

**Київський національний університет імені Тараса Шевченка**

**Економічний факультет  
Кафедра економічної кібернетики**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**ІННОВАЦІЙНЕ МОДЕЛЮВАННЯ БІЗНЕСУ З ВИКОРИСТАННЯМ  
ШТУЧНОГО ІНТЕЛЕКТУ**

студентки 4 курсу  
спеціальності 051 «Економіка»  
ОПП «Економічна кібернетика»  
денної форми навчання  
Ковбаси Анастасії Олександрівни

**Науковий керівник:**  
кандидат економічних наук, доцент  
Федоренко Ірина Костянтинівна

Засвідчую, що в цій роботі немає запозичень із  
праць інших авторів без відповідних посилань

Студент

Роботу допущено до захисту перед ЕК  
рішенням кафедри економічної кібернетики  
від 12 червня 2025 р., протокол № 15

Завідувач кафедри:  
доктор економічних наук, професор  
Ляшенко Олена Ігорівна

КИЇВ - 2025

## РЕФЕРАТ

Кваліфікаційна робота бакалавра містить: 126 ст., 20 рис., 54 джерела, додатки

Ключові слова: штучний інтелект, моделювання, інновації, бізнес-процеси, цифрова трансформація, економічна ефективність, чат-бот.

Об'єкт дослідження: процеси інноваційного моделювання бізнесу в умовах цифрової трансформації.

Мета дослідження: розробка теоретико-методичних засад та практичних рекомендацій щодо інноваційного моделювання бізнесу з використанням штучного інтелекту.

Методи дослідження: аналіз літератури, порівняльний аналіз, програмна реалізація (Python), економічне моделювання.

Наукова новизна, теоретична значимість дослідження: новизна роботи полягає у розробці доступної та економічно ефективної моделі впровадження ШІ для малого та середнього бізнесу в Україні, що враховує специфіку місцевого ринку та обмежені ресурси підприємств.

Практична цінність: можливість використання розроблених теоретико-методичних положень та практичних рекомендацій для впровадження ШІ-технологій у діяльність підприємств з метою підвищення їхньої інноваційної активності, оптимізації бізнес-процесів та формування конкурентних переваг.

## RESUME

Taras Shevchenko National University of Kyiv,

Faculty of Economics, Department of Economic Cybernetics

Key words: artificial intelligence, innovative modeling, business processes, digital transformation, economic efficiency.

The graduation research of student Anastasiia Kovbasa deals with innovative business modeling using artificial intelligence.

The work is interesting for entrepreneurs, business owners, and researchers in Ukraine seeking practical strategies for leveraging AI to enhance competitiveness and optimize business processes.

Pages 126, figures 20, bibliog. 54, append. available.

## ЗМІСТ

ВСТУП .....	4
РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ .....	6
1.1 <i>Поняття та сутність штучного інтелекту.</i> .....	6
1.2. <i>Роль штучного інтелекту в трансформації сучасних бізнес-моделей.</i> .....	8
1.3. <i>Застосування штучного інтелекту в різних галузях.</i> .....	11
1.4. <i>Переваги та недоліки використання штучного інтелекту.</i> .....	15
РОЗДІЛ 2. ОСНОВНІ КОНЦЕПЦІЇ ШТУЧНОГО ІНТЕЛЕКТУ, ЩО ВИКОРИСТОВУЮТЬСЯ В БІЗНЕСІ.....	20
2.1. <i>Особливості впровадження штучного інтелекту в бізнес-діяльність.</i> ....	20
2.2. <i>Аналіз основних тенденцій використання сервісів штучного інтелекту в бізнесі.</i> .....	23
2.3. <i>Аналіз найпопулярніших сервісів штучного інтелекту.</i> .....	26
РОЗДІЛ 3. ІННОВАЦІЙНЕ МОДЕЛЮВАННЯ БІЗНЕСУ .....	33
3.1. <i>Аналіз інноваційної бізнес-моделі компанії.</i> .....	33
3.2. <i>Оцінка ефективності штучного інтелекту для оптимізації бізнесу.</i> .....	36
3.3. <i>Пропозиції щодо впровадження штучного інтелекту для бізнесу в Україні.</i> .....	40
3.4. <i>Розробка рекомендацій та моделей.</i> .....	43
3.5. <i>Модель впровадження інноваційного ШІ-рішення.</i> .....	59
3.6. <i>Розрахунок потенційного економічного ефекту.</i> .....	64
ВИСНОВКИ.....	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	69
ДОДАТКИ .....	74

## Перелік умовних позначень і скорочень

ШІ - штучний інтелект

API - Application Programming Interface (інтерфейс прикладного програмування)

CRM - Customer Relationship Management (система управління відносинами з клієнтами)

ERP - Enterprise Resource Planning (система планування ресурсів підприємства)

## ВСТУП

Сучасний бізнес функціонує в умовах надзвичайної динаміки та конкуренції, що вимагає постійного пошуку інноваційних рішень для підвищення ефективності та адаптивності. Епоха цифрової трансформації, що характеризується стрімким розвитком інформаційних технологій та глобалізацією ринків, ставить перед підприємствами нові виклики, пов'язані з необхідністю швидкого реагування на зміни, оптимізації ресурсів та задоволення зростаючих потреб споживачів. У цьому контексті, штучний інтелект виступає не просто як технологічний тренд, а як ключовий каталізатор трансформаційних процесів, що докорінно переосмислюють традиційні бізнес-моделі.

В умовах стрімкої цифрової трансформації та глобальної конкуренції український бізнес стикається з необхідністю не просто впроваджувати інновації, а й робити це максимально ефективно. Штучний інтелект, який ще нещодавно здавався далеким від реального застосування у бізнесі, сьогодні стає одним із головних драйверів змін, здатних не лише оптимізувати процеси, а й відкрити нові ринкові можливості. Наукова новизна цієї роботи полягає у розробці адаптованої для українських реалій моделі впровадження ШІ, що враховує специфіку малого бізнесу, обмеженість ресурсів і потребу у швидкій віддачі інвестицій. Практична цінність - у створенні конкретного інструменту, який може бути використаний для реального підвищення ефективності, а також у розробці рекомендацій, які враховують не лише технічні, а й організаційні, кадрові й економічні аспекти впровадження. Структура роботи вибудована так, щоб послідовно - від теорії до практики - розкрити всі ключові етапи впровадження інноваційних рішень на основі ШІ для українського бізнесу.

Впровадження ШІ дозволяє компаніям не лише оптимізувати операційні процеси за рахунок автоматизації рутинних завдань, але й відкривати нові ринкові можливості завдяки здатності аналізувати величезні масиви даних, прогнозувати тенденції та ідентифікувати приховані закономірності. Персоналізація взаємодії зі споживачами, ефективніше управління ризиками, а

також підвищення якості прийняття управлінських рішень на основі глибоких аналітичних висновків стають досяжними завдяки інтеграції ШІ.

Актуальність теми дослідження обумовлена зростаючою потребою підприємств у розробці та впровадженні інноваційних стратегій, які враховують потенціал штучного інтелекту для підвищення ефективності діяльності та формування конкурентних переваг. Стрімкий розвиток технологій ШІ, їх інтеграція у різні сфери бізнесу, від виробництва та логістики до маркетингу та управління персоналом, робить дослідження інноваційного моделювання бізнесу з використанням штучного інтелекту надзвичайно важливим для забезпечення сталого розвитку та адаптивності компаній в умовах цифровізації економіки.

Об'єктом дослідження є процеси інноваційного моделювання бізнесу в умовах сучасних економічних трансформацій.

Предметом дослідження є сукупність теоретико-методичних та прикладних аспектів використання штучного інтелекту для формування та розвитку інноваційних бізнес-моделей.

Практична цінність роботи полягає у розробці конкретних рекомендацій та функціональної моделі ШІ-рішення, які можуть бути використані українськими підприємствами, зокрема малими та середніми, для оптимізації бізнес-процесів, підвищення рівня клієнтського сервісу та досягнення конкурентних переваг. Запропоновані підходи та розрахунки економічного ефекту надають практичне обґрунтування для інвестицій у технології штучного інтелекту.

## РОЗДІЛ 1. ТЕОРЕТИЧНІ АСПЕКТИ ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ

### *1.1 Поняття та сутність штучного інтелекту.*

Тема штучного інтелекту є однією з найгарячіших і найскладніших у сучасній науці. На сьогоднішній день він вийшов за межі лабораторій і став невід'ємною частиною нашого повсякденного життя, охоплюючи різноманітні сфери - від особистих пристроїв до складних виробничих процесів. Важливо зазначити, що, незважаючи на його значну популярність, серед науковців досі немає єдиного, загальноприйнятого визначення терміну «штучний інтелект». Це можна пояснити швидким розвитком цієї галузі, безперервним розширенням її можливостей та різними філософськими поглядами на природу самого інтелекту.

Вважається, що термін «штучний інтелект» вперше офіційно прозвучав у пропозиції до Дартмутського літнього дослідницького проекту в 1955 році, авторами якої були Джон Маккарті, Марвін Мінський, Натаніель Рочестер та Клод Шеннон. Хоча сама конференція відбулася в 1956 році, саме цей документ став основою для визначення ШІ як окремої галузі знань. Ця подія стала своєрідною точкою відліку для цілої наукової дисципліни, яка з того часу пройшла довгий і складний шлях розвитку, переживаючи періоди піднесення та розчарування. [1]

Щоб краще зрозуміти суть штучного інтелекту, корисно звернути увагу на класифікацію, запропоновану відомими дослідниками Стюартом Расселом і Пітером Норвігом. У своїй основоположній праці «Artificial Intelligence: A Modern Approach» вони ділять ШІ на чотири основні категорії відповідно до цілей його розробників, тож ми зупинимось на кожній з них.

Почнемо з систем, що мислять подібно до людей. Цей підхід спрямований на розробку комп'ютерних систем, котрі можуть виконувати когнітивні процеси, схожі на людське мислення, включаючи прийняття рішень, розв'язання проблем і навчання. Проте, як показує практика, повне відтворення людського мислення є надзвичайно складним завданням.

Говорячи про системи, що діють подібно до людей, то тут акцент робиться на поведінці системи. Якщо комп'ютерна програма може виконувати завдання, аналогічно людині, її можна вважати штучно інтелектуальною, навіть якщо її внутрішні процеси відрізняються. Яскравим прикладом є проходження тесту Тюрінга.

Системи, що мислять раціонально, цей підхід орієнтований на створення систем, які здатні логічно і послідовно мислити, дотримуючись законів логіки. Ідеалом тут є розробка формальних моделей мислення, які гарантують правильні висновки на основі заданих передумов.

Останніми розглянемо системи, що діють раціонально. Раціональна поведінка передбачає здатність системи досягати поставлених цілей, використовуючи наявну інформацію та ресурси способом, який є найбільш ефективним. Така система не обов'язково повинна мислити «по-людськи» - головне, щоб вона досягала бажаного результату. [2]

На мою думку, саме останній підхід, створення раціонально діючих систем, є найбільш затребуваним у практичному застосуванні ШІ сьогодні. Бізнес та промисловість зацікавлені перш за все в ефективних рішеннях, а не в імітації людського інтелекту як такого.

Оглядаючи різні погляди на визначення штучного інтелекту, можна виділити кілька основних характеристик, які є невід'ємними для цього поняття, а саме: здатність до навчання, логічних міркувань та прийняття рішень, розпізнавання образів, взаємодії природньою мовою, а також адаптивність.

Розглядаючи логічне мислення та прийняття рішень, слід підкреслити, що ШІ може обробляти великі обсяги даних, виявляти причинно-наслідкові зв'язки та на їх основі формувати обґрунтовані висновки. У сфері розпізнавання образів штучний інтелект здатен аналізувати візуальну, звукову та текстову інформацію, виявляючи закономірності та класифікуючи об'єкти. Яскравими прикладами цього є системи розпізнавання облич, голосові помічники та фільтри спаму. Технології обробки природної мови (NLP) дозволяють штучному інтелекту розуміти, інтерпретувати та генерувати людську мову. Це має велике значення

для чат-ботів, систем автоматичного перекладу, голосових інтерфейсів та цифрових асистентів. Однією з ключових переваг сучасних ШІ-систем є їхня адаптивність, тобто здатність змінювати поведінку на основі нової інформації чи змін у середовищі.

Важливим є також міждисциплінарний характер ШІ, що об'єднує знання з комп'ютерних наук, математики, статистики, психології та інших галузей. Застосування методів ШІ навіть у фундаментальних наукових дослідженнях, як це згадується у статті про Нобелівські премії 2024 року, підкреслює його універсальність та значущість для розвитку науки в цілому. [3]

В українському контексті, окрім наукових досліджень, важливу роль відіграє державна політика у сфері розвитку ШІ. Концепція розвитку штучного інтелекту в Україні визначає ШІ як комплексну сукупність інформаційних технологій, що використовуються для виконання складних завдань шляхом обробки інформації, створення баз знань та моделей прийняття рішень. [4]

Аналізуючи все вищезгадане, можна зробити висновок, що штучний інтелект є міждисциплінарною галуззю науки та технологій, що спрямована на розробку та створення комп'ютерних систем, здатних імітувати когнітивні функції людини, такі як навчання, розпізнавання образів, логічне мислення, прийняття рішень та обробка природної мови, з метою автономного виконання завдань та досягнення поставлених цілей.

### *1.2. Роль штучного інтелекту в трансформації сучасних бізнес-моделей.*

Сучасний світ бізнесу переживає значні зміни через швидкий прогрес та впровадження технологій штучного інтелекту, якщо ще декілька років тому ШІ сприймали як технологію майбутнього, то сьогодні він є рушійною силою для інновацій та трансформацій бізнес-моделей. Він проникає у всі сфери діяльності компаній, від стратегічного управління до покращення операційної ефективності та взаємодії з клієнтами.

Одним із найбільш помітних напрямів застосування ШІ на сьогодні є предиктивна аналітика, тобто прогнозований аналіз, на відміну від класичних

статистичних, що часто обмежуються аналізом лише наявних даних, ШІ дозволяє оцінювати не лише оцінювати минулі події, але і будувати досить точні прогнози майбутніх результатів. Так, у книзі Еріка Сігеля «Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die», зазначено, що сучасні алгоритми машинного навчання здатні працювати з величезними обсягами як структурованої, так і неструктурованої інформації, саме це дає змогу виявляти складні й, на перший погляд непомітні закономірності, які традиційними методами виявити практично неможливо. [5]

У сфері маркетингу такі можливості вже активно використовуються. Наприклад, компанія Adobe застосовує інструменти ШІ для прогнозування поведінки користувачів, що дозволяє точніше налаштовувати рекламні кампанії, краще розуміти потреби аудиторії та персоналізувати контент. Розглядаючи фінансовий сектор, слід згадати JPMorgan Chase, які використовують штучні інтелектуальні системи для аналізу ринкових тенденцій, оцінки кредитоспроможності клієнтів, а також виявлення потенційно шахрайських транзакцій. Такий підхід не тільки знижує фінансові ризики, але й дозволяє приймати більш обґрунтовані інвестиційні рішення. [6]

Окрім цього, доцільно зазначити, що ШІ відіграє значну роль в оптимізації операційної діяльності. Завдяки здатності швидко обробляти великі масиви інформації про виробництво, логістику, стан обладнання чи рівень запасів, ШІ допомагає компаніям не просто бачити ситуацію в реальному часі, а й заздалегідь реагувати на потенційні проблеми, що дає змогу уникати затримок, зменшувати витрати та прискорювати роботу. Наприклад, компанія Siemens активно впроваджує технології штучного інтелекту на своїх виробництвах для оптимізації процесів, прогнозування поломок обладнання та в результаті отримує вищу продуктивність і менші витрати. Можна також згадати компанію Maersk, яка використовує ШІ у своїй логістиці для оптимізації маршрутів, прогнозування затримок, управління ланцюгами постачання в реальному часі, тим самим значно покращуючи своє представлення на міжнародному ринку.

Штучний інтелект також змінює і підхід до роботи з клієнтами. У статті «The Impact of Conversational AI on Customer Experience» показано як ШІ допомагає компаніям зрозуміти індивідуальні потреби, вподобання, поведінку, настрої своєї цільової аудиторії, що, в свою чергу, дає можливість створювати персоналізовані пропозиції, а також забезпечувати клієнтську підтримку через чат-боти та віртуальних асистентів. Наприклад, Sephora використовує ШІ для підбору косметичних засобів за індивідуальними параметрами користувача, надаючи можливість віртуального “примірювання”, що суттєво покращує онлайн та офлайн досвід своїх клієнтів. [7]

Ще однією сферою застосування штучного інтелекту є боротьба з шахрайством, особливо у фінансовому секторі. На відміну від традиційних систем, що базуються на жорстких правилах, системи ШІ постійно навчаються та здатні аналізувати великі масиви даних про транзакції у реальному часі, виявляючи аномалії та підозрілу активність, що дозволяє вчасно реагувати на загрози. Згідно із дослідженнями The Global Treasurer, 73% фінансових організацій вже використовують ШІ для боротьби з шахрайством, а 23% планують впровадження найближчим часом. Так тайська компанія Krungthai Card PCL (КТС) використовує ШІ для оцінки ризиків транзакцій у мілісекундах, що дозволяє зменшити кількість несанкціонованих операцій, підвищити рівень безпеки та задоволеність клієнтів. [8]

Крім того, штучний інтелект сприяє створенню абсолютно нових бізнес-моделей, кидаючи виклик традиційним підходам і відкриваючи незвідані можливості. Завдяки унікальним аналітичним можливостям штучного інтелекту компанії мають змогу виявляти потреби ринку, які раніше залишалися непоміченими або недостатньо задоволеними, розробляти нові продукти та послуги, а також оптимізувати вже наявні пропозиції. Яскравим прикладом цієї тенденції є бізнес у сфері економіки спільного використання, наприклад, Airbnb та Uber, де складні алгоритми штучного інтелекту допомагають ефективно балансувати попит і пропозицію в режимі реального часу та уможливають динамічне ціноутворення. Також значний потенціал ШІ вже можна побачити у

секторі охорони здоров'я, розробляючи більш точні методи діагностики, створюючи персоналізовані плани лікування на основі генетичних даних та історії хвороби пацієнта, а шляхом аналізу великих обсягів біомедичних даних та прогнозування ефективності потенційних сполук, процес відкриття нових лікарських засобів прискорюється.

Українські компанії також слідують світовим трендам та впроваджують штучний інтелект в різних галузях. Наприклад Sense Bank застосовує ШІ для автоматизації підтримки клієнтів через чат-боти й віртуальних асистентів, а також для аналізу даних, прогнозування та виявлення шахрайства. В банку впроваджені біометричні технології на основі ШІ для швидкої та безпечної ідентифікації клієнтів, що дозволяє оптимізувати внутрішні процеси й автоматизувати рутинні завдання. Ще одним яскравим прикладом є компанія Grammarly, яка створила платформу, що допомагає покращувати якість англійських текстів, використовуючи штучний інтелект для аналізу стилю, граматики та тону. Український онлайн-сервіс Rozetka впровадив системи для прогнозування попиту і оптимізації доставлення, які допомагають ефективніше планувати запаси та знижувати витрати на логістику. [9]

Таким чином, можна з упевненістю стверджувати, що штучний інтелект став одним із ключових інструментів трансформування бізнес-моделей, який охоплює безліч процесів: від прогнозування продажів до взаємодії з клієнтами. Світовий та український досвід його використання показує, що ця технологія надає компаніям можливості для підвищення ефективності та відкриття нових горизонтів.

### *1.3. Застосування штучного інтелекту в різних галузях.*

Сьогодні штучний інтелект є не лише теоретичним концептом, а й активно застосовується на практиці, демонструючи свої можливості у вирішенні складних завдань, оптимізації процесів та створенні нових можливостей у різних галузях людської діяльності. Тому розгляд конкретних прикладів впровадження ШІ є важливим для кращого розуміння його потенціалу та впливу на економіку та суспільство в цілому.

Говорячи про сучасну промисловість, то все більш важливою стає концепція Промисловість 5.0, на відміну від попередньої концепції, вона зосереджується на гармонійному співіснуванні людини та інтелектуальних технологій, підвищеній стійкості, екологічної відповідальності.

Одним із ключових напрямків застосування ІІІ у промисловості є предиктивне обслуговування обладнання (predictive maintenance). Завдяки датчикам інтернету речей (ІоТ), які збирають дані про стан обладнання в режимі реального часу, та алгоритмам машинного навчання, підприємства можуть прогнозувати можливі поломки ще до їх виникнення, що дозволяє завчасно планувати технічне обслуговування, мінімізувати простой та витрати на аварійний ремонт. Дослідження Deloitte показують, що впровадження таких систем дозволяє знизити витрати на обслуговування до 40% і підвищити надійність обладнання на 30-50%. [10]

Оптимізація виробничих процесів є ще одним важливим напрямом, завдяки системам ІІІ є можливість аналізувати значні обсяги даних з різних етапів виробництва, виявляти вразливі місця, підвищувати продуктивність і скорочувати кількість відходів. Контроль якості продукції суттєво покращується завдяки використанню комп'ютерного зору та алгоритмів для аналізу зображень, такі системи можуть автоматично виявляти дефекти на виробничих лініях із точністю, яка перевершує людські можливості, що допомагає зменшити кількість бракованої продукції та витрати на її виправлення. Окрім цього, у рамках концепції Промисловість 5.0 особлива увага приділяється сталому розвитку та енергоефективності. Системи штучного інтелекту аналізують, скільки енергії споживається, і оптимізують роботу обладнання, що допомагає знизити витрати на енергію та зменшити викиди вуглекислого газу. [11]

У фінансовому секторі ІІІ також демонструє значний вплив. Завдяки можливості обробляти великі обсяги даних, фінансові установи можуть швидше приймати рішення, зменшувати витрати та підвищувати безпеку. Так JPMorgan Chase використовує COiN, програму на основі ІІІ, для аналізу фінансових та юридичних документів, тим самим пришвидшуючи процес та мінімізуючи

людські помилки. Також алгоритми машинного навчання здатні аналізувати транзакції, виявляти певні аномалії, що можуть свідчити про шахрайські дії. Доречно також додати, що штучний інтелект дозволяє оцінювати кредитні ризики, оскільки він може аналізувати не лише традиційні фінансові показники, а й інші дані про клієнтів, наприклад дії у соціальних мережах або історії покупок. Окрім цього чат-боти та віртуальні асистенти надають цілодобову клієнтську підтримку без залучення людини. Впровадження ШІ також дозволяє фінансовим установам ефективніше дотримуватися регуляторних вимог, автоматизуючи процеси моніторингу та звітності. [13]

Значною мірою штучний інтелект впливає й на сучасну медицину, він змінює підходи як до діагностики, так і до лікування пацієнтів. Системи на базі ШІ допомагають ранньому діагностуванню захворювань аналізуючи медичні зображення, такі як рентгенівські знімки, МРТ чи КТ з точністю, яка в деяких випадках може перевищувати можливості досвідчених лікарів. Дослідження, опубліковане в *Nature Medicine*, підтверджує, що такі системи суттєво підвищують шанси на ранню діагностику раку молочної залози, що напряду впливає на успішність лікування. Також доцільно буде згадати про персоналізовану медицину, де завдяки штучному інтелекту лікарі можуть враховувати не лише звичайні клінічні показники, але й генетичну інформацію, спосіб життя та інші індивідуальні характеристики пацієнта, що допомагає спрогнозувати, як пацієнт реагуватиме на певні методи лікування, а також призначити найефективніше лікування. Звісно, не можна оминати увагою внесок ШІ у створення нових ліків. Оцінюючи складні біологічні процеси та прогножуючи ефективність потенційних препаратів, штучний інтелект може значно прискорити цю процедуру, яка зазвичай є тривалою і дорогою. [14-15]

З розвитком електронної комерції та зростання конкуренції у сфері роздрібної торгівлі, штучний інтелект стає одним із важливих інструментів для підвищення ефективності у цьому бізнесі. Одне з найпоширеніших застосувань штучного інтелекту в електронній комерції - персоналізація, що допомагає створювати індивідуальні пропозиції для покупців. Алгоритми машинного

навчання вивчають інформацію про попередні покупки, перегляди товарів та поведінку на сайті, щоб пропонувати покупки, які найкраще зможуть задовільнити потреби користувачів. Окрім цього алгоритми аналізують історичні дані про продажі, сезонні тенденції та зовнішні фактори, такі як погода чи економічні зміни, щоб передбачити, які товари будуть популярними. Наприклад, американська мережа Walmart використовує ШІ для прогнозування попиту, що дозволяє зменшити надлишки товарів на складах, оптимізувати логістику, знизити витрати та підвищити ефективність. Також ШІ допомагає покращувати ціноутворення, беручи до уваги попит, конкуренцію та звички покупців. Алгоритми динамічного ціноутворення дозволяють змінювати ціни в режимі реального часу, щоб отримати максимальний прибуток. [17]

Як й інші галузі, логістика також зазнала змін під впливом штучного інтелекту. Сучасні технології аналізують такі фактори як трафік, погода, стан доріг, щоб прокладати найкоротші та найзручніші маршрути. До того ж, ШІ широко використовується для автоматизації складів і управління запасами. Завдяки цьому компанії можуть прогнозувати попит, оптимізувати рівень запасів та автоматично розподіляти товари між складами, а використання роботизованих систем і технологій машинного навчання допомагає знизити кількість помилок при комплектуванні замовлень і пришвидшити обробку вантажів. В Україні такі технології вже застосовує компанія «Нова Пошта», яка активно впроваджує ШІ для сортування посилок і оптимізації маршрутів доставки. Завдяки цьому вона може швидше обробляти великі обсяги відправлень, підвищувати точність доставки та оперативно реагувати на зміни в логістиці, що особливо важливо в умовах зростаючого попиту та складної інфраструктури. [9; 18]

Штучний інтелект швидко проникає і в освіту, змінюючи способи навчання, викладання та управління освітніми процесами. Він дозволяє створювати індивідуальні навчальні плани, адаптуючи матеріали до рівня знань та потреб кожного студента, алгоритми аналізують прогрес учнів, щоб пропонувати завдання відповідної складності або додаткові ресурси. Також ШІ допомагає автоматизувати процес оцінювання, системи на його основі можуть перевіряти

тести, есе й навіть творчі роботи, аналізуючи текст чи відповіді. Окрім цього, ШІ використовується для аналізу даних про студентів, викладачів і навчальні програми, щоб оптимізувати управління освітою. Алгоритми можуть виявляти студентів, які ризикують відстати, або пропонувати зміни в навчальних планах. Не варто забувати і про чат-боти, які можуть відповідати на питання студентів або допомагати викладачам із плануванням. Наприклад UNICEF розробив доступні цифрові підручники з функціями, які адаптують контент для дітей з інвалідністю, забезпечуючи інклюзивність, а у Бразилії програма Letrus використовує ШІ для аналізу письмових робіт учнів, надаючи миттєвий зворотний зв'язок і покращуючи рівень грамотності. [19-21]

Сільське господарство є однією з ключових сфер економіки, яка стикається з різними викликами, такими як зміни клімату, зростання населення і потреба в раціональному використанні ресурсів. Штучний інтелект відіграє важливу роль у вирішенні цих проблем, надаючи сучасні інструменти для аналізу даних, автоматизації процесів і підвищення врожайності. Точне землеробство активно використовує штучний інтелект для аналізу даних, отриманих зі супутників, дронів і датчиків, щоб оптимізувати процеси посіву, поливу та внесення добрив. Алгоритми машинного навчання допомагають фермерам розпізнавати ділянки, які потребують особливої уваги, а також виявляти хвороби рослин і шкідників на ранніх етапах через комп'ютерний зір та аналіз зображень. Штучний інтелект також використовують для прогнозування врожайності, беручи до уваги дані про погоду, стан ґрунту та історичні показники, що дозволяє фермерам більш ефективно планувати посіви та використовувати ресурси. Крім того, вже існують системи, які керують роботами та автоматизованими системами для виконання таких завдань, як посів, збір врожаю та догляд за тваринами. [22]

#### *1.4. Переваги та недоліки використання штучного інтелекту.*

Перш за все, важливо підкреслити, що однією із найсуттєвіших переваг ШІ є підвищення ефективності та продуктивності бізнес-процесів. Автоматизація стандартних і повторюваних операцій, таких як обробка даних, управління запасами чи планування, дозволяє значно скоротити час виконання завдань і

зменшити вплив людського фактору. Наприклад, у виробничих компаніях системи прогнозованого обслуговування на базі ШІ допомагають вчасно виявляти потенційні несправності обладнання, що запобігає дорогим простоям. Таким чином, підприємства можуть оптимізувати свої ресурси і зосередити увагу на стратегічних питаннях, що вимагають творчого підходу та людського досвіду.

