

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота
на здобуття освітнього рівня бакалавра
за спеціальністю 121 Інженерія програмного забезпечення
на тему:

**РОЗРОБКА ВЕБ-СИСТЕМИ
ДЛЯ ДИСТАНЦІЙНОЇ ОСВІТИ**

Виконала студентка 4-го курсу
Ангеліна ГОНЧАРУК

(підпис)

Науковий керівник:
доцент, кандидат фізико-математичних наук
Лариса КАТЕРИНИЧ

(підпис)

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

Студент

(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри інтелектуальних
програмних систем

«__» _____ 202_р.,

протокол № _____

Завідувач кафедри

Олександр ПРОВОТАР

(підпис)

РЕФЕРАТ

Обсяг роботи 60 сторінок, 23 використаних джерела, 36 рисунків.

ДИСТАНЦІЙНА ОСВІТА, ДИСТАНЦІЙНЕ НАВЧАННЯ, ІНТЕРФЕЙС, НАВЧАЛЬНА СЕРЕДОВИЩЕ РОЗРОБКИ, СИСТЕМА, ФРЕЙМВОРК.

Об'єктом кваліфікаційної роботи на здобуття освітнього рівня бакалавра є створення системи для дистанційного навчання.

Мета кваліфікаційної роботи полягає у показі корисних можливостей даного проекту відносно класичних методів здобуття освіти та впровадження даної системи для навчання серед тих, хто цього потребує.

Предметом роботи є веб-система для навчання, розроблена за допомогою фреймворку Django.

Результат роботи: виконано загальний огляд дистанційного навчання в Україні та світі, виділено його переваги та недоліки, розглянуті технології розробки систем та обрано найоптимальніші, розроблено систему для здобуття освіти дистанційно.

Для виконання роботи було використано середовище розробки PyCharm. Керування базами даних здійснювалось з використанням об'єктно-реляційної системи PostgreSQL. Відображення сторінок здійснювалося за допомогою мови тегів HTML і спеціальної мови стилю сторінок CSS.

Проект, що є результатом роботи, може отримати застосування в навчальних закладах для організації дистанційного навчання.

ЗМІСТ

ВСТУП	5
РОЗДІЛ 1 ДИСТАНЦІЙНА ОСВІТА.....	7
1.1 Становлення та розвиток дистанційного навчання в Україні та за її межами	7
1.2 Сутність та зміст поняття «Дистанційне навчання».....	9
1.3 Види дистанційного навчання	10
1.4 Підстави для введення дистанційного навчання	11
1.5 Переваги та недоліки	12
РОЗДІЛ 2. ТЕХНОЛОГІЇ РОЗРОБКИ.....	14
2.1 Мова та середовище розробки	14
2.2 Обґрунтування вибору фреймворку	14
2.2.1 Переваги веб-розробки Django.....	16
2.2.2 Недоліки веб-розробки Django.....	17
2.3 Робота з базами даних.....	19
2.3.1 Порівняння SQL та NoSQL	20
2.3.2 Порівняння MongoDB та PostgreSQL	23
2.3.3 Порівняння PostgreSQL та MySQL.....	25
2.3.4 Чому було використано саме PostgreSQL.....	27
2.4 Розробка Frontend частини	28
РОЗДІЛ 3. ОСНОВНА СТРУКТУРА ПРОЕКТУ	30
3.1 Архітектура системи.....	30
3.2 Можливості незареєстрованого користувача	30
3.3 Реєстрація та вхід в обліковий запис.....	34
3.4 Можливості «Студента»	37
3.5 Можливості «Викладача».....	50

3.6 Можливості «Адміністратора» 54

ВИСНОВКИ..... 56

ВИКОРИСТАНІ ДЖЕРЕЛА 57

ВСТУП

Актуальність роботи та підстави її виконання. Чим далі рухається технічний прогрес, тим поширенішою стає так звана дистанційна діяльність. В наші дні віддалено можна робити практично все: спілкуватися, працювати, робити покупки, отримувати освіту. Останній пункт буде розглянуто детальніше.

Стрімкий розвиток та зростаюча популярність глобальної мережі Інтернет дає можливість вдосконалити систему освіти. На сьогоднішній день класичні методи навчання удосконалюються використанням сучасних здобутків науки, до яких входять Інтернет, засоби телекомунікації та інше.

Сучасний світ потребує нової системи неперервного навчання, яке дасть людині доступ до світових інформаційних ресурсів та баз даних, що дозволять покращувати її професійні навички та підвищать її професійну мобільність.

Мета й завдання роботи. Мета роботи полягає у показі корисних можливостей даного проекту відносно класичних методів здобуття освіти та впровадження даної системи для навчання серед тих, хто цього потребує. Завдання роботи – створити веб-систему для освіти дистанційно.

Об'єкт, методи й засоби розроблення. Об'єктом роботи є розробка веб-системи для здобуття дистанційної освіти. Для виконання роботи було використано мову програмування Python та середовище розробки PyCharm. Щоб керувати базами даних використано об'єктно-реляційної системи PostgreSQL. Веб-сторінки відображаються за допомогою мови тегів HTML і спеціальної мови стилю сторінок CSS.

Можливі сфери застосування. Можливість навчання на відстані досить давно цікавила педагогів та учнів. Це навчання може мати різний формат залежно від технологій та способу організації навчання. У минулому таке навчання проходило за допомогою обміну освітніми матеріалами та

періодичних зустріч учнів та викладачів для перевірки знань перших на залікових та екзаменаційних сесіях в типовому для навчального закладу форматі. Дана форма навчання називається заочною.

Через постійне збільшення об'ємів навчальних матеріалів учні мають вивчати частину навчальних матеріалів власноруч. У випадку очної форми навчання самостійна робота учня займає дещо менше половини часу, виділеного на освітній процес. Варто зауважити, що розмір роботи без сторонньої підготовки поступово збільшується.

Переваги розподіленого дистанційного навчання полягають в його нових функціях, розширенні можливостей і послуг наданих студентам, його розподілу по різних ринкових областях і регіонах, а також використання гнучкої системи безперервної освіти, гнучкого розкладу, проведення занять в різних режимах. Основна особливість гнучких моделей навчання полягає в тому, що вони не визначають строго часові та просторові рамки для спілкування в класі і між учителем і учнем. Це дозволяє вчителям та учням більш гнучко планувати час навчання і навчальний процес. Це особливо важливо для студентів-заочників та студентів, що навчаються у філіях ВНЗ в інших містах і навіть країнах.

РОЗДІЛ 1 ДИСТАНЦІЙНА ОСВІТА

1.1 Становлення та розвиток дистанційного навчання в Україні та за її межами

Для вирішення нових задач, що постають перед українською системою освіти, потрібно виробити адекватну структуру системи освіти, що в подальшому дала б можливість громадянам отримувати освіту протягом всього життя.

Рішення даної задачі можливо знайти за допомогою використання системи дистанційного навчання.

Дистанційна освіта в Україні почала розвиватись досить пізно порівняно з більш розвиненими країнами, оскільки українське суспільство мало низький рівень інформатизації, також більшість шкіл були не достатньо оснащені необхідною технікою та загалом методики дистанційного навчання були відсутні.

Розробка різних аспектів дистанційного навчання повністю не завершена. Створення та використання певних курсів дистанційного навчання у дослідницьких та вищих навчальних закладах України на даний момент не надто поширені [1].

Дистанційне навчання України почало розвиватись з використанням вже існуючих здобутків світу. Даний процес можна поділити на декілька етапів. Початок становлення дистанційної освіти України припав на кінець дев'яностих років ХХ століття. Закон України «Про національну програму інформатизації» був прийнятий Верховною Радою лише наприкінці зими 1998 року. Від часу прийняття даного Закону українська система освіти зазнала деяких позитивних змін у сферу інформатизації та розвитку Інтернету [2].

У 1997 році по всій Україні почали систематично проводити конференції «Інтернет – технології в інформаційному просторі держави». Наступного року

вченими було організовано та проведено подібну конференцію в Одесі, на якій підписали договір про співробітництво між двадцяти семи вищими навчальними закладами України [3].

Загалом за межами України дистанційне навчання розпочалось набагато раніше. Дистанційне навчання або дистанційна освіта насправді існує вже більше ста років.

Справжнє походження дистанційного навчання вже довгий час викликає суперечки, оскільки початок не був добре задокументований. Деякі джерела стверджують, що дистанційна освіта розпочалася ще в 1700-х роках. Ця форма навчання практикувалась на заочній формі навчання і широко розрослась по країнах світу. Під час такого навчання студенти отримували інструкції та матеріали поштою, а потім відповідали на запитання чи виконували завдання. Хоча даний процес був неймовірно повільним, він продовжував набувати обертів, оскільки на той час студенти так прагнули вчитися, що затримка не мала значення [4].

Деякі джерела стверджують, що дистанційна освіта почалася в 1840 р., Коли англійський педагог Ісаак Пітман почав викладати стенографію поштою. Він розсилав картки студентам, а вони відправляли завдання назад. Заочне навчання по-справжньому почало набувати сили в 1858 р. Лондонський університет став першим коледжем, який запропонував отримати навчальний ступінь дистанційно. Тільки через 30 років була заснована велика приватна школа, що знаходиться в штаті Пенсильванія, яка проводила подібну форму навчання з метою підготовки іммігрантів-видобувачів вугілля, що бажали стати державними інспекторами шахт. Протягом 18 років кількість учнів на даній формі навчання зросла з 2500 до 900 000. З цього можна зробити висновок, що дистанційна освіта була затребуваною вже в досить далекому минулому [5].

Із часом і технології стали значно прогресивнішими, формат навчання почав суттєво змінюватися. Радіо дозволяло університетам своєчасно

передавати інформацію та курси студентам. Університети навіть почали пропонувати телевізійні курси та навчальні програми для студентів, починаючи з 1930-х років. Ці методи навчання поширилися на середину століття. Багато університетів почали пропонувати телевізійні курси в кредит.

Протягом 90-х років винахід персонального комп'ютера та Інтернету революціонізував спосіб дистанційного навчання. Інтернет-університет став звичним, і люди могли здобувати ступінь бакалавра та ступінь магістра в завдяки Інтернету. Інтернет-навчання було важливим у 90-х роках, оскільки університети не могли задовольнити високий попит на очні зустрічі. Поєднання віртуальних класів, мобільних телефонів, відеоконференцій та Інтернету було ключем до дистанційної освіти.

Сьогодні навчання в Інтернеті досить поширене. Більшість великих світових університетів мають онлайн-програми для різноманітних курсів. Окрім університетів, наприклад, багато торгових шкіл також пропонують дистанційні курси. Сьогодні в Інтернеті можна дізнатись практично будь-що та здобути освіту майже для будь-якої професії. Порівняно з минулими часами можливості – безмежні [6].

1.2 Сутність та зміст поняття «Дистанційне навчання»

Дистанційне навчання має відмінності від очного та заочного навчання. Основні відмінності полягають у використанні нових засобів та методик навчання, організації навчання дещо іншим способом. Форми комунікації та взаємодії педагога та учня, учня з іншими учнями досить відрізняються від класичних. Дистанційне навчання набагато вимогливіше і потребує більших затрат зусиль. Однак воно пропонує хороші перспективи для студентів, які прагнуть вчитися, але поки що не мають можливості відвідувати звичайні заняття. Як і у випадку з іншими типовими та більш поширеними формами

навчання, дистанційне навчання ставить цілі, що враховують потреби суспільства. Способи організації дистанційного навчання прямо залежать від конкретних технологій та засобів, використаних під час процесу здобуття знань.

Дистанційна освіта задовольняє наступні цілі:

- професійна підготовка, перепідготовка та підвищення кваліфікації кадрів;
- вивчення навчальних предметів до здачі заочних іспитів;
- підготовка до вступу в профільні навчальні певного;
- поглиблене вивчення матеріалів зі програми навчання або за її межами;
- заповнення прогалин в знаннях;
- додаткова освіта за бажанням [7].

1.3 Види дистанційного навчання

На даний момент можна виділити п'ять основних видів дистанційного навчання.

1. Засобами зв'язку під час якого є електронна пошта і факс. Учні отримують навчальні матеріали по електронній пошті та насилають назад письмові звіти та результати власної практичної роботи і завдань. Доступні відео- та аудіокасети, диски та дискети з навчальними програмами для комп'ютера.

2. Коли в процесі навчання використовуються освітні телепрограми, які інтегровані в навчальний графік денних курсів, доповнюючи навчальні програми. Канали електронної пошти використовуються в якості зворотного зв'язку, по якій учні отримують допомогу від вчителів і відправляють звіти.

3. Навчання за допомогою освітніх телеконференцій і відеоконференцій. Дані два типи конференцій часто поєднуються в процесі навчання: телеконференції використовуються на початкових етапах викладацької діяльності для трансляції теоретичних матеріалів, аудіо- та відеоконференцій, для семінарської або проектної роботи в невеликих групах. Учні виконують свої проекти, і за допомогою конференцій вони збираються разом, щоб продемонструвати та обговорити звіти, координувати навчальну і дослідницьку діяльність, отримати пораду від вчителя і т. д..

4. Навчання на базі комп'ютерних освітніх систем. З електронними навчальними виданнями, зазвичай включеними в навчально-методичний комплекс та складаються з підручника, навчальних планів, студент може працювати самостійно на своєму комп'ютері або безпосередньо в Інтернеті. Електронна пошта та телеконференції широко використовуються для зворотного зв'язку.

