

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

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)
на тему:

**«Веб-орієнтована система освітнього процесу
університету»**

Виконав студент групи КН-19-2
КРАВЧУК Дмитро Андрійович

Керівник роботи:
КОРОТУН Ольга Володимирівна

Рецензент:
МАРЧУК Галина Вікторівна

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

Марина ГРАФ

«05» січня 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)

Здобувач вищої освіти: **КРАВЧУК Дмитро Андрійович**

Керівник роботи: **КОРОТУН Ольга Володимирівна**

Тема роботи: **«Веб-орієнтована система освітнього порталу університету»**

затверджена наказом закладу вищої освіти від **«05» січня 2023 р., №10/с**

Термін здачі закінченої роботи: **«10» червня 2023 р.**

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): об'єктом дослідження є веб-сторінка інформаційного порталу університету. Предметом дослідження є веб-система розміщення та огляду навчальної інформації, її побудова на технологічних методах та середовищ розробки.

Консультанти випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
1	КОРОТУН О.В.	05.02.2023	07.02.2023
2	КОРОТУН О.В.	12.03.2023	15.03.2023
3	КОРОТУН О.В.	08.04.2023	10.04.2023

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі. Пошук, огляд та аналіз аналогічних розробок. Формулювання технічного завдання. Опрацювання літературних джерел.	5 лютого 2023 – 10 березня 2023	Виконано
2	Проектування структури системи	12 березня 2023 – 23 березня 2023	Виконано
3	Написання програмного коду	25 березня 2023 – 22 квітня 2023	Виконано
4	Тестування системи	27 квітня 2023 – 10 травня 2023	Виконано
5	Оформлення пояснювальної записки	15 травня 2023 – 1 червня 2023	
6	Попередній захист	5 червня 2023	Виконано

Здобувач вищої освіти **Дмитро КРАВЧУК**

Керівник **Ольга КОРОТУН**

РЕФЕРАТ

Випускна робота бакалавра складається з веб-орієнтованого програмного додатку та пояснювальної записки. Пояснювальна записка до випускної роботи містить 72 сторінок, 81 ілюстрацію, 1 таблицю, 2 додатка та 25 використаних джерел.

Метою роботи є розробка веб-орієнтованого інформаційного порталу для навігації студентів по системі навчання університету.

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

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНИЙ ПОРТАЛ, ВЕБ-САЙТ, БАЗИ ДАНИХ, VISUAL STUDIO.

					КН.КР.Б – 122 – 23 – ПЗ		
Змн.	Арк.	№ докум.	Підпис	Дата	Веб-орієнтована система освітнього порталу університету Пояснювальна записка		
Розроб.		Д.А.КРАВЧУК					
Керівник		О.В.КОРОТУН					
Рецензент		Г.В.МАРЧУК					
Н. контр.							
Зав. каф.		М.С. ГРАФ			Літ. Арк. Аркуші		
						3	72
					Житомирська політехніка, група КН-19-2		

ABSTRACT

The bachelor's thesis consists of a web-based software application and an explanatory note. The explanatory note to the thesis contains 72 pages, 81 illustrations, 1 table, 2 appendices and 25 references.

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

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

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

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ	3
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	7
1 АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ.....	9
1.1 Постановка задачі	9
1.2 Аналіз аналогів	10
1.3 Вибір архітектури для створення веб-орієнтованої системи.....	14
Висновки до першого розділу.....	17
2 ПРОЕКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ОСВІТНЬОГО ПРОЦЕСУ УНІВЕРСИТЕТУ	18
2.1 Визначення варіантів використання та структури системи.....	18
2.2 Розробка структури бази даних системи.....	22
2.3 Розробка структури файлів системи.....	30
Висновки до другого розділу	45
3 ПОРЯДОК РОБОТИ ТА ТЕСТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ОСВІТНЬОГО ПРОЦЕСУ УНІВЕРСИТЕТУ	47
3.1 Архітектура веб-системи та її розгортання	47
3.2 Проектування загального інтерфейсу системи	48
3.3 Тестування роботи системи.....	53
Висновки до третього розділу.....	65
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	67
ДОДАТКИ.....	70

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

HTTP – протокол передачі гіпертексту (Hypertext Transfer Protocol)

HTML – мова розмітки гіпертексту (HyperText Markup Language)

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

PHP – мова програмування PHP (Hypertext Preprocessor)

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

VS – безкоштовний та потужний текстовий редактор, розроблений компанією Microsoft (Visual Studio)

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. В сучасному світі освітній процес стає більш доступним та зручним завдяки веб-орієнтованим системам. Це дозволяє зібрати та організувати велику кількість інформації, спілкуватись між студентами та викладачами та отримувати доступ до навчального матеріалу з будь-якого місця і в будь-який час.

Створення веб-орієнтованої системи для університету є важливим кроком у покращенні якості освіти та забезпеченні ефективного навчання. Це дозволить студентам 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 зручним. Користувачі можуть легко знайти необхідну інформацію про курси, предмети, викладачів та інші аспекти навчання. Вони можуть також залишати свої відгуки про предмети та викладачів, що дозволяє абітурієнтам отримувати більш об'єктивну інформацію та приймати рішення щодо вступу до університету.

У зв'язку з пандемією COVID-19 [15] та переходом на дистанційне навчання, веб-орієнтована система освітнього процесу університету може стати незамінним інструментом для забезпечення ефективного та організованого процесу навчання.

Отже, створення веб-орієнтованої системи для університету є важливим завданням, яке може забезпечити студентам всю необхідну інформацію в одному місці, зменшити витрати часу та покращити якість навчання. Вона може стати важливим інструментом для збільшення ефективності і організації навчального процесу та відповідати на сучасні вимоги технологічного розвитку та дистанційного навчання.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

Мета створення. Метою кваліфікаційної роботи є розробка веб-орієнтованої системи освітнього процесу університету для поліпшення якості освіти студентів, що зможе забезпечити доступ до інформації про курси, дисципліни та викладачів. Крім того, студенти зможуть ділитися 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.

Об'єктом дослідження є розробка веб-орієнтованої системи. Це дозволить забезпечити швидке та ефективне виконання завдань з будь-якого пристрою з доступом до Інтернету. Крім того, додаток буде мати дружній та інтуїтивно зрозумілий інтерфейс, що дозволить користувачам з легкістю здійснювати потрібні їм дії. Система освітнього процесу університету буде повністю автоматизована, що зменшить кількість ручної роботи та забезпечить точність та швидкість отримання потрібної інформації. Система матиме можливість зберігання та обробки великої кількості даних, що дозволить студентам та викладачам з легкістю знаходити інформацію та аналізувати результати освітнього процесу.

Предметом дослідження є розробка веб-орієнтованої системи освітнього процесу університету за допомогою використання HTML, CSS, JavaScript, PHP та середовища Visual Studio Code і MAMP. В рамках дослідження розглядаються основні принципи роботи з веб-технологіями, такі як функціонування клієнт-серверної моделі, засоби взаємодії користувача з веб-додатком, інструменти відладки та оптимізації веб-сторінок. Дослідження також передбачає пояснення різниці між мовами програмування та їхнє використання у веб-розробці.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

1 АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ

1.1 Постановка задачі

Етапи роботи

Постановка задачі полягає у створенні веб-орієнтованої системи освітнього процесу університету, яка забезпечуватиме студентам доступ до корисної інформації про дисципліни. Для досягнення цієї мети, система буде демонструвати інформацію про навчальні курси, дисципліни та викладачів, а також оцінки студентів. Важливо, щоб система мала користувацький профіль, що дозволить студентам переглядати у ньому головну інформацію про свої біографічні дані та отримані оцінки по дисциплінам. Це допоможе студентам відстежувати свій прогрес та використовувати цю інформацію для підвищення своєї продуктивності у навчанні та самодисципліні.

Крім того, система має містити відгуки про викладачів на основній формі. Це надасть студентам більше контексту до інформації про самі дисципліни, 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 зворотнього зв'язку. Це дозволить адміністратором взаємодіяти зі студентами та відправляти запрошення або рекомендації до вступу в університет.

Також, система повинна мати базу знань з корисною інформацією про курси, їх предмети та викладачів. Ці дані допоможуть студентам більш ефективно організовувати свій час та знаходити необхідну 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 покращенню процесу навчання та допоможе студентам досягнути високих результатів у своїй навчальній діяльності.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Для досягнення мети проекту, необхідно вирішити наступні задачі:

1. Розробка веб-орієнтованої системи, яка буде працювати на основі веб-технологій зі зручним і простим інтерфейсом для користувачів.
2. Збір, обробка та систематизація інформації про навчальні дисципліни, викладачів.
3. Розробка системи користувацького профілю, що дозволить користувачам переглядати свої навчальні досягнення та біографічні дані.
4. Розробка системи зворотного зв'язку для користувачів, що дозволить отримати певну інформацію від адміністраторів сайту.
5. Розробка бази знань, що містить інформацію про дисципліни, викладачів та інші корисні ресурси, що допоможуть користувачам отримати детальнішу інформацію про університет та його систему навчання.

1.2 Аналіз аналогів

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

Blackboard

Одна з найпоширеніших веб-орієнтованих систем освітнього процесу [19]. Вона містить багато функцій, включаючи користувацькі профілі, зворотній зв'язок, електронні засоби навчання та багато іншого.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

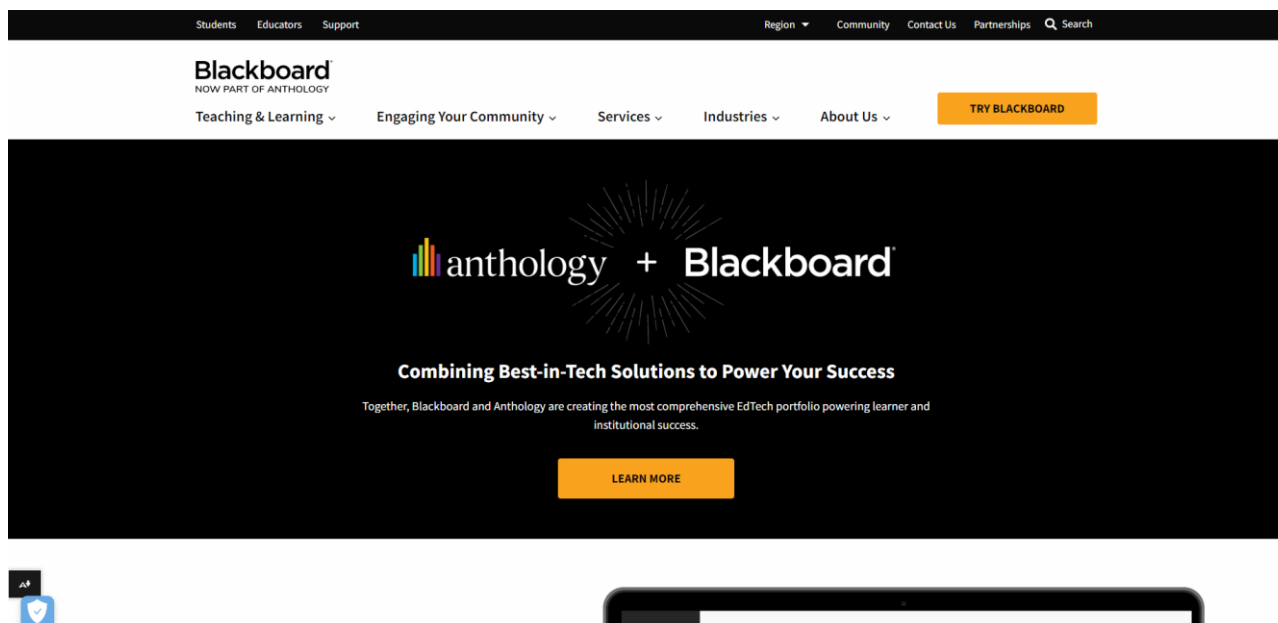


Рис.1.1. Головна сторінка Blackboard

Moodle

Ця система також широко використовується в школах та університетах [20]. 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.

