

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «магістр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Веб-орієнтована система освітнього процесу
університету з використанням штучного інтелекту»**

Виконав студент групи ІПЗм-23-1
КРАВЧУК Дмитро Андрійович

Керівник роботи:
ЄФРЕМОВ Юрій Миколайович

Рецензент:
ВЛАСЕНКО Олег Васильович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ “ЖИТОМИРСЬКА ПОЛІТЕХНІКА”
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Завідувач кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«01» жовтня 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **КРАВЧУК Дмитро Андрійович**

Керівник роботи: **ЄФРЕМОВ Юрій Миколайович**

Тема роботи: **«Веб-орієнтована система освітнього порталу університету з використанням штучного інтелекту»**, затверджена наказом закладу вищої освіти від **«01» жовтня 2024 р., №520/с**

Вихідні дані роботи (зазначається предмет і об’єкт дослідження):

полегшення роботи з системою освітнього процесу використання штучного інтелекту.

Керівники з магістерської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Керівник	Завдання видав	Завдання прийняв
1	Єфремов О.М.	02.10.2024	02.10.2024
2	Єфремов О.М.	02.10.2024	02.10.2024
3	Єфремов О.М.	02.10.2024	02.10.2024

РЕФЕРАТ

Кваліфікаційна робота освітнього ступеня «магістр» присвячена веб-орієнтованому програмного додатку, який буде полегшувати роботу та пошук освітньої інформації. Пояснювальна записка складається з вступу, трьох розділів та списку використаної літератури. Пояснювальна записка до кваліфікаційної магістерської роботи містить 80 сторінок, 67 ілюстрацій, 1 таблицю, 3 додатка та 20 використаних джерел.

Метою роботи є розробка веб-орієнтованого інформаційного порталу, завдання якого є навігація студентів по системі навчання університету.

В роботі було визначено основні завдання на розробку веб-сайту, проаналізовано аналоги, визначено основний функціонал та вимоги до системи. Розглянуто популярні інформаційні системи, та було обрано актуальний дизайн, спроектовано та реалізовано інтерфейс з повним функціоналом. Програмний продукт був протестований.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНИЙ ПОРТАЛ, БАЗИ ДАНИХ, VISUAL STUDIO, ШІ.

ABSTRACT

The qualification work of the master's degree is devoted to a web-based software application that will facilitate the work and search for educational information. The explanatory note consists of an introduction, three chapters and a list of references. The explanatory note to the qualification master's thesis contains 80 pages, 67 illustrations, 1 table, 3 appendices and 20 references.

The purpose of the work is to develop a web-based information portal, the task of which is to navigate students through the university's education system.

The paper identifies the main tasks for the development of the website, analyzes analogues, defines the main functionality and requirements for the system. Popular information systems were considered, and an up-to-date design was chosen, an interface with full functionality was designed and implemented. The software product has been tested.

KEYWORDS: INFORMATION PORTAL, DATABASES, VISUAL STUDIO, AI.

ЗМІСТ

РЕФЕРАТ	3
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	6
ВСТУП	7
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1. Постановка задачі	9
1.2. Аналіз аналогів.....	10
1.3. Вибір архітектури для створення проекту	14
1.4. Використання ІІІ в сучасних веб системах та їх інтеграція.....	17
Висновки до 1 розділу	20
РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-ОРІЄНТОВНОЇ СИСТЕМИ ІНФОРМАЦІЙНОГО ПОРТАЛУ	21
2.1. Визначення варіантів використання та структури системи	21
2.2. Розробка структури бази даних системи.....	24
2.3. Розробка структури файлів системи	31
2.4. Інтеграція ІІІ-асистента в систему.....	42
Висновки до 2 розділу	46
РОЗДІЛ 3. ПОРЯДОК РОБОТИ ТА ТЕСТУВАННЯ СИСТЕМИ	47
3.1. Архітектура додатку та його розгортання	47
3.2. Проектування загального інтерфейсу додатку	48
3.3. Тестування роботи веб-орієнтованого додатку	53
Висновки до 3 розділу	67
ВИСНОВКИ.....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	71
ДОДАТКИ.....	73

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

CSS - Каскадні таблиці стилів (Cascading Style Sheets)

СУБД – Система управління базами даних

ОС – Операційні системи

MySQL - Система управління базами даних на основі SQL (My Structured Query Language)

JS – Мова програмування JavaScript

MAMP - Програмне середовище, яке дозволяє встановити та запустити веб-сервер (Macintosh, Apache, MySQL, PHP)

VS - Текстовий редактор, розроблений компанією Microsoft (Visual Studio)

ІІІ – штучний інтелект

ВСТУП

Актуальність теми. В сучасному часі навчальний процес стає більш доступним та зручним завдяки веб орієнтованим системам. Це дозволяє зібрати та згрупувати велику кількість інформації, організувати діалог між студентами та викладачами, а також отримувати доступ до навчального матеріалу з будь-якого місця і в будь-який момент.

Створення веб орієнтованої системи для університету є важливим кроком у покращенні якості освіти та забезпеченні ефективності навчання. Це дозволить студентам зручно отримати доступ до інформації про навчальні курси, викладачів та академічні досягнення, що допоможе їм краще зорієнтуватись в навчальному процесі та планувати свій час.

Застосування веб-технологій дозволяє зробити навчальний процес більш доступним та оптимальним. Користувачі можуть легко знайти необхідну інформацію про курси, предмети, викладачів та інші аспекти навчання. Вони мають змогу залишати свої відгуки про предмети та викладачів, що дозволяє абітурієнтам отримати більш об'єктивну інформацію та приймати рішення щодо вступу до навчального закладу.

У зв'язку з пандемією COVID-19 та виходом на дистанційне навчання, веб орієнтована система освітнього процесу університету може стати незмінним інструментом для забезпечення ефективного і організованого процесу навчання.

Отож, створення веб орієнтованої системи для університету є важливим завданням, яке забезпечить студентам всю необхідну інформацію в одному місці, зменшить витрати часу та зможе покращити якість навчання. Вона стане важливим інструментом для збільшення ефективності і організації навчального процесу та відповідь на сучасні вимоги технологічного розвитку і дистанційного навчання.

Мета створення. Метою кваліфікаційної роботи є розробка веб орієнтованої системи освітнього процесу університету для покращення якості освіти студентів, вона зможе забезпечити доступ до інформації про курси,

предмети та викладачів. Крім того, студенти зможуть ділитись своїми думками про викладачів, а також переглядати свої досягнення в особистому кабінеті. Форма зворотного зв'язку та база знань допоможуть студентам вирішувати проблеми, що виникають у процесі навчання.

Завдання на роботу. Встановлена мета обумовлює зробити ряд задач, таких як проаналізувати вимоги до системи на основі аналогів, визначення функціональних вимог та потреб для різних ролей користувачів, вибір стеку потрібних технологій, проектування архітектури програмного застосунку, аналіз використання ІІІ у веб системах задля його подальшої інтеграції та тестування системи.

Об'єктом дослідження є веб-орієнтована система яка забезпечує автоматизацію освітнього процесу, зберігання та обробку великих обсягів даних, а також інтеграцію ІІІ-асистента для підвищення функціональності та зручності роботи. Крім того, додаток буде мати дружній та інтуїтивно зрозумілий інтерфейс, що дозволить користувачам з легкістю здійснювати потрібні їм дії.

Предметом дослідження є розробка веб орієнтованої системи освітнього процесу університету. В рамках дослідження будуть розглянуті основні принципи роботи з веб-технологіями, такі як функціонування клієнт-серверної моделі, засоби взаємодії користувача з веб-додатком. Дослідження передбачає пояснення різниці між мовами програмування та їх використанням у веб-розробці, а також професійній сфері діяльності загалом.

Практичне значення роботи. Система може бути використана в університетах для автоматизації освітніх процесів, полегшуючи управління курсами, зворотним зв'язком, обробкою даних та їх доступністю для студентів. Інтеграція ІІІ-асистента надає університетам інструмент для покращення навчального досвіду, забезпечуючи швидкий підхід вирішення простих питань кожного студента. Розробка може бути основою для подальших досліджень і покращень у сфері інтелектуальних освітніх платформ, що сприятиме інноваціям у галузі різного роду навчання.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Постановка задачі

Постановка задачі полягає у створенні веб орієнтованої системи освітнього процесу університету, яка забезпечуватиме студентам доступ до основної кількості потрібної інформації, надання даних до навчальних дисциплін, перегляд інформації про викладачів, а також зручний додаток для адміністрування. Для досягнення цієї мети, система буде висвітлювати інформацію про навчальні курси, предмети та викладачів, а також оцінки студентів. Важливо, щоб система мала користувацький профіль, що дозволить студентам переглядати в ньому основну інформацію про свої біографічні дані та оцінки за всі пройдені предмети. Це допоможе студентам відстежувати свій прогрес та використовувати надану інформацію для підвищення своєї продуктивності.

Крім того, система має містити відгуки про викладачів на основній формі додатку. Це надасть більше контексту до інформації про самі курси, що надаються в системі, і допоможе студентам створити більш обґрунтовану думку. Для забезпечення зворотного зв'язку має бути форма, де користувачі зможуть надсилати свою електронну пошту для зворотного зв'язку. Це дозволить адміністраторам взаємодіяти зі студентами та відправляти запрошення або рекомендації до вступу в університет.

Також, система повинна мати саму базу знань з корисною інформацією про курси, їх предмети та викладачів. Ці дані допоможуть студентам більш ефективно організовувати свій час та знаходити потрібну інформацію швидко та легко.

Основні задачі полягають у створенні функціональності системи, розробці дизайну та інтерфейсу, конфіденційності даних, тестуванні та підтримці на протязі певного часу. Потенційна система сприятиме процесу навчання та

допоможе студентам досягнути більш високих результатів у навчальній діяльності.

Для досягнення мети проекту, необхідно вирішити наступні питання:

- Розробка веб орієнтованої системи, яка буде працювати на основі веб-технологій зі зручним та інтуїтивним інтерфейсом для користувачів.
- Збір, обробка та систематизація інформації про курси, їх навчальні предмети, викладачів.
- Розробка системи користувацького профілю, що дозволить користувачам переглядати свої академічні досягнення та біографічні дані.
- Розробка системи зворотного зв'язку для користувачів, що дозволить отримати відповідь або інформацію від адміністраторів сайту.
- Розробка бази знань, що містить інформацію про курси, викладачів та інші корисні ресурси, що допоможуть користувачам отримати детальнішу інформацію про університет і його систему навчання.

1.2. Аналіз аналогів

Існує велика кількість аналогів до нашого проекту, оскільки більшість університетів та шкіл мають власні веб-орієнтовані системи освітнього процесу. Деякі з них мають схожу функціональну базу, а інші можуть мати свої унікальні особливості. Розглянемо кілька популярних аналогів:

Blackboard. Одна з найпоширеніших веб-орієнтованих систем освітнього процесу. Вона містить багато функцій, включаючи користувацькі профілі, зворотній зв'язок, електронні засоби навчання та багато іншого. Використовується в університетах та школах. Вона надає інструменти для створення курсів, завантаження матеріалів, проведення тестів, оцінювання та комунікації між студентами й викладачами. Її сильними сторонами є багатофункціональність і можливість доступу з будь-якого пристрою, що робить її зручною для дистанційного навчання.

Однак Blackboard має недоліки. Система може здаватися складною у використанні через застарілий інтерфейс, а її впровадження є доволі дорогим, що робить її менш доступною для невеликих навчальних закладів. Попри це, завдяки своїй функціональності та надійності вона залишається популярним вибором для великих освітніх установ.



Рис.1.1. Головна сторінка Blackboard

Moodle. Ця система також широко використовується в школах та університетах. Вона має схожі функції, такі як користувацькі профілі, зворотній зв'язок, електронні засоби навчання та багато іншого. Однак, Moodle має більше можливостей для індивідуалізованого навчання та підтримки студентів. Її головною перевагою є гнучкість та можливість кастомізації, адже вихідний код системи доступний для модифікації. Moodle дозволяє створювати курси, завантажувати матеріали, проводити тести та опитування, а також забезпечує комунікацію між студентами і викладачами.

Серед недоліків Moodle варто відзначити, що для реалізації всіх можливостей часто потрібна команда розробників. Інтерфейс може здатися складним для нових користувачів, а налаштування системи потребує часу та технічних знань. Водночас Moodle є безкоштовною, що робить її доступною навіть для невеликих організацій, і це одна з причин її популярності в освітньому середовищі.

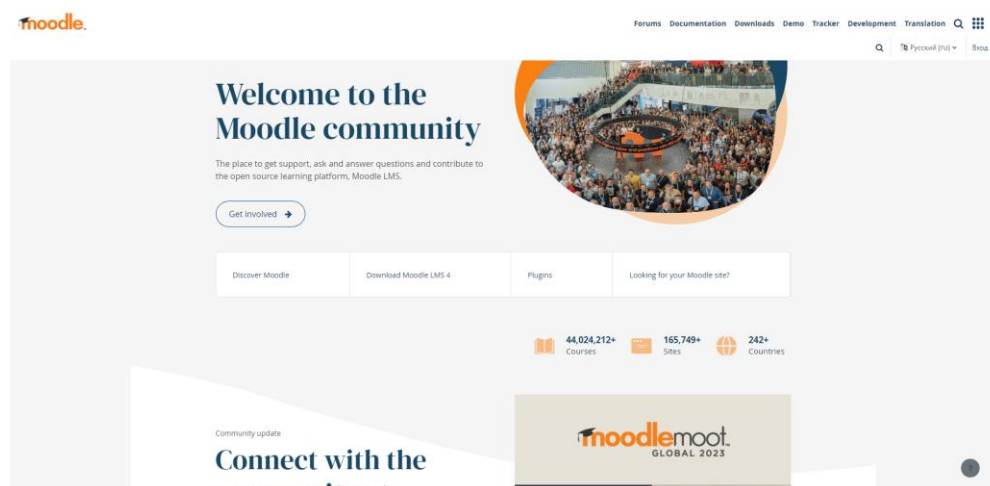


Рис.1.2. Головна сторінка Moodle

Canvas. Веб-орієнтована система освітнього процесу використовується багатьма школами та університетами. Вона має користувацькі профілі, зворотній зв'язок та електронні засоби навчання. Virізняється зручним і сучасним інтерфейсом, що робить її простою у використанні як для студентів, так і для викладачів. Canvas надає інструменти для створення курсів, інтеграції мультимедійних матеріалів, проведення тестів, а також аналітичні модулі для відстеження успішності студентів. Важливою перевагою є мобільна інтеграція, що дозволяє користуватися системою з будь-якого пристрою.

Головним недоліком Canvas є її висока вартість, що може бути недоступною для невеликих навчальних закладів.



Рис.1.3. Головна сторінка Canvas

Google Classroom. Це безкоштовна веб-орієнтована система, яка створена для використання в освітніх закладах. Вона має можливості для створення завдань, зворотнього зв'язку, співпраці та обміну документами. Вона також інтегрується з іншими інструментами Google, такими як Google Drive, Google Docs та Google Meet. Virізняється простим і зрозумілим інтерфейсом, що робить її ідеальною для шкіл та індивідуального навчання. Серед основних функцій – створення та управління завданнями, обмін матеріалами, комунікація між учасниками та інтеграція з іншими інструментами Google, такими як Документи, Таблиці й Meet.

Головним недоліком Google Classroom є обмежений функціонал у порівнянні з більш комплексними платформами, такими як Blackboard чи Canvas. Вона менше підходить для великих університетів, де потрібні складні системи оцінювання чи аналітики. Незважаючи на це, Google Classroom залишається популярним рішенням завдяки своїй доступності, простоті та широкому поширенню серед освітніх установ.

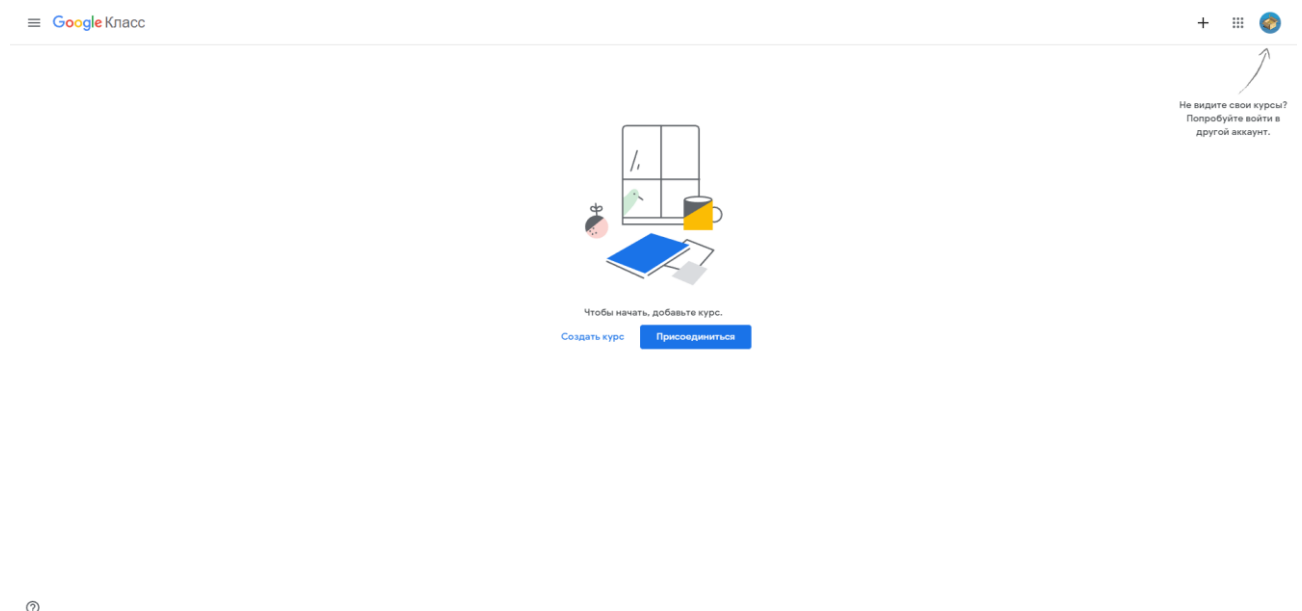


Рис.1.4. Головна сторінка Google Classroom

1.3. Вибір архітектури для створення проекту

Для створення веб-орієнтованої системи освітнього процесу університету, потрібно детальніше розглянути такий перелік технологій:

Frontend: HTML, CSS, і JavaScript для створення користувацького інтерфейсу та взаємодії з користувачами [2,3,4]. Для покращення користувацького досвіду ми можемо додати певну кількість анімацій, які зроблять сайт більш естетичним. Крім того, ми можемо додати більше функцій, які зроблять інтерфейс зручним для користувачів, наприклад, можливість використовувати вкладені сторінки.

Backend: PHP для обробки запитів користувачів та взаємодії з базою даних [6,7]. Однак, можна додати більше функцій для забезпечення високої продуктивності, наприклад, кешування запитів, що дозволить ефективно відповідати на запити користувачів.

Архітектура проекту буде включати наступні компоненти:

Користувацький інтерфейс: його можна створити за допомогою розмітки HTML, CSS (використовуючи оптимальні підходи до стилізації та розробки лейауту) та мови програмування JS [5]. Користувачі зможуть зареєструватись, увійти в систему, переглянути свій профіль, користуватись формою зворотного зв'язку та базою даних. Ми можемо додати більше функцій, які зроблять інтерфейс більш зручним для користувачів, наприклад, можливість отримувати розсилку від адміністраторів в особистому кабінеті.