5. Дистанційне навчання організоване в середовищі Інтернету з використанням інтерактивних веб-посібників, електронної пошти, чатів та відеоконференцій для надання зворотного зв'язку, комп'ютерних моделей та симуляцій [8].

1.4 Підстави для введення дистанційного навчання

Оскільки світ стрімко розвивається, постійно з'являються нові відкриття, то безліч знань, які студенти мають здобути для отримання майбутньої професії, старіють та стають не актуальними. На додаток до цього обсяг інформації, що мають вивчити студенти для успішного закінчення вищого навчального закладу, подвоюється кожні три-чотири роки. Оскільки без зміни освітніх технологій підготовка фахівців буде не достатньою, щоб

вони були затребуваними на ринку праці, то модернізація системи освіти безперечно необхідна [9].

Також дистанційна освіта особливо актуальна на сьогоднішній день, так як у ці перемінливі часи соціальне дистанціювання стало новою нормою. Людству довелося адаптуватися у кожному аспекті нашого життя - від носіння масок у продуктовому магазині до занять в Інтернеті. Зараз, як ніколи, ця форма навчання потрібна у більшості країнах світу.

1.5 Переваги та недоліки

Дистанційне навчання має як переваги, так і недоліки порівняно з традиційними формами навчання. Переваги включають, перш за все, можливість навчання у зручному для студента місці та в зручний час, а отже, зменшення витрат на проїзд, проживання та економію часу. Завдяки усуненню часових і просторових бар'єрів, ця форма навчання дозволяє навчатися багатьом людям, наприклад людям з інвалідністю та тим, для кого доступ до університету з якихось причин став ускладненим або неможливим (наприклад, жінки, які займаються вихованням дітей). Крім того, процес дистанційного навчання менш напружений та в основному індивідуалізований. Також, це розвиває самодисципліну та організованість учнів. Дистанційна освіта дозволяє збільшити різноманітність навчання завдяки використанню мультимедіа та різних добре сконструйованих додатків, мінімізувати страх і сором'язливість, які можуть виникнути в традиційному колективі. Однак існують також недоліки та обмеження процесу дистанційного навчання:

- тривалий аналіз мультимедійних навчальних матеріалів та форм спілкування з іншими студентами та викладачем;
- розробка електронних навчальних матеріалів займає більше часу, ніж класичних друкованих;

- складність організації навчального процесу: робота в групі, створення навчальної спільноти, користування послугами електронного деканату та цифрової бібліотеки;
- відокремлення від групи та вчителя зменшує соціальні зв'язки, мотивацію та наполегливість у навчанні;
- через відсутність спеціалізованого учень часто може відволікатися на побутові проблеми та подразники;
- неможливість взяти участь у роботах, наприклад, лабораторних, експериментальних та дизайнерських (маються на увазі роботи, що потребують знаходження людини або групи людей в приміщенні з потрібним обладнанням);
- відсутність навичок роботи з комп'ютером та використання мережі;
- відволікаючі можливості Інтернету для користувача (Інтернет має багато цікавої, не обов'язково навчальної інформації);
- відсутність стимулюючої атмосфери для навчання, характерної для школи чи групи, відчуття ізоляції;
- необхідність мати схильність до самоосвіти та самоконтролю [10].

РОЗДІЛ 2. ТЕХНОЛОГІЇ РОЗРОБКИ

2.1 Мова та середовище розробки

Для створення проекту була використана Python – потужна, динамічна мова програмування, що використовується для розробки веб-додатків високого рівня та програм машинного навчання.

Як середовище розробки системи було обрано PyCharm. PyCharm сумісний з обома версіями Python 2.x, а також 3.x. Хоча PyCharm спеціально розроблений для Python, проте в PyCharm є підтримка також файлів типів HTML, CSS та JavaScript. PyCharm надає підтримку цим мовам програмування, оскільки Python також використовується для створення веб-додатків. PyCharm також має прекрасний інтерфейс користувача з можливістю налаштувати його відповідно до потреб.

Як і інші IDE, присутні на ринку, PyCharm в основному використовується для редагування, написання, налагодження та управління кодом Python. І не тільки Python, також PyCharm надає можливість кодувати кількома мовами програмування під час використання. PyCharm має дуже інтерактивний та ефективний користувальницький інтерфейс, що дозволяє користувачеві керувати великими проектами. Легко створити новий проект, а функція віртуального середовища допомагає розробникам ізолювати один проект від іншого. PyCharm вважається найкращим Python IDE, і за допомогою нього можливо ефективно працювати над великими проектами, які містять кілька каталогів і тисячі рядків коду [11].

2.2 Обґрунтування вибору фреймворку

Фреймворком для розробки проекту було обрано Django.

Django – найбільш часто використовуваний фреймворк для розробки проектів на Python. Django здобув велику популярність завдяки своїм обчислювальним можливостям.

Сьогодні Django є однією з найбільш часто використовуваних платформ веб-розробки. Розробники широко віддають перевагу веб-розробці за допомогою Django, адже, розробляючи в Django не потрібно встановлювати окремі бібліотеки для додавання функціональних можливостей. Django надає безліч функцій для додавання стандартних функціональних можливостей у будь-який вид веб-додатків.

Django - це програма розробки веб-додатків з відкритим кодом на основі MVT, яка написана на Python. Це дозволяє чітко написання коду та зменшує складність веб-розробки.

Веб-розробка Python за допомогою Django стає простішою, оскільки в Python відсутні вбудовані можливості веб-розробки. Django надає функціональні можливості, які компенсують цей розрив.

Розробка Django користується високою популярністю завдяки філософії «батареї в комплекті». Він надає багато загальних функціональних можливостей для веб-розробки, що позбавляє від написання коду для кожної функції.

Веб-фреймворк Django також забезпечує розширені функції, такі як ORM, міграція баз даних, автентифікація користувачів, адміністративна панель та форми. Перевагу Django часто надають через те, що фреймворк спрощує роботу з базами даних. Це прискорює процес розробки та дозволяє розробникам створювати масштабовані програми. Розробники також використовують його, оскільки він дозволяє розділити проекти Django на кілька додатків.

Веб-фреймворк Django широко використовується у всьому світі, він має свої плюси і мінуси. Розуміння переваг та недоліків розробки Django має

важливе значення для розробників та підприємств, що мають сумніви в тому, яку структуру їм слід вибрати для розробки. [12]

2.2.1 Переваги веб-розробки Django

Можна побачити наступні переваги при розробці за допомогою вказаного веб-фреймворку:

- «Батареї в комплекті»

Перша і найголовніша перевага веб-розробки. Замість написання коду розробники можуть використовувати готові пакети для додавання функціональних можливостей. Це економить величезну кількість часу для розробників, і вони можуть зосередитися на додаванні розширених функцій.

Можна використовувати пакети, створені членами спільноти світового класу, оскільки веб-фреймворк Django є відкритим. Також можна реалізувати автентифікацію за допомогою пакета `auth`, взаємодію адміністратора з пакетом адміністратора, управління сесансами за допомогою пакета `Sessions` тощо. Оскільки всі функції включені в фреймворк, то можна просто імпортувати ці пакети і в найкоротші терміни розпочати створення додатків.

- Гнучкі рамки

Django не дотримується домовленостей щодо механізму конфігурації. Оскільки Django написаний на Python, універсальній мові програмування, веб-розробка має більше гнучкості та динамічності.

Можливо конфігурувати фреймворк на ходу відповідно до своїх потреб і створювати веб-додатки.

Будучи написаним на Python, Django забезпечує чудову підтримку зовнішніх бібліотек та пакетів. Можна легко імпортувати пакет у Django без будь-яких проблем із безпекою.

У Django більше уваги робиться на явне програмування, а не на неявне, що робить його одним з ідеальних фреймворків для додатків, які вимагають швидких змін.

- Швидший розвиток

Одним з найважливіших плюсів розробки Django є те, що він забезпечує швидший розвиток. Django має тести для перевірки внутрішньої швидкості розробки. За належної конфігурації веб-розробка Django забезпечує оптимізацію веб-додатків.

Хоча веб-розробку Python з Django можна вважати дещо повільною через конфігурації Python, Django це компенсує.

Конфігурація каркасу та архітектурний дизайн полягають у тому, що він дозволяє використовувати декілька компонентів одночасно, що забезпечує швидкий розвиток. Це дає можливість декільком розробникам працювати паралельно, не втрачаючи швидкості розробки [13].

2.2.2 Недоліки веб-розробки Django

Хоча веб-фреймворк Django надійний і відомий своїми можливостями розробки, він все ще не є основною структурою для багатьох програмістів. Є кілька недоліків Django, і давнім розробникам важко здійснити перехід. Деякі мінуси розробки Django перелічені нижче.

- Жодних домовленостей

Більшість програмістів не люблять веб-розробку Django через відсутність умов.

Фреймворк не має набору принципів, яким розробники можуть керуватися при веб-розробці.

Це ускладнює процес для розробників, які працювали над фреймворками з Конвенцією про конфігурацію.

Часто налаштування на ходу призводить до невідповідних компонентів, і багатьом розробникам це неприємно. Ранні розробники Django стикалися з цією проблемою. Оскільки все потрібно визначати в міру розвитку, іноді швидкість розвитку може бути набагато повільнішою, ніж передбачалося.

– Часто не для малих проектів

Веб-фреймворк Django не часто підходить для невеликих проектів та продуктів із лише декількома функціями та вимогами.

Django вимагає багато кодування, яке займає час обробки сервера та пропускну здатність під час розробки. Зазвичай він використовується для проектів, які потребують масштабування або будуть запущені у великих масштабах.

Веб-розробка за допомогою Django для невеликих проектів складна, оскільки фреймворк може заплутати розробників своїми унікальними функціональними можливостями.

– Монолітний каркас

Для всіх «батареї в комплекті» у Django, рамки є монолітними. У нього мізерна кількість залежностей, що ускладнює використання. Програмісти Django розглядають фреймворк як єдиний пакет, що погано. Чим менші залежності у Django, тим більше коду доведеться писати.

Django не заохочує розробників вивчати пакети і інструменти Python. Замість цього розробка Django фокусується на наданні програмістам все більшої і більшої кількості програмування, орієнтованого на код. В Django легко використовувати бібліотеки, але розглядати його як єдиний пакет для веб-розробки - помилка багатьох розробників.

– Крута крива навчання

Одним з основних недоліків є те, що веб-структура Django має круту криву навчання. Хоча це чітка та проста рамка, вона написана на Python, що ускладнює навчання. Django має безліч функцій та конфігурації, які розробники не можуть зрозуміти легко.

Розробникам, які переходять з інших мов на розробку Python за допомогою Django, часто важче вчитися. Фреймворк має інший синтаксис завдяки Python, і розробники не швидко його зрозуміють. Вони мусять навчитися цьому довгий час, перш ніж зможуть стати експертами в цьому. Незважаючи на те, що фреймворк застарів і розвивається повільно, його все-таки найскладніше освоїти [14].

– Випуск кількох запитів

Django відрізняється від більшості фреймворків, але іноді це також є недоліком. На відміну від більшості фреймворків веб-розробки, Django не може обробляти кілька запитів одночасно. Вони являють собою запити на окремі процеси, і для виконання кожного запиту потрібен час. У цьому сенсі Django не забезпечує швидкого розвитку.

Управління кількома запитами пропонується майже всіма системами ринку. Причина, по якій Django не підтримує кілька запитів, полягає в тому, що вона заохочує розробників Django досліджувати окремі процеси та приймати рішення [15].

2.3 Робота з базами даних

Для роботи з базами даних в даному проєкті було використано систему PostgreSQL.

У сучасному технологічному суспільстві цілісність та безпека даних стають головним пріоритетом. Особливо, якщо говорити про великі проекти, які вимагають зберігання сотень чи тисяч інформаційних блоків в одному безпечному місці.

Отже, вибір належної бази даних для майбутнього проекту – це не якась незначна річ. Потрібно знати всі варіанти, їх переваги і недоліки та зрозуміти, які з них найбільше підходять.

Протягом кількох останніх десятиліть лідером серед варіантів роботи з базами даних була система PostgreSQL. Це вдосконалена об'єктно-реляційна система з відкритим кодом, яка використовує мову SQL. PostgreSQL дозволяє безпечно зберігати об'ємні та складні дані. Це допомагає розробникам створювати найскладніші програми та цілісні середовища.

Починаючи з 1986 року, коли PostgreSQL був створений, у нього було багато прихильників та критиків. Щоб зберегти назву «найдосконалішої бази даних», PostgreSQL регулярно оновлюється, додає нові функції. Попри це зростає і кількість конкурентів PostgreSQL.

В даному розділі буде підсумовано все, що відомо про PostgreSQL, проаналізовано його визначні особливості, буде порівняно PostgreSQL з базами даних SQL та NoSQL, щоб зрозуміти, чому було обрано PostgreSQL серед безлічі інших баз даних. PostgreSQL працює на основі мови SQL [16].

2.3.1 Порівняння SQL та NoSQL

SQL – це декларативна мова програмування для створення та експлуатації даних у реляційній базі даних. У той же час NoSQL, скоріше, визначає набір підходів до зберігання даних не так, як це робить SQL.

Створені в 1960-х роках, бази даних NoSQL зуміли набути реальної популярності лише останнім часом. Цьому сприяло поширення таких баз даних NoSQL, як CouchDB, MongoDB, Apache Cassandra та Redis.

– Формат зберігання даних

Системи NoSQL зазвичай зберігають дані у полях, подібних до JSON. Більше того, будь-які схожі файли в системах NoSQL можна компілювати в колекції. Ці системи приймають документи будь-якого типу.