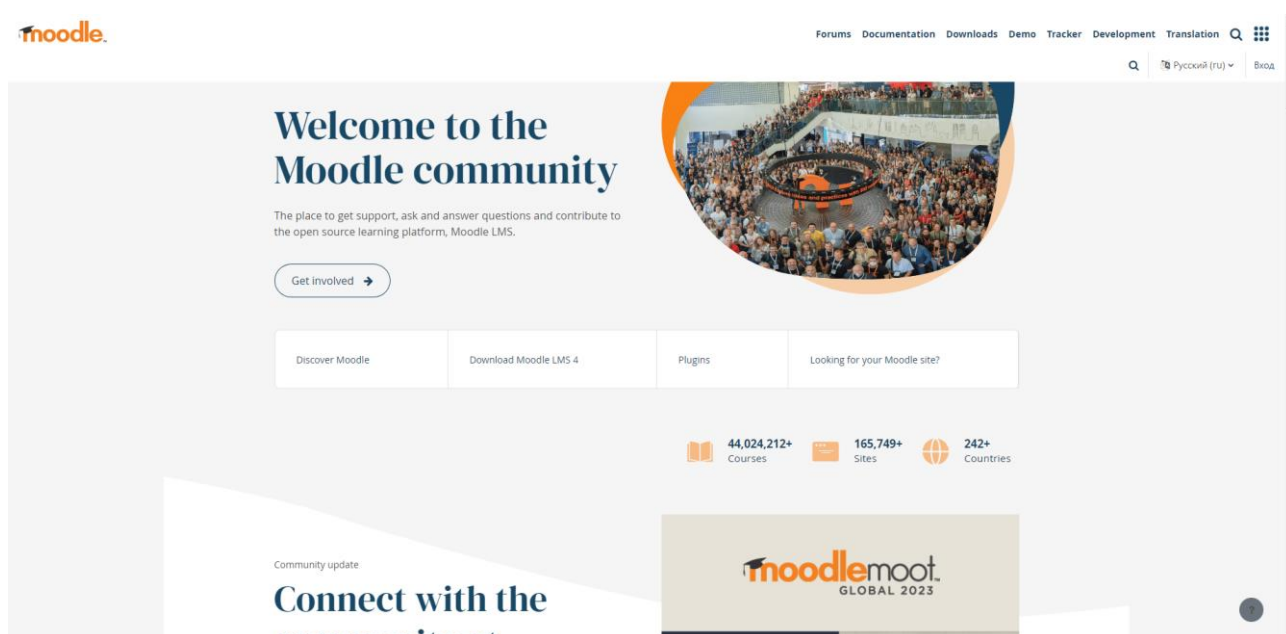


Рис.1.2. Головна сторінка Moodle

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Canvas

Веб-орієнтована система освітнього процесу використовується багатьма школами та університетами [21]. Вона має користувацькі профілі, зворотній зв'язок та електронні засоби навчання. Однак, вона має менше можливостей для індивідуалізованого навчання, порівняно з Moodle.



Рис.1.3. Головна сторінка Canvas

Schoology

Веб-орієнтована система, яка має багато функцій, включаючи зворотній зв'язок, електронні засоби навчання та інструменти для співпраці [22]. Вона також має можливості для індивідуалізованого навчання та підтримки студентів.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

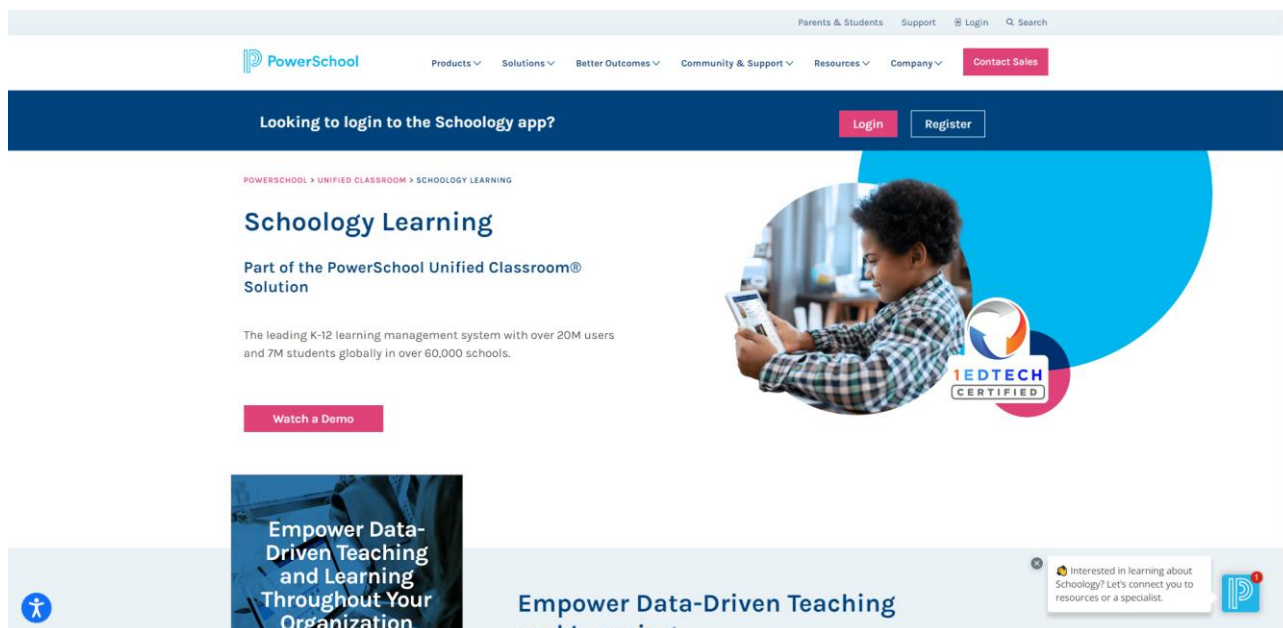


Рис.1.4. Головна сторінка Schoology

Google Classroom

Це безкоштовна веб-орієнтована система, яка створена для використання в освітніх закладах [23]. 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.

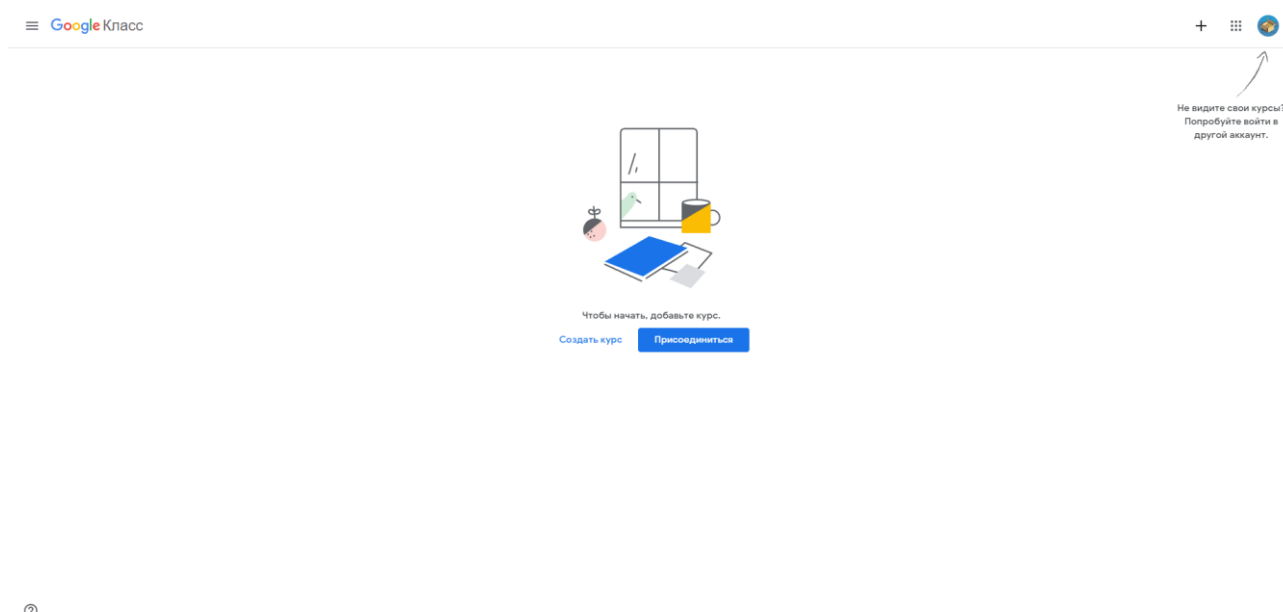


Рис.1.5. Головна сторінка Google Classroom

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

1.3 Вибір архітектури для створення веб-орієнтованої системи

Для створення веб-орієнтованої системи освітнього процесу університету, потрібно детальніше розглянути такі технології.

Frontend: HTML [26][27], CSS [9][29], та JavaScript [34] для створення користувацького інтерфейсу та взаємодії з користувачами. Для покращення користувацького досвіду можемо додати анімації, які зроблять сайт більш захоплюючим. Крім того, додамо більше функцій, які зроблять інтерфейс більш зручним для користувачів, наприклад, можливість використовувати вкладені сторінки.

Backend: PHP для обробки запитів користувачів та взаємодії з базою даних [11][31]. Однак, додамо більше функцій для забезпечення високої продуктивності, наприклад, кешування запитів, що дозволить швидко відповідати на запити користувачів тощо.

Архітектура проекту буде включати наступні компоненти.

Користувацький інтерфейс: можна створити за допомогою HTML [28], CSS [30] та JavaScript. Користувачі зможуть зареєструватися, увійти в систему, переглянути свій профіль, користуватися формою зворотнього зв'язку тощо. А також додамо більше функцій, які зроблять інтерфейс зручним для користувачів, наприклад, можливість отримувати розсилку від адміністраторів в особистому кабінеті тощо.

База даних: створимо за допомогою MySQL [3][32]. База даних міститиме таблиці для користувачів, рейтингу користувачів, навчальних дисциплін, розсилки повідомлень та електронних пошт відвідувачів. Розробимо більше таблиць, щоб зберігати додаткову інформацію. Також можна додати механізми резервного копіювання та відновлення бази даних, щоб забезпечити захист від втрати даних.

Бекенд: реалізуємо за допомогою PHP [33]. Бекенд буде відповідати за обробку запитів користувачів та взаємодію з базою даних. Крім того, буде

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

відповідати за обробку форми зворотнього зв'язку та оновлення даних про користувачів, курси, дисципліни та викладачів.

Таблиця 1.1 – Таблиця порівнянь 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 забезпечення всіх необхідних функцій, потрібно реалізувати наступні модулі.

Модуль реєстрації та авторизації користувачів

Реалізуємо зручні та безпечні форми реєстрації та 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

Додамо можливість перегляду особистого кабінету користувача, де будуть відображатись всі особисті дані, успіхи в навчанні, повідомлення від адміністраторів та функція переходу до потрібного курсу.

Модуль додавання, редагування та видалення навчальних курсів

Дозволяє створювати курси по певному шаблону, реалізувати функції редагування уже створених курсів та їх видалення.

Модуль присвоєння та редагування рейтингу по дисципліні для певного користувача

Створимо окрему форму для адміністратора, де він зможе виставити оцінки з окремих дисциплін для певних користувачів. Також треба реалізувати можливість редагування оцінки в будь-який час.