База даних: її можна створити за допомогою MySQL. База даних буде містити таблиці для переліку користувачів, їх рейтингу в списках, навчальних предметів, розсилки повідомлень для відвідувачів. Ми можемо додати більше таблиць, щоб зберігати додаткову інформацію. Також можна додати механізми резервного копіювання і відновлення бази даних, щоб забезпечити захист від втрати даних.

Порівняння потенційної архітектури проекту

Потенційні мови програмування	C#	Php	JavaScript
Використання	Популярна мова для розробки веб-орієнтованих додатків під платформу .NET	Широко використовується для розробки веб-додатків та машинного навчання	Використовується для розробки веб-додатків та веб-інтерфейсів
Легкість вивчення	Середній	Середній	Легкий
Швидкодія	Швидка	Швидка	Швидка
Екосистема	Потужне інтегроване середовище розробки Visual Studio та велика спільнота розробників	Широка спільнота розробників	Багата екосистема та спільнота розробників, проте менш потужна за C#
Надійність	Дуже надійна	Надійна	Середня
Структурованість	Дуже структурована	Менш структурована, але здатна до функціонального програмування	Менш структурована, проте здатна до функціонального програмування
Сумісність	Windows-орієнтована мова, проте здатна до розробки під інші платформи	Мова, що працює на будь-якій платформі	Мова, що працює на будь-якій платформі

Для реалізації всіх необхідних функцій, потрібно створити наступні модулі:

Модуль реєстрації та авторизації користувачів: потрібно реалізувати зручні та безпечні форми реєстрації і авторизації користувачів. Можливість швидко створити обліковий запис, вказавши свої особисті дані такі як: електронна пошта, пароль, номер залікової книжки, дата народження, фото, ім'я та прізвище. Також треба створити можливість авторизації за указаною поштою та паролем.

Модуль перегляду профілю користувача та перехід до потрібних курсів: ми можемо додати можливість перегляду особистого кабінету користувача, де будуть відображатись всі особисті дані, успіхи в навчанні, повідомлення від адміністраторів та функція переходу до потрібного курсу.

Модуль присвоєння та редагування рейтингу по предметам для певного користувача: потрібно створити окрему форму для адміністратора, де він зможе виставляти оцінки за окремі предмети для потрібних користувачів. Також треба реалізувати можливість редагування оцінки в будь-який момент.

Модуль додавання, редагування та видалення навчальних предметів: треба додати можливість створювати предмети по певному шаблону, реалізувати функції редагування уже створених предметів та їх видалення.

Модуль керування поточними користувачами: можна створити форму для адміністратора, де він буде мати змогу переглядати список всіх зареєстрованих студентів, та при потребі, наприклад коли студент буде відчислений, видалити його аккаунт з бази даних сайту.

Модуль форми зворотного зв'язку, який дозволить користувачам залишати свої контакти: можемо додати форму, яка буде надавати можливість відвідувачам та користувачам залишати свою електронну пошту для очікування на зворотну відповідь.

Наш проект буде реалізовуватись в середовищі розробки Visual Studio Code. Середовище розробки VS є безкоштовним та відкритим редактором коду, розробленим компанією Microsoft. VS надає широкий набір функцій, що

дозволяє зручно та швидко писати код, включаючи автодоповнення, перевірку і підсвічування синтаксису, підказки та багато іншого.

Зокрема, VS має велику кількість розширень, що дозволяють додавати нові функції та полегшувати роботу з багатьма мовами програмування.

Крім того, VS є кроссплатформовим, що дозволяє розробникам використовувати середовище незалежно від особистої операційної системи та співпрацювати з іншими розробниками, які використовують різні ОС. Наприклад, VS працює на операційних системах Windows, macOS та Linux.

Отже, VS є потужним та зручним середовищем розробки, що дозволяє ефективно та легко розробляти веб-орієнтовані системи і співпрацювати з іншими розробниками. Зокрема, завдяки великій кількості розширень та можливості інтеграції з Git, розробка проекту в VS може бути доволі продуктивною та цікавою.

Підсумовуючи, створення веб-орієнтованої системи освітнього процесу університету потребує використання різних технологій. Архітектура проекту буде включати такі компоненти як користувацький інтерфейс, база даних та бекенд, а також різні модулі, що забезпечать весь потрібний функціонал.

1.4. Використання ШІ в сучасних веб системах та їх інтеграція

Штучний інтелект стає все більш важливою складовою сучасних веб-орієнтованих систем [14]. Його впровадження дозволяє суттєво підвищити функціональність, забезпечуючи зручність і швидкість доступу до інформації [15].

За течією сучасних напрямків технологій ШІ використовується наприклад для автоматизації процесів, тобто чат-боти здатні надавати користувачам відповіді в режимі реального часу, що значно зменшує навантаження на адміністративний персонал. Персоналізація є ваговою складовою, алгоритми ШІ аналізують поведінку користувачів, щоб рекомендувати найбільш релевантний контент чи курси. Напрямок освітніх процесів відіграє важливу роль, адже

інтегровані ШІ-рішення допомагають студентам знаходити потрібні навчальні матеріали, відповідають на запитання щодо предметів та викладачів, а також можуть адаптувати рекомендації до рівня знань користувача.

У цьому проєкті ШІ виконує роль асистента, що відповідає на запити користувачів, надає інформацію про курси, викладачів та інші освітні ресурси. На сьогодні існує кілька перевірених та стабільних моделей ШІ, які використовуються у веб-середовищі [10]. Короткий огляд найбільш популярних з них:

OpenAI ChatGPT - генеративна модель мови, створена OpenAI, яка базується на архітектурі GPT (Generative Pre-trained Transformer) [11]. Вона призначена для розуміння та створення тексту на основі контексту запиту. Як перевагу визначимо підтримку генерації тексту високої якості, легко інтегрується через API, має широкий спектр застосувань. Однак наразі безкоштовні моделі мають обмеження на довжину контексту, обов'язкова наявність інтернет-з'єднання для отримання відповідей.

Google Gemini - мультимодальна модель штучного інтелекту, яка поєднує текстові та візуальні можливості [12]. Вона розроблена для забезпечення інтеграції тексту, зображень та інших форматів даних для вирішення складних задач. На стороні даної моделі стоїть інтеграція з іншими сервісами Google, підтримка мультимодальних запитів (текст + зображення), високий рівень продуктивності. Оскільки модель ще дороблюється, як і всі інші звісно, але в порівнянні у Google Gemini більш обмежена доступність API порівняно з OpenAI.

В ході дослідження ШІ-рішень в сучасних веб системах згадаємо про Claude - генеративна модель мови, створена компанією Anthropic, яка акцентує увагу на етичному використанні штучного інтелекту [13]. Вона оптимізована для безпечної та прозорої взаємодії з користувачами. З перевагах можемо розцінити орієнтовану на етичність та прозорість та гнучкість у налаштуваннях. Проте

менша популярність серед розробників, обмежений набір функцій у безкоштовних версіях.

Використання ІІІ як асистента у веб-системах є стратегічним та практичним напрямом, що обґрунтовується сучасними дослідженнями в галузі штучного інтелекту та його інтеграції у цифрові середовища. Наш проект передбачає розробку асистента, який зможе реагувати на запити користувачів.

Дослідження демонструють, що впровадження ІІІ у веб-системи значно підвищує їхню ефективність і доступність. Наприклад, алгоритми машинного навчання дозволяють аналізувати великий обсяг даних у реальному часі, визначаючи закономірності та адаптуючи відповіді до контексту запиту. Завдяки цьому, асистент може пропонувати користувачам саме ту інформацію чи інструменти, які є найбільш релевантними в конкретний момент.

У нашому проекті ІІІ буде інтегрований для спрощення простих процесів, для прикладу таких як пошук інформації, навігація системою та вирішення базових питань. Це дозволить значно зменшити когнітивне навантаження на користувачів, забезпечуючи швидкий доступ до необхідних ресурсів.

На основі сучасних досліджень також відомо, що інтерактивні ІІІ-рішення підвищують рівень залученості та задоволеності користувачів. У нашому випадку, це може бути реалізовано через асистента, який не лише відповідає на запити. Такий підхід відповідає найкращим практикам у веб-розробці та сприяє підвищенню конкурентоспроможності системи та дає підстави для подальшого розвитку.

Інтеграція ІІІ-асистента також дозволить в майбутньому впроваджувати інноваційні підходи до взаємодії з користувачами, такі як голосові команди, розпізнавання тексту чи адаптація вмісту до контексту. У перспективі це може стати фундаментом для створення багатофункціональних платформ, які будуть не просто інструментами, а й стати невід'ємною частиною у вирішенні завдань та цілей користувачів.

Висновки до 1 розділу

В першому розділі звіту ми розглянули важливі етапи для розробки проекту. Спочатку в ньому була поставлена задача, яка визначила напрямок дослідження та розробки.

Далі було проведено аналіз аналогів, що дозволило ознайомитись зі схожими проектами, визначити їх переваги, а також виявити недоліки, які необхідно уникнути в розробці власного проекту.

Останнім етапом був вибір та аналіз архітектури для створення проекту. Визначили оптимальне середовище розробки.

Також були проаналізовані сучасні вимоги до систем із використанням штучного інтелекту. Зроблено огляд популярних моделей ШІ, зокрема OpenAI ChatGPT, Google Gemini та Claude. Вибір моделі Google Gemini для реалізації асистента зумовлений її мультимодальними можливостями та інтеграцією з іншими сервісами Google, а також безкоштовну можливість використання API.

Отже, перший розділ дозволив визначити напрямок роботи, ознайомитися з аналогами, а також вибрати зручну та функціональну архітектуру для розробки проекту.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБ-ОРІЄНТОВНОЇ СИСТЕМИ ІНФОРМАЦІЙНОГО ПОРТАЛУ

2.1. Визначення варіантів використання та структури системи

Система веб-орієнтованого інформаційного порталу університету створюється для забезпечення зручного доступу до інформації для відвідувачів та студентів.

Користувачі порталу мають можливість створювати облікові записи, заповнюючи необхідні дані у формі реєстрації. Після успішної реєстрації вони можуть авторизуватись на сайті, використовуючи свої облікові дані.

Авторизовані користувачі отримують доступ до особистого кабінету та інших функцій порталу.

На порталі також доступна загальна інформація про проєкт. На головній сторінці користувачі можуть ознайомитися з метою, завданнями та іншими деталями.

Для зручності студентів на порталі є функція перегляду навчальних курсів. Відвідувачі можуть ознайомитися зі списком доступних курсів, обрати конкретний курс і переглянути перелік навчальних предметів, які до нього входять.

Інформація про навчальні предмети також доступна для перегляду. Користувачі можуть ознайомитися з описом предмета, інформацією про викладача та відгуками студентів.

Особистий кабінет надає доступ до персональних даних, таких як ім'я, прізвище, електронна пошта, а також до рейтингу з навчальних предметів, який виставляє адміністратор.

Крім того, користувачі можуть переглядати розсилки та отримувати повідомлення від адміністратора. Особистий кабінет дозволяє легко переходити до обраних курсів безпосередньо з інтерфейсу.



Рис.2.1. Варіанти використання порталу користувачем

Система інформаційного порталу також повинна забезпечувати адміністратора широким спектром можливостей для управління сайтом. Це дозволяє ефективно стежити за користувачами та оперативно додавати або оновлювати інформацію на сторінках порталу.

Адміністратор має можливість управляти обліковими записами користувачів, включаючи видалення непотрібних записів. Також система дозволяє додавати нові навчальні предмети до бази даних за допомогою спеціальної форми, редагувати вже створені шаблони та видаляти непотрібні предмети.

Однією з важливих функцій є управління оцінками студентів. Адміністратор може виставляти рейтинги студентам з навчальних предметів, які відображаються в їхніх особистих кабінетах, а також має можливість видаляти ці дані за потреби.

Окрім того, система підтримує функцію розсилки повідомлень. Адміністратор може надсилати індивідуальні або масові повідомлення, які автоматично з'являються в особистих кабінетах користувачів.

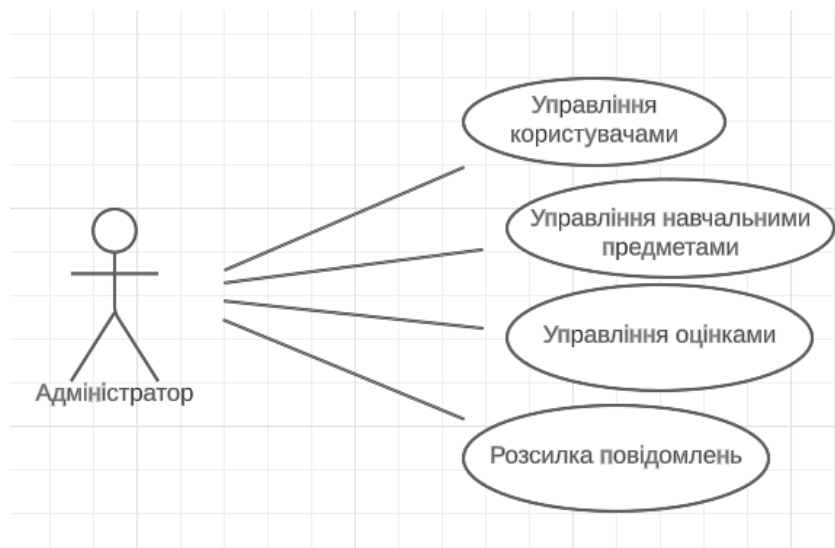


Рис.2.2. Варіанти використання порталу адміністратором

Функціональні вимоги до проєкту включають забезпечення авторизації та реєстрації користувачів. Система дозволяє створювати облікові записи, авторизуватись на порталі та підтримує різні рівні доступу, зокрема для звичайних користувачів і адміністратора.

На головній сторінці порталу відображається інформація про проєкт, його мету, а також форма зворотного зв'язку для користувачів.

Система підтримує функціонал для роботи з навчальними курсами та предметами. Користувачі можуть переглядати список доступних курсів, ознайомлюватися з їхнім описом, а також отримувати детальну інформацію про кожен навчальний предмет. Передбачено можливість переходу на сторінку конкретного предмета, де відображається інформація про викладача та відгуки студентів.

Особистий кабінет користувача містить дані, зазначені під час реєстрації. У ньому можна переглянути оцінки з навчальних предметів, отримувати повідомлення та розсилки від адміністратора, а також здійснювати перехід до обраних курсів.

Для адміністратора передбачено функції управління навчальними предметами. Це включає можливість додавання, редагування та видалення предметів, а також встановлення оцінок для користувачів за певні предмети.

Нефункціональні вимоги охоплюють кілька ключових аспектів. Час ознайомлення з порталом для звичайних користувачів не перевищує 10 хвилин, а для досвідчених користувачів — 3 хвилини. Час відповіді системи на запити становить не більше 0.015 секунди для звичайних запитів і 0.03 секунди для складніших.

Інтерфейс portalу розроблено таким чином, щоб він був інтуїтивно зрозумілим і простим у використанні навіть для новачків. Забезпечується конфіденційність особистої інформації користувачів та захист системи від несанкціонованого доступу. Система сумісна з популярними веб-браузерами, такими як Chrome, Firefox, Safari тощо.

Розробка веб-орієнтованої системи освітнього процесу університету базується на аналізі передових аналогів, таких як Canvas та Blackboard. Це дозволяє впроваджувати сучасний функціонал, спрямований на покращення навчального досвіду студентів.

Ці вимоги визначають основну функціональність і характеристики проєкту, забезпечуючи відповідність потребам користувачів і поставленій меті.

2.2. Розробка структури бази даних системи

В проєкті веб-орієнтованого інформаційного portalу університету нам потрібно організувати зручну систему зберігання та читання великого обсягу даних. Для цього ми розглянемо MySQL як одну з найпопулярніших систем управління базами даних та порівняємо її з конкурентами [1].

MySQL є відкритою реляційною системою управління базами даних, яка надає широкий функціонал для зберігання та обробки даних.

1	•	USE productsdb;
2		
3	•	INSERT Products(ProductName, Manufacturer, ProductCount, Price)
4		VALUES ('iPhone X', 'Apple', 5, 76000);

#	Time	Action	Message	Duration / Fetch
✓ 1	20:15:54	USE productsdb	0 row(s) affected	0.000 sec
✓ 2	20:15:54	INSERT Products(ProductName, Manufactu...	<u>1 row(s) affected</u>	0.219 sec

Рис.2.3. Приклад бази даних MySQL

MySQL є однією з найпопулярніших систем управління базами даних завдяки її численним перевагам. Однією з головних переваг є простота використання. Завдяки зрозумілій і простій структурі, MySQL дозволяє легко створювати таблиці, виконувати запити та керувати базою даних [16].

Ще однією перевагою є висока швидкодія. MySQL добре працює з великими обсягами даних і забезпечує ефективну обробку запитів, що робить його ідеальним вибором для багатьох проєктів.

Система також відзначається масштабованістю, підтримуючи як горизонтальне, так і вертикальне масштабування. Це дозволяє розширювати систему для роботи з більшими обсягами даних або забезпечувати більшу кількість одночасних з'єднань.

Надійність і стабільність є ще однією сильною стороною MySQL. Система добре справляється з великим навантаженням і має механізми відновлення після збоїв, що забезпечує її стабільну роботу навіть у складних умовах [20].

Попри численні переваги, MySQL має і деякі недоліки. Наприклад, у функціональному плані система поступається деяким конкурентам, особливо у підтримці складних операцій або типів даних.

Також, хоча MySQL загалом є нескладною у використанні, налаштування та розгортання бази даних можуть викликати труднощі у початківців. Крім того, залежно від налаштувань і обсягів даних, система може споживати значні ресурси, такі як пам'ять і процесорний час.

На ринку систем управління базами даних MySQL має декількох сильних конкурентів. Наприклад, PostgreSQL є потужною відкритою реляційною СУБД, яка підтримує широкий набір функцій і розширень [9]. У порівнянні з MySQL, PostgreSQL пропонує більш розширені можливості, проте є складнішою у використанні.



Рис.2.4. Система управління базами даних PostgreSQL

Oracle є комерційною СУБД, яка підтримує широкий спектр функцій та масштабованість. Oracle володіє високою продуктивністю та надійністю, але вимагає значних витрат на ліцензії.



Рис.2.5. Система управління базами даних Oracle

SQL Server є комерційною СУБД, розробленою компанією Microsoft. Він має високий рівень інтеграції з іншими продуктами Microsoft та підтримує широкий спектр функцій.



