

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 – Комп’ютерні науки,
освітньо-наукова програма «Управління проєктами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження процесів управління проєктом створення вебплатформи для
діяльності салонів краси»

Студентки 2-го курсу групи УП-22

Олександри МИРУТЕНКО

(ім'я, прізвище)

(підпис студента)

Науковий керівник:

К.Т.Н., доцент

(науковий ступінь, вчене звання)

Вадим ЗЮЗЮН

(ім'я, прізвище)

(дата)

(підпис)

Попередній захист:

(Висновок: "До захисту в Екзаменаційній комісії")

Завідувач кафедри
технологій управління

(підпис)

Віктор МОРОЗОВ

(ім'я, прізвище)

(дата)

Київ 2025

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій**

Кафедра технологій управління

Освітній рівень Магістр

Спеціальність 122 Комп'ютерні науки

Освітньо-наукова програма Управління проєктами

ЗАТВЕРДЖУЮ

Завідувач кафедри

професор Віктор МОРОЗОВ

“27” листопада 2024 року

**З А В Д А Н Н Я
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

Студент: Мирутенко Олександра Володимирівна

Група: УП-22

1. Тема кваліфікаційної роботи: «Дослідження процесів управління проєктом створення вебплатформи для діяльності салонів краси».

Затверджена протоколом кафедри ТУ від 26 листопада 2024 року № 5.

2. Строк подання студентом готової роботи – “19” травня 2025 року.

3. Цільова установка та вихідні дані до роботи: дослідження різних методів та інструментів для управління проєктом розробки інформаційної системи, їх використання у плануванні проєкту, ресурсів, бюджету та управлінні ризиками.

4. Зміст роботи: дослідження конкурентного середовища, обґрунтування доцільності та життєздатності проєкту, загальний опис проєкту, аналіз зацікавлених сторін проєкту, проведення SWOT-аналізу, вибір методології управління проєктом, формування беклогу продукту проєкту, організаційна структура проєкту, планування робіт проєкту у вигляді спринтів, управління ресурсами та бюджетування проєкту, управління ризиками проєкту. Розробка інформаційної структури проєкту, створення концептуальної моделі бази даних, проєктування інтерфейсу користувача.

5. Перелік графічного матеріалу: актуальність, концептуальна та математична модель інформаційної системи, процес функціонування системи, User Story проєкту, беклог продукту проєкту, OBS структура проєкту, планування спринтів та діаграма Ганта, ризики проєкту, архітектура інформаційної системи, інформація про базу даних, сторінки сайту.

6. Календарний план виконання роботи

№ з/п	Назва частин роботи	План виконання роботи
1	Вивчення літературних джерел з предмету дослідження	22.01.25-22.02.25
2	Збір і вивчення матеріалів досліджуваного підприємства	22.01.25-26.01.25
3	Складання плану кваліфікаційної роботи	15.01.25-22.01.25
4	Ознайомлення наукового керівника з розгорнутим планом кваліфікаційної роботи. Внесення змін	22.01.25
5	Підготовка розділу 1	05.02.25-26.02.25
6	Підготовка розділу 2	27.02.25-16.03.25
7	Підготовка розділу 3	17.03.25-06.04.25
8	Підготовка розділу 4	07.04.25-25.04.25
9	Оформлення кваліфікаційної роботи	28.04.25-07.05.25

10	Передача кваліфікаційної роботи науковому керівникові	08.05.25
11	Передача кваліфікаційної роботи рецензенту для рецензування	12.05.25
12	Захист кваліфікаційної роботи	26.05.24-28.05.25

Дата видачі завдання «28» листопада 2024 р.

Керівник роботи _____ доцент Вадим ЗЮЗІОН
(посада, ім'я, прізвище)

 (підпис)

Завдання прийняв до виконання студент групи УП-22

_____ Олександра МИРУТЕНКО
(ім'я, прізвище)

 (підпис)

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему

«Дослідження процесів управління проєктом створення вебплатформи для діяльності салонів краси»

Студент: Мирутенко Олександра Володимирівна

Науковий керівник: Зюзюн Вадим Ігорович

Рік захисту – 2025

Метою кваліфікаційної роботи є розробка ефективної та зручної у використанні вебплатформи для салону краси «MOV Hair Studio», яка забезпечить автоматизацію процесів запису на прийом, покращить комунікацію з клієнтами, підвищить їхню задоволеність послугами, а також сприятиме збільшенню клієнтської бази і прибутковості салону через впровадження онлайн-маркетингових інструментів та програм лояльності.

Ціль проєкту – розробити та впровадити інтерактивну вебплатформу для салону краси «MOV Hair Studio» з метою повної автоматизації процесу бронювання, підвищення якості обслуговування клієнтів, покращення управління клієнтськими взаєминами та підтримки маркетингової діяльності, що дозволить забезпечити високий рівень задоволеності користувачів, збільшити кількість онлайн-бронювань та оптимізувати внутрішні бізнес-процеси впродовж дев'яти місяців від старту проєкту.

Наукова новизна дослідження полягає в розробці обґрунтованої концептуальної моделі вебпорталу діяльності салону краси, що передбачає інтеграцією адаптованих методів NLP (обробки та аналізу відгуків) з джерел синхронізованих з вебпорталом.

Практична цінність полягає в: 1) автоматизації запису на прийом, яка зменшить адміністративне навантаження на персонал салону, знизить ймовірність помилок і дозволить ефективніше керувати розкладом; 2) інтеграції онлайн-маркетингових інструментів та програм лояльності, що сприятиме залученню нових клієнтів і утриманню постійних, що дозволить салону виділитися серед конкурентів; 3) завдяки покращенню обслуговування клієнтів і оптимізації бізнес-процесів, платформа сприятиме зростанню доходів салону за рахунок збільшення кількості клієнтів, ефективнішого управління ресурсами та можливості придбати засоби догляду через інтернет-магазин, як ресурс

вебплатформи; 4) система збору та аналізу відгуків клієнтів допоможе салону вчасно реагувати на побажання і зауваження, що сприятиме постійному покращенню якості послуг.

Кваліфікаційної робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, переліку використаних інформаційних джерел та додатків.

У першому розділі проведено комплексне дослідження предметної області, зокрема аналіз існуючих програмних рішень, літературних та інформаційних джерел щодо способів вирішення виявлених проблем. Також виконано SWOT-аналіз проєкту, ідентифіковано зацікавлені сторони, поставлено задачі дослідження та сформульовано технічне завдання у вигляді паспорту проєкту.

У другому розділі розроблено концептуальну модель інформаційної системи, здійснено математичну постановку задачі відповідно до розробленої моделі та представлено застосування методів моделювання у контексті функціонування салону краси «MOV Hair Studio».

В третьому розділі було обґрунтовано доцільність вибору технології управління проєктом, визначено функціональні та нефункціональні вимоги до майбутнього продукту, сформовано беклог, описано організаційну структуру управління проєктом, сплановано ресурси, бюджет та спринти з використанням MS Project. Окрему увагу приділено ідентифікації ризиків та розробці карти їх управління.

В четвертому розділі запропоновано технічну реалізацію вебплатформи як кінцевого продукту проєкту. В межах цього розділу розроблено алгоритм роботи сайту, побудовано концептуальну та даталогічну моделі бази даних, охарактеризовано фреймворк Python Django як основний інструмент розробки, а також реалізовано back-end та front-end частини вебплатформи.

Робота містить 100 сторінок без додатків, 26 рисунки та 26 таблиць. Додатки складають 16 сторінок.

Ключові слова: інформаційна система, вебплатформа, Django, математичне моделювання, управління проєктом, беклог, ризики, салон краси, технічне завдання, база даних.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ	12
1.1 Дослідження існуючих програмних рішень та формування предметної області	12
1.2 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем	14
1.3 Ідентифікація зацікавлених сторін проєкту	17
1.4 SWOT-аналіз проєкту	19
1.5 Постановка задачі дослідження. Формулювання технічного завдання на розробку у вигляді паспорту проєкту	25
РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ ...	30
2.1 Розробка концептуальної моделі інформаційної системи	30
2.2 Постановка задачі в математичному вигляді відповідно до концептуальної моделі	34
2.3 Використання методів моделювання розроблених моделей в аспекті діяльності салону краси «MOV Hair Studio»	37
РОЗДІЛ 3. ВИБІР ТЕХНОЛОГІЇ УПРАВЛІННЯ ПРОЄКТОМ ТА РОЗРОБКА ЙОГО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ	41
3.1 Вибір технології управління проєктом	41
3.2 Визначення функціональних та нефункціональних вимог до продукту проєкту	44
3.3 Формування беклогу продукту проєкту	48
3.4 Організаційна структура управління проєктом	52
3.5 Планування ресурсного забезпечення проєкту та бюджету	55
3.6 Планування спринтів та візуалізація їх виконання за допомогою інструментів MS Project	58
3.7 Ідентифікація проєктних ризиків та розробка карти управління ними .	61

РОЗДІЛ 4. РОЗРОБКА ВЕБПЛАТФОРМИ ЯК ПРОДУКТУ РЕАЛІЗАЦІЇ	
ПРОЄКТУ	69
4.1 Формування алгоритмів роботи вебпорталу	69
4.2 Концептуальне та даталогічне моделювання бази даних продукту проєкту	72
4.3 Характеристика Python Django як інструменту розробки вебплатформи	81
4.4 Розробка back-end частини вебплатформи	85
4.5 Розробка front-end частини вебплатформи	88
ВИСНОВКИ	93
ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ	95
ДОДАТКИ	102

ВСТУП

Актуальність теми дослідження. Актуальність розробки вебплатформи для салону краси з догляду за волоссям «MOV Hair Studio» полягає в її здатності задовольнити зростаючі потреби сучасних клієнтів у зручних та ефективних онлайн-сервісах. З появою пандемії COVID-19 цифровізація бізнесу стала критично важливою, оскільки клієнти все більше надають перевагу онлайн-бронюванню та консультаціям. Вебплатформа дозволить автоматизувати процес запису на прийом, зменшуючи адміністративне навантаження на працівників салону та мінімізуючи помилки, пов'язані з людським фактором. Крім того, клієнти матимуть можливість отримувати персоналізовані рекомендації щодо догляду за волоссям, що підвищить їхню задоволеність та лояльність до салону. Інтеграція з соціальними мережами та системами відгуків дозволить ефективно збирати та аналізувати зворотний зв'язок від клієнтів, що сприятиме підвищенню якості послуг. Враховуючи швидкий розвиток технологій та конкуренцію в індустрії краси, наявність сучасної вебплатформи стане суттєвою перевагою для «MOV Hair Studio». Платформа також сприятиме розширенню клієнтської бази за рахунок онлайн-маркетингу та рекламних кампаній. Вона дозволить впровадити програми лояльності та спеціальні пропозиції, що забезпечить додатковий прибуток для бізнесу. Загалом, створення такої платформи є важливим кроком для підвищення конкурентоспроможності салону краси «MOV Hair Studio» в умовах сучасного ринку. На основі вище зазначеного було сформовано мету та завдання дослідження.

Метою даного дослідження є розробка ефективної та зручної у використанні вебплатформи для салону краси «MOV Hair Studio», яка забезпечить автоматизацію процесів запису на прийом, покращить комунікацію з клієнтами, підвищить їхню задоволеність послугами, а також сприятиме збільшенню клієнтської бази і прибутковості салону через впровадження онлайн-маркетингових інструментів та програм лояльності.

Для досягнення мети потрібно виконати наступні **задачі**:

- провести дослідження програмних рішень у сфері керування салонами краси, визначити їхні сильні та слабкі сторони, а також сформувати предметну область для подальшої розробки;
- проаналізувати літературні та інформаційні джерела щодо можливостей вирішення виявлених проблем, а також здійснити SWOT-аналіз проєкту та ідентифікувати зацікавлені сторони;
- сформулювати технічне завдання у вигляді паспорта проєкту, розробити концептуальну модель інформаційної системи та визначити математичну постановку задачі;
- обґрунтувати вибір методології управління проєктом, визначити організаційну структуру управління, сформувати беклог продукту та спланувати спринти з використанням інструментів MS Project;
- ідентифікувати можливі ризики, які можуть виникнути під час розробки та впровадження вебплатформи, а також розробити карту управління ризиками;
- створити алгоритм роботи сайту, розробити концептуальну та даталогічну модель бази даних, а також визначити особливості використання Python Django як основного інструменту розробки;
- реалізувати back-end частину вебплатформи на основі Django, розробити front-end інтерфейс та забезпечити інтеграцію компонентів для коректної роботи системи.

Об'єктом дослідження є процеси розробки і впровадження вебплатформи для діяльності салону краси з догляду за волоссям «MOV Hair Studio».

Предметом дослідження є методи та процеси управління проєктом, щодо розробки і впровадження вебплатформи для діяльності салону краси з догляду за волоссям «MOV Hair Studio».

Наукова новизна дослідження полягає в розробці обґрунтованої концептуальної моделі вебпорталу діяльності салону краси, що передбачає інтеграцією адаптованих методів NLP (обробки та аналізу відгуків) з джерел синхронізованих з вебпорталом.

Практичне значення отриманих результатів полягатиме у:

1) автоматизації запису на прийом, яка зменшить адміністративне навантаження на персонал салону, знизить ймовірність помилок і дозволить ефективніше керувати розкладом;

2) інтеграції онлайн-маркетингових інструментів та програм лояльності, що сприятиме залученню нових клієнтів і утриманню постійних, що дозволить салону виділитися серед конкурентів;

3) завдяки покращенню обслуговування клієнтів і оптимізації бізнес-процесів, платформа сприятиме зростанню доходів салону за рахунок збільшення кількості клієнтів, ефективнішого управління ресурсами та можливості придбати засоби догляду через інтернет-магазин, як ресурс вебплатформи;

4) система збору та аналізу відгуків клієнтів допоможе салону вчасно реагувати на побажання і зауваження, що сприятиме постійному покращенню якості послуг.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

1.1 Дослідження існуючих програмних рішень та формування предметної області

Розробка вебпорталу для салону краси вимагає ґрунтовного дослідження існуючих програмних рішень на ринку та формування чіткої предметної області. На сьогоднішній день існує чимало платформ, які пропонують автоматизацію різних аспектів діяльності салонів краси. Вони зазвичай включають функціонал для бронювання послуг, управління даними клієнтів, розкладу працівників, а також аналізу зворотного зв'язку від клієнтів.

Одним з найбільш відомих рішень є система Booksy, яка дозволяє клієнтам бронювати послуги онлайн, отримувати нагадування про майбутні візити та залишати відгуки. Наприклад, клієнти можуть через мобільний додаток Booksy переглядати розклад майстрів та обирати зручний для себе час, що значно спрощує процес бронювання і зменшує кількість пропущених записів. Fresha також є популярною платформою, що надає функціонал для управління розкладом працівників, фінансами та програмами лояльності. Вона пропонує безкоштовну модель для салонів з можливістю оплати тільки за фактично надані послуги, що робить її привабливою для невеликих бізнесів.

Salon Iris пропонує повний спектр інструментів для автоматизації бізнесу, включаючи облік витрат, управління запасами та інтеграцію з POS-системами. Ця система, наприклад, дозволяє відстежувати запаси косметичних засобів і автоматично замовляти нові, коли рівень запасів знижується до певного значення, що забезпечує безперервну роботу салону без затримок.

Однак, незважаючи на широкий вибір програмних продуктів, більшість з них має певні обмеження. Вони часто пропонують стандартний набір функцій без можливості гнучкого налаштування під конкретні потреби

салону. Також багато рішень не враховують сучасні методи обробки природної мови (NLP) для автоматичного аналізу клієнтських відгуків, що є важливим аспектом для виявлення прихованих тенденцій та потреб клієнтів.

В нашому дослідженні ми зосередимося на розробці вебпорталу, який буде враховувати всі ці аспекти та надаватиме більш інноваційні та адаптивні функції. Основною метою є створення платформи, яка не лише автоматизує рутинні процеси, але й забезпечить салон краси інструментами для аналізу та покращення обслуговування на основі клієнтських відгуків.

Формування предметної області включає визначення ключових компонентів системи, таких як модулі для бронювання послуг, управління даними клієнтів та працівників, обробка зворотного зв'язку та аналіз ефективності роботи салону. Кожен з цих модулів буде розроблений з використанням сучасних технологій, включаючи алгоритми NLP для автоматичного аналізу текстових даних.

Для забезпечення високої якості обслуговування клієнтів, вебплатформа буде інтегруватися з соціальними мережами та іншими платформами для збору відгуків. Наприклад, інтеграція з Facebook та Instagram дозволить автоматично збирати коментарі клієнтів про послуги салону і аналізувати їх за допомогою NLP-алгоритмів для виявлення ключових тем та емоцій. Це дозволить отримувати більш повну картину про задоволеність клієнтів та оперативно реагувати на їхні потреби.

Використання технологій машинного навчання дозволить виявляти приховані тенденції у відгуках та пропонувати рекомендації щодо покращення сервісу. Наприклад, якщо аналіз відгуків виявить, що багато клієнтів незадоволені часом очікування, система може рекомендувати збільшити кількість майстрів у певні години пік.

Крім того, важливою складовою є управління програмами лояльності та проведення маркетингових кампаній. Вебплатформа буде включати інструменти для створення таргетованих рекламних матеріалів та управління спеціальними пропозиціями для постійних клієнтів. Наприклад, на основі

аналізу історії відвідувань та вподобань клієнтів, система зможе автоматично генерувати індивідуальні пропозиції та знижки для конкретних клієнтів, підвищуючи їхню лояльність.

Таким чином, дослідження в кваліфікаційній роботі та розробка вебпорталу для салону краси зосереджуються на створенні комплексного рішення, яке забезпечить високу якість обслуговування, ефективне управління бізнес-процесами та використання сучасних технологій для аналізу клієнтських відгуків. Це дозволить салону краси оперативно адаптуватися до змінних очікувань клієнтів та підвищити рівень їхньої задоволеності.

1.2 Проведення аналізу літературних та інформаційних джерел щодо можливостей вирішення виявлених проблем

Проблематику яка розглядається в межах курсового проєкту досліджували велика кількість науковців та практиків, тому проведено певне узагальнення опрацьованих інформаційних та літературних джерел.

Авторський колектив праці [1] зазначає, що розробка веб-сторінок є надзвичайно кропіткою працею, але правильно вибране середовище розробки з необхідними плагінами, здатність логічно структурувати майбутній сайт, а також використання різноманітних фреймворків або їх окремих компонентів значно спрощують цю роботу та роблять її більш доступною. Аналіз тенденцій розвитку інструментів для створення веб-сторінок свідчить про подальше спрощення процесу розробки та підвищення якості кінцевого продукту в майбутньому.

У дослідженні [2] автор розглядає поняття надійності веб-сайту в аспекті його «вузького» і «широкого» трактування, а також критерії оцінювання надійності та метрики її вимірювання. Автором проаналізовано методи оцінювання надійності веб-сайтів, розроблено багатокритеріальну модель оцінювання надійності веб-сайтів, проведено апробацію моделі оцінювання надійності.

У дослідженні [3] авторами приділено увагу розробці безпечних та ефективних механізмів інтеграції платіжних систем, які підвищують надійність та зручність використання.

В науковому дослідженні [4] зазначено, що ефективність функціонування сайтів залежить від завчасно коректно сформованої його структури, а також відповідно чіткості дизайнерських аспектів і функціоналу. Також важливе значення має логіка побудови функціоналу.

А в праці [5] проаналізовано наявні на ринку платні і безкоштовні програмні продукти для створення веб-сайтів на предмет їх функціональності, розглянуто їх переваги та недоліки.

Досліджуючи публікацію [6] можна ознайомитися із п'яти категоріями веб-сайтів та особливостями роботи із кожним із них.

В публікації [7] зазначається, що ландшафт створення веб-сайтів різко змінився з появою штучного інтелекту (AI) та безкодових рішень. Вміння кодувати або залежність від веб-дизайнерів відійшли в минуле. Сьогодні платформи без коду, керовані штучним інтелектом, дозволяють швидко та легко створювати привабливу онлайн-присутність.

Також варто виділити ресурси [8-14] функціонування яких направлено на надання корисної та актуальної інформації щодо створення веб-ресурсів, а також забезпечення їх якісної та продуктивної роботи.

У джерелі [15] розглядається ефективність використання технологій обробки природної мови (NLP) та систем штучного інтелекту (ШІ) для аналізу рекламних кампаній у середовищі інтернет-реклами. Автори досліджують можливості сучасних інструментів для виявлення ключових слів у текстах рекламних оголошень, що дозволяє підвищити релевантність та безпеку рекламного контенту. Особливу увагу приділено комбінації NLP-інструментів із моделями машинного навчання, що спеціалізуються на ідентифікації шахрайських або потенційно шкідливих оголошень, які можуть з'являтися на вебсерверах систем обміну реклами.

У публікації наведено приклади метрик, які застосовуються для автоматичної оцінки вмісту оголошень з метою виявлення небажаного чи загрозового контенту. Автори здійснюють порівняльний аналіз наявних рішень та демонструють результати їх застосування на основі реального кейсу з оголошенням, що має підвищений рівень ризику. На основі отриманих результатів сформульовано висновки щодо ефективності та доцільності інтеграції таких технологій у бізнес-логіку рекламних мереж з метою покращення якості контенту та зменшення ризиків для кінцевих користувачів. В праці [16] звертається увага на проблему неконсистентного оцінювання NLP-моделей, що використовується для вирішення завдань у програмній інженерії. Відсутність єдиного підходу до метрик та інтерпретації результатів ускладнює порівняння ефективності моделей. Автори наголошують на необхідності розробки методологічного та стандартизованого підходу до оцінювання, що дозволить забезпечити об'єктивність і покращити якість досліджень у цій галузі.

У дослідженні [17] розглядається створення інтелектуального чат-бота для CRM-системи на основі технологій NLP. Було протестовано кілька підходів до навчання моделей, використано сучасні бібліотеки та алгоритми, зокрема RASA, Chatterbot, косинусну схожість і векторні представлення від Google. Проведено аналіз переваг та недоліків різних методів інтеграції та розгортання, визначено оптимальні рішення для практичного використання чат-бота в автоматизованій підтримці користувачів. У праці [18] представлено інноваційну систему парсингу резюме, що враховує макет документа. Система поєднує NLP та rule-based підходи для вилучення змісту за розділами, забезпечуючи повноцінний аналіз усієї інформації, на відміну від традиційних рішень, які фокусуються лише на сутностях. Результати парсингу можуть бути використані для автоматичної перевірки та оцінювання резюме, що особливо актуально для кандидатів в IT-сфері.

Створення універсальної NLP-платформи для аналізу текстових даних у сфері бізнесу розглянуто у праці [19]. Вона об'єднує функції визначення

емоцій, видобування інформації та резюмування, дозволяючи будувати візуалізації, що полегшують інтерпретацію даних. Платформа особливо корисна в освітньому середовищі, де студенти та викладачі зможуть ефективніше аналізувати ринкові тенденції та приймати обґрунтовані рішення на основі текстового контенту.

У дослідженні [20] представлено вебплатформу Text Processing Hub, яка об'єднує ключові інструменти NLP у єдиному середовищі, з орієнтацією на освітній сектор. Вона вирішує проблему розпорошення інструментів, надаючи доступні, ефективні та безпечні функції для роботи з текстами. Платформа підтримує аналіз емоцій, генерацію тексту, перевірку орфографії та допомагає студентам і викладачам у навчанні, аналізі та прийнятті рішень, роблячи її незамінною в міждисциплінарному контексті.

Таким чином, аналіз показує, що наявні наукові дослідження та прикладні рішення здатні суттєво сприяти вирішенню проблем, пов'язаних із ефективністю, надійністю, безпечністю та функціональністю вебплатформ. Особливо перспективним є інтеграція сучасних технологій NLP і ШІ в архітектуру таких систем, що відкриває нові горизонти для створення інтелектуальних, адаптивних і масштабованих онлайн-рішень у різних сферах діяльності.

1.3 Ідентифікація зацікавлених сторін проєкту

Успішна реалізація будь-якого ІТ-проєкту значною мірою залежить від точного визначення та глибокого розуміння зацікавлених сторін (stakeholders). Вебплатформа для салону краси охоплює різні аспекти – від надання зручного сервісу клієнтам до оптимізації внутрішніх процесів управління. Кожна група стейкхолдерів має свої очікування, потреби та вплив на проєкт. Аналіз зацікавлених сторін дозволяє виявити основні категорії учасників, оцінити їх інтереси, рівень впливу на проєкт та ступінь залученості, що в подальшому стане основою для ефективної комунікації та управління проєктними

ризиками. В працях [21-25] розглянуто питання щодо управління зацікавленими сторонами, оцінки комунікаційних аспектів роботи із зацікавленими сторонами та інші актуальні питання.

Аналіз зацікавлених сторін демонструє багаторівневу структуру взаємодії між учасниками проєкту. Найвищий вплив на проєкт мають власник салону, розробники платформи та клієнти, адже саме їхні потреби є ключовими для досягнення цілей. Водночас адміністратори та майстри безпосередньо працюють з платформою, тому доцільно забезпечити їхню активну участь на етапах тестування і впровадження.

Таблиця 1.1

Аналіз зацікавлених сторін проєкту

№	Зацікавлена сторона	Роль у проєкті	Інтереси та очікування	Вплив на проєкт	Ступінь залученості	Стратегія взаємодії
1	2	3	4	5	6	7
1	Власник салону краси	Ініціатор, фінансист	Підвищення прибутковості, автоматизація, контроль процесів	Високий	Висока	Регулярні зустрічі, затвердження рішень
2	Адміністратори салону	Користувачі системи	Зручний інтерфейс, зменшення рутинної роботи	Середній	Висока	Тестування, опитування
3	Майстри (перукарі, візажисти тощо)	Виконавці послуг	Чіткий графік, мінімізація помилок у бронювання	Середній	Середня	Збір фідбеку, тренінги
4	Клієнти салону	Кінцеві користувачі	Зручність онлайн-бронювання, знижки, нагадування	Високий	Низька	UX-дослідження, аналіз відгуків
5	Розробники платформи	Технічні виконавці	Реалізація техзавдання, стабільність, підтримка	Високий	Висока	Щоденні стендапи, технічна документація
6	Маркетинговий відділ	Промоція, аналіз користувацьких даних	Доступ до аналітики, інтеграція з соцмережами	Середній	Середня	Обговорення функціоналу звітності
7	Постачальники програмного забезпечення	Партнери	Ліцензування, підтримка API	Низький	Низька	Контракти, технічна підтримка

1	2	3	4	5	6	7
8	Дизайнери інтерфейсу	UX/UI розробка	Якість візуального досвіду, адаптивність	Середній	Висока	Етапи прототипування та юзабіліті-тестування
9	Регуляторні органи	Контроль законодавчої відповідності	Виконання норм зберігання персональних даних (GDPR тощо)	Середній	Низька	Юридичний аудит

Окрему увагу слід приділити UX/UI-дизайнерам та маркетологам, оскільки вони впливають на залучення нових клієнтів та підтримку візуальної привабливості сервісу. Хоча регуляторні органи мають низьку залученість, ігнорування юридичних вимог може створити суттєві ризики, зокрема, штрафи та блокування сервісу.

Таким чином, комплексний аналіз дозволяє сформувати ефективну стратегію управління стейкхолдерами, забезпечити прозору комунікацію між усіма сторонами та зменшити ризики, пов'язані з непорозумінням чи конфліктом інтересів.

1.4 SWOT-аналіз проєкту

У сучасних умовах цифровізації бізнес-процесів SWOT-аналіз є одним із найефективніших інструментів стратегічного планування, який дозволяє системно оцінити внутрішні переваги та недоліки проєкту, а також зовнішні можливості і загрози, що можуть вплинути на його реалізацію. Особливої актуальності такий аналіз набуває в проєктах, пов'язаних з інформаційними технологіями та сферою послуг, де успішність залежить як від технічної реалізації, так і від гнучкості в реагуванні на ринкові зміни та потреби споживача.

Проєкти зі створення вебплатформ для салонів краси поєднують у собі дві ключові складові: технічну (розробка функціонального та надійного цифрового продукту) та бізнесову (вдосконалення сервісу, підвищення

лояльності клієнтів, оптимізація операційної діяльності). Такий тип рішень покликаний забезпечити нову якість обслуговування, надати зручний інструмент для запису, комунікації та персоналізації взаємодії з клієнтами. Водночас розробка платформи потребує врахування ряду факторів: конкурентного середовища, вимог користувачів, можливих ризиків та ресурсних обмежень. Застосування SWOT-аналізу в цьому контексті дозволяє не лише сформулювати стратегічні орієнтири розвитку системи, а й проаналізувати потенційні сценарії її впровадження [26-31]. Це дає змогу оптимізувати підхід до проєктування платформи, ефективно використовувати наявні ресурси та заздалегідь підготуватися до ймовірних викликів. Аналіз базується як на результатах дослідження ринку послуг краси, так і на вивченні найкращих практик у сфері веброзробки для малого та середнього бізнесу.

В табл. 1.2 наведено характеристику сильних сторін проєкту.

Таблиця 1.2

Сильні сторони (Strengths) проєкту

Код	Характеристика сильних сторін
S01	Індивідуальна розробка під конкретний бізнес-процес (платформа створюється з урахуванням специфіки саме обраного салону).
S02	Можливість онлайн-запису клієнтів (зручний інтерфейс для бронювання послуг у будь-який час).
S03	Інтеграція з CRM (облік клієнтів, історії відвідувань, формування статистики та звітності).
S04	Покращення клієнтського досвіду (нагадування про записи, знижки, персоналізовані пропозиції).
S05	Підвищення ефективності управління (оптимізація графіків майстрів, управління ресурсами).
S06	Можливість SEO-оптимізації (вебплатформа може бути ефективним маркетинговим інструментом).
S07	Мобільна адаптивність (коректне відображення на смартфонах і планшетах).
S08	Можливість розширення функціоналу (додавання нових модулів (магазин косметики, відгуки тощо)).
S09	Підвищення іміджу бренду (сучасний веб-сайт створює враження професіоналізму).
S10	Автоматизація внутрішніх процесів (формування фінансових звітів, контроль розкладу, онлайн-оплата).