Модуль керування поточними користувачами

Розробимо форму для адміністратора, де він буде мати змогу переглядати список всіх зареєстрованих студентів, та при потребі, наприклад коли студент був відчислений, видаляти його аккаунт з бази даних сайту.

Модуль форми зворотнього зв'язку, який дозволить користувачам залишати свої контакти

Додамо форму, яка буде давати можливість відвідувачам та користувачам залишати свою електронну пошту для очікування на зворотній зв'язок.

Проект будемо розробляти у середовищі розробки **Visual Studio** [14]. 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 синтаксису, підсвічування синтаксису, підказки та багато іншого.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

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

Крім того, VS є кросплатформовим, що дозволяє розробникам використовувати VS незалежно від своєї 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 швидко та легко розробляти веб-орієнтовані системи та співпрацювати з іншими розробниками. Зокрема, завдяки великій кількості розширень та можливості інтеграції з Git [7], розробка проекту в VS може бути доволі продуктивною та ефективною.

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

ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ

В першому розділі звіту кваліфікаційної роботи розглянуті важливі етапи розробки веб-сайту. Поставлена задача, яка визначила напрямок дослідження та розробки.

Проведено аналіз аналогів, що дозволило ознайомитись зі схожими проектами, визначити їх переваги, а також виявити недоліки, які необхідно уникнути в розробці власного веб-сайту.

Останнім етапом зроблено вибір та аналіз архітектури для створення веб-сайту. Визначили оптимальне середовище розробки.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЕКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ОСВІТНЬОГО ПРОЦЕСУ УНІВЕРСИТЕТУ

2.1 Визначення варіантів використання та структури системи

Веб-орієнтована система освітнього процесу університету створюється з метою забезпечення зручного доступу до інформації для відвідувачів та студентів.

Основні доступні функції для користувачів.

Реєстрація

Користувачі можуть створити обліковий запис на порталі, заповнивши необхідні дані у формі реєстрації.

Після успішної реєстрації користувач має можливість авторизуватись на сайті.

Авторизація

Користувачі можуть увійти на портал, використовуючи свої облікові дані у формі авторизації.

Після успішної авторизації користувач отримує доступ до особистого кабінету та інших функцій portalу.

Перегляд довідки про проект

Користувачі можуть переглянути загальну інформацію про наш проект на головній сторінці portalу.

Вони можуть ознайомитись з метою, завданнями та іншими деталями проекту.

Перегляд навчальних курсів

Користувачі можуть переглядати список доступних курсів навчання на порталі.

Вони мають можливість обрати конкретний курс та переглянути список навчальних предметів, які включені до нього.

Перегляд інформації про навчальні дисципліни

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

Користувачі можуть переглянути загальну інформацію про конкретну навчальну дисципліну, включаючи опис, викладача та відгуки студентів.

Перегляд особистих даних та оцінок

Користувачі мають доступ до особистого кабінету, де можуть переглянути свої особисті дані, такі як ім'я та прізвище, електронна пошта тощо.

Вони також можуть переглядати свої оцінки з навчальних предметів, які були виставлені адміністратором.

Перегляд розсилки та повідомлень

Користувачі можуть переглядати розсилку та отримувати повідомлення від адміністратора.

Перехід до курсу

Користувачі можуть перейти з особистого кабінету безпосередньо до потрібного курсу навчання.

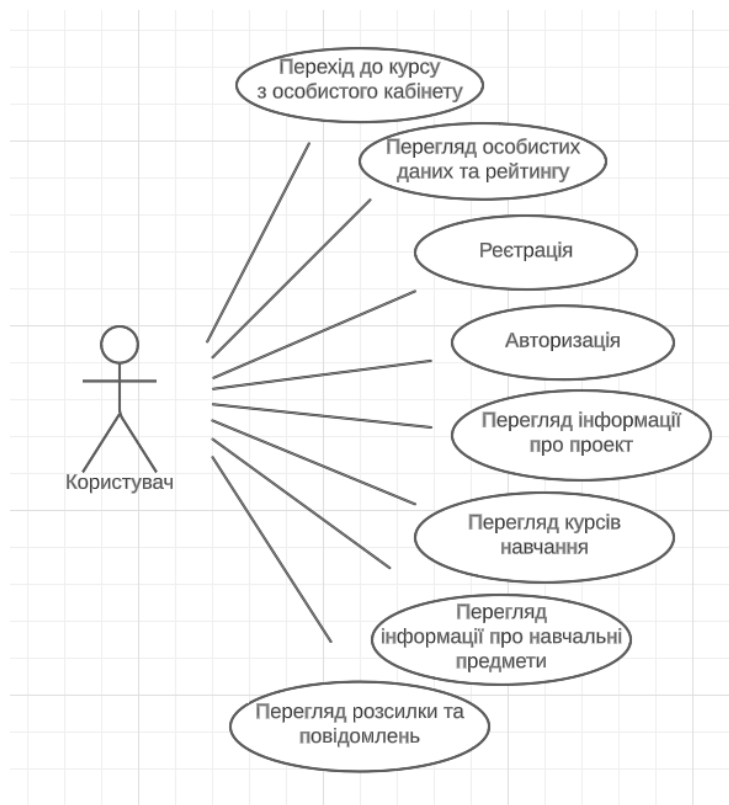


Рис.2.1. Варіанти використання порталу користувачем

Також наш веб-орієнтована система повинна мати широкий функціонал для адміністратора сайту, щоб можна було слідкувати за користувачами, додавати та оновлювати інформацію на сторінках сайту.

Основні доступні функції для адміністратора:

Управління користувачами

Можливість видалення облікових записів користувачів.

Управління навчальними дисциплінами

Форма додавання нових навчальних дисциплін до системи та модулі редагування інформації в уже створених шаблонах. Також можливість видалення навчальних дисциплін з системи.

Управління оцінками

Виставлення оцінок студентам з навчальних дисциплін в їх особисті кабінети та можливість видалення адміністратором.

Розсилка повідомлень

Надсилання повідомлень користувачам в їх особисті кабінети та можливість видалення адміністратором.

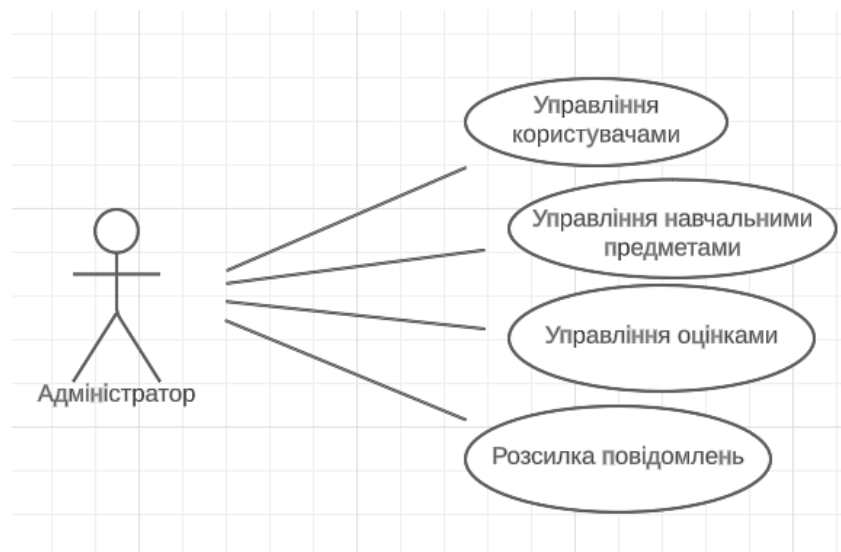


Рис.2.2. Варіанти використання порталу адміністратором

Функціональні вимоги до веб-сайту.

Авторизація та реєстрація

- Користувачі можуть створювати облікові записи та авторизуватись на веб-сайті.
- Різні рівні доступу, зокрема для звичайних користувачів та адміністратора.

Головна сторінка

- Відображення інформації про проект та його мету.

- Форма зворотного зв'язку для користувачів.

Навчальні курси та дисципліни

- Перегляд списку та опис доступних курсів.
- Відображення інформації про кожну навчальну дисципліну.
- Перехід на сторінку певної дисципліни з детальною інформацією, викладачем та відгуками студентів.

Особистий кабінет користувача

- Перегляд особистих даних, зазначених під час реєстрації.
- Перегляд оцінок з навчальних дисциплін.
- Отримання повідомлень та розсилок від адміністратора.
- Можливість переходу до потрібного курсу.

Управління навчальними курсами (адміністратор)

- Додавання, редагування та видалення навчальних курсів.
- Встановлення оцінок з певних дисциплін для користувачів.

Нефункціональні вимоги до веб-сайту.

Час навчання

- Час, необхідний для ознайомлення з порталом для звичайних користувачів - не більше 15 хвилин, для досвідчених - не більше 5 хвилин.

Швидкість реакції системи

- Час відповіді системи на запити користувачів - не більше 0.017 секунди для звичайних запитів, не більше 0.05 секунди для складніших запитів.

Інтуїтивність інтерфейсу

- Інтерфейс повинен бути зрозумілим та простим у використанні навіть для користувачів без попереднього досвіду використання подібних систем.

Безпека

- Забезпечення конфіденційності особистої інформації користувачів.
- Захист системи від несанкціонованого доступу та зловживання.

Сумісність

- Сумісність системи з різними веб-браузерами (Chrome, Firefox, Safari, etc.).

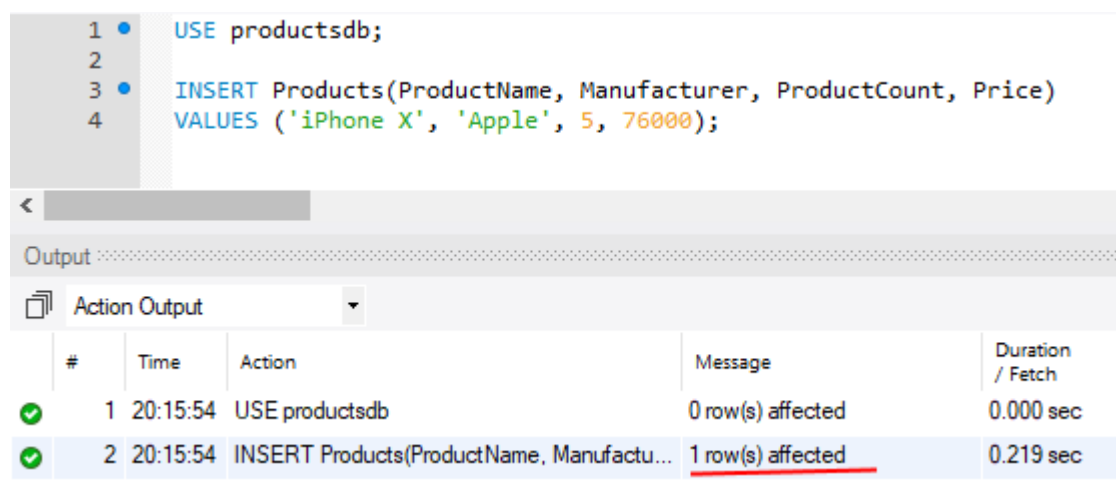
					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

Ці вимоги відображають основну функціональність та характеристики, які має мати проект для відповідності потребам користувачів та поставленій меті.

2.2 Розробка структури бази даних системи

У веб-орієнтованій системі освітнього процесу університету потрібно організувати надійну систему зберігання та читання великого обсягу даних. Для цього розглянемо MySQL як одну з найпопулярніших систем управління базами даних та порівняємо її з конкурентами.

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



#	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, Manufacturer, ProductCount, Price) VALUES ('iPhone X', 'Apple', 5, 76000);	1 row(s) affected	0.219 sec

Рис.2.3. Приклад бази даних MySQL

Основні переваги MySQL.