Крім того, ШІ значно прискорює процес прийняття рішень. Сучасні алгоритми здатні аналізувати величезні обсяги даних у реальному часі, виявляючи приховані закономірності, тренди та потенційні ризики, що дає змогу керівникам отримувати глибокі аналітичні інсайти, що базуються на фактах, а не лише на інтуїції. Завдяки цьому бізнес може своєчасно реагувати на зміни ринку, адаптувати стратегії і знижувати рівень невизначеності. Водночас, варто підкреслити, що якість прийнятих рішень безпосередньо залежить від якості даних і правильності налаштувань моделей ШІ.

Ще одним важливим аспектом є покращення взаємодії з клієнтами. Системи на основі ШІ, такі як чат-боти, голосові помічники і персоналізовані рекомендації, забезпечують індивідуальний підхід до кожного користувача. Вони здатні працювати безперервно, що є особливо цінним для міжнародних компаній із клієнтами в різних часових поясах. Завдяки цьому підвищується рівень задоволеності споживачів, зростає лояльність і, відповідно, збільшуються продажі. Проте, доречно зазначити, що повна заміна людського спілкування ШІ не може забезпечити, особливо у випадках, коли потрібна емпатія або складні переговори.

Оптимізація витрат є ще одним аргументом на користь впровадження ШІ. Автоматизація дозволяє скоротити витрати на оплату праці за рахунок зменшення потреби у виконанні рутинних завдань, а також знижує ризик помилок, які можуть призводити до фінансових втрат. Це добре помітно у виробничих галузях, де ШІ-системи прогнозують технічні несправності обладнання, що дозволяє планувати профілактичне обслуговування і уникати дорогих простоїв і ремонтів, що позитивно впливає на бюджет підприємства.

Що стосується розвитку нових бізнес-моделей, ШІ відкриває безпрецедентні можливості для інновацій. Наприклад, у сфері електронної комерції системи рекомендацій на основі машинного навчання дозволяють пропонувати клієнтам товари, які максимально відповідають їхнім уподобанням, що суттєво підвищує конверсію. Крім того, фінансові технології використовують ШІ для автоматизованого аналізу ризиків і персоналізованого консультування. Завдяки цьому компанії можуть швидше адаптуватися до змін ринку і створювати унікальні пропозиції, що зміцнюють їх конкурентні позиції. [24]

Однак, незважаючи на численні переваги, застосування ШІ в бізнесі супроводжується низкою викликів і недоліків. По-перше, високі початкові витрати на впровадження технологій є серйозною перешкодою, особливо для підприємств із обмеженим бюджетом. Вартість не обмежується лише придбанням обладнання і програмного забезпечення, а включає також навчання персоналу, адаптацію бізнес-процесів і подальшу підтримку систем.

По-друге, автоматизація на базі ШІ може призводити до скорочення робочих місць, особливо у сферах, де домінує рутинна праця. Це викликає соціальні та етичні питання, пов'язані з безробіттям і необхідністю перепідготовки кадрів. Відповідно, компаніям доводиться балансувати між економічною вигодою і соціальною відповідальністю, що вимагає розробки стратегій підтримки працівників.

Ще одним суттєвим викликом є етичні та правові аспекти використання ШІ. Алгоритми можуть містити упередження, що відображаються у прийнятті дискримінаційних рішень, а також існує ризик порушення конфіденційності даних клієнтів. Відсутність прозорості у роботі складних моделей ШІ ускладнює визначення відповідальності за помилки або несправедливі дії. Це ставить бізнес перед необхідністю розробки чітких політик і стандартів етичного використання ШІ.

Додатково, слід зазначити, що ШІ не володіє емоційним інтелектом, який є критично важливим у багатьох сферах бізнесу, особливо в обслуговуванні клієнтів і управлінні персоналом. Людська інтуїція, креативність і здатність до

емпатії залишаються незамінними, що означає необхідність інтеграції ШІ як допоміжного інструменту, а не повної заміни людини.

Крім того, залежність від технологій створює ризики, пов'язані з кібербезпекою і стабільністю роботи систем. Збої або атаки на інфраструктуру ШІ можуть призвести до значних збитків і порушення бізнес-процесів. Тому питання захисту даних і надійності систем є пріоритетними у стратегіях впровадження ШІ. Також існує проблема дефіциту кваліфікованих фахівців, здатних розробляти, впроваджувати і підтримувати ШІ-рішення. Це уповільнює темпи цифрової трансформації і вимагає активних інвестицій у освіту і підвищення кваліфікації. [25]

Отже, застосування штучного інтелекту у бізнесі є багатогранним явищем, що відкриває широкі перспективи для розвитку і оптимізації, але одночасно ставить перед компаніями складні виклики. Лише комплексний і продуманий підхід, який враховує як технологічні, так і соціальні аспекти, дозволить максимально ефективно інтегрувати ШІ у бізнес-процеси і забезпечити довгостроковий успіх.

У першому розділі було здійснено ґрунтовний аналіз теоретичних засад інноваційного моделювання бізнесу та визначено сутність і місце штучного інтелекту в сучасній економіці. Зокрема, систематизовано ключові характеристики ШІ як міждисциплінарної галузі, що здатна до імітації когнітивних функцій людини для автономного виконання завдань. У цьому контексті було розкрито трансформаційну роль ШІ у формуванні нових бізнес-моделей, зокрема через можливості предиктивної аналітики, оптимізації операційної діяльності, персоналізації взаємодії з клієнтами та ефективної боротьби з шахрайством.

Разом з тим, розглянуто переваги впровадження ШІ, такі як підвищення загальної ефективності та прискорення прийняття рішень, а також окреслено потенційні недоліки, серед яких високі початкові інвестиції та етичні виклики. Отже, даний розділ заклав міцний теоретичний фундамент для розуміння ключових аспектів інтеграції штучного інтелекту у бізнес-процеси, що, у свою

чергу, дозволяє перейти до більш детального розгляду практичних аспектів його застосування.

## РОЗДІЛ 2. ОСНОВНІ КОНЦЕПЦІЇ ШТУЧНОГО ІНТЕЛЕКТУ, ЩО ВИКОРИСТОВУЮТЬСЯ В БІЗНЕСІ

### *2.1. Особливості впровадження штучного інтелекту в бізнес-діяльність.*

Як вже було визначено в попередньому розділі, впровадження штучного інтелекту в бізнесі стає не просто трендом, а необхідністю для підтримання конкурентоспроможності на сучасному ринку. Проте необхідно розуміти, що цей процес вимагає системного підходу, ретельного планування та зусиль різних підрозділів компанії, оскільки не всі проекти можуть досягнути очікуваного результату.

Будь-яке впровадження ШІ розпочинається з глибокого аналізу потреб підприємства. На даному етапі керівництво та ключові відділи мають дійти до конкретизації бізнес-завдань, які можуть бути вирішені за допомогою ШІ: це найчастіше ними є автоматизація рутинних операцій, підвищення точності прогнозування, персоналізація клієнтських пропозицій або оптимізація логістичних процесів. Слід акцентувати, що успішність даного етапу залежить від залучення представників різних департаментів (маркетингу, фінансів, IT, HR), адже лише комплексний підхід дозволяє врахувати всі нюанси бізнес-процесів і уникнути хибних очікувань. Український досвід свідчить про ефективність такого підходу, так, ПриватБанк розпочав впровадження ШІ з аналізу клієнтських даних для створення персоналізованих пропозицій у програмі «Pryvit», що дозволило підвищити лояльність клієнтів. [26; 27]

Наступним кроком є підготовка даних, який часто виявляється найбільш трудомістким і критично важливим. Якість, повнота й актуальність даних мають прямий вплив на ефективність майбутньої ШІ-системи. Підприємство має організувати збір, очищення, структурування та захист даних, забезпечити їхню відповідність вимогам законодавства щодо приватності, наприклад, GDPR. В Україні цей процес частково полегшується завдяки ініціативам Мінцифри, які сприяють розвитку відкритих даних. Водночас, навіть великі компанії, як

ПриватБанк, стикаються з викликами щодо якості даних, особливо коли йдеться про інтеграцію різнорідних джерел інформації.

Далі слідує етап вибору оптимального ІІІ-рішення на основі визначених потреб та наявних даних. Компанія має вирішити, чи використовувати готові рішення, доступні на ринку, чи розробляти власні моделі. Вибір залежить від низки факторів, включаючи бюджет, наявні технічні можливості, специфіку бізнес-процесів та конфіденційність даних. На цьому етапі важливо залучити фахівців із машинного навчання, дата-саєнтистів, а також бізнес-аналітиків, які оцінюють економічну доцільність та рентабельність інвестицій. Міжнародний досвід компаній, таких як Commonwealth Bank of Australia, показує, що поєднання внутрішніх розробок і зовнішніх рішень дозволяє швидко масштабувати АІ-проекти і зменшувати ризики.

Після вибору рішення необхідно здійснити належні інфраструктурні налаштування, вони передбачають підготовку апаратного забезпечення, мережевого середовища та платформ для розгортання ІІІ. Залежно від обраного рішення, це може бути локальний дата-центр, хмарні сервіси або гібридні варіанти. Важливо, щоб інфраструктура забезпечувала надійність, масштабованість і безпеку, а також підтримувала інтеграцію з існуючими інформаційними системами підприємства.

Інтеграція з наявними системами - ще один виклик, який часто недооцінюють на початку проекту. ІІІ-рішення мають належним чином взаємодіяти з CRM, ERP, ВІ-системами та іншими корпоративними платформами. Для цього використовуються АРІ, middleware та інші інструменти, що забезпечують обмін даними в режимі реального часу. Важливо також враховувати питання сумісності, безпеки і масштабованості інтеграції, адже навіть незначні технічні збої можуть призвести до втрати даних чи порушення бізнес-процесів. [28]

Навчання персоналу є обов'язковою умовою успішного впровадження ІІІ. Працівники мають розуміти не лише базові принципи роботи нової системи, але й її можливості, обмеження, а також правила безпеки та етики. Особливо це

стосується тих, хто безпосередньо взаємодіє з ІІІ-рішеннями - аналітиків, операторів, менеджерів.

Запуск і моніторинг результатів є завершальним, але не менш важливим етапом. Після впровадження системи необхідно постійно оцінювати її ефективність за допомогою ключових показників таких як точність прогнозів, рівень автоматизації чи задоволеність клієнтів. Моніторинг дозволяє вчасно виявляти проблеми, оновлювати моделі та адаптувати рішення до змін у бізнес-середовищі. Наприклад, ПриватБанк відстежує активність користувачів у програмі «Privity» і коригує алгоритми персоналізації відповідно до отриманих даних.

Організаційні, кадрові та технічні особливості впровадження ІІІ тісно пов'язані між собою. Зміна корпоративної культури та подолання опору змінам є ключовими організаційними викликами. У багатьох українських компаніях працівники побоюються втрати роботи через автоматизацію, тому важливо забезпечити комунікацію переваг і створити стратегії підтримки персоналу. Кадрові виклики пов'язані з дефіцитом фахівців, що вимагає інвестицій у навчання та залучення зовнішніх експертів. Технічні аспекти включають забезпечення надійності, безпеки даних, масштабованості інфраструктури та сумісності з існуючими системами.

Важливо пам'ятати, що кожен відділ підприємства відіграє свою унікальну роль і вимагає специфічних знань та навичок, необхідних для впровадження ІІІ. Таким чином, ІТ-відділ повинен розуміти архітектуру систем, АРІ, моделі машинного навчання, забезпечувати кібербезпеку й підтримку хмарних платформ. Дата-саєнтисти займаються підготовкою та аналізом даних, розробкою і тестуванням моделей. Бізнес-аналітики відповідають за формулювання вимог, оцінку економічної доцільності, аналіз рентабельності і ризиків. HR-відділ організує навчання, долає опір змінам, розробляє плани перекваліфікації. Юридичний відділ забезпечує відповідність обробки даних законодавству й етичним стандартам.

Водночас, впровадження ІІІ не завжди є ефективним. Однією з основних проблем є недостатня якість даних: неточні чи упереджені дані призводять до помилкових результатів, як це сталося у випадку з американською компанією Zillow, яка була змушена припинити програму купівлі будинків через неякісні прогнози цін. Недооцінка складності інтеграції, опір змінам серед співробітників, а також недостатній аналіз економічної доцільності - все це може звести нанівець очікувані переваги. Українські стартапи, такі як Reface, також стикалися з викликами масштабування та адаптації технологій до різних ринків, але змогли подолати ці труднощі завдяки використанню хмарних рішень і гнучкому підходу до розвитку. [29; 30]

## *2.2. Аналіз основних тенденцій використання сервісів штучного інтелекту в бізнесі.*

Останнє десятиліття стало переломним для використання штучного інтелекту у бізнесі, змінивши підходи до автоматизації, аналізу даних і створення нових продуктів. Якщо у 2015 році лише 10% організацій використовували або планували впроваджувати ІІІ, то, за даними Gartner, цей показник зріс до 37% у 2019 році, а за останніми звітами McKinsey у 2024 році вже досяг 78%, що свідчить про глибоку трансформацію бізнес-процесів. [31; 32]

Спочатку компанії покладалися на традиційні алгоритми машинного навчання, такі як лінійна регресія, дерева рішень і кластеризація, для прогнозування продажів, сегментації клієнтів і оцінки ризиків. З появою глибокого навчання, зокрема конволюційних нейронних мереж (CNN) і рекурентних нейронних мереж (RNN), стали можливими складніші задачі, як-от розпізнавання зображень і обробка природної мови. Трансформери, такі як BERT (Bidirectional Encoder Representations from Transformers) і GPT (Generative Pre-trained Transformer), революціонізували обробку тексту, сприяючи розвитку чат-ботів, віртуальних асистентів і автоматизації контенту. Разом із цим, зросла доступність ІІІ через платформи AI-as-a-Service (AIaaS) і low-code/no-code інструменти, що дозволило навіть малим підприємствам використовувати ІІІ без значних інвестицій. [33]

Однією з найпомітніших тенденцій є поширення генеративного ШІ, представленого такими інструментами, як ChatGPT, Claude і Gemini. Ці технології дозволяють автоматизувати створення текстового контенту, розробку маркетингових матеріалів, написання коду та навіть генерацію дизайнерських рішень. Наприклад, компанії в маркетингу використовують генеративний ШІ для створення персоналізованих рекламних кампаній, що значно підвищує конверсію. За даними McKinsey у 2024 році, 71% організацій застосовують генеративний ШІ у щонайменше одній бізнес-функції, водночас лише 1% компаній досягли зрілості в його використанні, що вказує на виклики масштабування. Наприклад, створення якісних даних для тренування моделей залишається проблемою, адже неякісні дані можуть призвести до упереджених або неточних результатів. [32]

Ще однією ключовою тенденцією є AI-as-a-Service (AIaaS), яка робить ШІ доступним для бізнесів будь-якого розміру. Платформи, такі як AWS, Azure AI та Google Cloud AI, надають готові рішення для машинного навчання, обробки природної мови та комп'ютерного зору, усуваючи потребу в дорогій інфраструктурі чи спеціалізованих кадрах. Згідно зі звітом Grand View Research, ринок AIaaS у 2024 році оцінювався в 16,08 млрд доларів із прогнозованим зростанням на 36,1% до 2030 року. Це дозволяє малим і середнім підприємствам конкурувати з великими корпораціями. Наприклад, невелика роздрібна компанія може використовувати AIaaS для аналізу клієнтських даних і створення персоналізованих пропозицій без значних витрат. Водночас компанії стикаються з викликами, такими як інтеграція цих сервісів із застарілими системами, що вимагає ретельного планування. [34; 35]

Перехід від експериментальних пілотних проектів до повної інтеграції ШІ в бізнес-процеси є ще однією з важливих тенденцією. Раніше компанії тестували ШІ в окремих функціях, але у 2025 році вони дедалі частіше вбудовують його в операційну діяльність. Наприклад, логістична компанія UPS використовує ШІ для оптимізації маршрутів доставки, що, за даними Harvard Data Science Review, дозволяє заощаджувати сотні мільйонів доларів щорічно. Дослідження вказують,

що значна частина великих підприємств інтегрувала ІІІ у щонайменше три бізнес-функції, такі як маркетинг, фінанси та логістика. Цей перехід вимагає перебудови робочих процесів і навчання персоналу, що може викликати опір серед працівників, проте компанії, які успішно інтегрують ІІІ, отримують значні конкурентні переваги. [32, 36]

Полегшення доступу до ІІІ через low-code та no-code платформи є ще одним трендом. Ці платформи дозволяють користувачам без глибоких технічних знань створювати ІІІ-додатки, залучаючи так званих «громадянських розробників». За прогнозами Gartner, low-code розвиток до 2024 року вже відповідає за понад 65% діяльності з розробки додатків. Платформи, як-от Bubble та Appian, дозволяють створювати інтелектуальні додатки, наприклад, чат-боти для клієнтської підтримки, без написання коду. Це знижує бар'єри для впровадження ІІІ, особливо для малих та середніх бізнесів. Наприклад, невелика маркетингова агенція може розробити власний інструмент для аналізу даних клієнтів, використовуючи no-code платформу, що економить час і ресурси, але компанії повинні забезпечити якість даних і відповідність регуляторним вимогам при використанні таких платформ. [34]

З поширенням ІІІ-технологій у бізнесі зростає увага до етичних аспектів їх використання та регуляторних механізмів. У 2025 році уряди та міжнародні організації посилюють зусилля для створення стандартів відповідального використання ІІІ. Наприклад, Європейський Союз прийняв AI Act, який встановлює рамки для розробки та використання ІІІ з урахуванням етичних норм, таких як прозорість і захист даних, із початком дії окремих положень із лютого 2025 року. Принципи OECD AI також закликають до людиноцентричного підходу до ІІІ. Бізнеси, своєю чергою, створюють ради з етики ІІІ та впроваджують політики прозорості, щоб уникнути упередженості та захистити конфіденційність даних. Наприклад, компанії в фінансовому секторі використовують інструменти для моніторингу ІІІ, щоб запобігти дискримінаційним рішенням у кредитуванні. Ця тенденція вимагає від компаній адаптації до складного регуляторного ландшафту. [38]

Варто згадати і про формування гібридних команд, де ШІ та люди співпрацюють, є ще однією ключовою тенденцією. ШІ-агенти, такі як Microsoft 365 Copilot, автоматизують рутинні завдання, дозволяючи працівникам зосередитися на стратегічних цілях. Наприклад, у кол-центрах ШІ-агенти обробляють запити клієнтів, тоді як люди вирішують складні питання, що створює нові можливості для підвищення продуктивності, але вимагає перекваліфікації працівників і адаптації корпоративної культури.

Отже, аналізуючи тенденції використання сервісів ШІ в бізнесі у 2025 році можна стверджувати, що технологія стає невід'ємною частиною бізнес-стратегій. Генеративний ШІ, AIaaS, інтеграція в процеси, демократизація через low-code/no-code платформи, етичні аспекти та гібридні команди формують майбутнє бізнесу. Компанії, які адаптуються до цих трендів, отримують конкурентні переваги, але повинні подолати виклики, пов'язані з даними, регуляцією та організаційними змінами.

### *2.3. Аналіз найпопулярніших сервісів штучного інтелекту.*

Стрімкий розвиток штучного інтелекту трансформував бізнес-операції, дозволивши компаніям автоматизувати процеси, аналізувати величезні масиви даних і створювати інноваційні продукти. За останні кілька років деякі сервіси штучного інтелекту стали лідерами в бізнес-середовищі, пропонуючи рішення для різноманітних завдань, тому буде доцільним детальніше познайомитись з ними.

Одним із найвідоміших сервісів штучного інтелекту є ChatGPT, розроблений компанією OpenAI, який став універсальним інструментом для автоматизації текстових завдань у бізнесі. Його основним призначенням є генерація тексту для таких застосувань, як підтримка клієнтів, створення контенту і навіть генерація коду. Наприклад, Octopus Energy інтегрувала ChatGPT у свою службу підтримки, обробляючи 44% запитів клієнтів, виконуючи роботу, яку раніше робили 250 співробітників. Ба більше, клієнти залишаються задоволені спілкуванням із ботом навіть більше, ніж із живими операторами. А компанія Freshworks застосовує ChatGPT для прискорення розробки програмного забезпечення,

скоротивши час створення складних додатків із 8-10 тижнів до менш ніж одного тижня. [39]

ChatGPT можна інтегрувати у різні бізнес-процеси через REST API - це дозволяє підключати його до CRM-систем, месенджерів, чат-ботів чи будь-яких інших платформ. Крім того, ним можна користуватися через веб-інтерфейс або мобільні додатки. Сервіс підтримує збереження історії розмов, а також дозволяє налаштовувати запити (так званий *prompt engineering*) та адаптувати відповіді під стиль і потреби конкретного бізнесу. В основі ChatGPT лежить сучасна архітектура GPT (Generative Pre-trained Transformer). Завдяки механізму самоуваги (*self-attention*) модель розуміє контекст і може генерувати логічні, зв'язні відповіді. Для навчання таких моделей використовують величезні обсяги текстових даних, що дозволяє їм працювати з різними тематиками й стилями. [40]

Наступним розглянемо Microsoft Azure AI, який вирізняється універсальністю та глибокою інтеграцією з іншими продуктами Microsoft, що робить його привабливим для компаній, які вже використовують цю екосистему. Його застосовують для створення рекомендаційних систем, автоматизації рутинних процесів і розробки інтелектуальних додатків у фінансах, електронній комерції та охороні здоров'я. Наприклад, у фінансовому секторі Azure AI допомагає аналізувати транзакції в реальному часі, виявляючи підозрілі операції, для цього використовуються трансформери - моделі, які «розуміють» контекст тексту. А от для роботи зі зображеннями (наприклад, у медицині для аналізу рентгенівських знімків) застосовують конволюційні нейромережі (CNN), які вміють витягувати з картинок шари ознак - від простих ліній до складних об'єктів.

Для тих, хто не хоче заглиблюватися в код, Azure пропонує Cognitive Services, тобто набір готових API для розпізнавання мови, аналізу текстів або навіть створення віртуальних помічників. Наприклад, можна зібрати чат-бота для підтримки клієнтів, який розумітиме запити українською та адаптуватиметься до стилю бренду. У разі якщо потрібна кастомна модель, Azure Machine Learning

дозволяє тренувати її на власних даних, завдяки AutoML система сама підбере оптимальні алгоритми. [41]

Ще одним популярним сервісом є Google Cloud AI, який надає широкий спектр інструментів для роботи зі штучним інтелектом і аналізом даних. Особливо корисним він є для компаній, які мають справу з великими обсягами інформації та прагнуть автоматизувати процеси, пов'язані з аналітикою, обробкою природної мови або зображень. Серед основних напрямків застосування Google Cloud AI можна виділити прогнозування ринкових тенденцій, аналіз настроїв клієнтів і автоматизацію обробки документів.