В табл. 1.3 наведено характеристику слабких сторін проєкту.

Таблиця 1.3

Слабкі сторони (Weaknesses) проєкту

Код	Характеристика слабких сторін
W01	Висока початкова вартість розробки (індивідуальна вебплатформа потребує інвестицій).
W02	Потреба в постійному технічному супроводі (оновлення, безпека, підтримка).
W03	Низький рівень цифрової грамотності частини персоналу (потреба в навчанні).
W04	Можливі труднощі з інтеграцією в існуючі бізнес-процеси (необхідність адаптації).
W05	Ризик перенавантаження функціоналом (зайва складність інтерфейсу для клієнтів чи адміністратора).
W06	Обмежені ресурси команди на етапі запуску (кадрові або часові обмеження).
W07	Можлива технічна застарілість при довгій розробці (ризик втрати актуальності технологій).

В табл. 1.4 наведено характеристику можливостей проєкту.

Таблиця 1.4

Можливості (Opportunities) проєкту

Код	Характеристика можливостей
O01	Залучення нових клієнтів через онлайн-присутність (розширення аудиторії).
O02	Інтеграція з соцмережами та месенджерами (зручна комунікація з клієнтами).
O03	Аналітика поведінки клієнтів (персоналізація сервісу на основі зібраних даних).
O04	Партнерські програми з косметичними брендами (додаткові джерела доходу).
O05	Впровадження штучного інтелекту (розумне планування завантаження майстрів).
O06	Розробка мобільного додатку (розширення функціональності платформи).
O07	Підключення платіжних систем та розстрочок (зростання кількості замовлень).
O08	Автоматизація маркетингових кампаній (push-сповіщення, розсилки).
O09	Розширення на інші види послуг (SPA, масаж, косметологія).
O10	Можливість виходу на ринок франшиз (масштабування рішень на інші салони).

В табл. 1.5 наведено характеристику загроз для проєкту.

Таблиця 1.4

Загрози (Threats) для проєкту

Код	Характеристика загроз для проєкту
1	2
T01	Конкуренція з готовими SaaS-рішеннями (платформи типу YCLIENTS чи Glossy).
T02	Швидкі зміни в цифрових технологіях (необхідність постійної адаптації).
T03	Кібератаки та загрози безпеці персональних даних.

1	2
T04	Економічна нестабільність (зниження платоспроможності клієнтів).
T05	Низька лояльність клієнтів у цифровому середовищі (висока конкуренція за увагу).
T06	Можливе законодавче регулювання щодо обробки персональних даних
T07	Сезонність попиту на окремі види послуг (нерівномірне навантаження).

SWOT-аналіз (Strengths, Weaknesses, Opportunities, Threats) є ключовим інструментом стратегічного планування, який дозволяє виявити внутрішні та зовнішні чинники, що впливають на ефективність проєкту. У випадку розробки вебплатформи для салону краси, цей вид аналізу допомагає всебічно оцінити як технічні, так і бізнесові аспекти системи, а також передбачити виклики, які можуть виникнути при впровадженні. Через високий рівень конкуренції в індустрії краси, необхідність цифровізації бізнес-процесів, а також зростаючі очікування клієнтів щодо якості обслуговування, проведення SWOT-аналізу дає змогу сформувати ефективну стратегію реалізації проєкту та мінімізувати ризики.

Після детального визначення сильних і слабких сторін, а також аналізу зовнішніх можливостей і загроз, доцільно здійснити попарне зіставлення відповідних елементів SWOT-аналізу. Такий підхід дозволяє виявити критичні взаємозв'язки між внутрішніми характеристиками проєкту та зовнішніми умовами його реалізації. Наведені нижче приклади демонструють, як сильні сторони можуть бути використані для реалізації можливостей або протистояння загрозам, а також як слабкі сторони можуть посилювати зовнішні ризики або бути пом'якшені за рахунок наявних шансів для розвитку.

S01–O06. Індивідуальна розробка під специфіку салону (S01) дозволяє безперешкодно реалізувати розширення функціональності через мобільний застосунок (O06), що забезпечить більшу залученість клієнтів та покращення сервісу.

S02–O01. Можливість онлайн-запису клієнтів (S02) і наявність онлайн-присутності (O01) у синергії сприяють значному розширенню клієнтської бази, особливо серед молодшої аудиторії, яка віддає перевагу цифровим каналам обслуговування.

S03–O03. Інтеграція з CRM (S03) дозволяє ефективно використовувати аналітику поведінки клієнтів (O03) для створення персоналізованих пропозицій та підвищення лояльності.

S05–O05. Оптимізація управління персоналом (S05) у поєднанні з можливостями впровадження штучного інтелекту (O05) створює передумови для автоматизованого формування розкладів та зменшення конфліктів у записах.

S08–O09. Можливість розширення функціоналу (S08) узгоджується з потенціалом виходу на ринки суміжних послуг (SPA, масаж, косметологія) (O09), що відкриває перспективи масштабування бізнесу.

S04–T05. Покращення клієнтського досвіду (S04) слугує бар'єром проти загрози низької цифрової лояльності клієнтів (T05), дозволяючи утримувати користувачів за рахунок зручності та персоналізації.

S06–T02. SEO-оптимізація (S06) зменшує ризик втрати конкурентоспроможності через технологічні зміни (T02), забезпечуючи стабільну видимість ресурсу в пошукових системах навіть при частих оновленнях алгоритмів.

S10–T03. Автоматизація внутрішніх процесів (S10) може бути вразливою до кіберзагроз (T03), однак при належному рівні безпеки зменшує людський фактор у зберіганні критичних даних.

S09–T01. Підвищення іміджу бренду через сучасний вебсайт (S09) дає конкурентну перевагу навіть на тлі присутності SaaS-рішень (T01), оскільки дозволяє створити унікальний стиль комунікації з клієнтом.

W01–O04. Висока початкова вартість розробки (W01) може бути частково компенсована завдяки партнерським програмам з косметичними брендами (O04), які зацікавлені у просуванні через цифрові платформи.

W03–O08. Низький рівень цифрової грамотності персоналу (W03) може бути нівельований через впровадження автоматизованих маркетингових кампаній (O08), які потребують мінімального втручання з боку працівників.

W06–O07. Обмежені ресурси команди на етапі запуску (W06) можуть бути оптимізовані за рахунок впровадження онлайн-оплати та розстрочок (O07), що забезпечить оперативні надходження коштів на підтримку процесу запуску.

W04–T06. Складнощі з інтеграцією в існуючі бізнес-процеси (W04) посилюються при запровадженні законодавчих вимог до обробки персональних даних (T06), що потребує ретельного юридичного супроводу.

W07–T02. Можливе технологічне застарівання (W07) на фоні стрімкого розвитку технологій (T02) створює серйозну загрозу для стабільності проєкту та вимагає впровадження модульної архітектури з можливістю оновлень.

Аналіз комбінацій елементів SWOT показує, що проєкт має потужну базу для розвитку за рахунок власних сильних сторін, які узгоджуються з ключовими можливостями цифрового ринку. Особливо перспективними є напрями мобільної адаптації, автоматизації аналітики та виходу на нові категорії послуг. Водночас наявні слабкі сторони потребують цілеспрямованої роботи над адаптацією внутрішніх процесів, цифровим навчанням персоналу та гнучкістю технічного рішення.

Загрози зовнішнього середовища (технологічна турбулентність, конкуренція SaaS, вимоги до безпеки) вимагають формування стратегії сталого оновлення, підвищення надійності платформи та акценту на унікальні особливості бренду. Саме поєднання індивідуалізованого підходу, автоматизації сервісів і персоналізації обслуговування дозволяє створити конкурентоспроможну платформу, здатну не лише адаптуватися до змін, а й формувати нові стандарти у сфері цифрових послуг для салонів краси.

1.5 Постановка задачі дослідження. Формулювання технічного завдання на розробку у вигляді паспорту проєкту

Назва проєкту: «Розробка проєкту створення вебплатформи для діяльності салону краси».

Мета проєкту – створення вебпорталу для салону краси з надання послуг з догляду за волоссям, який автоматизує процеси бронювання послуг, управління даними клієнтів та працівників, аналіз зворотного зв'язку та проведення маркетингових кампаній для підвищення якості обслуговування і задоволеності клієнтів.

Інноваційність проєкту. Застосування сучасних технологій обробки природної мови (NLP) для автоматичного аналізу клієнтських відгуків і виявлення прихованих тенденцій та потреб. Це дозволить салону оперативно адаптуватися до змінних очікувань клієнтів та підвищити рівень їхньої задоволеності.

Цілі проєкту:

1. Розробити вебплатформу для салону краси «MOV Hair Studio» з функціями онлайн-бронювання, управління клієнтськими даними, аналізу зворотного зв'язку, маркетингових активностей та адміністративного управління.
2. Досягти 90% задоволеності клієнтів через 6 місяців після запуску вебпорталу, вимірюючи її через регулярні опитування та аналіз відгуків.
3. Забезпечити 100% автоматизацію процесу бронювання послуг через вебплатформу протягом перших 3 місяців з використанням сучасних технологій для автоматичного призначення вільного часу та майстра.
4. Підвищити кількість бронювань через вебплатформу на 50% протягом першого року після впровадження, зосереджуючись на зручності використання та маркетингових активностях для залучення нових користувачів.

5. Завершити розробку та запустити вебплатформу протягом 9 місяців з моменту початку проєкту, з детальним планом з ключовими етапами та контрольними точками для відстеження прогресу.

Для кожної із вище зазначених цілей сформуємо перелік задач для їх досягнення. Даний перелік відповідно до цілей наведено в табл. 1.5.

Таблиця 1.5

Формулювання переліку задач відповідно до цілей проєкту

Ціль 1. Розробити вебплатформу для салону краси «MOV Hair Studio» з функціями онлайн-бронювання, управління клієнтськими даними, аналізу зворотного зв'язку, маркетингових активностей та адміністративного управління.
Задача 1.1. Провести збір і аналіз вимог від власників салону та цільових користувачів. Задача 1.2. Сформувати архітектуру системи та структуру бази даних для ключових функцій. Задача 1.3. Розробити інтерфейс онлайн-бронювання з інтерактивним календарем. Задача 1.4. Реалізувати систему управління клієнтськими профілями (реєстрація, історія візитів, переваги). Задача 1.5. Впровадити модуль збору та візуалізації відгуків клієнтів (рейтинг, коментарі). Задача 1.6. Розробити панель адміністратора для керування послугами, майстрами, графіками. Задача 1.7. Інтегрувати маркетингові інструменти (e-mail розсилки, купони, акції). Задача 1.8. Забезпечити адаптивний дизайн для десктопної та мобільної версій. Задача 1.9. Провести інтеграцію з календарями майстрів (Google Calendar або подібне). Задача 1.10. Провести тестування системи на функціональність, зручність та безпеку.
Ціль 2. Досягти 90% задоволеності клієнтів через 6 місяців після запуску вебпорталу, вимірюючи її через регулярні опитування та аналіз відгуків.
Задача 2.1. Розробити та впровадити регулярні онлайн-опитування користувачів (NPS, CSAT). Задача 2.2. Реалізувати форму зворотного зв'язку на платформі. Задача 2.3. Налаштувати аналітику клієнтської взаємодії (Hotjar, Google Analytics). Задача 2.4. Створити систему оперативної обробки скарг та побажань користувачів. Задача 2.5. Аналізувати отримані відгуки та формувати щомісячні звіти. Задача 2.6. Запустити програму лояльності на основі фідбеку. Задача 2.7. Оптимізувати інтерфейс на основі поведінкових даних користувачів. Задача 2.8. Провести навчання персоналу салону з використання платформи для покращення сервісу. Задача 2.9. Визначити ключові показники задоволеності (KPI) та регулярно їх вимірювати. Задача 2.10. Адаптувати функціонал платформи відповідно до відгуків користувачів.

Ціль 3. Забезпечити 100% автоматизацію процесу бронювання послуг через вебплатформу протягом перших 3 місяців з використанням сучасних технологій для автоматичного призначення вільного часу та майстра.
Задача 3.1. Розробити алгоритм автоматичного підбору вільного часу та відповідного майстра. Задача 3.2. Сформуванати розклади майстрів з урахуванням можливих змін. Задача 3.3. Впровадити перевірку конфліктів розкладу та механізм блокування часу. Задача 3.4. Реалізувати push/e-mail повідомлення про бронювання, зміни та нагадування. Задача 3.5. Надати користувачам можливість змінювати/скасовувати запис самостійно. Задача 3.6. Інтегрувати бронювання з CRM або Google Calendar для синхронізації. Задача 3.7. Провести тестування сценаріїв автоматичного бронювання. Задача 3.8. Впровадити дашборд для моніторингу бронювань у реальному часі. Задача 3.9. Навчити персонал роботі з автоматизованою системою. Задача 3.10. Створити резервний сценарій ручного втручання у разі збою.
Ціль 4. Підвищити кількість бронювань через вебплатформу на 50% протягом першого року після впровадження, зосереджуючись на зручності використання та маркетингових активностях для залучення нових користувачів.
Задача 4.1. Провести рекламну кампанію з запуску платформи (соцмережі, Google Ads). Задача 4.2. Реалізувати SEO-оптимізацію сайту для кращої видимості в пошукових системах. Задача 4.3. Запустити реферальну програму для клієнтів. Задача 4.4. Налаштувати систему знижок/бонусів при онлайн-бронюванні. Задача 4.5. Запровадити персоналізовані пропозиції на основі історії бронювань. Задача 4.6. Аналізувати воронку бронювання та оптимізувати її (UX, СТА, швидкість). Задача 4.7. Запустити email-маркетинг з регулярними пропозиціями. Задача 4.8. Впровадити A/B-тестування для маркетингових елементів платформи. Задача 4.9. Проводити регулярні акції/іvents для залучення нових користувачів. Задача 4.10. Відслідковувати динаміку бронювань і коригувати маркетингову стратегію.
Ціль 5. Завершити розробку та запустити вебплатформу протягом 9 місяців з моменту початку проєкту, з детальним планом з ключовими етапами та контрольними точками для відстеження прогресу.
Задача 5.1. Розробити детальний план-графік проєкту з розбивкою на етапи. Задача 5.2. Сформуванати команду проєкту та визначити відповідальних за кожен напрям. Задача 5.3. Виконати етап UI/UX дизайну платформи з прототипуванням. Задача 5.4. Провести поетапну розробку модулів (Agile/Scrum-підхід). Задача 5.5. Встановити систему контролю версій (Git) та CI/CD. Задача 5.6. Забезпечити регулярну перевірку результатів на контрольних точках (місяць/етап). Задача 5.7. Впровадити юніт- та інтеграційне тестування. Задача 5.8. Провести бета-тестування з реальними користувачами. Задача 5.9. Здійснити запуск MVP (мінімально життєздатного продукту). Задача 5.10. Підготувати документацію, навчальні матеріали та технічну підтримку.

Життєвий цикл продукту проєкту для вебпорталу салону краси «MOV Hair Studio» може бути розділений на наступні етапи:

Етап 1. Аналіз та планування.

1.1. Консультації та визначення вимог (спілкування з власниками салону та працівниками для виявлення ключових процесів і потреб; аналіз поведінки

та очікувань клієнтів (анкетування, досвід конкурентів); визначення функціоналу MVP та пріоритетних функцій).

1.2. Створення технічного завдання (формалізація вимог у вигляді структурованого ТЗ з блок-схемами і сценаріями; опис архітектури системи (UI, бізнес-логіка, база даних); визначення ключових показників ефективності (KPI) та контрольних точок).

Етап 2. Дизайн, розробка та тестування.

2.1. UI/UX дизайн (розробка інтерактивних прототипів та макетів; проведення юзабіліті-тестів прототипів з представниками цільової аудиторії; адаптація інтерфейсу під мобільні пристрої).

2.2. Бекенд розробка (побудова API для обробки запитів клієнтів і адміністраторів; розробка алгоритмів автоматичного призначення послуг; забезпечення ролей доступу: клієнт, майстер, адміністратор).

2.3. Фронтенд розробка (реалізація сторінок онлайн-бронювання, профілю користувача та реєстрації; інтеграція відгуків, рейтингів, форм зворотного зв'язку; впровадження маркетингових елементів (банери, СТА, акції)).

2.4. Інтеграція з базою даних (створення моделі даних для зберігання інформації про користувачів, послуги, розклади; забезпечення зв'язності між модулями; впровадження системи резервного копіювання).

Етап 3. Фінальне функціональне тестування.

3.1. Функціональне тестування (перевірка виконання всіх сценаріїв користувача (бронювання, реєстрація, зміна даних); тестування логіки розкладів, знижок, відгуків).

3.2. Тестування користувацького інтерфейсу (оцінка зручності навігації для різних ролей; UX-тестування на мобільних пристроях).

3.3. Тестування безпеки (аудит захищеності персональних даних; тестування автентифікації, ролей, шифрування; виявлення XSS, SQL-ін'єкцій та інших вразливостей).

Етап 4. Запуск та навчання.

4.1. Розгортання системи (встановлення вебпорталу на хостинг із налаштуванням домену, HTTPS; налаштування моніторингу та логування).

4.2. Навчання персоналу (підготовка інструкцій і відео-гайдів; проведення тренінгів для адміністраторів, майстрів; створення служби підтримки першої лінії (FAQ, контакт-форма).

Етап 5. Підтримка та розвиток.

5.1. Технічна підтримка (реагування на звернення користувачів та усунення помилок; ведення журналу інцидентів та вирішення проблем).

5.2. Оновлення та розвиток (впровадження нових функцій (наприклад, інтеграція з Telegram-ботом); розширення маркетингового функціоналу (аналітика, A/B тестування); періодичні оновлення безпеки та сумісності).

Етап 6. Моніторинг та аналіз.

6.1. Моніторинг ефективності: (відстеження KPI: кількість бронювань, задоволеність, повторні звернення; аналітика користувацької поведінки (Google Analytics, Hotjar)).

6.2. Звітність та аналіз (формування щомісячних та квартальних звітів; розробка рекомендацій щодо покращення функціоналу на основі зворотного зв'язку; прийняття стратегічних рішень щодо подальшої автоматизації та масштабування).

Життєвий цикл продукту вебпорталу для салону краси «MOV Hair Studio» охоплює повний спектр робіт – від початкового аналізу та планування до запуску, підтримки та стратегічного розвитку. Така структурована модель забезпечує системний підхід до створення цифрового рішення, що враховує як бізнес-цілі власників салону, так і потреби кінцевих користувачів. Ретельне тестування, навчання персоналу, постійна підтримка та моніторинг ефективності дозволяють не лише гарантувати стабільну роботу вебпорталу, а й адаптувати його до змін ринку, зберігаючи конкурентоспроможність і високий рівень задоволеності клієнтів.

РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1 Розробка концептуальної моделі інформаційної системи

Розробка концептуальної моделі є ключовим етапом у процесі створення будь-якої інформаційної системи. Ця модель визначає загальну архітектуру системи, її складові елементи та взаємозв'язки між ними. У цьому підрозділі ми розглянемо процес розробки концептуальної моделі для системи управління салоном краси «MOV Hair Studio»..

Перед визначенням концептуальної моделі ми проаналізуємо вхідні дані та основні процеси, які визначають функціонал системи. Після цього ми розглянемо ключові аспекти архітектури системи та встановимо зв'язки між її складовими елементами. Крім того, будуть розглянуті можливі обмеження, які впливають на розробку концептуальної моделі, а також важливі вимоги щодо безпеки та захисту даних.

В цьому контексті створення концептуальної моделі стає важливим кроком у напрямку розробки ефективної та надійної інформаційної системи, яка відповідає потребам сучасного салону краси та забезпечує зручний та продуктивний взаємозв'язок між клієнтами, майстрами та адміністрацією.

Отже, для побудови концептуальної моделі розробки проєкту створення вебплатформи для діяльності салону краси «MOV Hair Studio» нам потрібно чітко розуміти, що є вхідними даними (табл. 2.1), що є процесами (табл. 2.2), що є вихідними результатами (табл. 2.3), які можливі обмеження (табл. 2.4) та елементи зворотного зв'язку (табл. 2.5).

Таблиця 2.1

Вхідні дані

Блок моделі	Підблоки моделі	Елементи, які входять до підблоку
1	2	3
Вхідні дані	Дані клієнтів	1. Ім'я, контактні дані (телефон, email). 2. Історія відвідувань та послуг. 3. Персональні вподобання та алергії.

Вхідні дані	Дані послуг	1. Перелік послуг, доступних у салоні. 2. Ціни та тривалість послуг. 3. Доступні майстри та їх розклад.
	Дані працівників	1. Ім'я, контактні дані. 2. Графік роботи. 3. Спеціалізація та досвід.
	Зовнішні дані	1. Відгуки з соціальних мереж та інших платформ. 2. Дані про акції та знижки від постачальників.

Таблиця 2.2

Дані щодо процесів

Блок моделі	Модулі які потребують розробки	Функціонал модулів
Процеси	Автоматизація запису на прийом	1. Клієнти можуть бронювати послуги онлайн через вебплатформу. 2. Система автоматично призначає вільний час та майстра на основі доступності та вподобань клієнта.
	Управління даними клієнтів та послуг	1. Збереження та оновлення історії відвідувань. 2. Надання персоналізованих рекомендацій на основі минулих записів та вподобань. 3. Автоматичні нагадування про майбутні прийоми через SMS або email.
	Збір та аналіз зворотного зв'язку (форум)	1. Збір відгуків після кожного візиту. 2. Аналіз відгуків з використанням алгоритмів обробки природної мови (NLP) для виявлення ключових тенденцій та проблем.
	Маркетингові активності та програми лояльності	1. Запуск таргетованих рекламних кампаній на основі даних клієнтів. 2. Управління програмами лояльності та спеціальними пропозиціями для постійних клієнтів.
	Адміністративне управління вебплатформи	1. Управління розкладом майстрів. 2. Ведення обліку фінансових показників (дохід, витрати). 3. Генерація звітів для аналізу ефективності роботи салону.

Таблиця 2.3

Інформація щодо вихідних даних

Блок моделі	Підблоки моделі	Елементи, які входять до підблоку
Вихідні дані	Результати бронювань	1. Підтверджені записи на прийом. 2. Автоматизовані нагадування для клієнтів.
	Аналітичні звіти	1. Звіти про задоволеність клієнтів та якість обслуговування. 2. Статистичні дані про використання послуг та фінансові показники.
	Маркетингові результати	1. Дані про ефективність рекламних кампаній. 2. Інформація про успішність програм лояльності.

Дані щодо зворотного зв'язку

Блок моделі	Підблоки моделі	Елементи, які входять до підблоку
Зворотний зв'язок	Клієнтські відгуки	1. Інтерактивні опитування після кожного візиту. 2. Аналіз коментарів та пропозицій для покращення сервісу.
	Дані про використання вебплатформи	1. Відстеження активності користувачів на вебплатформі. 2. Аналіз проблемних зон та оптимізація інтерфейсу.
	Аналіз ефективності персоналу	1. Збір даних про продуктивність майстрів. 2. Оцінка роботи персоналу на основі відгуків клієнтів та внутрішніх показників.

Параметри можливих обмежень

Блок моделі	Підблоки моделі	Елементи, які входять до підблоку
Можливі обмеження	Технічні обмеження	1. Відсутність належної інфраструктури для підтримки великої кількості одночасних користувачів. 2. Проблеми з інтеграцією з існуючими системами салону.
	Фінансові обмеження	1. Високі витрати на розробку та підтримку платформи. 2. Обмежений бюджет на маркетингові активності.
	Організаційні обмеження	1. Опір персоналу до впровадження нових технологій. 2. Необхідність навчання персоналу для роботи з новою системою.
	Юридичні обмеження	1. Забезпечення відповідності нормативним вимогам щодо зберігання та обробки персональних даних. 2. Виконання вимог щодо захисту прав споживачів.

Підходи до концептуального моделювання інформаційних систем, а також їх математичного опису розглядалися авторами наукових робіт [32-37].

Таким чином маючи всю сукупність необхідних даних щодо елементів які можуть складати інформаційну систему (у випадку дослідження вебплатформу для сфери надання послуг), а також проаналізувавши існуючі літературні та наукові джерела щодо створення таких моделей, стала можливою побудова концептуальної моделі розробки та функціонування досліджуваної вебплатформи. Розроблену модель представлено на рис. 2.1.

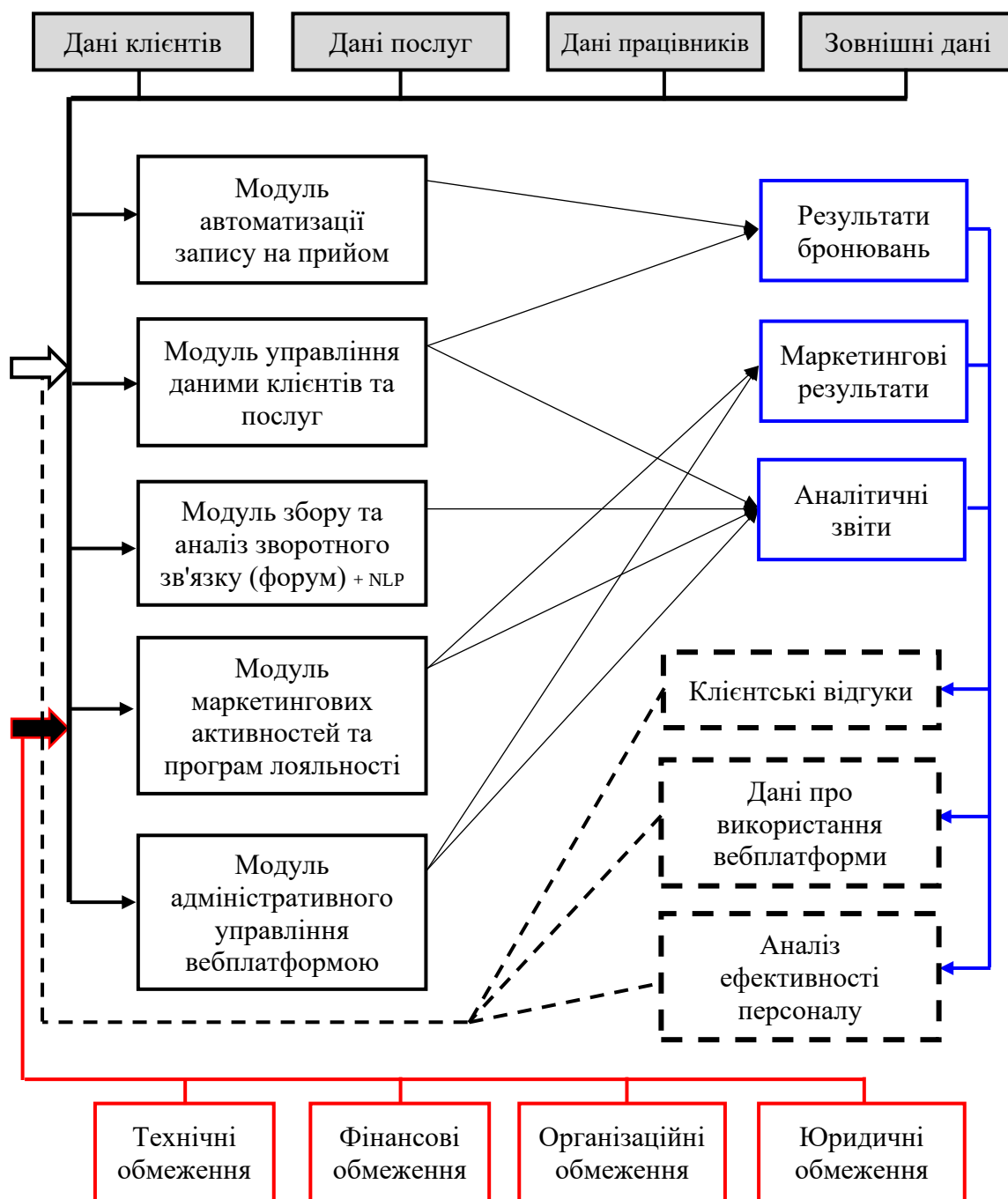


Рис. 2.1. Концептуальна модель розробки та функціонування даної вебплатформи

Розроблена та представлена на рис. 2.1 концептуальна модель стане основою для подальшого дослідження.

2.2 Постановка задачі в математичному вигляді відповідно до концептуальної моделі

В першу чергу здійснимо опис необхідних параметрів і змінних для побудови математичної моделі відповідно до дослідження, яке необхідно провести.

1. Дані клієнтів:

- C_i – клієнт, де i - унікальний індекс клієнта;
- D_i – набір даних клієнта C_i , включаючи ім'я, контактні дані, історію відвідувань, персональні вподобання та інше (d_i).

Параметр D_i можна представити як сукупність множин:

$$D_i = \{d_{i1}, d_{i2}, d_{i3}, \dots, d_{in}\} \quad (2.1)$$

2. Дані послуг:

- S_j – послуга, що надається, де j – її унікальний індекс;
- P_j – інформація щодо послуги S_j . Інформація включає дані (p_i) щодо ціни послуги; майстрів, що можуть надати дану послугу, тривалість сеансу та інші дані.

Параметр P_j можна представити, як вираз сукупності множин:

$$P_j = \{p_{j1}, p_{j2}, p_{j3}, \dots, p_{jn}\} \quad (2.2)$$

3. Дані працівників:

- E_k – працівник салону, де k – його унікальний індекс;
- W_k – сукупність даних щодо характеризують працівника E_k , а саме: його ім'я, контактні дані, графік роботи, приклад виконаних замовлень клієнта, спеціальні вміння та інше (w_i).