Простота використання

MySQL має зрозумілу і просту структуру, що дозволяє легко створювати таблиці, виконувати запити та керувати базою даних.

Швидкодія

MySQL відомий своєю високою швидкодією та продуктивністю, добре працює з великими обсягами даних та забезпечує ефективну обробку запитів.

Масштабованість

MySQL підтримує горизонтальне та вертикальне масштабування, що дозволяє розширювати систему для роботи з більшим обсягом даних або забезпечити більшу кількість одночасних з'єднань.

Надійність та стабільність

MySQL є стабільною та надійною системою, яка добре впорається з обробкою великого навантаження та має механізми відновлення після збоїв.

Але незважаючи на багато переваг, MySQL також має свої недоліки.

Обмеження в функціональності

Порівняно з деякими конкурентами, MySQL може бути менш функціональним в плані підтримки складних операцій або типів даних.

Складність розгортання

Хоча сама система MySQL нескладна у використанні, налаштування та розгортання бази даних можуть бути складними завданнями, особливо для початківців.

Використання ресурсів

Залежно від налаштувань та обсягу даних, MySQL може вимагати значних ресурсів, таких як пам'ять та процесорний час.

MySQL має кілька сильних конкурентів на ринку систем управління базами даних. Розглянемо деякі з них.

PostgreSQL

PostgreSQL [16] є потужною відкритою реляційною СУБД, яка підтримує багато функцій та розширень. В порівнянні з MySQL, PostgreSQL має більш розширений набір можливостей, але може бути складнішим у використанні.



Рис.2.4. Система управління базами даних PostgreSQL

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

Oracle

Oracle [17] 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

Microsoft SQL Server

SQL Server [18] 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

MAMP є середовищем для розробки веб-додатків, яка поєднує в собі сервер Apache, базу даних MySQL та мову програмування PHP [2]. Саме його ми будемо використовувати в нашому проекті.

Особливості MAMP включають в себе [13]:

Простота встановлення

MAMP надає простий спосіб встановлення та налаштування серверного середовища на комп'ютері.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Локальна розробка

МAMP дозволяє розробляти та тестувати веб-додатки локально на комп'ютері перед розгортанням на живий сервер.

Інтегровані компоненти

МAMP поєднує сервер Apache, базу даних MySQL та мову програмування PHP, що дозволяє легко розробляти веб-додатки, які використовують ці технології.

Враховуючи всі фактори за та проти, вибрали оптимальне СУБД та середовище розробки, де будуть зберігатись складові нашого проекту. На рис. 2.7 представлено перелік таблиць бази даних веб-орієнтованої системи освітнього процесу університету у MySQL.

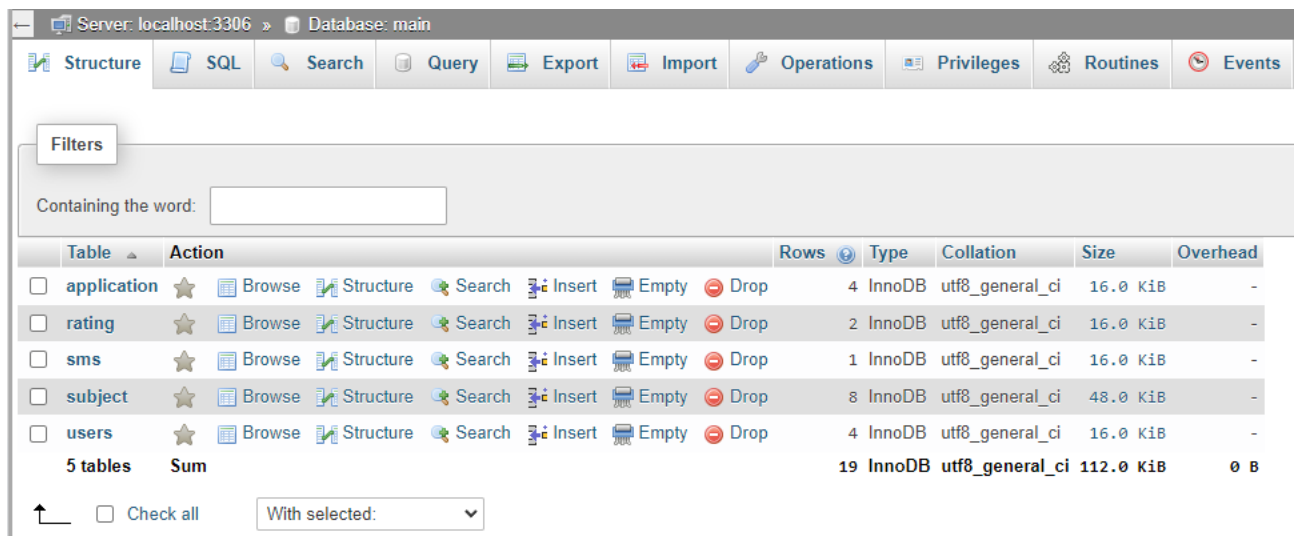


Table	Action	Rows	Type	Collation	Size	Overhead
☐ application	★ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16.0 KiB	-
☐ rating	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8_general_ci	16.0 KiB	-
☐ sms	★ Browse Structure Search Insert Empty Drop	1	InnoDB	utf8_general_ci	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

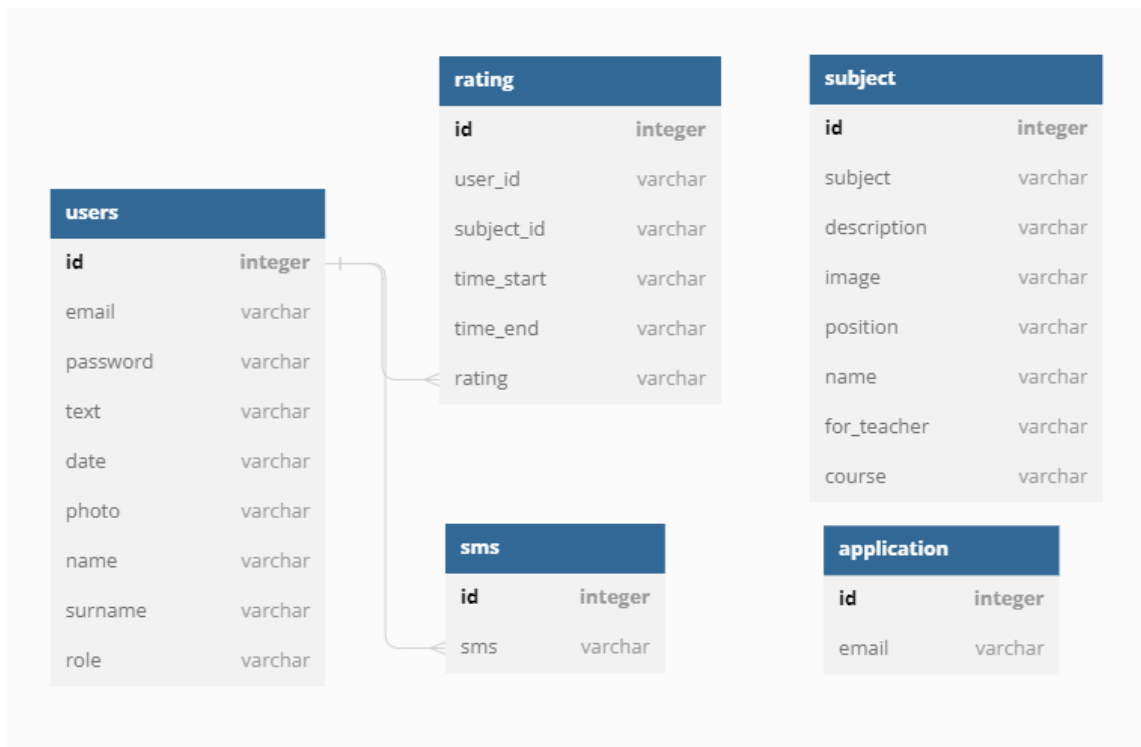


Рис.2.8. Діаграма бази даних веб-орієнтованої системи освітнього процесу університету

Отже, створена база даних з назвою main, що складається з п'яти таблиць: "users", "rating", "sms", "subject" і "application". Кожна таблиця має свою структуру та поля для зберігання відповідних даних.

У загальному, база даних містить інформацію, необхідну для функціонування нашого інформаційного порталу. Вона забезпечує зберігання та організацію даних про користувачів, рейтинги, повідомлення, предмети та заявки. Кожна таблиця має відповідні поля для зберігання необхідних атрибутів і взаємозв'язки з іншими таблицями за допомогою зовнішніх ключів.

Ознайомимось детальніше з кожною з таблиць.

Таблиця "users" містить дані про користувачів, такі як їхні ідентифікатори, електронні пошти, паролі, тексти, дати, фотографії, імена, прізвища та ролі.

Таблиця "users"

id (цілочисельний): унікальний ідентифікатор користувача.

email (рядок): електронна адреса користувача.

password (рядок): пароль користувача.

text (рядок): текстове поле.

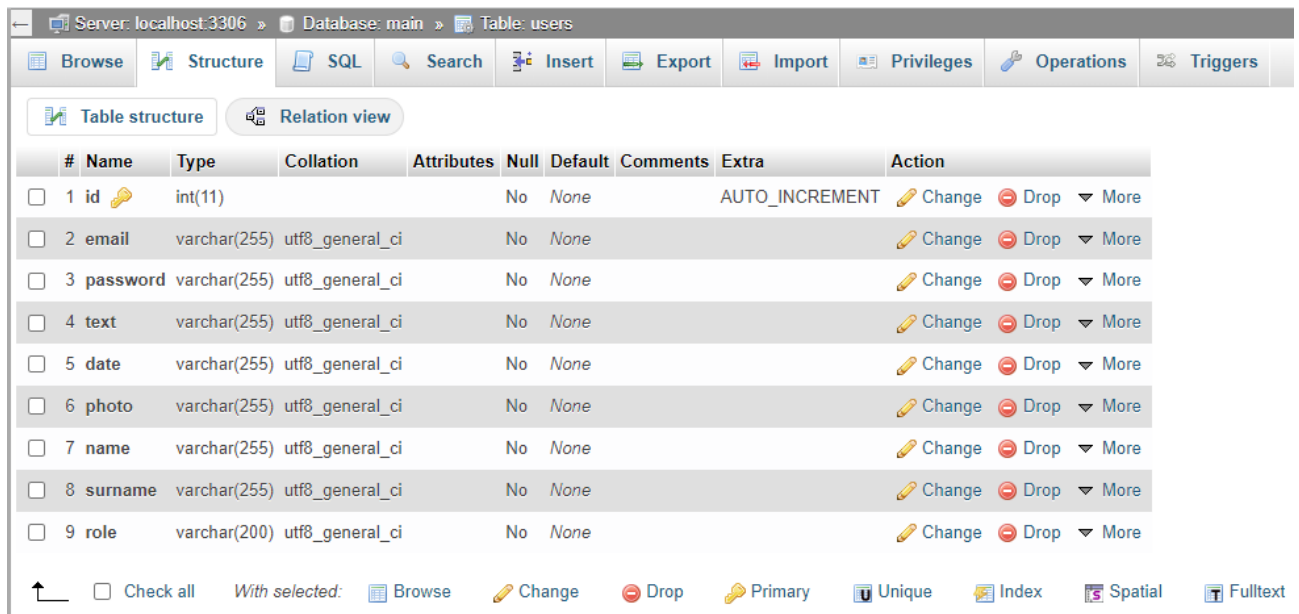
date (рядок): дата народження.

photo (рядок): шлях до фотографії користувача.

name (рядок): ім'я користувача.

surname (рядок): прізвище користувача.

role (рядок): роль користувача.