Системи SQL зазвичай зберігають інформацію в таблицях даних. Таблиці взаємопов'язані та мають фіксований шаблон даних. Така строгість здається менш привабливою, проте вона мінімізує можливість будь-яких помилок.

– Схема

У SQL схема містить типи полів і таблиці бази даних. Це означає, що для запуску проекту в SQL потрібно розробити базу даних перед будь-якою логікою. Після того, як створено схему, буде досить складно внести будь-які зміни.

Для NoSQL таких обмежень немає. Користувачі можуть додавати зміни у свою базу даних у будь-який час. База даних NoSQL не має схеми, а тому не вимагає заздалегідь встановлювати жорсткий дизайн бази даних.

– Операція JOIN

JOIN – ще одна корисна функція SQL, що дозволяє отримати всі необхідні дані з вибраних таблиць за допомогою механізмів SQL. NoSQL подібного не має, але є можливість отримати необхідну інформацію вручну.

– Безпека та практичність

Проблеми, пов'язані з безпекою, та проблеми з функціонуванням можуть обернутися головним болем для користувачів NoSQL. Проте ці

проблеми, як правило, спричинені не певними системними прогалинами, а скоріше браком знань. Будучи не надто досвідченими у роботі з молодими системами NoSQL, розробники досить часто не дотримуються простих правил безпеки.

- Масштабованість

Для баз даних SQL масштабованість може викликати проблеми. Якщо буде потрібно розподілити пов'язані дані, то можна розглянути можливість кластеризації декількох серверів навколо одного центрального сховища.

При використанні NoSQL це набагато простіше зробити. Системи NoSQL пропонують функцію масштабування, яку можна запустити з самого першого дня вашої роботи в системі.

- Транзакції

Найкраща цілісність та точність даних є головним пріоритетом для систем SQL. Ще одним доказом цього є їх механізми транзакцій. Ці бази даних дозволяють розміщувати два або більше оновлень під час будь-якої транзакції. Це означає, що за будь-якою транзакцією група оновлень може бути або прийнята, або відхилена. У будь-якому випадку, точність даних буде збережена.

Цей механізм не працює для баз даних NoSQL, де кожне оновлення приймається або відхиляється окремо. Це може призвести до дрібних наслідків: зрештою дані можуть бути неактуальними.

- Цілісність даних

Більшість баз даних SQL забезпечують цілісність даних, блокуючи будь-які недійсні дані чи записи-сироти. Для баз даних NoSQL це не працює таким чином – є можливість зберігати будь-які дані, які необхідно без будь-яких обмежень. Відсутність обмежень може здатися хорошим рішенням, але це може призвести до помилок у роботі з базою даних.

– Висновок

Для роботи зі звичайними програмами чи проектами середнього розміру, то краще обрати SQL. Для проектів з надвисоким навантаженням, можна використовувати NoSQL, або почати працювати з SQL, а потім перейти на NoSQL. Для будь-яких проектів на стадії ініціалізації SQL буде розумним рішенням. Причина полягає в тому, що нещодавно розпочаті проекти зазвичай не мають структурованих вимог і можуть змінювати їх з часом. У рамках NoSQL ці зміни важко здійснити, тоді як бази даних SQL є більш гнучкими [17].

2.3.2 Порівняння MongoDB та PostgreSQL

MongoDB – це розподілена база даних на основі документів, яка добре підходить для додатків та хмар. MongoDB підтримує документи JSON і пропонує гнучкі та динамічні схеми. Більше того, він має потужну мову запитів. За допомогою нього можливо фільтрувати та сортувати дані, а також виконувати численні типи пошуку.

PostgreSQL може легко керувати найбільш важливими запитами найбільших компаній та установ. Його ефективність доведена роками. Крім того, PostgreSQL не зупиняється на оптимізації своєї роботи, випускаючи нові та нові версії. Деякі з них, до речі, включали також вдосконалення неструктурованих типів даних. Найгучнішим прикладом є введення функції JSONB.

– Перевірка дійсності

Постійно вдосконалюючись у нових версіях, PostgreSQL пропонує нові інструменти та функції для забезпечення достовірності даних. База даних

забезпечує перевірку даних для будь-якого поля JSON, тому введення будь-яких неправильних даних в підсумку призведе до помилки.

- Програмування на стороні сервера

Незважаючи на MongoDB, PostgreSQL працює з різними мовами процедур, такими як Python, JavaScript, C, C ++, Tcl, Perl та багатьма іншими.

Формати даних: Postgres NoSQL та MongoDB

Postgres NoSQL тепер підтримує всі загальні формати даних: JSON (документ), ключ-значення, XML. Такі самі формати підтримує MongoDB. Це робить Postgres NoSQL гарною альтернативою деяким системам NoSQL.

- Інтеграція з іншими джерелами

Postgres NoSQL також забезпечує взаємодію з іншими джерелами даних, тоді як MongoDB цього не робить. Отже, Postgres може витягувати дані з Oracle, MySQL, MongoDB (так), CouchDB, Redis, Neo4j, Twitter, LDAP, File, Hadoop та інших.

- Бізнес-логіка

У MongoDB ділова логіка поширюється серед клієнтських додатків. У PostgreSQL він одночасно розповсюджується серед клієнтських програм і централізується за допомогою тригерів та збережених процедур.

- Наявність навчальних ресурсів в MongoDB та PostgreSQL

Якщо буде потреба почати зберігати свої дані у PostgreSQL, то легко можна знайти безліч освітніх ресурсів. Вони доступні на офіційному веб-сайті PostgreSQL та просто на інших веб-сайтах. Проте навчитися працювати в MongoDB може бути дещо складнішим, оскільки його посібники та інструкції не так легко знайти.

Отже, загалом, Postgres NoSQL має ряд переваг перед іншими базами даних NoSQL, такими як MongoDB. Причиною цього домінування є прийняття деяких функцій NoSQL [18].

2.3.3 Порівняння PostgreSQL та MySQL

У цьому розділі буде порівняно основні технічні відмінності двох SQL - PostgreSQL та MySQL. Причини надання переваги PostgreSQL:

– Продуктивність

Продуктивність Postgres набагато вища порівняно з будь-якою системою NoSQL або SQL? Загалом, як MySQL, так і PostgreSQL демонструють швидке функціонування навіть під час роботи з великими навантаженнями. Однак не завжди було так. У минулому MySQL мав репутацію швидкої бази даних за рахунок її одночасності. Натомість PostgreSQL, як правило, демонструє збалансовані результати разом із обробкою паралелізму. З останніми версіями обох баз даних усі відмінності були стерті.

– Паралельність

Переможцем в цьому питанні стане PostgreSQL. Ця база даних обробляє паралельність набагато краще, ніж її конкурент MySQL. Це завдяки Multiversion Concurrency Control, який реалізує PostgreSQL.

– Розширюваність

PostgreSQL надзвичайно розширювана, може підтримувати численні типи даних. Серед них: геометричні / ГІС, типи мережевих адрес, JSONB, власний UUID, часові позначки з урахуванням часових поясів. MySQL не пропонує аналогічних опцій.

– Реплікація

MySQL пропонує односторонню асинхронну реплікацію, коли одна база даних вважається домінуючою, а інші – другорядними. Для PostgreSQL типовою є синхронна або безпечна реплікація. Цей тип реплікації передбачає синхронізацію основної бази даних з другорядною. Така реплікація забезпечує більший захист, оскільки ваші дані можуть бути втрачені лише в тому випадку, якщо обидві бази даних зламаються в один момент.

– Документація

Документація – ще одна сильна сторона PostgreSQL. У документації PostgreSQL можна знайти відповідь майже на кожне запитання. Хоча прихильники Oracle іноді критикують документацію PostgreSQL, вона все одно значно витісняє MySQL.

– Адміністрування

Адміністрування набагато простіше в MySQL. Однак це досить відносна перевага – така простота є результатом гіршої функціональності. Отже, це означає, що складні запити навряд чи можна обробляти в MySQL. Навпаки, PostgreSQL може легко з ними справлятися. Щоб правильно його встановити, потрібно витратити трохи часу. Однак з часом усі ваші зусилля окупляться.

– Тригери

PostgreSQL підтримує тригери, які можуть реагувати на більшість типів команд, за винятком тих, що впливають на базу даних у всьому світі, наприклад, ролі та табличні простори. MySQL, у свою чергу, обмежений лише деякими командами [19].

2.3.4 Чому було використано саме PostgreSQL

Отже, в підсумку PostgreSQL чудовою базою даних для розробки проекту роблять:

- *Типи даних.* PostgreSQL підтримує всі необхідні типи даних.
- *Цілісність даних.* PostgreSQL забезпечує цілісність даних, вводячи обмеження та регулюючи дані.
- *Продуктивність.* Паралелізація запитів на читання, потужні методи індексації, контроль паралельності Multiversion. Це лише деякі з численних функцій, реалізованих PostgreSQL для підвищення та оптимізації його роботи.
- *Відновлення після катастрофи та надійність.* PostgreSQL дбає про забезпечення найвищого рівня надійності ваших даних. Завдяки вишуканим варіантам реплікації ваші дані абсолютно безпечні. Крім того, завжди можливо створити резервну копію найціннішої інформації.
- *Розширюваність.* У цій базі даних не потрібно обмежуватися певними типами документів. База даних пропонує широкий вибір типів даних для вашого розпорядження.
- *Інтернаціоналізація та пошук тексту.* PostgreSQL підтримує міжнародні набори символів, що також є доступним в повнотекстовому пошуку, щоб пришвидшити сам процес пошуку, та інтегрує нечутливі до регістру та нечутливі до наголосу порівняння.
- *Підтримка нереляційних даних.* Це, мабуть, найважливіше оновлення бази даних. Підтримка документів JSON, XML, Hstore та Cstore фактично перетворює PostgreSQL на базу даних NoSQL [20].

2.4 Розробка Frontend частини

Для відображення веб-сторінок у проекті були використані HTML та CSS.

«HTML» розшифровується як HyperText Markup language. В основному HTML використовується для розмітки веб-сторінки за допомогою характеристик HTML. HTML-код зберігається в файлах розширення html, що може відображатись будь-яким розвинутим браузером (Google Chrome, Firefox тощо).

Переваги HTML:

- Файли HTML – це текстові файли, тому їх можна редагувати на будь-якому типі комп'ютера;
- Кожен браузер підтримує мову HTML;
- Простота у вивченні та використанні;
- Майже всі засоби розробки підтримують HTML;
- Він безкоштовний і за замовчуванням поставляється в кожній Windows [21, 22].

CSS, каскадні таблиці стилів, визначається як мова таблиці методів, яка забезпечує контроль над тим, як веб-сайт взаємодіє з веб-браузерами, включаючи форматування та відображення їх HTML-документів.

Переваги CSS:

- CSS відіграє важливу роль, використовуючи CSS, просто потрібно один раз вказати повторюваний стиль для елемента і використовувати його кілька разів, оскільки CSS автоматично застосовуватиме необхідні стилі;
- головною перевагою CSS є те, що стиль застосовується послідовно на різних сайтах;

- потрібно використовувати кілька рядків програмування для кожної сторінки, що покращує швидкість сайту;
- не тільки спрощує розробку веб-сайту, але й спрощує обслуговування, оскільки зміна одного рядка коду впливає на весь веб-сайт;
- простота використання, тому зусилля на розробку значно зменшуються;
- допомагає формувати спонтанні та послідовні зміни;
- CSS має силу для повторного позиціонування. Це допомагає визначити зміни в позиції веб-елементів, які є на сторінці;
- економія пропускної здатності;
- користувачеві легко налаштувати Інтернет-сторінку;
- зменшує розмір передачі файлів [23].

РОЗДІЛ 3. ОСНОВНА СТРУКТУРА ПРОЕКТУ

3.1 Архітектура системи

Розроблена система має клієнт-серверну архітектуру (рисунок 3.1). Архітектура має клієнт для взаємодії з користувачем та сервер для надання функції або послуги одному або кільком клієнтам.

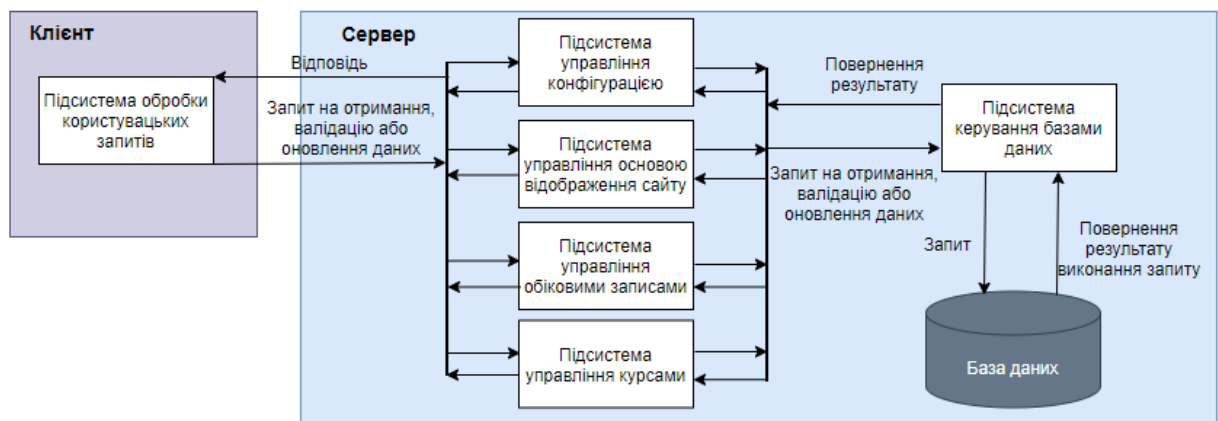


Рисунок 3.1 – Архітектура даної веб-системи

3.2 Можливості незареєстрованого користувача

Веб-система зустрічає незареєстрованого користувача сторінкою «Головна» (рисунок 3.2).