Рис.2.6. Система управління базами даних Oracle

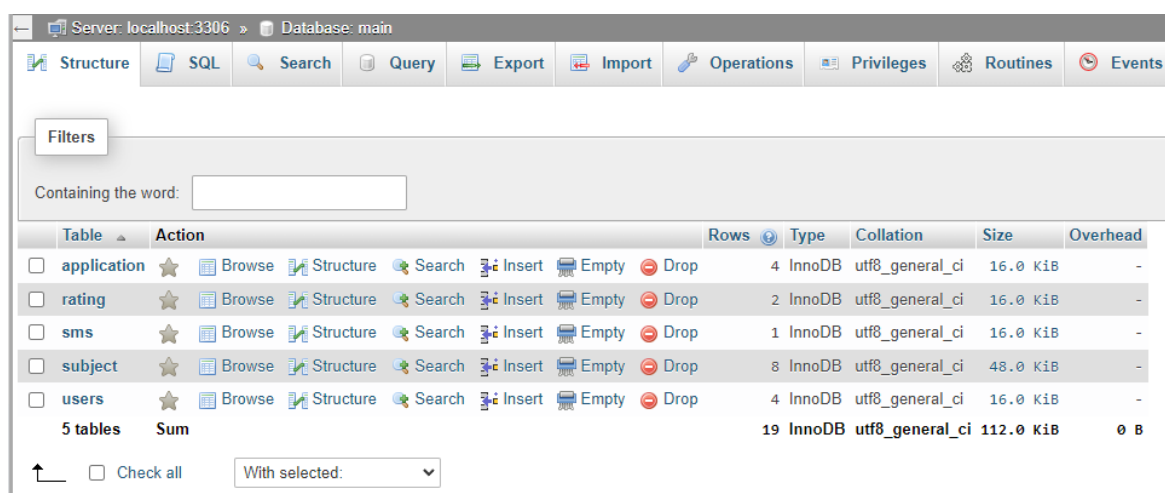
МAMP є зручним середовищем для розробки веб-додатків, яке поєднує сервер Apache, базу даних MySQL та мову програмування PHP [8]. У нашому проєкті це середовище обране для забезпечення ефективної розробки та тестування.

Однією з головних особливостей MAMP є простота встановлення, яка дозволяє швидко налаштувати серверне середовище на комп'ютері. Це особливо важливо для забезпечення швидкого старту роботи над проєктом.

Середовище забезпечує можливість локальної розробки, що дозволяє тестувати веб-додатки без необхідності розгортання їх на живому сервері. Це значно полегшує процес налагодження й тестування.

Інтегровані компоненти, такі як сервер Apache, база даних MySQL та мова програмування PHP, створюють єдину платформу для розробки веб-додатків. Така інтеграція спрощує процес налаштування та дозволяє використовувати популярні технології для створення веб-додатків.

З урахуванням усіх переваг MAMP, це середовище розробки було вибрано як оптимальне для зберігання та реалізації складових нашого проєкту.



The screenshot shows the MySQL database structure view for a database named 'main'. It lists five tables: 'application', 'rating', 'sms', 'subject', and 'users'. Each table has a 'Rows' column indicating the number of records, a 'Type' column (all are InnoDB), a 'Collation' column (all are utf8_general_ci), a 'Size' column, and an 'Overhead' column. The 'Sum' row at the bottom shows a total of 19 rows and 112.0 KiB size.

Table	Action	Rows	Type	Collation	Size	Overhead
☐ application	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16.0 KiB	-
☐ rating	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	16.0 KiB	-
☐ sms	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	16.0 KiB	-
☐ subject	★ Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci	48.0 KiB	-
☐ users	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16.0 KiB	-
5 tables	Sum	19	InnoDB	utf8_general_ci	112.0 KiB	0 B

Рис.2.7. Таблиці бази даних

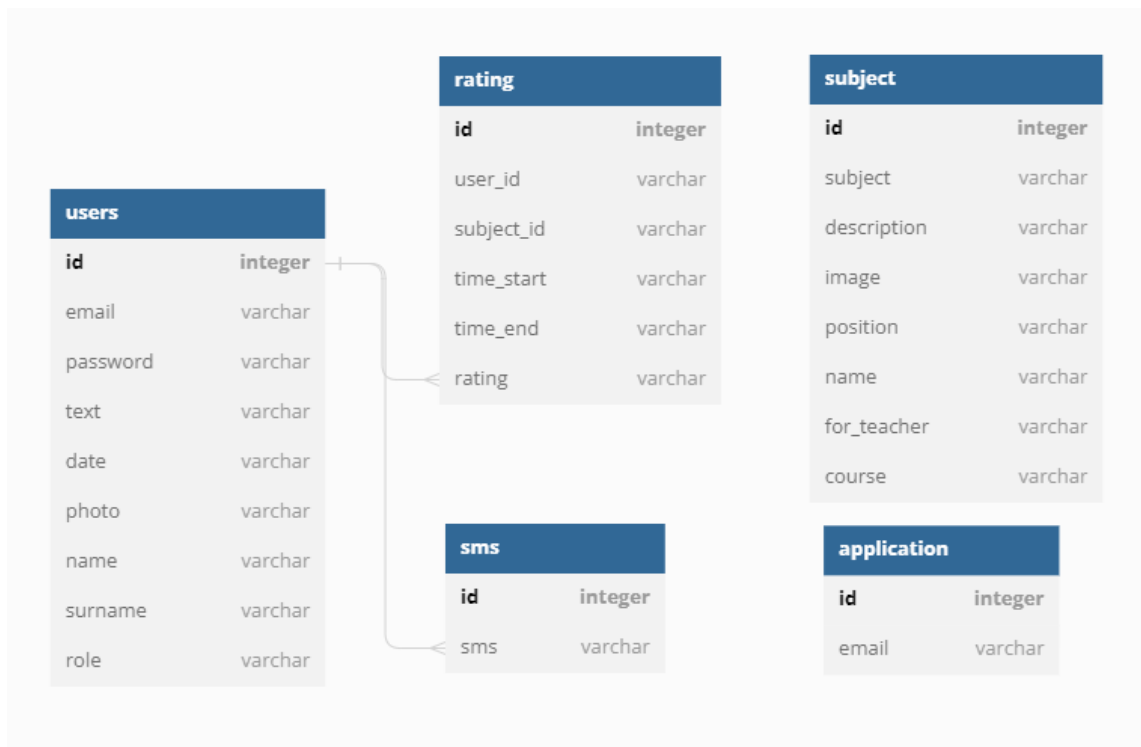


Рис.2.8. Діаграма бази даних

У рамках проекту була створена база даних MySQL з назвою main, яка складається з п'яти таблиць: users, rating, sms, subject та application.

Ця база даних організовує й зберігає інформацію, необхідну для функціонування інформаційного порталу. Кожна таблиця має свою структуру, яка включає відповідні атрибути для зберігання даних, а також встановлені зв'язки між таблицями через зовнішні ключі. Це забезпечує ефективну організацію інформації про користувачів, рейтинги, повідомлення, предмети та заявки.

Таблиця users зберігає інформацію про користувачів порталу. У ній містяться дані, такі як унікальні ідентифікатори, електронні адреси, паролі, текстові поля для збереження додаткової інформації, дати народження, шляхи до фотографій, імена, прізвища та ролі користувачів. Кожен запис у цій таблиці відображає окремого користувача, забезпечуючи можливість управління доступом до системи та зберігання персональної інформації.

Server: localhost:3306 » Database: main » Table: users

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐ 2	email	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 3	password	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 4	text	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 5	date	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 6	photo	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 7	name	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 8	surname	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 9	role	varchar(200)	utf8_general_ci		No	None			Change Drop More

☐ Check all
 With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Рис.2.9. Таблица users

Таблица rating зберігає інформацію про оцінки, які пов'язані з користувачами та предметами. У цій таблиці фіксуються унікальні ідентифікатори рейтингів, ідентифікатори користувачів і предметів, початковий і кінцевий час навчання, а також самі оцінки.

Дані, що зберігаються в таблиці, дозволяють визначати, які оцінки були виставлені конкретним користувачам за певні предмети, а також відстежувати часові рамки навчання. Зв'язки між таблицями забезпечуються за допомогою зовнішніх ключів, що дозволяє інтегрувати інформацію з таблиць users і subject.

Server: localhost:3306 » Database: main » Table: rating

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
☐ 2	user_id	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 3	subject_id	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 4	time_start	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 5	time_end	varchar(255)	utf8_general_ci		No	None			Change Drop More
☐ 6	rating	varchar(255)	utf8_general_ci		No	None			Change Drop More

☐ Check all
 With selected: Browse Change Drop Primary Unique Index Spatial Fulltext

Рис.2.10. Таблица rating

Таблиця sms призначена для зберігання повідомлень, які надсилаються адміністратором користувачам. У ній міститься інформація про унікальні ідентифікатори повідомлень, текст самих повідомлень, а також ідентифікатори користувачів, яким вони адресовані.

Зв'язок із таблицею users здійснюється через відповідні зовнішні ключі, що дозволяє ефективно інтегрувати дані про повідомлення з даними про користувачів.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	sms	varchar(255)	utf8_general_ci		No	None			Change Drop More

Рис.2.11. Таблиця sms

Таблиця subject використовується для зберігання даних про навчальні предмети. Вона включає інформацію про унікальні ідентифікатори кожного предмета, його назву, опис та пов'язані дані про викладачів.

У цій таблиці міститься шлях до фотографії викладача, його посада, повне ім'я, загальна інформація про викладача, а також курс, до якого відноситься конкретний предмет. Ця структура дозволяє забезпечити зручний доступ до всіх необхідних відомостей про навчальні дисципліни та їх викладачів, що сприяє ефективному управлінню освітнім процесом.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	int(11)			No	None		AUTO_INCREMENT	
☐ 2	subject	varchar(255)	utf8_general_ci		No	None			
☐ 3	title	varchar(255)	utf8_general_ci		No	None			
☐ 4	description	varchar(3000)	utf8_general_ci		No	None			
☐ 5	image	varchar(255)	utf8_general_ci		No	None			
☐ 6	position	varchar(255)	utf8_general_ci		No	None			
☐ 7	name	varchar(255)	utf8_general_ci		No	None			
☐ 8	for_teacher	varchar(3000)	utf8_general_ci		No	None			
☐ 9	course	varchar(255)	utf8_general_ci		No	None			

Рис.2.12. Таблица subject

Таблиця application призначена для зберігання заявок, що містять електронні адреси користувачів. Вона включає унікальний ідентифікатор кожної заявки та електронну пошту користувача. Ця таблиця дозволяє зберігати контактні дані користувачів, що подали заявку, для подальшої обробки чи комунікації.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐ 1	id	int(11)			No	None		AUTO_INCREMENT	
☐ 2	email	varchar(255)	utf8_general_ci		No	None			

Рис.2.13. Таблица application

2.3. Розробка структури файлів системи

Тепер ми опишемо організацію файлів у нашому проекті. Наведемо основні файли та їх реалізовані функції, які будуть розглянуті досить детально.

Загальна структура файлів проекту визначає організацію та розміщення файлів у нашій веб-орієнтованій системі. Наведемо опис існуючих папок з файлами, які використовуються в проекті:

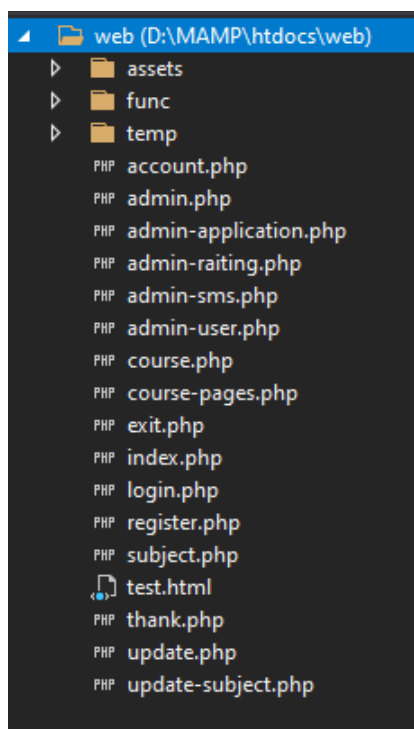


Рис.2.14. Загальна структура файлів проекту

Папка `assets` є однією з основних директорій для зберігання ресурсів, таких як зображення, шрифти, скрипти та стилі, які використовуються для оформлення нашого порталу. Вона дозволяє централізовано організувати всі статичні файли, необхідні для відображення та функціонування веб-додатку.

Структура папки забезпечує організацію ресурсів, дозволяючи розміщувати різні типи файлів у відповідних підкаталогах. Наприклад, зображення можна зберігати в підпапці `img`, а скрипти — в `script`. Це забезпечує зручність у пошуку та управлінні файлами.

Використання окремої директорії для статичних ресурсів також сприяє розширюваності проекту. Це запобігає можливному перекриттю файлів між різними частинами проекту та дозволяє легко додавати нові ресурси.

Централізоване розташування ресурсів дозволяє спростити доступ до них у коді. Використовуючи відносні посилання на файли з директорії `assets`, можна

легко змінювати їх, якщо це буде необхідно, без порушення загальної структури проекту.

Структура папки `assets` з підкаталогами, такими як `fonts`, `img`, `script` та `style`, дозволяє логічно організувати ресурси проекту, що полегшує їх використання та підтримку під час розробки та експлуатації веб-орієнтованого інформаційного порталу.

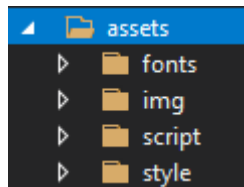


Рис.2.15. Структура папки `assets`

Детальніше розглянемо внутрішню структуру папки `assets`.

Папка `fonts` призначена для зберігання шрифтів. Вона дозволяє додавати файли шрифтів у різних форматах, таких як `.ttf` або `.woff`, що забезпечує сумісність з різними браузерами та пристроями. Це дає можливість використовувати шрифти на порталі з гарною підтримкою на різних платформах.

Папка `img` містить зображення, які використовуються на порталі. Для зручності можна створювати підкаталоги, що відповідають різним категоріям зображень. Наприклад, можна створити підпапки `avatars` для профільних фотографій користувачів або `banners` для рекламних банерів. Це дозволяє легко організувати і знайти необхідні файли.

Папка `script` використовується для зберігання JavaScript-скриптів, які використовуються в проекті. Тут можуть знаходитися як власноруч написані скрипти, так і зовнішні бібліотеки або фреймворки. Це дозволяє централізовано керувати усіма скриптами, що використовуються на порталі.

Папка `style` містить файли CSS, які визначають зовнішній вигляд порталу. Можна створити окремі стилі для різних сторінок або компонентів, а також організувати стилі за допомогою підпапок для полегшення навігації та управління.

Завдяки такій організації папок ми зможемо ефективно керувати ресурсами проекту, такими як шрифти, зображення, скрипти та стилі, що підвищує зручність розробки та подальшої підтримки веб-орієнтованого інформаційного порталу.



Рис.2.16. Папка func

Папка func містить набір PHP-файлів, що відповідають за різні функції та операції на інформаційному порталі. Кожен файл у цій папці реалізує певну функціональність у системі. Ось опис деяких з них:

Файл add-subject.php відповідає за додавання нового предмета до бази даних. Він обробляє дані, отримані з форми для додавання предмета, і додає ці дані до таблиці subject у базі даних.

```
<?php  
  
require_once 'db.php';  
  
mysqli_query($connect, "INSERT INTO `subject` (`id`, `subject`, `title`,  
`description`, `image`, `position`, `name`, `for_teacher`, `course`) VALUES (NULL,  
'$subject', '$title', '$description', '$image', '$position', '$name', '$for_teacher',  
'$course');");  
header('Location: ../admin.php') ?>
```

Лістинг 1.1. Додавання нового предмету

application.php. Продемонстрований файл відповідає за обробку заявок користувачів на потенційну відповідь від адміністратора на залишену електронну пошту. Він містить код, який обробляє контакти від користувачів у відповідній таблиці бази даних, а саме "application".

```
<?php  
require_once 'db.php';  
  
$email = $_POST['email'];  
mysqli_query($connect, "INSERT INTO `application` (`id`, `email`) VALUES (NULL,  
'$email');");  
header('Location: ../thank.php') ?>
```

Лістинг 1.2. Обробка заявок

db.php. Це модуль, який встановлює з'єднання з базою даних MySQL. Він містить функцію, яка дозволяє підключитися до бази даних, після чого ми можемо виконувати запити та інші операції, пов'язані з базою даних.

```
<?php
$connect = mysqli_connect('localhost', 'root', 'root', 'main');

if(!$connect){
    echo 'Errors!';
} ?>
```

Лістинг 1.3. Підключення до БД

delete-rating.php. Цей файл відповідає за видалення оцінок з бази даних. Він містить функції, які виконують запити до бази даних для видалення конкретної оцінки з таблиці "rating".

```
<?php
require_once 'db.php';

$id = $_GET['id'];
mysqli_query($connect, "DELETE FROM `rating` WHERE `rating`.`id` = '$id' ");
header('Location: ../admin-rating.php'); ?>
```

Лістинг 1.4. Видалення оцінок

delete-sms.php. Модуль відповідає за видалення повідомлень (sms) з бази даних. Він містить код, який виконує запити до бази даних для видалення конкретного повідомлення з таблиці "sms".

```
<?php require_once 'db.php';

$id = $_GET['id'];
mysqli_query($connect, "DELETE FROM `sms` WHERE `sms`.`id` = '$id' ");

header('Location: ../admin-sms.php'); ?>
```

Лістинг 1.5. Видалення повідомлень

delete-subject.php. Цей модуль відповідає за видалення предметів з бази даних. Він містить функції, які виконують запити до бази даних для видалення конкретного предмета з таблиці "subject".

```
<?php require_once 'db.php';
```

```
$id = $_GET['id'];
mysqli_query($connect, "DELETE FROM `subject` WHERE `subject`.`id` = '$id' ");
header('Location: ../admin.php'); ?>
```

Лістинг 1.6. Видалення предметів

delete-user.php. Файл, аналогічно попереднім, відповідає за видалення користувачів з бази даних. Він містить код, який виконує запити до бази даних для видалення конкретного користувача з таблиці "users".

```
<?php require_once 'db.php';

$id = $_GET['id'];
mysqli_query($connect, "DELETE FROM `users` WHERE `users`.`id` = '$id' ");
header('Location: ../admin-user.php'); ?>
```

Лістинг 1.7. Видалення користувача

login.php. Цей файл відповідає за обробку процесу авторизації користувача. Він містить функції, які перевіряють введені дані користувача, їх збіг з даними в базі даних і забезпечують доступ користувача до системи.

```
<?php session_start();
require_once 'db.php';
$login = $_POST['email'];
$password = $_POST['password'];

$res = mysqli_query($connect, "SELECT * FROM users WHERE email = '$login' and
password = '$password' ");
$result = mysqli_fetch_assoc($res); ?>
```

Лістинг 1.8. Авторизація

rating.php. Файл відповідає за отримання та відображення оцінок користувача. Він містить функції, які виконують запити до бази даних для отримання оцінок користувача з таблиці "rating" і їх подальший вивід на веб-сторінку.

```
<?php require_once 'db.php';

mysqli_query($connect, "INSERT INTO `rating` (`id`, `user_id`, `subject_id`,
`time_start`, `time_end`, `rating`) VALUES (NULL, '$user_id', '$subject_id',
'$time_start', '$time_end', '$rating');");
header('Location: ../admin-raiting.php') ?>
```

Лістинг 1.9. Відображення оцінок

register.php. Цей файл відповідає за обробку процесу реєстрації нового користувача. Він містить функції, які перевіряють введені дані користувача на унікальність і створюють новий запис користувача в таблиці "users", що забезпечує його реєстрацію в системі.

```
<?php require_once 'db.php';

mysqli_query($connect, "INSERT INTO `users` (`id`, `email`, `password`, `text`,
`date`, `photo`, `name`, `surname`, `role`) VALUES (NULL, '$email', '$password',
'$number_book', '$date', '$photo', '$name', '$surname', '1');");
header('Location: ../login.php') ?>
```

Лістинг 1.10. Логіка реєстрації

sms.php. Модуль відповідає за обробку повідомлень, що надсилаються користувачам. Він містить код, який додає повідомлення до бази даних, генерує повідомлення для відправки та забезпечує доставку повідомлення користувачам в особистий кабінет.

```
<?php require_once 'db.php';

$sms = $_POST['sms'];
mysqli_query($connect, "INSERT INTO `sms` (`id`, `sms`) VALUES (NULL, '$sms ');");
header('Location: ../admin-sms.php') ?>
```

Лістинг 1.11. Обробка повідомлень

update-rating.php. Цей модуль відповідає за оновлення оцінок у базі даних. Він містить функції, які виконують запити до бази даних для оновлення оцінок користувача в таблиці "rating".

```
<?php require_once 'db.php';

$id = $_POST['id'];
$rating = $_POST['rating'];
mysqli_query($connect, "UPDATE `rating` SET `rating` = '$rating' WHERE
`rating`.`id` = '$id';");
header('Location: ../admin-raiting.php'); ?>
```

Лістинг 1.12. Оновлення оцінок

update-sub.php. Цей файл відповідає за оновлення інформації про предмет у базі даних. Він містить код, який виконує запити до бази даних для оновлення даних про конкретний предмет в таблиці "subject".

```
<?php require_once 'db.php';
```

```
$id = $_POST['id'];  
$subject = addslashes($_POST['subject']);  
$title = addslashes($_POST['title']);  
$description = addslashes($_POST['description']);  
$position = addslashes($_POST['position']);  
$name = addslashes($_POST['name']);  
$for_teacher = addslashes($_POST['for_teacher']);  
  
mysqli_query($connect, "UPDATE `subject` SET `subject` = '$subject', `title` =  
`$title`, `description` = '$description', `position` = '$position', `name` =  
`$name`, `for_teacher` = '$for_teacher' WHERE `subject`.`id` = '$id';");  
header('Location: ../admin.php'); ?>
```

Лістинг 1.13. Оновлення інформації про предмет

Кожен з цих файлів відповідає за певну функціональність проекту, а їх узгоджена робота дозволяє забезпечити відповідну функціональність та взаємодію всіх компонентів системи.