Параметр W_k можна виразити у сукупності множин, як:

$$W_k = \{w_{k1}, w_{k2}, w_{k3}, \dots, w_{kn}\} \quad (2.3)$$

4. Зовнішні дані:

- R_m – відгуки, де m - індекс відгуку.
- A_n – акції та знижки, де n - індекс акції.

Відповідно наступним кроком є побудова математичних моделей, які будуть відповідати за формування модулів, які складають основні процеси функціональної складової концептуальної моделі інформаційної системи.

1. Автоматизація запису на прийом.

Нехай B_{ij} – бронювання клієнта C_i на послугу S_j . Тоді цільова функція автоматизації запису:

$$Avz = \min \sum_{i=1}^N \sum_{j=1}^M |t_{ij} - t_{optimal}| \quad (2.4)$$

де t_{ij} – час запису клієнта C_i на послугу S_j , $t_{optimal}$ – оптимальний час на основі доступності майстрів та вподобань клієнта.

2. Управління даними клієнтів та послуг.

Оновлення історії відвідувань можна визначити як:

$$H_i = H_i \cup \{B_{ij}\} \quad (2.5)$$

де H_i – історія відвідувань клієнта C_i .

Персоналізовані рекомендації можна визначити як:

$$R_{ij} = f(D_i, H_i, P_j) \quad (2.6)$$

де R_{ij} – рекомендація для клієнта C_i щодо послуги S_j на основі його даних і історії відвідувань.

3. Збір та аналіз зворотного зв'язку.

Відгуки після кожного візиту можна визначити як:

$$F_i = \sum_{m=1}^M r_{im} \quad (2.7)$$

де r_{im} – відгук клієнта C_i .

Один із основних наукових аспектів даного дослідження це саме застосування NLP. І саме із застосування NLP буде здійснюватися аналіз впровадження сучасних методів обробки природної мови для автоматичного аналізу клієнтських відгуків та виявлення ключових тенденцій і проблем. Даний процес можна представити як сукупність множин (t_i):

$$NLP_T = \{t_1, t_2, t_3, \dots, t_q\} \quad (2.8)$$

де T – набір тем, виявлених в відгуках, з використанням методів обробки природної мови.

4. Маркетингові активності та програми лояльності.

Таргетовані рекламні кампанії можна визначити як:

$$M_i = f(D_i, H_i, P) \quad (2.9)$$

де M_i – маркетингова кампанія для клієнта C_i .

Програми лояльності можна визначити як:

$$L_i = h(H_i) \quad (2.10)$$

де L_i – програма лояльності для клієнта C_i на основі його історії відвідувань.

5. Адміністративне управління.

Управління розкладом майстрів можна визначити як сукупність множин (s_i):

$$S_k = \{s_{k1}, s_{k2}, s_{k3}, \dots, s_{kp}\} \quad (2.10)$$

де S_k – розклад майстра E_k .

Процес генерації звітів можна визначити як:

$$FR = \sum_{i=1}^N \sum_{j=1}^M B_{ij} \quad (2.11)$$

де FR – загальний звіт про записи на прийом.

Вище наведене математичне моделювання постановки задачі даного дослідження деталізує вхідні дані, процеси, вихідні результати, зворотний зв'язок та можливі обмеження, що сприятиме забезпеченню ефективному впровадженню та функціонуванню вебплатформи для салону краси «MOV Hair Studio».

2.3 Використання методів моделювання розроблених моделей в аспекті діяльності салону краси «MOV Hair Studio»

З огляду на важливість виявлення прихованих тенденцій та потреб клієнтів, необхідно перейти до моделювання запропонованого процесу.

Для математизації процесу застосування методів обробки природної мови (NLP) для аналізу клієнтських відгуків у вебплатформі салону краси «MOV Hair Studio» можна використати кілька етапів, кожен з яких описується відповідними математичними моделями і формулами.

Даний процес моделювання буде здійснено в декілька етапів.

Етап 1. Збір даних відгуків.

$$T = \{t_1, t_2, t_3, \dots, t_i\} \quad (2.12)$$

де t_i – текст відгуку i -го клієнта.

Етап 2. Попередня обробка даних.

1) Очищення тексту. Функція очищення тексту матиме наступний вигляд:

$$f_{clean} = (t_i) \rightarrow t'_i \quad (2.13)$$

2) Токенізація. Функція токенизації тексту матиме наступний вигляд:

$$f_{tokeniz} = t'_i \rightarrow \{v_1, v_2, v_3, \dots, v_m\}, \quad (2.15)$$

де v_m – j -ий токен.

3) Видалення стоп-слів:

$$Del = \{del_1, del_2, del_3, \dots, del_i\} \quad (2.15)$$

де del_i – стоп-слово.

$$\{v_1, v_2, v_3, \dots, v_m\} \setminus D \rightarrow \{v'_1, v'_2, v'_3, \dots, v'_m\} \quad (2.16)$$

4) Лематизація / стемінг. Функція лематизації матиме наступний вигляд:

$$f_{lemma/stem}(v'_i) \rightarrow l_j \quad (2.17)$$

Етап 3. Аналіз тексту.

1) Векторизація тексту може відбуватися згідно TF-ID.

TF-IDF (Term Frequency-Inverse Document Frequency) – це статистична міра, яка використовується для оцінки важливості слова в документі відносно до колекції документів. Вона складається з двох компонентів: TF (частота терміну) та IDF (зворотна частота документів).

Для нашого дослідження формула для обчислення TF-IDF терміну (t) у документі (d) матиме наступний вигляд:

$$TF - IDF(t, d) = tf(d, t) \cdot idf(t), \quad (2.18)$$

де, $tf(d, t)$ – це частота терміну t у документі d , яка показує, скільки разів термін з'являється у документі; $idf(t)$ – це зворотна частота документів для терміну t , яка зменшує вагу часто вживаних термінів у всіх документах і збільшує вагу не часто вживаних термінів.

В іншому вигляді даний вираз може мати вигляд:

$$tf(d, t) = \frac{\text{Кількість появ терміну } t \text{ у документі } d}{\text{Загальна кількість термінів у документі } d} \quad (2.19a)$$

$$idf(t) = \log\left(\frac{N}{1 + \text{Кількість документів } (D), \text{ що містять } t}\right) \quad (2.19a)$$

де, N – це загальна кількість документів у колекції.

За допомогою TF-IDF можна визначити найважливіші терміни у відгуках клієнтів. Наприклад, якщо в багатьох відгуках часто зустрічаються слова «стрижка», «перукар» та «обслуговування», але в конкретному відгуку слово «погано» зустрічається рідко, його вагу буде високою. Це дозволить виділити відгуки з негативним змістом для подальшого аналізу.

Також, аналізуючи відгуки за допомогою TF-IDF, можна зрозуміти, які послуги найбільше цінуються клієнтами. Наприклад, якщо клієнт часто згадує терміни «фарбування» і «стрижка» у своїх відгуках, вебплатформа може пропонувати акції або нові послуги, пов'язані з цими процедурами.

2) Виявлення тем. Тематичне моделювання можна зобразити у наступному вигляді:

$$I_{tem} = \max \prod_{d \in D} \prod_{v \in d} \sum_z P(v|z) P(z|d) \quad (2.20)$$

Етап 4. Класифікація та категоризація відгуків.

Елемент класифікації може бути врахований використовуючи технології машинного навчання та може мати наступний математичний вигляд:

$$Q_{klas}(t_i; \theta) \rightarrow c_i, \quad (2.21)$$

де, Q_{klas} – це модель класифікації; θ – параметри; c_i – категорія відгуку (позитивний, негативний, нейтральний).

Етап 5. Врахування настрою клієнтів.

Модель врахування настроїв (G) може мати наступний вигляд:

$$G(t_i) \rightarrow \sigma_i, \quad (2.22)$$

де, σ_i – оцінка настрою (позитивний, негативний, нейтральний).

Етап 6. Інтеграція результатів у систему зворотного зв'язку.

1) Автоматичне генерування звітів.

2) Надавання рекомендацій.

Розглянемо декілька прикладів розроблених інструментів застосування NLP в аспекті функціонування вебплатформі салону краси «MOV Hair Studio».

1. Персоналізовані рекомендації на основі аналізу відгуків.

Опис ситуації: Клієнти залишають відгуки про надані послуги, зазначаючи, що їм сподобалось, а що ні. Використовуючи NLP, система

аналізує ці відгуки та надає персоналізовані рекомендації для кожного клієнта при наступному візиті.

Відгук клієнта: «Я дуже задоволена стрижкою від Марії, але колір волосся від Ірини виявився занадто темним для мене».

Аналіз NLP: Виявлення позитивної оцінки стрижки та негативної оцінки кольору волосся.

Рекомендація: Наступного разу рекомендувати клієнту стрижку від Марії та запропонувати обговорити бажаний колір з іншим спеціалістом.

2. Автоматичні відповіді на відгуки.

Опис: Система автоматично генерує відповіді на відгуки клієнтів, використовуючи алгоритми NLP, що дозволяє швидко реагувати на запити та підвищує задоволеність клієнтів.

Відгук клієнта: «Мені дуже сподобалось обслуговування, але я чекала на свій прийом 15 хвилин».

Автоматична відповідь: «Дякуємо за ваш відгук! Ми раді, що вам сподобалось обслуговування. Вибачте за очікування, ми докладемо зусиль, щоб таке більше не повторилось».

Отже, застосування методів NLP для функціонування вебплатформи «MOV Hair Studio» дозволить не тільки автоматизувати процес аналізу відгуків клієнтів, але й надавати цінну інформацію для покращення обслуговування, персоналізованих рекомендацій, оцінки роботи персоналу, виявлення ключових проблем та розробки ефективних маркетингових стратегій. Це забезпечує підвищення якості обслуговування та задоволеності клієнтів, що є важливим аспектом наукової новизни даного дослідження.

РОЗДІЛ 3. ВИБІР ТЕХНОЛОГІЇ УПРАВЛІННЯ ПРОЄКТОМ ТА РОЗРОБКА ЙОГО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Вибір технології управління проєктом

Успішність реалізації програмного проєкту значною мірою залежить від обраного підходу до його управління. Особливо це актуально в умовах обмежених ресурсів, стиснутих термінів і необхідності гнучкого реагування на зміну вимог. Методологія управління проєктом визначає організацію робочого процесу, порядок виконання завдань, взаємодію між учасниками та контроль за результатами [38]. У контексті розробки вебплатформи для салону краси ключовими вимогами до управління проєктом є:

- адаптивність до змін у бізнес-процесах,
- швидке постачання робочого функціоналу,
- можливість залучення замовника до кожного етапу розробки,
- прозорість виконання завдань і контролю прогресу.

З огляду на ці фактори, було прийнято рішення використовувати гнучку методологію Scrum, яка найкраще відповідає вищезазначеним критеріям.

Scrum – це фреймворк для гнучкого управління проєктами, орієнтований на інкрементальну та ітеративну розробку [39-41]. Його ключові компоненти:

1. Ролі:

- Product Owner – визначає функціональні пріоритети, формує беклог;
- Scrum Master – забезпечує дотримання принципів Scrum, усуває перешкоди;
- Scrum Team – кросфункціональна команда, що самостійно організовується.

2. Артефакти:

- Product Backlog – перелік вимог і функцій;
- Sprint Backlog – завдання на поточну ітерацію;
- Increment – готовий до використання функціонал.

3. Події (церемонії) (характеристика подій наведена в табл. 3.1):

- Sprint Planning;
- Daily Scrum;
- Sprint Review;
- Sprint Retrospective.

Таблиця 3.1

Характеристика подій Scrum

№	Подія Scrum	Мета	Учасники	Вхідні/формат роботи	Результат	Приклад у досліджуваному проєкті
1	Sprint planning	Визначити, що буде реалізовано у спринті, і як цього досягти	Вся Scrum-команда: Product Owner, Scrum Master, розробники	- оновлений Product Backlog; - мета продукту; - результати попередніх спринтів.	- Sprint backlog. - Sprint goal.	Планування задач: онлайн-бронювання, реєстрація клієнтів, форма зворотного зв'язку
2	Daily scrum	Синхронізувати дії команди, виявити перешкоди	Scrum Team (розробники), Scrum Master (фасилітатор)	Кожен відповідає на 3 питання: 1. Що зробив учора? 2. Що робитиму сьогодні? 3. Які є перешкоди?	- Спільне розуміння статусу роботи. - Оперативне реагування на затримки.	Наприклад, оновлення щодо верстки інтерфейсу запису або створення API для бронювання
3	Sprint review	Продемонструвати готовий інкремент і зібрати зворотний зв'язок	Scrum Team + замовник (власник/адміністратор салону)	- демонстрація готового функціоналу; - обговорення; досягнень і відхилень - збір нових ідей.	- Зворотний зв'язок від замовника. - Оновлений Product backlog.	Демонстрація: реалізована сторінка послуг, форма запису, реєстрація клієнтів
4	Sprint retrospective	Аналіз ефективності процесу роботи, пошук шляхів покращення	Scrum Team	Обговорення: 1. Що працювало добре? 2. Що не працювало? 3. Що покращити у наступному спринті?	- Список ідей для вдосконалення. - План покращень процесу в наступному спринті.	Вирішення змінити порядок ревізії, покращити розподіл задач, уточнити опис критеріїв готовності

Розглянемо переваги Scrum у контексті проєкту, що розглядається.

1. *Гнучкість до змін.* У б'юті-індустрії часто змінюються бізнес-потреби: введення акцій, змін у ціноутворенні, нових послуг тощо. Scrum дозволяє змінювати пріоритети в кожному спринті, не порушуючи загальної логіки розробки.

2. *Швидке отримання результату.* Кожен спринт триває 2 тижні та завершується робочим інкрементом. Це дозволяє замовнику (власнику салону) бачити поступове зростання функціоналу платформи, надавати зворотний зв'язок і вносити корективи.

3. *Постійна залученість замовника.* Product Owner бере участь у плануванні, демонстраціях результатів і ретроспективах. Це сприяє точнішому визначенню очікувань і мінімізує ризики непорозумінь.

4. *Підвищення якості продукту.* Завдяки ретроспективам, регулярному тестуванню та фокусі на завершеному інкременті Scrum створює умови для постійного вдосконалення процесу і кінцевого продукту.

5. *Мотивація команди.* Команда сама обирає обсяг роботи, бере участь у плануванні, і завдяки автономності відчуває більше відповідальності за результат. Це покращує моральний клімат і продуктивність.

6. *Зменшення ризиків.* Раннє виявлення проблем на щоденних зустрічах і демонстраціях знижує ймовірність критичних помилок на пізніх етапах.

Scrum – це не просто методологія або інструмент, а ефективна система управління змінами, що дозволяє адаптувати процес розробки до динамічного середовища [42]. Його впровадження в проєкті створення вебплатформи для салону краси є обґрунтованим рішенням, що сприяє:

- поступовій реалізації функціоналу відповідно до пріоритетів замовника;
- оперативному внесенню змін у проєкт;
- мінімізації ризиків;
- покращенню якості продукту;
- ефективній командній роботі.

Отже, використання Scrum забезпечить гнучке, контрольоване та результативне управління реалізацією даного програмного проєкту.

3.2 Визначення функціональних та нефункціональних вимог до продукту проєкту

Ефективна реалізація вебплатформи для салону краси «MOV Hair Studio» потребує чіткого визначення як функціональних, так і нефункціональних вимог до майбутнього програмного продукту. На цьому етапі проєкту відбувається формалізація очікувань кінцевих користувачів та власників бізнесу у вигляді конкретних характеристик системи, що мають бути реалізовані в межах розробки.

Функціональні вимоги описують конкретні дії та поведінку системи, зокрема можливості, які вона повинна надавати користувачам. До таких вимог відносяться, зокрема: реєстрація та авторизація клієнтів, онлайн-бронювання послуг, управління персональним кабінетом, збір і обробка відгуків, генерація звітів, а також інструменти маркетингової взаємодії. Визначення цих вимог дозволяє деталізувати очікуваний функціонал, що забезпечить повноцінну цифрову підтримку операцій салону.

Нефункціональні вимоги, у свою чергу, стосуються якісних характеристик платформи, що не пов'язані безпосередньо з функціями, але є критичними для задоволення користувацького досвіду, безпеки, продуктивності та надійності. Сюди входить забезпечення високої швидкодії системи, відповідність вимогам безпеки, доступність для користувачів з особливими потребами, масштабованість, зручність інтерфейсу, стабільність роботи тощо.

У даному підрозділі представлено систематизований перелік як функціональних, так і нефункціональних вимог до вебплатформи, сформований на основі аналізу бізнес-потреб, цілей проєкту, технічних можливостей та очікувань кінцевих користувачів. Визначення цих вимог

слугуватиме основою для подальшого проєктування, розробки, тестування та впровадження вебпорталу.

В табл. 3.2 наведено короткий перелік функціональних вимог визначених для даного проєкту який досліджується.

Таблиця 3.2

Функціональні вимоги проєкту

Код ФВ	Назва ФВ	Характеристика функціональної вимоги
1	2	3
ФВ01	Реєстрація користувачів	Система повинна надавати можливість реєстрації нових користувачів з подальшим створенням особистого кабінету, де зберігається історія бронювань, вподобання та профільні дані.
ФВ02	Авторизація користувачів	Користувачі повинні мати можливість входу до свого акаунта за допомогою email та паролю або через соціальні мережі.
ФВ03	Бронювання послуг	Користувачі повинні мати змогу онлайн-бронювання послуг (стрижка, фарбування тощо) з вибором дати, часу, майстра та типу послуги.
ФВ04	Автоматичний підбір вільного часу	Система має автоматично пропонувати вільні часові слоти відповідно до графіків майстрів та обраної послуги.
ФВ05	Персоналізовані рекомендації	Платформа повинна аналізувати історію візитів і вподобання клієнтів для надання індивідуальних рекомендацій щодо послуг або продуктів.
ФВ06	Нагадування про прийоми	Клієнти повинні отримувати автоматичні нагадування про майбутні прийоми через SMS та/або Email з можливістю налаштування часу сповіщення.
ФВ07	Збір відгуків після візиту	Після завершення послуги система повинна надсилати запит на оцінку якості обслуговування та збирати відгуки, які зберігаються в базі даних.
ФВ08	Аналіз зворотного зв'язку	Адміністрація повинна мати доступ до інструментів аналізу відгуків клієнтів (загальний рейтинг, проблемні точки, тренди).
ФВ09	Управління маркетингом	Система має підтримувати функції створення рекламних кампаній, купонів, акцій, e-mail розсилок та програм лояльності.
ФВ10	Управління розкладом майстрів	Майстри повинні мати змогу самостійно оновлювати свою доступність, переглядати власні бронювання та залишати нотатки.
ФВ11	Панель адміністратора	Адміністратори повинні мати доступ до керування працівниками, списком послуг, цінами, розкладами, акціями, користувачами.
ФВ12	Генерація звітів	Платформа повинна надавати можливість формувати звіти щодо записів, популярності послуг, завантаження майстрів, доходів.

1	2	3
ФВ13	Адаптивний інтерфейс	Вебплатформа повинна коректно працювати на різних пристроях (десктопи, планшети, смартфони) з адаптованим UI/UX.
ФВ14	Безпека та захист даних	Система повинна забезпечувати безпечне зберігання персональних даних користувачів та відповідати сучасним стандартам (SSL, хешування паролів, контроль доступу).
ФВ15	Журнал подій	Система повинна фіксувати ключові події (бронювання, відміни, оновлення профілю тощо) для моніторингу та аудиту.

Перелік нефункціональних вимог наведено в табл. 3.3.

Таблиця 3.3

Нефункціональні вимоги проєкту

Код НФВ	Назва НФВ	Характеристика нефункціональної вимоги
1	2	3
НФВ01	Безпека даних	Система повинна забезпечувати високий рівень захисту персональних та платіжних даних користувачів відповідно до стандартів GDPR/ЗУ «Про захист персональних даних», включаючи використання SSL, хешування паролів, контроль доступу та регулярні аудити безпеки.
НФВ02	Продуктивність	Платформа повинна забезпечувати завантаження основних сторінок не більше ніж за 2 секунди при середньому навантаженні, а обробку запитів бронювання — не більше ніж за 1 секунду.
НФВ03	Доступність	Інтерфейс вебпорталу повинен бути доступним для користувачів з інвалідністю відповідно до стандарту WCAG 2.1 (мінімум рівень AA).
НФВ04	Надійність	Платформа має забезпечувати стабільну роботу з показником доступності (uptime) не менше 99,5% на місяць, включаючи механізми резервного копіювання та аварійного відновлення.
НФВ05	Простота інтерфейсу	Інтерфейс повинен бути інтуїтивно зрозумілим для користувачів без спеціальної підготовки; час освоєння базових функцій не повинен перевищувати 5 хвилин для нового користувача.
НФВ06	Масштабованість	Архітектура платформи повинна дозволяти горизонтальне та вертикальне масштабування для підтримки зростання кількості користувачів, даних і функцій без зниження продуктивності.

1	2	3
НФВ07	Сумісність	Платформа має підтримувати стабільну роботу у популярних браузерів (Chrome, Safari, Firefox, Edge) та бути адаптованою під десктопи, планшети та смартфони.
НФВ08	Легкість підтримки та розвитку	Кодова база має бути добре документованою, а архітектура — модульною, що дозволить легко впроваджувати нові функції, оновлювати компоненти та усувати помилки без ризику для основного функціоналу.
НФВ09	Час реакції системи	Ключові дії користувачів (авторизація, бронювання, перегляд розкладу) повинні виконуватись без помітних затримок — у межах 1–2 секунд для стандартного сценарію використання.
НФВ10	Аудит та логування	Система повинна зберігати журнали ключових подій (авторизація, створення/зміна бронювання, адміністрування) з можливістю їх подальшого аналізу та перегляду адміністратором у разі потреби.

Визначення функціональних та нефункціональних вимог є ключовим етапом у розробці вебплатформи, що забезпечує структуроване бачення майбутнього продукту та знижує ризики невідповідності очікуванням користувачів і бізнес-цілям. Сформульовані функціональні вимоги відображають перелік основних можливостей системи, що забезпечують реалізацію основних сервісів для клієнтів та адміністрації салону – від бронювання послуг до аналітики та маркетингу. У свою чергу, нефункціональні вимоги визначають якісні характеристики платформи, які гарантують її безперебійну, безпечну, зручну й масштабовану роботу.

Отриманий набір вимог стане фундаментом для подальшого проєктування архітектури, побудови інтерфейсів, розробки програмного коду та проведення тестування. Крім того, чітко окреслені вимоги дозволяють ефективно планувати ресурсне забезпечення, строки реалізації та контрольні точки проєкту. Таким чином, цей підрозділ формує методологічну та технічну основу для створення високоякісного та конкурентоспроможного цифрового рішення для сфери краси.

3.3 Формування беклогу продукту проєкту

Формування беклогу продукту є одним із визначальних етапів у процесі розробки вебплатформи для діяльності салону краси «MOV Hair Studio». Беклог продукту (Product Backlog) у контексті нашого проєкту виступає впорядкованим переліком функціональностей, сервісів та покращень, які мають бути реалізовані для забезпечення ефективного функціонування платформи, задоволення потреб клієнтів салону та підтримки бізнес-процесів адміністрації.

Ключовою передумовою для побудови якісного та структурованого беклогу є формулювання User Story – коротких описів функціональностей з точки зору кінцевого користувача. Саме вони дозволяють врахувати реальні очікування та цілі таких категорій користувачів, як адміністратори салону, стилісти та клієнти, що здійснюють онлайн-бронювання послуг. Завдяки User Story кожен елемент беклогу отримує чіткий фокус на користь, яку він приносить, що, у свою чергу, забезпечує пріоритезацію та ефективність подальших розробок.

Проведемо формулювання User Story (US) для проєкту, а також визначимо критерії їх досягнення (Acceptance Criteria). Фрагмент переліку User Story наведено в табл. 3.4. Повний перелік User Story наведено в табл. А.1 Додатку А кваліфікаційної роботи.

Таблиця 3.4

Фрагмент переліку User Story проєкту

Код US	Назва та формулювання US	Acceptance Criteria
1	2	3
US01 (ФВ01)	Реєстрація користувачів. Як новий відвідувач, я хочу мати можливість зареєструватися на платформі, щоб створити обліковий запис і зберігати особисті дані, історію візитів і вподобання.	Scenario: Успішна реєстрація. Given: Я на сторінці реєстрації. When: Заповнюю поля та натискаю «Зареєструватися». Then: Створюється обліковий запис, відкривається особистий кабінет.

US02 (ФВ02)	<i>Авторизація користувачів.</i> Як зареєстрований користувач, я хочу входити до свого акаунта за допомогою email, пароля або соцмереж, щоб мати доступ до персональних функцій.	<i>Scenario:</i> Вхід у систему. <i>Given:</i> Я маю акаунт. <i>When:</i> Ввожу email/пароль або авторизуюсь через соцмережу. <i>Then:</i> Потрапляю до свого кабінету.
US03 (ФВ03)	<i>Онлайн-бронювання послуг.</i> Як клієнт, я хочу мати можливість бронювати послуги онлайн, щоб зручно планувати візити до салону.	<i>Scenario:</i> Бронювання послуги. <i>Given:</i> Я авторизований. <i>When:</i> Вибираю послугу, дату, майстра. <i>Then:</i> Отримую підтвердження бронювання.
...	...	...
US07 (ФВ07)	<i>Збір відгуків після візиту.</i> Як клієнт, я хочу залишати відгук після отриманої послуги, щоб поділитися своїм досвідом.	<i>Scenario:</i> Надсилання відгуку. <i>Given:</i> Я завершив візит. <i>When:</i> Отримую запит на оцінку. <i>Then:</i> Маю можливість залишити відгук через форму.
US08 (ФВ08)	<i>Аналіз зворотного зв'язку.</i> Як адміністратор, я хочу бачити аналітику за відгуками клієнтів, щоб виявляти проблемні зони та покращувати якість сервісу.	<i>Scenario:</i> Перегляд аналітики. <i>Given:</i> Відгуки вже залишені. <i>When:</i> Я відкриваю розділ зворотного зв'язку. <i>Then:</i> Бачу зведення з оцінками, трендами і коментарями.
...	...	...
US10 (ФВ10)	<i>Управління розкладом майстрів.</i> Як майстер, я хочу самостійно оновлювати свій графік, щоб контролювати свій робочий час та запис клієнтів.	<i>Scenario:</i> Редагування розкладу. <i>Given:</i> Я авторизований майстер. <i>When:</i> Відкриваю календар і вношу зміни. <i>Then:</i> Новий розклад зберігається і синхронізується з системою бронювання.
...	...	...
US12 (ФВ12)	<i>Генерація звітів.</i> Як адміністратор, я хочу генерувати звіти щодо записів, популярності послуг та доходів, щоб контролювати ефективність роботи салону.	<i>Scenario:</i> Формування звіту. <i>Given:</i> Є дані про записи та послуги. <i>When:</i> Я запускаю генерацію звіту. <i>Then:</i> Отримую PDF або таблицю з потрібною статистикою.
...	...	...
US15 (ФВ15)	<i>Журнал подій.</i> Як адміністратор, я хочу бачити журнал подій (бронювання, скасування, зміни профілю), щоб мати змогу контролювати дії користувачів.	<i>Scenario:</i> Перегляд журналу. <i>Given:</i> Я відкриваю розділ журналу. <i>When:</i> Система показує список подій. <i>Then:</i> Я бачу хронологію дій користувачів у системі.

Сформульовані користувацькі історії (User Story) відіграють ключову роль у побудові беклогу продукту, адже саме вони є вихідною точкою для планування та реалізації функціональностей платформи. Кожна історія відображає конкретну потребу користувача – клієнта, адміністратора або

майстра салону і транслює їх у вимогу до майбутньої системи. Такий підхід дозволяє не лише враховувати реальні сценарії використання, а й зробити розробку більш гнучкою, цілеспрямованою та орієнтованою на результат.

У контексті Scrum методології, User Story є основними елементами беклогу продукту проєкту, впорядкованого списку функціональних одиниць, які команда має реалізувати. Вони допомагають Product Ownery встановлювати пріоритети, а Scrum-команді краще зрозуміти очікування користувачів і визначати обсяг роботи в межах кожного спринту. Завдяки чітко визначеним Acceptance Criteria, історії стають основою для перевірки досягнення цілей, визначених у кожному етапі розробки.

Таким чином, перехід від формулювання User Story до формування беклогу є не просто механічним кроком, а стратегічним інструментом структурування вимог, що забезпечує прозорість, передбачуваність і контрольованість процесу створення вебплатформи для салону краси «MOV Hair Studio». Наступним логічним етапом стає побудова та пріоритезація беклогу продукту на основі вже сформульованих історій. Фрагмент сформованого беклогу продукту проєкту наведено в табл. 3.5. Повний беклог продукту проєкту наведено в табл. А.2 Додатку А кваліфікаційної роботи.

Таблиця 3.5

Фрагмент беклогу продукту проєкту

User Story	Задача / Task	Підзадача/ Subtask
1	2	3
US01	Т.01.1 Створити функціонал реєстрації користувачів з особистим кабінетом.	ST.01.11 Розробити інтерфейс реєстрації.
		ST.01.12 Зберігати користувача в базі даних.
		ST.01.13 Створювати особистий кабінет при реєстрації.
	Т.01.2 Забезпечити збереження даних із валідацією та хешуванням паролів.	ST.01.21 Додати валідацію email і пароля.
		ST.01.22 Застосувати хешування паролів.
		ST.01.23 Обробляти помилки при збереженні.
US02	Т.02.1 Реалізувати вхід через email і пароль.	ST.02.11 Створити форму входу.
		ST.02.12 Перевірити введені дані.
		ST.02.11 Створити сесію при успішному вході.