Структурно платформа складається з кількох ключових модулів, так Vertex AI дає змогу створювати й розгортати власні моделі машинного навчання, а AutoML дозволяє автоматизувати підбір оптимальних алгоритмів і гіперпараметрів, що особливо зручно для користувачів без глибоких знань у сфері програмування. Для обробки зображень використовується Vision AI, а для аналізу тексту - Natural Language API. Усі ці сервіси працюють на хмарній інфраструктурі Google, що забезпечує гнучкість, масштабованість і високу продуктивність. Додатково Google Cloud AI використовує спеціалізовані процесори TPU сьомого покоління (Ironwood), які значно пришвидшують обчислення при роботі з великими моделями штучного інтелекту. З технічної точки зору, для задач обробки природної мови тут застосовуються трансформери такі як BERT і Gemini. Ці моделі здатні аналізувати контекст тексту, що дає змогу краще розуміти значення слів у різних ситуаціях і досягати високої точності при аналізі настроїв або автоматичній класифікації текстів. Для роботи із зображеннями Google Cloud AI використовує конволюційні нейронні мережі, які виділяють ознаки на різних рівнях складності - від простих контурів до складних об'єктів. AutoML, у свою чергу, автоматизує процес вибору моделей і налаштування параметрів, що значно спрощує запуск ШІ-проектів навіть для невеликих команд. [42]

Тепер перейдемо до AWS AI/ML Services від Amazon, що відомі своєю масштабованістю та гнучкістю, що робить їх привабливими для бізнесів у

роздрібній торгівлі, фінансах і логістиці. Центральним елементом тут є Amazon SageMaker, який фактично став стандартом для створення, тренування та розгортання моделей машинного навчання на AWS. SageMaker охоплює повний цикл роботи з ML: від підготовки даних до запуску моделі, і все це доступно через хмарну інфраструктуру AWS з інтеграцією через REST API та SDK. Також SageMaker підтримує автоматичне налаштування гіперпараметрів, розподілене тренування на GPU-інстансах, а також має такі модулі, як Studio для розробки, Autopilot для автоматизації побудови моделей, Ground Truth для маркування даних, Clarify для виявлення упередженості й Model Monitor для моніторингу моделей у процесі створення. [43]

Окремо варто згадати Amazon Rekognition - сервіс для аналізу зображень і відео. Він дозволяє автоматично визначати об'єкти, розпізнавати обличчя, витягувати текст із картинок, а також створювати власні моделі для специфічних бізнес-завдань через Custom Labels. Rekognition працює через API, що дає змогу інтегрувати його у будь-які бізнес-процеси, а також використовувати власні набори даних для навчання моделей під конкретні задачі. Ще одним важливим сервісом є Amazon Lex, який використовується для створення чат-ботів і голосових асистентів. Він поєднує автоматичне розпізнавання мови (ASR) та розуміння природної мови (NLU), що дозволяє створювати дійсно «розумні» боти для різних платформ: від контакт-центрів до мобільних додатків. [44]

Говорячи про IBM Watson, який тепер є частиною платформи watsonx, то він вирізняється своєю універсальністю, надаючи рішення для аналізу даних, автоматизації прийняття рішень і створення інтелектуальних асистентів у таких галузях, як охорона здоров'я, фінанси та підтримка клієнтів. Наприклад, Humana, провідна американська страхова компанія, застосовує Watson Assistant для обробки 7000 щоденних запитів про страхування, що скоротило витрати на контакт-центр і підвищило задоволеність клієнтів.

Сама платформа watsonx складається з кількох основних модулів, так watsonx.ai дозволяє створювати, тренувати й розгортати моделі штучного інтелекту, watsonx.data відповідає за управління даними в гібридних хмарних

середовищах, а `watsonx.governance` допомагає стежити за етичністю використання ІІ та відповідністю стандартам, наприклад, GDPR. Модуль `Watson Assistant` є інструментом для створення чат-ботів, які розуміють природну мову й можуть працювати кількома мовами в режимі реального часу, що ідеально підходить для сучасних контакт-центрів, а для глибшого аналізу неструктурованих даних, наприклад, документів чи медичних записів, застосовується `Watson Discovery`. [45]

Розглядаючи сферу логістики, то тут `FourKites` є провідною платформою для забезпечення прозорості та прогнозування для ланцюгів постачання. Завдяки використанню штучного інтелекту, ця система дозволяє компаніям не лише відстежувати місцезнаходження вантажів у реальному часі, а й отримувати аналітику щодо можливих затримок, оптимізувати управління запасами та швидко реагувати на будь-які збої. Наприклад, компанія `Kimberly-Clark` впровадила рішення `FourKites Dynamic Yard`, що дозволило скоротити витрати на штрафи за простій вантажівок на 50% лише за місяць, зекономивши майже пів мільйона доларів.

Технічна архітектура `FourKites` побудована так, щоб інтегруватися з існуючими бізнес-системами через API, збираючи дані з GPS-трекерів, IoT-пристроїв і ERP-систем, таких як `SAP` чи `Oracle`. Це забезпечує централізовану інформаційну панель для візуалізації всіх логістичних процесів, а також систему сповіщень і звітності для оперативного аналізу ефективності. Важливою складовою є `Intelligent Control Tower` - цифровий диспетчерський центр, який не лише відображає поточний стан ланцюга постачання, а й пропонує варіанти вирішення проблемних ситуацій на основі даних у реальному часі.

Також варто згадати функцію `Fin AI`, яка дозволяє користувачам отримувати відповіді на запитання щодо ланцюга постачання у звичній для них формі, використовуючи природну мову, що, в свою чергу, значно спрощує роботу з платформою для менеджерів і логістів, які не мають технічної підготовки. Ще одним важливим елементом є алгоритм `DynamicETA`, який прогнозує час

прибуття вантажу, аналізуючи понад 150 різних факторів - від погодних умов до поточного трафіку. [46]

У агросекторі однією з найуспішніших ШІ-платформ є Taranis, яка поєднує високотехнологічні методи збору даних, машинне навчання та глибокі агрономічні знання для оптимізації виробничих процесів. Ядро системи становить субміліметрова аерозйомка, що забезпечує роздільну здатність до 0,3 мм на піксель. Це дозволяє виявляти мікроскопічні ознаки хвороб, шкідників або дефіциту поживних речовин на рівні окремих листків, що є унікальною перевагою порівняно з аналогами. Разом із цим, Taranis не просто робить знімки - вона обробляє величезні обсяги даних, збирає тисячі зображень і аналізує їх за допомогою ШІ-моделей, натренованих на більш ніж 500 мільйонах точок даних, щоб генерувати рекомендації через Ag Assistant - генеративний ШІ, який інтегрує текст, зображення та навіть аудіо. Наприклад, якщо система виявляє початок грибкової інфекції на пшениці, вона не лише попереджає фермера, а й розраховує оптимальний час обробки, враховуючи прогноз погоди та економічну доцільність.

До того ж, Taranis не працює ізольовано, так партнерство з Syngenta, оголошене в жовтні 2024 року, дозволяє інтегрувати ШІ-рішення із платформою Cropwise, надаючи фермерам рекомендації безпосередньо в інтерфейсі управління господарством. Це спрощує процес прийняття рішень, тобто фермери можуть швидко визначити, які ділянки потребують додаткового поливу чи внесення добрив, без необхідності перемикатися між різними програмами. Така інтеграція робить технологію більш зручною для великих господарств, які вже використовують цифрові платформи. [47]

Однією з найпомітніших ШІ-платформ у сфері віртуальної первинної медицини є K Health. Її головна ідея якої полягає в тому, щоб зробити медичну допомогу доступнішою, швидшою та більш персоналізованою - користувачі можуть отримати діагностичні поради та рекомендації щодо лікування через чат-бот, не виходячи з дому. K Health використовує складну багаторівневу ШІ-архітектуру, алгоритми якої навчені на величезних масивах медичних даних -

понад 400 мільйонів записів від Massabi Healthcare Services та мільйони кейсів від Mayo Clinic. Це дозволяє системі аналізувати симптоми, історію хвороби, вік, стать, навіть генетичні особливості, і пропонувати ймовірні діагнози з урахуванням індивідуального контексту. Крім того, система застосовує алгоритм «наступного найкращого питання», який допомагає уточнити діагноз, задаючи релевантні запитання на основі попередніх відповідей користувача. Наприклад, якщо пацієнт скаржиться на біль у грудях, система послідовно запитуватиме про характер болю, тривалість, супутні симптоми, а потім порівнюватиме дані з мільйонами схожих випадків. Але K Health не обмежується лише автоматизованими рекомендаціями, якщо ШІ визначає, що потрібна додаткова оцінка, користувачі можуть зв'язатися з лікарями, тим самим забезпечуючи баланс між технологічною ефективністю та людським досвідом. [48]

Другий розділ дав детальніше розуміння про практичні сервіси та тенденції застосування ШІ в компаніях різних галузей. Вивчення успішних кейсів з різних галузей дозволило виявити спільні закономірності та визначити ключові технології, що демонструють найбільший потенціал для оптимізації процесів, персоналізації взаємодії з клієнтами та створення нових джерел доходу. Водночас, було проаналізовано значні переваги від впровадження ШІ, такі як підвищення операційної ефективності та покращення якості прийняття рішень, а також ідентифіковано низку викликів, серед яких значні інвестиції, потреба у кваліфікованих кадрах та етичні аспекти.

Отримані результати аналізують сучасний стан інтеграції ШІ у бізнес, підкреслюючи як значний потенціал, так і нагальну потребу в розробці обґрунтованих методичних рекомендацій. Це створює міцну основу для формування практичних пропозицій, які стануть предметом розгляду у заключному розділі роботи, орієнтуючись на подолання виявлених викликів та максимальне використання переваг.

## РОЗДІЛ 3. ІННОВАЦІЙНЕ МОДЕЛЮВАННЯ БІЗНЕСУ

### *3.1. Аналіз інноваційної бізнес-моделі компанії.*

Для аналізу успішної інноваційної бізнес-моделі розглянемо приклад компанії Amazon - одного з лідерів глобального ринку електронної комерції та хмарних технологій. Amazon функціонує як екосистема, що охоплює такі ключові напрямки діяльності як електронна комерція, хмарні обчислення через Amazon Web Services (AWS), цифровий контент (Prime Video, Kindle), розумні пристрої (Alexa), логістичні рішення та фізична роздрібна торгівля (Amazon Go, Whole Foods). Географічно компанія представлена в Північній Америці, Європі, Азії та активно розширюється на ринки, що розвиваються, включаючи Індію та Латинську Америку.

Ключовими принципами бізнес-моделі Amazon є клієнтоорієнтованість, постійна інноваційність та максимальна операційна ефективність. Компанія обслуговує надзвичайно різноманітні сегменти клієнтів: від мільйонів кінцевих покупців по всьому світу, які здійснюють покупки через її канали, до численних продавців, які використовують платформу Amazon для реалізації своїх товарів широкій аудиторії. Окрему, але не менш важливу категорію, складають розробники та підприємства будь-якого розміру, що користуються хмарною обчислювальною платформою Amazon Web Services (AWS). При цьому Amazon ретельно відстежує своїх клієнтів, аналізуючи їхні інтереси, рівень залученості та персональну інформацію, що дозволяє формувати максимально релевантні пропозиції.

Інноваційність бізнес-моделі Amazon полягає в її здатності постійно адаптуватися та впроваджувати нові технології для покращення клієнтського досвіду та операційної ефективності, що значною мірою є заслугою впровадження ШІ в різні сфери діяльності компанії. Розглянемо рекомендаційну систему Amazon, що є основою її електронної комерції, генеруючи до 35% продажів. Вона базується на методі спільної фільтрації «item-to-item collaborative filtering», який визначає зв'язки між товарами на основі взаємодій користувачів,

створюючи матрицю схожості на основі спільних покупок. Технічно система використовує методи матричної факторизації, такі як Singular Value Decomposition (SVD) і Alternating Least Squares (ALS), для розкладання матриці взаємодії користувач-товар, що дозволяє виявити приховані фактори, які пояснюють поведінку користувачів. [49]

Крім того, Amazon застосовує контентну фільтрацію, аналізуючи характеристики продуктів, такі як назви, категорії, описи та специфікації. Для цього використовуються методи обробки природної мови (NLP) для вилучення інформації з текстових даних, таких як описи продуктів і відгуки клієнтів. Система працює в реальному часі, адаптуючись до змін у поведінці та вподобаннях користувачів. Якщо клієнт раптово виявляє новий інтерес, рекомендації негайно коригуються. Система також використовує механізм зворотного зв'язку, постійно вдосконалюючи точність і релевантність рекомендацій шляхом аналізу відгуків, оцінок і історії покупок.

Тепер розглянемо систему оптимізації логістики Amazon, яка використовує передові алгоритми ШІ для управління одною з найскладніших мереж постачання у світі. Компанія розробила власну логістичну інфраструктуру для зменшення залежності від зовнішніх партнерів з доставки та забезпечення швидших і дешевших варіантів доставки для своїх клієнтів. Основою логістичної оптимізації є алгоритми машинного навчання, які аналізують величезні обсяги даних про попит, географічне розташування клієнтів, доступність товарів на складах, транспортні маршрути та погодні умови. Ці алгоритми використовують методи прогнозного аналізу для передбачення попиту на товари в різних регіонах, що дозволяє заздалегідь розміщувати популярні товари у стратегічно важливих складських центрах. Система динамічної маршрутизації використовує алгоритми оптимізації для планування найефективніших маршрутів доставки в реальному часі, враховуючи трафік, відстань, пріоритетність замовлень і завантаженість транспортних засобів.

ШІ-технології також використовуються і для оптимізації роботи складів Amazon. Для цього використовуються автономні мобільні роботи такі як Proteus,

які переміщують стелажі чи товари на складі, завдяки комбінації датчиків, включаючи лідар і камери, для навігації в навколишньому середовищі. Ці роботи використовують алгоритми одночасної локалізації і картографування (SLAM) для побудови карт і визначення свого місцезнаходження, що дозволяє їм ефективно переміщатися і уникати перешкод, що має вирішальне значення для підтримки робочого процесу в динамічних складських умовах. Роботизовані маніпулятори, такі як Sparrow, оснащені комп'ютерним зором і можливостями машинного навчання, щоб вибирати і розміщувати окремі предмети, використовуючи згорткові нейронні мережі (convolutional neural networks) для розпізнавання об'єктів на зображеннях, знятих камерами. Після ідентифікації об'єкта робот використовує навчання з підкріпленням або інші алгоритми планування, щоб визначити найкращий спосіб захопити і керувати ним, забезпечуючи точність і зменшуючи кількість помилок. Інтегровані системи, такі як Sequoia, поєднують в собі кілька технологій, включаючи робототехніку, штучний інтелект і автоматизацію, щоб впорядкувати весь процес виконання замовлення, координуючи рух роботів, керуючи запасами і оптимізуючи робочі процеси для мінімізації часу обробки. [50]

Разом із цим, система управління складом використовує алгоритми машинного навчання для оптимізації розташування товарів та планування маршрутів роботів. Ці алгоритми аналізують частоту замовлень різних товарів, сезонні тренди та фізичні характеристики продуктів для визначення оптимального розміщення у складському просторі. До того ж, Amazon розробляє нові покоління робототехніки, включаючи роботів-маніпуляторів для сортування та упакування товарів. Ці роботи використовують передові технології машинного зору та штучного інтелекту для розпізнавання різних типів товарів, оцінки їхніх фізичних властивостей та виконання делікатних операцій упакування.

Ще одним аспектом використання ШІ компанією Amazon є голосовий асистент Alexa, що базується на передових технологіях обробки природної мови, розпізнавання мовлення та машинного навчання. Архітектура Alexa включає декілька ключових компонентів: Automatic Speech Recognition для перетворення

звукових сигналів у текст, Natural Language Understanding для інтерпретації намірів користувача та Text-to-Speech для генерації відповідей. Серцем системи є хмарна інфраструктура Amazon, яка обробляє мільйони голосових запитів щодня. Коли користувач промовляє команду, звукові дані передаються до AWS, де спеціалізовані алгоритми машинного навчання аналізують акустичні характеристики мовлення, розпізнають слова та фрази, а потім інтерпретують контекст та намір запиту. Alexa інтегрована з величезною екосистемою навичок та сервісів Amazon, включаючи електронну комерцію, музичні сервіси, розумний дім та сторонні додатки. Ця інтеграція створює синергетичний ефект, де голосовий інтерфейс стає природним способом взаємодії з усією цифровою екосистемою Amazon, від замовлення товарів до управління розумними пристроями. [51]

### *3.2. Оцінка ефективності штучного інтелекту для оптимізації бізнесу.*

Ефективність штучного інтелекту в бізнесі - це багатоаспектне поняття, яке визначає здатність технології не просто вирішувати задачі, а й змінювати організаційні процеси, створюючи відчутні економічні та стратегічні переваги. Варто пам'ятати, що ефективне впровадження штучного інтелекту вимагає кардинально нового підходу до планування та оцінювання результативності. Звичні показники окупності інвестицій (ROI) не завжди відображають весь масштаб впливу ШІ на діяльність компанії, оскільки ця технологія нерідко створює непрямі ефекти, зокрема, покращення якості ухвалених рішень або підвищення гнучкості бізнес-процесів.

Перед стартом будь-якого ШІ-проекту важливо провести ретельну оцінку готовності організації та потенційної ефективності ініціативи. Індекс готовності до ШІ (AI Readiness Index) становить комплексний інструмент, що оцінює організацію за шістьма ключовими критеріями з різною ваговою значимістю. У ньому стратегічне бачення та цілі складають 15% загальної оцінки, відображаючи наскільки чітко керівництво розуміє роль ШІ в досягненні бізнес-цілей. IT-інфраструктура займає найвагомішу частку в 25%, оскільки саме

технічні можливості організації визначають змогу реалізації проекту. 20% становить якість даних, адже ШІ-моделі можуть бути ефективними лише за наявності структурованих, релевантних і достатніх за обсягом даних. Управління та керівництво, займає 15% та оцінює готовність топ-менеджменту підтримувати проект ресурсами та рішеннями. Таку ж частку складають і навички команди, визначаючи спроможність персоналу ефективно працювати з новою технологією. Нарешті, корпоративна культура - 10% - відображає готовність організації до змін та інновацій. Згідно з дослідженням Cisco AI Readiness Index, лише 27% компаній входять до категорії «Pacesetters» (з оцінкою 76-100 балів), демонструючи повну готовність до масштабування ШІ завдяки інтеграції технології в стратегію, тоді як 15% («Laggards», 0-25 балів) ризикують втратити до 45% бюджету через невдалі ініціативи, зокрема через низьку якість даних та відсутність підтримки керівництва. [52]

Період окупності, в свою чергу, визначає часовий горизонт, необхідний для відшкодування початкових інвестицій і розраховується як співвідношення початкових інвестицій до річного грошового потоку. Водночас важливо розуміти, що цей метод має обмеження, оскільки не враховує вигоди після періоду окупності. Успішні ШІ-проекти зазвичай демонструють період окупності 12-24 місяці, при цьому проекти з періодом понад 36 місяців потребують додаткового обґрунтування та можуть сигналізувати про завищені очікування або недооцінку складності впровадження.

Варто згадати й про сукупну вартість володіння, оскільки цей показник враховує всі витрати, пов'язані з впровадженням та експлуатацією ШІ-системи протягом її повного життєвого циклу. Він включає не лише очевидні витрати на розробку та придбання обладнання, але й приховані - на навчання персоналу, інтеграцію з існуючими системами, ліцензії, постійну підтримку та регулярні оновлення моделей, тим самим доводячи, що початкова вартість придбання, як правило, становить лише половину від загальної суми витрат протягом усього життєвого циклу системи.

Після впровадження ШІ у бізнес-процеси необхідно провести оцінку його реального впливу, використовуючи конкретні метрики, які відображають як кількісні, так і якісні результати. У контексті машинного навчання, точність моделі (accuracy) є однією з базових метрик, яка вимірює, наскільки часто модель ШІ правильно передбачає результати. Наприклад, у системах рекомендацій точність може показувати, скільки відсотків запропонованих товарів клієнти дійсно придбали. Для класифікаційних моделей вона розраховується за формулою (3.1).

$$\text{Точність} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.1)$$

де TP - кількість правильно передбачених позитивних випадків;

TN - кількість правильно передбачених негативних випадків;

FP - кількість помилково передбачених позитивних випадків;

FN - кількість помилково передбачених негативних випадків.

Однак точність може бути оманливою при роботі з незбалансованими наборами даних, де один клас значно переважає інші, тому потребує доповнення іншими метриками.

Прецизійність (precision) та повнота (recall) формують взаємопов'язану пару показників, що дозволяє глибше проаналізувати якість роботи моделі. Прецизійність відображає частку об'єктів, правильно віднесених до позитивного класу, серед усіх об'єктів, які модель класифікувала як позитивні і відображається формулою

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.2)$$

Формула (3.3) відображає підрахунок повноти, яка характеризує здатність моделі виявляти всі позитивні випадки.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.3)$$

Ці метрики є особливо важливими у сферах з високою ціною помилок, таких як медична діагностика, де пропуск захворювання (низька повнота) може мати

фатальні наслідки, або виявлення шахрайства, де помилкове блокування легітимної транзакції (низька прецизійність) може шкодити клієнтському досвіду.

Для забезпечення збалансованої оцінки продуктивності моделі, особливо коли одна з метрик (прецизійність або повнота) значно нижча за іншу, використовується F1-міра. Вона розраховується за формулою (3.4) та є гармонічним середнім між прецизійністю і повнотою

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3.4)$$

Цей показник є особливо цінним для оцінки моделей, що працюють з незбалансованими даними, оскільки він надає єдине число, яке враховує обидва аспекти точності.

Переходячи до бізнес-метрик, коефіцієнт віддачі від інвестицій (ROI) є основним показником економічної ефективності ІІІ-проектів, що розраховується як:

$$ROI = \frac{\text{Прибуток} - \text{Витрати}}{\text{Витрати}} \times 100\% \quad (3.5)$$

Де прибуток включає як прямі доходи, так і заощадження від впровадження ІІІ. Згідно з дослідженнями, в середньому підприємства отримують 3,7 долара прибутку на кожен долар, вкладений у ІІІ, а 5% організацій демонструє результат у 10 доларів прибутку на кожен інвестований долар. [53]

Економія витрат відображає пряме зниження операційних витрат завдяки автоматизації та оптимізації процесів, а індекси задоволеності та утримання клієнтів є досить схожими та пов'язаними між собою. Перший вимірює рівень задоволеності клієнтів покращеними послугами, а другий - здатність ІІІ утримувати клієнтів протягом тривалого періоду. Підвищення цих показників досягається через персоналізацію досвіду, прогнозування потреб клієнтів та активне вирішення проблем. Навіть невелике 5% збільшення утримання може підвищити прибуток на 25-95%. [54Помилка! Джерело посилання не знайдено.]

Таким чином, ефективна оцінка ІІІ потребує збалансованого поєднання як технічних метрик, так і бізнес-показників. Саме такий комплексний підхід забезпечує можливість постійного вдосконалення ІІІ-систем через виявлення вразливих місць, оптимізацію алгоритмів та адаптацію до змінних бізнес-потреб, що робить інвестиції в штучний інтелект не лише виправданими, але й стратегічно необхідними для підтримання конкурентоспроможності в сучасному бізнес-середовищі.

### *3.3. Пропозиції щодо впровадження штучного інтелекту для бізнесу в Україні.*

Впровадження штучного інтелекту в бізнес-процеси українських підприємств є не лише питанням сучасної технологічної трансформації, а й стратегічною необхідністю, що зумовлена глобальними тенденціями. Український бізнес, має свої особливості, що варто враховувати при розробці пропозицій щодо інтеграції ІІІ. Загалом, українські підприємства демонструють значну гнучкість, орієнтованість на інновації та здатність до швидкого реагування на ринкові зміни, що особливо помітно в останні роки. Разом із цим, бізнес часто стикається з низкою специфічних проблем, які можуть уповільнювати або ускладнювати повномасштабне впровадження передових технологій.

Однією з ключових перешкод на шляху до широкого впровадження штучного інтелекту є проблема кадрів, існує відчутний дефіцит кваліфікованих кадрів саме у сфері ІІІ, зокрема фахівців з машинного навчання, аналізу даних та інтеграції ІІІ-систем. З іншого боку, часто бракує розуміння можливостей та переваг ІІІ серед керівного складу підприємств, що також може гальмувати інвестиції у відповідні технології. Водночас важливо зазначити, що значні ресурси є ще одним фактором, який стримує впровадження. Розробка та інтеграція складних ІІІ-рішень потребує значних фінансових вливань, що може бути недоступним для багатьох українських підприємств.

Оскільки в Україні більша частина бізнесів є малими та середніми, буде доречним розглянути пропозиції щодо впровадження ІІІ саме для них. Перш за

все, доцільним є впровадження простих, готових ШІ-рішень, що не потребують значних інвестицій у розробку з нуля. Наприклад інтеграція чат-ботів на веб-сайти та в соціальні мережі дозволяє автоматизувати відповіді на типові питання, обробку замовлень та надання інформації, що значно розвантажує персонал та забезпечує цілодобову доступність сервісу.

Для онлайн-магазинів та сервісів корисними можуть бути системи персоналізації та рекомендацій, що надають можливість аналізувати поведінку користувачів та пропонувати їм релевантні товари чи послуги. Так, невелика кав'ярня може використовувати просту ШІ-систему для аналізу історії замовлень клієнтів та пропонувати їм акції на улюблені напої або десерти, а ефект буде виражатися у збільшенні середнього чека, підвищенні лояльності клієнтів та зростанні конверсії.