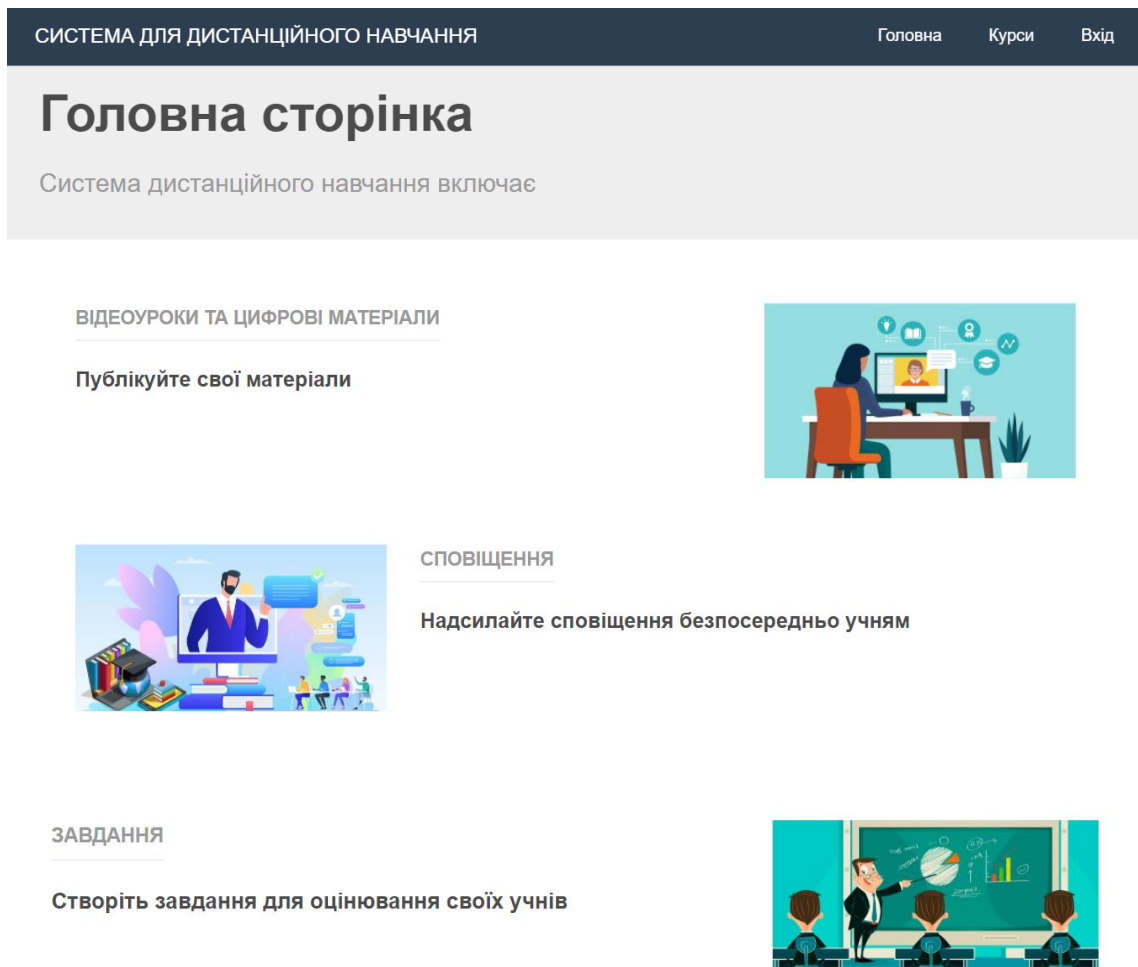


Рисунок 3.2 – Сторінка «Головна»

Зверху сторінки можна побачити блок з посиланнями на сторінки «Головна», «Курси» та «Вхід».

Незареєстрований користувач може переходити на сторінку «Курси» та переглядати список курсів (рисунок 3.3).

ПЕРЕЛІК КУРСІВ

Список доступних курсів



АЛГОРИТМИ

Цей курс охоплює важливу інформацію, яку кожен серйозний програміст повинен знати про алгоритми та структури даних, з акцентом на додатки та науковий аналіз продуктивності реалізацій Java.



ОСНОВИ ПРОГРАМУВАННЯ НА PYTHON

Метою курсу є вивчення основних конструкцій мови Python, які стануть в нагоді при вирішенні широкого кола завдань - від аналізу даних до розробки нових програмних продуктів.



ПРОГРАМУВАННЯ НА JAVA

Зробіть свій перший крок до кар'єри в розробці програмного забезпечення завдяки цьому введенню в Java - одну з найбільш затребуваних мов програмування та основу операційної системи Android.*

Рисунок 3.3 – Сторінка «Курси»

Натиснувши на блок одного з курсів зі сторінки «Курси», користувач переходить на сторінку з детальнішою інформацією про обраний курс (рисунок 3.4). На даній сторінці також присутні кнопка «Приєднатися до курсу» та можливість задати запитання щодо курсу в формі знизу. Оскільки даний користувач є неавторизованим, то, при натисканні на кнопку «Приєднатися до курсу», користувача перенесе на сторінку авторизації.

Програмування на Java

Зробіть свій перший крок до кар'єри в розробці програмного забезпечення завдяки цьому введенню в Java - одну з найбільш затребуваних мов програмування та основу операційної системи Android.

[Приєднатися до курсу](#)

ПРО КУРС

Розроблена для початківців, ця спеціалізація навчить вас основним концепціям програмування та забезпечить написання програм для вирішення складних проблем. Крім того, ви отримаєте основні навички, необхідні інженеру програмного забезпечення для вирішення реальних проблем, від проектування алгоритмів до тестування та налагодження ваших програм.

Проект прикладного навчання

Орієнтована на проекти, ця спеціалізація допоможе вам створити портфоліо робіт, щоб продемонструвати свої нові навички програмування. У головному камені ви створите механізм рекомендацій, подібний до тих, що використовуються Netflix або Amazon. Додаткові проекти у вашому портфоліо включатимуть: інтерактивну веб-сторінку, яка застосовує фільтри до зображень, аналіз файлів CSV, програму шифрування та генератор прогнозного тексту.



Сумніви?

[Зв'яжіться з нами](#)

Задайте своє запитання щодо курсу

Ім'я:

E-mail:

Повідомлення :

[Надіслати](#)

Рисунок 3.4 – Сторінка з детальною інформацією про курс

Таким чином неавторизований користувач може:

- переглядати головну сторінку;
- переглядати список наявних курсів;
- переглядати опис курсів;
- надіслати запитання щодо курсу;

- зареєструватися на платформі;
- авторизуватися.

3.3 Реєстрація та вхід в обліковий запис

При натисканні на кнопку «Вхід» зверху сторінки або «Приєднатися до курсу» на сторінці одного з курсів неавторизований користувач переходить на сторінку авторизації (рисунок 3.5).

Якщо користувач має обліковий запис в системі, то він може авторизуватися, ввівши коректні дані в поля «Ім'я користувача» та «Пароль» (рисунок 3.5). Ввівши некоректні дані входу та натиснувши «Увійти», під надписом «Введіть дані для авторизації» користувач побачить повідомлення червоним текстом «Введіть правильне ім'я користувача та пароль. Зауважте, що обидва поля можуть бути чутливими до регістру літер.» (рисунок 3.6).

Якщо користувач не має облікового запису, то є можливість зареєструватись. Для цього потрібно натиснути на посилання «Зареєструватися» під кнопкою «Увійти» (рисунок 3.7).

СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ Головна Курси Вхід

Введіть свої дані для авторизації

Ім'я користувача:

Пароль:

[Увійти](#)

[Ви зареєстровані? Зареєструйтесь](#)
[Забули пароль? Відновити пароль](#)

Рисунок 3.5 – Сторінка авторизації

Введіть правильне ім'я користувача та пароль. Зауважте, що обидва поля можуть бути чутливими до регістру літер.

Рисунок 3.6 – Повідомлення при невірному введенні даних

Далі система перенаправить користувача на сторінку реєстрації (рисунок 3.7), де він може, ввівши власні дані в поля та натиснувши «Зареєструватись», зареєструватись.

СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ Головна Курси Вхід

Введіть свої дані

Ім'я користувача:

E-mail:

Пароль:

Підтвердження паролю:

Рисунок 3.7 – Сторінка реєстрації

Якщо в поля «Пароль» та «Підтвердження паролю» введені різні дані, з'явиться повідомлення «Некоректне підтвердження» над кнопкою «Зареєструватись» (рисунок 3.8).

Ім'я користувача:

E-mail:

Пароль:

Підтвердження паролю:

Некоректне підтвердження

[Зареєструйтесь](#)

Рисунок 3.8 – Повідомлення при некоректному підтвердженні пароля

Якщо електронна адреса буде введена не вірно, то з'явиться повідомлення, що сповістить про це (рисунок 3.9).

E-mail:

! Електронна адреса має містити знак "@". В електронній адресі "userknu.ua" знака "@" немає.

Рисунок 3.9 – Повідомлення при невірному введенні електронної адреси

При коректності введення даних реєстрації користувач стає зареєстрованим та автоматично переходить на сторінку «Головна» (рисунок 3.10), де на верхньому блоці з'являються посилання «Мої курси» і «Вихід» та зникає «Вхід». Аналогічно відбувається, якщо користувач успішно авторизується.

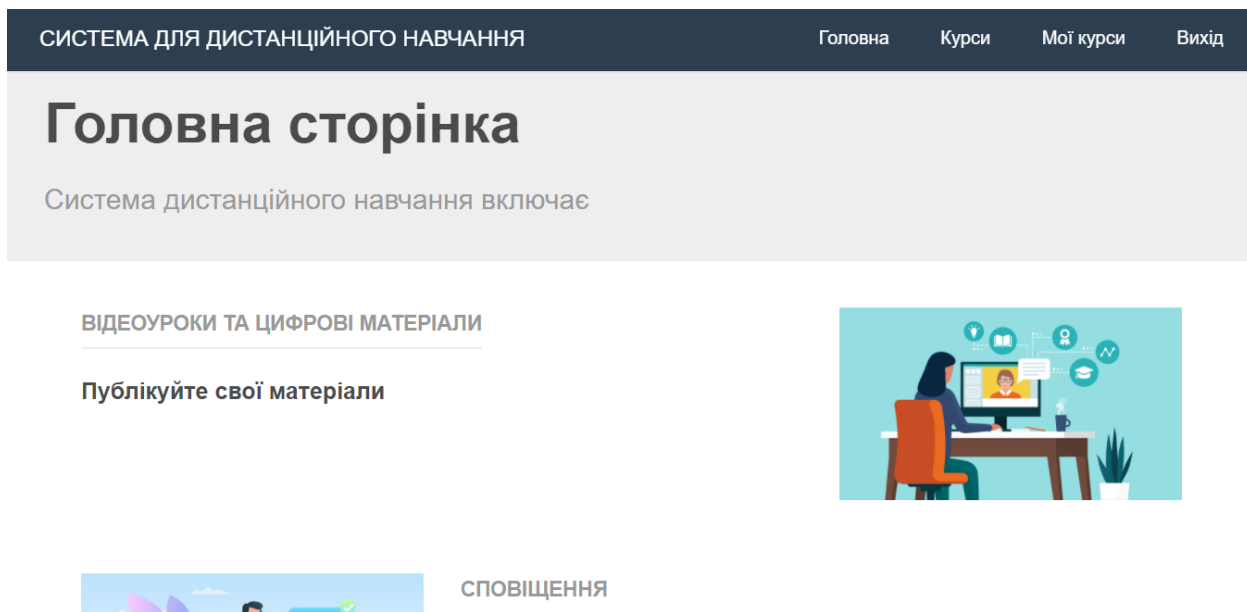


Рисунок 3.10 – Головна сторінка авторизованого користувача

3.4 Можливості «Студента»

Аналогічно до незареєстрованого користувача користувач «Студент» може перейти на сторінку «Курси» (рисунок 3.11) та обрати курс із переліку та потрапити на сторінку обраного курсу (рисунок 3.11). Ознайомившись з деталями курсу, користувач може приєднатися до курсу, натиснувши «Приєднатися до курсу» (рисунок 3.12).