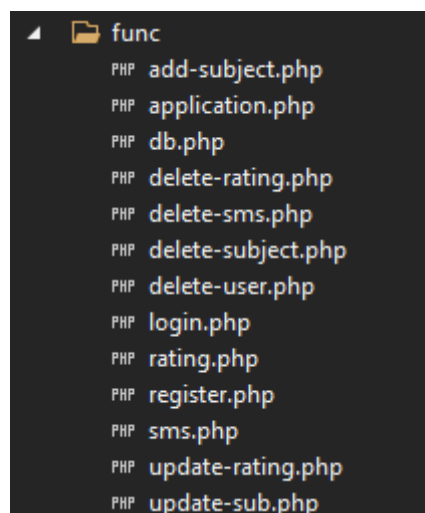


Рис.2.17. Структура папки func

Далі у нас йде папка "temp", яка містить файли "header.php" і "admin-header.php", відповідальні за вигляд та структуру заголовку веб-сторінок в проєкті.

header.php. Файл з функціями, який відображає заголовок сторінки для звичайних користувачів. Він включає логотип, навігаційне меню, форму довідки та контактів, які відображаються на кожній сторінці проекту. Файл "header.php" включений в інші файли для створення єдиної структури заголовка на всіх сторінках.

```
<?php session_start();
require_once './func/db.php';
$user_id = $_SESSION['id'];
$new = mysqli_query($connect, "SELECT * FROM `users` WHERE `id` = '$user_id'");
$new = mysqli_fetch_assoc($new);
?>

    <?php
    if ($new['role'] == 2) {
        echo ' <span><a href="./admin.php">Адмінка</a></span>';
    } else {
        echo '<span><a href="./account.php"></a></span>';
    }
    if (isset($_SESSION['id'])) {
        echo ' <span><a href="./exit.php">Вийти</a></span>';
    }
    ?>
</div>
</div>
```

Лістинг 1.14. Навігація, частина визначення ролі користувача

admin-header.php. Цей файл містить код, який відображає заголовок сторінки для адміністратора. Він має подібний вигляд до "header.php", але додатково містить елементи адміністративного доступу, такі як посилання на панель адміністрування, додаткові функції та інформацію, специфічну для адміністратора. Файл "admin-header.php" також включений в інші файли, щоб забезпечити єдину структуру заголовка для сторінок, доступних тільки адміністратору.

Ці файли дозволяють забезпечити однорідний та консистентний вигляд заголовка на всіх сторінках проекту, спрощують розробку та підтримку коду, а також покращують користувацький досвід, забезпечуючи логічну навігацію та зручність використання.

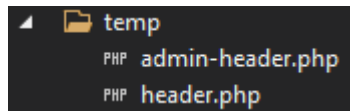


Рис.2.18. Структура папки temp

Також ми маємо загальний функціонал сайту, який знаходиться в основній папці "web". Розглянемо головні реалізовані можливості.

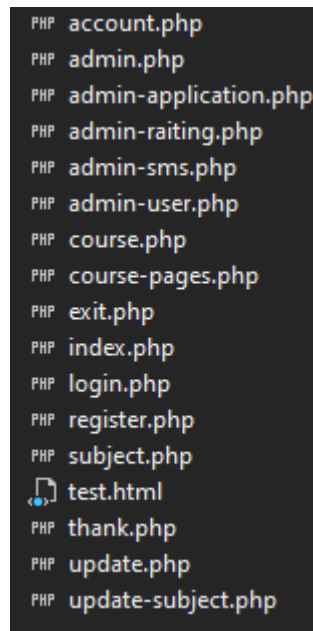


Рис.2.19. Структура папки web

account.php. Цей файл відповідає за відображення особистого кабінету користувача. В ньому реалізовані функції для перегляду особистих даних студента, оцінок з предметів, розсилки від адміністратора та навігації до потрібних курсів.

```
<?php
```

```
include './temp/header.php';
require_once './func/db.php';
```

```
$new = mysqli_query($connect, "SELECT * FROM `users` WHERE `id` = '$user_id'");
$new = mysqli_fetch_assoc($new); ?>
```

```
<div class="wrapper-user">
    <div class="user-information">
        
        <p class="name">Студент</p>
```

```

<div class="user-date-wrap">
    <div class="user-date">
        <p>Ім'я та прізвище</p>
        <p><?= $new['name'] ?> <?= $new['surname'] ?></p>

```

Лістинг 1.15. Особистий кабінет, приклад верстки

admin.php. Даний файл відповідає за відображення адміністративної панелі. В ньому реалізовані функції для додавання, редагування та видалення навчальних предметів, виставлення та редагування оцінок для користувачів, надсилання повідомлень до особистих кабінетів студентів. Також реалізована можливість видалення аккаунта певного користувача.

```

<div class="wrapper-admin">
    <div class="admin-nav">
        <?php include './temp/admin-header.php'; ?>
    </div>
</div>
<div class="admin-form">
    <form action="./func/add-subject.php" method="POST"
enctype="multipart/form-data">
        <h3>Додавання предмета</h3>
        <input type="text" placeholder="Предмет" name="subject" required>
        <input type="text" placeholder="Заголовок" name="title" required>
        <textarea name="description" placeholder="Опис" id="" cols="30"
rows="10"></textarea>

```

Лістинг 1.16. Адміністративна панель, частина верстки

course.php. Цей модуль відповідає за відображення списку курсів навчання. Через нього можна перейти до переліку навчальних предметів для перегляду детальної інформації.

course-pages.php. Модуль відповідає за відображення основної сторінки певного предмета курсу. Він містить загальну інформацію про предмет, викладача, відгуки студентів тощо.

```

<?php include './temp/header.php';
require_once './func/db.php';

$news_id = $_GET['id'];
$new = mysqli_query($connect, "SELECT * FROM `subject` WHERE `id` =
'$news_id'");

```

```
$new = mysqli_fetch_assoc($new);  
$id_blog = $new['id']; ?>
```

Лістинг 1.17. Основна сторінка певного курсу, витяг даних

index.php. Це один з важливих файлів проекту, який відображає головну сторінку інформаційного порталу університету. Він містить загальну інформацію про проект, форму зворотнього зв'язку, а також навігаційне меню для переходу до різних розділів та функцій порталу. Код сторінки можна знайти в додатку В.

Наведені файли представляють основні сторінки та функціональність проекту. Вони виконують роль контролерів, які обробляють запити користувача та відповідають за відображення відповідної інформації на веб-сторінках.

2.4. Інтеграція ШІ-асистента в систему

Інтеграція ШІ-асистента у веб-орієнтовану систему здійснюється за допомогою API Google Gemini [17]. Асистент надає користувачам можливість отримувати відповіді на запитання, що стосуються навчальних курсів, предметів та викладачів. Крім того, він може пропонувати рекомендації щодо навігації між різними розділами сайту, а також допомагає швидко знаходити необхідну та релевантну інформацію завдяки інтеграції з базою даних.

Процес інтеграції передбачає використання PHP для обробки запитів до API, створення структурованих запитів, що включають контекст, зокрема початкову інструкцію для асистента, та використання локального сховища для збереження стану взаємодії з користувачем.

Для взаємодії з ШІ використовується REST API, що забезпечує передачу запитів і отримання відповідей у форматі JSON [18]. Запити формуються на серверній частині проекту за допомогою мови програмування PHP.

Перший запит до асистента включає інструкцію та список доступних предметів, що зберігається в базі даних. Це дозволяє забезпечити точність відповідей, орієнтованих на специфіку університету та надає чату напрямок за

яких не варто заходити, генерація відповідей буде стосуватися лише актуальної інформації, таким чином зменшуємо ймовірність надання вигаданої або банально некоректної інформації.

Для підтримки безперервності діалогу між користувачем і асистентом використовується локальне сховище, що забезпечує контекстну зв'язність.

Інтеграція з базою даних - запити користувачів обробляються за допомогою SQL-запитів до бази даних, після чого результати передаються в ШІ для формування відповіді. Наприклад, запит на пошук предмета генерує вибірку з таблиці subject.

Для роботи з API, потрібно скористуватися Google AI Studio [19]. Там ми отримаємо ключ для роботи з API. Також там можна корегувати обрану модель, проаналізувати інші. В кваліфікаційній роботі використовується безкоштовна модель gemini 1.5-flash.

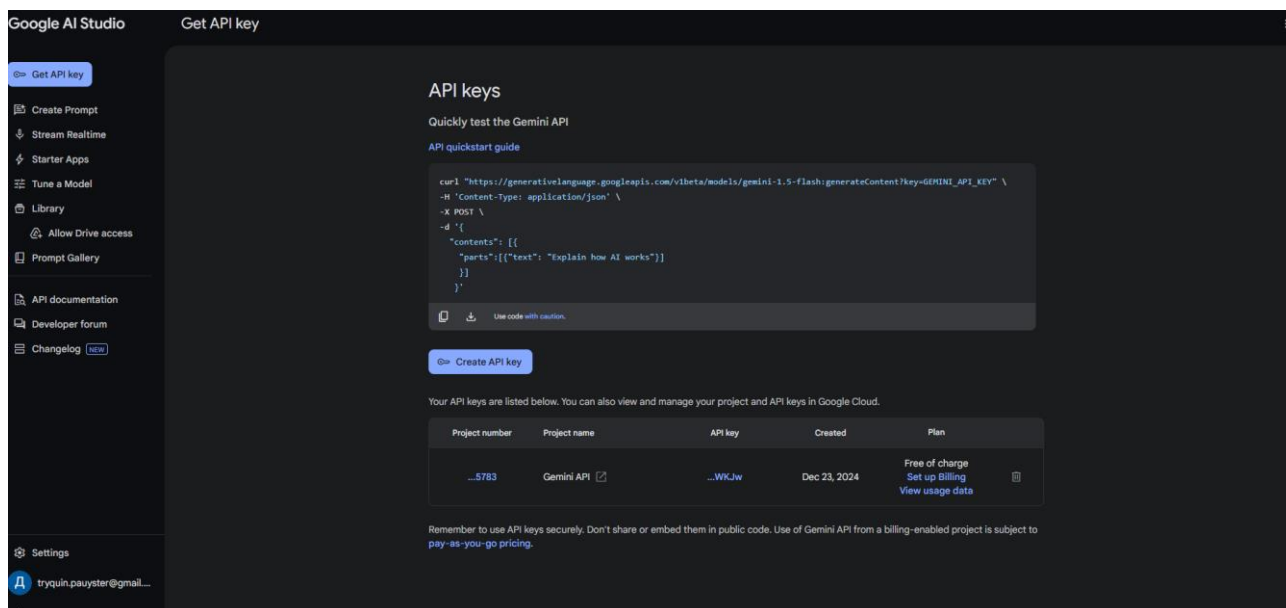


Рис.2.20. Google AI Studio

Також в Google Cloud можна переглянути квоти на використання моделей, лімітів. Досить корисна сторінка, адже дозволяє точно відслідкувати помилку сервера, наприклад, якщо було досягнуто ліміту моделі.

The screenshot shows the Google Cloud Gemini API console. At the top, there's a search bar and a 'Gemini API' tab. Below the header, a message states: 'Secure your account and protect your data by turning on multi-factor authentication. Add an extra layer of security today.' A 'LEARN MORE' button is next to it. The main content area is titled 'API/Service Details' and includes a 'DISABLE API' button. A sidebar on the left lists 'Enabled APIs & services', 'Library', 'Credentials', 'OAuth consent screen', and 'Page usage agreements'. The main area contains a warning: 'Get alerted if a quota is close to reaching its maximum. Click on more actions button ⓘ in a row to get started, or click "Learn more" to view documentation.' Below this, three summary boxes show: 'Current usage > 90%' (0), '7 day peak usage > 90%' (2), and 'All quotas & system limits' (651). A 'MANAGE ALERT POLICIES' link is also present. A table below lists various quotas with columns for Name, Type, Dimensions, Value, Current usage percentage, Current usage, and Adjustable. The table includes rows for request limits per day and per minute for different regions and models.

Name	Type	Dimensions (e.g. location)	Value	Current usage percentage	Current usage	Adjustable
Request limit per model per day for a project in the free tier	Quota	model: gemini-1.5-flash	1,500	1.93%	29	Yes
Batch embed contents request limit per minute for a region	Quota	region: africa-south1	150	0%	0	Yes
Batch embed contents request limit per minute for a region	Quota	region: asia-east1	150	0%	0	Yes
Batch embed contents request limit per minute for a region	Quota	region: asia-east2	150	0%	0	Yes
Batch embed contents request limit per minute for a region	Quota	region: asia-northeast1	150	0%	0	Yes
Batch embed contents request limit per minute for a region	Quota	region: asia-northeast2	150	0%	0	Yes
Batch embed contents request limit per minute for a region	Quota	region: asia-northeast3	150	0%	0	Yes
Batch embed contents request limit per minute for a region	Quota	region: asia-south1	150	0%	0	Yes

Рис.2.21. Google Cloud

Для початку було зроблено окремо сторінку чату з асистентом. Тут у нас звичайна верстка та слухачі подій, реалізовані мовою JavaScript.

```
<?php include './temp/header.php'; ?>
```

```
<script>
function saveHistoryToLocalStorage(history) {
    localStorage.setItem('chatHistory', JSON.stringify(history));
}

function getHistoryFromLocalStorage() {
    const history = localStorage.getItem('chatHistory');
    return history ? JSON.parse(history) : [];
}
```

Лістинг 1.18. Окрема сторінка чату з ШІ-асистентом

Серверна логіка прописана в окремому файлі, там реалізований запит до API, також вказані інструкції для асистента. Весь код логіки роботи з Gemini API можна переглянути в додатку А.

```
$data = array(
    'contents' => [
        ['parts' => [
            ['text' => $initialPrompt],
            ['text' => $contextPrompt],
            ['text' => $userMessage]
        ]]
    ]
);
```

Лістинг 1.19. Приклад роботи з Gemini API

Зручним та хорошим рішенням також було зробити доступ до асистента на всіх сторінках, де буде можливість відкрити чат та за бажанням перейти на окрему сторінку.

Це було реалізовано додаванням коду в header.php. Повний код слухачів подій в додатку Б.

```
<?php
```

```
$currentPage = basename($_SERVER['PHP_SELF']);  
if ($currentPage != 'ai-page.php') {  
    echo '  
        <div class="ai-assistant">  
            <div class="ai-assistant-button" onclick="toggleChat()">  
                <i class="fa-solid fa-question"></i>  
            </div>  
        </div>
```

Лістинг 1.20. Доповнення хедера, міні чат на всіх сторінках

Отже, інтеграція ІІІ-асистента дозволить підвищити зручність використання порталу, зменшити навантаження на адміністративний персонал, забезпечить швидкий доступ до релевантної інформації для студентів і викладачів. Gemini API є зручним рішенням для розробки ІІІ-асистента.

Як один з висновків варто відзначити перспективи розвитку ІІІ-асистента. На фоні розвитку даного роду технологій, інтеграція такого роду функціоналу може стати фундаментальною частиною для будь якого проекту, особливо розрахованого на роботу з користувачами. Як перспективи розвитку, на тлі інформаційного порталу можемо зазначити такі варіанти: масштабування системи, а конкретніше перехід на хмарну основу; персоналізація взаємодії, використовувати алгоритми машинного навчання для адаптації відповідей, мати свою власну базу, налаштування рекомендацій залежно від академічного прогресу студента; розширити функціонал ІІІ-асистента, інтеграція голосових команд, додавання мультимодальних можливостей, наприклад обробки зображень; інтеграція з іншими системами, підключення до платформ Google Classroom, Moodle.

Висновки до 2 розділу

В другому розділі проводилось детальне визначення варіантів використання системи та її структури. Варіанти використання описують типові сценарії взаємодії користувачів з системою, їхні права та обмеження. Структура системи визначає логічну організацію модулів, компонентів та їх зв'язки.

Також велась розробка структури бази даних системи. Відбувся аналіз вимог до зберігання та обробки даних, визначення сутностей, атрибутів та зв'язків між ними. Розроблялись схеми таблиць, визначались типи даних та обмеження. Розглядалась оптимізація запитів та забезпечення цілісності даних.

Останнім етапом була розробка структури файлів системи. Визначались основні файли, їх функціональність та взаємозв'язок. Описувались ролі та обов'язки кожного файлу, їх призначення та відповідальність. Розглядались папки та підпапки, у яких розташовувались файли проекту.

Також було використано API Google Gemini для обробки запитів і формування відповідей. Інтеграція ШІ забезпечила ефективну взаємодію з користувачами.

Таким чином, другий розділ дозволяє детально спроектувати веб-орієнтовану систему інформаційного порталу, враховуючи висновки першого розділу та вимоги проекту.

РОЗДІЛ 3. ПОРЯДОК РОБОТИ ТА ТЕСТУВАННЯ СИСТЕМИ

3.1. Архітектура додатку та його розгортання

Архітектура додатку є фундаментальною складовою його розробки та успішного функціонування. Цей аспект стає ключовим при створенні веб-орієнтованих систем інформаційних порталів, таких як наш проект. Архітектура визначає спосіб організації компонентів системи, їх взаємозв'язки та взаємодію, що забезпечує досягнення поставлених цілей.