До того ж, ШІ може допомогти у сегментації аудиторії, оптимізації рекламних кампаній, визначенні найбільш ефективних каналів реклами та автоматичному створенні персоналізованих розсилок для різних груп клієнтів, тим самим оптимізуючи маркетингові бюджети та підвищуючи ефективність залучення нових клієнтів. Прості системи для аналізу даних та прогнозування є незамінними для відслідковування попиту на товари, тенденцій продажів або оптимізації управління запасами. Наприклад, невелика пекарня може використовувати ШІ для прогнозування кількості необхідних інгредієнтів на наступний день, щоб уникнути надлишків або нестачі продукції, тим самим зменшуючи витрати, оптимізуючи робочі процеси та підвищуючи прибуток.

Для аграрного сектору, який є однією з ключових галузей української економіки, ШІ також відкриває значні можливості. Системи точного землеробства, що використовують машинне навчання та аналіз супутникових даних, дозволяють оптимізувати використання добрив, води та інших ресурсів. Наприклад, українська компанія AgriEye розробила ШІ-платформу, яка допомагає фермерам прогнозувати врожайність і виявляти захворювання рослин на ранніх стадіях. Впровадження таких технологій не лише підвищує

продуктивність, а й сприяє сталому розвитку, що є важливим у контексті європейських екологічних стандартів.

Для України, де відповідність міжнародним стандартам AML та загальна прозорість фінансового сектора є ключовою для залучення іноземних інвестицій та повноцінної інтеграції з європейськими інституціями, ШІ-рішення у фінансах є не просто питанням підвищення внутрішньої ефективності, а стратегічним кроком для зміцнення міжнародної довіри та економічної стабільності. Наприклад, застосування ШІ для виявлення шахрайства шляхом аналізу платіжних транзакцій у режимі реального часу, з метою виявлення незвичайних або підозрілих зв'язків, що можуть свідчити про шахрайську діяльність. Крім того, ШІ може використовуватися для оптимізації фінансового управління, зокрема, для глибокого аналізу ризиків, ефективного управління інвестиційними портфелями та точного прогнозування змін на фінансових ринках.

Окрема увага має приділятися застосуванню ШІ у сфері охорони здоров'я, особливо в умовах військового стану та післявоєнної відбудови країни. Медичні платформи, інтегровані з ШІ-алгоритмами діагностики, можуть значно розширити доступність медичних послуг у віддалених регіонах та зонах з пошкодженою інфраструктурою. Системи машинного навчання здатні допомагати лікарям у постановці попередніх діагнозів на основі аналізу симптомів, медичної історії пацієнта та результатів обстежень, що особливо важливо в умовах дефіциту спеціалістів у певних медичних галузях. До того ж, ШІ-рішення для управління медичними запасами та планування ресурсів можуть оптимізувати роботу лікарень та амбулаторій, забезпечуючи більш ефективне використання обмежених коштів охорони здоров'я.

Успішна реалізація зазначених пропозицій потребує комплексного підходу, що включає створення відповідного нормативно-правового середовища, розвиток освітніх програм для підготовки кадрів у сфері ШІ та стимулювання партнерства між державою, бізнесом та науковими установами.

### *3.4. Розробка рекомендацій та моделей.*

В умовах динамічного економічного середовища України, де значну частку становлять малі та середні підприємства, питання ефективності та оптимізації бізнес-процесів набуває особливої актуальності. Показово, що ці підприємства часто функціонують з обмеженими бюджетами, що робить впровадження інноваційних рішень, зокрема технологій штучного інтелекту, можливим лише за умови їхньої фінансової доступності та чіткої економічної обґрунтованості. Разом із цим, інвестиції у ШІ повинні не просто мінімізувати витрати, а й гарантовано генерувати відчутний економічний прибуток, що є ключовим критерієм для бізнесу з обмеженими ресурсами. Таким чином, розробка та імплементація ШІ-систем для українських малих і середніх підприємств має бути орієнтована на забезпечення максимального впливу на прибутковість при мінімальних початкових витратах.

З огляду на ці обставини, для практичної демонстрації можливостей впровадження доступних ШІ-рішень було обрано умовний місцевий магазин. На мій погляд, цей вибір є доречним, оскільки традиційно основним джерелом збуту для таких невеликих підприємств є фізичний магазин, що обмежує охоплення аудиторії. Водночас, диджиталізація бізнесу через створення веб-сайту дозволяє значно розширити коло потенційних клієнтів, але супроводжується необхідністю ефективної взаємодії з ними. При цьому, наймання додаткового консультанта для онлайн-підтримки може бути економічно недоцільним для малого бізнесу. Саме тому, для оптимізації процесу обслуговування клієнтів на сайті та підвищення його ефективності, доцільно інтегрувати функціонал чат-бота. Такий підхід дозволяє забезпечити цілодобову доступність інформації, швидке вирішення типових запитів та персоналізовані рекомендації без значних кадрових витрат, що відповідає основним принципам економії та максимізації прибутку для малого бізнесу в Україні. Водночас важливо зазначити, що такий інструмент, за умови його ретельної розробки, здатен ефективно взаємодіяти з користувачами, виявляючи їхні потреби та пропонуючи оптимальні рішення, що сприятиме підвищенню лояльності клієнтів.

Варто також підкреслити, що імплементація подібних ІІІ-рішень, як чат-бот для книжкового магазину, є перспективною не лише для роздрібної торгівлі, але й для інших секторів малого та середнього бізнесу в Україні. До прикладу, невеликі локальні кафе можуть використовувати чат-ботів для приймання замовлень, надання інформації про меню та акції, а також збору зворотного зв'язку відвідувачів. Салони краси чи фітнес-студії можуть автоматизувати процес запису на послуги та нагадування про візити. Водночас, для сфери послуг, таких як юридичні консультації або персональні заняття, чат-боти можуть стати першою точкою контакту з клієнтами, надаючи відповіді на типові питання та перенаправляючи складніші запити до менеджерів. Такі рішення є економічно вигідними, адже дозволяють зменшити витрати на рутинні операції, прискорити обслуговування, зробити сервіси доступнішими, а в результаті підвищити задоволеність клієнтів і збільшити можливий прибуток.

Для побудови моделі взято умовний локальний книжковий магазин «БукЛайф», який, як і більшість малих підприємств, мав обмежене охоплення аудиторії, зосереджене переважно на фізичній точці продажу. Найкращим рішенням для вирішення головної проблеми та збільшення прибутку без значних початкових витрат є створення веб-сайту з інтегрованим чат-ботом.

Проект розроблено на основі архітектури клієнт-сервер, де серверна частина реалізована за допомогою фреймворку Flask на мові програмування Python, а клієнтська частина представлена стандартними веб-технологіями (HTML, CSS та JavaScript).

Серцем серверної логіки є файл `app.py` (див. додаток А), який виконує роль основного контролера та обробника запитів. Він відповідає за управління базою даних, логіку роботи кошика покупок, оформлення замовлень, а також є точкою входу для взаємодії з чат-ботом.

Ініціалізація додатку починається з виклику функції `load_database()`, Рис. 3.1. показує її реалізацію в коді, вона відкриває і завантажує базу даних з файлу `database.json` у глобальну змінну `data`. Якщо файл відсутній або пошкоджений,

система передбачає створення порожньої структури даних, що підвищує її стійкість до збоїв.

```
def load_database():
    global data
    try:
        with open('database.json', 'r', encoding='utf-8') as f:
            data = json.load(f)
        logger.info("Базу даних успішно завантажено.")
    except FileNotFoundError:
        logger.error("ПОМИЛКА: Файл database.json не знайдено! Створюю порожню базу даних.")
        data = {"books": [], "clients": [], "orders": [], "categories": [], "reviews": [], "chatbot_training_data": []}
    except json.JSONDecodeError as e:
        logger.error(f"ПОМИЛКА: Некоректний формат JSON в database.json: {e}")
        data = {"books": [], "clients": [], "orders": [], "categories": [], "reviews": [], "chatbot_training_data": []}
```

Рис. 3.1. Вигляд функції load\_database() у коді програми

Джерело: розроблено автором

Чат-бот, реалізований у класі WebsiteChatbot, є ключовим елементом сайту, що демонструє застосування ШІ для автоматизації клієнтської взаємодії. Його основне завдання - обробка текстових запитів користувачів, визначення їхніх намірів і надання релевантних відповідей, таких як пошук книг, рекомендації за віком, настроєм чи жанром, перевірка статусу замовлення або інформація про доставку. Опис архітектури чат-боту представлений на Рис.3.2., ілюструючи, як він обробляє вхідні запити користувача, розпізнає наміри, витягує контекст і генерує відповідь.

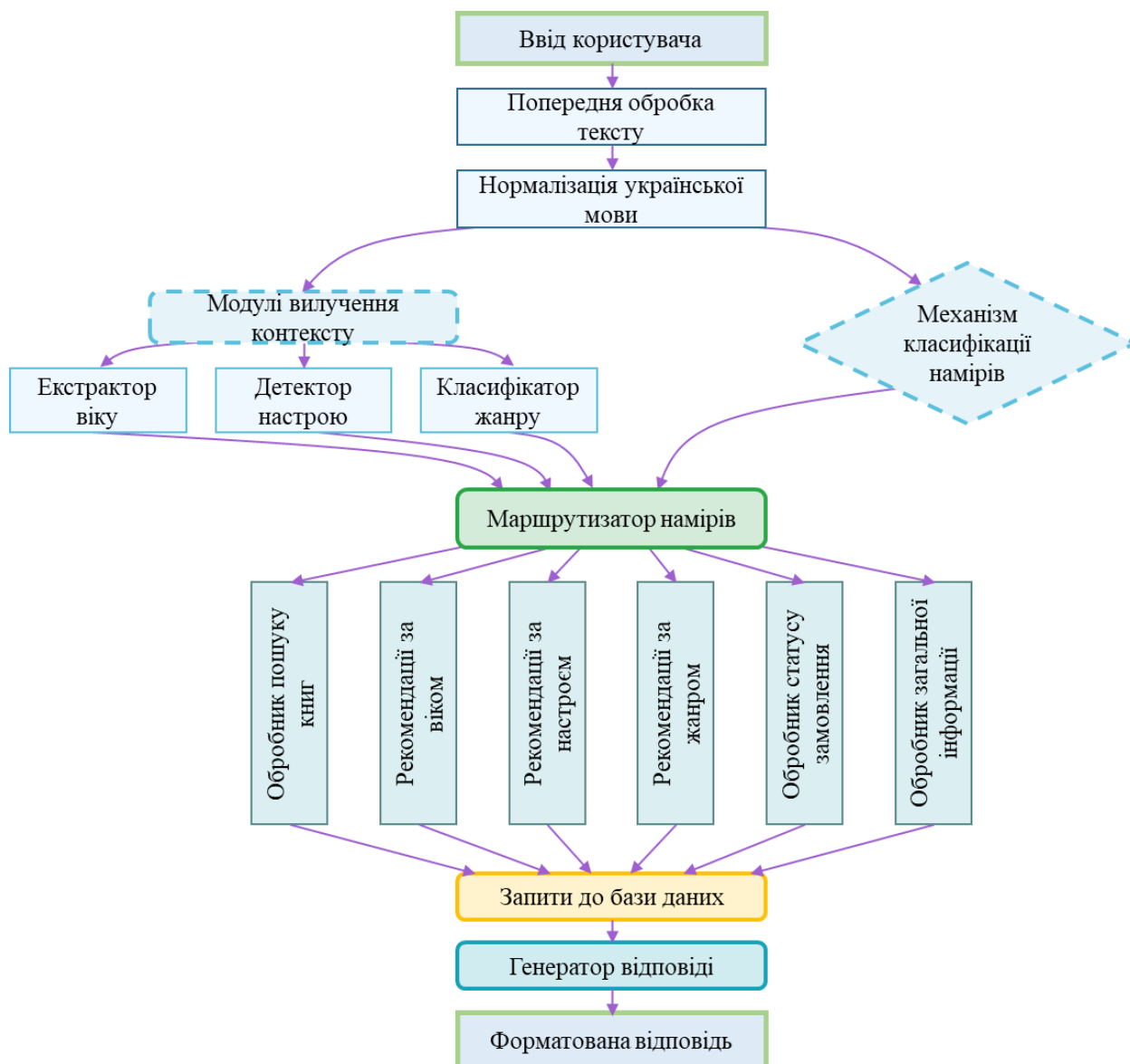


Рис. 3.2. Архітектура WebsiteChatbot

Джерело: розроблено автором

Тепер розглянемо логіку роботи чат-бота детальніше, першим етапом клас WebsiteChatbot ініціалізується з навчальними даними та налаштуваннями для обробки української мови. Конструктор класу ініціалізує тренувальні дані (self.intents) із бази даних. Ці дані включають наміри (intents) та шаблони (patterns) фраз, що відповідають цим намірам. Важливою особливістю є інтеграція правил для нормалізації українських слів (ukr\_endings), код можна побачити на рис. 3.3., це дозволяє боту краще розуміти різні форми одного слова (наприклад, "книга", "книги", "книгу"). Крім цього, ініціалізуються розширені наміри (enhanced\_intents) та категорії настроїв (mood\_categories) з їхніми

ключовими словами та тегами, що підвищує гнучкість системи у розпізнаванні запитів.

```
class WebsiteChatbot:
    def __init__(self, training_data):
        self.intents = training_data
        self.context = {}

        self.ukr_endings = ['а', 'и', 'е', 'о', 'у', 'ї', 'я', 'ю', 'є', 'ї',
                             'им', 'ій', 'ою', 'их', 'их', 'огд', 'ому', 'ій',
                             'ти', 'ся', 'ої', 'ій', 'ами', 'ями']
```

Рис. 3.3. Фрагмент коду, що демонструє завантаження тренувальних даних та параметрів нормалізації мови для чат-бота

Джерело: розроблено автором

На рис. 3.4. показано метод `_normalize_word`, який реалізує спрощену нормалізацію слів для української мови, видаляючи типові закінчення (з переліку `self.ukr_endings`). Це допомагає боту розпізнавати корінь слова незалежно від його граматичної форми.

```
def _normalize_word(self, word):
    word = word.lower().strip()
    for ending in sorted(self.ukr_endings, key=len, reverse=True):
        if word.endswith(ending) and len(word) > len(ending):
            return word[:-len(ending)]
    return word
```

Рис. 3.4. Метод `_normalize_word`

Джерело: розроблено автором

Цей підхід не є повноцінним морфологічним аналізом, але ефективний для малого проекту, де потрібно обробляти обмежений набір слів. Метод `_get_normalized_forms` застосовує нормалізацію до списку слів, а `preprocess_text` рис. 3.5. очищає вихідний текст користувача від зайвих символів, привести його до нижнього регістру та нормалізувати всі слова. Це уніфікує вхідні дані, значно спрощуючи їхнє подальше зіставлення з шаблонами.

```
def preprocess_text(self, text):
    text = text.lower().strip()
    text = re.sub(r'^\w\s', ' ', text)
    text = re.sub(r'\s+', ' ', text)
    words = text.split()
    normalized_words = [self._normalize_word(word) for word in words]
    return ' '.join(normalized_words)
```

Рис. 3.5. Метод preprocess\_text для нормалізації слів

Джерело: розроблено автором

Також система містить спеціалізовані функції для витягування контекстної інформації з повідомлень користувачів. Логіка визначення віку реалізована в коді рис. 3.6. в методі extract\_age. Він шукає числові значення з контекстом «років» (наприклад, «10 років»). Якщо число не знайдено, метод аналізує ключові слова, такі як «дитина» чи «підліток», повертаючи відповідну вікову групу (“children”, “teenager”, “elderly”, “adult”).

```
def extract_age(self, message):
    processed_message = self.preprocess_text(message)
    age_patterns = re.findall(r'\b(\d{1,2})\s*(?:років?|річ|рок|рочки|б)\b', message)
    if age_patterns:
        return int(age_patterns[0])
    if any(w in processed_message for w in self._get_normalized_forms(['дитина', 'малюк', 'дошкільник'])):
        return "children"
    if any(w in processed_message for w in self._get_normalized_forms(['школяр', 'підліток', 'юнак', 'юначка'])):
        return "teenager"
    if any(w in processed_message for w in self._get_normalized_forms(['дідусь', 'бабуся', 'літній', 'старший', 'пенсіонер'])):
        return "elderly"
    if any(w in processed_message for w in self._get_normalized_forms(['дорослий', 'для дорослих'])):
        return "adult"
    return None
```

Рис. 3.6. Код функції extract\_age для ідентифікації віку в повідомленні

Джерело: розроблено автором

Метод detect\_mood, його код наданий на рис. 3.7., визначає настрій користувача, зіставляючи нормалізований текст із ключовими словами настрою (наприклад «веселий» для good\_mood). Структура self.mood\_categories містить категорії настрою з тегами й описами, що дозволяють підбирати книги за емоційним станом. Наприклад, для “relax\_mood” теги включають «поезія» та «лірика».

```
def detect_mood(self, message):
    processed_message = self.preprocess_text(message)
    detected_moods = []

    for mood, data in self.mood_categories.items():
        if any(keyword in processed_message for keyword in data['keywords']):
            detected_moods.append(mood)
    return detected_moods
```

Рис. 3.7. Код функції виявлення настрою detect\_mood

Джерело: розроблено автором

Система жанрової класифікації реалізована через функцію extract\_genre\_keywords, яка використовує словник відповідності між нормалізованими ключовими словами та ідентифікаторами категорій книг. Алгоритм підтримує 17 основних жанрових категорій, включаючи пригоди, детективи, фентезі, наукову літературу, поезію та інші.

Механізм визначення наміру find\_intent є центральним логічним блоком, що відповідає за класифікацію запиту користувача. Він переглядає всі визначені наміри (enhanced\_intents) та їхні шаблони, порівнюючи їх з обробленим повідомленням користувача. Його робота базується на системі оцінювання (скорингу), бот обчислює рейтинг відповідності, порівнюючи нормалізоване повідомлення користувача з нормалізованими патернами наміру. Нарахування балів відбувається за правилами, що показані на рис. 3.8., де просте співпадіння слова додає 1 бал, точне співпадіння фрази - 5 балів, а часткове співпадіння кореня слова - 2 бали. Якщо виявлено специфічну інформації (вік, настрій, жанр) значно підвищується пріоритет відповідного наміру (додатково 10 балів), що дозволяє боту більш точно реагувати на конкретні запити.

```

for pattern_word in patterns:
    if pattern_word in processed_message.split():
        score += 1 # Просте співпадіння слова
    if pattern_word == processed_message:
        score += 5 # Точне співпадіння
    elif len(pattern_word) > 3 and pattern_word in processed_message:
        score += 2 # Співпадіння кореня слова

if intent == "age_based_recommendations" and self.extract_age(message):
    score += 10
if intent == "mood_based_recommendations" and self.detect_mood(message):
    score += 10
if intent == "genre_based_recommendations" and self.extract_genre_keywords(message):
    score += 10

```

Рис. 3.8. Код обчислення балу відповідності наміру у функції `find_intent`

Джерело: розроблено автором

Це дозволяє боту віддати перевагу більш конкретному запиту. Наприклад, у фразі «порадьте книгу для дитини» намір `age_based_recommendations` отримує вищий пріоритет, ніж загальний `recommendations`.

Після того, як функція `find_intent` визначила намір користувача, управління передається відповідній `handle`-функції, яка відповідає за генерацію релевантної відповіді. На рис. 3.9. показаний код методу `handle_book_search`, який відповідає за пошук книг за запитом користувача. Вона нормалізує текст повідомлення, розбиває його на слова, а потім фільтрує та ранжує книги з бази даних (`data['books']`) за кількістю співпадінь.

```

def handle_book_search(self, message):
    processed_message = self.preprocess_text(message)
    words = processed_message.split()

    matching_books = []
    for book in data['books']:
        book_text = f"{book['title']} {book['author']} {book.get('description', '')} {book.get('tags', '')}".lower()
        book_text_normalized = self.preprocess_text(book_text) # Нормалізуємо текст книги

        matches = sum(1 for word in words if len(word) > 2 and word in book_text_normalized)
        if matches > 0:
            matching_books.append((book, matches))

    matching_books.sort(key=lambda x: x[1], reverse=True)

    if matching_books:
        top_books = matching_books[:3]
        response = "Ось що я знайшов:\n\n"
        for book, _ in top_books:
            response += f"■ {book['title']} - {book['author']}\n"
            response += f"■ Ціна: {book['price']} грн\n"
        response += "Бажаєте дізнатися більше про якусь книгу?"
    return response

```

Рис. 3.9. Алгоритм пошуку книг за ключовими словами у класі `WebsiteChatbot`

Джерело: розроблено автором

Результатом його роботи є виведення до трьох найбільш релевантних книг.

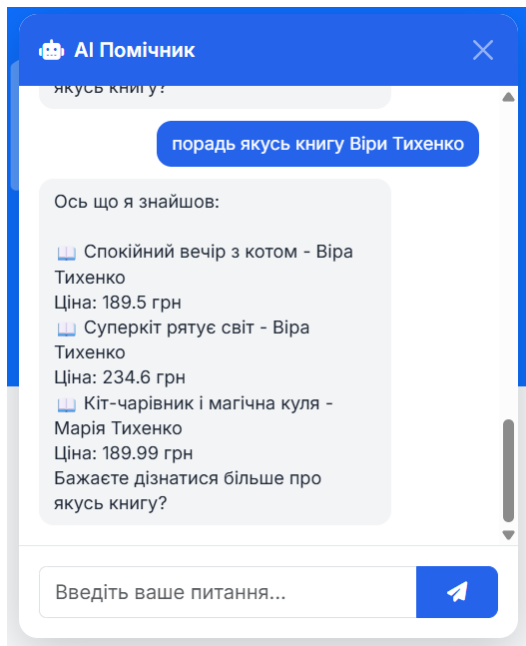


Рис. 3.10. Діалог з чат-ботом для отримання відповідних книг у базі даних

Джерело: розроблено автором

Метод `handle_age_based_recommendations` є ключовим для персоналізованих рекомендацій, оскільки він ефективно обробляє запити, фільтруючи книги за відповідними категоріями та виводячи найбільш рейтингові. Його особливістю є здатність розуміти як числові показники віку (наприклад, "10 років"), так і словесні позначення вікових груп (наприклад, "для підлітків", "для бабусі"), забезпечуючи високу гнучкість взаємодії з користувачем. Після визначення віку рис. 3.11., функція підбирає найбільш релевантні книги з бази даних, які відповідають вказаним критеріям.

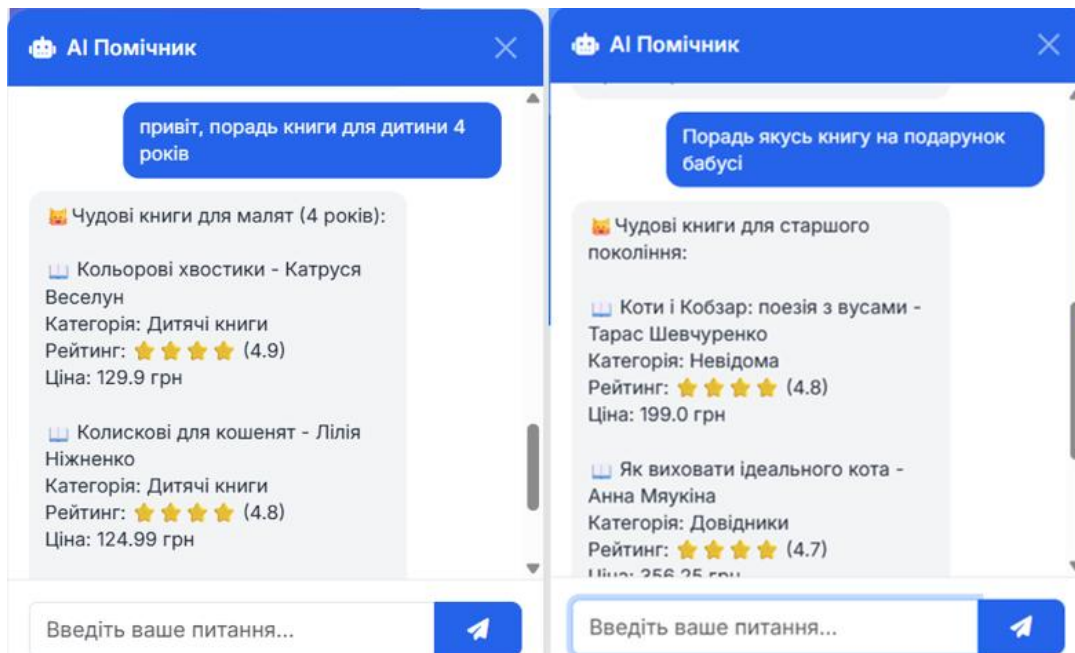


Рис. 3.11. Демонстрація роботи функції `handle_age_based_recommendations` для підбору книг за віком

Джерело: розроблено автором

Функція `handle_mood_based_recommendations` обробляє запити, пов'язані з настроєм користувача, використовуючи `detect_mood`. Вона спочатку шукає книги за тегами, пов'язаними з виявленими настроями (наприклад, "гумор" для "веселого" настрою). Якщо за тегами не знайдено достатньо книг, система просить користувача детальніше пояснити свій настрій рис. 3.12. та розширює пошук, включаючи книги за категоріями, що асоціюються з певним настроєм. Після взаємодії з ШІ-помічником користувач отримує 3 найбільш відповідних до настрою книги, що відсортовані за рейтингом.