Рисунок 3.11 – Сторінка «Курси» авторизованого користувача

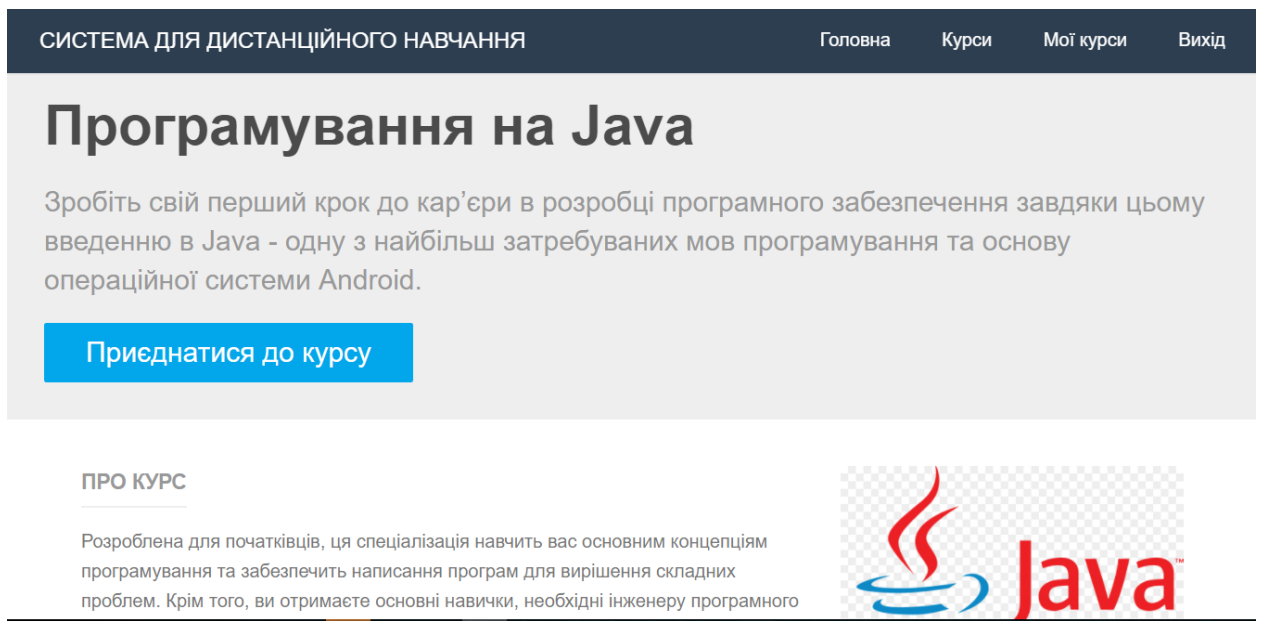


Рисунок 3.12 – Сторінка курсу авторизованого користувача

Після виконання даної дії користувач переміщається на сторінку «Мої курси» (рисунок 3.13), де в списку, в правій частині сторінки, «Мої курси» може побачити доданий курс, до якого він щойно приєднався. Також короткий список доданих курсів можна побачити зліва блоці «Моя інформаційна панель» в списку «Курси». В цьому ж блоці «Студент» може вибрати зі списку

«Налаштування акаунта» пункти «Редагувати інформацію» та «Змінити пароль».

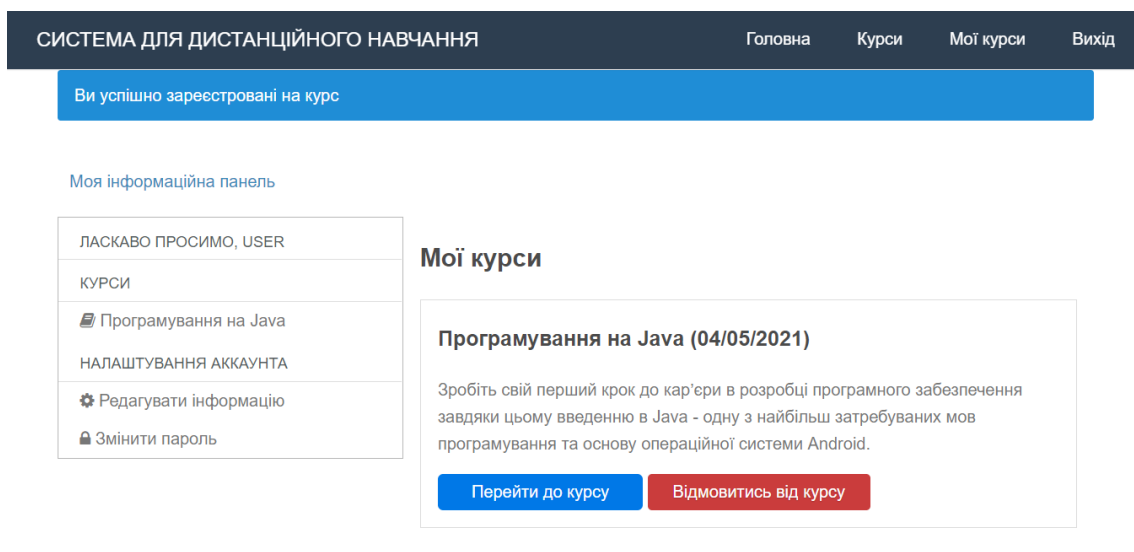


Рисунок 3.13 – Сторінка «Мої курси»

При натисканні на пункт «Редагувати інформацію» «Студент» переходить на сторінку, де може змінити дані свого облікового запису, заповнивши потрібні поля та натиснувши «Зберегти» (рисунок 3.14).

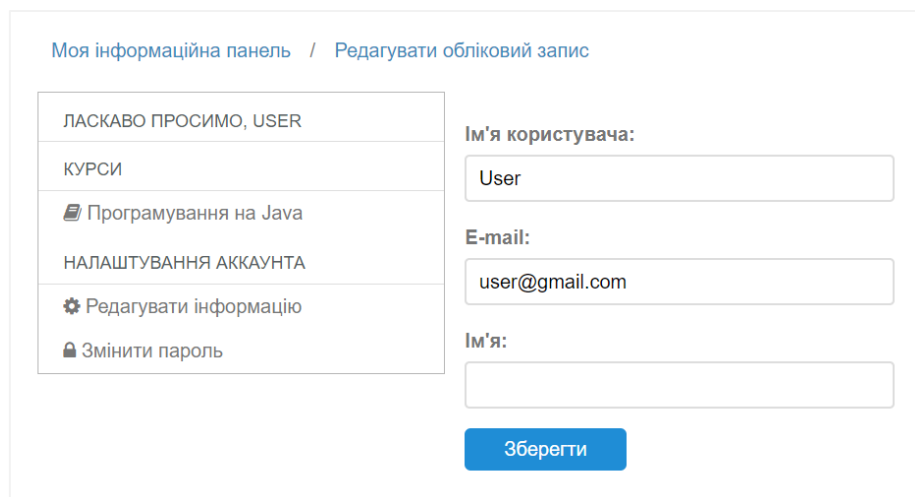
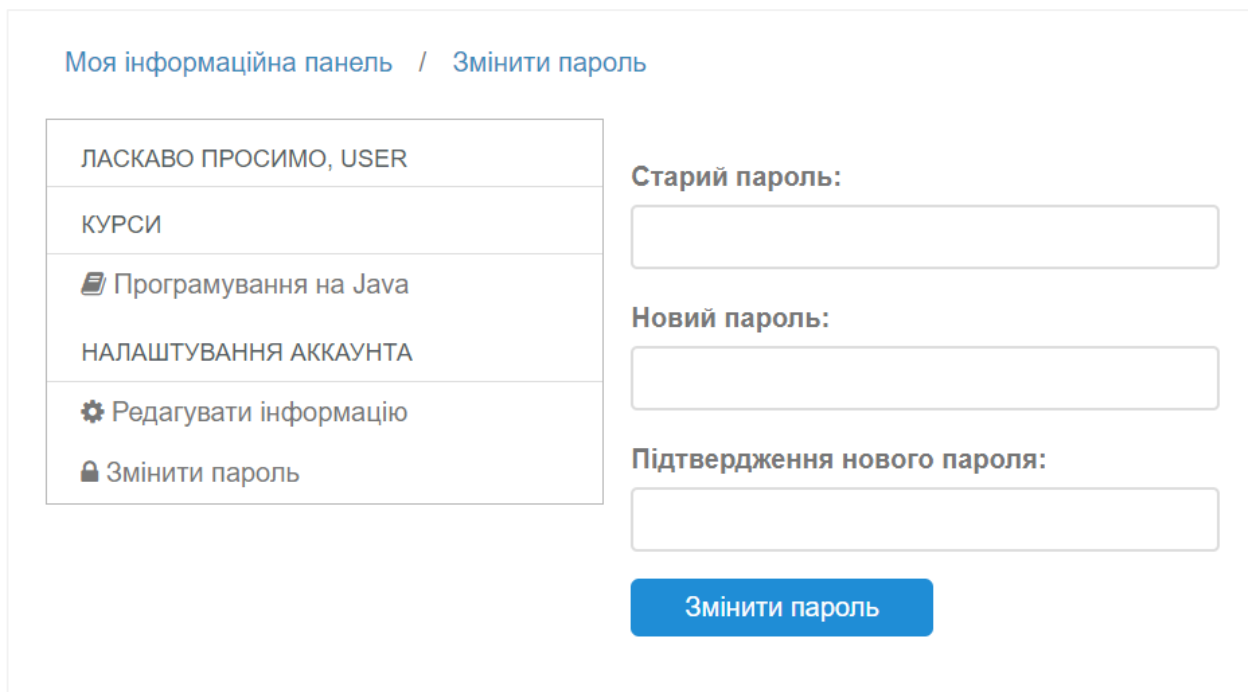


Рисунок 3.14 - «Редагувати інформацію»

При натисканні на пункт «Змінити пароль» користувач потрапляє на сторінку, де має можливість змінити власний пароль, ввівши запрошені дані та натиснувши «Змінити пароль» (рисунок 3.15).



The screenshot shows a user dashboard with a breadcrumb trail: "Моя інформаційна панель / Змінити пароль". On the left is a sidebar menu with items: "ЛАСКАВО ПРОСИМО, USER", "КУРСИ", "Програмування на Java", "НАЛАШТУВАННЯ АККАУНТА", "Редагувати інформацію", and "Змінити пароль". The main content area contains three password input fields labeled "Старий пароль:", "Новий пароль:", and "Підтвердження нового пароля:". A blue button labeled "Змінити пароль" is positioned below the fields.

Рисунок 3.15 – Форма для зміни пароля

Також «Студент» може повернутися на сторінку «Мої курси». Далі, якщо натиснути на назву курсу в списку зліва сторінки або на кнопку «Перейти до курсу» в блоці потрібного курсу (рисунок 3.13), то можна перейти на сторінку обраного курсу (рисунок 3.16). Після переходу на обрану сторінку користувач може побачити сторінку «Сповідання», де знаходяться сповідання залишені «Викладачем» в даному курсі, та кнопка «Перейти до чату». Натиснувши «Перейти до чату» (рисунок 3.16), користувач може перейти до чату курсу (рисунок 3.17).

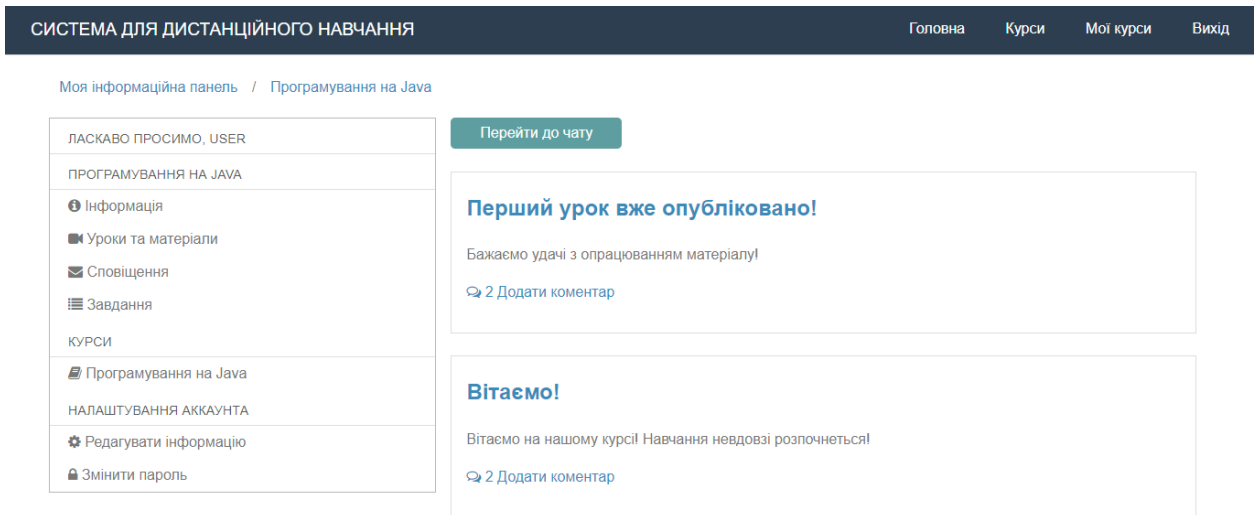


Рисунок 3.16 – Сторінка «Сповідення»

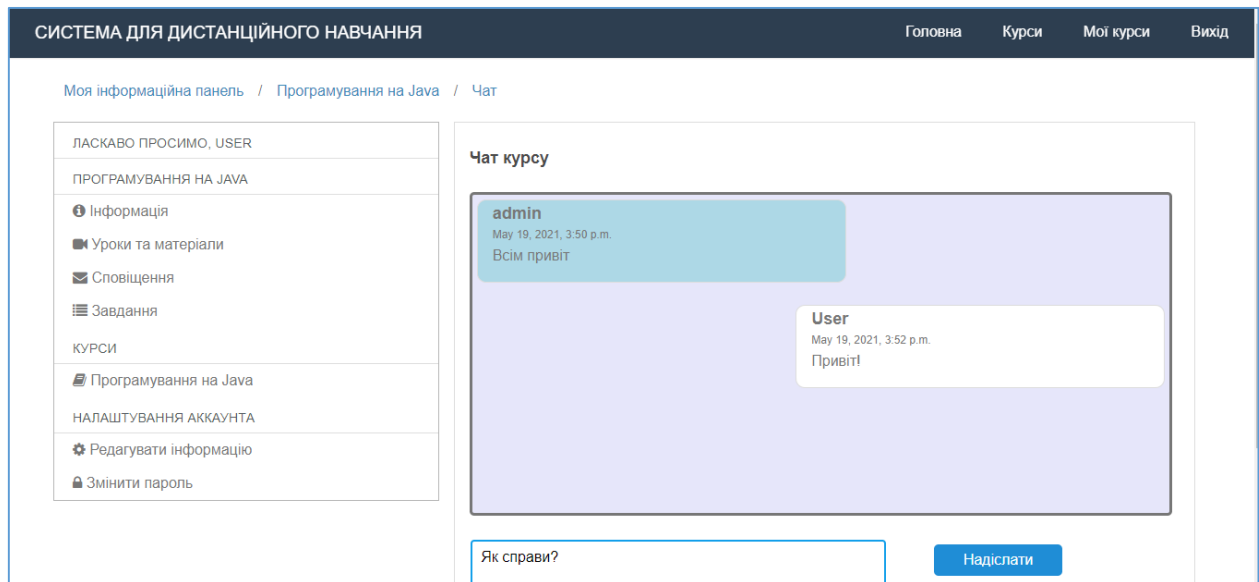


Рисунок 3.17 – Сторінка «Чат»

При натисканні на один із блоків можна переглянути інформацію сповіщення детальніше та переглянути коментарі (рисунок 3.18).