#	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

Рис.2.9. Таблиця users

Таблиця "rating" зберігає дані про оцінки, пов'язані з користувачами та предметами. Вона містить ідентифікатори рейтингів, ідентифікатори користувачів, ідентифікатори предметів, час початку і закінчення навчання та самі оцінки.

Таблиця "rating"

id (цілочисельний): унікальний ідентифікатор рейтингу.

user_id (рядок): ідентифікатор користувача, пов'язаний з таблицею "users".

subject_id (рядок): ідентифікатор предмета, пов'язаний з таблицею "subject".

time_start (рядок): початок навчання.

time_end (рядок): кінець навчання.

rating (рядок): оцінка.

#	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

Рис.2.10. Таблица rating

Таблица "sms" використовується для зберігання повідомлень, відправлених адміністратором. Вона містить ідентифікатори повідомлень та ідентифікатори користувачів.

Таблица "sms"

id (цілочисельний): унікальний ідентифікатор повідомлення.

sms (рядок): текст повідомлення.

user_id (рядок): ідентифікатор користувача, пов'язаний з таблицею "users".

#	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. Таблица sms

Таблица "subject" містить дані про навчальні предмети, такі як назви, описи, фото викладачів, посади, ПІБ, загальна інформація про викладачів, та курс до якого відноситься предмет.

Таблица "subject"

id (цілочисельний): унікальний ідентифікатор предмета.

subject (рядок): назва предмета.

description (рядок): опис предмета.

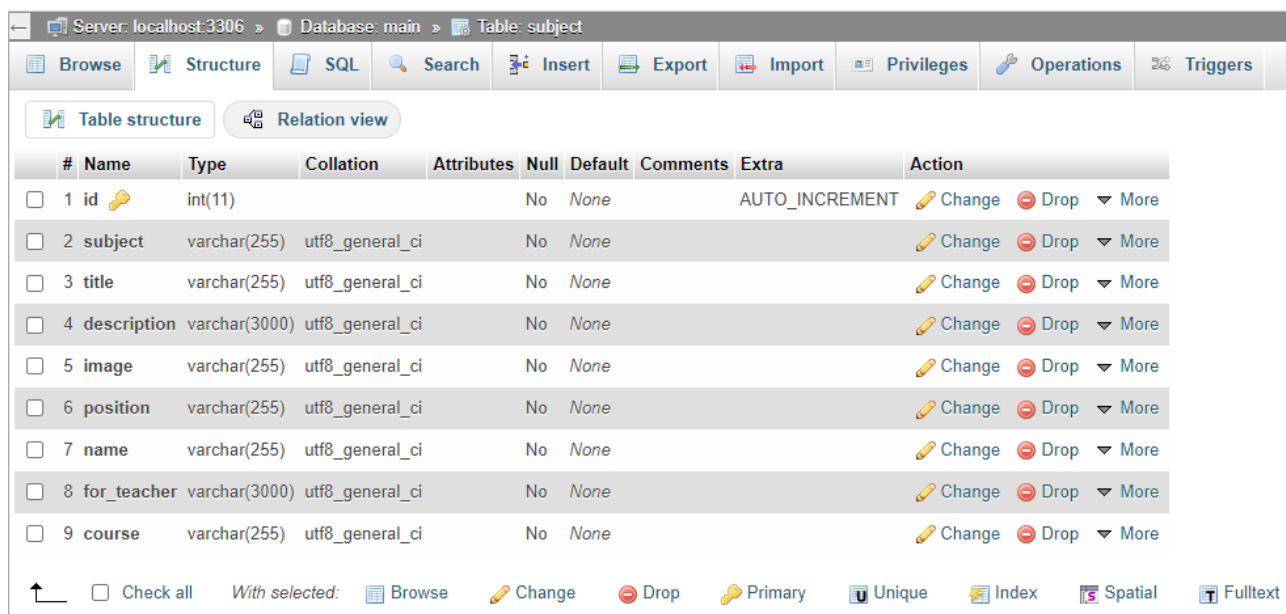
image (рядок): шлях до фото викладача.

position (рядок): посада викладача.

name (рядок): ПІБ викладача.

for_teacher (рядок): загальна інформація про викладача.

course (рядок): курс, до якого відноситься предмет.



The screenshot shows the MySQL Table structure for the 'subject' table. The table has 9 columns: id (int(11), primary key, AUTO_INCREMENT), subject (varchar(255)), title (varchar(255)), description (varchar(3000)), image (varchar(255)), position (varchar(255)), name (varchar(255)), for_teacher (varchar(3000)), and course (varchar(255)). All columns are of type utf8_general_ci and have no default values or comments.

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

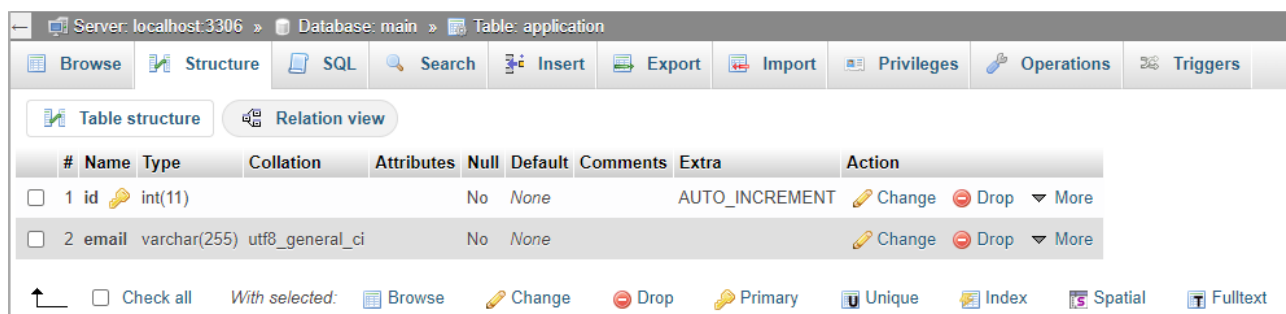
Рис.2.12. Таблиця subject

Таблиця "application" використовується для зберігання заявок, які містять електронні адреси.

Таблиця "application"

id (цілочисельний): унікальний ідентифікатор заявки.

email (рядок): електронна пошта користувача.



The screenshot shows the MySQL Table structure for the 'application' table. The table has 2 columns: id (int(11), primary key, AUTO_INCREMENT) and email (varchar(255)). Both columns are of type utf8_general_ci and have no default values or comments.

#	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

Рис.2.13. Таблиця application

2.3 Розробка структури файлів системи

Опишемо організацію файлів у розробленому веб-сайті. Наведемо основні файли та їх реалізовані функції, які будуть розглянуті досить детально.

Загальна структура файлів проекту визначає організацію та розміщення файлів. Наведемо опис існуючих папок з файлами, які використовуються в проєкті:

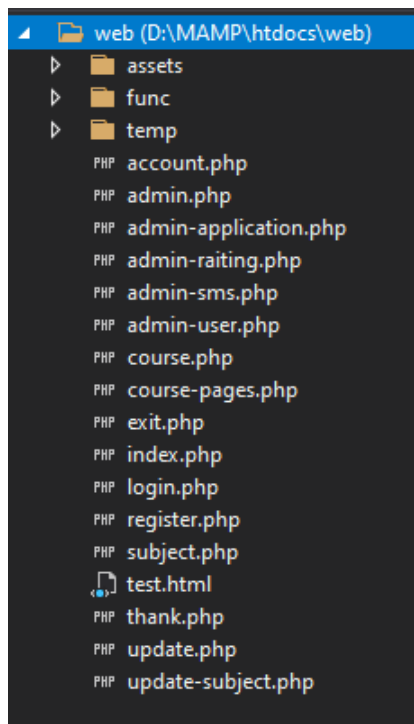


Рис.2.14. Загальна структура файлів проєкту

Папка "assets" є однією з основних папок для зберігання ресурсів, таких як зображення, шрифти, скрипти та стилі, які використовуються для оформлення нашого порталу. Ця папка зазвичай використовується для централізованого управління всіма статичними файлами, які потрібні для відображення та функціонування нашого веб-додатку.

Використання папки "assets" має кілька переваг.

Організація ресурсів

Розділення різних типів ресурсів у відповідних підпапках дозволяє легко знайти та керувати файлами. Наприклад, можна швидко знайти всі зображення в папці "img" або всі скрипти в папці "script".

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Підтримка розширюваності

Використання окремої папки для ресурсів уникне перекривання файлів між різними частинами розробленого веб-сайту. Це дозволить легко додавати нові ресурси та зберігати їх відокремлено від інших файлів.

Швидкий доступ

Розташування ресурсів у централізованій папці дозволяє легко посилатися на них у коді. Можна використовувати відносні посилання на ресурси, починаючи з папки "assets", що спрощує пошук та зміну посилань при необхідності.

Описана структура папки "assets" з підпапками "fonts", "img", "script" та "style" дозволяє логічно організувати ресурси проекту та забезпечити легку навігацію та управління ними під час розробки та підтримки веб-орієнтованого інформаційного порталу.

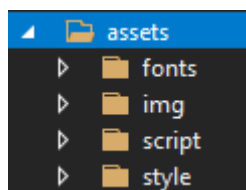


Рис.2.15. Структура папки assets

Пройдемося більш детально по внутрішнім папкам assets.

Папка "fonts"

У цій папці зберігаються шрифти. Можна додати файли шрифтів різних форматів, таких як .ttf або .woff, щоб забезпечити сумісність з різними браузерами та пристроями.

Папка "img"

Тут зазвичай знаходяться зображення, які використовуються на порталі. Можна створити підпапки для різних категорій зображень, наприклад, "avatars" для профільних фотографій користувачів або "banners" для рекламних банерів.

Папка "script"

В даній директорії можна зберігати файли скриптів JavaScript [35], які використовуються в проєкті. Це можуть бути власноруч написані скрипти або залежності, такі як бібліотеки або фреймворки JS.

Папка "style"

В цій папці зазвичай знаходяться файли стилів CSS, які керують зовнішнім виглядом системи. Можна створити окремі файли для різних сторінок або компонентів, а також створити підпапки для організації різних стилів.

Ці папки допоможуть нам керувати ресурсами проєкту, такими як шрифти, зображення, скрипти та стилі. Вони також сприяють організованості та зручності під час розробки та підтримки веб-орієнтованої системи.