1	2	3
	T.02.2 Інтегрувати входи через Google та Facebook.	ST.02.21 Налаштувати авторизацію через Google.
		ST.02.22 Налаштувати авторизацію через Facebook.
		ST.02.23 Прив'язати соцмережу до профілю користувача.
US03	T.03.1 Розробити інтерфейс бронювання послуг.	ST.03.11 Створити форму вибору послуги.
		ST.03.12 Інтегрувати календар та вибір майстра.
		ST.03.13 Додати перевірку доступного часу.
	T.03.2 Реалізувати збереження бронювань і підтвердження користувачу.	ST.03.22 Зберігати бронювання в базі.
		ST.03.22 Надсилати підтвердження користувачу.
US04	T.04.1 Імплементувати автоматичний підбір вільного часу.	ST.03.22 Відображати бронювання у кабінеті.
		ST.04.11 Зібрати графік роботи майстрів.
		ST.04.12 Розрахувати тривалість послуг.
	T.04.2 Виводити список доступних слотів в UI.	ST.04.13 Сформулювати алгоритм підбору слотів.
		ST.04.21 Розробити UI-компонент для показу слотів.
		ST.04.22 Інтегрувати з даними про наявність.
		ST.04.23 Оновлювати доступні слоти в реальному часі.
US05	T.05.1 Розробити алгоритм персоналізованих рекомендацій.	ST.05.11 Аналізувати історію візитів клієнта.
		ST.05.12 Формувати релевантні послуги.
		ST.05.13 Оптимізувати швидкість обробки запитів.
	T.05.2 Інтегрувати блок рекомендацій на головну сторінку.	ST.05.21 Створити окремий компонент рекомендацій.
		ST.05.22 Виводити персоналізований контент.
		ST.05.23 Адаптувати блок під мобільні пристрої.
...	...	...
US07	T.07.1 Автоматично запитувати відгук після візиту.	ST.07.11 Реалізувати тригер після візиту.
		ST.07.12 Надсилати запит на оцінку.
		ST.07.13 Забезпечити підтримку мультимедіальності.
	T.07.2 Зберігати та обробляти оцінки і коментарі.	ST.07.21 Зберігати фідбек у базі.
		ST.07.22 Валідувати вхідні дані.
		ST.07.23 Інтегрувати з аналітикою.
US08	T.08.1 Створити дашборд для аналізу відгуків.	ST.08.11 Реалізувати таблиці і графіки.
		ST.08.12 Показувати середні оцінки.
		ST.08.13 Побудувати агреговану статистику.
	T.08.2 Реалізувати фільтри за майстрами та періодами.	ST.08.21 Додати фільтри по датах.
		ST.08.22 Фільтрація за іменем майстра.
		ST.08.22 Оновлення даних дашборду динамічно.

Розроблений перелік задач і підзадач, структурований за відповідними User Story, дозволяє забезпечити повну простежуваність між функціональними вимогами, очікуваннями користувачів і технічним плануванням проєкту. Такий підхід сприяє чіткому розподілу обов'язків у

команді, полегшує контроль за виконанням спринтів у рамках Scrum-методології та забезпечує адаптивність процесу розробки до змін у вимогах або пріоритетах. Структура задач створює надійне підґрунтя для подальшого планування релізів та управління ризиками проєкту.

3.4 Організаційна структура управління проєктом

Розробка вебплатформи для діяльності салону краси є складним інноваційним проєктом, який поєднує сучасні IT-рішення, специфіку сфери обслуговування та високі очікування кінцевих користувачів щодо зручності, швидкодії та якості сервісу. Успішна реалізація такого проєкту вимагає чітко структурованого управління, яке забезпечить злагоджену роботу фахівців, оперативність прийняття рішень, ефективне розв'язання проблем і гнучке реагування на зміни вимог замовника.

З урахуванням цих характеристик, найоптимальнішою моделлю управління для даного проєкту є проєктна організаційна структура. Така структура передбачає створення тимчасової, але повністю автономної команди, яка сконцентрована винятково на досягненні конкретної цілі – запуску та впровадженню вебплатформи. Всі учасники мають чітко визначені функціональні ролі, сфери відповідальності та орієнтуються на єдиний результат. Це забезпечує максимально ефективне використання ресурсів, спрощує комунікації та підвищує відповідальність кожного члена команди.

Проєктна структура є особливо актуальною для цифрових розробок, які виконуються в обмежені терміни, вимагають високої технічної координації та передбачають поетапну перевірку та вдосконалення функціоналу. У нашому випадку, вебплатформа для «MOV Hair Studio» включає широкий спектр функціональних модулів (реєстрація користувачів, бронювання послуг, система зворотного зв'язку, маркетинг тощо), що передбачає активну співпрацю між технічною, дизайнерською та бізнес-орієнтованою частинами команди.

У рамках обраної Scrum-методології, робота організована ітеративно, з чіткими часовими межами (спринтами) та постійною взаємодією з користувачами. В центрі цієї структури є Product Owner, який керує продуктом і формує бачення системи, та Scrum Master, який забезпечує методологічну підтримку команди та усуває організаційні бар'єри. Решта команди формується з фахівців різних напрямків, які безпосередньо реалізують функціональні компоненти платформи.

Використання проєктної структури в умовах такого типу розробки дозволяє:

- скоротити цикл прийняття рішень;
- зменшити ризики невідповідності очікувань замовника;
- забезпечити командну концентрацію на конкретному результаті;
- ефективно планувати ресурси й терміни.

Отже, проєктна структура є не просто доцільним вибором, а ключовим чинником успішного управління розробкою та впровадженням вебплатформи «MOV Hair Studio».

На рис. 3.1 наведено графічне представлення проєктної організаційної структури, а в табл. 3.6 наведено інформацію щодо ролей в команді.

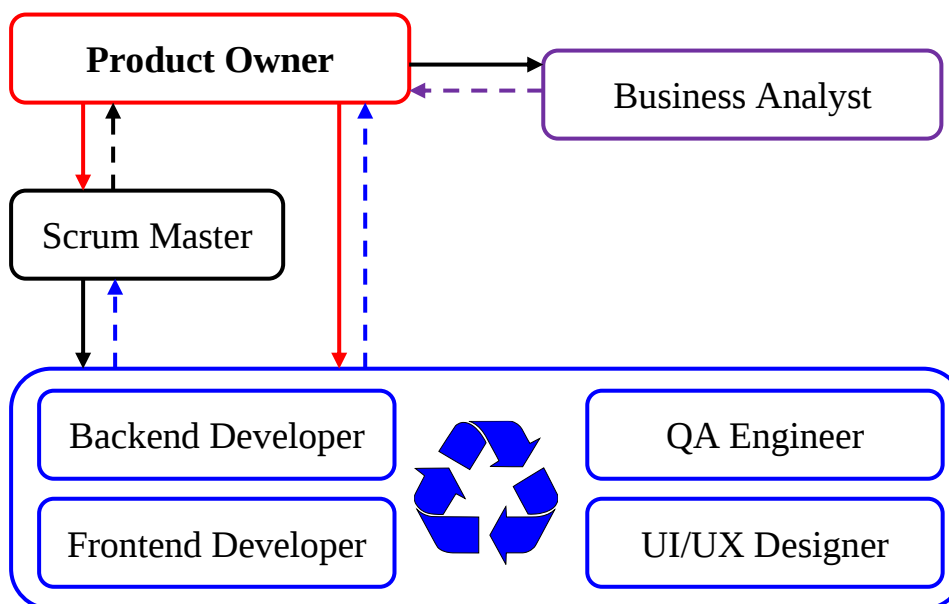


Рис. 3.1. Організаційна структура управління проєктом

Склад команди проєкту та функціональні обов'язки

№	Роль в команді	Основні функціональні обов'язки
1	Product Owner	- Формує бачення продукту. - Працює з пріоритетами беклогу. - Узгоджує вимоги з клієнтом. - Приймає результати спринтів.
2	Scrum Master	- Організовує Scrum-події. - Сприяє самоорганізації команди. - Усуває перешкоди в роботі. - Забезпечує дотримання Agile-практик.
3	Backend Developer	- Створює серверну логіку та API. - Інтегрує базу даних та сторонні сервіси. - Забезпечує безпеку й масштабованість.
4	Frontend Developer	- Розробляє адаптивний інтерфейс. - Реалізує інтерактивність у браузері. - Інтегрує UI з API.
5	UI/UX Designer	- Створює макети й прототипи інтерфейсу. - Розробляє логіку користувацького шляху. - Супроводжує реалізацію дизайну.
6	QA Engineer	- Пише тест-кейси й автотести. - Проводить тестування (функціональне, регресійне, навантажувальне). - Виявляє та верифікує помилки.
7	Business Analyst	- Аналізує бізнес-процеси салону. - Формулює вимоги до функціональності. - Підтримує Product Owner у комунікації з замовником.

Таким чином, обрана проєктна структура управління у поєднанні з використанням Scrum-методології дозволяє сформувати згуртовану, кросфункціональну команду, орієнтовану на результат. Така модель забезпечує ефективне управління проєктом, гнучкість при зміні вимог, а також максимальну адаптацію до потреб замовника й кінцевих користувачів. Чіткий розподіл ролей та відповідальностей у команді створює умови для стабільної, прозорої та контрольованої реалізації всіх етапів проєкту – від аналізу потреб до впровадження готової вебплатформи в реальну діяльність салону краси.

3.5 Планування ресурсного забезпечення проєкту та бюджету

Реалізація проєкту з розробки вебплатформи для діяльності салону краси потребує ретельного планування як людських, так і матеріально-технічних ресурсів. Враховуючи масштаб задач, багатофункціональність платформи, орієнтацію на гнучку методологію Scrum та постійні ітерації, ключовим питанням є формування адекватного бюджету. З огляду на те, що обсяги завдань можуть змінюватися під час розробки, найраціональнішою моделлю обліку витрат є Time & Material.

Модель Time & Material (T&M) передбачає оплату на основі фактично витраченого часу та використаних ресурсів. Такий підхід дозволяє:

- уникнути жорсткої фіксації бюджету на старті;
- адаптуватися до змін у пріоритетах чи нових вимог замовника;
- отримати більш прогнозовані витрати у рамках контрольованих ітерацій;
- забезпечити прозору звітність по кожній годині роботи.

Окрім цього, модель Time & Material надає проєктній команді більшу свободу в плануванні спринтів та розподілі завдань відповідно до їх пріоритетності. Це особливо важливо у контексті нашого проєкту, де платформа має підтримувати адаптивність інтерфейсу, інтеграцію з зовнішніми сервісами та персоналізовану взаємодію з клієнтами. Завдяки цій моделі Product Owner отримує змогу гнучко коригувати напрямки розробки залежно від зворотного зв'язку користувачів або бізнес-потреб. Крім того, застосування T&M дозволяє уникнути перевитрати бюджету на непотрібні або передчасні функції. У підсумку, це забезпечує економічно обґрунтовану розробку без шкоди для якості продукту.

Для проєкту «MOV Hair Studio», де важливу роль відіграє UX-досвід, інтеграції з зовнішніми сервісами, зворотний зв'язок клієнтів, така модель дозволяє балансувати між якістю, швидкістю та гнучкістю реалізації.

В табл. 3.7 наведено склад проєктної команди та оцінено завантаженість ключових фахівців. Ролі визначено відповідно до організаційної структури, розробленої на попередньому етапі. Команда формується з урахуванням принципів Scrum-організації, включаючи обов’язкові ролі Product Owner та Scrum Master, а також ключових технічних спеціалістів.

Таблиця 3.7

Людські ресурси проєкту

Роль	Кількість осіб	Орієнтовна завантаженість	Тривалість участі
Product Owner	1	50%	4 міс.
Scrum Master	1	50%	4 міс.
Frontend Developer	1	100%	4 міс.
Backend Developer	1	100%	4 міс.
UI/UX Designer	1	70%	2 міс.
QA Engineer	1	50%	3 міс.
Business Analyst	1	50%	1,5 міс.

Успішна реалізація проєкту вимагає використання сучасних інструментів і сервісів, які забезпечують ефективну розробку, тестування та розгортання продукту (табл. 3.8).

Таблиця 3.8

Технічна інфраструктура

Категорія	Потреба	Коментарі
Хостинг/сервери	VPS/Cloud сервер на 6 міс.	Окремі середовища: dev/stage/prod.
Система контролю версій	Git (GitHub/Bitbucket)	Інструменти для командної розробки.
Таск-трекер	Jira/Trello	Підтримка Scrum-процесів.
Засоби комунікації	Slack, Zoom	Віддалена співпраця.
Дизайн-середовище	Figma, Zeplin	Для UI/UX прототипування.
CI/CD	GitHub Actions, Docker	Автоматизація тестів і розгортання.

Сформована калькуляція орієнтовного бюджету (табл. 3.9) включає як витрати на роботу команди (табл. 3.9), так і матеріальні витрати на інфраструктуру (табл. 3.10).

Таблиця 3.9

Бюджет проєкту (витрати на команду)

Роль	К-ть	Год/міс	Міс.	Ставка (USD/год)	Вартість (USD)
Product Owner	1	80	4	45	14400
Scrum Master	1	80	4	40	12800
Frontend Developer	1	160	4	35	22400
Backend Developer	1	160	4	40	25600
UI/UX Designer	1	112	2	35	7840
QA Engineer	1	80	3	30	7200
Business Analyst	1	80	1,5	40	4800
Загальна сума	-	-	-	-	95040

Таблиця 3.10

Бюджет проєкту (витрати на матеріали та інфраструктуру)

Матеріал	К-ть	Міс.	Вартість (USD)
VPS/Cloud сервер	1	6	600
GitHub Teams	5 користувачів	6	180
Jira/Trello (платна версія)	5 користувачів	6	300
Figma (Pro-акаунт)	1	2	30
Zoom/Slack (Pro-акаунти)	5 користувачів	4	200
Docker Hub Pro	1	6	90
Загальна сума	-	-	1400

Отже, орієнтовний бюджет проєкту буде становити 96440 доларів.

Планування ресурсного забезпечення в контексті Time & Material дозволяє досягти балансу між фінансовою ефективністю та гнучкістю реалізації проєкту. Обрана структура команди повністю відповідає вимогам Scrum, а використані технічні засоби сприяють оперативній координації розробки та швидкому реагуванню на зміни. Бюджет проєкту є реалістичним для IT-рішення середнього масштабу, охоплює всі ключові етапи життєвого циклу платформи та може адаптуватися під час подальших ітерацій. Такий підхід створює надійне підґрунтя для стабільної, ефективної та контрольованої реалізації вебплатформи для салону краси «MOV Hair Studio».

3.6 Планування спринтів та візуалізація їх виконання за допомогою інструментів MS Project

Для планування спринтів нам потрібно здійснити деталізацію беклогу на роботи які потрібно буде виконати. Фрагмент результатів наведено в табл. 3.11, а повні результати для US01-05 (так як проєкт дуже об'ємний всю інформацію в межах кваліфікаційної роботи представити важко, тому для подальшого представлення було обрано саме ці сторі), наведені в табл. А.3 Додатку А кваліфікаційної роботи.

Таблиця 3.11

Фрагмент деталізації беклогу продукту до рівня робіт

Задача / Task (User Story)	Підзадача / Subtask	Роботи які потрібно виконати
1	2	3
T.01.1 (US01)	ST.01.11 Розробити інтерфейс реєстрації	1. Створити HTML-форму з полями: ім'я, email, пароль. 2. Застосувати клієнтську валідацію (JavaScript). 3. Забезпечити адаптивність для мобільних пристроїв. 4. Протестувати правильність відображення форми на різних екранах.
	ST.01.12 Зберігати користувача в базі даних	1. Налаштувати модель користувача у базі даних. 2. Реалізувати збереження даних за допомогою ORM. 3. Обробити помилки при збереженні (у тому числі дублі email). 4. Протестувати успішне створення користувача.
	ST.01.13 Створювати особистий кабінет при реєстрації	1. Створити модель особистого кабінету. 2. Прив'язати кабінет до користувача після реєстрації. 3. Створити початковий шаблон кабінету. 4. Протестувати автоматичне створення кабінету.
T.01.2 (US01)	ST.01.21 Додати валідацію email і пароля	1. Написати функції перевірки формату email. 2. Додати перевірку складності пароля (довжина, символи). 3. Виводити повідомлення про помилки введення. 4. Протестувати валідацію різних варіантів введення.
	ST.01.22 Застосувати хешування паролів	1. Встановити бібліотеку хешування (наприклад, bcrypt). 2. Реалізувати хешування пароля перед збереженням. 3. Зберігати лише хеш у базі даних. 4. Протестувати коректність хешування пароля.
	ST.01.23 Обробляти помилки при збереженні	1. Реалізувати механізм обробки винятків. 2. Налаштувати логування помилок. 3. Повернення користувачу інформативних повідомлень. 4. Створити unit-тести для виняткових ситуацій.

1	2	3
US02 (T.02.1)	ST.02.11 Створити форму входу	1. Дизайн UI для логіну (email, пароль, кнопка входу). 2. Реалізувати надсилання даних на сервер. 3. Обробити помилки у формі. 4. Протестувати поведінку форми при різних сценаріях.
	ST.02.12 Перевірити введені дані	1. Перевірити наявність email у БД. 2. Порівняти хешовані паролі. 3. Повернути повідомлення про успіх або помилку. 4. Протестувати успішні/неуспішні сценарії входу.
	ST.02.13 Створити сесію при успішному вході	1. Налаштувати сесію для авторизованого користувача. 2. Створити механізм збереження токена. 3. Забезпечити перехід на особистий кабінет після входу. 4. Протестувати створення сесії.
US02 (T.02.2)	ST.02.21 Налаштувати авторизацію через Google	1. Зареєструвати застосунок у Google Cloud. 2. Інтегрувати Google OAuth2.0 через SDK. 3. Налаштувати маршрути перенаправлення. 4. Протестувати авторизацію через Google.
	ST.02.22 Налаштувати авторизацію через Facebook	1. Зареєструвати застосунок у Facebook Developers. 2. Інтегрувати Facebook SDK. 3. Обробити callback від Facebook. 4. Протестувати авторизацію через Facebook.
	ST.02.23 Прив'язати соцмережу до профілю користувача	1. берегти токен у профілі користувача. 2. Реалізувати перевірку існуючого облікового запису. 3. Зв'язати профіль із соцмережею. 4. Протестувати сценарій прив'язки акаунтів.

Тепер маючи достатню кількість вихідної інформації сплануємо два спринти.

В табл. 3.12-3.13 наведено планування двох спринтів.

Таблиця 3.12

Інформація щодо планування Спринту 1

Задача / Підзадача	Назва роботи	Тривалість (год)	Виконавець	Матеріал, технології
1	2	3	4	5
ST.01.11	Створити HTML-форму з полями	4	Frontend Developer	Figma, VS Code
ST.01.11	Застосувати клієнтську валідацію	4	Frontend Developer	JavaScript
ST.01.11	Адаптивність для мобільних пристроїв	4	Frontend Developer	VS Code
ST.01.11	Протестувати форму	4	QA Engineer	BrowserStack
ST.01.12	Модель користувача в БД	5	Backend Developer	SQL

1	2	3	4	5
ST.01.12	ORM-збереження	5	Backend Developer	Django ORM
ST.01.12	Обробка помилок	4	Backend Developer	Python, Django
ST.01.12	Тестування збереження	4	QA Engineer	Postman
ST.01.13	Модель особистого кабінету	5	Backend Developer	Django Models
ST.01.13	Прив'язка кабінету	5	Backend Developer	Python, Django
ST.01.13	Шаблон кабінету	5	Frontend Developer	HTML, CSS
ST.01.13	Тестування створення кабінету	5	QA Engineer	Postman

Таблиця 3.13

Інформація щодо планування Спринту 2

Задача / Підзадача	Назва роботи	Тривалість (год)	Виконавець	Матеріал, технології
ST.01.21	Перевірка формату email	4	Frontend Developer	JavaScript
ST.01.21	Перевірка складності пароля	4	Frontend Developer	JavaScript
ST.01.21	Повідомлення про помилки	3	Frontend Developer	VS Code
ST.01.21	Тестування валідації	3	QA Engineer	BrowserStack
ST.01.22	Встановлення bcrypt	4	Backend Developer	bcrypt
ST.01.22	Реалізація хешування	4	Backend Developer	Python
ST.01.22	Збереження хешу в БД	4	Backend Developer	SQL
ST.01.22	Тестування хешування	4	QA Engineer	Postman
ST.02.11	UI-дизайн форми логіну	4	UI/UX Designer	Figma
ST.02.11	Надсилення даних	4	Frontend Developer	JavaScript
ST.02.11	Обробка помилок	4	Frontend Developer	VS Code
ST.02.11	Тестування логіну	4	QA Engineer	BrowserStack

Для візуалізації виконання спринтів у проєкті було використано інструмент MS Project, який дозволяє ефективно представити план-графік задач у вигляді діаграми Ганта. Кожна задача та підзадача, включена до спринтів, була внесена до проєктного плану з визначеними тривалістю,

виконавцем та логічними зв'язками між роботами. Такий підхід забезпечує наочне відображення обсягу робіт, черговості їх виконання, можливих паралельних дій, а також контроль за перевищенням встановлених часових меж кожного спринту. За допомогою інструментів MS Project команда може відстежувати прогрес виконання, виявляти критичний шлях та завчасно виявляти ризики, пов'язані з відставанням від графіка. Це сприяє більш злагодженому управлінню розробкою та підвищує прозорість у взаємодії між учасниками проєкту.

Для всіх спринтів проєкту візуалізація виконання спринтів буде здійснюватися в середовищі MS Project, а на рис. 3.2 наведено приклад візуалізації Спринту 1 та діаграми Ганта.

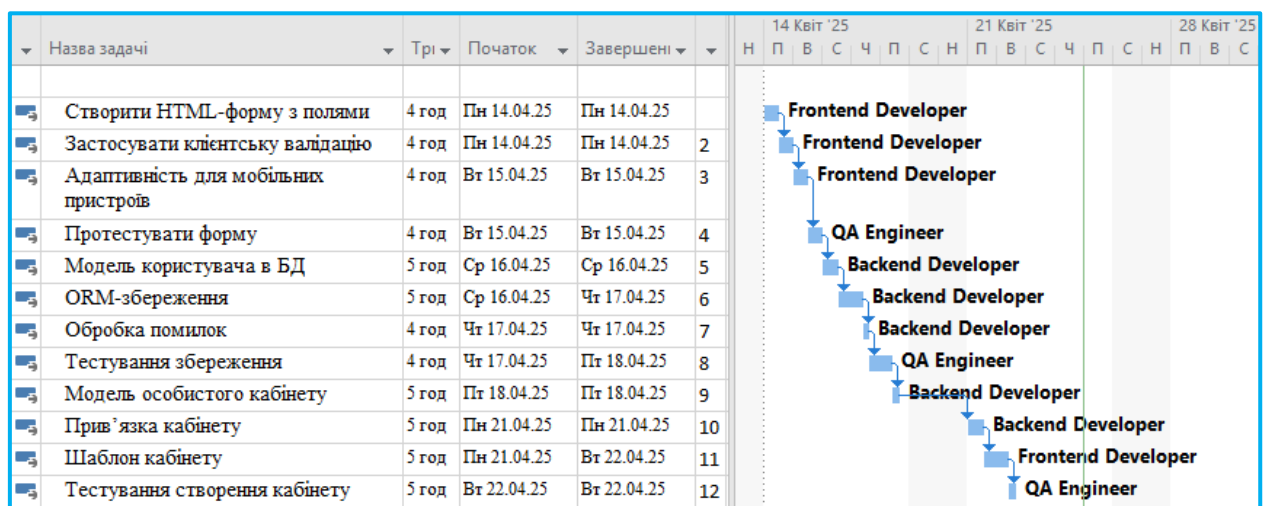


Рис. 3.2. Візуалізація Спринту 1 та побудова діаграми Ганта в MS Project

3.7 Ідентифікація проєктних ризиків та розробка карти управління ними

Успішна реалізація будь-якого інформаційного проєкту, зокрема в галузі розробки вебплатформ, тісно пов'язана з ефективним управлінням ризиками, які можуть виникнути на всіх етапах життєвого циклу. Особливо це актуально в умовах високої динаміки ІТ-середовища, обмежених часових та фінансових ресурсів, а також зростаючих очікувань кінцевих користувачів. Проєкт

створення вебплатформи для діяльності салону краси, як приклад сучасного цифрового рішення у сфері малого бізнесу, також потребує комплексного підходу до виявлення, аналізу та контролю ризиків.

У цьому проєкті платформа виконує функції онлайн-бронювання послуг, управління записами клієнтів, збору зворотного зв'язку, адміністрування послуг і працівників, а також підтримки маркетингових активностей. Складність таких функціональних вимог зумовлює підвищену вразливість проєкту до ризиків технічного, організаційного, фінансового та правового характеру. У зв'язку з цим ідентифікація ризиків та побудова карти управління ними стає невід'ємною складовою якісного проєктного менеджменту.

Процес управління ризиками передбачає послідовне проходження таких етапів: виявлення ризиків, їх кількісна або якісна оцінка, визначення ймовірності виникнення та рівня впливу, розробка відповідних стратегій реагування, а також постійний моніторинг і контроль [43-44]. Кожен із цих етапів має бути адаптованим до специфіки обраного проєкту, враховуючи його цілі, зацікавлені сторони, структуру команди та технологічну базу.

Ідентифікація ризиків – це перший і надзвичайно критичний крок. На цьому етапі проводиться збір потенційних загроз із залученням членів команди, експертів, аналізом подібних реалізованих проєктів та оцінкою внутрішніх і зовнішніх чинників [45]. У контексті вебплатформи для салону краси важливо враховувати як загальні ІТ-ризики (недоступність хостингу, помилки в коді, кіберзагрози), так і специфічні ризики, пов'язані з бізнес-середовищем (затримки з боку власника салону, зміни у вимогах, низька цифрова обізнаність персоналу тощо).

Подальшим кроком є класифікація ризиків за групами – технічні, управлінські, зовнішні, юридичні, комунікаційні, ресурсні тощо. Такий поділ дозволяє систематизувати інформацію та полегшити розробку відповідних стратегій реагування. Важливим інструментом на цьому етапі є матриця ризиків, яка поєднує ймовірність настання події з очікуваним рівнем її впливу

на проєкт. Формування карти управління ризиками дозволяє визначити пріоритетні загрози та розробити відповідні превентивні та коригувальні заходи [46-48]. Наприклад, ризик втрати даних можна мінімізувати через регулярне резервне копіювання, а ризик неузгодженості у вимогах клієнта – шляхом проведення регулярних зустрічей і затвердження специфікацій на початку кожного етапу.

Особливе значення в управлінні ризиками має культура проєктного середовища – готовність команди відкрито говорити про можливі загрози, фіксувати відхилення та швидко адаптуватися до змін. У проєкті створення вебплатформи для салону краси гнучкість і проактивність у роботі з ризиками є критичними факторами, що забезпечують відповідність очікуванням замовника та стабільну роботу продукту після його впровадження.

Таким чином, управління ризиками в рамках цього проєкту є не лише засобом зниження невизначеності, але й інструментом підвищення якості кінцевого продукту, оптимізації витрат і досягнення стратегічних цілей. У наступному підрозділі буде представлено деталізовану карту ідентифікованих ризиків, класифікацію за категоріями та опис заходів щодо їх мінімізації.

Ідентифікацію, оцінку та розробку карти протиризикових заходів для проєкту «Розробка проєкту створення вебплатформи для діяльності салону краси» будемо здійснювати згідно рекомендацій представлених у вказівках [49].

Найбільш типовими класами ризиків для ІТ-проєктів є технічні, управлінські, операційні, юридичні та регуляторні, фінансові ризики; ризики пов'язані із замовником; ризики пов'язані із користувачами та форс-мажорні ризики.

Зазначені класи ризиків охоплюють найпоширеніші джерела невизначеності, з якими стикаються ІТ-команди в процесі планування, розробки, впровадження та супроводу цифрових рішень. Кожен із цих типів ризиків може мати як безпосередній, так і відкладений вплив на результат проєкту, затримуючи його реалізацію, знижуючи якість продукту або

створюючи додаткові витрати на виправлення проблем. Наприклад, технічні ризики можуть призвести до нестабільної роботи платформи, а управлінські – до втрати контролю над строками виконання.

Оцінка та управління цими ризиками потребує системного підходу та постійної взаємодії між усіма учасниками проєкту. Важливо не лише вчасно ідентифікувати можливі загрози, але й розробити дієві стратегії реагування на них, виходячи з пріоритетності та потенційного впливу. Зокрема, форс-мажорні ризики, як-от техногенні аварії, збої в електропостачанні чи політична нестабільність, хоч і мають низьку ймовірність, можуть мати катастрофічні наслідки, якщо їх не враховано у плануванні. Отже, глибоке розуміння природи ризиків та їх інтеграція в систему проєктного управління є критичним чинником успішного виконання ІТ-проєктів.

Проведемо ідентифікацію ризиків для проєкту «Розробка проєкту створення вебплатформи для діяльності салону краси», а результат представимо у вигляді табл. 3.14.