Моя інформаційна панель / Програмування на Java / Перший урок вже опубліковано!

ЛАСКАВО ПРОСИМО, USER
ПРОГРАМУВАННЯ НА JAVA
📌 Інформація
📖 Уроки та матеріали
✉ Сповідання
📋 Завдання
КУРСИ
📖 Програмування на Java
📖 Основи програмування на Python
НАЛАШТУВАННЯ АККАУНТА
⚙ Редагувати інформацію
🔒 Змінити пароль

Перший урок вже опубліковано!

Бажаємо удачі з опрацюванням матеріалу!

Коментарі

[Прокоментуйте](#)

Ангеліна надіслав 1 week, 1 day тому:
Чудово! Приступаю до навчання!

user надіслав 1 week, 1 day тому:
Добре, вже переглянув.

Рисунок 3.18 – Сторінка обраного сповіщення

Натиснувши на «Прокоментуйте» або прогорнувши сторінку вниз, можна потрапити до форми для написання коментарів та відповідно додати власний коментар, ввівши текст в поле та натиснувши «Надіслати» (рисунок 3.19).

Прокоментуйте

Ангеліна надіслав 1 week, 1 day тому:
Чудово! Приступаю до навчання!

user надіслав 1 week, 1 day тому:
Добре, вже переглянув.

Коментувати:

Цікавий урок! Вже недочекаюсь наступних!

[Надіслати](#)

Рисунок 3.19 – Введення та додавання коментаря

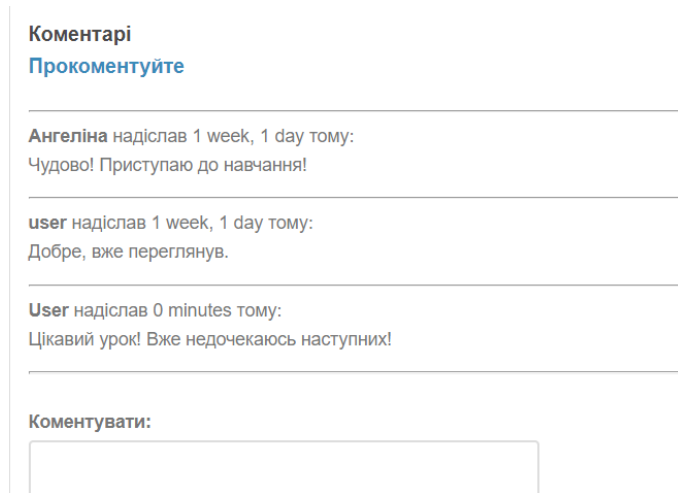


Рисунок 3.20 – Результат надсилання коментаря з рисунку 3.19

Натиснувши на пункт «Інформація» можна потрапити на сторінку «Інформація» поточного курсу та переглянути інформацію про курс (рисунок 3.21).

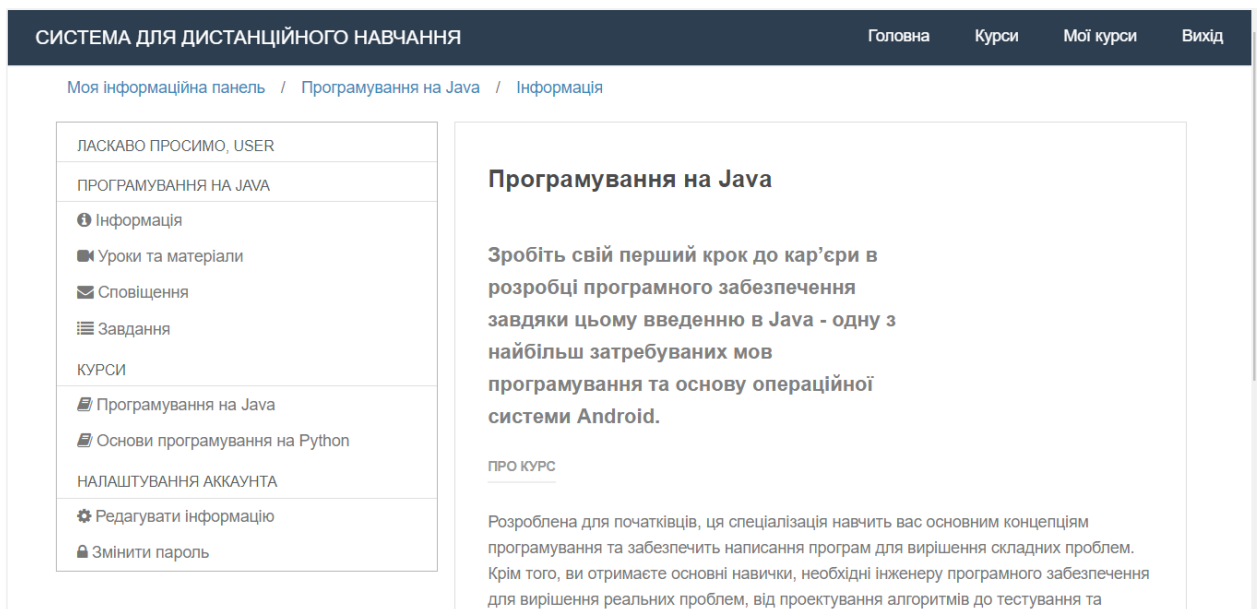
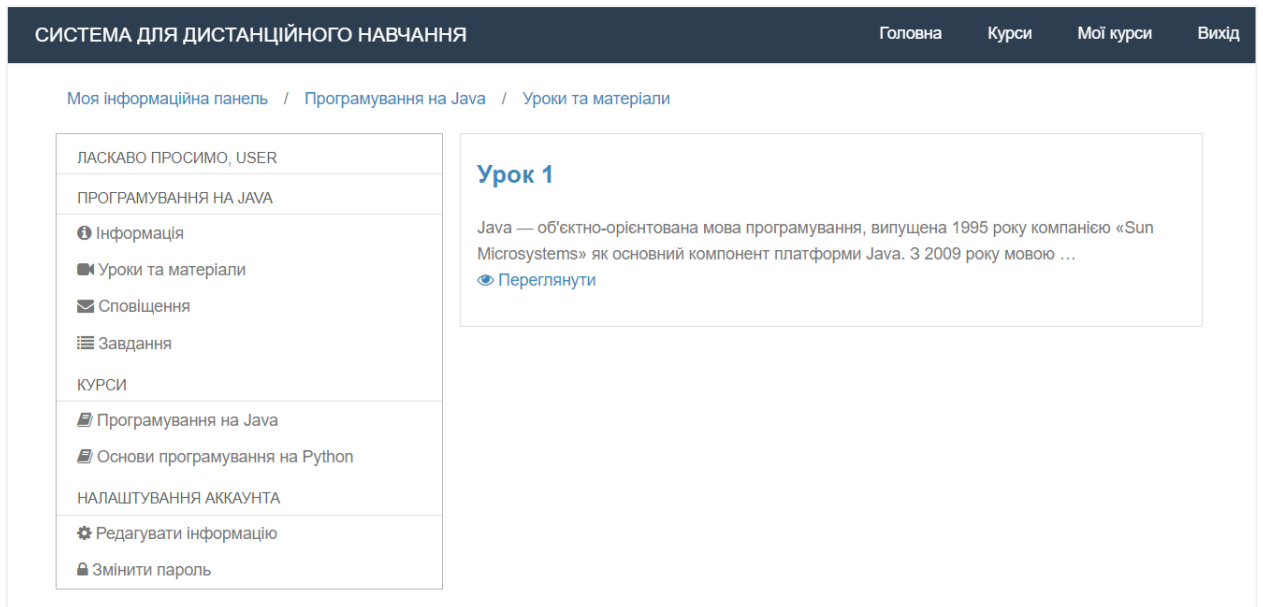


Рисунок 3.21 – Сторінка «Інформація»

Натиснувши на пункт «Уроки та матеріали» можна потрапити на сторінку «Уроки та матеріали» поточного курсу (рисунок 3.22). На даній сторінці відображається список уроків.



The screenshot displays a web interface for a distance learning system. At the top, there is a dark navigation bar with the text 'СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ' on the left and 'Головна', 'Курси', 'Мої курси', and 'Вихід' on the right. Below this, a breadcrumb trail reads 'Моя інформаційна панель / Програмування на Java / Уроки та матеріали'. On the left side, there is a vertical menu with the following items: 'ЛАСКАВО ПРОСИМО, USER', 'ПРОГРАМУВАННЯ НА JAVA', 'Інформація', 'Уроки та матеріали', 'Сповідання', 'Завдання', 'КУРСИ', 'Програмування на Java', 'Основи програмування на Python', 'НАЛАШТУВАННЯ АККАУНТА', 'Редагувати інформацію', and 'Змінити пароль'. The main content area on the right features a lesson card titled 'Урок 1'. The card contains the text: 'Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою ...' and a blue link with an eye icon labeled 'Переглянути'.

Рисунок 3.22 – сторінка «Уроки та матеріали»

При натисканні на блок одного із уроків користувач переходить до сторінки даного уроку (рисунок 3.23). На ній відображаються назва уроку, опис уроку та таблиця із назвою та посиланнями на матеріали. Щоб переглянути матеріал потрібно натиснути на посилання «Переглянути» навпроти потрібного матеріалу.

СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ Головна Курси Мої курси Вихід

Моя інформаційна панель / Програмування на Java / Уроки та матеріали / Урок 1

ЛАСКАВО ПРОСИМО, USER

ПРОГРАМУВАННЯ НА JAVA

- 📄 Інформація
- 📖 Уроки та матеріали
- ✉️ Сповідання
- 📋 Завдання

КУРСИ

- 📖 Програмування на Java
- 📖 Основи програмування на Python

НАЛАШТУВАННЯ АККАУНТА

- ⚙️ Редагувати інформацію
- 🔒 Змінити пароль

Урок 1

Java — об'єктно-орієнтована мова програмування, випущена 1995 року компанією «Sun Microsystems» як основний компонент платформи Java. З 2009 року мовою займається компанія «Oracle», яка того року придбала «Sun Microsystems». В офіційній реалізації Java-програми компілюються у байт-код, який при виконанні інтерпретується віртуальною машиною для конкретної платформи.

Сьогодні ми розберемо основний синтаксис Java.


Матеріал уроку

Назва	Дія
Основний синтаксис. Відеоурок	Переглянути
Основний синтаксис. Текстовий матеріал	Переглянути

Рисунок 3.23 – Сторінка обраного уроку з матеріалами

Обравши бажаний матеріал із таблиці, користувач може побачити на сторінці назву матеріалу в заголовку, прикріплене відео (рисунок 3.14) або посилання на текстовий документ (рисунок 3.15), короткий опис матеріалу та посилання «Повернутись» для того щоб знову потрапити до поточного уроку.