У цьому розділі дипломної роботи ми детально розглянемо архітектуру нашого додатку, спроектовану для реалізації функціональності інформаційного порталу. Ми дослідимо різні аспекти архітектури, такі як клієнт-серверна модель, шарування, модульність та взаємодію між компонентами системи.

Процес проектування архітектури включає в себе аналіз вимог до системи, вибір оптимальних технологій та інструментів, а також розподіл функціональності на логічні компоненти. Метою є створення структурної та масштабованої системи, яка забезпечує зручне використання та підтримку:

Клієнтська частина:

- Веб-браузер, який використовуватимуть користувачі для взаємодії з системою.
- HTML, CSS і JS для створення інтерфейсу користувача та взаємодії з сервером.

Серверна частина:

- Web-сервер, такий як Apache, для обробки HTTP-запитів від клієнтів.
- MAMP (Mac, Apache, MySQL, PHP) або аналогічний стек (WAMP або LAMP) для локальної розробки і тестування системи.
- PHP для обробки динамічного контенту, роботи з базою даних і здійснення логіки бізнес-процесів.
- MySQL база даних для зберігання і керування даними системи.

Структура файлів і каталогів:

- У кореневій папці проекту розташовується файл `index.php`, який є точкою входу до серверної частини.
- Каталог `assets` містить підкаталоги `fonts`, `img`, `script`, `style`, де знаходяться файли шрифтів, зображень, скриптів та стилів відповідно.
- Каталог `func` містить файли з функціями для обробки різних операцій системи, такі як додавання, видалення, оновлення даних.
- Каталог `temp` містить файли, які використовуються для створення загальних компонентів інтерфейсу, такі як шапка сайту або панель адміністратора.
- Каталог `web` містить окремі файли та папки для кожної сторінки або функціональної області системи.

Розгортання:

- Після локальної розробки та тестування проекту з використанням MAMP, можна розгорнути його на веб-сервері. Для цього нам знадобиться веб-хостинг, який підтримує PHP і MySQL.
- Завантажимо файли проекту на сервер, налаштуємо з'єднання з базою даних, імпортуємо схему таблиць.

Ця архітектура дозволяє розділити логіку клієнтської та серверної частини, забезпечує зручну організацію файлів та папок проекту і забезпечує ефективну роботу з базою даних. Для розгортання можемо використовувати будь-який веб-хостинг, що підтримує необхідні технології.

3.2. Проектування загального інтерфейсу додатку

Також звернемо увагу на проектування інтерфейсу нашого проекту з використанням потужного інструменту Figma. Розібравшись з аналізом вимог та розумінням потреб користувачів, ми перейдемо до створення прототипів, дизайну та розташування елементів інтерфейсу для забезпечення зручної та ефективної взаємодії користувача з додатком.

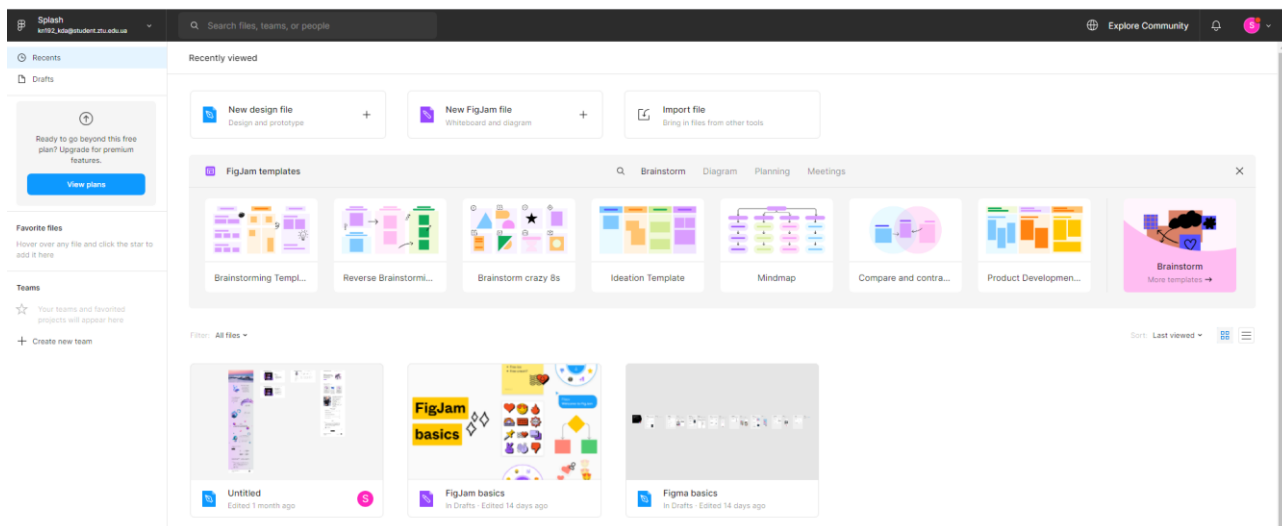


Рис.3.1. Головна сторінка онлайн-сервісу Figma

Ми будемо використовувати різні аспекти проектування інтерфейсу, такі як вибір кольорової палітри, типографіка, розташування елементів, структура навігації та взаємодія з користувачем. Також переглянемо принципи візуального дизайну, які допоможуть створити привабливий та сучасний інтерфейс.

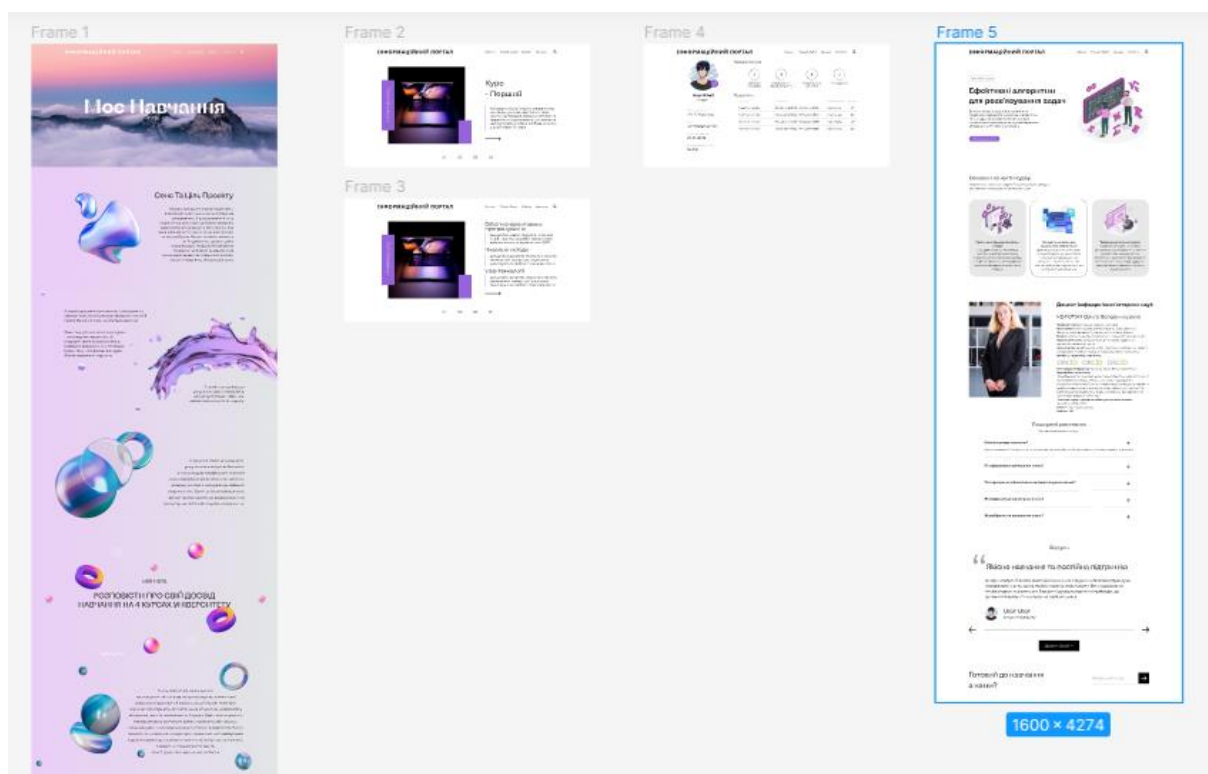


Рис.3.2. Розроблений прототип проекту

У нас є створений інтерфейс нашого порталу, тому детально розглянемо кожен із фреймів, що використовується в проекті.

Фрейми є візуальними контейнерами, які містять елементи інтерфейсу та відображають різні сторінки, екрани або функціональні блоки додатку.

Докладно опишемо кожен фрейм, починаючи з його назви та призначення. Розглянемо структуру фреймів, їх складові елементи, такі як кнопки, текстові поля, списки, зображення та інші важливі елементи інтерфейсу.

Frame 1. Головна сторінка має приємні постільні кольори, що підкреслюють м'якість та елегантність дизайну. Вона використовує градієнтні ефекти, які створюють плавний перехід між кольорами фотографії заднього фону та градієнту, надаючи сторінці вишуканого вигляду.

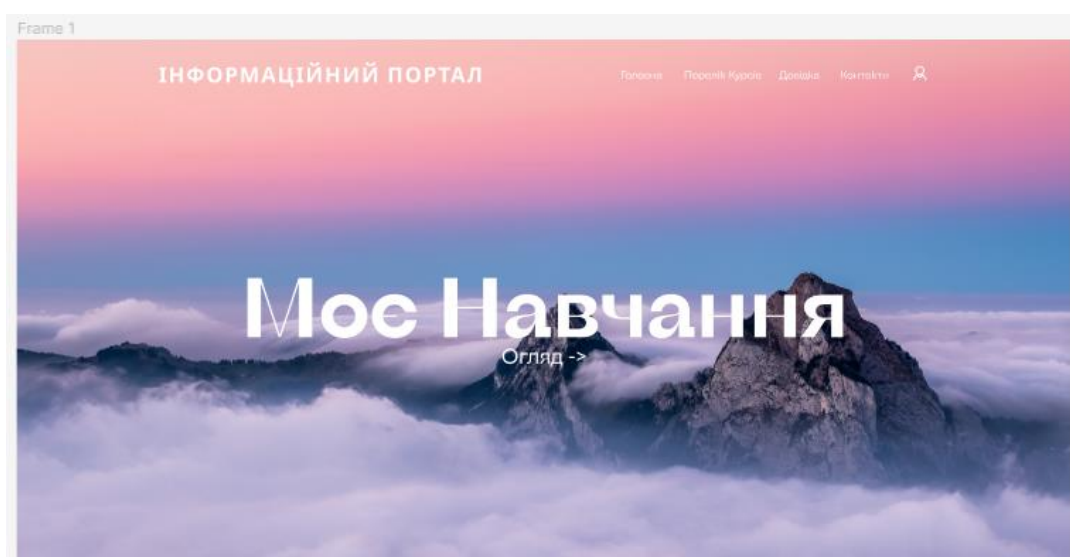


Рис.3.3. Frame 1. Головна сторінка

У футорі головної сторінки розміщена мінімалістична форма зворотного зв'язку, що дозволяє користувачам легко зв'язатись з адміністратором.

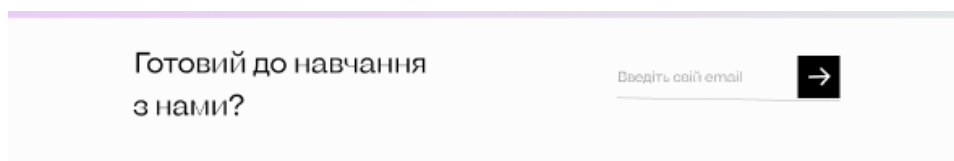


Рис.3.4. Frame 1. Футор головної сторінки

Frame 2. Сторінка з курсами реалізована за допомогою CSS-стилів, що передають ніжні та збалансовані кольори. HTML-розмітка та CSS-стилі дозволяють створити чітку структуру сторінки з окремими блоками для кожного курсу. Це дозволяє відобразити просту та зрозумілу інформацію про курси для користувачів.

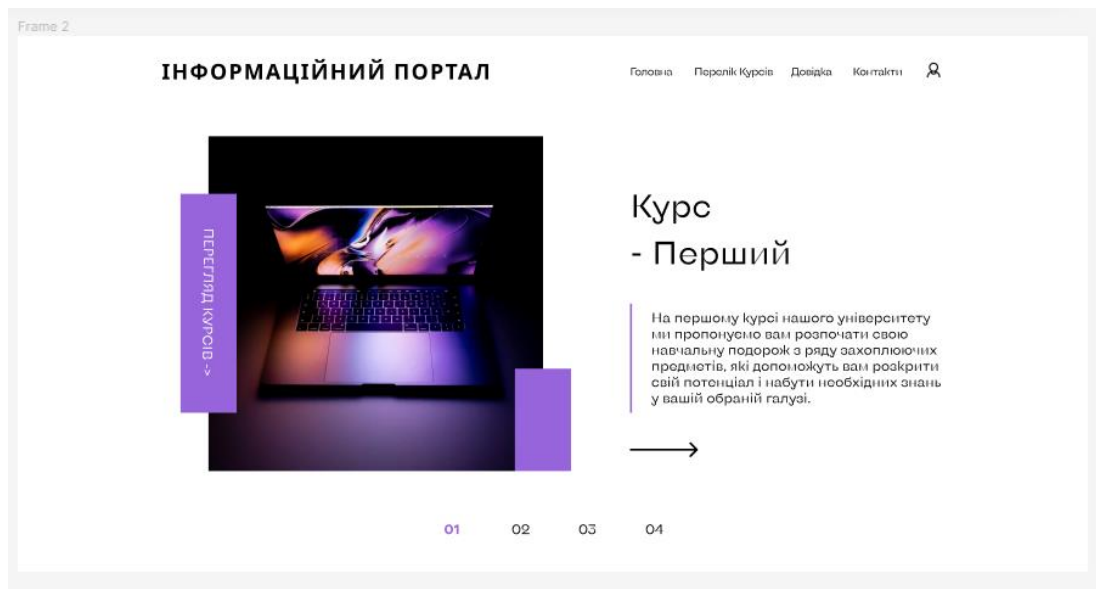


Рис.3.5. Frame 2. Сторінка з переліком курсів

Frame 3. Сторінка з навчальними предметами підкреслює мінімалістичний стиль та елегантність. Вона використовує м'які кольори та прості форми, подібні попередньому фрейму, для створення зручного вигляду. Інформація про кожен предмет відображається зрозуміло та естетично.

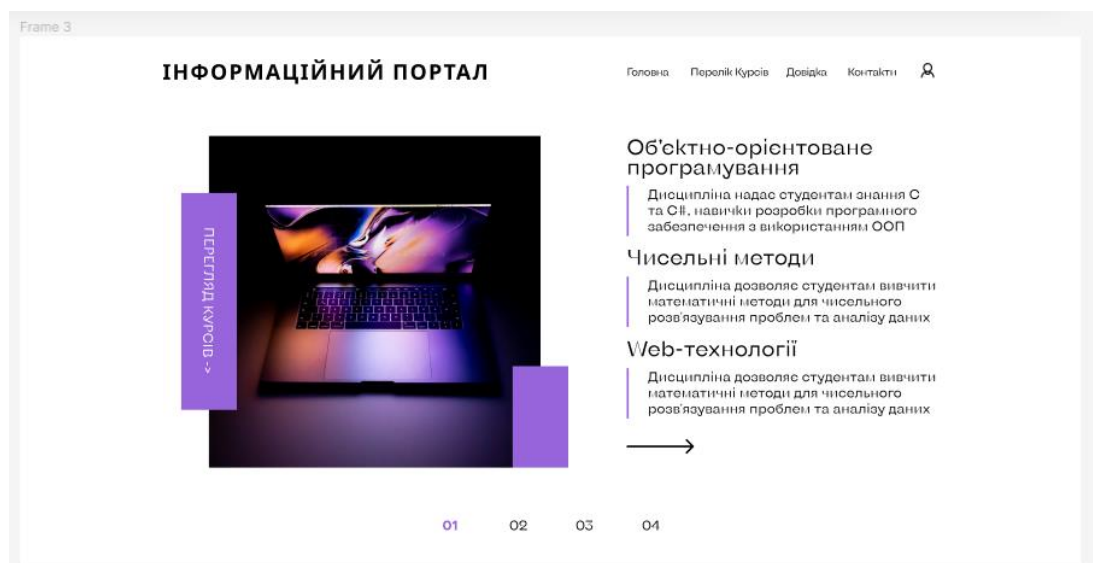


Рис.3.6. Frame 3. Сторінка з навчальними предметами курсів

Frame 4. Особистий кабінет виконаний в стриманій стилістиці, що надає йому спокійний вигляд. Дизайн зосереджений на простоті та лаконічності, забезпечуючи зручний користувацький досвід. Профіль орієнтований на легку функціональність, сприяючи комфортній взаємодії з особистим кабінетом.

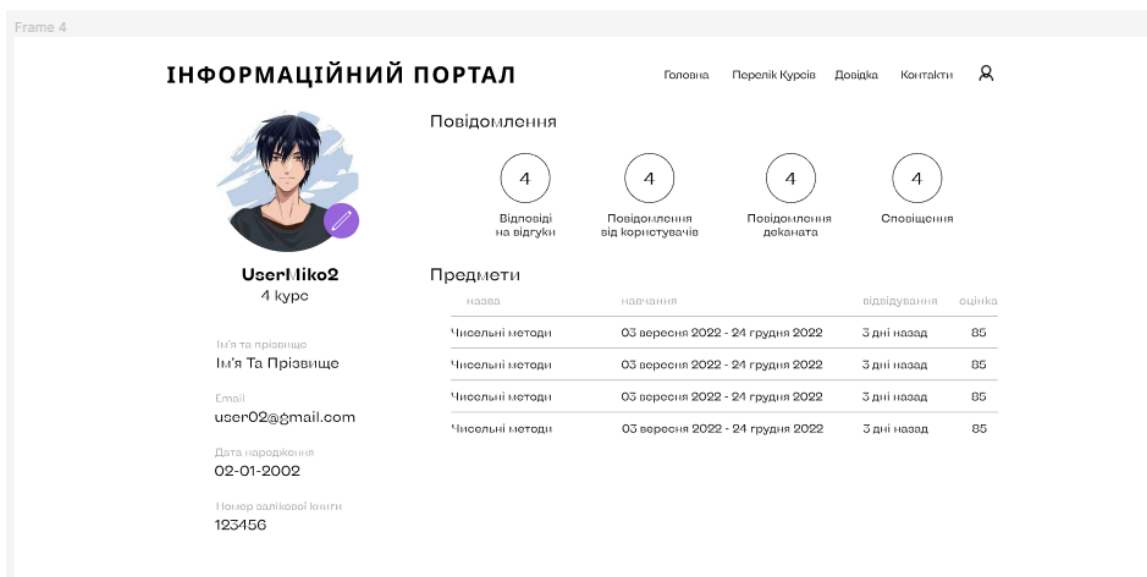


Рис.3.7. Frame 4. Сторінка особистого кабінету користувача

Frame 5. Основна сторінка предмету оформлена в стилі мінімалізму з використанням світлих кольорів та м'яких форм. Вона містить всю необхідну інформацію про предмет, яка відображається зрозуміло та естетично. Лаконічний та збалансований дизайн створює приємну атмосферу для вивчення предмету.