Таблиця 3.14

Ідентифікація ризиків проєкту

Код ризику	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	2	3	4	5
P01	Технічні	Вибір застарілої технології	висока	середня
P02	Технічні	Помилки в програмному коді	висока	середня
P03	Технічні	Складність інтеграції з API	середня	середня
P04	Технічні	Низька продуктивність вебплатформи	висока	середня
P05	Технічні	Вразливість до кіберзагроз	висока	низька
P06	Технічні	Відсутність мобільної адаптації	середня	висока
P07	Управлінські	Погано визначені вимоги	висока	висока
P08	Управлінські	Часті зміни в специфікації	середня	середня
P09	Управлінські	Недостатній досвід керівника проєкту	висока	середня
P10	Управлінські	Низький рівень комунікації в команді	висока	висока
P11	Операційні	Збій хостингу	висока	низька
P12	Операційні	Недоступність у пікові години	середня	середня
P13	Операційні	Проблеми з оновленням платформи	середня	середня

1	2	3	4	5
P14	Операційні	Відсутність плану відновлення	висока	низька
P15	Юридичні та регуляторні	Порушення GDPR (обробка даних без згоди)	висока	середня
P16	Юридичні та регуляторні	Використання неліцензованих компонентів	висока	висока
P17	Юридичні та регуляторні	Відсутність політики конфіденційності	середня	висока
P18	Фінансові	Перевищення бюджету	висока	середня
P19	Фінансові	Затримка фінансування	висока	низька
P20	Фінансові	Недооцінка вартості підтримки	середня	середня
P21	Пов'язані із замовником	Зміна бачення проєкту з боку замовника	висока	низька
P22	Пов'язані із замовником	Невчасне надання інформації	середня	середня
P23	Пов'язані із замовником	Пасивність у прийнятті рішень	середня	низька
P24	Пов'язані із користувачами	Складний інтерфейс	висока	висока
P25	Пов'язані із користувачами	Небажання переходити на нову систему	середня	низька
P26	Пов'язані із користувачами	Відсутність навчання персоналу	середня	висока
P27	Форс-мажорні	Відключення електроенергії	висока	низька
P28	Форс-мажорні	Військові дії або політична нестабільність	висока	низька

Після завершення етапу ідентифікації ризиків, коли було виявлено та класифіковано найбільш вірогідні загрози для проєкту створення вебплатформи для діяльності салону краси, наступним кроком є їх оцінка з точки зору рівня впливу та ймовірності виникнення. Цей процес дозволяє не лише усвідомити потенційну загрозу кожного ризику, а й визначити, на які з них потрібно звернути першочергову увагу при плануванні заходів реагування.

Оцінювання ризиків передбачає використання як якісних, так і кількісних підходів, залежно від доступності даних, досвіду команди та складності ризикових подій. Якісна оцінка базується на експертних судженнях, ранжуванні ризиків за шкалою (низький, середній, високий) та їх візуальному представленні у вигляді матриці. Такий підхід є ефективним на ранніх етапах, коли ще відсутня детальна статистика або точні числові параметри.

Результат якісної та кількісної оцінки ризиків наведено в табл. 3.15.

Таблиця 3.15

Оцінювання ризиків проєкту

№	Ризикова подія	Затримки у часі		Фінансові втрати		Ймовір- ність		Частота (за проєкт)		Важливість ризиків
		ЯО	КО	ЯО	КО	ЯО	КО	ЯО	КО	
1	2	3	4	5	6	7	8	9	10	11
01	Вибір застарілої технології	СС	5	СВ	6	СН	4	СС	5	24
02	Помилки в програмному коді	СВ	5	СС	5	ВС	6	ВС	6	30
03	Складність інтеграції з API	НС	2	СС	5	СС	4	НС	2	20
04	Низька продуктивність вебплатформи	ВС	6	ВС	6	ВС	6	СВ	5	36
05	Вразливість до кіберзагроз	ВВ	9	ВВ	9	ВС	6	ВС	6	54
06	Відсутність мобільної адаптації	НС	2	НС	2	СС	4	СВ	5	8
07	Погано визначені вимоги	СВ	5	СВ	5	ВС	6	СВ	5	30
08	Часті зміни в специфікації	СВ	5	СВ	5	СВ	5	НС	2	25
09	Недостатній досвід керівника проєкту	ВС	6	ВС	6	ВС	6	СВ	5	36
10	Низький рівень комунікації в команді	СВ	5	ВС	6	СВ	5	СВ	5	30
11	Збій хостингу	ВВ	9	ВВ	9	СВ	5	ВС	6	45
12	Недоступність у пікові години	НС	2	СВ	5	СВ	5	СВ	5	25
13	Проблеми з оновленням платформи	СВ	5	НС	2	СВ	5	НС	2	10
14	Відсутність плану відновлення	ВВ	8	ВВ	8	ВВ	8	ВВ	8	64
15	Порушення GDPR (обробка даних без згоди)	ВВ	8	ВВ	8	СВ	5	СВ	5	40
16	Використання неліцензованих компонентів	СВ	5	ВВ	8	СВ	5	НС	2	40
17	Відсутність політики конфіденційності	НС	2	НС	2	СВ	5	НС	2	10
18	Перевищення бюджету	СВ	5	ВС	6	ВС	6	ВС	6	36
19	Затримка фінансування	ВВ	8	ВВ	8	ВС	6	ВС	6	48
20	Недооцінка вартості підтримки	СВ	5	СВ	5	СВ	5	СВ	5	25
21	Зміна бачення проєкту з боку замовника	ВВ	8	ВВ	8	СВ	5	СВ	5	40
22	Невчасне надання інформації	СВ	5	НС	2	СВ	5	НС	2	10
23	Пасивність у прийнятті рішень	НС	2	НС	2	СВ	5	НС	2	10
24	Складний інтерфейс	СВ	5	СВ	5	ВС	6	СВ	5	30
25	Небажання переходити на нову систему	НС	2	НС	2	НС	2	НС	2	4
26	Відсутність навчання персоналу	НС	2	НС	2	СВ	5	НС	2	10
27	Відключення електроенергії	ВВ	8	СВ	5	ВВ	8	ВВ	8	40
28	Військові дії або політична нестабільність	ВВ	8	ВВ	8	ВВ	8	ВВ	8	64

Після проведення кількісної та якісної оцінки проєктних ризиків стає можливим здійснити наступний критично важливий крок – розробку карти протиризикових заходів. Оцінювання дозволило не лише виявити найуразливіші точки проєкту, а й чітко визначити рівень впливу кожного ризику на його хід, що створює основу для формування ефективних стратегій реагування. Таким чином, на основі отриманих даних доцільно визначити пріоритети в управлінні ризиками та розробити конкретні дії, спрямовані на запобігання, зниження, прийняття або передавання кожного із них. У подальшому підрозділі буде детально представлено карту протиризикових заходів, що відповідає характеру і рівню кожного оціненого ризику в межах проєкту розробки вебплатформи для салону краси. Фрагмент карти протиризикових заходів наведено в табл. 3.16, наведені в табл. Б.1 Додатку Б кваліфікаційної роботи.

Таблиця 3.16

Фрагмент карти протиризикових заходів

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		профілактика		при симптомі	при проблемі
1	2	3	4	5	6
02	Помилки в програмному коді	Код-рев'ю, модульне тестування	Часті збої, падіння сторінок	Фіксація помилок через логери	Рефакторинг або перерозробка модуля
04	Низька продуктивність вебплатформи	Оптимізація запитів	Затримки відгуку понад 2 с	Аналіз продуктивності	Використання кешування, редизайн запитів
05	Вразливість до кіберзагроз	Впровадження брандмауєру та шифрування	Підозрілі дії або скарги користувачів	Аудит безпеки	Блокування, відкат, інформування користувачів
09	Недостатній досвід керівника проєкту	Навчання або консультації	Помилки в управлінні, затримки	Підтримка з боку Scrum Master	Заміна/делегування керівника
11	Збій хостингу	Резервні сервери	Неможливість підключення до сайту	Перемикання на резервний хостинг	Аналіз причин збою, масштабування
15	Порушення GDPR (обробка даних без згоди)	Аналіз політики обробки даних	Скарги на використання персональних даних	Консультація з юристом	Зміна логіки збирання/обробки
18	Перевищення бюджету	Буфер бюджету	Перевищення планових витрат	Аналіз бюджету	Оптимізація витрат, корекція бюджету
19	Затримка фінансування	Погодження графіку фінансування	Відтермінування платежів	Резервне фінансування	Зміна пріоритетів або етапів реалізації

1	2	3	4	5	6
21	Зміна бачення проєкту з боку замовника	Фіксація бачення на старті	Невідповідність очікувань	Демонстрація результатів	Переговори та документування змін
27	Відключення електроенергії	Джерело безперебійного живлення	Нестабільна робота системи	Активація аварійного режиму	Аналіз втрат, автоматичне відновлення

У межах підрозділу, присвяченого управлінню ризиками, було здійснено повний цикл аналізу потенційних загроз, які можуть вплинути на реалізацію проєкту створення вебплатформи для діяльності салону краси. Розпочинаючи з ідентифікації ризиків, було визначено 28 ключових ризикових подій, класифікованих за типами: технічні, управлінські, операційні, юридичні, фінансові, а також ризики, пов'язані з користувачами, замовником і зовнішніми факторами. Такий підхід дозволив систематизувати інформацію та охопити всі критичні зони потенційної вразливості проєкту.

Подальша якісна і кількісна оцінка ризиків дала змогу виявити найбільш значущі загрози, що потребують пріоритетної уваги. Зокрема, було визначено рівень впливу кожного ризику на строки реалізації, бюджет, імовірність виникнення та частоту прояву.

Завершальним етапом стало формування карти протиризикових заходів, у якій для кожного ризику запропоновано набір дій у трьох площинах: профілактика, реагування на симптоми та дії при виникненні проблеми. Цей інструмент підвищує готовність команди до оперативного реагування, забезпечує зниження впливу негативних подій та зміцнює загальну стійкість проєкту до непередбачених ситуацій.

Таким чином, системне управління ризиками не лише сприяє досягненню стратегічних цілей проєкту, а й створює передумови для його успішної реалізації в рамках визначених ресурсів, часу та очікувань замовника. З урахуванням отриманих результатів, команда проєкту має чіткий план дій, що дозволяє мінімізувати негативні наслідки та забезпечити стабільний хід виконання робіт.

РОЗДІЛ 4. РОЗРОБКА ВЕБПЛАТФОРМИ ЯК ПРОДУКТУ РЕАЛІЗАЦІЇ ПРОЄКТУ

4.1 Формування алгоритмів роботи вебпорталу

Кожна комп'ютерна програма функціонує згідно з певною логікою, що визначається чіткою послідовністю дій – алгоритмом. Поняття алгоритм походить від латинізованого варіанта імені арабського вченого аль-Хорезмі (Algorithmi) та означає впорядкований набір інструкцій, що описують дії виконавця для досягнення конкретного результату у процесі розв'язання задачі за скінченну кількість кроків. Це набір правил, які забезпечують дискретний процес виконання з досягненням поставленої мети у межах обмеженого часу. Для наочності та кращого розуміння таких процесів часто застосовуються блок-схеми, які дозволяють візуалізувати логіку виконання алгоритму.

У контексті розробки програмного забезпечення алгоритм являє собою послідовність формалізованих дій, що реалізують певні обчислення або логіку. Цей процес стартує з визначеного початкового стану, проходить через низку логічних етапів і завершується досягненням кінцевого стану. У деяких випадках алгоритми можуть містити елементи випадковості, наприклад, у стохастичних процесах чи адаптивних системах.

З огляду на те, що дана система тісно інтегрована з базою даних, доцільно розглянути приклади алгоритмів виконання базових SQL-команд. Однією з найуживаніших команд є SELECT – її алгоритм зображено на рис. 4.1. Ця команда призначена для витягання даних із заданої таблиці або з кількох таблиць, представлень (view) чи тимчасових структур бази даних. Як результат виконання, повертається набір рядків, який може бути як порожнім, так і містити множину записів залежно від умов вибірки.

Іншою важливою командою є DELETE, алгоритм виконання якої наведено на рис. 4.2. У мовах, подібних до SQL, ця команда забезпечує видалення записів із таблиці. Записи, які підлягають видаленню, визначаються за допомогою умови, зазначеної у виразі WHERE. Якщо ж така умова відсутня, команда призводить до повного очищення таблиці від усіх записів.

Особливу увагу слід звернути на алгоритм підключення до бази даних, адже цей процес є критичним для функціонування всієї системи. Без належного встановлення з'єднання з базою даних робота з інформацією стає неможливою. Алгоритм цього процесу наведено на рис. 4.3. Він охоплює етапи ініціалізації з'єднання, перевірки доступності ресурсу, аутентифікації користувача та встановлення стабільного каналу обміну даними з БД.

Таким чином, описані алгоритми демонструють основні операції, які лежать в основі функціонування програмної системи, і забезпечують коректну взаємодію з базою даних, що є її ключовим компонентом.

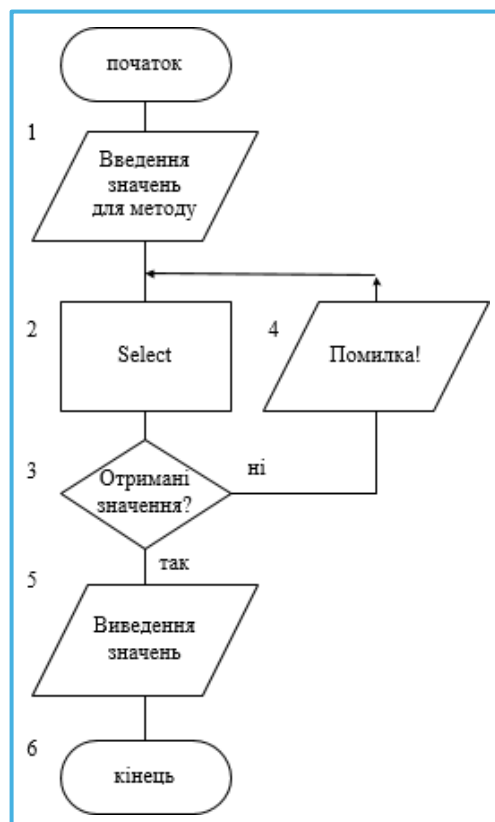


Рис. 4.1. Алгоритм перевірки роботи команди «Select»

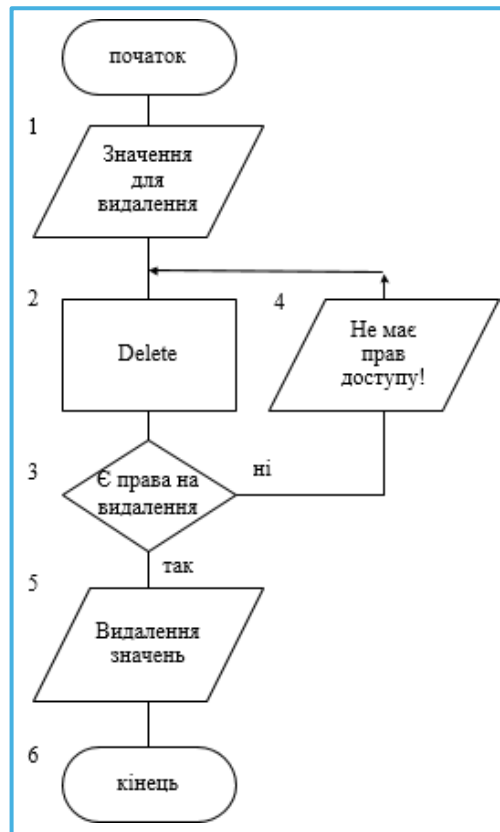


Рис. 4.2. Алгоритм перевірки роботи команди «Delete»

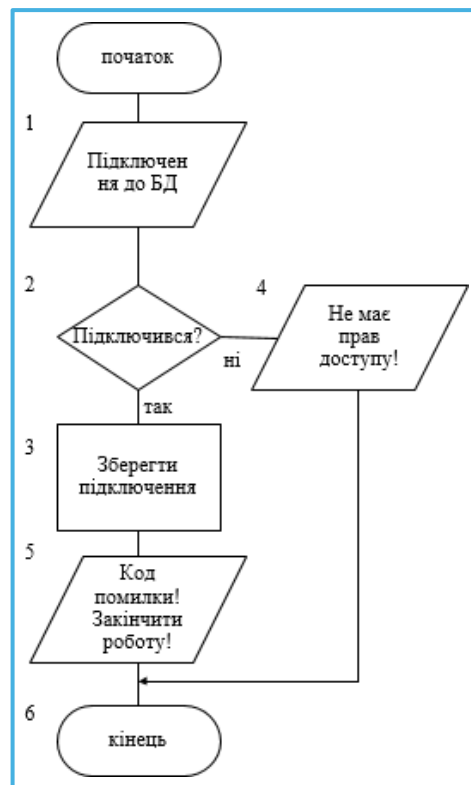


Рис. 4.3. Алгоритм, що здійснює перевірку підключення до бази даних

У зв'язку з великою кількістю функціональних компонентів додатку та значним обсягом програмного коду, у Додатку В кваліфікаційної роботи представлено алгоритми реалізації окремих ключових функцій вебплатформи. Зокрема, наведено алгоритм перевірки логіна та пароля під час авторизації користувача (рис. В.1), що є критично важливою функцією для забезпечення захищеного доступу до системи.

Також подано алгоритм перевірки наявності співробітника в базі даних при реєстрації нового працівника (рис. В.2). Ця функція дозволяє уникнути дублювання записів та підтримувати цілісність облікової інформації.

Вказані алгоритми є невід'ємною частиною забезпечення стабільної та безпечної взаємодії системи з клієнтами та персоналом, а їх правильна реалізація має принципове значення для функціонування вебплатформи в цілому.

4.2 Концептуальне та даталогічне моделювання бази даних продукту проєкту

Першим кроком у процесі проєктування бази даних виступає етап концептуального моделювання, який полягає у формуванні загальної моделі даних відповідно до специфіки досліджуваної предметної області. Його головне завдання – ідентифікувати основні об'єкти (сутності), що є значущими для функціонування майбутньої системи, а також визначити взаємозв'язки між ними на логічному рівні [50-51].

У результаті аналізу предметної області для інформаційної системи салону краси було виокремлено такі сутності: client – клієнти; employee – співробітники; register – запис на прийом; specialty – спеціальність; users – користувачі системи; service – сервіси, що надаються.

Далі для кожної сутності були проаналізовані взаємозв'язки.

Сутності «Клієнт – Запис» (рис. 4.4): зв'язок типу один до багатьох (1:Б), оскільки один клієнт може мати багато записів, але кожен запис належить лише одному клієнту, який ідентифікується за номером телефону.

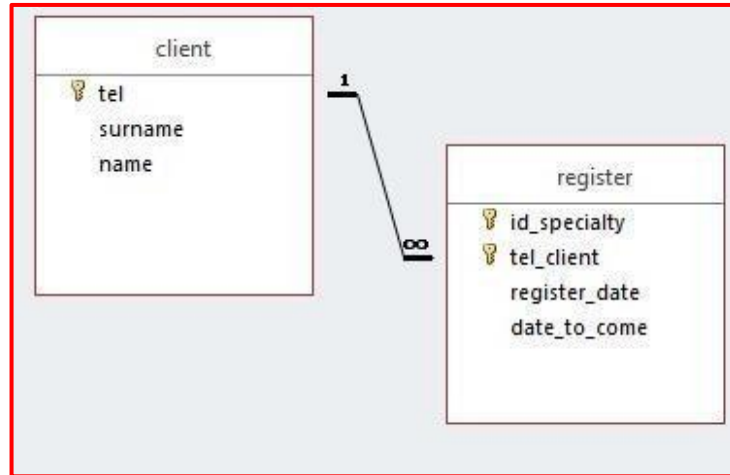


Рис. 4.4. Взаємозв'язок сутностей «Клієнт – Запис»

Сутності «Співробітник – Спеціальність» (рис. 4.5): зв'язок 1:Б, адже один співробітник (ідентифікований за унікальним податковим номером) може володіти кількома спеціальностями.

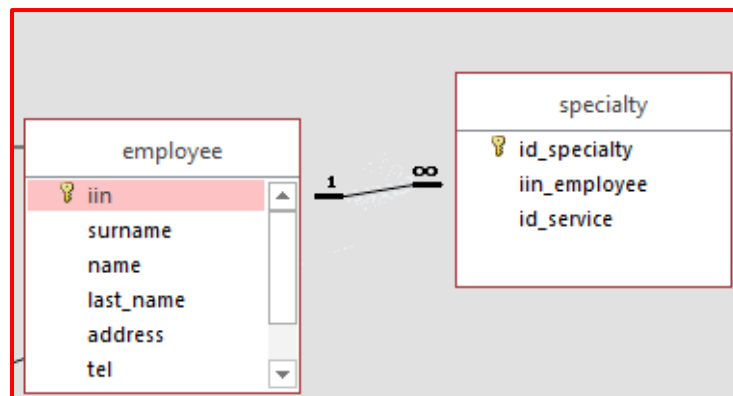


Рис. 4.5. Взаємозв'язок сутностей «Співробітник – Спеціальність»

Сутності «Сервіс – Спеціальність» (рис. 4.6): зв'язок 1:Б, тому що кожен сервіс має унікальну назву, але може бути пов'язаний із багатьма спеціальностями, які реалізують даний сервіс.

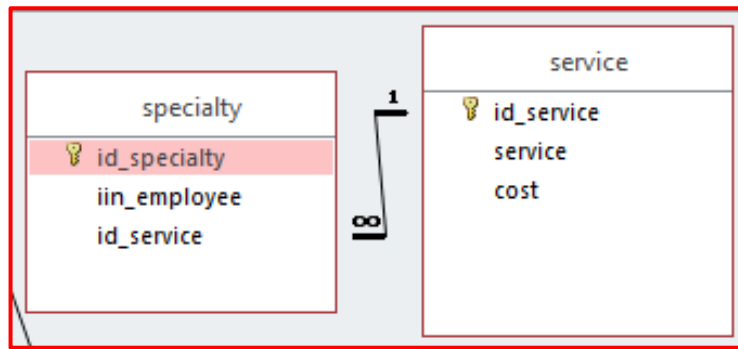


Рис. 4.6. Взаємозв'язок сутностей «Сервіс – Спеціальність»

Сутності «Співробітник – Користувач» (рис. 4.7): зв'язок 1:1, оскільки кожному співробітнику відповідає єдиний обліковий запис у системі з унікальним логіном.

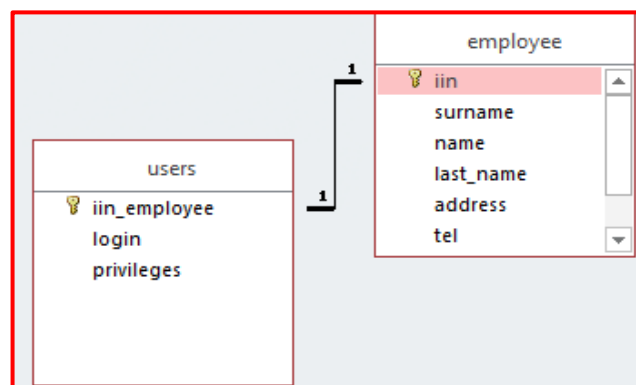


Рис. 4.7. Взаємозв'язок сутностей «Співробітник – Користувач»

Сутності «Спеціальність – Запис» (рис. 4.8): зв'язок 1:Б, бо одна спеціальність може використовуватись у великій кількості записів, проте кожен запис відповідає лише одній спеціальності.

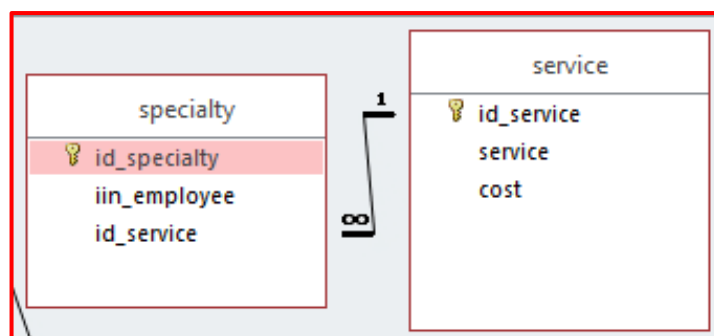


Рис. 4.8. Взаємозв'язок сутностей «Спеціальність – Запис»

Сумарна концептуальна модель бази даних, побудована на основі вказаних сутностей та зв'язків, представлена у Додатку Г кваліфікаційної роботи (рис. В.1).

Даталогічне (логічне) моделювання – це наступний етап після концептуального, що полягає у трансформації концептуальної моделі у логічну модель, орієнтовану на специфіку обраної системи управління базами даних (СУБД).

Метою логічного моделювання є побудова логічної схеми бази даних, яка враховує вимоги конкретної СУБД. Оскільки більшість сучасних СУБД побудовані на основі реляційної моделі, саме до такої моделі й приводиться концептуальне подання даних.

На основі розробленої концептуальної моделі, логічна структура бази даних повинна включати наступні таблиці:

- client – зберігає дані про клієнтів;
- employee – інформація про співробітників;
- register – облік записів на послуги;
- specialty – спеціалізації персоналу;
- users – облікові записи користувачів;
- service – надані сервіси.

Ці таблиці формують основу майбутньої бази даних і будуть реалізовані відповідно до норм реляційного підходу.

Деталізація переліку атрибутів та відповідних їм таблиць, наведено на рис. 4.9 – 4.15.

Field	Type	Null	Key	Default	Extra
tel	varchar(20)	NO	PRI	NULL	
surname	varchar(20)	YES		NULL	
name	varchar(20)	YES		NULL	

Рис. 4.9. Перелік атрибутів, що складаю сутність «Клієнти»

Field	Type	Null	Key	Default	Extra
iin	varchar(12)	NO	PRI	NULL	
surname	varchar(20)	YES		NULL	
name	varchar(20)	YES		NULL	
last_name	varchar(20)	YES		NULL	
address	varchar(80)	YES		NULL	
tel	varchar(20)	YES		NULL	
birth_date	date	YES		NULL	
employee_type	varchar(30)	YES		NULL	

Рис. 4.10. Перелік атрибутів, що складаю сутність «Співробітники»

Field	Type	Null	Key	Default	Extra
id_specialty	int(10) unsigned	NO	PRI	NULL	
tel_client	varchar(20)	YES	MUL	NULL	
register_date	date	YES		NULL	
date_to_come	timestamp	NO	PRI	NULL	

Рис. 4.11. Перелік атрибутів, що складаю сутність «Запис»

Field	Type	Null	Key	Default	Extra
id_service	int(10) unsigned	NO	PRI	NULL	auto_increment
service	varchar(80)	YES		NULL	
cost	int(10) unsigned	YES		NULL	

Рис. 4.12. Перелік атрибутів, що складаю сутність «Сервіс»

Field	Type	Null	Key	Default	Extra
id_specialty	int(10) unsigned	NO	PRI	NULL	auto_increment
iin_employee	varchar(12)	YES	MUL	NULL	
id_service	int(10) unsigned	YES	MUL	NULL	

Рис. 4.13. Перелік атрибутів, що складаю сутність «Спеціальність»

Field	Type	Null	Key	Default	Extra
iin_employee	varchar(12)	NO	PRI	NULL	
login	varchar(20)	YES		NULL	
privileges	enum('a','u')	YES		NULL	

Рис. 4.15. Перелік атрибутів, що складаю сутність «Користувачі»

У результаті логічного проектування, що базується на попередньо сформованій концептуальній моделі, була побудована логічна структура бази

даних, яка відповідає реляційній моделі. Усі відношення (таблиці) цієї моделі приведені до третьої нормальної форми (3НФ). Це означає, що:

- усі атрибути у кожному відношенні знаходяться у повній функціональній залежності (ФЗ) від первинного ключа;
- відсутні повторювані групи атрибутів;
- усунені транзитивні та часткові залежності, що забезпечує цілісність і несуперечливість даних, а також підвищує ефективність обробки запитів.

Для створення таблиць та управління ними використовується мова структурованих запитів SQL (Structured Query Language), яка виконує не лише обробку даних, але й дозволяє реалізувати повний спектр операцій над об'єктами бази даних, зокрема:

- створення та модифікація таблиць;
- встановлення обмежень цілісності;
- організація зв'язків між таблицями;
- управління правами доступу.

Існує два основні підходи до створення таблиць у системах управління базами даних (СУБД):

1. Інтерактивний підхід – більшість сучасних СУБД, зокрема MySQL, PostgreSQL, MS SQL Server, пропонують візуальні графічні інтерфейси (GUI), які дають змогу створювати таблиці вручну без написання SQL-коду. Ці інструменти автоматично генерують відповідні SQL-команди на основі дій користувача. Такий підхід є зручним на початкових етапах розробки та для користувачів без глибоких знань SQL.
2. Програмний підхід – передбачає написання SQL-інструкцій вручну для повного контролю над створенням і параметрами таблиць. Цей спосіб є рекомендованим при розробці промислових проєктів, автоматизації процесів та інтеграції БД у програмне забезпечення. При цьому розробник самостійно визначає:

- назву таблиці;
- перелік її атрибутів (стовпців) із зазначенням типів даних;
- первинні та зовнішні ключі;
- додаткові обмеження та опції (наприклад, поведінка при оновленні або видаленні записів).

Варто зазначити, що у візуальному середовищі, навіть при використанні графічних інструментів, насправді відбувається генерація SQL-коду у фоновому режимі, аналогічно до того, як у середовищі Microsoft Excel макрорекордер перетворює дії користувача у команди VBA [52–54].

У середовищі СУБД MySQL створення таблиць зазвичай виконується вручну за допомогою інструкції CREATE TABLE. Синтаксис цієї команди передбачає:

- вказання імені таблиці;
- визначення стовпців та типів даних;
- зазначення первинного ключа (PRIMARY KEY);
- встановлення зовнішніх ключів (FOREIGN KEY) та поведінки при оновленні чи видаленні пов'язаних записів (ON UPDATE, ON DELETE).

Наведемо приклад створення таблиці register (рис. 4.16), яка використовується для зберігання інформації про реєстрацію клієнтів:

sql

Копіювати

Редагувати

```
CREATE TABLE register (
    id_specialty INT UNSIGNED,
    tel_client VARCHAR(20),
    register_date DATE,
    date_to_come TIMESTAMP,
    PRIMARY KEY (id_specialty, date_to_come),
    FOREIGN KEY (id_specialty)
    REFERENCES specialty(id_specialty)
```

```

ON DELETE CASCADE
ON UPDATE CASCADE,
FOREIGN KEY (tel_client)
REFERENCES client(tel)
ON DELETE CASCADE
ON UPDATE CASCADE
);

```

У цьому прикладі:

- `id_specialty` – зовнішній ключ, який посилається на первинний ключ таблиці `specialty`;
- `tel_client` – зовнішній ключ, пов’язаний із таблицею `client`;
- обрано складений первинний ключ (`id_specialty`, `date_to_come`), що гарантує унікальність записів;
- використано механізми каскадного оновлення та видалення для збереження цілісності пов’язаних таблиць.