Основний синтаксис. Відеоурок



Об'єкт - об'єкти мають стан і поведінку. Наприклад: собака може мати стан - колір, ім'я, а також і поведінка - кивати, бігти, гавкати, їсти. Об'єкт є екземпляром класу. Клас - може бути визначений як шаблон, який описує поведінку об'єкта. Метод - є в основному

Рисунок 3.24 – Матеріал уроку типу відео

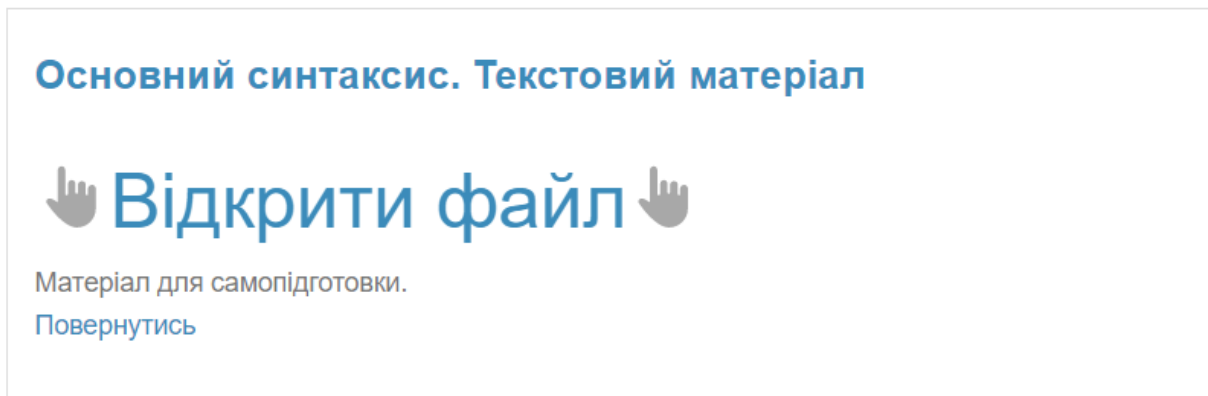


Рисунок 3.25 – Матеріал уроку типу текстовий документ

Для перевірки засвоєння навчальних матеріалів курсу, користувач може обрати завдання із списку вкладки «Завдання» (рисунок 3.26)

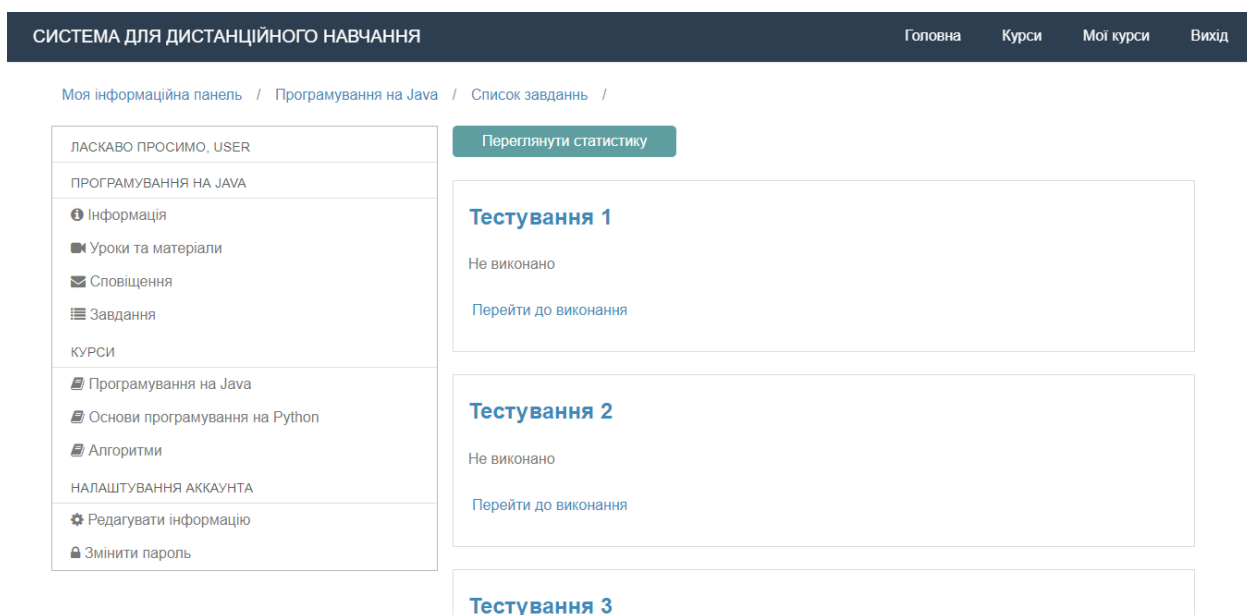


Рисунок 3.26 – Відкрита вкладка «Завдання» зі списком завдань та посиланням на сторінку зі статистикою їх виконання

Обравши одне із завдань, користувач може пройти тестування: відмітити відповіді на питання та підтвердити їх, натиснувши «Відправити» (рисунок 3.27). В результаті цього користувач автоматично повернеться на сторінку зі

списком завдань, де побачить отриманий бал в блоці із завданням (рисунок 3.28)

СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ

[Головна](#)
[Курси](#)
[Мої курси](#)
[Вихід](#)

Моя інформаційна панель / Програмування на Java / Список завдань / Тестування /

ЛАСКАВО ПРОСИМО, USER

ПРОГРАМУВАННЯ НА JAVA

- 📄 Інформація
- 📖 Уроки та матеріали
- ✉️ Сповідження
- 📋 Завдання

КУРСИ

- 📖 Програмування на Java
- 📖 Основи програмування на Python
- 📖 Алгоритми

НАЛАШТУВАННЯ АККАУНТА

- ⚙️ Редагувати інформацію
- 🔒 Змінити пароль

Тестування 1

Деяка сутність у віртуальному просторі, що має певний стан і поведінку, задані значення властивостей (атрибутів) та операцій над ними (методів), це:

Клас

Об'єкт

Атрибут класу

Java -

високорівнева мова програмування

низькорівнева мова програмування

Розрядність типу int

64

32

16

Змінна, опис якої створює програміст при створенні класу . Всі дані об'єкта зберігаються в його полях. Доступ до полів здійснюється за їх імені, це:

Клас

Об'єкт

Атрибут класу

Відправити

Рисунок 3.27 – Відкрите завдання «Тестування 1» (Сторінка із тестовими завданнями)

СИСТЕМА ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ Головна Курси Мої курси Вихід

Моя інформаційна панель / Програмування на Java / Список завдань /

ЛАСКАВО ПРОСИМО, USER

ПРОГРАМУВАННЯ НА JAVA

- Інформація
- Уроки та матеріали
- Сповідання
- Завдання

КУРСИ

- Програмування на Java
- Основи програмування на Python
- Алгоритми

НАЛАШТУВАННЯ АККАУНТА

- Редагувати інформацію
- Змінити пароль

Переглянути статистику

Тестування 1

Складено на: 75%

Тестування 2

Не виконано

[Перейти до виконання](#)

Тестування 3

Рисунок 3.28 – Оновлена сторінка зі списком завдань

Загальний прогрес проходження курсу користувач може побачити, натиснувши «Переглянути статистику» (рисунок 3.28). В наслідок цього користувач потрапить на сторінку зі статистикою (рисунок. 3.29), де відображені оцінки за пройдені завдання, середня успішність виконання завдань курсу та загальний рейтинг учасників курсу.

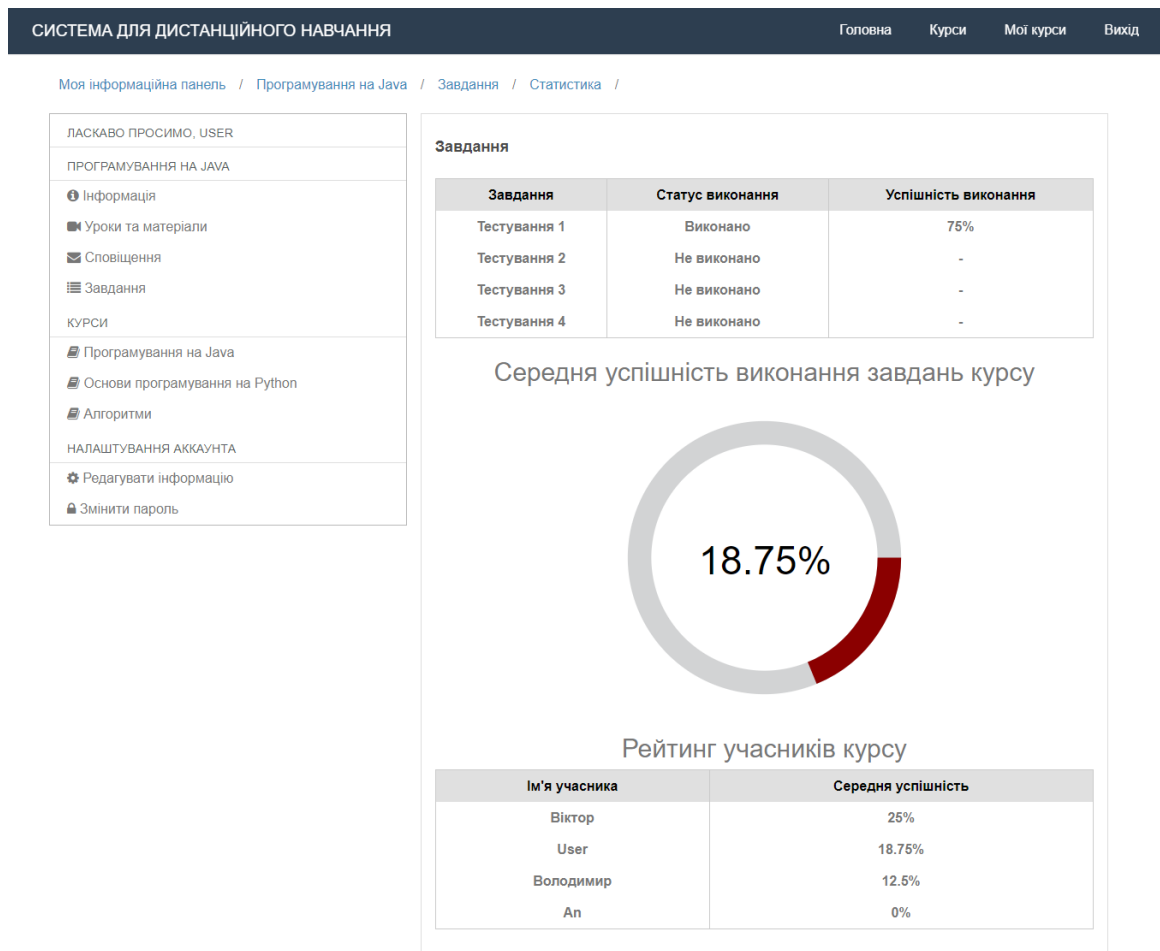


Рисунок 3.29 – Сторінка із статистикою проходження курсу

Якщо користувач бажає покинути курс, то він має знову перейти на сторінку «Мої курси» (рисунок 3.13) та натиснути кнопку «Відмовитися від курсу», після чого підтвердити своє бажання, натиснувши «Підтвердити» (рисунок 3.30)

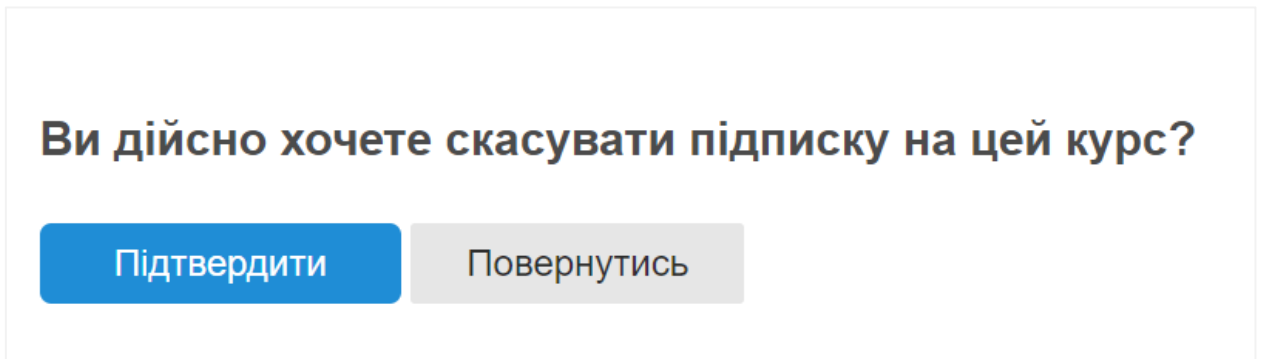


Рисунок 3.30 – Сторінка скасування підписки на курси

Отже, можливості «Студента»:

- підписуватись та скасовувати підписку на навчальні курси;
- переглядати навчальні курси;
- переглядати уроки до курсів;
- переглядати матеріали до уроків;
- переглядати оголошення;
- коментувати оголошення та спілкуватися в чаті;
- переглядати завдання;
- проходити тестування;
- переглядати статистику;
- вийти з системи.

3.5 Можливості «Викладача»

«Викладач» має всі можливості «Студента» та доступ до часткового редагування сайту.

На сторінці для редагування (рисунок 3.31) «Викладач» може побачити в таблиці зліва дванадцять вкладок («Вибір відповіді», «Відповіді», «Вікторини», «Зарахування», «Коментарі», «Коректність відповіді», «Курси», «Матеріали», «Оголошення», «Оцінки», «Питання», «Уроки») для

редагування даних сайту. Також на цій же сторінці, в блоці справа, відображається історія змін інформації сайту, зроблена даним користувачем.

Site administration

COURSES		
Вибір відповіді	+ Add	Change
Відповіді	+ Add	Change
Вікторини	+ Add	Change
Зарахування	+ Add	Change
Коментарі	+ Add	View
Коректність відповіді	+ Add	Change
Курси	+ Add	Change
Матеріали	+ Add	Change
Оголошення	+ Add	Change
Оцінки	+ Add	Change
Питання	+ Add	Change
Уроки	+ Add	Change

Recent actions

My actions

- ✘ Курс
Курс
- ✎ Перший урок вже опубліковано!
Оголошення
- ✎ Enrollment object (13)
Зарахування
- ✘ Comment object (3)
Коментар
- + Урок 1
Урок
- ✎ Програмування на Java
Курс
- ✎ Анастасія
Користувач
- ✎ user
Користувач
- ✎ Володимир
Користувач
- ✎ Володимир
Користувач

Рисунок 3.31 – Сторінка для редагування даних сайту

Для виконання дій над курсами сайту користувач має обрати вкладку «Курси» в таблиці. Після цієї дії «Викладач» потрапляє до сторінки зі списком курсів (рисунок 3.32).