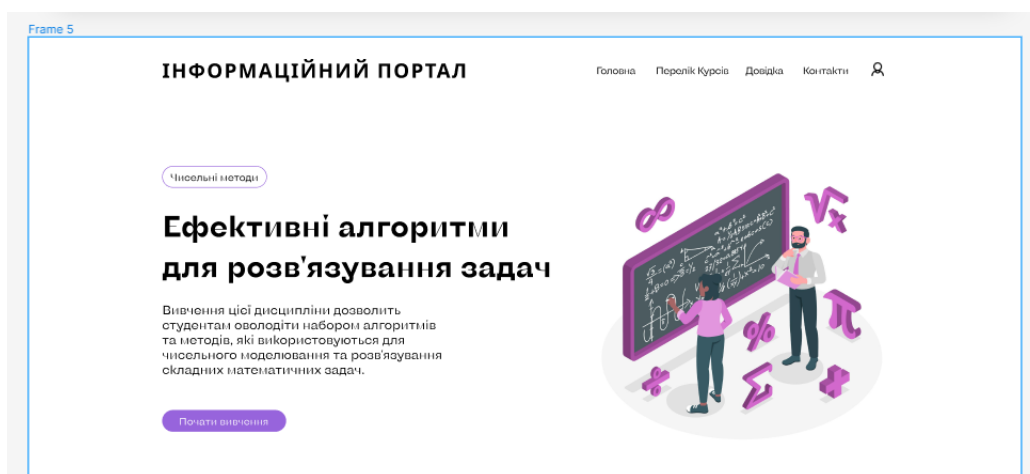
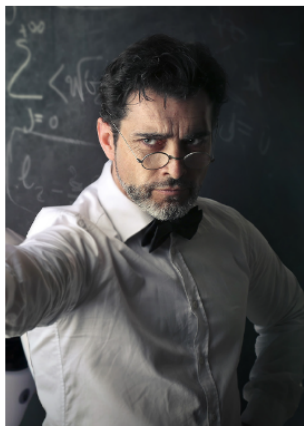


Рис.3.8. Frame 5. Частина основної сторінки з описом предмету

На другій частині сторінки відображається інформація про викладача, яка також відповідає мінімалістичному стилю та забезпечує чітку та зручну навігацію для користувачів. Тут представлена основна інформація про викладача, його кваліфікація та досвід, яка допомагає студентам зрозуміти його роль та значення в навчальному процесі.



Старший викладач

Шелест Ш.Ш.

Науковий ступінь: доктор філософії (Ph.D) з інженерії програмного забезпечення
Освіта: спеціаліст, спеціальність «Інформаційні управляючі системи та технології», кваліфікація "Інженер-програміст"
Наукова діяльність: автор (співавтор) понад 50 наукових та навчально-методичних праць
Наукові інтереси: інформаційні технології, інтелектуальний аналіз даних
Профілі у наукометричних базах:
Державні нагороди та відзнаки: Подяка первинної профспілкової організації викладачів та співробітників ЖДТУ за багаторічну сумлінну працю, вагомий особистий внесок у розвиток університету та з нагоди річниці факультету інформаційно-комп'ютерних технологій. Грамота Житомирського державного технологічного університету за сумлінну та плідну працю, високий професіоналізм та вагомий внесок у підготовку та виховання студентської молоді



Рис.3.9. Frame 5. Частина основної сторінки з інформацією про викладача

Кожен фрейм працює відповідно до логічної структури та функціональності. Всі фрейми орієнтовані на забезпечення комфортної взаємодії з користувачем та надання комплексної інформації.

В результаті детального розгляду кожного фрейму ми забезпечили чітке розуміння структури та функціональності нашого проекту, що є важливим етапом в процесі розробки програмного коду інтерфейсу.

3.3. Тестування роботи веб-орієнтованого додатку

У цьому розділі ми детально розглянемо процес тестування нашого веб-орієнтованого додатку. Ми проведемо різні види функціональних тестів, з метою переконатися в його надійності та оптимальній продуктивності.

Функціональні тести перевіряють, чи працюють окремі функції, модулі чи компоненти програмного продукту згідно з очікуваннями.

Основна мета – забезпечити ефективне виконання потрібних операцій і повертання очікуваних результатів.

Процес тестування головної сторінки проекту включає кроки, що дозволяють перевірити, чи працює сторінка згідно з очікуваннями та заданими вимогами. Основна мета цього тестування - забезпечити, щоб користувач міг легко орієнтуватись на сторінці, розумів її функціональність та взаємодівав з нею без будь-яких проблем.

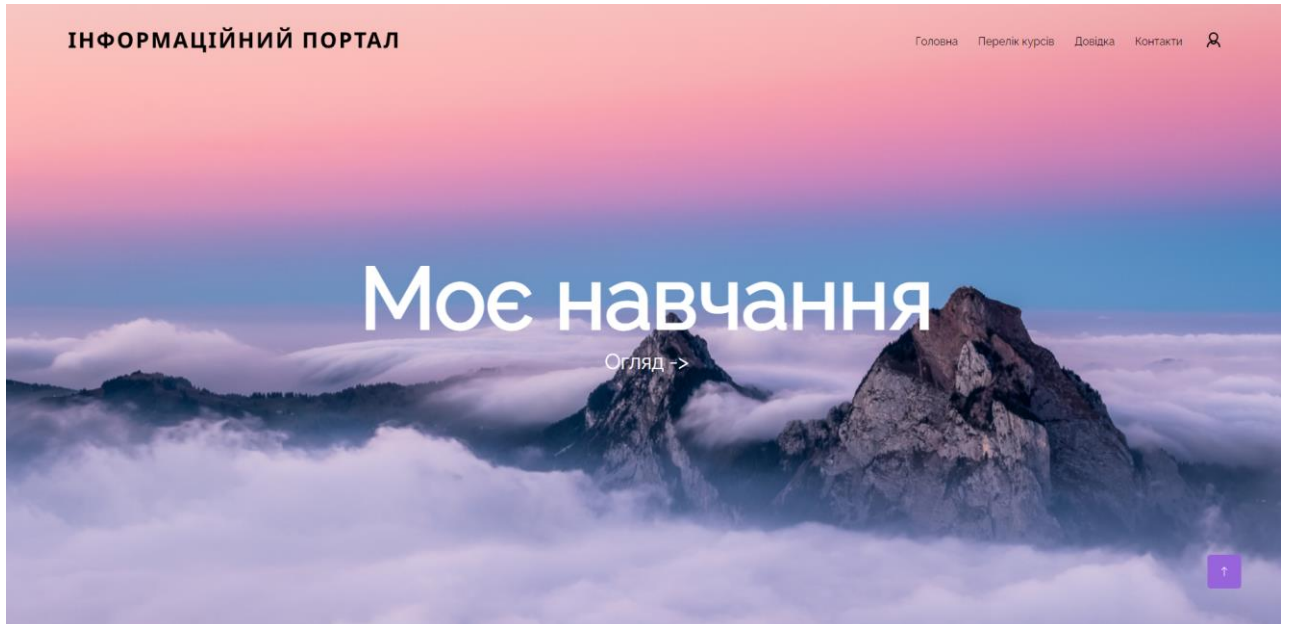


Рис.3.10. Вигляд головної сторінки

Першим елементом, який привертає увагу користувача при відкритті сайту, є кнопка "Огляд". Ця кнопка створена з метою надати користувачу можливість перейти до форми авторизації, після чого він отримає доступ до основних ресурсів проекту.

При натисканні на кнопку користувача перенаправляє на форму авторизації.

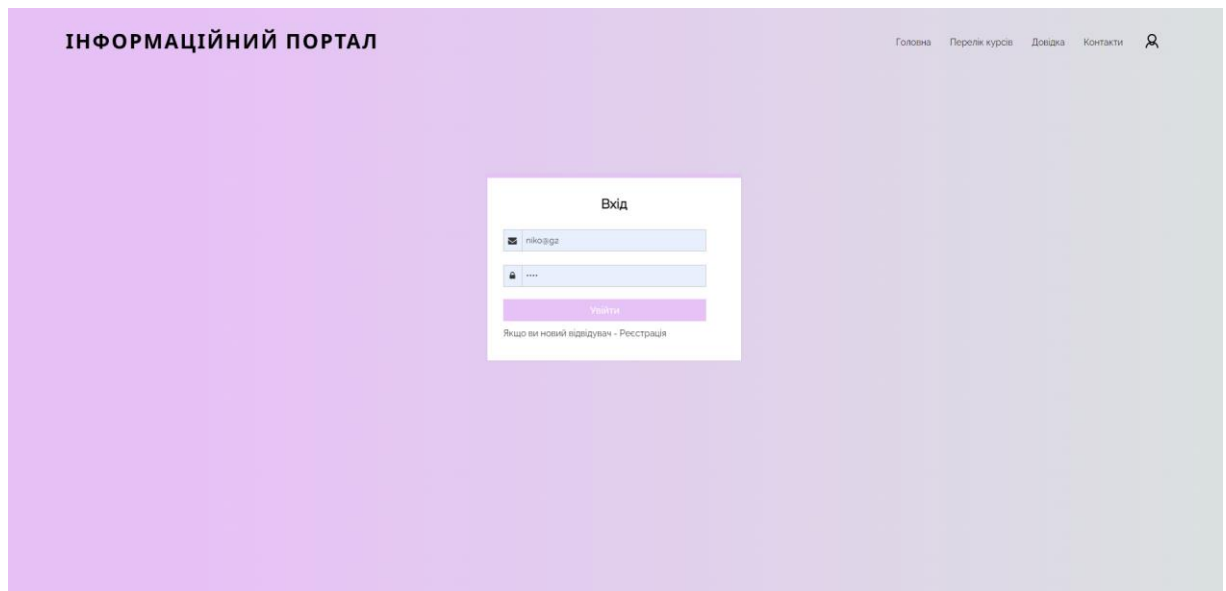


Рис.3.11. Форма авторизації

Якщо у користувача немає особистого кабінету, він може його створити прямо з цієї сторінки, натиснувши на кнопку "Реєстрація" під формою входу.

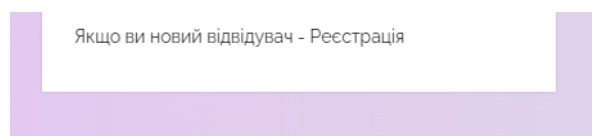


Рис.3.12. Кнопка переходу до форми реєстрації

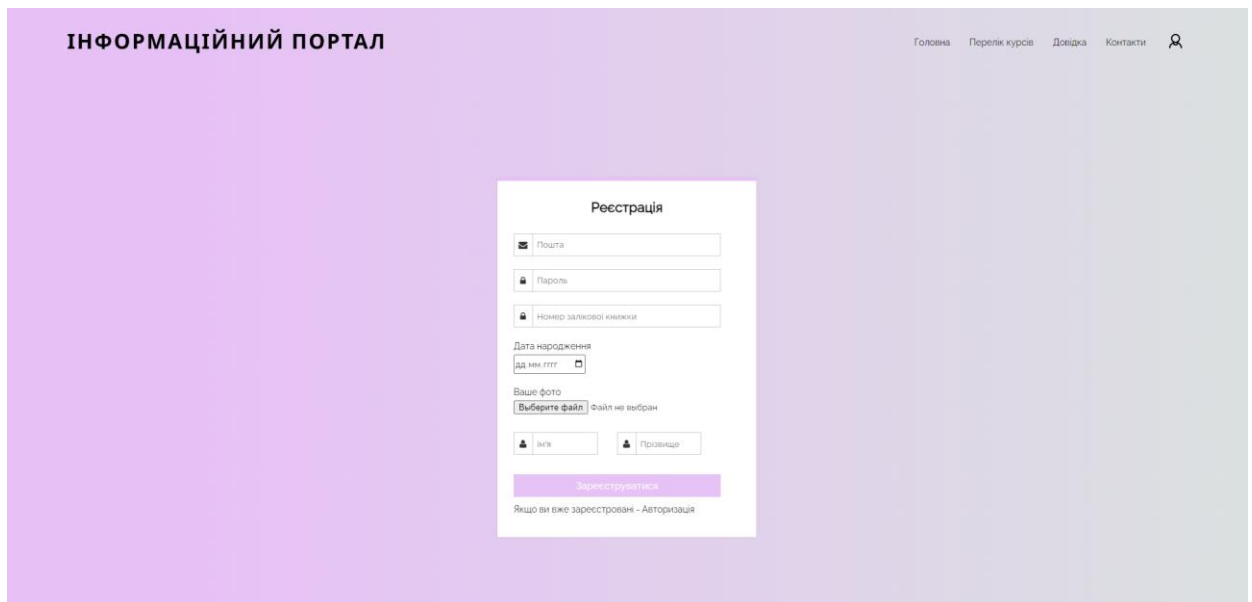


Рис.3.13. Форма реєстрації

Також, до форм авторизації/реєстрації можна перейти, натиснувши іконку особистого кабінету, яка знаходиться в верхньому правому куту сайту.

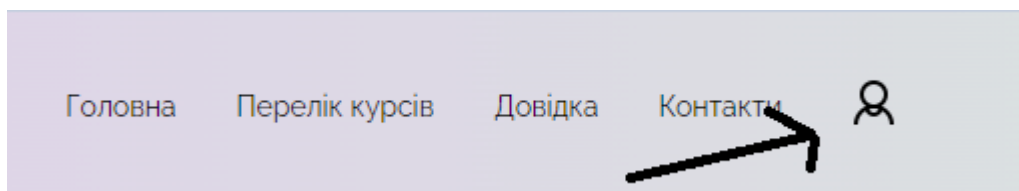


Рис.3.14. Кнопка особистого кабінету

При натисканні на елементи, такі як "Інформаційний портал" та "Головна" виконується функція якірного посилання, реалізована в коді проекту. Це спеціальний механізм, який дозволяє перехопити подію натискання на елемент та здійснити перехід або плавну прокрутку до відповідної частини сторінки, на яку вказує якірне посилання.



Рис.3.15. Елементи "Інформаційний портал" та "Головна"

При натисканні на елементи "Довідка" та "Контакти" також виконується функція якірного посилання, яка реалізована аналогічно до "Інформаційного порталу" та "Головної". Це означає, що при натисканні на ці елементи, користувач буде перенаправлений до відповідних розділів на головній сторінці проекту.



Рис.3.16. Елементи "Довідка" та "Контакти"

"Довідка" перенаправить на інформаційний розділ головної сторінки, де користувач зможе ознайомитись з метою проекту та загальною інформацією про портал.

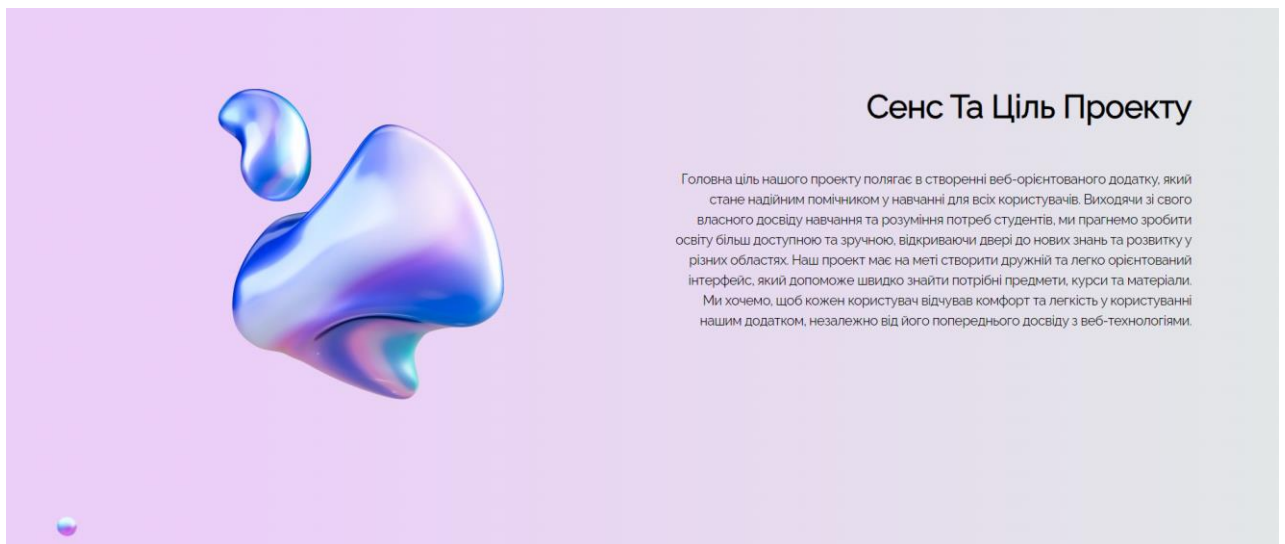


Рис.3.17. Довідка сайту

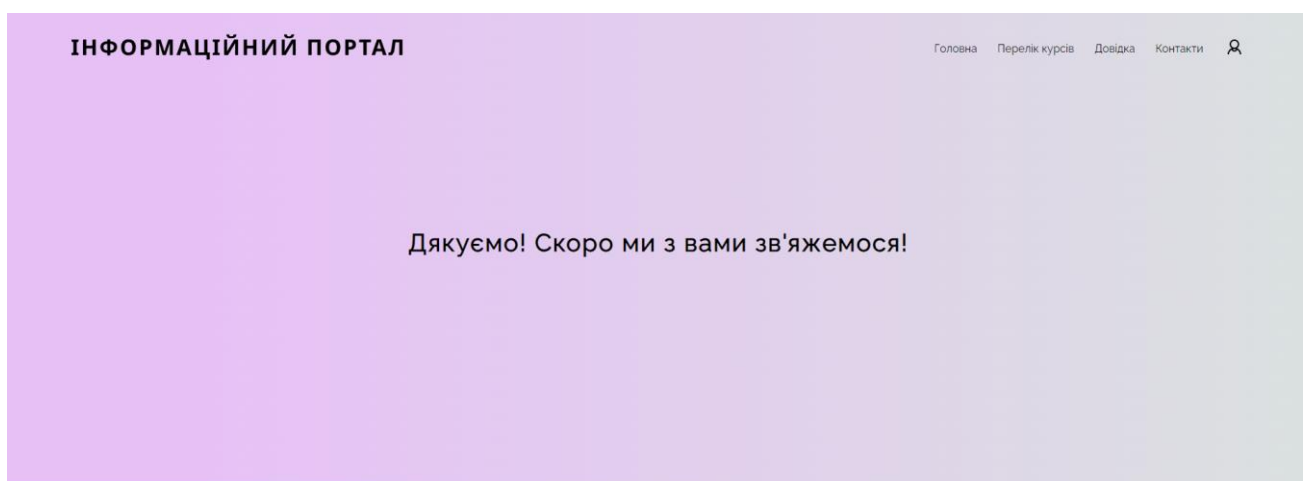
"Контакти" перенаправляють користувача на форму зворотнього зв'язку, де він може залишити свою електронну пошту для адміністратора проекту.

ГОТОВИЙ ДО НАВЧАННЯ З НАМИ?