Такий підхід до створення таблиць забезпечує не лише структурованість даних, але й підтримку надійних зв’язків між сутностями, що є основою реляційної моделі баз даних.

Field	Type	Null	Key	Default	Extra
<code>id_specialty</code>	<code>int(10) unsigned</code>	NO	PRI	NULL	
<code>tel_client</code>	<code>varchar(20)</code>	YES	MUL	NULL	
<code>register_date</code>	<code>date</code>	YES		NULL	
<code>date_to_come</code>	<code>timestamp</code>	NO	PRI	NULL	

Рис. 4.16. Дані таблиці «Запис» (`register`)

У процесі створення таблиці в SQL спочатку вказується її назва після ключових слів `CREATE TABLE`. Далі, у круглих дужках, перелічуються всі стовпці (атрибути), які входять до таблиці. Важливо, що імена стовпців мають бути унікальними в межах однієї таблиці, оскільки повторення назв призведе до помилок при створенні.

Для кожного стовпця обов'язково вказується тип даних, який визначає, які саме значення можуть зберігатися у відповідному полі (наприклад, INT, VARCHAR, DATE, BOOLEAN тощо). Крім типу, зазначається, чи допускається наявність порожніх значень у полі за допомогою ключових слів NULL або NOT NULL. Якщо вказано NOT NULL, це означає, що поле обов'язкове для заповнення.

Також під час створення таблиці визначається первинний ключ (Primary Key) – один або кілька стовпців, значення яких однозначно ідентифікують кожен запис у таблиці. Первинний ключ гарантує унікальність і цілісність записів [55].

Окрім обов'язкових характеристик полів, мова SQL надає можливість задавати для стовпців значення за замовчуванням, що автоматично встановлюються у випадку, якщо користувач не вводить відповідного значення. Це реалізується за допомогою ключового слова DEFAULT, яке зазначається після типу даних у визначенні стовпця. Наприклад:

```
sql
```

Копіювати/Редагувати

```
status VARCHAR(20) DEFAULT 'новий'
```

Важливим аспектом у структурі реляційної бази даних є встановлення зв'язків між таблицями. Це досягається шляхом оголошення зовнішніх ключів (Foreign Keys). Зовнішній ключ у таблиці-підлеглий (child table) вказує на первинний ключ або унікальний стовпець у таблиці-господарі (parent table), забезпечуючи таким чином референційну цілісність даних.

Після ключового слова FOREIGN KEY зазначається перелік стовпців поточної таблиці, які утворюють зовнішній ключ. Ключове слово REFERENCES визначає таблицю, з якою встановлюється зв'язок, і перелік стовпців, на які посилається зовнішній ключ. Крім того, за допомогою ON DELETE і ON UPDATE можна налаштувати поведінку СУБД у разі видалення або оновлення записів у головній таблиці. Наприклад:

- ON DELETE CASCADE – автоматичне видалення пов'язаних записів у підлеглий таблиці;
- ON UPDATE CASCADE – автоматичне оновлення зовнішніх ключів відповідно до змін у головній таблиці;
- SET NULL, SET DEFAULT, RESTRICT – інші варіанти обробки.
- Практичне застосування в межах даного проєкту

У межах розробки інформаційної системи для салону краси всі таблиці бази даних були створені вручну за допомогою SQL-операторів. Такий підхід забезпечив повний контроль над структурою бази, визначенням типів полів, ключів та зв'язків між сутностями. Зокрема, було реалізовано як первинні ключі, що гарантують унікальність записів, так і зовнішні ключі, що встановлюють логічні зв'язки між таблицями, забезпечуючи узгодженість даних.

Фрагмент SQL-коду, який ілюструє процес створення таблиць та організацію міжтабличних зв'язків, наведено у Додатку Д кваліфікаційної роботи. Він демонструє застосування типових конструкцій SQL при побудові бази даних у середовищі MySQL.

4.3 Характеристика Python Django як інструменту розробки вебплатформи

Python Django – це високорівневий вебфреймворк з відкритим вихідним кодом, який широко використовується для створення динамічних, масштабованих та безпечних вебдодатків. Він базується на мові програмування Python і дотримується парадигм швидкої розробки (rapid development) та чистої архітектури програмного забезпечення. Серед ключових характеристик Django як інструменту для розробки вебплатформи можна виділити наступні [56-57]:

1. Повноцінний стек функціональності.

Django надає розробнику повний набір інструментів для розробки вебзастосунку «з коробки»:

- маршрутизація URL (*URL Dispatcher*);
- система шаблонів (*Template Engine*);
- ORM (*Object-Relational Mapping*);
- інтеграція з базами даних;
- обробка форм і валідація даних;
- автентифікація користувачів;
- підтримка сесій та cookie;
- адмін-панель тощо.

Це дозволяє уникнути використання сторонніх бібліотек на базовому рівні, що пришвидшує розробку та зменшує технічний борг.

2. Принцип «Batteries Included».

Django реалізує концепцію «усе необхідне в комплекті», надаючи вбудовані інструменти для:

- кешування;
- відправки електронних листів;
- роботи з медіафайлами;
- локалізації;
- обробки помилок;
- REST-підтримки (за допомогою *Django REST framework*).

Це дозволяє команді зосередитись на бізнес-логіці, мінімізуючи час на інфраструктурні налаштування.

3. Безпека на першому місці.

Фреймворк включає вбудований захист від найпоширеніших вебзагроз:

- SQL-ін'єкції;
- міжсайтові скриптові атаки (XSS);
- міжсайтові підробки запитів (CSRF);
- клікджекинг.

Розробники Django слідують принципу «Secure by default», що означає безпечну поведінку компонентів за замовчуванням.

Django однаково добре підходить для малих проєктів і великих платформ, завдяки модульній структурі, підтримці горизонтального масштабування, кешування та розділення логіки за принципом MVC (у *Django* – *MTV: Model-Template-View*). При правильному налаштуванні його можна ефективно використовувати у високонавантажених середовищах (наприклад, Instagram, Pinterest, Disqus – великі платформи, побудовані на Django).

Django ORM забезпечує високий рівень абстракції у роботі з базами даних, дозволяючи писати запити за допомогою об'єктно-орієнтованого синтаксису замість сирого SQL. Це:

- спрощує перенесення на інші СУБД;
- підвищує безпеку;
- покращує читабельність коду;

Однією з найсильніших сторін Django є його *admin interface*, який автоматично генерується на основі моделей. Цей інструмент дозволяє не тільки керувати контентом, а й реалізувати систему прав доступу, історію змін, кастомні дії над об'єктами тощо- без додаткового кодування.

Django добре інтегрується з іншими технологіями та сервісами:

- Redis, Celery (асинхронні задачі);
- REST API (через Django REST framework);
- GraphQL (через Graphene);
- фронтенд-фреймворки як React або Vue.js;
- сторонні системи автентифікації (OAuth, LDAP, JWT);

Також можлива розробка багатоарендних (multi-tenant) рішень, SaaS-платформ, мікросервісних архітектур.

Django має вбудовану систему для юніт-тестування, включаючи підтримку модульного, інтеграційного та функціонального тестування. Це значно полегшує автоматизацію перевірки якості коду та впровадження практик CI/CD у проєкті.

Фреймворк сприяє написанню лаконічного, повторно використовованого та підтримуваного коду:

- DRY (Don't Repeat Yourself) дозволяє уникнути дублювання логіки.
- KISS (Keep It Simple, Stupid) наголошує на важливості простоти та очевидності у проектуванні.

Django має одну з найактивніших спільнот у світі Python. Велика кількість навчальних матеріалів, готових рішень, плагінів та активних форумів (StackOverflow, Reddit, офіційна документація) робить процес навчання та розробки комфортним як для новачків, так і для досвідчених інженерів..

Таким чином, використання Django як основного інструменту для створення вебплатформи «MOV Hair Studio» є обґрунтованим вибором з кількох ключових причин. По-перше, Django забезпечує швидку розробку завдяки своїй філософії «Batteries included», що дозволяє швидко реалізувати функціонал онлайн-бронювання, управління клієнтами, збору зворотного зв'язку та адміністрування. По-друге, наявність автоматичної адміністративної панелі значно спрощує керування послугами, записами та користувачами без необхідності створення окремого інтерфейсу. По-третє, Django ORM дозволяє ефективно працювати з базою даних, що критично важливо для збереження історії записів та аналітики. Завдяки вбудованим механізмам безпеки, платформа буде захищена від поширених кіберзагроз, що особливо важливо для обробки персональних даних клієнтів. Масштабованість фреймворку дозволить легко розширювати платформу разом із розвитком бізнесу, включаючи додавання нових філій чи розширення функціоналу. Django також підтримує інтеграцію з зовнішніми сервісами – наприклад, системами онлайн-платежів, email-розсилками та SMS-сповіщеннями. Завдяки активній спільноті та документації, розробка і супровід проєкту будуть ефективними навіть у довгостроковій перспективі. Високий рівень повторного використання коду та дотримання принципів DRY/KISS сприятимуть створенню підтримуваного та масштабованого рішення. Зрештою, Django ідеально поєднує гнучкість, безпеку та швидкість розробки – саме те, що потрібно для якісного цифрового сервісу у сфері краси.

4.4 Розробка back-end частини вебплатформи

У сучасному веброзробленні back-end частина відіграє ключову роль у забезпеченні функціональності, безпеки, обробки запитів користувача, взаємодії з базами даних і логіки бізнес-процесів. Саме серверна логіка є тією невидимою, але критично важливою основою, яка забезпечує узгоджену та безперебійну роботу всієї платформи. Незалежно від складності вебплатформи – від простої сторінки до багатофункціональної платформи – якісний back-end є необхідною умовою ефективного функціонування сервісу, особливо якщо мова йде про обробку персональних даних, бронювання, взаємодію з зовнішніми API або створення адміністративних інтерфейсів.

Одним із найбільш популярних рішень для реалізації back-end у сучасних проєктах є фреймворк Django, написаний мовою програмування Python. Django забезпечує високу швидкість розробки, масштабованість, безпеку та має чітку архітектуру, що базується на шаблоні MTV (Model-Template-View). Завдяки цій архітектурі логіка проєкту поділяється на незалежні компоненти, що значно спрощує процес розробки, тестування та супроводу. Django надає велику кількість вбудованих інструментів, серед яких система керування користувачами, ORM для взаємодії з базами даних, система шаблонів, механізми автентифікації, маршрутизації, кешування та багато іншого.

Особливо важливими для побудови якісного back-end є вбудовані засоби захисту від загроз, таких як CSRF, XSS, SQL-ін'єкції, що у випадку роботи з персональними даними клієнтів та фінансовою інформацією є беззаперечною перевагою. Ще однією ключовою особливістю Django є автоматично генерований адміністративний інтерфейс, який дозволяє розробникам швидко створювати потужні панелі управління даними, що критично важливо для адміністрації бізнесу – як-от салонів краси, які постійно працюють із клієнтськими записами, послугами та історією взаємодії.

Окрім функціональності, Django підтримує гнучку інтеграцію з іншими сервісами та бібліотеками, що дає змогу реалізовувати надбудови – від email-розсилок до підключення платіжних систем, чатів, SMS-нотифікацій або аналітичних інструментів. Завдяки активній спільноті розробників, проєкти, побудовані на Django, мають широку базу підтримки, що мінімізує ризики зупинок розвитку внаслідок технічних складнощів [58-59].

Таким чином, використання Django як інструменту для створення back-end частини вебплатформи є стратегічно вдалим рішенням, що дозволяє швидко та якісно реалізувати як базовий функціонал, так і масштабовані бізнес-логіки. Такий підхід дозволяє сконцентрувати зусилля на розвитку унікальної цінності продукту, залишаючи рутинні аспекти на опрацювання перевіреними інструментами фреймворку.

Створення як back-end так і front-end частини вебплатформи здійснюється з використання середовища PyCharm Community (рис. 4.17).

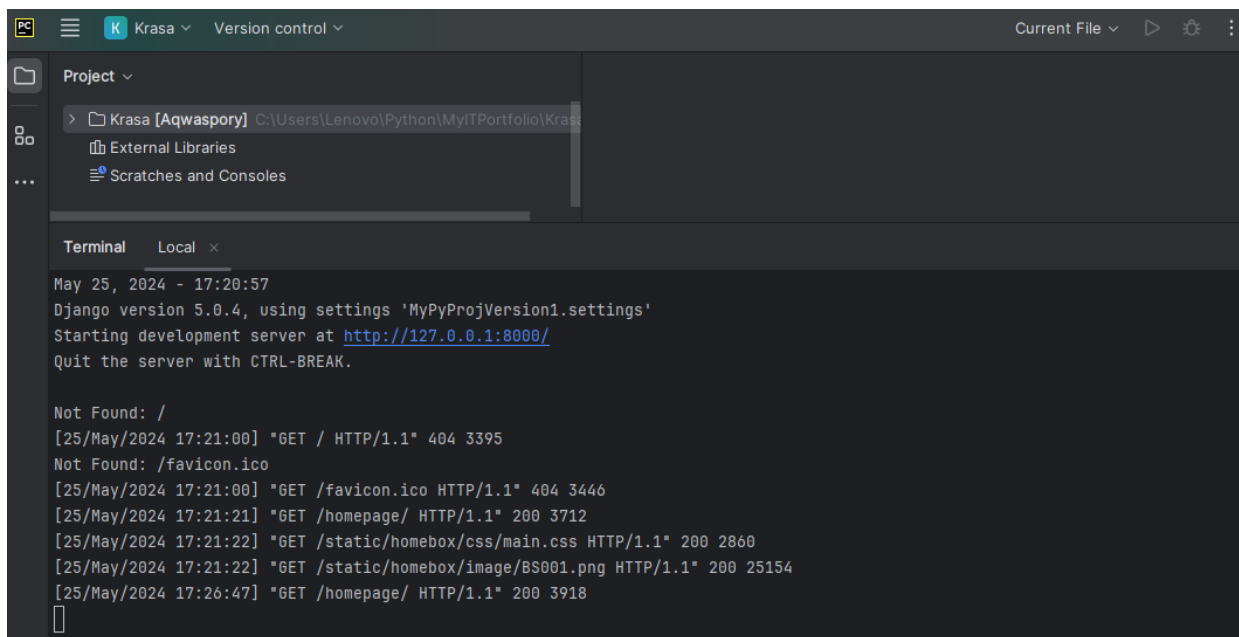


Рис. 4.17. Середовище PyCharm Community

Архітектура back-end частини вебплатформи представлена на рис. 4.18.

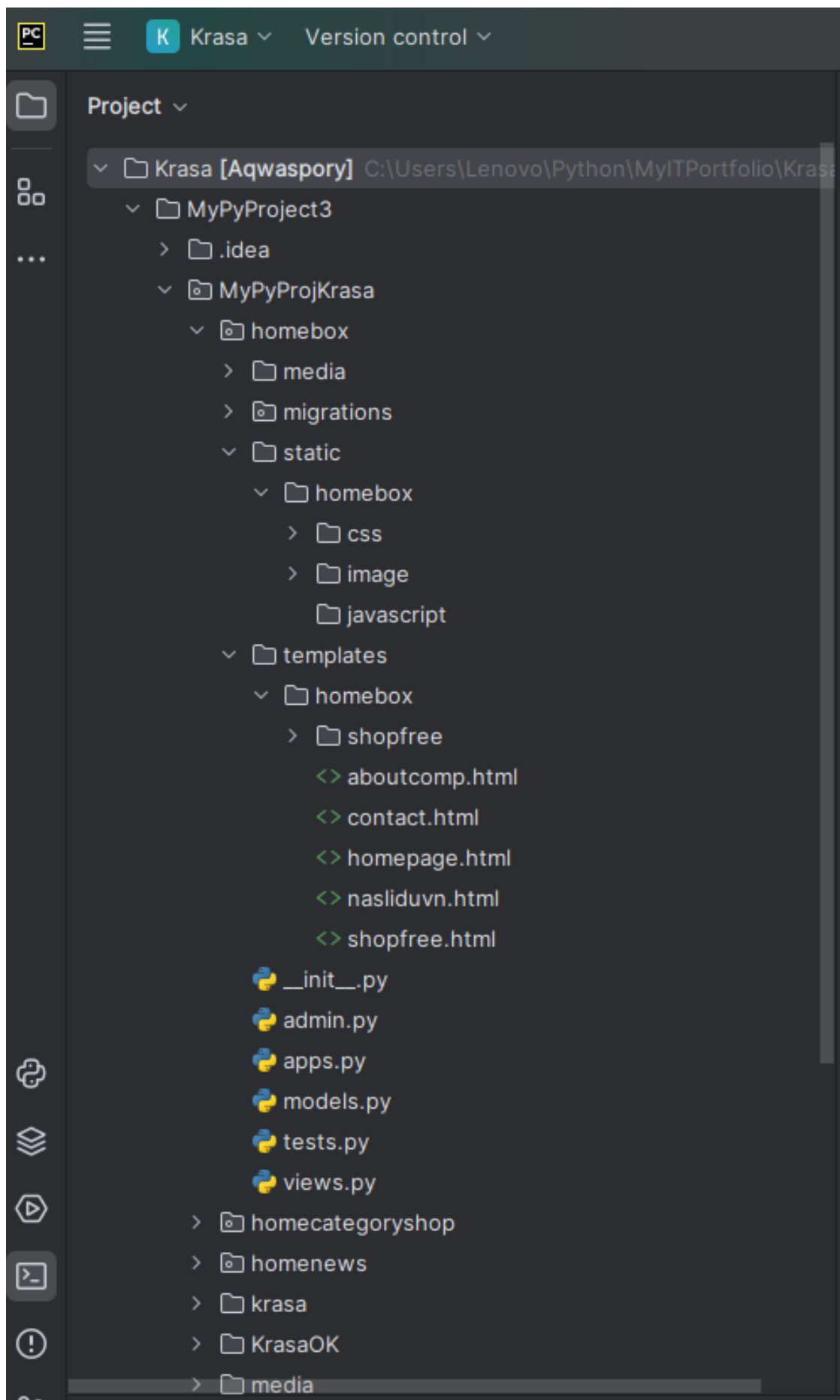


Рис. 4.18. Архітектура back-end частини вебплатформи

Фрагмент коду back-end частини вебплатформи наведено на рис. 4.19.

```
1 from django.http import HttpResponse
2 from django.shortcuts import render
3
4 def index(request):
5     return render(request, "homebox/homepage.html")
6
7 def aboutc(request):
8     return render(request, "homebox/aboutcomp.html")
9
10 def contact(request):
11     return render(request, "homebox/contact.html")
12
13 def magazyn(request):
14     return render(request, "homebox/shopfree.html")
15
16 def summ(request):
17     return render(request, "homebox/shopfree/summer/sum.html")
18
```

Рис. 4.19. Фрагмент коду back-end частини вебплатформи

4.5 Розробка front-end частини вебплатформи

Front-end частина вебплатформи є ключовою точкою взаємодії користувача з системою, оскільки саме вона відповідає за зовнішній вигляд, доступність функцій та зручність використання. Від якості реалізації інтерфейсу напряду залежить загальне враження клієнта про сервіс, рівень його задоволення, частота повторних взаємодій та ефективність бізнесу загалом. Для платформи, орієнтованої на надання послуг у сфері краси, важливо забезпечити інтуїтивний та візуально привабливий інтерфейс, що стимулюватиме клієнтів до активного використання функціоналу – зокрема, онлайн-бронювання, перегляду прайс-листа, ознайомлення з портфоліо фахівців, залишення відгуків тощо.

У сучасній практиці розробки вебінтерфейсів головним завданням є створення адаптивного, динамічного, естетично привабливого інтерфейсу, який однаково добре відображається на пристроях будь-якого типу – від мобільних телефонів до великих екранів. Особливу увагу приділяють

респонсивній верстці, яка автоматично адаптує вміст під розмір вікна користувача. У поєднанні з mobile-first дизайном, такий підхід забезпечує зручну навігацію та швидкий доступ до основних функцій саме з мобільних пристроїв, що є критично важливим для сфери послуг, де більшість користувачів здійснюють бронювання «на ходу» [60].

У рамках обраної технологічної бази, фронтенд частина платформи розробляється на основі Django, з використанням його вбудованої системи шаблонів. Це дозволяє інтегрувати динамічний HTML-контент безпосередньо з серверної логіки, що забезпечує цілісність архітектури проєкту. Django-шаблони підтримують змінні, умовні оператори, цикли, включення блоків, а також легко розширюються через створення власних фільтрів та тегів. Такий підхід особливо ефективний у проєктах, де front-end і back-end реалізуються в межах одного фреймворку, забезпечуючи узгодженість, менші витрати на підтримку, зменшення кількості проміжних API-запитів і пришвидшення рендерингу сторінок [61].

Для стилізації інтерфейсу використовуються сучасні CSS-фреймворки, такі як Tailwind CSS, що забезпечує високий рівень контролю над оформленням елементів і підтримує створення компонентоорієнтованої структури верстки. Використання Tailwind дозволяє створювати гнучкі дизайн-рішення з мінімальним обсягом кастомного коду, що прискорює розробку і полегшує внесення змін у майбутньому. Також можуть застосовуватись інтерактивні компоненти на основі JavaScript або бібліотек, таких як Alpine.js або HTMX, які добре інтегруються з Django і не потребують окремого API-шару, як у випадку з React або Vue.

Таким чином, розробка front-end частини вебплатформи із використанням шаблонної системи Django, у поєднанні з гнучкими CSS-фреймворками, дозволяє досягти високої швидкості створення, стилістичної узгодженості та ефективної інтеграції з серверною логікою. Це є надзвичайно важливим для динамічного розвитку малого бізнесу в цифровому просторі.

Наведений нижче шаблон наслідування дозволить створювати нові сторінки, додаючи лише унікальний контент відповідно до кожного конкретного вмісту. Такий підхід полегшить управління та підтримку даної вебплатформи, оскільки основна структура та стилі залишаються незмінними, а відповідно зміні підлягає лише те, що відображається на кожній окремій сторінці.

```
{% load static %}
```

```
<!doctype html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
    minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>{% block title %}{% endblock %}</title>
  <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="{% static 'homebox/css/main.css' %}">
  <link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.8.2/css/all.css">
</head>
<body>
  <aside>
    
    <h3>Пошук по сайту</h3>
    <ul>
      <li><a href="{% url 'homefd' %}"><i class="fas fa-home"></i>Головна сторінка</a></li>
      <li><a href="{% url 'shopfr' %}"><i class="fas fa-regular fa-bars"></i>Наша
команда</a></li>
      <li><a href="{% url 'homenews' %}"><i class="fas fa-light fa-newspaper"></i>Корисна
інформація для догляду за волоссям</a></li>
      <li><a href="{% url 'shop' %}"><i class="fas fa-regular fa-store"></i>Інтернет-магазин
салону</a></li>
      <li><a href="{% url 'contfd' %}"><i class="fas fa-paper-plane"></i>Контакти</a></li>
    </ul>
    <li><a href="{% url 'create' %}"><button class="btn btn-dark"><i class="fas fa-plus-
circle"></i>Записатися до майстра</button></a></li>
  </ul>
</aside>
<main>
  {% block content %}
  {% endblock %}
</main>
```

</body>
</html>

На рис. 4.20-4.23 наведено приклади сторінок вебплатформи для салону краси «MOV Hair Studio».



Рис. 4.20. Вигляд головної сторінки вебплатформи

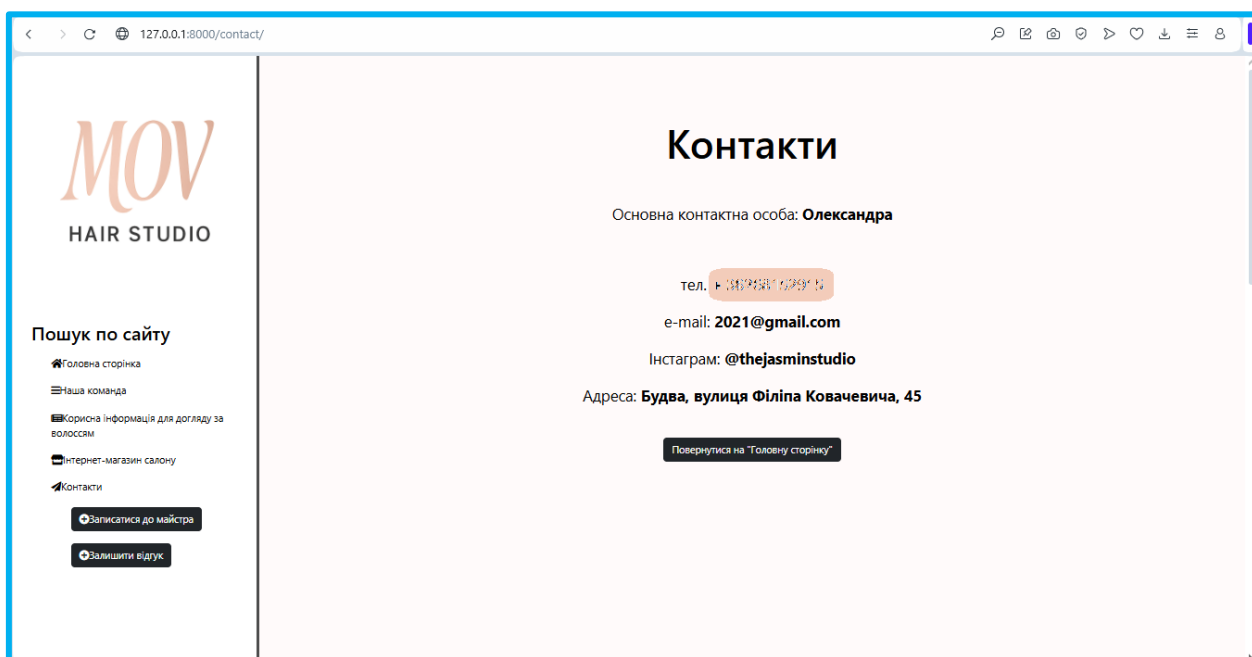


Рис. 4.21. Вигляд сторінки «Контакти»

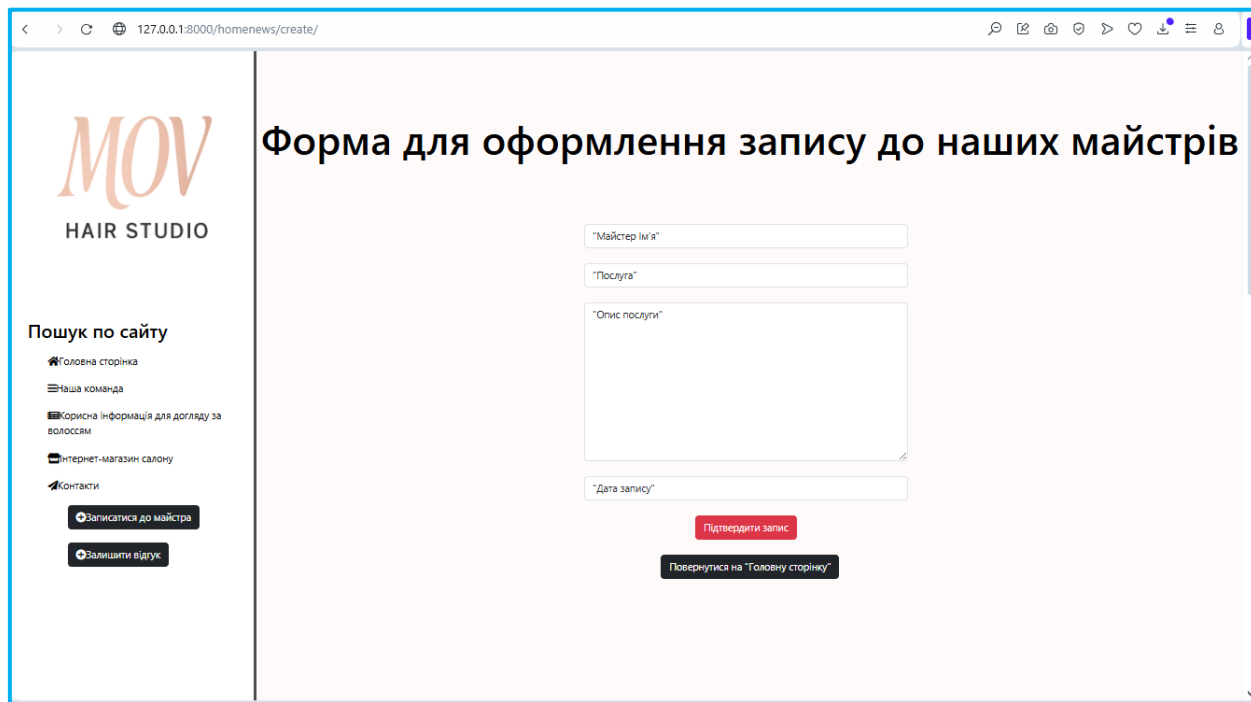


Рис. 4.22. Вигляд сторінки «Запису до майстра»

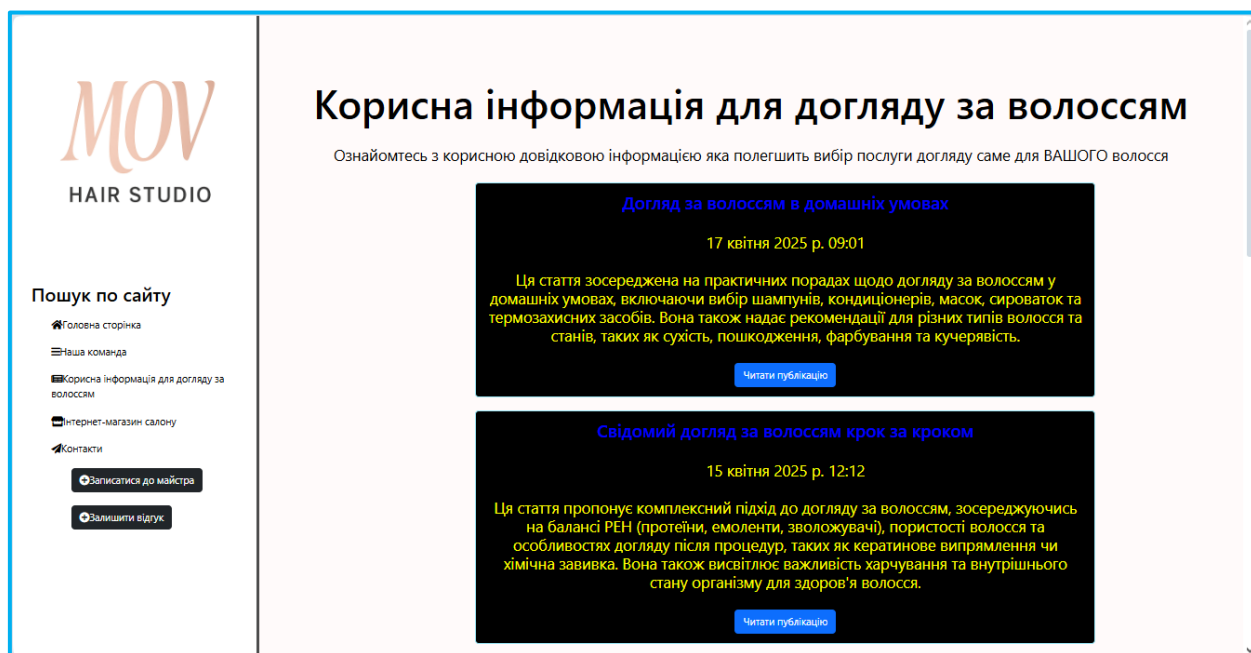


Рис. 4.22. Вигляд сторінки «Корисна інформація для догляду за волоссям»

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було досягнуто поставлену мету та послідовно виконано всі поставлені задачі:

1. Дослідження програмних рішень у сфері керування салонами краси дозволило виявити сильні сторони сучасних платформ (автоматизація запису, CRM-функціонал) та їхні слабкі місця (обмежена гнучкість, недостатня аналітика зворотного зв'язку). Це обґрунтувало необхідність створення власної вебплатформи із розширеними можливостями.
2. Аналіз літературних та інформаційних джерел забезпечив глибоке розуміння актуальних технологій розробки вебсистем, методів аналізу клієнтських відгуків за допомогою NLP, а також дозволив здійснити SWOT-аналіз проєкту та ідентифікувати ключових зацікавлених сторін – власників бізнесу, працівників салону, клієнтів та розробників.
3. Сформульовано технічне завдання у вигляді паспорта проєкту, розроблено концептуальну модель інформаційної системи та визначено математичну постановку задачі, що охоплює моделювання процесів бронювання, управління клієнтськими даними та аналізу зворотного зв'язку.
4. Обґрунтовано вибір методології Scrum для управління проєктом, сформовано організаційну структуру команди, розроблено продукт-беклог та сплановано серію спринтів із використанням MS Project для контролю виконання задач.
5. Ідентифіковано основні ризики розробки та впровадження вебплатформи, серед яких – ризики кібератак, технологічного застарівання та обмежень бюджету. Для кожного ризику розроблено заходи управління та пом'якшення наслідків.
6. Розроблено алгоритм роботи сайту, концептуальну і даталогічну моделі бази даних, а також визначено особливості використання Python Django

як базового фреймворку для створення стабільної та масштабованої архітектури платформи.