Рис.2.16. Папка func

Папка "func" містить набір файлів PHP, які відповідають за різні функції та операції інформаційного порталу. Кожен файл в цій папці має свою відповідну функціональну роль в системі. Опишемо кожен файл окремо:

add-subject.php

Цей файл відповідає за додавання нової дисципліни до бази даних та містить код, який обробляє дані, отримані з форми додавання предмета і вставляє їх в таблицю "subject" розробленої бази даних.

```
add-subject.php
1 <?php
2
3 if ($FILES['image']['error'] === UPLOAD_ERR_OK) {
4     $tmp_name = $FILES['image']['tmp_name'];
5     $name = $FILES['image']['name'];
6     $destination = '../assets/img/teacher/' . $name;
7
8     if (move_uploaded_file($tmp_name, $destination)) {
9         echo 'Картинка успішно завантажена!';
10        echo '<br>';
11        echo '';
12    } else {
13        echo 'Помилка при завантаженні картинки.';
14    }
15 } else {
16     echo 'Виникла помилка при завантаженні файлу.';
17 }
18
19
20 require_once 'db.php';
21
22 $subject = addslashes($_POST['subject']);
23 $title = addslashes($_POST['title']);
24 $description = addslashes($_POST['description']);
25 $image = $name;
26 $position = addslashes($_POST['position']);
27 $name = addslashes($_POST['name']);
28 $for_teacher = addslashes($_POST['for_teacher']);
29 $course = $_POST['course'];
30
31 mysql_query($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')");
32
33
34 header('Location: ../admin.php')
35 }
```

Рис.2.17. Структура файлу add-subject.php

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

application.php

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

```
application.php  + X
1  <?php
2
3
4  require_once 'db.php';
5
6  $email = $_POST['email'];
7
8
9
10 mysqli_query($connect, "INSERT INTO `application` (`id`, `email`) VALUES (NULL, '$email ');");
11
12
13 header('Location: ../thank.php')
14 ?>
```

Рис.2.18. Структура файлу application.php

db.php

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

```
db.php  + X
1  <?php
2  $connect = mysqli_connect('localhost', 'root', 'root', 'main');
3
4  if(!$connect){
5      echo 'Errors!';
6  }
7  ?>
```

Рис.2.19. Структура файлу db.php

delete-rating.php

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

```

delete-rating.php
1  <?php
2
3  require_once 'db.php';
4
5  $id = $_GET['id'];
6
7
8  mysqli_query($connect, "DELETE FROM `rating` WHERE `rating`.`id` = '$id' ");
9
10 header('Location: ../admin-raiting.php');
11
12 ?>

```

Рис.2.20. Структура файлу delete-rating.php

delete-sms.php

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

```

delete-sms.php
1  <?php
2
3  require_once 'db.php';
4
5  $id = $_GET['id'];
6
7
8  mysqli_query($connect, "DELETE FROM `sms` WHERE `sms`.`id` = '$id' ");
9
10 header('Location: ../admin-sms.php');
11
12 ?>

```

Рис.2.21. Структура файлу delete-sms.php

delete-subject.php

Цей модуль відповідає за видалення предметів з бази даних. Складається з функцій, які виконують запити до бази даних для видалення конкретного предмета з таблиці "subject".

```

delete-subject.php
1  <?php
2
3  require_once 'db.php';
4
5  $id = $_GET['id'];
6
7
8  mysqli_query($connect, "DELETE FROM `subject` WHERE `subject`.`id` = '$id' ");
9
10 header('Location: ../admin.php');
11
12 ?>

```

Рис.2.22. Структура файлу delete-subject.php

delete-user.php

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

```
delete-user.php  ▢ ×
1  <?php
2
3  require_once 'db.php';
4
5  $id = $_GET['id'];
6
7
8  mysqli_query($connect, "DELETE FROM `users` WHERE `users`.`id` = '$id' ");
9
10 header('Location: ../admin-user.php');
11
12 ?>
```

Рис.2.23. Структура файлу delete-user.php

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.

```
login.php*  ▢ ×
1  <?php
2  session_start();
3
4
5  require_once 'db.php';
6
7
8  $login = $_POST['email'];
9  $password = $_POST['password'];
10
11
12
13 $res = mysqli_query($connect, "SELECT * FROM users WHERE email = '$login' and password = '$password' ");
14
15
16
17 $result = mysqli_fetch_assoc($res);
18
19 $id = $result['id'];
20 $role = $result['role'];
21
22 $count = mysqli_num_rows($res);
23
24
25 if ($count == 1) {
26     if ($role == 1) {
27         $_SESSION['id'] = $id;
28         header('Location: ../account.php');
29     } else if ($role == 2) {
30         $_SESSION['id'] = $id;
31         header('Location: ../admin.php');
32     }
33 }
34 else {
35     echo "Wrong login or password!";
36 }
37 ?>
```

Рис.2.24. Структура файлу login.php

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

rating.php

Файл відповідає за отримання та відображення оцінок користувача. Вміщує функції для виконання запитів до бази даних для отримання оцінок користувача з таблиці "rating" і їх подальший вивід на веб-сторінку.

```
rating.php  ✕
1  <?php
2
3
4  require_once 'db.php';
5
6  $user_id = $_POST['user_id'];
7  $subject_id = $_POST['subject_id'];
8  $time_start = $_POST['time_start'];
9  $time_end = $_POST['time_end'];
10 $rating = $_POST['rating'];
11
12
13
14
15 mysqli_query($connect, "INSERT INTO `rating` (`id`, `user_id`, `subject_id`,
16
17
18 header('Location: ../admin-rating.php')
19 ?>
```

Рис.2.25. Структура файлу rating.php

register.php

Цей файл відповідає за обробку процесу реєстрації нового користувача. Складається з функцій, що перевіряють введені дані користувача на унікальність і створюють новий запис користувача в таблиці "users" тим самим забезпечують його реєстрацію у системі.

```
register.php  ✕
1  <?php
2
3  if ($FILES['image']['error'] == UPLOAD_ERR_OK) {
4      $tmp_name = $FILES['image']['tmp_name'];
5      $name = $FILES['image']['name'];
6      $destination = '../assets/img/user/' . $name;
7
8      if (move_uploaded_file($tmp_name, $destination)) {
9          echo 'Картинка успішно завантажена!';
10         echo '<br>';
11         echo '<img src="" . $destination . "" alt="Завантажена картинка">';
12     } else {
13         echo 'Помилка при завантаженні картинки.';
14     }
15 } else {
16     echo 'Виникла помилка при завантаженні файлу.';
17 }
18
19
20 require_once 'db.php';
21
22 $email = $_POST['email'];
23 $password = $_POST['password'];
24 $number_book = $_POST['number_book'];
25 $date = $_POST['date'];
26 $photo = $name;
27 $name = $_POST['name'];
28 $surname = $_POST['surname'];
29
30
31 mysqli_query($connect, "INSERT INTO `users` (`id`, `email`, `password`, `text`, `date`, `photo`, `name`, `surname`, `role`) VALUES (NULL, '$email', '$password', '$number_book', '$date',
32
33
34 header('Location: ../login.php')
35 ?>
```

Рис.2.26. Структура файлу register.php

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

sms.php

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

```
1 <?php
2
3
4 require_once 'db.php';
5
6 $sms = $_POST['sms'];
7
8
9
10 mysqli_query($connect, "INSERT INTO `sms` (`id`, `sms`) VALUES (NULL, '$sms ');");
11
12
13 header('Location: ../admin-sms.php')
14 ?>
```

Рис.2.27. Структура файлу sms.php

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".

```
1 <?php
2 require_once 'db.php';
3
4 $id = $_POST['id'];
5 $rating = $_POST['rating'];
6
7 mysqli_query($connect, "UPDATE `rating` SET `rating` = '$rating' WHERE `rating`.`id` = '$id';");
8
9
10 header('Location: ../admin-rating.php');
11 ?>
```

Рис.2.28. Структура файлу update-rating.php

update-sub.php

Цей файл відповідає за оновлення інформації про предмет у базі даних та складається з коду, який виконує запити до бази даних для оновлення даних про конкретний предмет в таблиці "subject".

```

1  <?php
2  require_once 'db.php';
3
4  $id = $_POST['id'];
5  $subject = addslashes($_POST['subject']);
6  $title = addslashes($_POST['title']);
7  $description = addslashes($_POST['description']);
8  $position = addslashes($_POST['position']);
9  $name = addslashes($_POST['name']);
10 $for_teacher = addslashes($_POST['for_teacher']);
11
12
13
14
15 mysqli_query($connect, "UPDATE `subject` SET `subject` = '$subject', `title` = '$title', `description` = '$description', `position` = '$position', `name` = '$name', `for_teacher` = '$for_teach
16
17
18 header('Location: ../admin.php');
19 ?>

```

Рис.2.29. Структура файлу update-sub.php

Кожен з цих файлів відповідає за певну функціональність проекту, а їх узгоджена робота дозволяє забезпечити відповідну функціональність та взаємодію всіх компонентів веб-орієнтованої системи освітнього процесу університету.

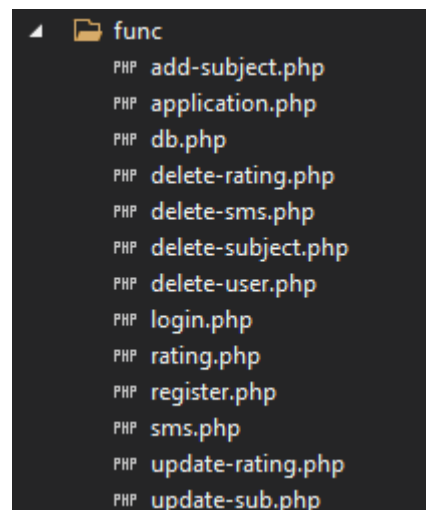


Рис.2.30. Структура папки func

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

header.php

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

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

header.php  X
1  <?php
2  session_start();
3
4  require_once './func/db.php';
5
6  $user_id = $_SESSION['id'];
7  $new = mysqli_query($connect, "SELECT * FROM `users` WHERE `id` = '$user_id'");
8  $new = mysqli_fetch_assoc($new);
9  ?>
10 <header>
11   <div class="wrap-header">
12     <div class="header__logo">
13       <a href="./index.php">
14         
15       </a>
16     </div>
17     <div class="header__menu">
18       <span><a href="./index.php">Головна</a></span>
19       <span><a href="./course.php">Перелік курсів</a></span>
20       <span><a href="./index.php#dovidka">Довідка</a></span>
21       <span><a href="./index.php#contact">Контакти</a></span>
22     </div>
23     <?php
24     if($new['role'] == 2){
25       echo ' <span><a href="./admin.php">Адмінка</a></span>';
26     }else{
27       echo ' <span><a href="./account.php"></a></span>';
28     }
29     if (isset($_SESSION['id'])) {
30       echo ' <span><a href="./exit.php">Вийти</a></span>';
31     }
32     ?>
33   </div>
34 </div>
35 </header>

```

Рис.2.31. Структура файлу header.php

admin-header.php

Цей файл містить код, що відображає заголовок сторінки для адміністратора та має подібний вигляд до "header.php", але додатково складається з елементів 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.

```

admin-header.php  X
1  <a href="./index.php">Головна</a>
2  <a href="./admin.php">Додавання предметів</a>
3  <a href="./admin-user.php">Керування користувачами</a>
4  <a href="./admin-sms.php">Відправка смс</a>
5  <a href="./admin-raiting.php">Оцінки</a>
6  <a href="./admin-application.php">Заявки</a>
7  <a href="./exit.php">Вихід</a>

```

Рис.2.32. Структура файлу admin-header.php

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

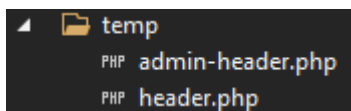


Рис.2.33. Структура папки temp

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

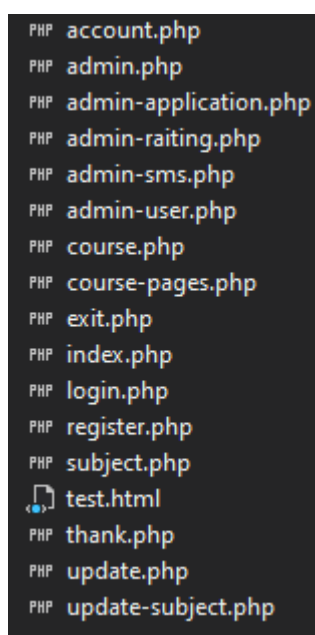


Рис.2.34. Структура папки web

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.