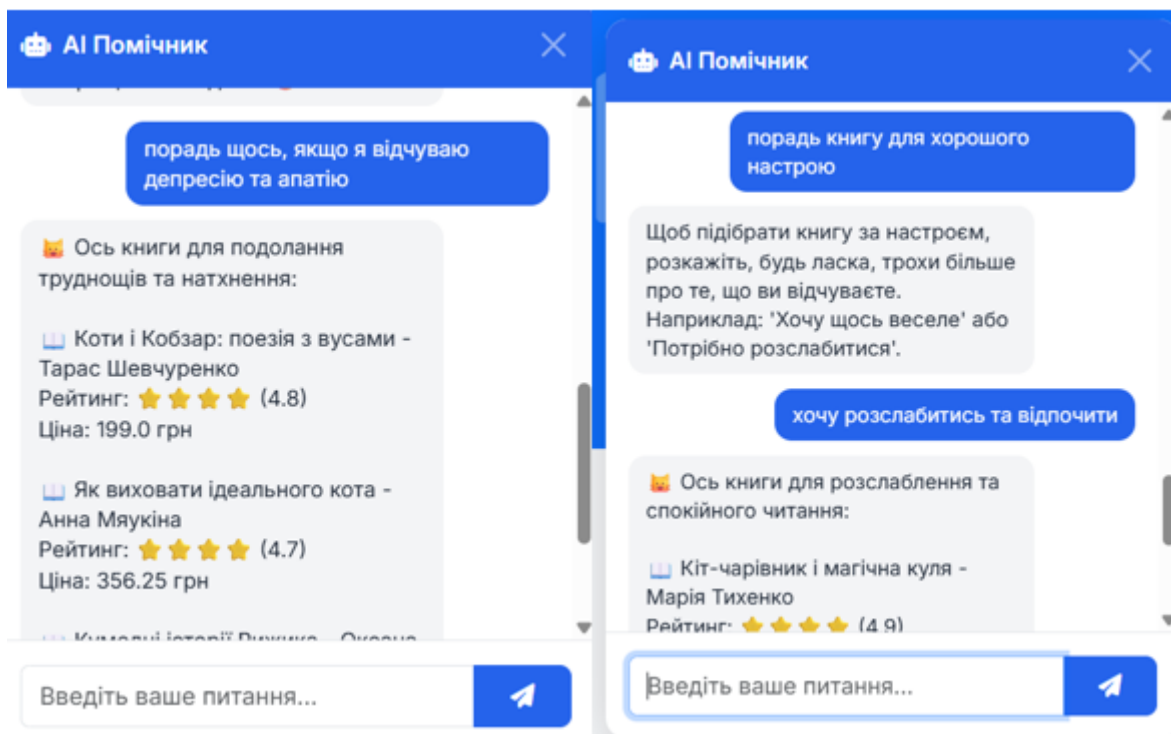


Рис. 3.12. Адаптивні рекомендації книг чат-ботом з урахуванням настрою користувача

Джерело: розроблено автором

На рис. 3.13. показано роботу методу `handle_genre_based_recommendations`, він отримує визначені жанри за допомогою `extract_genre_keywords` з повідомлення користувача та фільтрує книги з бази даних за відповідними `category_id`, після чого користувач отримує назви, авторів, рейтинги та ціни книг в улюбленому жанрі.

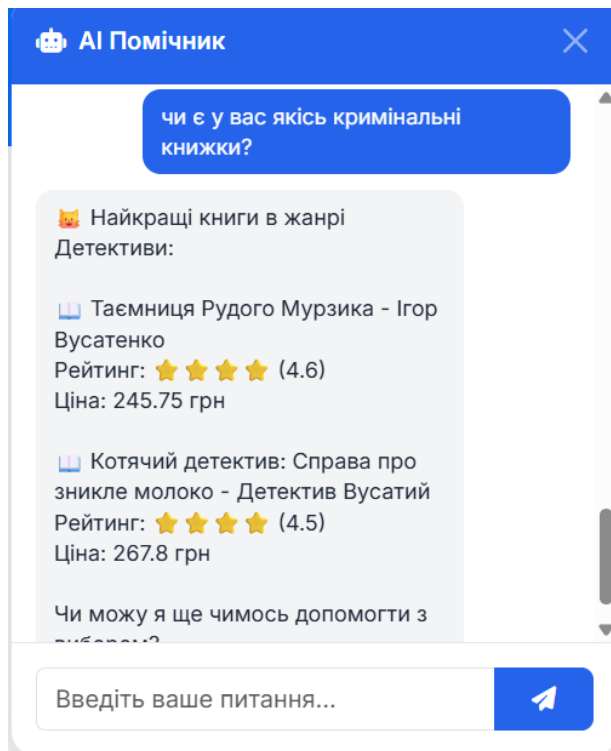


Рис. 3.13. Результат роботи функції `handle_genre_based_recommendations` у чат-боті

Джерело: розроблено автором

Метод `handle_popular_books` виводить список книг, які позначені як «бестселери» або мають найвищий рейтинг. Метод `handle_order_status` витягує номер замовлення з повідомлення та надає інформацію про його статус, дату, суму, доставку. Якщо користувач під час взаємодії одразу не вказує номер замовлення, то чат-бот просить його це зробити для надання релевантної відповіді, що й показано на рис. 3.14.

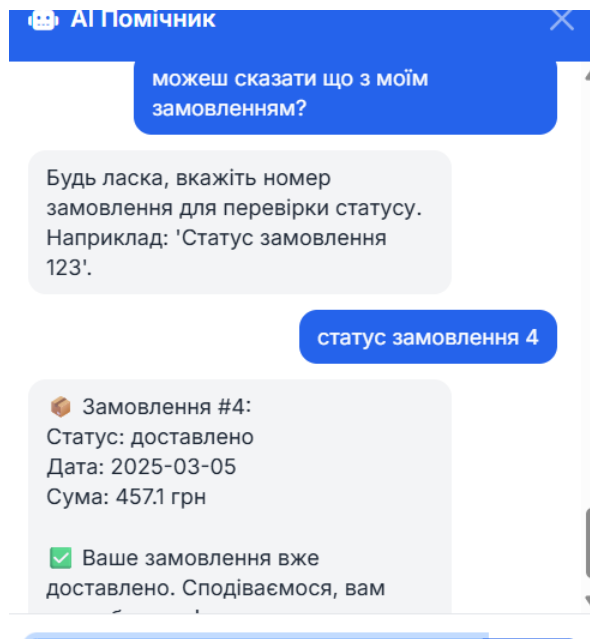


Рис. 3.14. Демонстрація роботи методу `handle_order_status` для перевірки статусу замовлення

Джерело: розроблено автором

Функція `handle_recommendations` виступає своєрідним «маршрутизатором» для рекомендацій. На рис 3.15. можна побачити, що спершу вона намагається знайти більш специфічні критерії (вік, настрій, жанр, популярність) у повідомленні користувача. Якщо такі виявлено, викликається відповідний, більш точний обробник.

```
def handle_recommendations(self, message):
    processed_message = self.preprocess_text(message)
    age_info_detected = self.extract_age(message)
    if age_info_detected:
        return self.handle_age_based_recommendations(message)
    moods_detected = self.detect_mood(message)
    if moods_detected:
        return self.handle_mood_based_recommendations(message)
    genres_detected = self.extract_genre_keywords(message)
    if genres_detected:
        return self.handle_genre_based_recommendations(message)
    if any(word in processed_message for word in self._get_normalized_forms(['популярні', 'бестселери', 'топ', 'хіти'])):
        return self.handle_popular_books(message)
    featured_books = [book for book in data['books'] if book.get('featured', False)]
    if not featured_books:
        featured_books = sorted(data['books'], key=lambda x: x['rating'], reverse=True)
    top_books = featured_books[:3]
```

Рис. 3.15. Фрагмент коду Python: функція `handle_recommendations` для персоналізованих рекомендацій чат-бота

Джерело: розроблено автором

Якщо ж запит є загальним, надається список найбільш рейтингових книг, а потім пропонується користувачу уточнити свої вподобання. Для простих інформаційних запитів (`delivery_info`, `payment_info`, `contact_info`), бот просто повертає одну з попередньо визначених випадкових відповідей, що робить взаємодію більш природною.

Функція `get_response` є головною точкою входу для взаємодії з ботом. Вона приймає повідомлення користувача, викликає `find_intent` для визначення найбільш ймовірного наміру, а потім перенаправляє запит до відповідної `handle`-функції для генерації відповіді. Якщо намір не визначено, бот відповідає загальною фразою, пропонуючи уточнити запит. Цей інтегрований підхід дозволяє чат-боту ефективно обробляти широкий спектр запитів, надаючи релевантні та корисні відповіді, імітуючи діалог з реальним співробітником.

Маршрути Flask забезпечують функціонал сайту, вони схожі на «адреси» сайту, які визначають, яка частина коду виконається, коли користувач заходить на певну сторінку або надсилає запит. Кожен маршрут створюється за допомогою декоратора `@app.route`, і під ним розміщується функція, що відповідає за обробку запиту та формування відповіді. Наприклад, коли користувач заходить на головну сторінку `/`, спрацьовує функція `index()`, яка підбирає популярні та рекомендовані книги й відображає їх через шаблон `index.html`, Рис. 3.16. показує код цієї функції.

```

@app.route('/')
def index():
    """Головна сторінка магазину"""
    featured_books = [book for book in data['books'] if book.get('featured', False)][:6]
    bestsellers = [book for book in data['books'] if book.get('bestseller', False)][:6]

    if not featured_books:
        featured_books = sorted(data['books'], key=lambda x: x['rating'], reverse=True)[:6]

    if not bestsellers:
        bestsellers = sorted(data['books'], key=lambda x: x['rating'], reverse=True)[6:12]

    return render_template('index.html',
                           featured_books=featured_books,
                           bestsellers=bestsellers,
                           categories=data['categories'])

```

Рис. 3.16. Серверна логіка Flask для відображення головної сторінки онлайн-магазину

Джерело: розроблено автором

Якщо користувач переходить у каталог `/catalog`, то функція `catalog()` фільтрує книги за жанром, пошуковим запитом чи ціною, і повертає відповідну сторінку з результатами. Маршрут `/book/` показує деталі книги, включаючи відгуки та схожі книги, а `/add_to_cart`, `/cart`, `/checkout` і `/order_success/` реалізують кошик і оформлення замовлень. Маршрут `/chat` відповідає за взаємодію з чат-ботом.

Сам інтерфейс користувача реалізовано за допомогою HTML-файлів, які використовують шаблонізатор Jinja2 для динамічного відображення даних з серверної частини. Файл `base.html` (див. додаток Б) є основним шаблоном, який визначає загальну структуру всіх сторінок веб-сайту. Він містить спільні елементи, такі як мета-теги для адаптивного відображення на різних пристроях, посилання на стилі (Bootstrap) та іконки (Font Awesome), а також загальні компоненти інтерфейсу, такі як навігаційна панель (`<nav>`) та нижній колонтитул (`<footer>`). Важливо зазначити, що саме у цьому файлі інтегровано основний JavaScript-код, що відповідає за інтерактивність сайту, зокрема за функціонал чат-бота та кошика покупок, забезпечуючи динамічну взаємодію користувачів з інтелектуальним помічником та функціоналом магазину. Рис. 3.17. показує реалізацію коду `index.html` (див. додаток В), що представляє собою головну

сторінку веб-сайту, яка розширює функціонал base.html, надаючи відвідувачам первинну інформацію про магазин та його основні пропозиції.

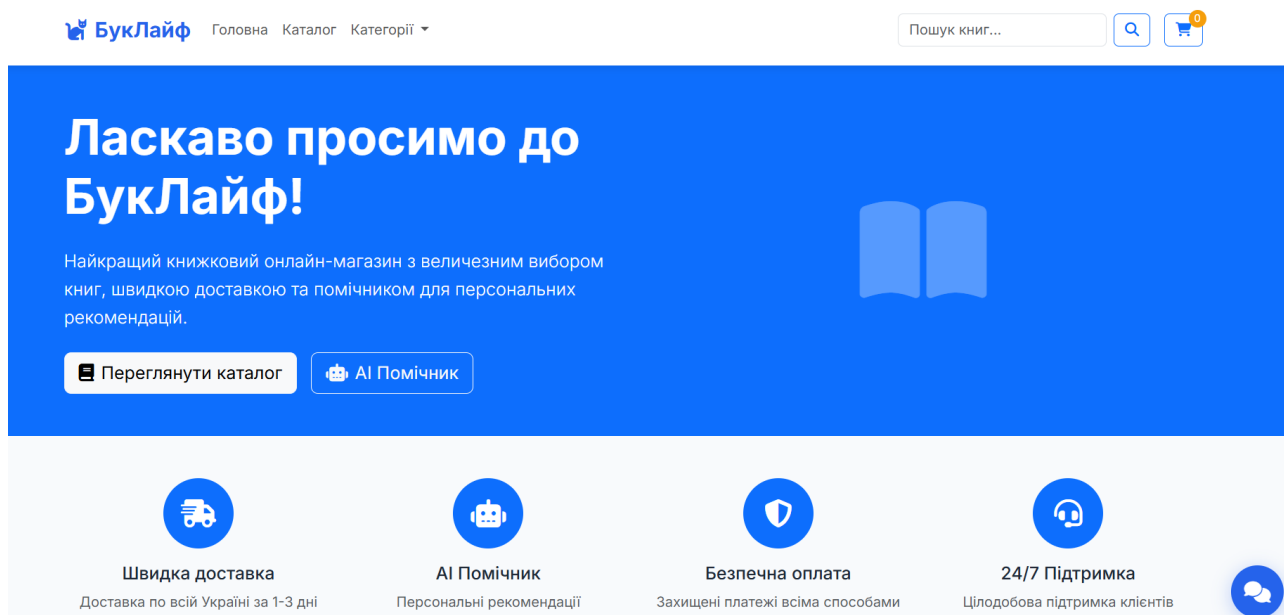


Рис. 3.17. Реалізація HTML-коду головної сторінки для користувачів

Джерело: розроблено автором

Відповідно файли catalog.html, book\_detail.html, cart.html, checkout.html, order\_success.html відповідають за інтерфейс сторінок які відображаються користувачам при переході до каталогу, детальнішому ознайомленні з конкретною книгою, переглядом кошику та оформленням замовлення. На рис. 3.18 показано що бачить користувач при переході на сторінку каталогу магазину. Ці файли відповідають виключно за візуальне представлення даних та взаємодію з користувачем на рівні інтерфейсу. Вони не містять складної бізнес-логіки чи алгоритмів, а лише відправляють запити на сервер та відображають отримані дані.

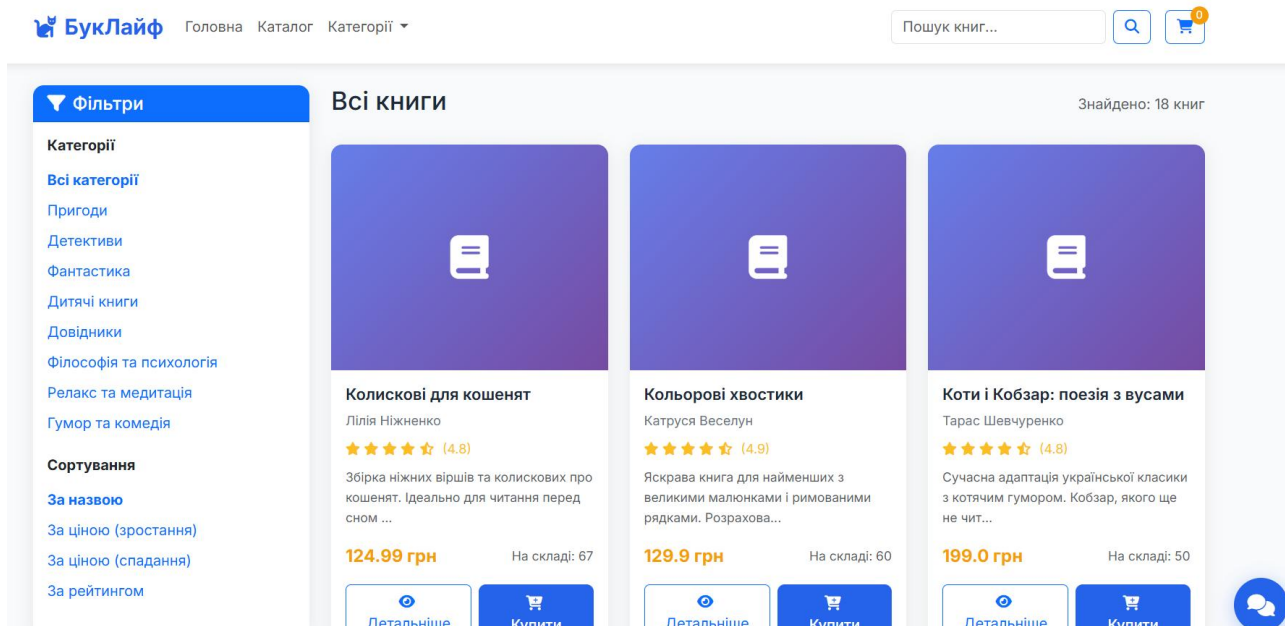


Рис. 3.18. Реалізація HTML-коду сторінки каталогу для користувачів

Джерело: розроблено автором

### 3.5. Модель впровадження інноваційного ІІІ-рішення.

Впровадження штучного інтелекту в бізнес-процеси потребує системного підходу, який охоплює послідовне виконання взаємопов'язаних етапів – від ідентифікації бізнес-задачі до постійного моніторингу та безперервного вдосконалення системи. Це не просто технічна інтеграція, а комплексна трансформація, що вимагає ретельного планування та адаптації бізнес-процесів. На рис. 3.19. можна побачити схему впровадження ІІІ, де добре видно всі етапи і те, що цей процес вимагає постійного вдосконалення.



Рис. 3.19. Схема етапів впровадження ІІІ-рішення

Джерело: розроблено автором

Початковим етапом для будь-якого ІІІ-рішення має стати чітке визначення бізнес-задачі та вибір найбільш доречного підходу для її розв'язання. На цій стадії ключовим завданням є ідентифікація конкретної проблеми, яку штучний інтелект здатен вирішити з максимальною економічною ефективністю. Для умовного магазину «БукЛайф», такою проблемою була необхідність розширення ринку збуту через онлайн-канал при одночасному дотриманні обмежень у витратах на клієнтську підтримку. З огляду на ці обставини, рішення розробити чат-бота, що базується на правилах, видається найбільш оптимальним для подібних малих та середніх підприємств.

Такий підхід, на відміну від складних нейромережових моделей, не вимагає значних обчислювальних ресурсів та об'ємних наборів даних для навчання, що робить його економічно вигідним для малого бізнесу. Показово, що правило-орієнтовані чат-боти забезпечують високу контрольованість відповідей та передбачуваність поведінки, що є важливим для бізнесу, який прагне підтримувати стабільний рівень обслуговування. З іншого боку, швидкість впровадження таких систем значно вища, оскільки вони не потребують тривалого періоду навчання моделі та постійного тонкого налаштування, що характерно для рішень на основі глибокого навчання. При впровадженні для реального бізнесу, навіть на етапі планування, необхідно враховувати не лише поточні, а й майбутні потреби у масштабуванні. Це означає, що попри вибір простого, але ефективного рішення на старті, необхідно закладати можливість його подальшої модернізації, наприклад, переходу до більш складних ШІ-моделей у міру зростання обсягів даних та запитів.

Наступним критично важливим етапом у процесі впровадження є збір, підготовка та структуризація даних, адже якість роботи будь-якої інтелектуальної системи безпосередньо залежить від інформації, на якій вона базується. Для даного проекту було створено централізовану базу даних у форматі JSON-файлу (`database.json`), що продемонструвало концептуальну можливість ефективної організації всієї необхідної інформації. Однак для реального бізнесу варто передбачити перехід до більш надійних та масштабованих рішень, як-от реляційні (PostgreSQL, MySQL) або NoSQL бази даних (MongoDB, Firestore).

Наступним кроком у моделі впровадження є розроблення безпосередньо інтелектуальної моделі, що в цьому випадку передбачає створення архітектури чат-бота, здатної обробляти природну мову та генерувати релевантні відповіді. Ядро розробленої системи, клас `WebsiteChatbot`, слугує прототипом та демонструє принципи, на яких може бути побудована повноцінна система.

Архітектура такої системи має включати декілька ключових компонентів, що працюють у взаємодії для забезпечення функціональності бота. Це модуль попередньої обробки тексту для очищення та нормалізації українських слів, що

є критично важливим для коректного розпізнавання намірів. Поряд із цим, необхідні модулі для вилучення специфічної інформації, як-от вік, настрої чи жанри, які аналізують текст на наявність конкретних паттернів. Центральне місце посідає система визначення намірів, що працює на основі скорингової моделі, оцінюючи відповідність вхідного повідомлення та намагаючись зрозуміти контекст запиту. Нарешті, після визначення наміру, управління передається спеціалізованим обробникам запитів, кожен з яких взаємодіє з базою даних для формування релевантної відповіді. При реалізації для живого бізнесу слід врахувати необхідність розробки надійних API для інтеграції бота з іншими корпоративними системами (CRM, ERP), а також залучення спеціалізованих розробників або партнерів, що мають досвід у створенні та впровадженні ШІ-рішень.

Етап впровадження та інтеграції є критично важливим, адже саме тут відбувається «вбудовування» розробленого ШІ-рішення у реальний бізнес-інструмент. У випадку інтернет-магазину, метою є створення безшовної та інтуїтивно зрозумілої взаємодії між чат-ботом та основними функціями інтернет-магазину.

Технічно інтеграція відбувається на двох рівнях: серверній стороні (Flask-додаток з API-ендпоінтом /chat) та клієнтській стороні (JavaScript-клас ChatBot для візуалізації та взаємодії). Для реального бізнесу важливо проаналізувати існуючу IT-інфраструктуру, спланувати поетапне розгортання (наприклад, через пілотні групи) та навчити персонал. Це забезпечить безперервність обслуговування та ефективне перенаправлення складних запитів. Результатом інтеграції ШІ-рішення в умовному магазині «БукЛайф» став сайт з впровадженим чат-ботом, який зображений на рис. 3.20. та з яким користувачі можуть легко взаємодіяти через привабливий інтерфейс.

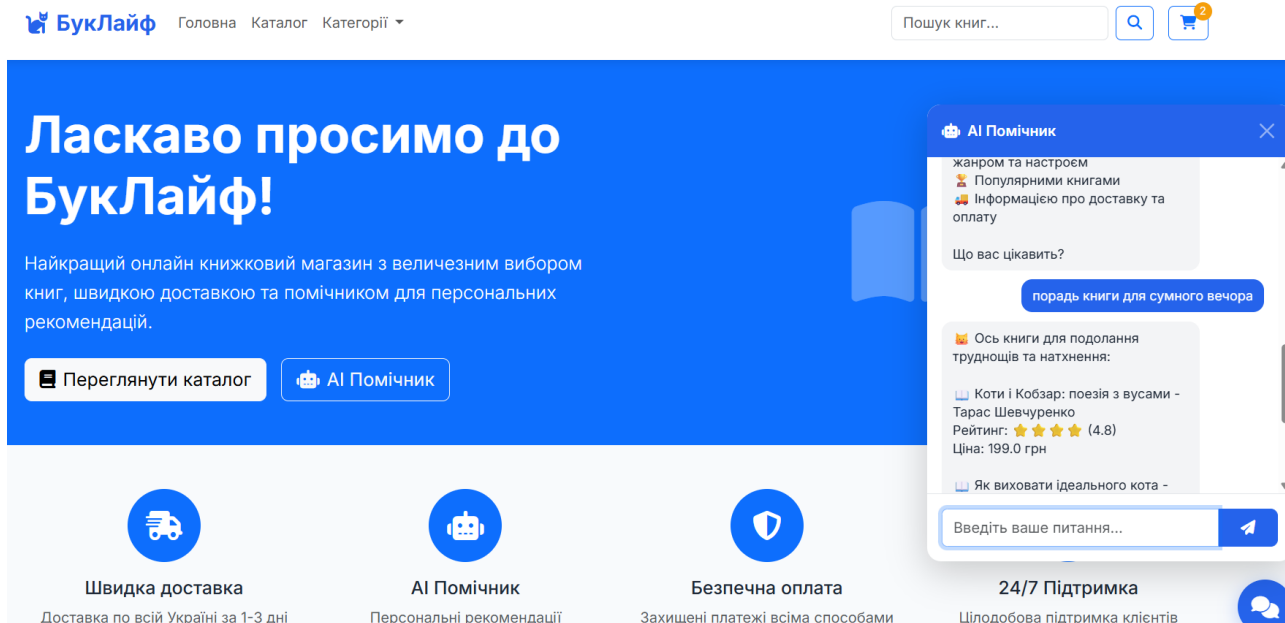


Рис. 3.20. Чат-бот, впроваджений в сайт магазину для взаємодії з клієнтами  
Джерело: розроблено автором

Впровадження ШІ-рішення не є фінальною точкою, а лише початком процесу його безперервного вдосконалення. Інтелектуальна система, як живий організм, потребує постійного моніторингу, аналізу та адаптації до динамічних умов зовнішнього середовища та мінливих потреб користувачів.

Для оцінки ефективності чат-бота необхідно відстежувати технічні (час відповіді, успішність запитів) та бізнес-орієнтовані метрики (відсоток нерозпізнаних намірів, середня тривалість діалогу, коефіцієнт конверсії), що демонструють його цінність. При впровадженні для реального бізнесу доцільно розробити аналітичні дашборди для оперативного реагування. Водночас, ключовою є організація циклу зворотного зв'язку (оцінка корисності, відгуки клієнтів і операторів). Діалоги, де бот не зміг дати адекватну відповідь, мають регулярно аналізуватися, і на основі цього база знань поповнюється новими правилами. Такий ітеративний підхід забезпечує постійне підвищення точності та цінності системи, дозволяючи їй еволюціонувати. Крім того, для реального бізнесу необхідно враховувати питання регуляторної відповідності, зокрема щодо захисту персональних даних клієнтів, що є важливим для уникнення правових ризиків.