7. Реалізовано back-end вебплатформи на базі Django із забезпеченням функцій бронювання, управління профілями клієнтів та адміністрування салону, розроблено front-end інтерфейс із адаптивним дизайном та проведено інтеграцію всіх компонентів для забезпечення повної функціональності системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Kotenko N., Zhyrova T., Chybaievskyi V., Desiatko A. Doslidzhennja osnovnyh tendencij suchasnoi' rozrobky veb-sajtiv. Elektronne fahove naukove vydannja «Kiberbezpeka: osvita, nauka, tehnika», 2019, 1(5), 6–15. <https://doi.org/10.28925/2663-4023.2019.5.615>
2. Могильська М. Модель оцінювання надійності веб-сайтів. URL: <http://dspace.wunu.edu.ua/bitstream/316497/47171/1/МОГИЛЬСЬКА%20Марія%20Богданівна.pdf>
3. Димитров А. Розробка веб-сайту з функціональною адміністративною панеллю та інтеграцією платіжних систем : магістерська дис. : 171 Електроніка. - Київ, 2023. - 108 с.
4. Аронов А.О. Аналіз підходів до створення сайтів та вимог щодо їх інформаційного наповнення. Слово науковця. Зв'язок, №2, 2018, с. 40-44.
5. Гніденко І.А., Воробйов І.Є. Аналіз сучасних програмних продуктів для створення web-сайтів, їх переваги та недоліки. Проблеми інформатизації та управління, 1(61), 2019.
6. Топ популярних веб-сайтів. Особливості успіху. URL: <https://mas-agency.com.ua/blog/article/?id=2>
7. 10 найкращих конструкторів веб-сайтів зі штучним інтелектом (травень 2024 р.) URL: <https://www.unite.ai/uk/best-ai-website-builders/>
8. Етапи створення веб сайтів. URL: <https://webtune.com.ua/statti/web-rozrobka/etapy-stvorennya-veb-sajtiv/>
9. Складні веб-проекти. URL: <https://webcase.com.ua/uk/>
10. Як обрати найкращу платформу для створення сайту для вашого бізнесу. URL: <https://krakatets.pro/iak-obraty-najkrashchu-platformu-dlia-stvorennia-sajtu-dlia-vashoho-biznesu.html>
11. 7 речей, які слід враховувати під час створення веб-сайту з портфоліо. URL: <https://webnus.net/uk/things-to-consider-when-building-a-portfolio-website/>

12. Етапи розробки сайту: як пройти шлях від ідеї до релізу.
URL: <https://wezom.com.ua/ua/blog/etapy-razrabotki-sajta>
13. Як створити веб-сайт для свого бізнесу. URL: <https://www.doola.com/uk/blog/how-to-start-a-website-for-your-business/>
14. Low-code та No-code платформи - це майбутнє створення веб-сайтів.
URL: <https://abordazh.net/uk/uproshchonnaye-sozdaniya-veb-saytov/>
15. Круціцький, В. Я., & Сугоняк, І. І. (2023). Оцінка ефективності використання інструментів NLP та систем AI для аналізу рекламних оголошень у системах обміну інтернет-рекламою. Технічна інженерія, (1(91), 161–165. [https://doi.org/10.26642/ten-2023-1\(91\)-161-165](https://doi.org/10.26642/ten-2023-1(91)-161-165).
16. Izadi M., Ahmadabadi M.N. On the Evaluation of NLP-based Models for Software Engineering. IEEE/ACM 1st International Workshop on Natural Language-Based Software Engineering (NLBSE), Pittsburgh, PA, USA, 2022, pp. 48-50, <https://doi.org/10.1145/3528588.3528665>
17. Doshi J. Chatbot User Interface for Customer Relationship Management using NLP models. International Conference on Artificial Intelligence and Machine Vision (AIMV), Gandhinagar, India, 2021, pp. 1-4, <https://doi.org/10.1109/AIMV53313.2021.9670914>
18. Warusawithana S.P., Perera N.N., Weerasinghe R.L., Hindakaraldeniya T.M., Ganegoda G.U. Layout Aware Resume Parsing Using NLP and Rule-based Techniques. 8th International Conference on Information Technology Research (ICITR), Colombo, Sri Lanka, 2023, pp. 1-5, <https://doi.org/10.1109/ICITR61062.2023.10382773>
19. Mishra P., Ninawe K., Dhawle K., Prasad S., Band A., Dubey P. NLP in Business Analytics: Generating Insights from Textual Data Using NLP Models. IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN), Indore, India, 2024, pp. 1472-1478, <https://doi.org/10.1109/CICN63059.2024.10847530>
20. Mishra P., Bankar S., Madankar M. Text Processing Hub: NLP Applications. International Conference on Innovations and Challenges in Emerging

- Technologies (ICICET), Nagpur, India, 2024, pp. 1-5, <https://doi.org/10.1109/ICICET59348.2024.10616266>
21. Amantova-Salmane L. Stakeholder concerns and management principles in project management. *Journal of Regional Economic and Social Development*, 2024, 16, 21-32. <https://doi.org/10.17770/jresd2024vol16.8294>
 22. Eskola A., Hundal S. Organizational Storytelling Translated to Successful Business. *Asian Journal of Business and Management*, 2021, 9 (2), <https://doi.org/10.24203/ajbm.v9i2.6408>
 23. Sanyaolu T.O., Adeleke A.G., Efunniyi C.P., Akwawa L.A., Azubuko C.F. Stakeholder management in IT development projects: Balancing expectations and deliverables. *International Journal of Management & Entrepreneurship Research* P-ISSN: 2664- 3588, E-ISSN: 2664-3596 Volume 5, Issue 12, P.No.1239-1255, December 2023, <https://doi.org/10.51594/ijmer.v5i12.1535>.
 24. Suvvari S.K., Saxena V.D. Stakeholder Management in Projects: Strategies for Effective Communication. *Innovative Research Thoughts*, 9(5):188-201, <https://10.36676/irt.v9.i5.1479>
 25. Romero de Miguel A., Pérez-Ezcurdia M.A., Gimena-Ramos, F.N. The Project Book: A Toolkit for Stakeholder Management in International Development Projects. *Revista Iberoamericana De Estudios De Desarrollo = Iberoamerican Journal of Development Studies*, 2024, 1-36. https://doi.org/10.26754/ojs_ried/ijds.10140
 26. Meskauskas Z., Jasinevicius R., Kazanavicius E., Petrauskas V. XAI-Based Fuzzy SWOT Maps for Analysis of Complex Systems. *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Glasgow, UK, 2020, pp. 1-8, <https://doi.org/10.1109/FUZZ48607.2020.9177792>
 27. Zhibin X., Jiangsheng Z., Lvqing Y. Innovation Strategy and Development Plan of E-Sign Wechat System Based on SWOT-AHP Analysis. *IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, 2020, pp. 1640-1644, <https://doi.org/10.1109/ITNEC48623.2020.9084953>

28. Larsson D., Chandima Ratnayake R.M. SWOT Analysis for Implementation of Lean-Agile Mindset: A Case Study from an ETO Organisation. IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), Kuala Lumpur, Malaysia, 2022, pp. 1107-1113, <https://doi.org/10.1109/IEEM55944.2022.9989718>.
29. Ebrahimi M. SWOT Analysis for 2025 Success. Cambridge Open Engage, 2025. <https://doi.org/10.33774/coe-2025-h44tr>
30. Qu Ru. Strategy analysis and optimization of SWOT matrix model in science and technology development. Applied Mathematics and Nonlinear Sciences, 2025, vol. 10, no. 1, <https://doi.org/10.2478/amns-2025-0371>
31. Yun T. SWOT Analysis of Game Science Company. Advances in Economics, Management and Political Sciences, 2025, 149, 129-134, <https://doi.org/10.54254/2754-1169/2024.19243>
32. Kolomiets A., Miroshnychenko I., Ziuziun V., Datsenko N., Kmytiuk T. Development of Project Management Models for Information Systems to Improve Website SEO Metrics. XI International Scientific Conference «Information Technology and Implementation» (IT&I 2024). CEUR Workshop Proceedings, 2024, Vol-3909, pp. 334-345. https://ceur-ws.org/Vol-3909/Paper_27.pdf
33. Toxanov S., Sharipova S., Biloshchytskyi A., Abzhanova D., Bakytkereily B. Analysis of the impact of sharding on the scalability and efficiency of blockchain technologies for the creation of information-analytical systems for environmental monitoring of emissions into the environment. Scientific Journal of Astana IT University, 2024, 18, 66–74. <https://doi.org/10.37943/18VIFS4010>
34. Neftissov A., Biloshchytskyi A., Toxanov S., Ordabayev S, Kuchansky O., Andrashko Y., Vatskel V. Mathematical, software and hardware support of the conceptual model of the information system of precision agriculture. Scientific Journal of Astana IT University, 2023, 15(15), 55–70. <https://doi.org/10.37943/15TKFW1223>

35. Borgida A. Conceptual Modeling of Information Systems. In: Brodie, M.L., Mylopoulos, J. (eds) On Knowledge Base Management Systems. Topics in Information Systems. Springer, New York, NY. https://doi.org/10.1007/978-1-4612-4980-1_36
36. Wand Y., Weber R. Research Commentary: Information Systems and Conceptual Modeling. A Research Agenda. Information Systems Research, 13(4), 363-376, <https://doi.org/10.1287/isre.13.4.363.69>
37. Kononenko I., Sushko H. Mathematical model of software development project team composition optimization with fuzzy initial data. Radioelectronic and computer systems, 2021, no. 3, pp. 149-159. <https://doi.org/10.32620/reks.2021.3.12>.
38. Mihailides G., Young W. Is Agile Scrum Improving IT Project Outcomes?. International Journal of Management Science and Business Administration, 2024, 10(2), 41-55, <https://doi.org/10.18775/ijmsba.1849-5664-5419.2014.102.1004>
39. Agile Practice Guide: Paperback. USA, Project Management Institute, 2017, 210 p.
40. Швабер К., Сазерленд Дж. Посібник зі Скраму (Повний навчальний посібник зі Скраму: правила гри). Режим доступу: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Ukrainian.pdf>
41. Кон Майкл. Оцінювання і планування в Agile / пер. з англ. Г. Якубовська. Харків : Вид-во «Ранок» : Фабула, 2019, 336 с.
42. Рябчиков О.М., Ганущак-Єфіменко Л.М. Integration of six sigma dmaic methods for effective middle-level team management in IT projects of SCRUM methodology. Journal of Strategic Economic Research, 2024, (3), 150–162, <https://doi.org/10.30857/2786-5398.2024.3.15>
43. Dong X. Best Practices in Project Risk Management. Transactions on Economics, Business and Management Research, 2024, 11, 511-518. <https://doi.org/10.62051/qkf9h233>

44. Tak A., Sunil Chahal S.C. Risk Management in Agile AI/ML Projects: Identifying and Mitigating Data and Model Risks. *Journal of Technology and Systems*, 2024, 6(3), 1–18. <https://doi.org/10.47941/jts.1824>
45. Bisikirskienė L., Čeponienė L., Vilutis G., Nečionytė A. Software Project Risk Identification Based on Scrum Artifacts. *Applied Sciences*, 2025, 15(2), 824. <https://doi.org/10.3390/app15020824>
46. Bahaaudeen A., Nazneen A., Haque R. Risk and sustainability in project management with special reference to the IT sector. *International Journal of Innovative Research and Scientific Studies*, 2025, 8(2), 4219–4231. <https://doi.org/10.53894/ijirss.v8i2.6272>
47. Goman M. How to Improve Risk Management in IT Frameworks. 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), 2021, Riga, Latvia, 2021, pp. 1-6, <https://doi.org/10.1109/ITMS52826.2021.9615327>
48. Pa N.C., Anthony B. A model of mitigating risk for IT organisations. 4th International Conference on Software Engineering and Computer Systems (ICSECS), Kuantan, Malaysia, 2015, pp. 49-54, <https://doi.org/10.1109/ICSECS.2015.7333082>
49. Тімінський О.Г., Коломієць А.С. Управління ризиками та можливостями проекту [Текст]: методичні вказівки до виконання практичних, лабораторних робіт та самостійної роботи для студентів освітньої програми «Управління проектами» спеціальності 122 «Комп'ютерні науки» для денної і заочної форм навчання / Тімінський О.Г., Коломієць А.С. – К. : КНУ імені Тараса Шевченка, 2021. – 40 с.
50. Eessaar E., Voronova J. Using SQL Queries to Evaluate the Design of SQL Databases. In: Elleithy, K., Sobh, T. (eds) *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering. Lecture Notes in Electrical Engineering*, 2013, vol 312. Springer, Cham. https://doi.org/10.1007/978-3-319-06764-3_23

51. Guo Y., Li G., Hu R. et al. In-database query optimization on SQL with ML predicates. *The VLDB Journal*, 2025 34, 12, <https://doi.org/10.1007/s00778-024-00888-3>
52. Wei Z. The design of library database management system based on MySQL. *Applied and Computational Engineering*, 2024, 38, 41-50, <https://doi.org/10.54254/2755-2721/38/20230527>
53. Wahyudi J., Asbari M., Sasono I., Pramono T., Novitasari D. Database Management Education in MYSQL. *Edumaspul: Jurnal Pendidikan*, 2022, 6(2), 2413-2417, <https://doi.org/10.33487/edumaspul.v6i2.4570>
54. Valentine T. Installing and Using the MySQL Database Server. In: *Database-Driven Web Development*. Apress, Berkeley, CA., 2023, https://doi.org/10.1007/978-1-4842-9792-6_8
55. Ravi Kumar Y.V., Samayam A.K., Miryala N.K.. MySQL Server Administration. In: *Mastering MySQL Administration*. Apress, Berkeley, CA, 2024. https://doi.org/10.1007/979-8-8688-0252-2_3
56. Django makes it easier to build better web apps more quickly and with less code. URL: <https://www.djangoproject.com>
57. Встановлення проекту на Python/Django у cPanel. URL: <https://hostiq.ua/wiki/ukr/cpanel-django/>
58. Chandiramani A., Singh P. Management of Django Web Development in Python, 2021, *J. Manage. Serv. Sci.*, vol. 1, no. 2, pp. 1–17, <https://doi.org/10.54060/JMSS/001.02.005>
59. Ashley D. Django. In: *Foundation Dynamic Web Pages with Python*. Apress, Berkeley, CA, 2020. https://doi.org/10.1007/978-1-4842-6339-6_6
60. Chen S., Ahmmed S., Lal K., Deming C. Django Web Development Framework: Powering the Modern Web. *American Journal of Trade and Policy*, 2020, 7(3), 99-106. <https://doi.org/10.18034/ajtp.v7i3.675>
61. Rubio D. Django Model Forms and Class Views. In: *Beginning Django*. Apress, Berkeley, CA, 2017, https://doi.org/10.1007/978-1-4842-2787-9_9

ДОДАТКИ

Додаток А

Таблиця А.1

Перелік User story проєкту

Код US	Назва та формулювання US	Acceptance Criteria
1	2	3
US01 (ФВ01)	<i>Реєстрація користувачів.</i> Як новий відвідувач, я хочу мати можливість зареєструватися на платформі, щоб створити обліковий запис і зберігати особисті дані, історію візитів і вподобання.	<i>Scenario:</i> Успішна реєстрація. <i>Given:</i> Я на сторінці реєстрації. <i>When:</i> Заповнюю поля та натискаю «Зареєструватися». <i>Then:</i> Створюється обліковий запис, відкривається особистий кабінет.
US02 (ФВ02)	<i>Авторизація користувачів.</i> Як зареєстрований користувач, я хочу входити до свого акаунта за допомогою email, пароля або соцмереж, щоб мати доступ до персональних функцій.	<i>Scenario:</i> Вхід у систему. <i>Given:</i> Я маю акаунт. <i>When:</i> Ввожу email/пароль або авторизуюсь через соцмережу. <i>Then:</i> Потрапляю до свого кабінету.
US03 (ФВ03)	<i>Онлайн-бронювання послуг.</i> Як клієнт, я хочу мати можливість бронювати послуги онлайн, щоб зручно планувати візити до салону.	<i>Scenario:</i> Бронювання послуги. <i>Given:</i> Я авторизований. <i>When:</i> Вибираю послугу, дату, майстра. <i>Then:</i> Отримую підтвердження бронювання.
US04 (ФВ04)	<i>Автоматичний підбір вільного часу.</i> Як клієнт, я хочу, щоб система автоматично пропонувала вільні часові слоти, щоб швидко знайти зручний час.	<i>Scenario:</i> Автоматичний вибір часу. <i>Given:</i> Я вибрав послугу і майстра. <i>When:</i> Перехожу до вибору часу. <i>Then:</i> Відображаються лише доступні слоти.
US05 (ФВ05)	<i>Персоналізовані рекомендації.</i> Як клієнт, я хочу отримувати персоналізовані поради на основі моєї історії, щоб швидко знаходити потрібні послуги.	<i>Scenario:</i> Рекомендації. <i>Given:</i> Історія відвідувань є в системі. <i>When:</i> Відкриваю сторінку послуг. <i>Then:</i> Отримую персоналізовані пропозиції.
US06 (ФВ06)	<i>Нагадування про прийоми.</i> Як клієнт, я хочу отримувати автоматичні нагадування про прийоми, щоб не пропускати записані послуги.	<i>Scenario:</i> Отримання нагадування. <i>Given:</i> У мене є активне бронювання. <i>When:</i> Наближається час прийому. <i>Then:</i> Я отримую нагадування через SMS або Email.
US07 (ФВ07)	<i>Збір відгуків після візиту.</i> Як клієнт, я хочу залишати відгук після отриманої послуги, щоб поділитися своїм досвідом.	<i>Scenario:</i> Надсилання відгуку. <i>Given:</i> Я завершив візит. <i>When:</i> Отримую запит на оцінку. <i>Then:</i> Маю можливість залишити відгук через форму.
US08 (ФВ08)	<i>Аналіз зворотного зв'язку.</i> Як адміністратор, я хочу бачити аналітику за відгуками клієнтів, щоб	<i>Scenario:</i> Перегляд аналітики. <i>Given:</i> Відгуки вже залишені. <i>When:</i> Я відкриваю розділ зворотного зв'язку.

Код US	Назва та формулювання US	Acceptance Criteria
1	2	3
	виявляти проблемні зони та покращувати якість сервісу.	<i>Then:</i> Бачу зведення з оцінками, трендами і коментарями.
US09 (ФВ09)	<i>Управління маркетингом.</i> Як адміністратор, я хочу створювати акції, купони та розсилки, щоб залучати нових клієнтів і утримувати постійних.	<i>Scenario:</i> Створення маркетингової кампанії. <i>Given:</i> Я в панелі адміністратора. <i>When:</i> Я створюю нову акцію чи купон. <i>Then:</i> Кампанія зберігається і стає доступною для клієнтів.
US10 (ФВ10)	<i>Управління розкладом майстрів.</i> Як майстер, я хочу самостійно оновлювати свій графік, щоб контролювати свій робочий час та запис клієнтів.	<i>Scenario:</i> Редагування розкладу. <i>Given:</i> Я авторизований майстер. <i>When:</i> Відкриваю календар і вношу зміни. <i>Then:</i> Новий розклад зберігається і синхронізується з системою бронювання.
US11 (ФВ11)	<i>Панель адміністратора.</i> Як адміністратор, я хочу мати доступ до керування персоналом, послугами, цінами та користувачами, щоб ефективно адмініструвати платформу.	<i>Scenario:</i> Використання панелі адміністратора. <i>Given:</i> Я авторизований адміністратор. <i>When:</i> Я переходжу до панелі. <i>Then:</i> Маю доступ до всіх керуючих функцій (майстри, послуги, користувачі тощо).
US12 (ФВ12)	<i>Генерація звітів.</i> Як адміністратор, я хочу генерувати звіти щодо записів, популярності послуг та доходів, щоб контролювати ефективність роботи салону.	<i>Scenario:</i> Формування звіту. <i>Given:</i> Є дані про записи та послуги. <i>When:</i> Я запускаю генерацію звіту. <i>Then:</i> Отримую PDF або таблицю з потрібною статистикою.
US13 (ФВ13)	<i>Адаптивний інтерфейс.</i> Як користувач, я хочу, щоб інтерфейс коректно працював на різних пристроях, щоб мати зручний доступ незалежно від платформи.	<i>Scenario:</i> Відображення інтерфейсу. <i>Given:</i> Я відкриваю платформу на смартфоні або планшеті. <i>When:</i> Завантажується інтерфейс. <i>Then:</i> Усі елементи коректно масштабуються і працюють.
US14 (ФВ14)	<i>Безпека та захист даних.</i> Як користувач, я хочу бути впевненим у безпеці моїх персональних даних, щоб не хвилюватися про їх витік або несанкціонований доступ.	<i>Scenario:</i> Захищене збереження даних. <i>Given:</i> Я передаю особисті дані (email, пароль). <i>When:</i> Вони обробляються та зберігаються. <i>Then:</i> Вся інформація захищена (SSL, хешування, контроль доступу).
US15 (ФВ15)	<i>Журнал подій.</i> Як адміністратор, я хочу бачити журнал подій (бронювання, скасування, зміни профілю), щоб мати змогу контролювати дії користувачів.	<i>Scenario:</i> Перегляд журналу. <i>Given:</i> Я відкриваю розділ журналу. <i>When:</i> Система показує список подій. <i>Then:</i> Я бачу хронологію дій користувачів у системі.

Беклог продукту проєкту

User Story	Задача / Task	Підзадача/ Subtask
1	2	3
US01	Т.01.1 Створити функціонал реєстрації користувачів з особистим кабінетом.	ST.01.11 Розробити інтерфейс реєстрації.
		ST.01.12 Зберігати користувача в базі даних.
		ST.01.13 Створювати особистий кабінет при реєстрації.
	Т.01.2 Забезпечити збереження даних із валідацією та хешуванням паролів.	ST.01.21 Додати валідацію email і пароля.
		ST.01.22 Застосувати хешування паролів.
		ST.01.23 Обробляти помилки при збереженні.
US02	Т.02.1 Реалізувати вхід через email і пароль.	ST.02.11 Створити форму входу.
		ST.02.12 Перевірити введені дані.
		ST.02.11 Створити сесію при успішному вході.
	Т.02.2 Інтегрувати входи через Google та Facebook.	ST.02.21 Налаштувати авторизацію через Google.
		ST.02.22 Налаштувати авторизацію через Facebook.
		ST.02.23 Прив'язати соцмережу до профілю користувача.
US03	Т.03.1 Розробити інтерфейс бронювання послуг.	ST.03.11 Створити форму вибору послуги.
		ST.03.12 Інтегрувати календар та вибір майстра.
		ST.03.13 Додати перевірку доступного часу.
	Т.03.2 Реалізувати збереження бронювань і підтвердження користувачу.	ST.03.22 Зберігати бронювання в базі.
		ST.03.22 Надсилати підтвердження користувачу.
		ST.03.22 Відображати бронювання у кабінеті.
US04	Т.04.1 Імплементувати автоматичний підбір вільного часу.	ST.04.11 Зібрати графік роботи майстрів.
		ST.04.12 Розрахувати тривалість послуг.
		ST.04.13 Сформулювати алгоритм підбору слотів.
	Т.04.2 Виводити список доступних слотів в UI.	ST.04.21 Розробити UI-компонент для показу слотів.
		ST.04.22 Інтегрувати з даними про наявність.
		ST.04.23 Оновлювати доступні слоти в реальному часі.
US05	Т.05.1 Розробити алгоритм персоналізованих рекомендацій.	ST.05.11 Аналізувати історію візитів клієнта.
		ST.05.12 Формувати релевантні послуги.
		ST.05.13 Оптимізувати швидкість обробки запитів.
	Т.05.2 Інтегрувати блок рекомендацій на головну сторінку.	ST.05.21 Створити окремий компонент рекомендацій.
		ST.05.22 Виводити персоналізований контент.
		ST.05.23 Адаптувати блок під мобільні пристрої.
US06	Т.06.1 Реалізувати автоматичні SMS/Email-нагадування.	ST.06.11 Інтегрувати SMS та Email сервіси.
		ST.06.12 Створити шаблони повідомлень.
		ST.06.13 Реалізувати механізм планування сповіщень.
	Т.06.2 Додати налаштування часу та типу нагадувань.	ST.06.21 Додати параметри у профілі користувача.
		ST.06.22 Зберігати налаштування в базі.

User Story	Задача / Task	Підзадача/ Subtask
1	2	3
		ST.06.23 Обробляти зміни у налаштуваннях.
US07	T.07.1 Автоматично запитувати відгук після візиту.	ST.07.11 Реалізувати тригер після візиту.
		ST.07.12 Надсилати запит на оцінку.
		ST.07.13 Забезпечити підтримку мультимедіальності.
		ST.07.21 Зберігати фідбек у базі.
	T.07.2 Зберігати та обробляти оцінки і коментарі.	ST.07.22 Валідувати вхідні дані.
		ST.07.23 Інтегрувати з аналітикою.
US08	T.08.1 Створити дашборд для аналізу відгуків.	ST.08.11 Реалізувати таблиці і графіки.
		ST.08.12 Показувати середні оцінки.
		ST.08.13 Побудувати агреговану статистику.
	T.08.2 Реалізувати фільтри за майстрами та періодами.	ST.08.21 Додати фільтри по датах.
		ST.08.22 Фільтрація за іменем майстра.
US09	T.09.1 Реалізувати створення маркетингових кампаній.	ST.08.22 Оновлення даних дашборду динамічно.
		ST.09.11 Створення форми для кампаній.
		ST.09.12 Налаштування строків та цілей.
	T.09.2 Інтегрувати промокоди та email-розсилки.	ST.09.13 Збереження інформації у базі.
		ST.09.21 Генерація унікальних промокодів.
		ST.09.22 Імпорт контактів для розсилок.
US10	T.10.1 Розробити інтерфейс для оновлення розкладу майстрів.	ST.09.23 Створення email-шаблонів.
		ST.10.11 Календар із подіями.
		ST.10.12 Додавання/редагування слотів.
	T.10.2 Синхронізувати розклад із системою бронювання.	ST.10.13 Валідація при збереженні.
		ST.10.21 Оновлення даних в обох модулях.
		ST.10.22 Перевірка конфліктів у розкладі.
US11	T.11.1 Створити панель адміністратора для керування персоналом.	ST.10.23 Відображення актуального статусу.
		ST.11.11 Інтерфейс додавання працівників.
		ST.11.12 Редагування ролей та прав.
	T.11.2 Додати керування послугами, графіками, цінами.	ST.11.13 Сторінка перегляду інформації про працівників.
		ST.11.21 CRUD-функціонал для послуг.
		ST.11.22 Інтерфейс налаштувань цін.
US12	T.12.1 Реалізувати систему звітів про записи та доходи.	ST.11.23 Зв'язування послуг із майстрами.
		ST.12.11 Фільтрація за періодом і категоріями.
		ST.12.12 Побудова графіків і таблиць.
	T.12.2 Забезпечити експорт звітів у PDF/Excel.	ST.12.13 Збереження попередніх звітів.
		ST.12.21 Інтеграція з бібліотекою генерації PDF.
		ST.12.22 Опція завантаження у форматі Excel.
US13	T.13.1 Адаптувати інтерфейс під мобільні пристрої.	ST.12.23 Форматування звітів перед експортом.
		ST.13.11 Мобільна верстка UI-компонентів.
		ST.13.12 Тестування на різних розширеннях.
	T.13.2 Перевірити кросбраузерну та кросплатформну сумісність.	ST.13.13 Оптимізація навігації.
		ST.13.21 Тестування у Chrome, Safari, Firefox.
		ST.13.22 Перевірка на iOS та Android.
US14	T.14.1 Реалізувати SSL-захист і хешування паролів.	ST.13.23 Усунення стилістичних помилок.
		ST.14.11 Увімкнення SSL на сервері.
		ST.14.12 Реалізація шифрування паролів.
		ST.14.13 Тестування захищених запитів.