Користувач може перейти до створення нового курсу, натиснувши на «Додати курс» (рисунок 3.32).

Home · Courses · Курси

COURSES

- Вибір відповіді + Add
- Відповіді + Add
- Вікторини + Add
- Зарахування + Add
- Коментарі + Add
- Коректність відповіді + Add
- Курси + Add**
- Матеріали + Add
- Оголошення + Add
- Оцінки + Add
- Питання + Add
- Уроки + Add

Select course to change ADD COURSE +

Q Search

Action: Go 0 of 3 selected

<input type="checkbox"/>	ІМ'Я	ПОСИЛАННЯ	ДАТА ПОЧАТКУ	СТВОРЕНО
<input type="checkbox"/>	Алгоритми	algoritmi	May 15, 2021	May 14, 2021, 5:16 a.m.
<input type="checkbox"/>	Основи програмування на Python	osnovi-programuvannya-na-python	May 12, 2021	May 26, 2021, 4:45 a.m.
<input type="checkbox"/>	Програмування на Java	programuvannya-na-java	May 4, 2021	May 26, 2021, 4:54 a.m.

3 курси

Рисунок 3.32 – Вкладка «Курси»

Щоб додати курс користувач має заповнити дані сторінки (рисунок 3.5.3) та підтвердити введені дані, натиснувши «Зберегти».

Головна · Курси · Курси · Додати курс

КУРСИ

- Вибір відповіді + Додати
- Відповідь + Додати
- Вікторини + Додати
- Зарахування + Додати
- Коментарі + Додати
- Коректність відповіді + Додати
- Курси + Додати**
- Матеріали + Додати
- Оголошення + Додати
- Оцінки + Додати
- Питання + Додати
- Уроки + Додати

Додати курс

Ім'я:

Посилання:

Короткий опис:

Про курс:

Дата початку: Сьогодні | 📅
Примітка. Ви випереджаєте час сервера на 3 години.

Зображення: Файл не вибрано

Рисунок 3.33 – Сторінка для створення нового курсу

Натиснувши на назву курсу (рисунок 3.32), користувач може перейти на сторінку з інформацією про даний курс (рисунок 3.34) та змінити інформацію. можна видалити даний курс, натиснувши «Видалити» (рисунок 3.34).

Для видалення курсу зі списку «Вчитель» може поставити прапорець навпроти назви курсу або курсів та, вибравши із списку дій «Видалити обрані курси», підтвердити видалення (рисунок 3.32).

Головна > Курси > Курси > Програмування на Java

КУРСИ

- Вибір відповіді + Додати
- Відповідь + Додати
- Вікторини + Додати
- Зрахування + Додати
- Коментарі + Додати
- Коректність відповіді + Додати
- Курси** + Додати
- Матеріали + Додати
- Оголошення + Додати
- Оцінки + Додати
- Питання + Додати
- Уроки + Додати

Змінити курс

Програмування на Java ІСТОРИЯ ПЕРЕГЛЯНУТИ НА САЙТІ >

Ім'я:

Посилання:

Короткий опис:

Про курс:

Дата початку:
Примітка. Ви випереджаєте час сервера на 3 години.

Зображення:
зміню:

Рисунок 3.34 – Сторінка редагування одного з курсів

Аналогічно до курсів «Викладач» може створювати, редагувати та видаляти всі інші елементи сайту, назва яких прямо відповідає їх значенню, що знаходяться в таблиці зліва, окрім коментарів, їх він має право лише створювати та видаляти.

Підсумок

«Викладач», окрім можливостей «Студента», також може:

- створювати, редагувати та видаляти навчальні курси;
- створювати, редагувати та видаляти уроки до курсів;
- створювати, редагувати та видаляти матеріали до уроків;
- створювати, редагувати та видаляти оголошення;
- створювати та видаляти коментарі під оголошеннями та повідомлення чату;
- створювати, редагувати та видаляти завдання, питання та відповіді;
- створювати, редагувати та видаляти результати вибору відповідей інших учасників;
- міняти статус зарахування користувачів на курс.

3.6 Можливості «Адміністратора»

Адміністратор має всі права «Викладача» (рисунок 3.35), а також можливість редагувати коментарі та повідомлення інших користувачів, створювати, видаляти акаунти користувачів, редагувати інформацію з їх облікових записів, надавати та позбавляти користувачів прав «Викладачів» та «Адміністраторів» (рисунок 3.36).

Django administration

Site administration

The screenshot shows the Django administration interface. On the left is a navigation menu with three main sections: ACCOUNTS, AUTHENTICATION AND AUTHORIZATION, and COURSES. The ACCOUNTS section is expanded to show 'Користувачі' (Users) with '+ Add' and 'Change' links. The AUTHENTICATION AND AUTHORIZATION section shows 'Groups' with '+ Add' and 'Change' links. The COURSES section lists various items like 'Вибір відповіді', 'Відповіді', 'Вікторини', etc., each with '+ Add' and 'Change' links. On the right is a 'Recent actions' sidebar titled 'My actions' listing recent operations such as 'An Користувач', 'teacher Group', and several 'Exercisitem object' and 'Answer object' actions.

Рисунок 3.35 – Сторінка «Адміністратора»

The screenshot shows the 'Users' management page in the Django administration interface. The breadcrumb trail is 'Home > Accounts > Користувачі'. The left sidebar is the same as in Figure 3.35. The main content area is titled 'Select Користувач to change' and includes an 'ADD КОРИСТУВАЧ +' button. Below the title is an 'Action:' dropdown menu, a 'Go' button, and a selection count '0 of 9 selected'. A table lists the users with checkboxes for selection and columns for 'SUPERUSER STATUS' and 'IS STAFF?'. The users listed are User2, Anna, Victor, User, user, Anastasia, Volodymyr, teacher, and admin.

	SUPERUSER STATUS	IS STAFF?
<input type="checkbox"/> User2	✘	✘
<input type="checkbox"/> Anna	✘	✔
<input type="checkbox"/> Victor	✘	✘
<input type="checkbox"/> User	✘	✘
<input type="checkbox"/> user	✘	✘
<input type="checkbox"/> Anastasia	✘	✘
<input type="checkbox"/> Volodymyr	✘	✘
<input type="checkbox"/> teacher	✘	✔
<input type="checkbox"/> admin	✔	✔

Рисунок 3.36 – Відкрита вкладка на сторінці «Адміністратора»

«Користувачі» зі списком користувачів сайту

ВИСНОВКИ

Оскільки глобальна мережа Інтернет стрімко розвивається, набуває популярності та стає все більш доступною для широких мас, то найбільш корисну та ефективну систему навчання варто розробляти в форматі веб-системи або веб-додатка. Така система здатна охопити якнайширшу аудиторію. При ефективній, надійній та практично безперервній роботі системи та адаптації навчальних матеріалів до неї процес навчання значно спроститься та стане більш результативним.

Веб-система потребує розширення та вдосконалення щоб бути конкурентоспроможною до типових видів навчання. Також, з рухом прогресу, винаходженні нових технологій розробки систем та зміні потреб суспільства кожна система потребуватиме модифікації щоб йти в ногу з часом, тому дана версія системи не є остаточною.

У роботі було виконано загальний огляд дистанційної освіти, виділено переваги та недоліки здобуття освіти дистанційно, розглянуто технології розробки систем та порівняно їх між собою та обрано найоптимальніші, в результаті було розроблено веб-систему для здобуття дистанційної освіти.

ВИКОРИСТАНІ ДЖЕРЕЛА

1. Дистанційна освіта в Україні: історія та сучасний стан [Електронний ресурс] – Режим доступу: http://irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/Mir_2010_6_27.pdf
2. Дистанційна освіта. Дистанційне навчання. Дистанційна освіта - це можливість навчатися та отримувати необхідні знання віддалено від навчального закладу. [Електронний ресурс] – Режим доступу: <http://www.myshared.ru/slide/1394637/>
3. Дистанційна освіта в Україні: історія та сучасний стан [Електронний ресурс] – Режим доступу: https://www.academia.edu/43133254/%D0%94%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D1%86%D1%96%D0%B9%D0%BD%D0%B0_%D0%BE%D1%81%D0%B2%D1%96%D1%82%D0%B0_%D0%B2_%D0%A3%D0%BA%D1%80%D0%B0%D1%97%D0%BD%D1%96_%D1%96%D1%81%D1%82%D0%BE%D1%80%D1%96%D1%8F_%D1%82%D0%B0_%D1%81%D1%83%D1%87%D0%B0%D1%81%D0%BD%D0%B8%D0%B9_%D1%81%D1%82%D0%B0%D0%BD
4. ІСТОРІЯ РОЗВИТКУ ДИСТАНЦІЙНОЇ ОСВІТИ В СВІТІ [Електронний ресурс] – Режим доступу: http://irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?C21COM=2&I21DBN=UJRN&P21DBN=UJRN&IMAGE_FILE_DOWNLOAD=1&Image_file_name=PDF/Nz_p_2016_14_7_30.pdf
5. ДИСТАНЦІЙНЕ НАВЧАННЯ: ІСТОРІЯ ТА СУЧАСНІСТЬ [Електронний ресурс] – Режим доступу: <https://biblio.lib.kherson.ua/distantsiyne-navchannya-istoriya.htm>

6. ДИСТАНЦІЙНА ПЕДАГОГІЧНА ОСВІТА [Електронний ресурс] – Режим доступу:
http://ipood.com.ua/data/NDR/Information_technology/2012_Ogienko_recommended.pdf
7. ПРОБЛЕМИ ВИКОРИСТАННЯ ДИСТАНЦІЙНОЇ ФОРМИ НАВЧАННЯ В УМОВАХ СІЛЬСЬКОЇ МІСЦЕВОСТІ [Електронний ресурс] – Режим доступу: <http://2019.moodlemoot.in.ua/course/view.php?id=36>
8. Засоби дистанційної освіти. Методи дистанційного навчання. Види дистанційного навчання [Електронний ресурс] – Режим доступу: <https://santorpack.ru/uk/school-of-design/sredstva-distancionnogo-obrazovaniya-metody-distancionnogo.html>
9. СТРАТЕГІЯ РОЗВИТКУ ВИЩОЇ ОСВІТИ В УКРАЇНІ на 2021–2031 роки [Електронний ресурс] – Режим доступу: <https://mon.gov.ua/storage/app/media/rizne/2020/09/25/rozvitku-vishchoi-osviti-v-ukraini-02-10-2020.pdf>
10. Переваги та недоліки дистанційного навчання [Електронний ресурс] – Режим доступу: <https://kerivnyk.info/perevahy-ta-nedoliky-dystantsijnoho-navchannya>
11. PyCharm: IDE для Python [Електронний ресурс] – Режим доступу: <https://timeweb.com/ru/community/articles/pycharm-ide-dlya-python-1#:~:text=%D0%9F%D1%80%D0%B5%D0%B8%D0%BC%D1%83%D1%89%D0%B5%D1%81%D1%82%D0%B2%D0%B0%20PyCharm&text=PyCharm%20%D0%BF%D0%BE%D0%B7%D0%B2%D0%BE%D0%BB%D1%8F%D0%B5%D1%82%20%D0%B1%D1%8B%D1%81%D1%82%D1%80%D0%BE%20%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%B4%D0%B8%D1%82%D1%8C%20%D1%80%D0%B5%D1%84%D0%B0%D0%BA%D1%82%D0%BE%D1%80%D0%B8%D0%BD%D0%B3,PyGTK%20%D0%B8%20%D0%BC%D0%BD%D0%BE%D0%B3%D0%B8%D0%B5%20%D0%B4%D1%80%D1%83%D0%B3%D0%B8%D0%B5%20>

%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%8B.

12. Django: практика створення Web-сайтів на Python. / Дронов В. А. // БХВ-Петербург, 2016. – 528 с.
13. Django 3.0. Практика створення веб-сайтів на Python / Дронов В. А // ВHV-СПб, 2020 - 704 с.
14. Веб-фреймворк Django (Python) [Електронний ресурс] – Режим доступу: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django>
15. Керівництво Django Частина 8: Аутентифікація і авторизація користувача [Електронний ресурс] – Режим доступу: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Authentication>
16. Різниця між mysql і postgresql [Електронний ресурс] – Режим доступу: <https://svcministry.org/uk/dictionary/difference-between-mysql-and-postgresql/>
17. ПОРІВНЯННЯ SQL І NOSQL БАЗ ДАНИХ НА ПРИКЛАДІ ПРОЕКТУВАННЯ АФФІЛЕЙТ РЕПОРТ СИСТЕМ [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/339976138_PORIVNANNA_SQL_I_NOSQL_BAZ_DANIH_NA_PRIKLADI_PROEKTUVANNA_AFFILEJT_REPORT_SISTEM
18. PostgreSQL проти MongoDB: Порівняння баз даних [Електронний ресурс] – Режим доступу: <https://www.bmc.com/blogs/mongodb-vs-postgresql/>
19. PostgreSQL проти MySQL: критичні відмінності [Електронний ресурс] – Режим доступу: <https://www.xplenty.com/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/>
20. PostgreSQL: Documentation [Електронний ресурс] – Режим доступу: <https://www.postgresql.org/docs/>
21. Вивчаємо HTML 5 / Лоусон Б., Шарп Р. // Питер, 2011. – 253 с.

22. HTML & XHTML. Детальний керівництво / Ч. Муссіано, Б Кеннеді // 2014.
– 752 с.
23. HTML5, CSS3 и Web 2.0. Розробка сучасних Web-сайтів. / Дронов В. А. // БХВ, 2011. – 414 с.