Розроблений проект демонструє життєздатність та ефективність інтелектуального чат-бота на основі правил як доступного ШІ-рішення для малого та середнього бізнесу. Проте, важливо розуміти, що це лише початковий етап. Для реального бізнесу з плином часу та зростанням обсягів даних може виникнути потреба у подальшій еволюції системи. Це може включати перехід від простого правило-орієнтованого підходу до використання більш складних моделей машинного навчання та обробки природної мови, таких як трансформерні архітектури, для кращого розуміння контексту та генерації більш гнучких відповідей. Також, у перспективі, може бути розглянута інтеграція чат-бота з іншими каналами комунікації (наприклад, месенджерами, соціальними мережами) та розширення його функціоналу до повноцінного віртуального асистента, здатного виконувати транзакції, обробляти складніші запити та надавати більш глибокі персоналізовані рекомендації. Таким чином, розроблена модель слугує фундаментом, який може бути адаптований та масштабований відповідно до зростаючих потреб та стратегічних цілей реального підприємства.

### *3.6. Розрахунок потенційного економічного ефекту.*

Впровадження інноваційного ШІ-рішення, такого як інтелектуальний чат-бот, в операційну діяльність онлайн-книжкового магазину «БукЛайф» передбачає не лише покращення клієнтського досвіду, а й значний потенційний економічний ефект. Оцінка цього ефекту має вирішальне значення для обґрунтування інвестицій та демонстрації цінності технологічних інновацій. Потенційний економічний ефект може бути розглянутий з кількох ключових перспектив: зниження операційних витрат, зростання доходу, а також підвищення ефективності маркетингу та збору даних.

Однією з найбільш очевидних та безпосередньо вимірних переваг від впровадження чат-бота є скорочення витрат на клієнтську підтримку. Чат-бот здатний автоматично обробляти значну частину типових та повторюваних запитів клієнтів, що включають питання щодо статусу замовлень, умов доставки та оплати, наявності книг, базових рекомендацій. Припустимо, що в «БукЛайф» в середньому надходить 1000 запитів на місяць, з яких 70% є типовими. Якщо

бот обробляє 85% цих типових запитів, це означає скорочення навантаження на операторів на  $1000 \times 0,7 \times 0,85 = 595$  запитів на місяць. Якщо кожен оператор служби підтримки обробляє, наприклад, 300 запитів на місяць, автоматизація 595 запитів може потенційно вивільнити ресурс, еквівалентний  $595/300 \approx 1,98$  оператора. При середній місячній заробітній платі оператора в Україні (приблизно 17 500 грн) це може призвести до економії  $1,98 \times 17500 \approx 34650$  грн на місяць, або  $34650 \times 12 \approx 415800$  грн на рік. Навіть якщо це не призводить до негайного скорочення штату, це дозволяє перерозподілити робочий час операторів на більш складні та індивідуальні запити, підвищуючи якість підтримки.

Завдяки миттєвим та релевантним відповідям на запитання, а також персоналізованим рекомендаціям, чат-бот може значно покращити шлях клієнта до покупки. Користувачі, які отримують швидку допомогу у виборі книги, з більшою ймовірністю здійнять покупку. Якщо поточний коефіцієнт конверсії «БукЛайф» становить 3%, і бот дозволить збільшити його на 1% (до 4%), припустимому місячному трафіку в 2000 відвідувачів та середньому чеку в 300 грн, це призведе до додаткових продажів в розмірі  $2000 \times (0,04 - 0,03) \times 300 = 6000$  грн на місяць, або 72000 грн на рік.

Інтелектуальні рекомендації, що ґрунтуються на настрої, віці або жанрових вподобаннях, можуть спонукати клієнтів до покупки додаткових, можливо, не запланованих, книг. Якщо середній чек зростає на 3%, а річний обсяг продажів після підвищення конверсії становить  $2000 \times 0,04 \times 300 \times 12 = 288\ 000$  грн, то додатковий дохід від цього ефекту дорівнює  $288\ 000 \times 0,03 = 8640$  грн на рік.

Тепер підрахуємо витрати на розробку та підтримку ШІ-рішення (веб-сайт з чат-ботом). Припустимо що розробка сайту з чат-ботом, делегована фрілансеру, орієнтовно займе 250 годин. При середній вартості години роботи фрілансера-розробника в Україні 400 грн, початкові інвестиції становитимуть  $250 \times 400 = 100\ 000$  грн. Також для роботи невеликого сайту та бази даних потрібен віртуальний хостинг, витрати на який близько 250 грн/місяць ( $250 \times 12 = 3000$  грн/рік). Візьмемо до уваги і витрати на періодичні оновлення функціоналу,

виправлення помилок для чат-бота. Припустимо, на оновлення знадобиться 50 годин на рік, які також делегуємо на фріланс за ставкою 400 грн/год, це становитиме  $50 \times 400 = 20000$  грн/рік. Отже, загальні річні операційні витрати складуть  $3000 + 20000 = 23000$  грн.

Загальна річна вигода від впровадження чат-бота включає економію на кадрових ресурсах (415800 грн), додаткові продажі від підвищення конверсії (72000 грн) та збільшення середнього чека (8640 грн), що дає  $415800 + 72000 + 8640 = 496440$  грн. Після вирахування операційних витрат чиста вигода становить  $496440 - 23000 = 473440$  грн.

Підрахуємо рентабельність інвестицій (ROI):

$$\text{ROI} = \frac{\text{Прибуток} - \text{Початкові інвестиції}}{\text{Початкові інвестиції}} = \frac{473440 - 100000}{100000} = 373,44\%$$

Такий високий показник ROI свідчить про значну ефективність та швидку віддачу від вкладених коштів.

Період окупності визначає час, необхідний для того, щоб початкові інвестиції повністю окупилися за рахунок отриманої чистої вигоди.

$$\text{Період окупності} = \frac{\text{Початкові інвестиції}}{\text{Чиста річна вигода}} = \frac{100000}{473400} = 0,211 \text{ року}$$

$0,211 \times 12 \approx 2,53$  місяці, тобто період окупності становить приблизно два з половиною місяці.

Розрахований показник рентабельності інвестицій та період окупності підтверджують високу ефективність і доцільність впровадження ШІ-рішень для малого бізнесу. Отримані результати свідчать про значний потенціал подібних інновацій для оптимізації бізнес-процесів, підвищення конкурентоспроможності та фінансової стійкості підприємства. Це створює підґрунтя для подальшого масштабування моделі, розширення функціоналу чат-бота та адаптації аналогічних рішень для інших секторів малого та середнього бізнесу в Україні.

## ВИСНОВКИ

У сучасних умовах стрімкого розвитку цифрових технологій та глобалізаційних процесів, питання інноваційного моделювання бізнесу з використанням штучного інтелекту набуває особливої актуальності. Саме тому виконане дипломне дослідження було спрямоване на комплексний розгляд теоретичних та прикладних аспектів впровадження ШІ у бізнес-середовище, обґрунтування доцільності його використання та розробку практичних рекомендацій для підприємств.

За результатами досліджень першого розділу було ретельно систематизовано та узагальнено наукові погляди на сутність, ключові характеристики та класифікацію штучного інтелекту. Це дозволило чітко окреслити його трансформаційну роль у сучасних бізнес-моделях, зокрема у сферах предиктивної аналітики, оптимізації операційної діяльності, персоналізації взаємодії з клієнтами та ефективної боротьби з шахрайством. При цьому, було виявлено та проаналізовано як значні переваги впровадження ШІ, такі як підвищення загальної ефективності та прискорення прийняття рішень, так і потенційні виклики, що потребують уваги при його імплементації.

Далі, у другому розділі, було проведено детальний аналіз особливостей впровадження штучного інтелекту в бізнес-діяльність. Використання системного підходу дозволило виокремити ключові етапи та тенденції, включаючи поширення генеративного ШІ та концепції AI-as-a-Service. Огляд найпопулярніших сервісів штучного інтелекту, таких як ChatGPT, Microsoft Azure AI, Google Cloud AI та AWS AI/ML Services, продемонстрував розмаїття функціоналу та реальних можливостей для бізнесу, обґрунтовуючи доцільність їх використання.

Третій розділ містить аналіз інноваційної бізнес-моделі компанії Amazon як яскравого прикладу успішної інтеграції ШІ в електронну комерцію, логістику та взаємодію з клієнтами. Оцінка ефективності ШІ для оптимізації бізнесу проводилася з використанням як традиційних метрик якості моделей (точності, прецизійності, повноти та F1-міри), так і ключових економічних показників -

рентабельності інвестицій (ROI) та періоду окупності. На основі отриманих результатів, для малих та середніх підприємств України запропоновано та програмно реалізовано модель впровадження інтелектуального чат-бота для онлайн-магазину на мові Python з використанням фреймворку Flask. Це рішення, яке дозволяє автоматизувати клієнтську підтримку та надавати персоналізовані рекомендації, демонструє практичну ефективність застосування ШІ навіть для підприємств з обмеженими ресурсами.

Таким чином, проведене дослідження підтверджує доцільність та ефективність застосування ШІ в умовах сучасного бізнесу. Штучний інтелект є потужним інструментом для підвищення конкурентоспроможності, операційної ефективності та адаптивності підприємств до швидкозмінних ринкових умов. Він дозволяє автоматизувати рутинні процеси, підвищити точність прогнозів, персоналізувати взаємодію з клієнтами та оптимізувати використання ресурсів, що є критично важливим для виживання та розвитку компаній.

Розроблені в роботі рекомендації та практична модель чат-бота можуть бути безпосередньо використані малими та середніми підприємствами в Україні для впровадження доступних ШІ-рішень. Отримані розрахунки потенційного економічного ефекту слугуватимуть аргументацією для інвестицій у цифрову трансформацію. У перспективі, розроблену модель можна масштабувати та модернізувати, інтегруючи складніші алгоритми машинного навчання, розширюючи функціонал до повноцінного віртуального асистента та адаптуючи її для інших галузей, що потребують оптимізації клієнтської взаємодії.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1955). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence// AI Magazine. – 1955. – Vol. 27, No. 4.
2. Russell, S. J., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.).
3. Згуровський М. З., Хіміч О. М. Штучний інтелект: зміна парадигми у фундаментальних науках (Нобелівські премії з фізики і хімії 2024 року). Вісник Національної академії наук України. 2024. № 12.
4. Концепція розвитку штучного інтелекту в Україні, 2020 // Кабінет Міністрів України. <https://zakon.rada.gov.ua/go/1556-2020-%D1%80>
5. Siegel E., (2013). Predictive analytics: The power to predict who will click, buy, lie, or die. John Wiley & Sons.
6. Transforming Financial Services: The Impact of AI on JP Morgan Chase’s Operational Efficiency and Decision-Making. International Journal of Scientific Research & Engineering Trends. – 2024. – Vol. 10, Issue 1.
7. SmartDev. The Impact of Conversational AI on Customer Experience. URL: <https://smartdev.com/the-impact-of-conversational-ai-on-customer-experience/>.
8. How AI is Reshaping Fraud Detection in Payments. The Global Treasurer. 2024. URL: <https://www.theglobaltreasurer.com/2025/03/21/how-ai-is-reshaping-fraud-detection-in-payments/>.
9. Київстар Business Hub. Галузеві тренди. Штучний інтелект в Україні: як розвивається галузь. URL: <https://hub.kyivstar.ua/articles/galuzevi-trendi-shtuchnij-intelekt-v-ukrayini-yak-rozvivayetsya-galuz>.
10. Deloitte Analytics Institute. Deloitte Predictive Maintenance Position Paper, 2022. URL: <https://www2.deloitte.com/us/en/pages/operations/articles/predictive-maintenance-and-the-smart-factory.html>.
11. Sustainability in Industry 5.0: Theory and Applications, Edited by C Kishor Kumar Reddy, P. R. Anisha, Samiya Khan, Marlia Mohd Hanafiah, Lavanya Pamulaparty, R Madana Mohana

12. Karan Gupta, Prabhjyot Kaur, Application of Predictive Maintenance in Manufacturing with the utilization of AI and IoT Tools. URL: <https://www.techrxiv.org/users/870921/articles/1251882-application-of-predictive-maintenance-in-manufacturing-with-the-utilization-of-ai-and-iot-tools>.
13. Chirinos, J. Case Study: JPMorgan Chase's Contract Intelligence (COiN) platform for Document Analysis, 2024.
14. Esteva, A., Chou, K., Yeung, S., Naik, N., Madani, A., Mottaghi, A., Liu, Y., Topol, E., Dean, J., & Socher, R. Deep learning-enabled medical computer vision, 2021.
15. Artificial intelligence (AI) in personalized medicine: AI-generated personalized therapy regimens based on genetic and medical history: Short communication. *Annals of Medicine and Surgery*. URL: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10617817/>.
16. World Economic Forum. (2025). Artificial intelligence in financial services: <https://www.weforum.org/reports/artificial-intelligence-in-financial-service>.
17. Dimple Patil, Artificial intelligence in retail and e-commerce: Enhancing customer experience through personalization, predictive analytics, and real-time engagement. 2024.
18. Lindquist M., AI in logistics: Potential benefits and applications. Oracle. URL: <https://www.oracle.com/ua/scm/ai-in-logistics/>.
19. D2L. (2024). Brightspace Data Analytics: Impact on student success. D2L Research & Analytics.
20. How Artificial Intelligence is Reshaping Global Education, *Modern Diplomacy*, 2024. URL: <https://moderndiplomacy.eu/2024/04/30/how-artificial-intelligence-is-reshaping-global-education/>.
21. Tsou, C. Implementing Artificial Intelligence (AI) in Higher Education: A Narrative Literature Review, 2023. URL: [https://www.researchgate.net/publication/376553922\\_Implementing\\_Artificial\\_Intelligence\\_AI\\_in\\_Higher\\_Education\\_A\\_Narrative\\_Literature\\_Review](https://www.researchgate.net/publication/376553922_Implementing_Artificial_Intelligence_AI_in_Higher_Education_A_Narrative_Literature_Review).
22. Piddubna A, AI in Agriculture — The Future of Farming, 2024. URL: <https://intellias.com/artificial-intelligence-in-agriculture/>.

23. Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy: [https://www.researchgate.net/publication/335435943\\_Artificial\\_Intelligence\\_AI\\_Multidisciplinary\\_perspectives\\_on\\_emerging\\_challenges\\_opportunities\\_and\\_agenda\\_for\\_research\\_practice\\_and\\_policy](https://www.researchgate.net/publication/335435943_Artificial_Intelligence_AI_Multidisciplinary_perspectives_on_emerging_challenges_opportunities_and_agenda_for_research_practice_and_policy).
24. Forbes Business Council. Understanding The Benefits And Risks Of Using AI In Business, 2023. URL: <https://www.forbes.com/councils/forbesbusinesscouncil/2023/03/01/understanding-the-benefits-and-risks-of-using-ai-in-business/>.
25. Alawamlen M., Shammam N., Alawamlen K., Examining the limitations of AI in business and the need for human oversight, 2024
26. Андреев, А. Впровадження ШІ у бізнес-процеси, Apix-Drive, 2024. URL: <https://apix-drive.com/ua/blog/useful/vprovadzhenja-shi-u-biznes-procesi>
27. Shevko D., How Ukraine is setting new standards in digital innovation, The New Voice of Ukraine, 2025. URL: <https://english.nv.ua/business/how-ai-digitalization-and-innovation-are-reshaping-ukrainian-business-and-banking-50498278.html>.
28. AiBoost. Integration Challenges of AI in Businesses, the Future Prospect of the Industry / AiBoost, 2024. URL: <https://aiboost.co.uk/integration-challenges-of-ai-in-businesses-overcoming-hurdles-in-aligning-ai-with-existing-tech-infrastructures/>.
29. Gandel, S. Zillow, facing big losses, quits flipping houses and will lay off a quarter of its staff, The New York Times. 2021. URL: <https://www.nytimes.com/2021/11/02/business/zillow-q3-earnings-home-flipping-ibuying.html>.
30. Reface is cutting 35% of its workforce, affecting 60 employees , AIN.ua., 2024. URL: <https://en.ain.ua/2024/12/03/reface-is-cutting-60-employees/>.
31. Gartner. Gartner Survey Shows 37 Percent of Organizations Have Implemented AI in Some Form, Gartner Newsroom, 2019. URL: <https://www.gartner.com/en/newsroom/press-releases/2019-01-21-gartner-survey-shows-37-percent-of-organizations-have>.

32. Singla A., Sukharevsky A., Yee L., Chui M., & Hall B., The State of AI: How Organizations Are Rewiring to Capture Value, McKinsey & Company, 2025. URL: [https://www.mckinsey.com/~/media/mckinsey/business%20functions/quantumblack/our%20insights/the%20state%20of%20ai/2025/the-state-of-ai-how-organizations-are-rewiring-to-capture-value\\_final.pdf](https://www.mckinsey.com/~/media/mckinsey/business%20functions/quantumblack/our%20insights/the%20state%20of%20ai/2025/the-state-of-ai-how-organizations-are-rewiring-to-capture-value_final.pdf).
33. InfoWorld. 14 popular AI algorithms and their uses. URL: <https://www.infoworld.com/article/2338418/14-popular-ai-algorithms-and-their-uses.html>.
34. Tavakoli A., Harreis H., Rowshankish K., Charting a path to the data- and AI-driven enterprise of 2030, McKinsey & Company, 2024. URL: <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/charting-a-path-to-the-data-and-ai-driven-enterprise-of-2030>.
35. Artificial Intelligence As A Service market size, share & trends analysis report, Grand View Research, 2024. URL: <https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-as-a-service-market-report>.
36. Siegel E., To deploy machine learning, you must manage operational change - Here Is How UPS Got It Right, Harvard Data Science Review, 2023. URL: <https://hdsr.mitpress.mit.edu/pub/2z4dnhds/release/1>.
37. Mayer, H., Yee, L., Chui, M., & Roberts, R. Superagency in the workplace: Empowering people to unlock AI's full potential, McKinsey & Company, 2025. URL: <https://legalinstruments.oecd.org/en/instruments/oecd-legal-0449>.
38. Organisation for Economic Co-operation and Development (OECD). Recommendation of the Council on Artificial Intelligence. URL: <https://legalinstruments.oecd.org/en/instruments/oecd-legal-0449>.
39. Marr B. 10 Amazing Real-World Examples of How Companies Are Using ChatGPT in 2023. Forbes, 2023. URL: <https://www.forbes.com/sites/bernardmarr/2023/05/30/10-amazing-real-world-examples-of-how-companies-are-using-chatgpt-in-2023/>.
40. OpenAI. Platform documentation / OpenAI. URL: <https://platform.openai.com/docs/concepts> <https://arxiv.org/pdf/1706.03762>.

41. Microsoft Azure. Platform overview. URL: <https://azure.microsoft.com/en-us/explore/>.
42. Google Cloud. Platform overview. URL: <https://cloud.google.com/>.
43. Amazon SageMaker. What is SageMaker? Amazon Web Services. URL: <https://docs.aws.amazon.com/sagemaker/latest/dg/whatis.html>.
44. Amazon SageMaker. AI features. Amazon Web Services. URL: <https://aws.amazon.com/sagemaker-ai/features/>.
45. IBM. Watsonx-ai product page. URL: <https://www.ibm.com/products/watsonx-ai>
46. FourKites. Platform. URL: <https://www.fourkites.com/>.
47. The Three Key Benefits of Drone Scouting. Taranis. URL: <https://www.taranis.com/blogs/the-three-key-benefits-of-drone-scouting/>.
48. AI Physician Mode Study. K Health. URL: <https://khealth.com/ai-physician-mode-study>.
49. Baeldung. Amazon Recommendation System. URL: <https://www.baeldung.com/cs/amazon-recommendation-system>.
50. Amazon. About Us. URL: <https://www.aboutamazon.com/about-us>.
51. Amazon Developer. Alexa Documentation. URL: <https://developer.amazon.com/en-US/alexa>.
52. Cisco. Cisco AI Readiness Index 2024. URL: [https://www.cisco.com/c/dam/m/en\\_us/solutions/ai/readiness-index/2024-m11/documents/cisco-ai-readiness-index.pdf](https://www.cisco.com/c/dam/m/en_us/solutions/ai/readiness-index/2024-m11/documents/cisco-ai-readiness-index.pdf).
53. Microsoft Tech Community. A Framework for Calculating ROI for Agentic AI Apps 2024. URL: <https://techcommunity.microsoft.com/blog/machi>.
54. Neontri. How to Measure AI Performance. URL: <https://neontri.com/blog/measure-ai-performance/>.

## ДОДАТКИ

додаток А

Повний код сайту онлайн-магазину «БукЛайф» з чат-ботом (файл app.py)

```
import json
import re
import os
import logging
from datetime import datetime
from flask import Flask, request, jsonify, render_template, session, redirect,
url_for, flash
from werkzeug.security import generate_password_hash, check_password_hash
from werkzeug.utils import secure_filename
from collections import defaultdict
import uuid
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger("BukLifeStore")
app = Flask(__name__)
data = {}
shopping_cart = defaultdict(list)
def load_database():
    global data
    try:
        with open('database.json', 'r', encoding='utf-8') as f:
            data = json.load(f)
        logger.info("Базу даних успішно завантажено.")
    except FileNotFoundError:
        logger.error("ПОМИЛКА: Файл database.json не знайдено! Створюю
порожню базу даних.")
        data = {"books": [], "clients": [], "orders": [], "categories": [], "reviews": [],
"chatbot_training_data": []}
```

```

except json.JSONDecodeError as e:
    logger.error(f'ПОМИЛКА: Некоректний формат JSON у database.json:
    {e}')
    data = {"books": [], "clients": [], "orders": [], "categories": [], "reviews": [],
    "chatbot_training_data": []}

def save_database():
    try:
        with open('database.json', 'w', encoding='utf-8') as f:
            json.dump(data, f, ensure_ascii=False, indent=2)
            logger.info("Базу даних успішно збережено.")
    except Exception as e:
        logger.error(f'ПОМИЛКА: Не вдалося зберегти базу даних: {e}')

class WebsiteChatbot:
    def __init__(self, training_data):
        self.intents = training_data
        self.context = {}
        self.ukr_endings = ['a', 'и', 'е', 'о', 'у', 'і', 'я', 'ю', 'є', 'ї', 'ь',
            'им', 'ій', 'ою', 'их', 'их', 'ого', 'ому', 'ій',
            'ти', 'ся', 'ої', 'ій', 'ами', 'ями']
        self.enhanced_intents = [
            {
                "intent": "greeting",
                "patterns": self._get_normalized_forms([
                    "привіт", "добрий день", "здравствуйте", "вітаю", "hello", "hi",
                    "доброго дня",
                    "доброго ранку", "добрий вечір", "слухаю", "я тут", "доброго"
                ]),
                "responses": [

```

```

    "Привіт! Вітаю в книжковому магазині БукЛайф! 🐱 Як можу
допомогти?"
    ]
  },
  {
    "intent": "book_search",
    "patterns": self._get_normalized_forms([
      "шукаю книгу", "знайти книгу", "є така книга", "пошук",
"потрібна книга",
      "назва книги", "автор", "маю книгу", "знайти", "шукати", "книга",
"автора", "назву", "шукатиму"
    ]),
    "responses": [
      "З радістю допоможу знайти книгу! Скажіть назву або автора.",
      "Який жанр книг або автор вас цікавить?",
      "Можете назвати автора чи тему книги?"
    ]
  },
  {
    "intent": "mood_based_recommendations",
    "patterns": self._get_normalized_forms([
      "настрій", "почуття", "емоція", "настрою", "настроєм",
"самопочуття",
      "веселе", "сумний", "розслабитися", "енергія", "відпочити", "сум",
      "щасливий", "пригнічений", "радісний", "втомлений",
"натхнення", "смішне", "цікаве"
    ]),
    "responses": [
      "Розумію ваш настрій! Підберу ідеальну книгу.",
      "Для такого настрою маю відмінні варіанти історій!",

```

"Знаю саме те, що вам потрібно для приємного читання!",  
 "Щоб підібрати книгу за настроєм, розкажіть трохи більше про те, що ви відчуваєте.",  
 "Давайте знайдемо книгу, яка ідеально доповнить ваш поточний настрій!"

```

    ]
  },
  {
    "intent": "age_based_recommendations",
    "patterns": self._get_normalized_forms([
      "дитини", "років", "дитячі книги", "підліток", "малюк", "школяр",
      "дитині", "хлопчику", "дівчинці", "віку", "класу", "молодший",
    "старший",
      "дітям", "для дітей", "який вік", "дідусь", "бабуся", "дорослий"
    ]),
    "responses": [
      "Чудово! Я підберу відповідні книги за віком.",
      "Скажіть, будь ласка, точний вік дитини для кращого підбору.",
      "Маю чудові дитячі книги для різних вікових груп!",
      "Для якого віку ви шукаєте книгу? Це допоможе мені точніше підібрати.",
      "Книги для дітей та підлітків у нас є! Назвіть вік, будь ласка."
    ]
  },
  {
    "intent": "genre_based_recommendations",
    "patterns": self._get_normalized_forms([
      "пригодницький", "пригоди", "детектив", "фентезі", "комедія",
    "біографія",
      "роман", "новела", "класика", "навчальна", "довідник", "казка",
  
```

"фантастика", "трилер", "історія", "наука", "поезія", "драма",  
 "лірика",  
 "жахи", "містика", "любобний", "романтичний", "психологія",  
 "філософія"

]),

"responses": [

"Чудовий вибір жанру! Маю відмінні книги в цьому стилі.",

"Цей жанр дуже популярний! Ось найкращі варіанти.",

"Люблю цей жанр! Підберу для вас найкращі історії.",

"Назвіть ваш улюблений жанр, і я знайду щось цікаве!",

"У нас великий вибір за жанрами. Що саме вас цікавить?"

]

},

{

"intent": "popular\_books",

"patterns": self.\_get\_normalized\_forms([

"популярні", "найкращі", "топ", "хіти", "бестселери", "модні",

"найпопулярніші", "рейтинг", "найкраще", "що читають",

"тренди", "попит"

]),

"responses": [

"Ось найпопулярніші книги цього сезону!",

"Маю для вас топ найкращих бестселерів!",

"Ці книги зараз на піку популярності!",

"Хочете дізнатися, що зараз читають усі? Ось наш рейтинг!",

"Наші бестселери завжди до ваших послуг!"