User Story	Задача / Task	Підзадача/ Subtask
1	2	3
	Т.14.2 Налаштувати контроль доступу та аудит безпеки.	ST.14.21 Встановлення прав доступу до ресурсів.
		ST.14.22 Логування спроб несанкціонованого доступу.
		ST.14.23 Регулярна перевірка системи на вразливості.
US15	Т.15.1 Реалізувати логування дій користувачів у системі.	ST.15.11 Фіксація подій входу, бронювання, змін профілю.
		ST.15.12 Збереження логів у базі.
		ST.15.13 Встановлення обмежень на перегляд журналу.
	Т.15.2 Надати доступ адміністраторам до журналу подій.	ST.15.21 Створити UI для перегляду журналу.
		ST.15.22 Додати фільтрацію за типами подій.
		ST.15.23 Забезпечити лише адміністративний доступ.

Деталізація беклогу продукту до рівня робіт

Задача / Task (User Story)	Підзадача / Subtask	Роботи які потрібно виконати
1	2	3
T.01.1 (US01)	ST.01.11 Розробити інтерфейс реєстрації	1. Створити HTML-форму з полями: ім'я, email, пароль. 2. Застосувати клієнтську валідацію (JavaScript). 3. Забезпечити адаптивність для мобільних пристроїв. 4. Протестувати правильність відображення форми на різних екранах.
	ST.01.12 Зберігати користувача в базі даних	1. Налаштувати модель користувача у базі даних. 2. Реалізувати збереження даних за допомогою ORM. 3. Обробити помилки при збереженні (у тому числі дублі email). 4. Протестувати успішне створення користувача.
	ST.01.13 Створювати особистий кабінет при реєстрації	1. Створити модель особистого кабінету. 2. Прив'язати кабінет до користувача після реєстрації. 3. Створити початковий шаблон кабінету. 4. Протестувати автоматичне створення кабінету.
T.01.2 (US01)	ST.01.21 Додати валідацію email і пароля	1. Написати функції перевірки формату email. 2. Додати перевірку складності пароля (довжина, символи). 3. Виводити повідомлення про помилки введення. 4. Протестувати валідацію різних варіантів введення.
	ST.01.22 Застосувати хешування паролів	1. Встановити бібліотеку хешування (наприклад, bcrypt). 2. Реалізувати хешування пароля перед збереженням. 3. Зберігати лише хеш у базі даних. 4. Протестувати коректність хешування пароля.
	ST.01.23 Обробляти помилки при збереженні	1. Реалізувати механізм обробки винятків. 2. Налаштувати логування помилок. 3. Повернення користувачу інформативних повідомлень. 4. Створити unit-тести для виняткових ситуацій.
US02 (T.02.1)	ST.02.11 Створити форму входу	1. Дизайн UI для логіну (email, пароль, кнопка входу). 2. Реалізувати надсилання даних на сервер. 3. Обробити помилки у формі. 4. Протестувати поведінку форми при різних сценаріях.
	ST.02.12 Перевірити введені дані	1. Перевірити наявність email у БД. 2. Порівняти хешовані паролі. 3. Повернути повідомлення про успіх або помилку. 4. Протестувати успішні/неуспішні сценарії входу.
	ST.02.13 Створити сесію при успішному вході	1. Налаштувати сесію для авторизованого користувача. 2. Створити механізм збереження токена. 3. Забезпечити перехід на особистий кабінет після входу. 4. Протестувати створення сесії.
US02 (T.02.2)	ST.02.21 Налаштувати авторизацію через Google	1. Зареєструвати застосунок у Google Cloud. 2. Інтегрувати Google OAuth2.0 через SDK. 3. Налаштувати маршрути перенаправлення. 4. Протестувати авторизацію через Google.

Задача / Task (User Story)	Підзадача / Subtask	Роботи які потрібно виконати
1	2	3
	ST.02.22 Налаштувати авторизацію через Facebook	1. Зареєструвати застосунок у Facebook Developers. 2. Інтегрувати Facebook SDK. 3. Обробити callback від Facebook. 4. Протестувати авторизацію через Facebook.
	ST.02.23 Прив'язати соцмережу до профілю користувача	1. берегти токен у профілі користувача. 2. Реалізувати перевірку існуючого облікового запису. 3. Зв'язати профіль із соцмережею. 4. Протестувати сценарій прив'язки акаунтів.
US03 (T.03.1)	ST.03.11 Створити форму вибору послуги	1. Розробити інтерфейс для вибору категорії послуги. 2. Створити динамічне оновлення залежно від категорії. 3. Інтегрувати валідацію обов'язкових полів. 4. Протестувати UX форми вибору послуги.
	ST.03.12 Інтегрувати календар та вибір майстра	1. Імплементувати відображення доступного календаря. 2. Додати вибір зі списку майстрів. 3. Зв'язати дані майстра з розкладом. 4. Протестувати сценарії бронювання з різними майстрами.
	ST.03.13 Додати перевірку доступного часу	1. Реалізувати перевірку доступності слота. 2. Попереджати користувача про зайняті години. 3. Обмежити можливість вибору вже зайнятих слотів. 4. Протестувати логіку перевірки на прикладах.
US03 (T.03.2)	ST.03.22 Зберігати бронювання в базі	1. Налаштувати модель бронювання в базі. 2. Реалізувати механізм збереження бронювання. 3. Додати перевірку унікальності та обмежень. 4. Протестувати збереження з реальними даними.
	ST.03.22 Надсилати підтвердження користувачу	1. Розробити шаблон листа підтвердження. 2. Інтегрувати систему надсилання email. 3. Налаштувати події надсилання після збереження. 4. Протестувати надсилання підтвердження.
	ST.03.22 Відображати бронювання у кабінеті	1. Додати компонент перегляду записів у кабінеті. 2. Фільтрувати записи за користувачем. 3. Додати сортування та перегляд деталей. 4. Протестувати вивід у різних станах: з/без бронювання.
US04 (T.04.1)	ST.04.11 Зібрати графік роботи майстрів	1. Створити таблицю розкладу для кожного майстра. 2. Інтегрувати інтерфейс введення графіку. 3. Налаштувати облік вихідних та свят. 4. Протестувати правильність збереження графіку.
	ST.04.12 Розрахувати тривалість послуг	1. Додати параметр тривалості до кожної послуги. 2. Реалізувати логіку підрахунку часу бронювання. 3. Враховувати сумарну тривалість при комбінуванні послуг. 4. Протестувати точність розрахунку тривалості.
	ST.04.13 Сформувати алгоритм підбору слотів	1. Розробити алгоритм пошуку найближчих вільних вікон. 2. Врахувати графік майстра, обрані послуги, перерви. 3. Забезпечити повернення кількох альтернатив. 4. Написати юніт-тести для перевірки алгоритму.

Задача / Task (User Story)	Підзадача / Subtask	Роботи які потрібно виконати
1	2	3
US04 (T.04.2)	ST.04.21 Розробити UI-компонент для показу слотів	1. Створити UI-компонент у вигляді таблиці або списку. 2. Інтегрувати з бекендом для отримання слотів. 3. Забезпечити адаптивність компонента. 4. Протестувати відображення в браузері і на мобільних.
	ST.04.22 Інтегрувати з даними про наявність	1. Підключити дані про графік майстрів і бронювання. 2. Оновлювати доступні слоти динамічно. 3. Підключити оновлення по події. 4. Протестувати зміну слотів при оновленні графіку.
	ST.04.23 Оновлювати доступні слоти в реальному часі	1. Інтегрувати WebSocket або Long Polling. 2. Оновлювати UI без перезавантаження сторінки. 3. Тестування одночасних оновлень на різних пристроях. 4. Перевірити продуктивність при високому навантаженні.
US05 (T.05.1)	ST.05.11 Аналізувати історію візитів клієнта	1. Зібрати та зберегти історію візитів клієнтів. 2. Агрегувати дані за видами послуг, майстрами. 3. Побудувати модель частоти та пріоритетів. 4. Протестувати на прикладах з реальними даними.
	ST.05.12 Формувати релевантні послуги	1. Розробити алгоритм підбору послуг на основі історії. 2. Фільтрувати дублі, обрані послуги або доступність. 3. Оцінити релевантність на прикладах. 4. Протестувати якість результатів.
	ST.05.13 Оптимізувати швидкість обробки запитів	1. Проаналізувати вузькі місця алгоритму. 2. Імплементувати кешування результатів. 3. Застосувати асинхронну обробку запитів. 4. Провести тестування продуктивності.
US05 (T.05.2)	ST.05.21 Створити окремий компонент рекомендацій	1. Розробити UI-компонент блоку рекомендацій. 2. Підключити компонент до API з даними. 3. Забезпечити відповідність дизайну загальному стилю. 4. Протестувати відображення у різних браузерах.
	ST.05.22 Виводити персоналізований контент	1. Отримувати дані рекомендацій із бекенду. 2. Формувати динамічний список карток послуг. 3. Обробити сценарій відсутності даних. 4. Тестувати на різних облікових записах.
	ST.05.23 Адаптувати блок під мобільні пристрої	1. Оптимізувати UI-компонент для мобільного перегляду. 2. Використати медіа-запити та гнучкі сітки. 3. Перевірити доступність кнопок і тексту. 4. Протестувати на різних мобільних розширеннях.

Таблиця Б.1

Карта протиризикових заходів

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		профілактика		при симптомі	при проблемі
1	2	3	4	5	6
01	Вибір застарілої технології	Аналіз технологій перед вибором	Низька продуктивність, скарги розробників	Аудит стека	Перехід на актуальну технологію
02	Помилки в програмному коді	Код-рев'ю, модульне тестування	Часті збої, падіння сторінок	Фіксація помилок через логери	Рефакторинг або перерозробка модуля
03	Складність інтеграції з API	Пілотне підключення API	Помилки при передачі даних	Консультація з API-провайдером	Замінити API або створити проксі-рішення
04	Низька продуктивність вебплатформи	Оптимізація запитів	Затримки відгуку понад 2 с	Аналіз продуктивності	Використання кешування, редизайн запитів
05	Вразливість до кіберзагроз	Впровадження брандмауеру та шифрування	Підозрілі дії або скарги користувачів	Аудит безпеки	Блокування, відкат, інформування користувачів
06	Відсутність мобільної адаптації	Мобільне тестування	Скарги на зручність мобільної версії	Ітеративна адаптація інтерфейсу	Повна адаптація мобільної версії
07	Погано визначені вимоги	Складання чіткої специфікації	Постійні уточнення вимог	Зустрічі з замовником	Формалізація вимог і фіксація в системі
08	Часті зміни в специфікації	Пріоритезація змін	Постійні зміни задач у беклозі	Контроль змін Product Owner'ом	Переоцінка задач та планування спринту заново
09	Недостатній досвід керівника проєкту	Навчання або консультації	Помилки в управлінні, затримки	Підтримка з боку Scrum Master	Заміна/делегування керівника
10	Низький рівень комунікації в команді	Щоденні стендапи	Недорозуміння в задачах	Модерування комунікації	Зустріч 1-на-1, перегляд ролей
11	Збій хостингу	Резервні сервери	Неможливість підключення до сайту	Перемикання на резервний хостинг	Аналіз причин збою, масштабування
12	Недоступність у пікові години	Впровадження CDN	Зниження швидкості під час піків	Тимчасове обмеження запитів	Оптимізація навантаження, розширення каналів
13	Проблеми з оновленням платформи	План оновлень з тестуванням	Збій після оновлення	Ролбек оновлення	Випуск патчу, зміна стратегії оновлень
14	Відсутність плану відновлення	Розробка DR-плану	Збої без відновлення	Активація резервного механізму	Аналіз та повне відновлення з резерву
15	Порушення GDPR (обробка даних без згоди)	Аналіз політики обробки даних	Скарги на використання персональних	Консультація з юристом	Зміна логіки збирання/обробки

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		профілактика		при симптомі	при проблемі
1	2	3	4	5	6
			даних		
16	Використання неліцензованих компонентів	Перевірка ліцензій перед інтеграцією	Повідомлення про порушення	Заміна компонентів	Видалення компонентів, юридичні дії
17	Відсутність політики конфіденційності	Розміщення політики на сайті	Користувачі не бачать політики	Додавання посилань у всі форми	Публікація у footer + підтвердження згоди
18	Перевищення бюджету	Буфер бюджету	Перевищення планових витрат	Аналіз бюджету	Оптимізація витрат, корекція бюджету
19	Затримка фінансування	Погодження графіку фінансування	Відтермінування платежів	Резервне фінансування	Зміна пріоритетів або етапів реалізації
20	Недооцінка вартості підтримки	Деталізація сервісного контракту	Збільшення вартості підтримки	Контроль витрат	Зміна моделі підтримки
21	Зміна бачення проєкту з боку замовника	Фіксація бачення на старті	Невідповідність очікувань	Демонстрація результатів	Переговори та документування змін
22	Невчасне надання інформації	Графік надання даних	Затримки з боку замовника	Нагадування	Зміна етапів або тимчасова заміна
23	Пасивність у прийнятті рішень	Регулярні консультації	Відсутність рішення у дедлайн	Фасилітація прийняття рішень	Залучення відповідального представника
24	Складний інтерфейс	Юзабіліті-тестування	Скарги на складність	Збір відгуків	UX-редизайн
25	Небажання переходити на нову систему	Онбординг користувачів	Масове ігнорування	Покрокова інструкція	Промо-кампанія з демонстраціями
26	Відсутність навчання персоналу	Навчальні матеріали	Помилки користувачів	Підказки в інтерфейсі	Тренінг/онлайн-підтримка
27	Відключення електроенергії	Джерело безперебійного живлення	Нестабільна робота системи	Активация аварійного режиму	Аналіз втрат, автоматичне відновлення
28	Військові дії або політична нестабільність	Аналіз політичної обстановки	Новини про ескалацію	Перевірка плану дій	Тимчасове згортання або релокація

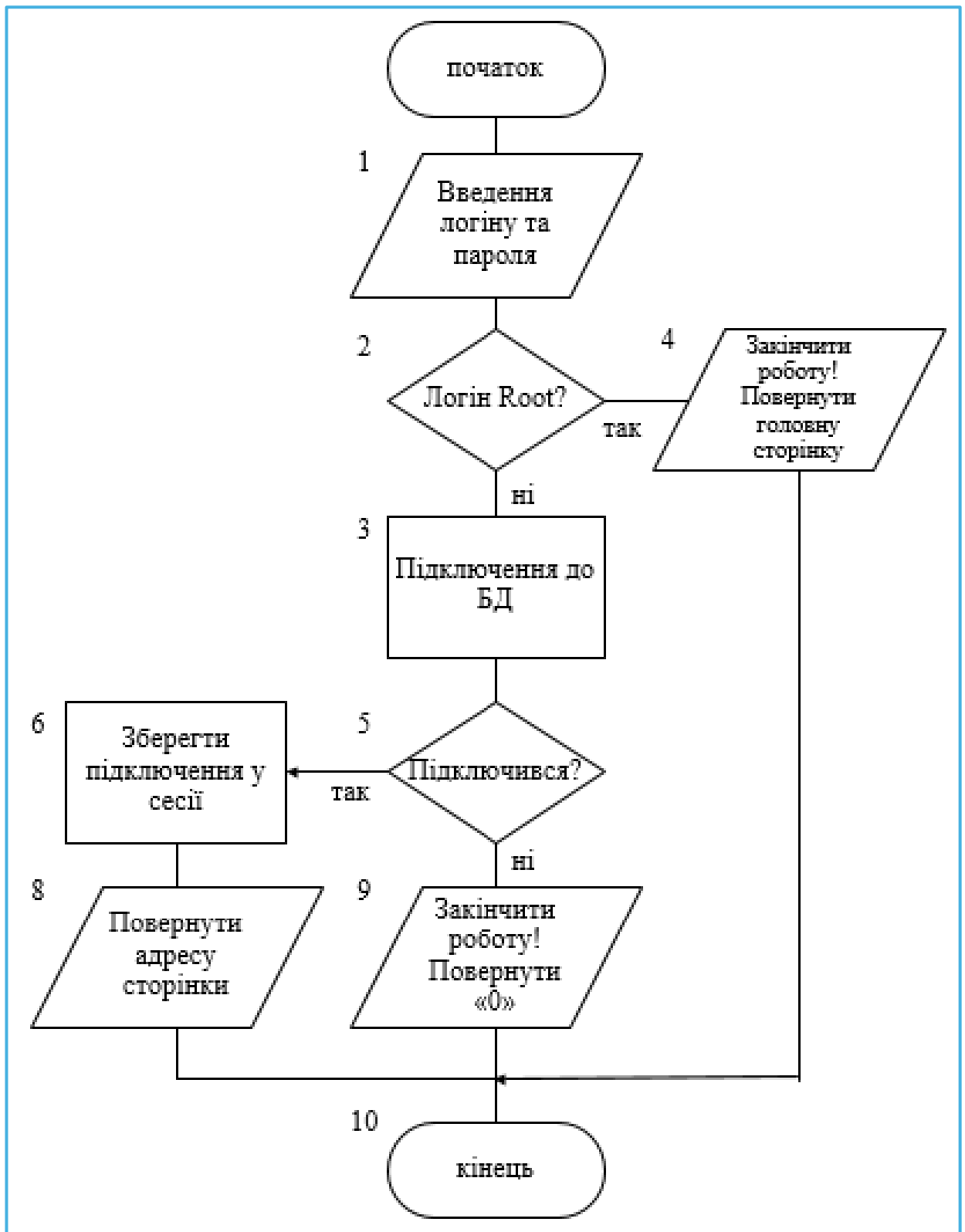


Рис. В.1. Алгоритм, який дозволяє здійснити перевірку логіну та паролю

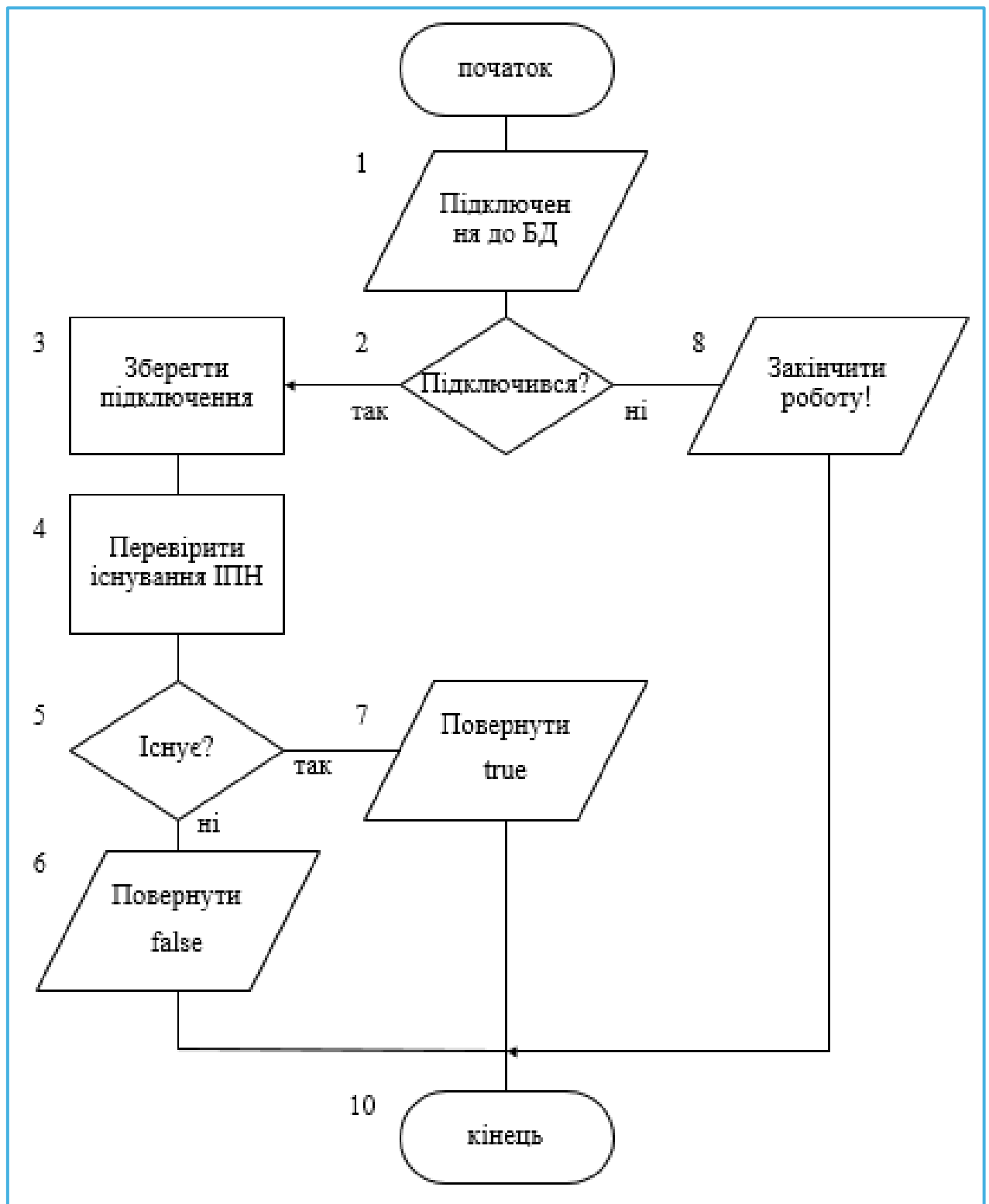


Рис. В.2. Алгоритм, який перевіряє наявність співробітника в системі

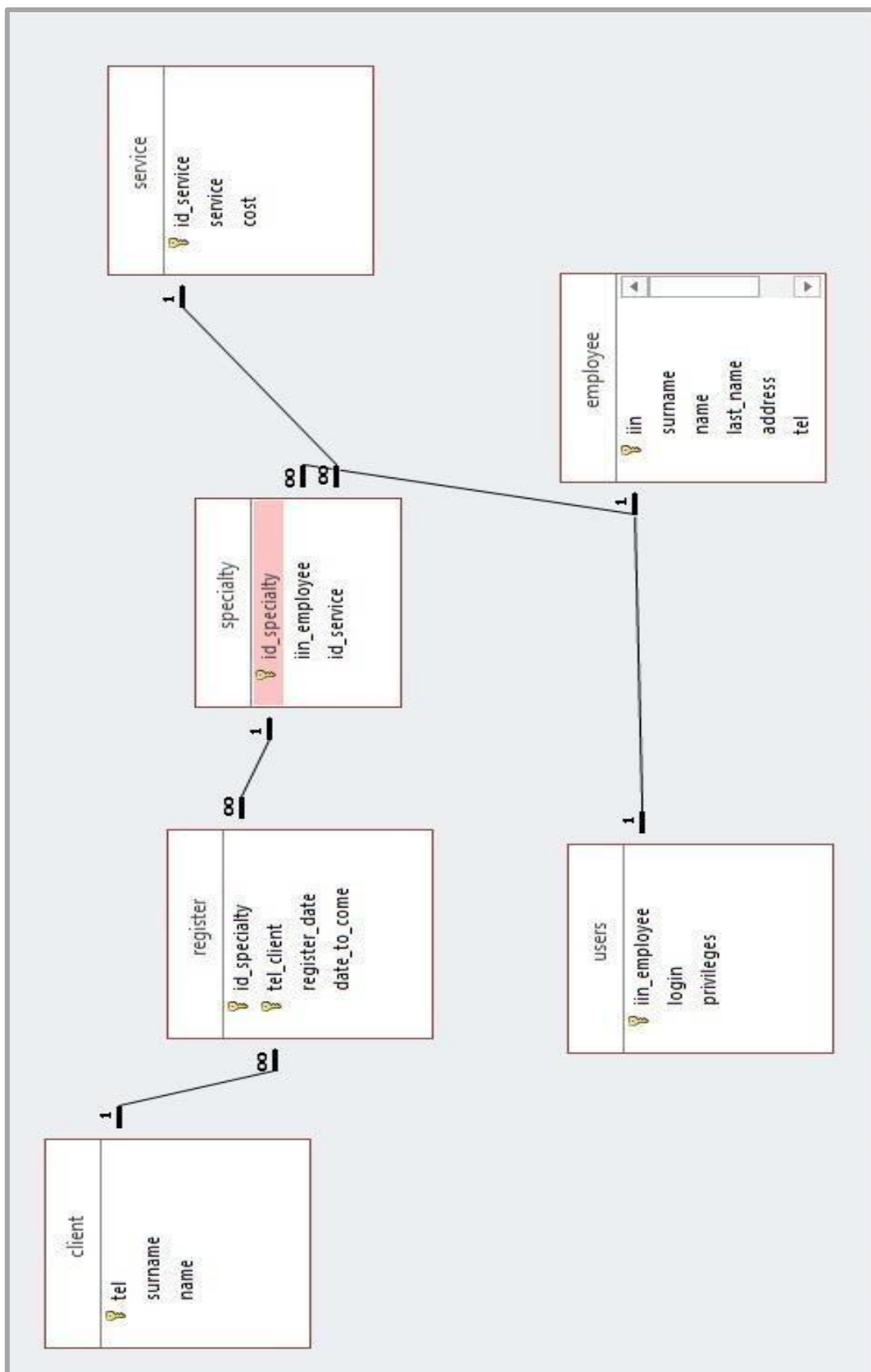


Рис. Г.1 Результируюча концептуальна модель бази даних

```

-- Створення бази даних
CREATE DATABASE BeautySalon;

-- Використання бази даних
USE BeautySalon;

-- Створення таблиці для зберігання даних клієнтів
CREATE TABLE Clients (
    ClientID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Phone VARCHAR(20),
    Email VARCHAR(100),
    Allergies TEXT,
    Preferences TEXT
);

-- Створення таблиці для зберігання даних про відвідування та послуги клієнтів
CREATE TABLE ClientVisits (
    VisitID INT AUTO_INCREMENT PRIMARY KEY,
    ClientID INT,
    VisitDate DATE,
    ServiceID INT,
    MasterID INT,
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),
    FOREIGN KEY (ServiceID) REFERENCES Services(ServiceID),
    FOREIGN KEY (MasterID) REFERENCES Masters(MasterID)
);

-- Створення таблиці для зберігання даних про послуги салону
CREATE TABLE Services (
    ServiceID INT AUTO_INCREMENT PRIMARY KEY,
    ServiceName VARCHAR(100) NOT NULL,
    Price DECIMAL(10, 2),

```

Фрагмент Коду стилів CSS

```
body {  
  background: #FFFAFA;  
}  
  
aside {  
  float: left;  
  background: #FFFFFF;  
  width: 20%;  
  padding: 1.5%;  
  min-height: 270vh;  
  color: #fff;  
  border-right: 5px solid #4d4d4d;  
}  
  
aside img {  
  width: 250px;  
  display: block;  
  margin: auto;  
}  
  
aside h3 {  
  margin-top: 50px;  
  font-size: 30px;  
  color: #000000;  
}  
  
aside ul {list-style: none}  
  
aside ul li {margin-top: 20px;}  
  
aside ul li a {  
  color: #000000;  
  text-decoration: none;  
  display: block;  
  transition: all .4s ease-in;  
}  
  
aside ul li a:hover {  
  color: #ff0000;  
  transform: scale(1.1);  
}  
  
main .features {  
  float: left;  
  color: #000000;  
  margin-top: 100px;  
  text-align: center;  
  width: 80%;
```

```

}

main .features h1 {
  font-size: 65px;
  color: black;
}

main .features p {
  max-width: 1800px;
  margin: 30px auto;
  text-align: center;
  line-height: 1.2;
  white-space: pre-line;
  font-size: 25px;
}

main .aboutus {
  float: left;
  color: #000000;
  margin-top: 100px;
  text-align: center;
  width: 80%;
}

main .aboutus h1 {
  font-size: 65px;
  color: black;
}

main .aboutus b {
  font-size: 25px;
  color: mediumblue;
}

main .aboutus p {
  max-width: 1400px;
  margin: 30px auto;
  text-align: justify;
  line-height: 1.2;
  white-space: pre-line;
  font-size: 25px;
}

main .fulltext {
  float: left;
  color: #000000;
  margin-top: 100px;
  margin-bottom: 20px;
  text-align: justify;
  line-height: 1.2;
  width: 80%;
}

```