Введіть свій email



Рис.3.18. Форма зворотнього зв'язку



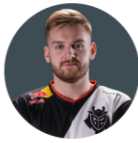
ГОТОВИЙ ДО НАВЧАННЯ З НАМИ?

Введіть свій email



Рис.3.19. Повідомлення про успішну відправку контактів

Перейшовши в кабінет користувач має можливість оцінити свій профіль, переглянувши особисті дані, рейтинг по навчальним предметам, розсилку від адміністратора та є можливість перейти до потрібного курсу.



Студент

Ім'я та прізвище
Никола КовачEmail
niko@gzДата народження
1997-02-16Номер залікової книги
12345

Повідомлення

1

Перейти до курсу

2

Перейти до курсу

3

Перейти до курсу

4

Перейти до курсу

Предмети

назва	навчання	прізвище та ім'я	оцінка
Об'єктно-орієнтоване програмування	2023-06-01 - 2023-07-01	Никола Ковач	75
Веб-технології	2023-07-01 - 2023-08-01	Никола Ковач	90
Чисельні методи	2023-08-01 - 2023-09-01	Никола Ковач	85

Повідомлення

6

Увага! Скоро сесія!

Рис.3.20. Сторінка особистого кабінету

При натисканні на кнопку "Перелік курсів" користувача перенаправляє на сторінку з переліком курсів.

Перелік курсів

Рис.3.21. Елемент "Перелік курсів"

Курс
- перший

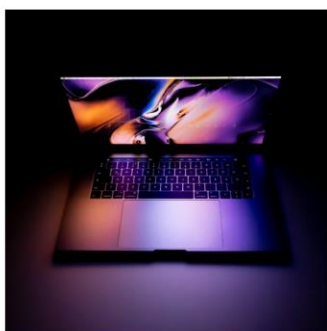
На першому курсі нашого університету ми пропонуємо вам розпочати свою навчальну подорож з ряду захоплюючих предметів, які допоможуть вам розкрити свій потенціал і набути необхідних знань у вашій обраній галузі.



01 02 03 04

Рис.3.22. Сторінка з переліком курсів

Після того, як користувач натисне на певний курс, він буде перенаправлений на сторінку з навчальними предметами, пов'язаними саме з цим курсом.

**Об'єктно-орієнтоване програмування**

| Програмування у стилі ООП: Відкрийте світ об'єктів та можливостей

Веб-технології

| Розкрийте потенціал веб-технологій: Вивчайте сучасні інструменти та тренди

Чисельні методи

| Освоєння потужних інструментів аналізу та розв'язання чисельних задач

Рис.3.23. Сторінка з переліком предметів

На основній сторінці можна переглянути загальну інформацію про обраний предмет, викладача, який його веде, та відгуки студентів про цього викладача. Вона допомагає користувачам отримати повну картину про предмет та викладача перед початком та на протязі навчання.

В першій частині сторінки можна побачити назву, короткий заголовок, та стислий опис до предмету.

Об'єктно-орієнтоване програмування

**Програмування у стилі ООП:
Відкрийте світ об'єктів та
можливостей**

Предмет "Об'єктно-орієнтоване програмування" відкриє вам захоплюючий світ програмування, де ви навчитеся розробляти програми, базовані на концепції об'єктів і класів. Ви отримаєте розуміння принципів ООП, таких як інкапсуляція, спадкування, поліморфізм, та зможете ефективно використовувати ці поняття для створення складних та масштабованих програмних рішень. Проходячи цей курс, ви розвинеєте свої навички проектування, абстракції, та рефакторингу коду, що допоможе вам стати успішним розробником програмного забезпечення.

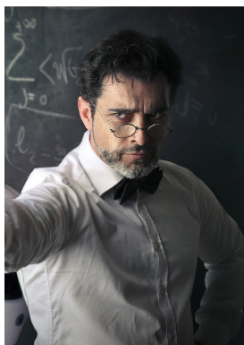
Почати вивчення



Рис.3.24. Основна сторінка з описом предмету

На другій частині сторінки присутня інформація про викладача, який веде предмет. Студенти можуть побачити його загальний опис: ім'я, професійний досвід, кваліфікацію, освіту і т.д. Ці дані допомагають користувачам проаналізувати експертність викладача.

Також розміщені відгуки студентів про викладача. Це дозволяє користувачам оцінити думки та враження інших студентів, які раніше навчалися під керівництвом цього викладача.



Старший викладач

Шелест Ш.Ш.

Науковий ступінь: доктор філософії (Ph.D) з інженерії програмного забезпечення Освіта: спеціаліст, спеціальність «Інформаційні управляючі системи та технології», кваліфікація "Інженер-програміст"
Наукова діяльність: автор (співавтор) понад 50 наукових та навчально-методичних праць Наукові інтереси: інформаційні технології, інтелектуальний аналіз даних Профілі у наукометричних базах: Державні нагороди та відзнаки: Подяка первинної профспілкової організації викладачів та співробітників ЖДТУ за багаторічну сумлінну працю, вагомий особистий внесок у розвиток університету та з нагоди річниці факультету інформаційно-комп'ютерних технологій. Грамота Житомирського державного технологічного університету за сумлінну та плідну працю, високий професіоналізм та вагомий внесок у підготовку та виховання студентської молоді

Відгуки молодих спеціалістів Житомирської Політехніки:

Педрюк П.І.

"Могутній"

Скала Д.О.

"Дуже добре веде дисципліну та подає матеріал"

Квас Р.Р.

"Нормальний"



Рис.3.25. Основна сторінка з інформацією про викладача

Не забуваємо про адміністратора проекту який має велику відповідальність за ефективне керування та розвиток системи. Тестування функціоналу адміністратора є важливим кроком для того, щоб переконатися, що всі потрібні функції працюють належним чином і забезпечують нам повний контроль над проектом.

Головна Додавання предметів Керування користувачами Відправка смс Оцінки Заявки Вихід

Рис.3.26. Панель керування адміністратора

Перше що ми бачимо, адміністратор має можливість додавати нові предмети до системи. Під час тестування перевірили, чи правильно додаються нові предмети з усією необхідною інформацією, такою як назва, опис, викладач тощо.

Додавання предмета

Графічний дизайн

Графічний дизайн

Графічний дизайн — це мистецтво та практика створення візуальних матеріалів для передачі ідей, інформації та емоцій. Він поєднує творчість і технічні навички для вирішення комунікаційних завдань, використовуючи елементи, як-от текст, зображення, кольори, форми і простір.

Фото викладача

Вибрати файл AdobeStock_58384118o_junghc1-scaled.jpeg

Старший викладач

Графічна О.В.

Завдяки його індивідуальному підходу, студенти розвивають вміння працювати в дизайні не тільки для задоволення естетичних вимог, а й з урахуванням практичних завдань: адаптація дизайну до різних платформ, створення брендів та розробка комунікаційних стратегій. Важливою рисою такого викладача є його здатність надихати на творчі пошуки, допомагаючи кожному знайти свій стиль і підхід.

Викладач графічного дизайну — це людина, яка відкриває студентам двері до світу візуальних комунікацій, надихаючи їх на створення справжніх шедеврів.

Виберіть курс

2

Надіслати

Рис.3.27. Приклад додавання предмету

10	Системний аналіз та теорія прийняття рішень	Системний аналіз та прийняття рішень: Розкриття потенціалу ефективного управління	Предмет "Системний аналіз та теорія прийняття рішень" допоможе освоїти методи та інструменти для аналізу та управління складними системами. Вивчення концепцій системного аналізу та теорії прийняття рішень допоможе вам збирати та аналізувати дані, оцінювати альтернативи та приймати ефективні рішення. Курс також надасть практичні навички роботи з інструментами системного аналізу, розвине критичне мислення та прийняття рішень, що є важливими у будь-якій сфері діяльності, де потрібно ефективно управляти складними системами.	Редагувати	Видалити
----	---	---	---	------------	----------

Рис.3.28. Результат додавання предмету

Також бачимо, що адміністратор може редагувати та видаляти існуючі предмети. Під час тестування ми переконались, що зміни вносяться коректно, а інформація про предмет оновлюється на всіх відповідних сторінках.

Редагування предмета

Об'єктно Орієнтовне Програмування

ООП

Об'єктно-орієнтоване програмування (ООП) — це парадигма програмування, яка організовує програмний код у вигляді об'єктів, кожен з яких містить як дані (властивості), так і методи (функції), що оперують цими даними. ООП спрощує розробку програм, підвищує їх зручність для підтримки і масштабування, а також дозволяє краще моделювати реальні системи.

Старший викладач

Шелест Ш.Ш.

Науковий ступінь: доктор філософії (Ph.D) з інженерії програмного забезпечення
Освіта: спеціаліст, спеціальність «Інформаційні управляючі системи та технології», кваліфікація "Інженер-програміст"
Наукова діяльність: автор (співавтор) понад 50 наукових та навчально-методичних праць
Наукові інтереси: інформаційні технології, інтелектуальний аналіз даних

Підтвердити

Рис.3.29. Модуль редагування предмету

Адміністратор має можливість переглядати базу даних користувачів та видаляти їх облікові записи. Ми переконались, що дані користувачів відображаються правильно, і видалення облікових записів користувачів відбувається безпомилково.

Головна	Додавання предметів	Керування користувачами	Відправка смс	Оцінки	Заявки	Вихід	
ID	Ім'я	Прізвище	Пошта	Дата народження	Номер залікової книжки	Видалити користувача	
4	Name	Surname	test@test	2023-05-31	testtest	Видалити	
7	John	Rembo	user@user	2023-06-15	usernumber	Видалити	
8	Никола	Ковач	niko@gg2	1997-02-16	12345	Видалити	

Рис.3.30. Форма керування користувачами

Адміністратор може створювати та видаляти розсилку повідомлень для користувачів у їх особисті кабінети. Під час тестування ми перевірили, що повідомлення досягають користувачів, та студенти мають змогу переглядати

повідомлення в своїх кабінетах. Також перевірили, що всі повідомлення видаляються належним чином.

SMS розсилка

Надіслати

ID	SMS	Видалення
6	Увага! Скоро сесія!	Видалити

Рис.3.31. Форма створення повідомлень та можливість їх видалення

Повідомлення

Увага! Скоро сесія!	6
---------------------	---

Рис.3.32. Результат відображення повідомлення в кабінеті користувача

Адміністратор може виставляти оцінки певним користувачам за певні предмети. Під час тестування перевірили, що оцінки виставляються правильно та вони відображаються у студентів в особистому кабінеті.

Додавання оцінки

Ім'я та прізвище студента

Никола Ковач

Предмет

Об'єктно-орієнтоване програмування

Дата початку навчання

дд.мм.гггг

Дата кінця навчання

дд.мм.гггг

Оцінка

від 0 до 100

Надіслати

Рис.3.33. Форма додавання оцінок

Ім'я та прізвище студента	Предмет	Навчання	Оцінка	Редагування	Видалення
John Rembo	Об'єктно-орієнтоване програмування	2023-06-01 - 2023-07-01	95	Редагувати	Видалити
Никола Ковач	Об'єктно-орієнтоване програмування	2023-06-01 - 2023-07-01	75	Редагувати	Видалити
Никола Ковач	Веб-технології	2023-07-01 - 2023-08-01	90	Редагувати	Видалити
Никола Ковач	Чисельні методи	2023-08-01 - 2023-09-01	85	Редагувати	Видалити

Рис.3.34. Відображення оцінок в системі адміністратора

Змінити бал

[Підтвердити](#)

Рис.3.35. Форма редагування оцінок

Предмети

назва	навчання	прізвище та ім'я	оцінка
Об'єктно-орієнтоване програмування	2023-06-01 - 2023-07-01	Никола Ковач	75
Веб-технології	2023-07-01 - 2023-08-01	Никола Ковач	90
Чисельні методи	2023-08-01 - 2023-09-01	Никола Ковач	85

Рис.3.36. Результат відображення оцінок в кабінеті користувача

І останнє, адміністратор має можливість переглядати контакти, які надійшли через форму зворотнього зв'язку від відвідувачів та користувачів. Під час тестування ми перевірили, що всі надіслані контакти в системі відображаються коректно.

ID заявки	Електронна пошта
1	123
2	123@123
3	dimonkravchuk151@gmail.com
4	dimonkravchuk151@gmail.com

Рис.3.37. Відображення контактів в системі адміністратора

Провівши тестування функцій адміністратора ми переконались, що всі вони працюють належним чином, забезпечують потрібний функціонал та безпроблемне управління системою.

Робота з асистентом. Приклади та варіанти використання.

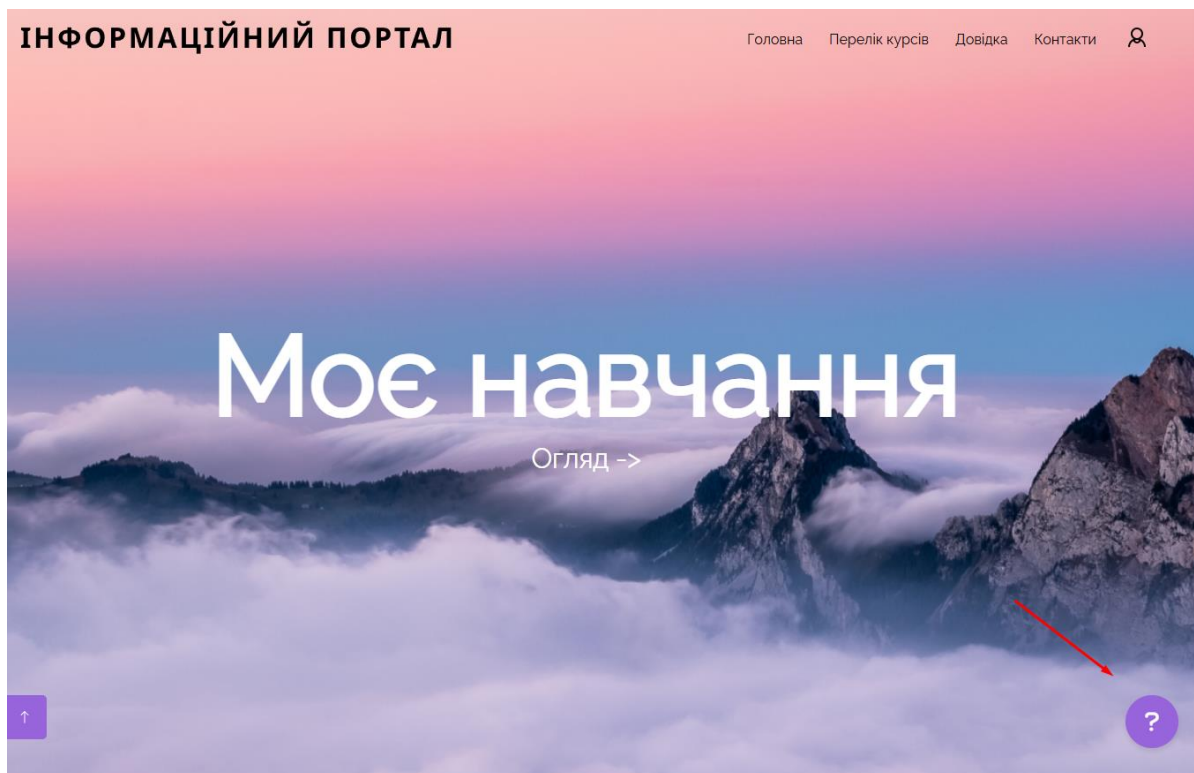


Рис.3.38. Головна сторінка, іконка асистента

Перевірка відповідей на типові запити користувачів.

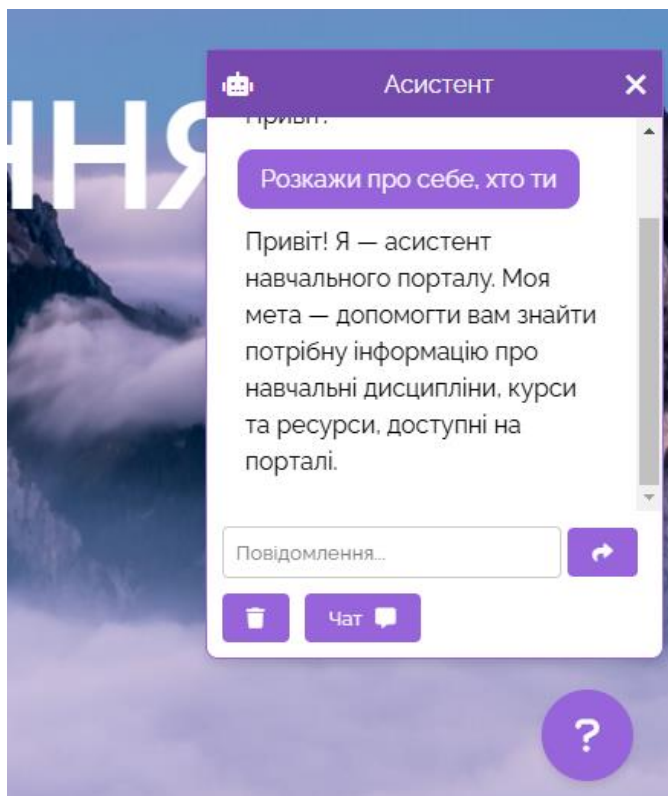


Рис.3.39. Приклад питання

З міні чату ми можемо перейти на окремо сторінку, історія повідомлень зберігається.

ІНФОРМАЦІЙНИЙ ПОРТАЛ

[Головна](#) [Перелік курсів](#) [Довідка](#) [Контакти](#) [Адмінка](#) [Вийти](#)

Розкажи про себе розгорнуто

Привіт! Я — ваш асистент на навчальному порталі. Моя мета — допомогти вам швидко та ефективно знайти необхідну інформацію про навчальні дисципліни, курси та ресурси, доступні на нашому порталі. Я працюю з базою даних, що містить інформацію про предмети, викладачів та структуру курсів. Якщо у вас виникнуть питання щодо того, як користуватися порталом, зверніть увагу на головну сторінку — там є вся необхідна інформація, зокрема, контакти для зв'язку. Для доступу до сторінки курсів та вашого особистого кабінету, вам потрібно авторизуватися на порталі. Я завжди радий допомогти!

Які предмети веде Шелест?

Добрий день! Шелест Ш.Ш. веде предмет "ООП" (Об'єктно-орієнтоване програмування). Це викладач високого рівня.

 [Очистити сесію](#)

Рис.3.40. Окрема сторінка чату

У разі введення питання, яке не стосується тематики проекту, ми отримаємо відповідь, яка буде про це нас свідчити.

ІНФОРМАЦІЙНИЙ ПОРТАЛ

[Головна](#) [Перелік курсів](#) [Довідка](#) [Контакти](#) 

Привіт як у тебе справи?

Привіт! У мене все добре, дякую за запитання. Чим можу допомогти?

Я мені знайти бегемота в Африці?

На жаль, я не можу допомогти з цим питанням. Мене запрограмовано надавати інформацію лише про навчальні дисципліни, курси та ресурси навчального portalу.

 [Очистити сесію](#)

Рис.3.41. Питання, яке не стосується тематики проекту

Приклад типового використання асистента щодо веб сайту.