]

},

{

"intent": "recommendations",

```

"patterns": self._get_normalized_forms([
    "порадьте книгу", "що почитати", "рекомендації", "цікаві книги",
    "новинки", "найкращі книги", "підберіть", "варіанти",
    "хочу щось", "обрати", "вибрати", "подивитися", "щось нове",
"для себе"
]),
"responses": [
    "З радістю порекомендую книги! Який жанр вам подобається?",
    "Маємо багато цікавих новинок. Що вас більше цікавить?",
    "Чи є у вас улюблені автори книг або жанри?",
    "Розкажіть, що вам до душі, і я підберу ідеальні варіанти!",
    "Для кращої рекомендації, скажіть, будь ласка, ваш улюблений
жанр, або настрої, або вік, для кого шукаєте книгу."
]
},
{
    "intent": "order_status",
    "patterns": self._get_normalized_forms([
        "статус замовлення", "замовлення", "де моє замовлення", "коли
прийде", "відстежити", "номер замовлення",
        "замовлення номер", "моє замовлення", "стан замовлення"
    ]),
    "responses": [
        "Щоб перевірити статус замовлення книг, назвіть номер
замовлення.",
        "Для відстеження замовлення потрібен його номер. Який у вас
номер?",
        "Будь ласка, повідомте номер замовлення для перевірки статусу.",
        "Введіть номер вашого замовлення, і я допоможу його
відстежити."
    ]
}

```

```

    ]
  },
  {
    "intent": "delivery_info",
    "patterns": self._get_normalized_forms([
      "доставка", "скільки часу доставка", "як отримати", "коли
прийде", "кур'єр",
      "вартість доставки", "самовивіз", "доставляєте", "доставити"
    ]),
    "responses": [
      "Доставка книг по Україні 1-3 робочих дні. Експрес-доставка 1
день.",
      "Стандартна доставка 2-4 дні, експрес 1-2 дні. Самовивіз
безкоштовно.",
      "Доставляємо книги по всій Україні. Київ та область - наступний
день.",
      "Ми доставляємо Новою Поштою та Укрпоштою. Вартість
розраховується за тарифами перевізника.",
      "Можливий самовивіз з нашого магазину за адресою: вул. Котяча,
12, Київ."
    ]
  },
  {
    "intent": "payment_info",
    "patterns": self._get_normalized_forms([
      "оплата", "як заплатити", "способи оплати", "карткою", "готівка",
      "розрахунок", "оплатити", "платіж", "розрахуватися"
    ]),
    "responses": [

```

"Приймаємо оплату карткою, готівкою при отриманні та банківським переказом.",

"Можна оплатити онлайн або готівкою кур'єру.",

"Підтримуємо всі популярні способи оплати: Visa, MasterCard, ПриватБанк.",

"Ви можете оплатити замовлення при отриманні або ж онлайн на сайті."

]

}

]

self.mood\_categories = {

"good\_mood": {

"keywords": self.\_get\_normalized\_forms([

"гарний", "чудовий", "прекрасний", "радісний", "щасливий",

"ентузіазм",

"позитив", "веселий", "піднесений", "оптимістичний", "смішне",

"кумедне",

"радість", "щастя", "позитивний", "піднесений"

]),

"tags": ["гумор", "комедія", "пригоди", "легке читання", "натхнення",

"оптимізм", "пригоди"],

"description": "для піднесення настрою та легкості"

},

"bad\_mood": {

"keywords": self.\_get\_normalized\_forms([

"поганий", "жахливий", "жалісний", "сумний", "смутний",

"депресивний",

"пригнічений", "важкий", "втомлений", "розчарований", "апатія",

"сум",

```

    "туга", "втома", "безнадія", "розгубленість"
  ],
  "tags": ["мотивація", "самодопомога", "психологія", "філософія",
"біографії успішних людей",
    "надихаючі історії", "саморозвиток", "історії успіху"],
  "description": "для подолання труднощів та натхнення"
},
"relax_mood": {
  "keywords": self._get_normalized_forms([
    "розслабитися", "відпочити", "релакс", "спокій", "тиша",
"медитація",
    "вечір", "ніч", "спокійний", "заспокоїти", "відволіктися", "спокій"
  ]),
  "tags": ["релаксація", "медитація", "природа", "поезія", "лірика",
    "художня література", "фантастика", "легке читання",
"духовний розвиток"],
  "description": "для розслаблення та спокійного читання"
},
"children_mood": {
  "keywords": self._get_normalized_forms([
    "дитина", "дитячий", "малюк", "школяр", "підліток", "дитині",
"дитиною",
    "маленький", "діти", "діткам", "для дітей", "юний", "підлітковий",
"дошкільник"
  ]),
  "tags": ["казки", "пригоди", "освітні", "розвиваючі", "фантастика
для дітей",
    "фентезі для підлітків", "дитяча проза", "юнацька література"],
  "description": "для дітей та підлітків"
},

```

```

"elderly_mood": {
    "keywords": self._get_normalized_forms([
        "пенсіонер", "літній", "похилий", "дорослий", "старший",
"зрілий",
        "дорослим", "дідусь", "бабуся", "старшого віку", "дід"
    ]),
    "tags": ["мемуари", "історія", "класика", "філософія",
"документалістика",
        "сімейні саги", "публіцистика", "біографії"],
    "description": "для старшого покоління"
},
"energetic_mood": {
    "keywords": self._get_normalized_forms([
        "енергійний", "активний", "бадьорий", "мотивований", "хочу
діяти", "натхнення",
        "продуктивний", "мотивація", "діяти", "розвиватися"
    ]),
    "tags": ["бізнес", "мотивація", "саморозвиток", "лідерство",
"продуктивність",
        "фінанси", "кар'єра", "управління"],
    "description": "для натхнення та нових звершень"
}
}

```

```

def _normalize_word(self, word):
    word = word.lower().strip()
    for ending in sorted(self.ukr_endings, key=len, reverse=True): # Довші
закінчення першими
        if word.endswith(ending) and len(word) > len(ending):
            return word[:-len(ending)]

```

```

    return word
def _get_normalized_forms(self, words):
    """Застосовує нормалізацію до всіх слів у списку"""
    return [self._normalize_word(word) for word in words]
def preprocess_text(self, text):
    """Попередня обробка тексту та нормалізація слів"""
    text = text.lower().strip()
    text = re.sub(r'^\w\s', ' ', text)
    text = re.sub(r'\s+', ' ', text)
    words = text.split()
    normalized_words = [self._normalize_word(word) for word in words]
    return ' '.join(normalized_words)
def extract_age(self, message):
    processed_message = self.preprocess_text(message)
    age_patterns = re.findall(r'\b(\d{1,2})\s*(?:років?|річ|рок|рочки|р)\b',
message)
    if age_patterns:
        return int(age_patterns[0])
        if any(w in processed_message for w in
self._get_normalized_forms(['дитина', 'малюк', 'дошкільник'])):
            return "children"
        if any(w in processed_message for w in
self._get_normalized_forms(['школяр', 'підліток', 'юнак', 'юначка'])):
            return "teenager"
        if any(w in processed_message for w in
self._get_normalized_forms(['дідусь', 'бабуся', 'літній', 'старший', 'пенсіонер'])):
            return "elderly"
        if any(w in processed_message for w in
self._get_normalized_forms(['дорослий', 'для дорослих'])):
            return "adult"

```

```

return None

def detect_mood(self, message):
    processed_message = self.preprocess_text(message)
    detected_moods = []
    for mood, data in self.mood_categories.items():
        if any(keyword in processed_message for keyword in data['keywords']):
            detected_moods.append(mood)
    return detected_moods

def extract_genre_keywords(self, message):
    processed_message = self.preprocess_text(message)
    genre_mapping = {
        'пригод': 1, 'пригодницьк': 1, 'мандр': 1,
        'детектив': 2, 'розслідуван': 2, 'кримінал': 2, 'трилер': 2,
        'фентез': 3, 'маг': 3, 'чарівник': 3, 'фантастик': 3,
        'дитяч': 4, 'малюк': 4, 'казк': 4, 'розвиваюч': 4,
        'навчальн': 5, 'навчан': 5, 'посібник': 5, 'освітн': 5, 'саморозвит': 5,
        'довідник': 6, 'енциклопеді': 6, 'інформаційн': 6,
        'комеді': 7, 'смішн': 7, 'весел': 7, 'кумедн': 7, 'гумор': 7,
        'біографі': 8, 'житт': 8, 'мемуар': 8, 'автобіографі': 8,
        'класик': 9, 'філософі': 9, 'літератур': 9,
        'новел': 10, 'коротк': 10,
        'історі': 11, 'історичн': 11,
        'науков': 12, 'наук': 12, 'досліджен': 12,
        'поезі': 13, 'вірш': 13, 'лірик': 13,
        'жахи': 14, 'жах': 14, 'містик': 15, 'містичний': 15,
        'любовн': 16, 'романтичн': 16,
        'психологі': 17, 'психологічний': 17
    }

    found_genres = []

```

```

for keyword_stem, category_id in genre_mapping.items():
    if keyword_stem in processed_message:
        found_genres.append(category_id)

return list(set(found_genres))

def find_intent(self, message):
    processed_message = self.preprocess_text(message)
    best_match = None
    best_score = 0
    if len(processed_message.split()) <= 2 and any(p in processed_message for
p in self._get_normalized_forms(["привіт", "добрий"])):
        return next((i for i in self.enhanced_intents if i['intent'] == 'greeting'),
None)

    for intent_data in self.enhanced_intents:
        intent = intent_data['intent']
        patterns = intent_data['patterns']
        score = 0
        for pattern_word in patterns:
            if pattern_word in processed_message.split():
                score += 1 # Просте співпадіння слова
            if pattern_word == processed_message:
                score += 5 # Точне співпадіння
            elif len(pattern_word) > 3 and pattern_word in processed_message:
                score += 2 # Співпадіння кореня слова
        if intent == "age_based_recommendations" and self.extract_age(message):
            score += 10
            if intent == "mood_based_recommendations" and
self.detect_mood(message):
                score += 10

```

```

        if intent == "genre_based_recommendations" and
self.extract_genre_keywords(message):
            score += 10
            if score > best_score:
                best_score = score
                best_match = intent_data
            elif score == best_score and best_match:
                specific_intents = ["age_based_recommendations",
"mood_based_recommendations", "genre_based_recommendations", "book_search",
"order_status"]
                if intent in specific_intents and best_match['intent'] not in
specific_intents:
                    best_match = intent_data
            return best_match if best_score > 0 else None

def handle_book_search(self, message):
    processed_message = self.preprocess_text(message)
    words = processed_message.split()
    matching_books = []
    for book in data['books']:
        book_text = f"{book['title']} {book['author']} {book.get('description', '')}
{book.get('tags', '')}".lower()
        book_text_normalized = self.preprocess_text(book_text)
        matches = sum(1 for word in words if len(word) > 2 and word in
book_text_normalized)
        if matches > 0:
            matching_books.append((book, matches))
    matching_books.sort(key=lambda x: x[1], reverse=True)
    if matching_books:
        top_books = matching_books[:3]

```

```

response = "Ось що я знайшов:\n\n"
for book, _ in top_books:
    response += f"📖 {book['title']} - {book['author']}\n"
    response += f"Ціна: {book['price']} грн\n"
response += "Бажаєте дізнатися більше про якусь книгу?"
return response
else:
    return "На жаль, не знайшов книг за вашим запитом. Спробуйте інший пошук або зверніться до консультанта. Можливо, вкажіть повну назву або автора?"

```

```

def handle_age_based_recommendations(self, message):
    age_info = self.extract_age(message)
    if age_info is None:
        return "Скажіть, будь ласка, для якого віку або вікової групи (наприклад, 'для підлітка', 'для дідуся') ви шукаєте книгу, щоб я міг підібрати найкращі варіанти!"
    target_categories = []
    age_group_desc = ""
    if isinstance(age_info, int):
        if age_info <= 6:
            target_categories = [4] # Дитячі книги
            age_group_desc = f"малят ({age_info} років)"
        elif age_info <= 12:
            target_categories = [4, 7, 1] # Дитячі, легкі пригоди, комедія
            age_group_desc = f"молодших школярів ({age_info} років)"
        elif age_info <= 16:
            target_categories = [1, 2, 3, 16] # Пригоди, фентезі, детективи, романтичні
            age_group_desc = f"підлітків ({age_info} років)"

```

```

else:
    target_categories = list(range(1, 18)) # Всі категорії
    age_group_desc = f"дорослих ({age_info} років)"
else:
    if age_info == "children":
        target_categories = [4]
        age_group_desc = "дітей"
    elif age_info == "teenager":
        target_categories = [1, 2, 3, 16]
        age_group_desc = "підлітків"
    elif age_info == "elderly":
        target_categories = [8, 9, 11, 13, 5] # Біографії, класика, історія,
        поезія, навчальна
        age_group_desc = "старшого покоління"
    elif age_info == "adult":
        target_categories = list(range(1, 18)) # Всі категорії для дорослих
        age_group_desc = "дорослих"
suitable_books = [book for book in data['books']
                   if book['category_id'] in target_categories]
suitable_books.sort(key=lambda x: (x.get('featured', False), x['rating']),
reverse=True)
top_books = suitable_books[:3]
if top_books:
    response = f" 🐱 Чудові книги для {age_group_desc}:\n\n"
    for book in top_books:
        category_name = next((cat['name'] for cat in data['categories']
                              if cat['id'] == book['category_id']), 'Невідома')
        response += f" 📖 {book['title']} - {book['author']}\n"
    response += f"Категорія: {category_name}\n"

```

```

response += f"Рейтинг: {'★' * int(book['rating'])} ({book['rating']})\n"
response += f"Ціна: {book['price']} грн\n\n"
response += "Сподіваюся, ці книги сподобаються!"
return response
else:
    return f"Вибачте, зараз немає книг для {age_group_desc}. Спробуйте розширити критерії пошуку."

```

```

def handle_mood_based_recommendations(self, message):
    moods = self.detect_mood(message)
    if not moods:
        return "Щоб підібрати книгу за настроєм, розкажіть, будь ласка, трохи більше про те, що ви відчуваєте. Наприклад: 'Хочу щось веселе' або 'Потрібно розслабитися'."
    suitable_books = []
    for mood in moods:
        mood_tags = self.mood_categories[mood]['tags']
        for book in data['books']:
            book_tags_normalized = self.preprocess_text(book.get('tags', "")).split()
            if any(t_norm in book_tags_normalized for t_norm in self._get_normalized_forms(mood_tags)):
                suitable_books.append(book)

    # Якщо не знайшли за тегами, шукаємо за категоріями, пов'язаними з настроєм
    if not suitable_books:
        category_ids = []
        for mood in moods:
            if mood == "good_mood":
                category_ids.extend([7, 1]) # Комедія, пригоди

```

```

elif mood == "bad_mood":
    category_ids.extend([8, 5, 9, 17]) # Біографія, навчальна, класика,
психологія
elif mood == "relax_mood":
    category_ids.extend([10, 9, 3, 13]) # Новели, класика, фентезі,
поезія
elif mood == "children_mood":
    category_ids.extend([4, 1]) # Дитячі, пригоди
elif mood == "elderly_mood":
    category_ids.extend([9, 8, 11]) # Класика, біографія, історія
elif mood == "energetic_mood":
    category_ids.extend([5, 8, 12]) # Навчальна, біографія, наукова

suitable_books.extend([book for book in data['books']
                        if book['category_id'] in category_ids])
unique_suitable_books = []
seen_book_ids = set()
for book in suitable_books:
    if book['id'] not in seen_book_ids:
        unique_suitable_books.append(book)
        seen_book_ids.add(book['id'])
unique_suitable_books.sort(key=lambda x: x['rating'], reverse=True)
top_books = unique_suitable_books[:3]
if top_books:
    mood_descriptions = [self.mood_categories[mood]['description'] for mood
in moods]
    response = f" 🐱 Ось книги {', '.join(mood_descriptions)}:\n\n"
    for book in top_books:
        response += f" 📖 {book['title']} - {book['author']}\n"

```

```

response += f"Рейтинг: {'★' * int(book['rating'])} ({book['rating']})\n"
response += f"Ціна: {book['price']} грн\n\n"
response += "Сподіваюся, ці рекомендації покращать ваш день! 😊"
return response

```

else:

```

return "Вибачте, зараз немає підходящих книг для вашого настрою.

```

Спробуйте інші критерії або попросіть загальні рекомендації."

```

def handle_genre_based_recommendations(self, message):

```

```

    genre_categories = self.extract_genre_keywords(message)

```

```

    if not genre_categories:

```

```

        return "Не зрозумів, який саме жанр книг вас цікавить. Можете

```

уточнити: пригоди, детективи, фентезі, комедії, історія?"

```

        suitable_books = [book for book in data['books']

```

```

            if book['category_id'] in genre_categories]

```

```

        suitable_books.sort(key=lambda x: x['rating'], reverse=True)

```

```

        top_books = suitable_books[:3]

```

```

        if top_books:

```

```

            genre_names = []

```

```

            for cat_id in genre_categories:

```

```

                cat_name = next((cat['name'] for cat in data['categories']

```

```

                    if cat['id'] == cat_id), 'Невідомий')

```

```

                genre_names.append(cat_name)

```

```

            response = f"🐱 Найкращі книги в жанрі {', '.join(genre_names)}:\n\n"

```

```

            for book in top_books:

```

```

                response += f"📖 {book['title']} - {book['author']}\n"

```

```

                response += f"Рейтинг: {'★' * int(book['rating'])} ({book['rating']})\n"

```

```

                response += f"Ціна: {book['price']} грн\n\n"

```

```

            response += "Чи можу я ще чимось допомогти з вибором?"

```

```

            return response

```

```

else:
    return "Вибачте, зараз немає книг у цьому жанрі. Спробуйте інший
жанр або загальні рекомендації."
def handle_popular_books(self, message):
    bestsellers = [book for book in data['books'] if book.get('bestseller', False)]
    if not bestsellers:
        bestsellers = sorted(data['books'], key=lambda x: x['rating'], reverse=True)
    top_books = bestsellers[:5]
    response = " 🐱 Найпопулярніші книги зараз:\n\n"
    if top_books:
        for i, book in enumerate(top_books, 1):
            response += f"{i}. {book['title']} - {book['author']}\n"
            response += f"    Рейтинг: {'★' * int(book['rating'])}
({book['rating'])}\n"
            response += f"    Ціна: {book['price']} грн\n\n"
            response += "Це справді найкращі книги, які обирають наші
клієнти! 🏆 "
    else:
        response = "На жаль, зараз не можу знайти популярні книги.
Спробуйте пізніше."
    return response
def handle_order_status(self, message):
    numbers = re.findall(r'\d+', message)
    if numbers:
        order_id = int(numbers[0])
        order = next((o for o in data['orders'] if o['id'] == order_id), None)
        if order:
            status_translations = {
                'pending': 'очікує обробки',

```

```

    'confirmed': 'підтверджено',
    'shipped': 'відправлено',
    'delivered': 'доставлено',
    'cancelled': 'скасовано'
}
status = status_translations.get(order['status'], order['status'])
response = f"📦 Заовлення #{order_id}:\n"
response += f"Статус: {status}\n"
response += f"Дата: {order['order_date'][:10]}\n"
response += f"Сума: {order['total_amount']} грн\n"
if order['status'] == 'shipped':
    response += "\n🚚 Очікуваний термін доставки: 1-2 дні"
elif order['status'] == 'delivered':
    response += "\n✅ Ваше замовлення вже доставлено.

```

Сподіваємося, вам сподобається!"

```

    return response

```

```

else:

```

```

    return f"Замовлення #{order_id} не знайдено. Перевірте номер."

```

```

else:

```

```

    return "Будь ласка, вкажіть номер замовлення для перевірки статусу.

```

Наприклад: 'Статус замовлення 123'."

```

def handle_recommendations(self, message):

```

```

    processed_message = self.preprocess_text(message)

```

```

    age_info_detected = self.extract_age(message)

```

```

    if age_info_detected:

```

```

        return self.handle_age_based_recommendations(message)

```

```

    moods_detected = self.detect_mood(message)

```

```

    if moods_detected:

```

```

        return self.handle_mood_based_recommendations(message)

```

```

genres_detected = self.extract_genre_keywords(message)
if genres_detected:
    return self.handle_genre_based_recommendations(message)
        if any(word in processed_message for word in
self._get_normalized_forms(['популярні', 'бестселери', 'топ', 'хіти')):
            return self.handle_popular_books(message)
            featured_books = [book for book in data['books'] if book.get('featured',
False)]
            if not featured_books:
                featured_books = sorted(data['books'], key=lambda x: x['rating'],
reverse=True)
            top_books = featured_books[:3]
            if top_books:
                response = " 🐱 Рекомендую ці книги, можливо, вони вас
зацікавлять:\n\n"
                for book in top_books:
                    response += f" 📖 {book['title']} - {book['author']}\n"
                    response += f"Рейтинг: {'★' * int(book['rating'])} ({book['rating']})\n"
                    response += f"Ціна: {book['price']} грн\n\n"
                response += "Чи хотіли б ви уточнити: за жанром, настроєм, чи для
певного віку?"
            return response
        else:
            return "Вибачте, зараз не можу запропонувати конкретних
рекомендацій. Можливо, ви бажаєте знайти книгу за назвою або автором?"

def get_response(self, message, user_session):
    try:
        intent_data = self.find_intent(message)

```

```

if not intent_data:
    return "Вибачте, не зрозумів ваш запит. Спробуйте перефразувати
    або оберіть один з варіантів:\n- Пошук книг\n- Статус замовлення\n-
    Рекомендації\n- Інформація про доставку"
    intent = intent_data['intent']
    if intent == 'greeting':
        return "Привіт! 😊 Вітаю в БукЛайф ❤️\nЯ ваш асистент і можу
        допомогти з:\n\n📖 Пошуком книг за назвою, автором або тематикою\n📦
        Статусом замовлення\n💡 Рекомендаціями за віком, жанром та настроєм\n🏆
        Популярними книгами\n🚚 Інформацією про доставку та оплату\n\nЩо вас
        цікавить?"
    elif intent == 'book_search':
        return self.handle_book_search(message)
    elif intent == 'order_status':
        return self.handle_order_status(message)
    elif intent == 'age_based_recommendations':
        return self.handle_age_based_recommendations(message)
    elif intent == 'mood_based_recommendations':
        return self.handle_mood_based_recommendations(message)
    elif intent == 'genre_based_recommendations':
        return self.handle_genre_based_recommendations(message)
    elif intent == 'popular_books':
        return self.handle_popular_books(message)
    elif intent == 'recommendations':
        return self.handle_recommendations(message)
    elif intent == 'delivery_info':
        import random
        return random.choice(intent_data['responses'])
    elif intent == 'payment_info':

```

```

        import random
        return random.choice(intent_data['responses'])
    else:
        import random
        return random.choice(intent_data['responses'])
except Exception as e:
    logger.error(f"Помилка обробки запиту: {str(e)}")
    return "Вибачте, сталася помилка. Спробуйте, будь ласка, пізніше."

load_database()
chatbot = WebsiteChatbot(data.get('chatbot_training_data', []))

@app.route('/')
def index():
    featured_books = [book for book in data['books'] if book.get('featured',
False)][:6]
    bestsellers = [book for book in data['books'] if book.get('bestseller', False)][:6]
    if not featured_books:
        featured_books = sorted(data['books'], key=lambda x: x['rating'],
reverse=True)[:6]
    if not bestsellers:
        bestsellers = sorted(data['books'], key=lambda x: x['rating'],
reverse=True)[6:12]
    return render_template('index.html',
        featured_books=featured_books,
        bestsellers=bestsellers,
        categories=data['categories'])

@app.route('/catalog')
def catalog():
    category_id = request.args.get('category', type=int)
    search_query = request.args.get('search', "").strip()
    sort_by = request.args.get('sort', 'title')

```

```

books = data['books'].copy()
if category_id:
    books = [book for book in books if book['category_id'] == category_id]
if search_query:
    search_lower = search_query.lower()
    search_normalized_words = chatbot.preprocess_text(search_query).split()
    filtered_books = []
    for book in books:
        book_title_normalized = chatbot.preprocess_text(book['title'])
        book_author_normalized = chatbot.preprocess_text(book['author'])
        book_description_normalized =
chatbot.preprocess_text(book.get('description', ""))

        if any(word in book_title_normalized.split() for word in
search_normalized_words) or \
            any(word in book_author_normalized.split() for word in
search_normalized_words) or \
            any(word in book_description_normalized.split() for word in
search_normalized_words):
            filtered_books.append(book)
    books = filtered_books
if sort_by == 'price_asc':
    books.sort(key=lambda x: x['price'])
elif sort_by == 'price_desc':
    books.sort(key=lambda x: x['price'], reverse=True)
elif sort_by == 'rating':
    books.sort(key=lambda x: x['rating'], reverse=True)
else:
    books.sort(key=lambda x: x['title'])

```

```

return render_template('catalog.html',
                       books=books,
                       categories=data['categories'],
                       current_category=category_id,
                       search_query=search_query,
                       sort_by=sort_by)

@app.route('/book/<int:book_id>')
def book_detail(book_id):
    book = next((book for book in data['books'] if book['id'] == book_id), None)
    if not book:
        flash('Книгу не знайдено', 'error')
        return redirect(url_for('catalog'))
    book_reviews = [review for review in data['reviews'] if review['book_id'] ==
book_id]
    for review in book_reviews:
        client = next((c for c in data['clients'] if c['id'] == review['client_id']), None)
        review['client_name'] = client['name'] if client else 'АНОНІМ'
    similar_books = [b for b in data['books']
                     if b['category_id'] == book['category_id'] and b['id'] != book_id][:4]
    return render_template('book_detail.html',
                           book=book,
                           reviews=book_reviews,
                           similar_books=similar_books)

@app.route('/add_to_cart', methods=['POST'])
def add_to_cart():
    book_id = request.form.get('book_id', type=int)
    quantity = request.form.get('quantity', 1, type=int)
    if 'user_session' not in session:
        session['user_session'] = str(uuid.uuid4())
    user_session = session['user_session']