```

account.php
1
2 <!DOCTYPE html>
3 <html lang="en">
4
5 <head>
6     <meta charset="UTF-8">
7     <meta http-equiv="X-UA-Compatible" content="IE=edge">
8     <meta name="viewport" content="width=device-width, initial-scale=1.0">
9     <title>Document</title>
10    <link rel="stylesheet" href="./assets/style/index.css">
11 </head>
12
13 <body>
14     <?php
15
16     include './temp/header.php';
17
18     require_once './func/db.php';
19
20     $user_id = $_SESSION['id'];
21
22
23     if (!isset($_SESSION['id'])) {
24         header('Location: ./login.php');
25     }
26
27     $new = mysqli_query($connect, "SELECT * FROM `users` WHERE `id` = '$user_id'");
28     $new = mysqli_fetch_assoc($new);
29
30     >
31
32     <div class="wrapper-user">
33         <div class="user-information">
34             
35             <p class="name">Студент</p>
36             <div class="user-date-wrap">
37                 <div class="user-date">
38                     <p>Ім'я та прізвище</p>
39                     <p><?=$new['name'] ?> <?=$new['surname'] ?></p>
40                 </div>
41                 <div class="user-date">
42                     <p>Email</p>
43                     <p><?=$new['email'] ?></p>
44                 </div>
45                 <div class="user-date">
46                     <p>Дата народження</p>
47                     <p><?=$new['date'] ?></p>

```

Рис.2.35. Структура файлу account.php

admin.php

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

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

admin.php
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Document</title>
9   <link rel="stylesheet" href="./assets/style/index.css">
10 </head>
11
12 <body>
13   <div class="wrapper-admin">
14     <div class="admin-nav">
15       <?php include './temp/admin-header.php'; ?>
16     </div>
17   </div>
18   <div class="admin-form">
19     <form action="./func/add-subject.php" method="POST" enctype="multipart/form-data">
20       <h3>Додавання предмета</h3>
21       <input type="text" placeholder="Предмет" name="subject" required>
22       <input type="text" placeholder="Заголовок" name="title" required>
23       <textarea name="description" placeholder="Опис" id="" cols="30" rows="10"></textarea>
24
25       <p>Фото викладача</p>
26       <input type="file" name="image" required />
27       <input type="text" placeholder="Посада" name="position" required>
28       <input type="text" placeholder="ПІБ викладача" name="name" required>
29       <textarea name="for_teacher" placeholder="Про викладача" id="" cols="30" rows="10"></textarea>
30       <p>Виберіть курс</p>
31       <select name="course" id="">
32         <option value="1">1</option>
33         <option value="2">2</option>
34         <option value="3">3</option>
35         <option value="4">4</option>
36       </select>
37       <button>Надіслати</button>
38     </form>
39   </div>
40   <div class="wrapper-admin">
41     <table>
42       <tr>
43         <td>ID</td>
44         <td>Предмет</td>
45         <td>Заголовок</td>
46         <td>Опис</td>

```

Рис.2.36. Структура файлу admin.php

course.php

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

```

course.php
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Document</title>
9     <link rel="stylesheet" href="./assets/style/index.css">
10 </head>
11
12 <body>
13     <?php
14         include './temp/header.php';
15
16         $user_id = $_SESSION['id'];
17
18         if (!isset($_SESSION['id'])) {
19             header('Location: ./login.php');
20         }
21     >
22
23     <main>
24         <div class="block-course">
25             <div class="block-course_images">
26                 <div class="wrapper-img">
27                     
28                 </div>
29             </div>
30             <div class="block-course_text">
31                 <h2 class="block-course_title">
32                     Курс <br> - перший
33                 </h2>
34                 <p class="block-course_description">
35                     На першому курсі нашого університету ми пропонуємо вам розпочати свою навчальну подорож з ряду
36                     захоплюючих предметів, які допоможуть вам розкрити свій потенціал і набути необхідних знань у вашій
37                     обраній галузі.
38                 </p>
39                 <a href="./subject.php?course=1">
40                     
41                 </a>
42             </div>
43             <div class="block-course_text">
44                 <h2 class="block-course_title">

```

Рис.2.37. Структура файлу course.php

course-pages.php

Модуль відповідає за відображення основної сторінки певної дисципліни курсу. Він містить загальну інформацію про дисципліну, викладача, відгуки студентів тощо.

```

course-pages.php
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Document</title>
9     <link rel="stylesheet" href="./assets/style/index.css">
10 </head>
11
12 <body>
13
14     <?php
15     include './temp/header.php';
16
17     require_once './func/db.php';
18
19     $news_id = $_GET['id'];
20     $new = mysqli_query($connect, "SELECT * FROM `subject` WHERE `id` = '$news_id'");
21     $new = mysqli_fetch_assoc($new);
22
23     $id_blog = $new['id'];
24     ?>
25
26     <main>
27         <div class="course-page">
28             <div class="block-course">
29                 <div class="course-page__desc">
30                     <p class="category"><?=$new['subject'] ?></p>
31                     <h2><?=$new['title'] ?></h2>
32                     <p class="txt"><?=$new['description'] ?></p>
33                     <a href="#">Почати вивчення</a>
34                 </div>
35                 <div class="course-page__img">
36                     
37                 </div>
38             </div>
39             <div class="block-teacher">
40                 <div class="block-teacher__img">
41                     
42                 </div>
43                 <div class="block-teacher__desc">
44                     <h2><?=$new['position'] ?></h2>
45                     <h3><?=$new['name'] ?></h3>
46                     <p>

```

Рис.2.38. Структура файлу course-pages.php

index.php

Це один з важливих файлів системи, який відображає головну сторінку університету та містить загальну інформацію про проект, форму зворотнього зв'язку, а також навігаційне меню для переходу до різних розділів та функцій порталу.

```

index.php
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Document</title>
9   <link rel="stylesheet" href="./assets/style/index.css">
10 </head>
11
12 <body>
13   <style>
14     body,
15     html {
16       background: linear-gradient(90deg, rgba(229, 189, 246, 0.74) 15.91%, rgba(216, 222, 222, 0.74) 100%);
17     }
18   </style>
19   <div class="main-screen">
20     <?php include './temp/header.php'; ?>
21     <div class="main-screen__description">
22       <h1>Моє навчання</h1>
23       <a href="./account.php">Огляд -></a>
24     </div>
25   </div>
26   <a name="dovidka"></a>
27   <div class="main-col2 screen-1">
28     <div class="main-col2__wrapper">
29       <div class="main-col2__img">
30         
31       </div>
32       <div class="main-col2__txt">
33         <h2>Сенс та ціль проекту</h2>
34         <p class="mt-65">Проект, про який я б хотів розповісти, пов'язаний з моїм навчанням на 4 курсах
35           університету. У цьому проекті я хочу поділитися з вами своїм досвідом навчання, розповісти про
36           предмети, які я вивчав, про своїх викладачів та про те, як це все впливає на моє майбутнє. Одним з
37           плюсів навчання на 4 курсах є те, що вони дуже спеціалізовані. Університет дозволяє студентам
38           вибирати предмети за їх власними інтересами, тому я зміг вивчати тільки ті предмети, які мене
39           цікавили.</p>
40       </div>
41     </div>
42   </div>
43   <div class="main-col2 screen-2">
44     <div class="main-col2__wrapper">
45       <div class="main-col2__img">
46         <p>

```

Рис.2.39. Структура файлу index.php

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.

ВИСНОВКИ ДО ДРУГОГО 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 висвітлена розробка структури бази даних системи. Відбувся аналіз вимог до зберігання та обробки даних, визначення сутностей, атрибутів та

зв'язків між ними. Розроблялись структури таблиць, визначались типи даних та обмеження. Розглядалась оптимізація запитів та забезпечення цілісності даних.

Описана розробка структури файлів системи. Визначали основні файли, їх функціональність та взаємозв'язок. Зробили опис ролей та обов'язків кожного файлу, їх призначення та відповідальність. Розглянуті папки та підпапки, у яких розташовувались файли веб-орієнтованої системи освітнього процесу університету.

Таким чином, другий розділ кваліфікаційної роботи дозволяє детально спроектувати веб-орієнтовану систему, враховуючи пророблені кроки першого розділу та вимоги до проекту.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

3 ПОРЯДОК РОБОТИ ТА ТЕСТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ СИСТЕМИ ОСВІТНЬОГО ПРОЦЕСУ УНІВЕРСИТЕТУ

3.1 Архітектура веб-системи та її розгортання

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

Детально розглянемо архітектуру нашої веб-орієнтованої системи, спроектованої для реалізації функціональності інформаційного порталу. 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.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

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 системи

Також звернемо увагу на проектування інтерфейсу нашого проекту з використанням потужного інструменту Figma. Розібравшись з аналізом вимог та розумінням потреб користувачів, перейдемо до створення прототипів, дизайну

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

та розташування елементів інтерфейсу для забезпечення зручної та ефективної взаємодії користувача з додатком.

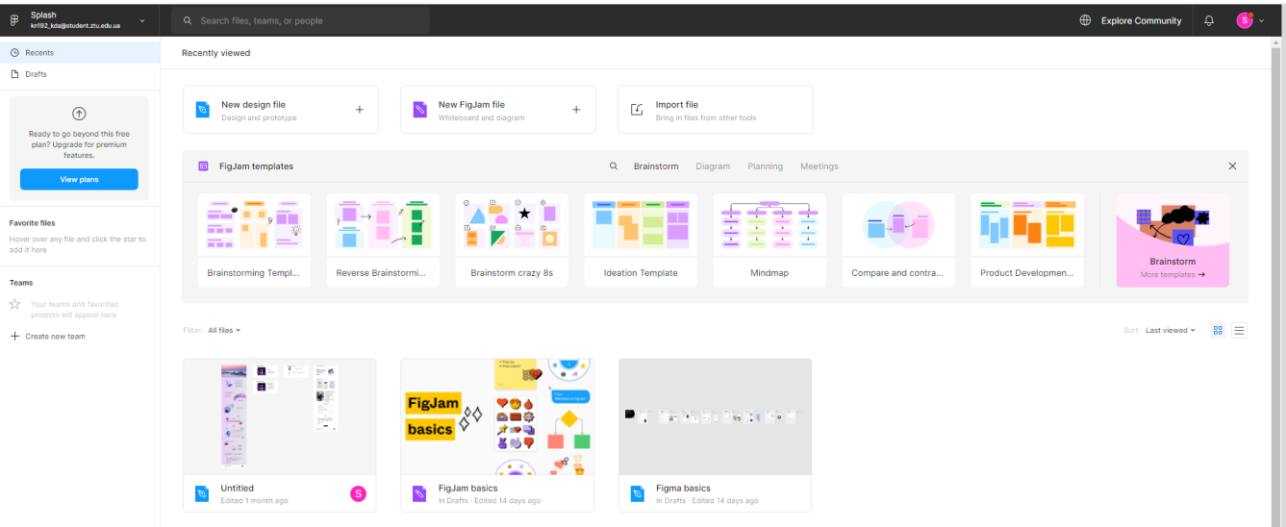


Рис.3.1. Головна сторінка онлайн-сервісу Figma

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.