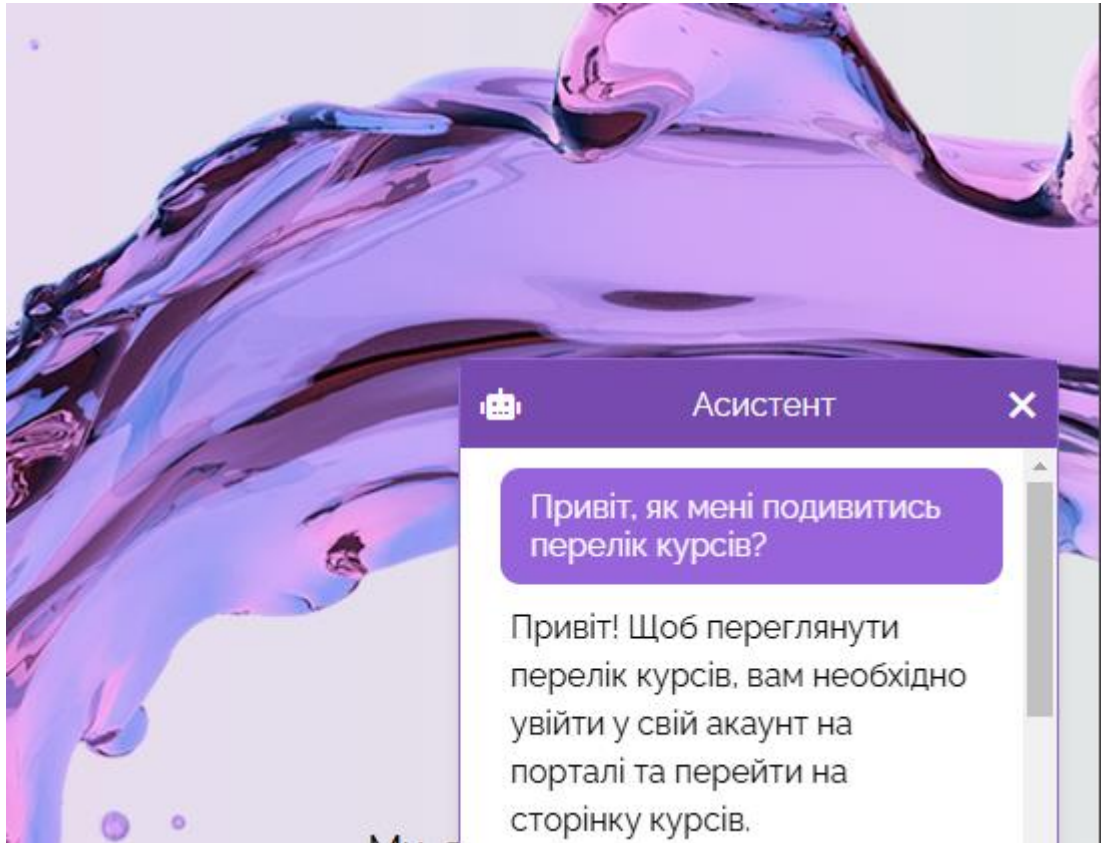


Рис.3.42. Стандартне ймовірне питання користувача

Провівши тестування функцій ШІ-асистента ми переконались, що всі вони працюють належним чином, забезпечують потрібний функціонал та безпроблемне управління системою.

Висновки до 3 розділу

У цьому розділі ми детально розглянули кілька важливих аспектів проекту – архітектура та його розгортання, проектування загального інтерфейсу та тестування роботи веб-орієнтованого додатку. За допомогою цих підрозділів ми покрили ключові етапи процесу розробки та впровадження додатку, забезпечуючи ефективність та якість його функціонування.

Ми розглянули важливі аспекти структури додатку, використовуючи модульний підхід, який сприяє зручності розробки, підтримці та розширенню. Також обговорили процес розгортання додатку, враховуючи вибір хостингової платформи та конфігурацію серверного середовища.

Далі, зосередились на розробці естетичного та користувацьки-орієнтованого інтерфейсу. Використовуючи мінімалістичний стиль з глассморфізмом, ми створили зручні та привабливі фрейми, враховуючи сучасні тенденції дизайну. Також впровадили функціональні елементи, які забезпечують зручну навігацію для користувачів та забезпечують ефективність використання додатку.

І звісно, ми провели роботу над перевіркою функціональності та взаємодії користувачів з додатком. Провели тестування головної сторінки, кабінету користувача та окремих функцій, включаючи адміністрування. Цей процес дозволив нам виявити та усунути помилки, забезпечити безперебійну роботу та зручність використання додатку користувачами.

Також було проведено тестування ШІ-компонентів веб-системи. Перевірено точність відповідей, швидкість обробки запитів та інтеграцію з базою даних. Результати тестування підтвердили високу ефективність роботи ШІ-асистента, а також його здатність забезпечувати якісну взаємодію з користувачами.

Загалом, завдяки архітектурі додатку, розробленому інтуїтивному інтерфейсу та тестуванню функціоналу, ми забезпечили створення веб-орієнтованого додатку, який надає зручний та задовільний користувацький досвід. Дотримуючись передових методів розробки та використовуючи найкращі практики, ми створили надійний та ефективний продукт, який готовий задовольнити потреби наших користувачів.

ВИСНОВКИ

В процесі роботи було проведено детальний аналіз вимог користувача і враховано різноманітні аспекти, що стосуються функціональності та зручності інтерфейсу. Шляхом розгляду існуючих аналогів були визначені основні функції та вимоги до майбутнього веб-додатку. Також було виявлено потенційні можливості для покращення та розширення функціональності.

У рамках дослідження було розглянуто сучасні можливості використання штучного інтелекту у веб-системах, зокрема його роль у підвищенні функціональності та автоматизації процесів. Особливу увагу приділено інтеграції ШІ в освітні платформи для забезпечення зручності доступу до навчальних ресурсів і персоналізації взаємодії з користувачами.

Модель Google Gemini обрана як основа для реалізації ШІ-асистента через її мультимодальні можливості, високу продуктивність та доступність. Використання Google Gemini у цьому проекті дозволило створити ефективного помічника, який забезпечує швидке надання відповідей на питання користувачів, пошук інформації в базі даних порталу університету та зручну навігацію по порталу для студентів і викладачів.

При проектуванні веб-орієнтованої системи інформаційного порталу, було ретельно розглянуто структуру та варіанти використання системи. Була розроблена архітектура системи та встановлено залежності між її компонентами. Також була розроблена структура бази даних та файлової системи, що забезпечують ефективне зберігання та організацію даних.

Порядок роботи та тестування системи були чітко визначені та виконані. Розроблені загальний та функціональний алгоритми веб-додатку, а також проведені тестування їх роботи. Цей процес дозволив перевірити функціональність, надійність та зручність використання додатку. Всі функції ШІ-асистента працюють коректно та правильно, забезпечуючи високу точність відповідей, стабільність роботи та відповідність очікуванням користувачів.

У підсумку, результатом нашої магістерської роботи є розроблений продукт, який задовольняє потреби користувача. Додаток має приємний та зручний інтерфейс, що сприяє його легкому використанню. Крім того, заздалегідь визначений вектор розвитку дозволить покращити функціонал додатку та привернути нову аудиторію в майбутньому.

Розроблений проект продемонстрував, що інтеграція штучного інтелекту у веб-системи відкриває нові можливості для автоматизації, покращення користувацького досвіду та підвищення ефективності освітніх процесів. Подальший розвиток у цьому напрямку дозволить створити ще більш досконалі рішення, які сприятимуть трансформації освітніх платформ у майбутньому.

В процесі роботи над проектом, був здобутий значний досвід у розробці веб-додатків та керуванні проектами. Вибір відповідних технологій та архітектури, а також ретельне аналізування вимог користувача дозволили досягти успішного результату.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Документація MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>
2. Документація HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/HTML>
3. Документація CSS [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/CSS>
4. Документація JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
5. Документація CSS Grid [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout
6. Документація PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/docs.php>
7. Платформа вивчення PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/tutorial.php>
8. Документація MAMP [Електронний ресурс] – Режим доступу до ресурсу: <https://documentation.mamp.info/>
9. Salunke, S., Ouda, A. (2024). A Performance Benchmark for the PostgreSQL and MySQL Databases. *Future Internet*, 16(10):382. DOI: 10.3390/fi16100382. License: CC BY 4.0.
10. Kokala, A., Kalluri, K. (2024). Performance Benchmarking of Generative AI Models: ChatGPT-4 vs. Google Gemini AI. *International Research Journal of Modernization in Engineering Technology and Science*, 6(11), 4673-4677. DOI: 10.56726/IRJMETs64283.
11. OpenAI. (2024). ChatGPT: Advanced AI Model for Language Understanding. OpenAI Research. [Електронний ресурс] – Режим доступу до ресурсу: <https://openai.com/research/chatgpt>
12. Google Gemini AI Chat [Електронний ресурс] – Режим доступу до ресурсу: <https://gemini.google.com/>

13. Anthropic. (2024). Claude AI: Ethical and Transparent Generative Model [Электронный ресурс] – Режим доступа до ресурсу: <https://www.anthropic.com/claude>
14. Jadhav, S. S., Gholve, S. S. (2024). The Impact of AI on Web Development. International Journal of Scientific Research in Modern Science and Technology, 3(8), 07-12. DOI: 10.59828/ijrmst.v3i8.240. License: CC BY-NC 4.0.
15. Bing Li, Miaomiao Peng. Integration of an AI-Based Platform and Flipped Classroom Instructional Model. Scientific Programming, 2022. <https://doi.org/10.1155/2022/2536382>.
16. Lee, K. (2021). Database Management Systems: MySQL in Education. O'Reilly Media. <https://www.oreilly.com/library/view/database-management-systems>
17. Google Gemini API [Электронный ресурс] – Режим доступа до ресурсу: <https://ai.google.dev/>
18. Yellavula (2020). Hands-On RESTful Web Services with Go: Develop elegant RESTful APIs with Golang for microservices and the cloud, Second Edition
19. Google AI Studio [Электронный ресурс] – Режим доступа до ресурсу: <https://aistudio.google.com/>
20. Fitzgerald, J. (2020). MySQL 8 Administrator's Guide: A Practical Guide to MySQL 8 Database Administration. Packt Publishing. <https://www.packtpub.com/product/mysql-8-administrators-guide/9781788621755>

ДОДАТКИ

Код інтеграції роботи з Gemini API

```

define('GEMINI_API_KEY', 'gemini-api-key');
define('MODEL', 'gemini-1.5-flash');
define('BASEURL', 'https://generativelanguage.googleapis.com/v1beta');

header('Content-Type: application/json');

$request = json_decode(file_get_contents('php://input'), true);

if (is_null($request)) {
    echo json_encode(['reply' => 'JSON decoding failed: ' .
    json_last_error_msg()]);
    exit;
}

$userMessage = $request['message'] ?? '';

if (!$userMessage) {
    echo json_encode(['reply' => 'No message provided.']);
    exit;
}

$subjects = [];
$query = "SELECT `title`, `description`, `name`, `course` FROM `subject`";
$result = mysqli_query($connect, $query);

if ($result && mysqli_num_rows($result) > 0) {
    while ($row = mysqli_fetch_assoc($result)) {
        $subjects[] = "Назва: {$row['title']}, Опис: {$row['description']},
Викладач: {$row['name']}, Курс: {$row['course']}";
    }
}

$subjectData = implode("\n", $subjects);

$contextPrompt = "Використовуй це тільки коли питання з цмм пов'язано. Ось список
предметів, доступних у базі:\n" . $subjectData;

$data = array(
    'contents' => [
        [
            'parts' => [
                ['text' => $initialPrompt],
                ['text' => $contextPrompt],
                ['text' => $userMessage]
            ]
        ]
    ]
]

```

```

);

$apiUrl = sprintf("%s/models/%s:generateContent?key=%s", BASEURL, MODEL,
GEMINI_API_KEY);

$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, $apiUrl);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, [
    'Content-Type: application/json',
]);
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
if ($response === false) {
    echo json_encode(['reply' => 'CURL error: ' . curl_error($ch)]);
    curl_close($ch);
    exit;
}

$httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);

curl_close($ch);

if ($httpCode !== 200) {
    echo json_encode(['reply' => "Error: Received HTTP code $httpCode"]);
    exit;
}
$responseData = json_decode($response, true);

if (isset($responseData['candidates'][0]['content']['parts'][0]['text'])) {
    $chatbotReply = $responseData['candidates'][0]['content']['parts'][0]['text'];
} else {
    $chatbotReply = 'Sorry, I could not process that.';
}
echo json_encode(['reply' => $chatbotReply]);

```

Слухачі подій для сторінки міні чату

```

<script>
  function saveHistoryToLocalStorage(history) {
    localStorage.setItem('chatHistory', JSON.stringify(history));
  }

  function getHistoryFromLocalStorage() {
    const history = localStorage.getItem('chatHistory');
    return history ? JSON.parse(history) : [];
  }

  function clearHistoryFromLocalStorage() {
    localStorage.removeItem('chatHistory');
  }

  function loadChatHistory() {
    const messages = document.getElementById('aiMessages');
    const history = getHistoryFromLocalStorage();

    history.forEach(entry => {
      const messageDiv = document.createElement('div');
      messageDiv.classList.add('message', entry.type);
      messageDiv.textContent = entry.text;
      messages.appendChild(messageDiv);
    });

    messages.scrollTop = messages.scrollHeight;
  }

  function updateChatHistory(type, text) {
    const history = getHistoryFromLocalStorage();
    history.push({
      type,
      text
    });
    saveHistoryToLocalStorage(history);
  }

  async function sendMessage() {
    const userInput = document.getElementById('aiUserInput');
    const messages = document.getElementById('aiMessages');

    if (!userInput.value.trim()) return;

    const userMessage = document.createElement('div');
    userMessage.classList.add('message', 'user');
    userMessage.textContent = userInput.value;
    messages.appendChild(userMessage);
  }

```

```

        updateChatHistory('user', userInput.value);

        messages.scrollTop = messages.scrollHeight;

        const response = await fetch('func/ai-assistant.php', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json'
            },
            body: JSON.stringify({
                message: userInput.value
            })
        });

        const result = await response.json();

        const botMessage = document.createElement('div');
        botMessage.classList.add('message', 'bot');
        botMessage.textContent = result.reply;
        messages.appendChild(botMessage);
        updateChatHistory('bot', result.reply);

        messages.scrollTop = messages.scrollHeight;
        userInput.value = '';
    }

    function clearChatHistory() {
        clearHistoryFromLocalStorage();
        const messages = document.getElementById('aiMessages');
        messages.innerHTML = '';
    }

    function toggleChat() {
        const chatBox = document.getElementById('aiChatBox');
        chatBox.style.display = chatBox.style.display === 'flex' ? 'none' :
'flex';
    }

    function goToFullChat() {
        window.location.href = '/ai-page.php';
    }

    document.addEventListener('DOMContentLoaded', loadChatHistory);
</script>

```

Головна сторінка веб сайту

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="./assets/style/index.css">
</head>

<body>
  <style>
    body,
    html {
      background: linear-gradient(90deg, rgba(229, 189, 246, 0.74) 15.91%,
        rgba(216, 222, 222, 0.74) 100%);
    }
  </style>
  <div class="main-screen">
    <?php include './temp/header.php'; ?>
    <div class="main-screen__description">
      <h1>Моє навчання</h1>
      <a href="./account.php">Огляд -></a>
    </div>
  </div>
  <a name="dovidka"></a>
  <div class="main-col2 screen-1">
    <div class="main-col2__wrapper">
      <div class="main-col2__img">
        
      </div>
      <div class="main-col2__txt">
        <h2>Сенс та ціль проекту</h2>
        <p class="mt-65">Головна ціль нашого проекту полягає в створенні
веб-орієнтованого додатку,
        який стане надійним помічником у навчанні для всіх
користувачів. Виходячи зі свого власного
        досвіду навчання та розуміння потреб студентів, ми прагнемо
зробити освіту більш доступною
        та зручною, відкриваючи двері до нових знань та розвитку у
різних областях.
        Наш проект має на меті створити дружній та легко орієнтований
інтерфейс,
        який допоможе швидко знайти потрібні предмети, курси та
матеріали. Ми хочемо, щоб кожен
        користувач відчував комфорт та легкість у користуванні нашим
додатком, незалежно від його

```

```

        попереднього досвіду з веб-технологіями.
    </p>
</div>
</div>
</div>
<div class="main-col2 screen-2">
    <div class="main-col2__wrapper">
        <div class="main-col2__img">
            <p>
                Ми також мали можливість вивчати ці предмети з викладачами, які
                дійсно були експертами в своїй галузі.
                Вони передавали свої знання та досвід з великим ентузіазмом,
                надихаючи нас на подальший
                розвиток і досягнення успіху.
            <br><br>
                Звісно, не все було ідеально, але кожен випадок неперфектності
                став для нас цінним досвідом та підказкою
                на шлях до подальшого удосконалення. Ми завжди прагнули до
                зростання і вдосконалення, враховуючи отримані уроки.
            </p>
        </div>
        <div class="main-col2__txt end-text">
            <p>Ми, як і всі, мали проблему з розумінням деяких предметів, але
            це було більше через нашу нецілеспрямованість
            та неувагу в деяких моментах.</p>
        </div>
    </div>
</div>
</div>
<div class="main-col2 screen-1">
    <div class="main-col2__wrapper">
        <div class="main-col2__img">
            
        </div>
        <div class="main-col2__txt">
            <p class="mt-65">Кожен предмет представлений зібраною інформацією,
            яка допомагає користувачам ознайомитись
            з його основними характеристиками, такими як назва, опис, обсяг
            матеріалу, та інші деталі. Крім того,
            наша платформа надає можливість переглянути інформацію про
            викладачів, їхні кваліфікації та досвід.</p>
        </div>
    </div>
</div>
<div class="main-col2 screen-4">
    <div class="text-center">
        <p>Наша мета</p>
        <h2>розповісти про поточну систему навчання на 4-х курсах
        університету</h2>
    </div>
</div>

```

```

<div class="main-col2 screen-5">
  <p class="text-right">
    Ми вважаємо, що цей проект може бути корисним для студентів, які мають
    намір вступати до університету
    або для тих, хто вже навчається на одному з 4-х курсів. Окрім того, в
    проекті ми плануємо детально розглянути
    кожен з предметів, які ми вивчали, і проаналізувати їх відповідно до
    інтересів та здібностей студентів. Ми хочемо показати,
    як ці предмети можуть бути корисні для вашого майбутнього кар'єрного
    розвитку.
  </p>
</div>

<a name="contact"></a>
<footer>
  <div class="block-course wrap-footer">
    <div>
      <h2>Готовий до навчання
        з нами?</h2>
    </div>
    <div class="footer-form">
      <form action="./func/application.php" method="POST">
        <input type="email" placeholder="Введіть свій email"
name="email" required>
        <button>
          
        </button>
      </form>
    </div>
  </div>
</footer>

<a class="btn-top" href="#top">
  <svg xmlns="http://www.w3.org/2000/svg" width="16" height="16"
fill="currentColor" class="bi bi-arrow-up"
viewBox="0 0 16 16">
    <path fill-rule="evenodd"
      d="M8 15a.5.5 0 0 0 .5-.5V2.707l3.146 3.147a.5.5 0 0 0 .708-.708l-
4-4a.5.5 0 0 0-.708 0l-4 4a.5.5 0 1 0 .708.708L7.5 2.707V14.5a.5.5 0 0 0 .5.5z" />
    </svg>
  </a>
</body>
</html>

```