```

```

book = next((book for book in data['books'] if book['id'] == book_id), None)
if not book:
    return jsonify({'success': False, 'message': 'Книгу не знайдено'})
if quantity > book['stock']:
    return jsonify({'success': False, 'message': f'На складі лише {book["stock"]}
шт.'})
cart_item = {
    'book_id': book_id,
    'title': book['title'],
    'author': book['author'],
    'price': book['price'],
    'quantity': quantity,
    'total': book['price'] * quantity
}
existing_item = next((item for item in shopping_cart[user_session]
                     if item['book_id'] == book_id), None)
if existing_item:
    existing_item['quantity'] += quantity
    existing_item['total'] = existing_item['price'] * existing_item['quantity']
else:
    shopping_cart[user_session].append(cart_item)
cart_count = sum(item['quantity'] for item in shopping_cart[user_session])
return jsonify({'success': True, 'message': 'Додано до кошика!', 'cart_count':
cart_count})

@app.route('/cart')
def cart():
    if 'user_session' not in session:
        session['user_session'] = str(uuid.uuid4())
    user_session = session['user_session']
    cart_items = shopping_cart.get(user_session, [])

```

```

        total_amount = sum(item['total'] for item in cart_items)
        return render_template('cart.html', cart_items=cart_items,
total_amount=total_amount)
@app.route('/checkout', methods=['GET', 'POST'])
def checkout():
    if 'user_session' not in session:
        return redirect(url_for('cart'))
    user_session = session['user_session']
    cart_items = shopping_cart.get(user_session, [])
    if not cart_items:
        flash('Кошик порожній', 'error')
        return redirect(url_for('catalog'))
    if request.method == 'POST':
        client_data = {
            'name': request.form.get('name'),
            'email': request.form.get('email'),
            'phone': request.form.get('phone'),
            'address': request.form.get('address'),
            'payment_method': request.form.get('payment_method'),
            'shipping_method': request.form.get('shipping_method'),
            'notes': request.form.get('notes', "")
        }
        new_order_id = max(order['id'] for order in data['orders']) + 1 if data['orders']
    else 1
        total_amount = sum(item['total'] for item in cart_items)
        new_order = {
            'id': new_order_id,
            'client_data': client_data,
            'order_date': datetime.now().isoformat(),
            'status': 'pending',

```

```

        'items': cart_items,
        'total_amount': total_amount,
        'shipping_address': client_data['address'],
        'payment_method': client_data['payment_method'],
        'shipping_method': client_data['shipping_method'],
        'notes': client_data['notes']
    }
    data['orders'].append(new_order)
    for item in cart_items:
        book = next((book for book in data['books'] if book['id'] ==
item['book_id']), None)
        if book:
            book['stock'] -= item['quantity']
    shopping_cart[user_session] = []
    save_database()
    flash(f'Замовлення #{new_order_id} успішно оформлено!', 'success')
    return redirect(url_for('order_success', order_id=new_order_id))
total_amount = sum(item['total'] for item in cart_items)
    return render_template('checkout.html', cart_items=cart_items,
total_amount=total_amount)

@app.route('/order_success/<int:order_id>')
def order_success(order_id):
    order = next((order for order in data['orders'] if order['id'] == order_id), None)
    if not order:
        flash('Замовлення не знайдено', 'error')
        return redirect(url_for('index'))
    return render_template('order_success.html', order=order)

@app.route('/chat', methods=['POST'])
def chat_endpoint():
    data_request = request.get_json()

```

```

message = data_request.get('message', '')
if 'user_session' not in session:
    session['user_session'] = str(uuid.uuid4())
user_session = session['user_session']
if not message.strip():
    return jsonify({'reply': 'Будь ласка, введіть ваше питання.'})
try:
    response = chatbot.get_response(message, user_session)
    return jsonify({'reply': response})
except Exception as e:
    app.logger.error(f"Помилка чат-бота: {str(e)}")
    return jsonify({'reply': 'Вибачте, сталася помилка. Спробуйте пізніше або
зверніться до служби підтримки.'})
@app.route('/api/cart_count')
def cart_count():
    if 'user_session' not in session:
        return jsonify({'count': 0})
    user_session = session['user_session']
    count = sum(item['quantity'] for item in shopping_cart.get(user_session, []))
    return jsonify({'count': count})
if __name__ == '__main__':
    if not os.path.exists('database.json'):
        print("УВАГА: Файл database.json не знайдено! Створіть базу даних
перед запуском.")
    app.run(debug=True, host='0.0.0.0', port=5000)

```

Повний код файлу base.html (включає відображення чатбота користувачам)

```
<!DOCTYPE html>
<html lang="uk">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>{% block title %}БукЛайф - Онлайн книжковий магазин{%
      endblock %}</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
      rel="stylesheet">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
      awesome/6.4.0/css/all.min.css">
    <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@300;400;500;600;700&
      display=swap" rel="stylesheet">
    <style>
      :root {
        --primary-color: #2563eb;
        --secondary-color: #1e40af;
        --accent-color: #f59e0b;
        --text-color: #1f2937;
        --bg-color: #f8f9fa;
        --border-color: #e5e7eb;
      }
      * {
        margin: 0;
        padding: 0;
```

```
    box-sizing: border-box;
    }
    body {
font-family: 'Inter', sans-serif;
    line-height: 1.6;
    color: var(--text-color);
background-color: var(--bg-color);
    }
    .navbar {
background: white;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
padding: 1rem 0;
    }
    .navbar-brand {
font-weight: 700;
font-size: 1.5rem;
color: var(--primary-color) !important;
    }
    .btn-primary {
background-color: var(--primary-color);
border-color: var(--primary-color);
border-radius: 8px;
padding: 0.5rem 1.5rem;
font-weight: 500;
transition: all 0.3s;
    }
    .btn-primary:hover {
background-color: var(--secondary-color);
border-color: var(--secondary-color);
transform: translateY(-2px);
```

```
    }  
    .card {  
        border: none;  
        border-radius: 12px;  
        box-shadow: 0 4px 20px rgba(0,0,0,0.08);  
        transition: all 0.3s;  
        overflow: hidden;  
    }  
    .card:hover {  
        transform: translateY(-5px);  
        box-shadow: 0 8px 30px rgba(0,0,0,0.12);  
    }  
    .book-card {  
        height: 100%;  
    }  
    .book-image {  
        height: 250px;  
        background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);  
        display: flex;  
        align-items: center;  
        justify-content: center;  
        color: white;  
        font-size: 3rem;  
        position: relative;  
    }  
    .book-title {  
        font-weight: 600;  
        color: var(--text-color);  
        margin-bottom: 0.5rem;  
        font-size: 1.1rem;
```

```
    }  
    .book-author {  
      color: #6b7280;  
      font-size: 0.9rem;  
      margin-bottom: 0.5rem;  
    }  
    .book-price {  
      font-weight: 700;  
      color: var(--accent-color);  
      font-size: 1.2rem;  
    }  
    .rating {  
      color: #fbbf24;  
      font-size: 0.9rem;  
      margin-bottom: 0.5rem;  
    }  
    .footer {  
background: var(--text-color);  
      color: white;  
      padding: 3rem 0 1rem;  
      margin-top: 4rem;  
    }  
    .chat-widget {  
      position: fixed;  
      bottom: 20px;  
      right: 20px;  
      z-index: 1000;  
    }  
    .chat-toggle {  
      width: 60px;
```

```
        height: 60px;
        border-radius: 50%;
        background: var(--primary-color);
        color: white;
        border: none;
        font-size: 1.5rem;
        box-shadow: 0 4px 20px rgba(37, 99, 235, 0.3);
        transition: all 0.3s;
    }
    .chat-toggle:hover {
        background: var(--secondary-color);
        transform: scale(1.1);
    }
    .chat-box {
        position: absolute;
        bottom: 80px;
        right: 0;
        width: 400px;
        height: 500px;
        background: white;
        border-radius: 15px;
        box-shadow: 0 10px 40px rgba(0,0,0,0.2);
        display: none;
        flex-direction: column;
    }
    .chat-header {
        background: var(--primary-color);
        color: white;
        padding: 1rem;
        border-radius: 15px 15px 0 0;
```

```
font-weight: 600;
    }
.chat-messages {
    flex: 1;
    padding: 1rem;
    overflow-y: auto;
    display: flex;
flex-direction: column;
    gap: 10px;
    }
    .message {
    max-width: 80%;
    padding: 8px 12px;
    border-radius: 12px;
    font-size: 0.9rem;
    line-height: 1.4;
    }
    .message.user {
background: var(--primary-color);
    color: white;
    align-self: flex-end;
    }
    .message.bot {
background: #f3f4f6;
    color: var(--text-color);
    align-self: flex-start;
    }
    .chat-input {
padding: 1rem;
border-top: 1px solid var(--border-color);
```

```
border-radius: 0 0 15px 15px;
    }
    .alert {
border-radius: 8px;
margin-bottom: 1rem;
    }
    .cart-badge {
position: absolute;
top: -5px;
right: -5px;
background: var(--accent-color);
color: white;
border-radius: 50%;
width: 20px;
height: 20px;
font-size: 0.75rem;
display: flex;
align-items: center;
justify-content: center;
    }
@media (max-width: 768px) {
    .chat-box {
width: 300px;
height: 400px;
    }
    .book-image {
height: 200px;
font-size: 2rem;
    }
}
```

```

</style>

{% block extra_css %} {% endblock %}
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light">
  <div class="container">
    <a class="navbar-brand" href="{{ url_for('index') }}">
      <i class="fas fa-cat me-2 "></i>БукЛайф
    </a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
      data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav me-auto">
        <li class="nav-item">
          <a class="nav-link" href="{{ url_for('index') }}">Головна</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="{{ url_for('catalog') }}">Каталог</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button"
            data-bs-toggle="dropdown">
            Категорії
          </a>
          <ul class="dropdown-menu">
            {% for category in categories %}
            {% if not category.parent_id %}

```

```

<li><a class="dropdown-item" href="{{ url_for('catalog',
category=category.id) }}">{{ category.name }}</a></li>
    {% endif %}
{% endfor %}
</ul>
</li>
</ul>

<div class="d-flex align-items-center gap-3">
<form class="d-flex" method="GET" action="{{ url_for('catalog')
    }}">
    <input class="form-control me-2" type="search" name="search"
placeholder="Пошук книг..." style="min-width: 200px;">
    <button class="btn btn-outline-primary" type="submit">
        <i class="fas fa-search"></i>
    </button>
</form>

<a href="{{ url_for('cart') }}" class="btn btn-outline-primary
    position-relative">
    <i class="fas fa-shopping-cart"></i>
    <span class="cart-badge" id="cart-count">0</span>
</a>
</div>
</div>
</div>
</nav>

{% with messages = get_flashed_messages(with_categories=true) %}
    {% if messages %}
        <div class="container mt-3">
            {% for category, message in messages %}

```

```

<div class="alert alert-{{ 'danger' if category == 'error' else 'success'
}} alert-dismissible fade show">
    {{ message }}
<button type="button" class="btn-close" data-bs-
dismiss="alert"></button>
</div>
{% endfor %}
</div>
{% endif %}
{% endwith %}
<main>
{% block content %} {% endblock %}
</main>
<footer class="footer">
<div class="container">
<div class="row">
<div class="col-md-4">
<h5>БукЛайф</h5>
<p>Ваш найкращий онлайн книжковий магазин з величезним
вибором книг.</p>
</div>
<div class="col-md-4">
<h6>Корисні посилання</h6>
<ul class="list-unstyled">
<li><a href="{{ url_for('catalog') }}" class="text-
light">Каталог</a></li>
<li><a href="#" class="text-light">Про нас</a></li>
<li><a href="#" class="text-light">Доставка та оплата</a></li>
<li><a href="#" class="text-light">Контакти</a></li>
</ul>

```

```

        </div>
        <div class="col-md-4">
            <h6>Контакти</h6>
            <p><i class="fas fa-phone me-2"></i> +380 44 123 4567</p>
            <p><i class="fas fa-envelope me-2"></i> info@buklife.ua</p>
            <p><i class="fas fa-map-marker-alt me-2"></i> Kyiv</p>
        </div>
    </div>
    <hr class="my-4">
    <div class="text-center">
        <p>&copy; 2025 БукЛайф</p>
    </div>
</div>
</div>
</footer>
<div class="chat-widget">
    <div class="chat-box" id="chatBox">
        <div class="chat-header">
            <div class="d-flex justify-content-between align-items-center">
                <span><i class="fas fa-robot me-2"></i> AI Помічник</span>
                <button type="button" class="btn-close btn-close-white"
                    id="closeChatBtn"></button>
            </div>
        </div>
        <div class="chat-messages" id="chatMessages">
            <div class="message bot">
                Привіт! 😊 Я ваш помічник БукЛайф 📖 Буду радий допомогти
                Вам з пошуком книг, статусом замовлення або рекомендаціями. Що вас
                цікавить?
            </div>
        </div>
    </div>
</div>

```

```

        <div class="chat-input">
            <div class="input-group">
                <input type="text" class="form-control" id="chatInput"
placeholder="Введіть ваше питання...">
                <button class="btn btn-primary" id="sendChatBtn">
                    <i class="fas fa-paper-plane"></i>
                </button>
            </div>
        </div>
    </div>
    <div class="chat-toggles">
        <button class="chat-toggle" id="chatToggle">
            <i class="fas fa-comments"></i>
        </button>
    </div>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></
script>
    <script>
        class ChatBot {
            constructor() {
                this.chatBox = document.getElementById('chatBox');
                this.chatToggle = document.getElementById('chatToggle');
                this.closeChatBtn = document.getElementById('closeChatBtn');
                this.chatInput = document.getElementById('chatInput');
                this.sendChatBtn = document.getElementById('sendChatBtn');
                this.chatMessages = document.getElementById('chatMessages');

                this.init();
            }
            init() {
                this.chatToggle.addEventListener('click', () => this.toggleChat());
            }
        }
    </script>

```

```

this.closeChatBtn.addEventListener('click', () => this.closeChat());
this.sendChatBtn.addEventListener('click', () => this.sendMessage());
this.chatInput.addEventListener('keypress', (e) => {
  if (e.key === 'Enter') this.sendMessage();
});
}
toggleChat() {
this.chatBox.style.display = this.chatBox.style.display === 'flex' ?
  'none' : 'flex';
if (this.chatBox.style.display === 'flex') {
  this.chatInput.focus();
}
}
closeChat() {
  this.chatBox.style.display = 'none';
}
addMessage(message, isUser = false) {
  const messageDiv = document.createElement('div');
  messageDiv.className = `message ${isUser ? 'user' : 'bot'}`;
  messageDiv.innerHTML = message.replace(/\n/g, '<br>');
  this.chatMessages.appendChild(messageDiv);
  this.chatMessages.scrollTop = this.chatMessages.scrollHeight;
}
async sendMessage() {
  const message = this.chatInput.value.trim();
  if (!message) return;
  this.addMessage(message, true);
  this.chatInput.value = "";
  this.addMessage('<i class="fas fa-spinner fa-spin"></i> Думаю...');
  try {

```

```

const response = await fetch('/chat', {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json'
    },
    body: JSON.stringify({ message: message })
});

const data = await response.json();
this.chatMessages.removeChild(this.chatMessages.lastChild);
this.addMessage(data.reply);
    } catch (error) {
this.chatMessages.removeChild(this.chatMessages.lastChild);
this.addMessage('Вибачте, сталася помилка. Спробуйте
    пізніше.');
```

```

    }
    }
    }

class ShoppingCart {
    constructor() {
        this.updateCartCount();
        this.init();
    }
    init() {
        document.addEventListener('click', (e) => {
            if (e.target.classList.contains('add-to-cart-btn')) {
                e.preventDefault();
                this.addToCart(e.target);
            }
        });
    }
}

```

```

        async addToCart(button) {
            const form = button.closest('form');
            const formData = new FormData(form);
                try {
const response = await fetch('/add_to_cart', {
                    method: 'POST',
                    body: formData
                });
            const data = await response.json();
                if (data.success) {
                    this.showSuccess(data.message);
                    this.updateCartCount();
                } else {
                    this.showError(data.message);
                }
            } catch (error) {
this.showError('Помилка додавання до кошика');
            }
        }
        async updateCartCount() {
                try {
const response = await fetch('/api/cart_count');
            const data = await response.json();
document.getElementById('cart-count').textContent = data.count;
                } catch (error) {
console.error('Помилка оновлення кошика:', error);
            }
        }
        showSuccess(message) {
            this.showToast(message, 'success');
        }
    }
}

```

```

    }
    showError(message) {
    this.showToast(message, 'error');
    }
    showToast(message, type) {
    const toast = document.createElement('div');
    toast.className = `alert alert-${type === 'error' ? 'danger' : 'success'}
    position-fixed`;
    toast.style.cssText = 'top: 20px; right: 20px; z-index: 9999; min-width:
    250px;';
    toast.innerHTML = `
    ${message}
    <button type="button" class="btn-close"
    onclick="this.parentElement.remove()"></button>
    document.body.appendChild(toast);
    setTimeout(() => {
    if (toast.parentElement) {
    toast.remove();
    }
    }, 3000);
    }
    }
    document.addEventListener('DOMContentLoaded', function() {
    new ChatBot();
    new ShoppingCart();
    });
</script>
{% block extra_js %} {% endblock %}
</body>
</html>

```

## Код для відображення головної сторінки сайту (index.html)

```

{% extends "base.html" %}
{% block content %}
<!-- Hero Section -->
<section class="bg-primary text-white py-5">
  <div class="container">
    <div class="row align-items-center">
      <div class="col-lg-6">
        <h1 class="display-4 fw-bold mb-4">Ласкаво просимо до
БукЛайф!</h1>
        <p class="lead mb-4">Найкращий книжковий онлайн-магазин з
величезним вибором книг, швидкою доставкою та помічником для персональних
рекомендацій.</p>
        <div class="d-flex gap-3">
          <a href="{{ url_for('catalog') }}" class="btn btn-light btn-lg">
            <i class="fas fa-book me-2"></i>Переглянути каталог
          </a>
          <button class="btn btn-outline-light btn-lg"
onclick="document.getElementById('chatToggle').click()">
            <i class="fas fa-robot me-2"></i>AI Помічник
          </button>
        </div>
      </div>
    </div>
  <div class="col-lg-6 text-center">
    <div class="hero-illustration">
      <i class="fas fa-book-open" style="font-size: 8rem; opacity:
0.3;"></i>
    </div>
  </div>

```

```

        </div>
    </div>
</div>
</section>

<!-- Features Section -->
<section class="py-5">
    <div class="container">
        <div class="row text-center">
            <div class="col-md-3 mb-4">
                <div class="feature-icon bg-primary text-white rounded-circle mx-auto
mb-3 d-flex align-items-center justify-content-center" style="width: 80px; height:
80px;">
                    <i class="fas fa-shipping-fast fa-2x"></i>
                </div>
                <h5>Швидка доставка</h5>
                <p class="text-muted">Доставка по всій Україні за 1-3 дні</p>
            </div>
            <div class="col-md-3 mb-4">
                <div class="feature-icon bg-primary text-white rounded-circle mx-auto
mb-3 d-flex align-items-center justify-content-center" style="width: 80px; height:
80px;">
                    <i class="fas fa-robot fa-2x"></i>
                </div>
                <h5>AI Помічник</h5>
                <p class="text-muted">Персональні рекомендації</p>
            </div>
            <div class="col-md-3 mb-4">

```

```
<div class="feature-icon bg-primary text-white rounded-circle mx-auto
mb-3 d-flex align-items-center justify-content-center" style="width: 80px; height:
80px;">
```

```
<i class="fas fa-shield-alt fa-2x"></i>
```

```
</div>
```

```
<h5>Безпечна оплата</h5>
```

```
<p class="text-muted">Захищені платежі всіма способами</p>
```

```
</div>
```

```
<div class="col-md-3 mb-4">
```

```
<div class="feature-icon bg-primary text-white rounded-circle mx-auto
mb-3 d-flex align-items-center justify-content-center" style="width: 80px; height:
80px;">
```

```
<i class="fas fa-headset fa-2x"></i>
```

```
</div>
```

```
<h5>24/7 Підтримка</h5>
```

```
<p class="text-muted">Цілодобова підтримка клієнтів</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</section>
```

```
<!-- Featured Books Section -->
```

```
{% if featured_books %}
```

```
<section class="py-5 bg-light">
```

```
<div class="container">
```

```
<div class="text-center mb-5">
```

```
<h2 class="fw-bold">Рекомендовані книги</h2>
```

```
<p class="text-muted">Наші експерти обрали найкращі книги для
вас</p>
```

```
</div>
```

```

<div class="row">
  {% for book in featured_books %}
  <div class="col-lg-4 col-md-6 mb-4">
    <div class="card book-card h-100">
      <div class="book-image">
        <i class="fas fa-book"></i>
      </div>
      <div class="card-body d-flex flex-column">
        <h5 class="book-title">{{ book.title }}</h5>
        <p class="book-author">{{ book.author }}</p>
        <div class="rating mb-2">
          {% for i in range(book.rating|int) %}
            <i class="fas fa-star"></i>
          {% endfor %}
          {% if book.rating % 1 %}
            <i class="fas fa-star-half-alt"></i>
          {% endif %}
          <span class="ms-1">({{ book.rating }})</span>
        </div>
        <p class="text-muted small mb-3">{{ book.description[:100] }} {%
if book.description|length > 100 %}...{% endif %}</p>
        <div class="mt-auto">
          <div class="d-flex justify-content-between align-items-center
mb-3">
            <span class="book-price">{{ book.price }} грн</span>
            <small class="text-muted">На складі: {{ book.stock
}}</small>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

        <a href="{{ url_for('book_detail', book_id=book.id) }}"
class="btn btn-outline-primary flex-fill">
        <i class="fas fa-eye me-1"></i>Переглянути
</a>
        {% if book.stock > 0 %}
<form method="POST" class="flex-fill">
        <input type="hidden" name="book_id" value="{{ book.id
}}">
        <input type="hidden" name="quantity" value="1">
        <button type="submit" class="btn btn-primary w-100 add-
to-cart-btn">
                <i class="fas fa-cart-plus me-1"></i>До кошика
        </button>
</form>
        {% else %}
<button class="btn btn-secondary flex-fill" disabled>
        Немає в наявності
        </button>
        {% endif %}
</div>
</div>
</div>
</div>
</div>
        {% endfor %}
</div>
<div class="text-center mt-4">
        <a href="{{ url_for('catalog') }}" class="btn btn-primary btn-lg">
                Переглянути весь каталог <i class="fas fa-arrow-right ms-2"></i>
        </a>

```

```

        </div>
    </div>
</section>
{% endif %}

<!-- Bestsellers Section -->
{% if bestsellers %}
<section class="py-5">
    <div class="container">
        <div class="text-center mb-5">
            <h2 class="fw-bold">Бестселери</h2>
            <p class="text-muted">Найпопулярніші книги серед наших
читачів</p>
        </div>
        <div class="row">
            {% for book in bestsellers %}
            <div class="col-lg-4 col-md-6 mb-4">
                <div class="card book-card h-100">
                    <div class="book-image">
                        <i class="fas fa-book"></i>
                        <div class="position-absolute top-0 end-0 p-2">
                            <span class="badge bg-warning">
                                <i class="fas fa-crown me-1"></i>Бестселер
                            </span>
                        </div>
                    </div>
                </div>
            <div class="card-body d-flex flex-column">
                <h5 class="book-title">{{ book.title }}</h5>
                <p class="book-author">{{ book.author }}</p>
                <div class="rating mb-2">

```

```

    {% for i in range(book.rating|int) %}
        <i class="fas fa-star"></i>
    {% endfor %}
    {% if book.rating % 1 %}
        <i class="fas fa-star-half-alt"></i>
    {% endif %}
    <span class="ms-1">({{ book.rating }})</span>
</div>
<p class="text-muted small mb-3">{{ book.description[:100] }} {%
if book.description|length > 100 %}...{% endif %}</p>
<div class="mt-auto">
    <div class="d-flex justify-content-between align-items-center
mb-3">
        <span class="book-price">{{ book.price }} грн</span>
        <small class="text-muted">На складі: {{ book.stock
}}</small>
    </div>
    <div class="d-flex gap-2">
        <a href="{{ url_for('book_detail', book_id=book.id) }}"
class="btn btn-outline-primary flex-fill">
            <i class="fas fa-eye me-1"></i>Переглянути
        </a>
        {% if book.stock > 0 %}
            <form method="POST" class="flex-fill">
                <input type="hidden" name="book_id" value="{{ book.id
}}">
                <input type="hidden" name="quantity" value="1">
                <button type="submit" class="btn btn-primary w-100 add-
to-cart-btn">
                    <i class="fas fa-cart-plus me-1"></i>До кошика

```

```

        </button>
    </form>
    {% else %}
    <button class="btn btn-secondary flex-fill" disabled>
        Немає в наявності
    </button>
    {% endif %}
</div>
</div>
</div>
</div>
</div>
    {% endfor %}
</div>
</div>
</section>
{% endif %}
<!-- Call to Action Section -->
<section class="py-5 bg-primary text-white">
    <div class="container text-center">
        <h2 class="fw-bold mb-4">Потрібна допомога з вибором книги?</h2>
        <p class="lead mb-4">Наш AI помічник допоможе знайти ідеальну книгу
саме для вас!</p>
        <button class="btn btn-light btn-lg"
onclick="document.getElementById('chatToggle').click()">
            <i class="fas fa-robot me-2"></i>Запитати у AI помічника
        </button>
    </div>
</section>
{% endblock %}

```