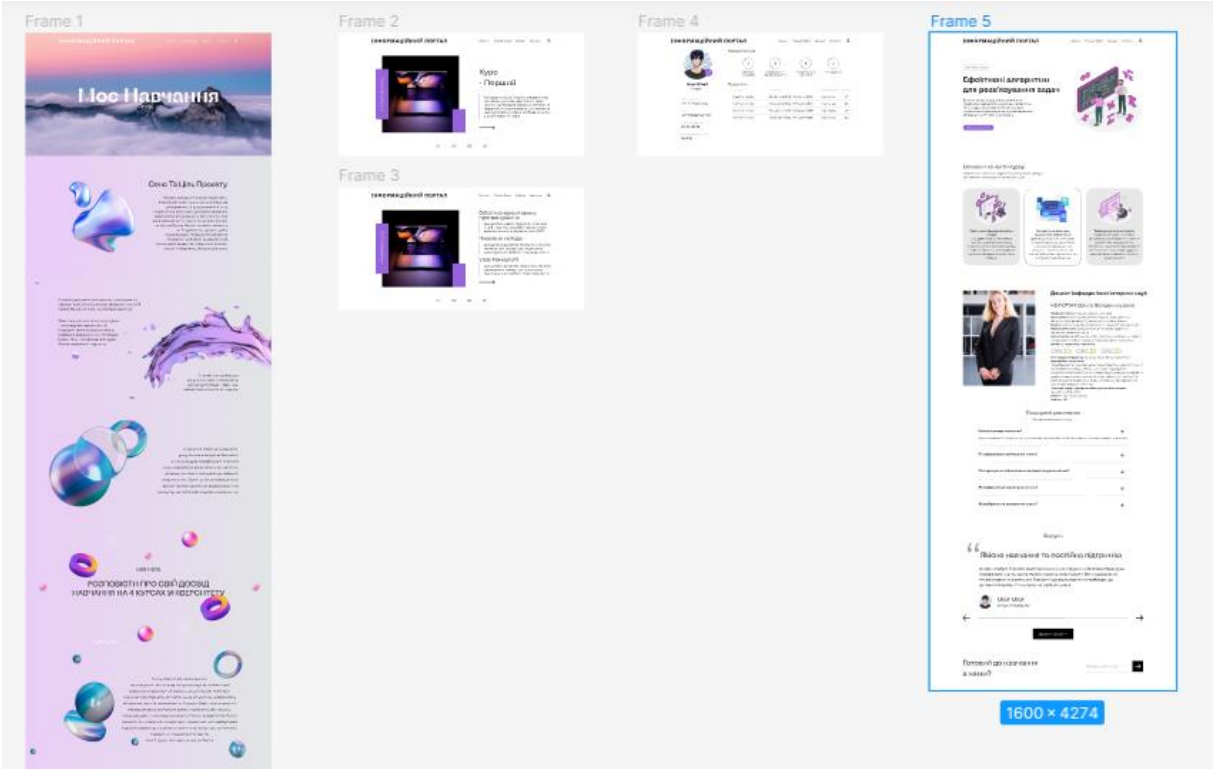


Рис.3.2. Розроблений прототип проекту

У нас є створений інтерфейс нашої веб-орієнтованої системи, тому детально розглянемо кожен із фреймів, що використовується.

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.

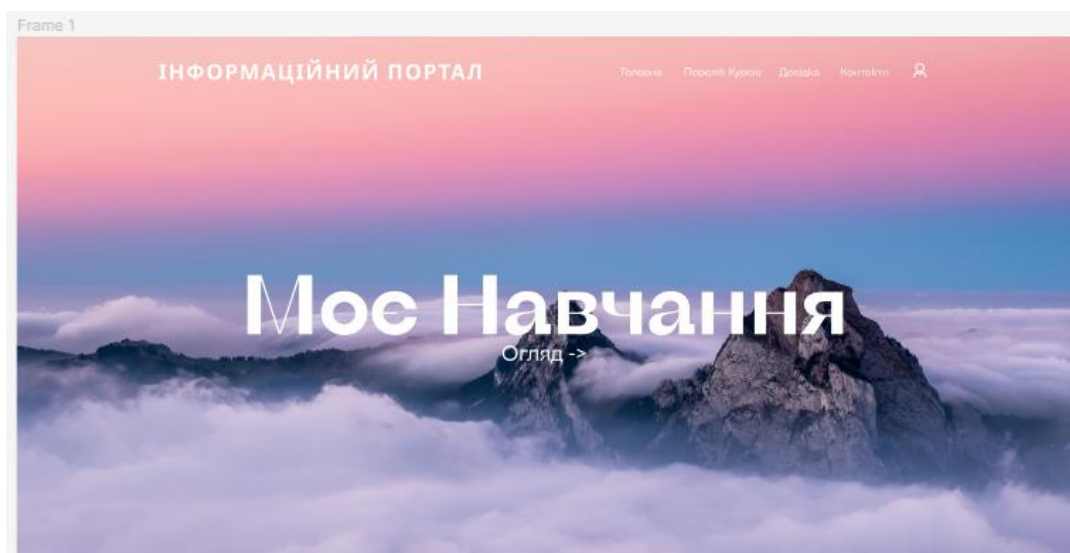


Рис.3.3. Frame 1. Головна сторінка

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

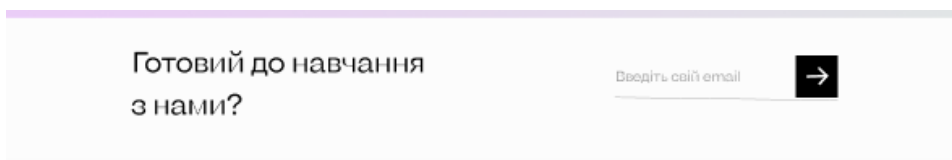


Рис.3.4. Frame 1. Футор головної сторінки

Frame 2

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

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.

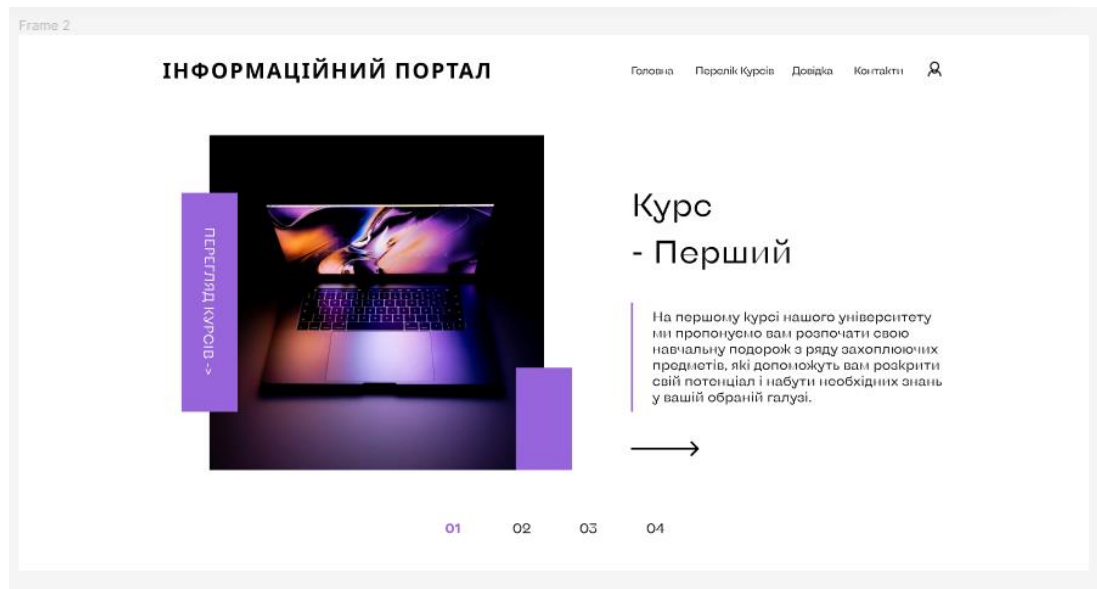


Рис.3.5. Frame 2. Сторінка з переліком курсів

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 кожну дисципліну відображається зрозуміло та естетично.

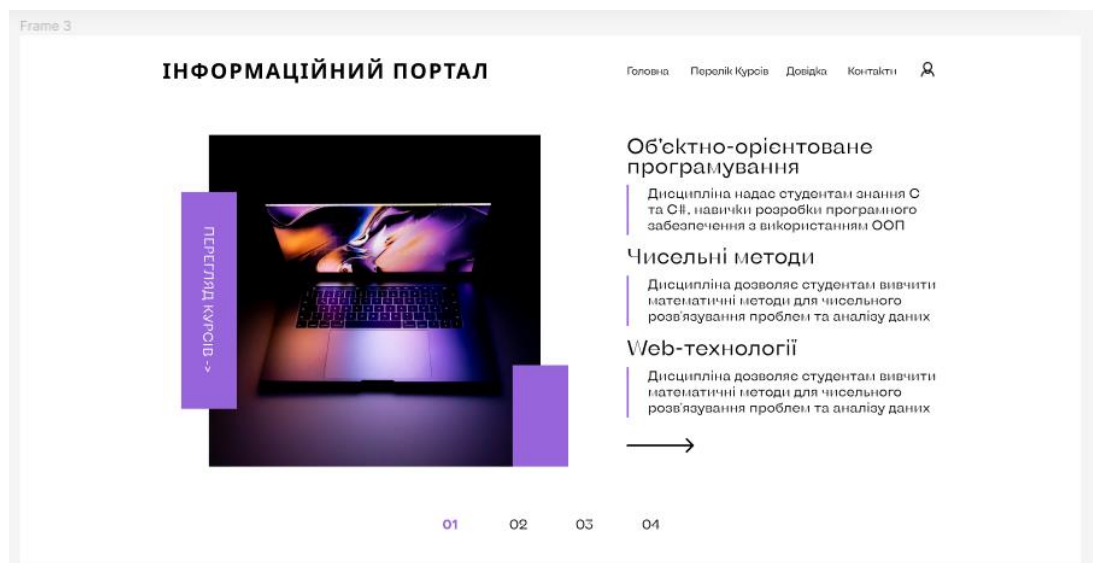


Рис.3.6. Frame 3. Сторінка з навчальними предметами курсів

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.

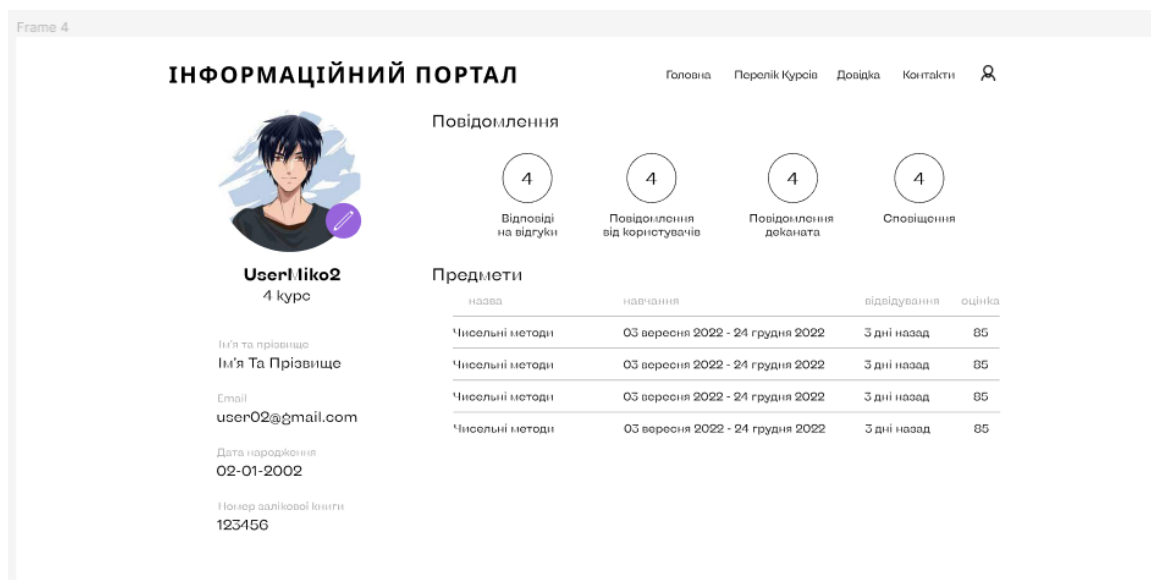


Рис.3.7. Frame 4. Сторінка особистого кабінету користувача

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.

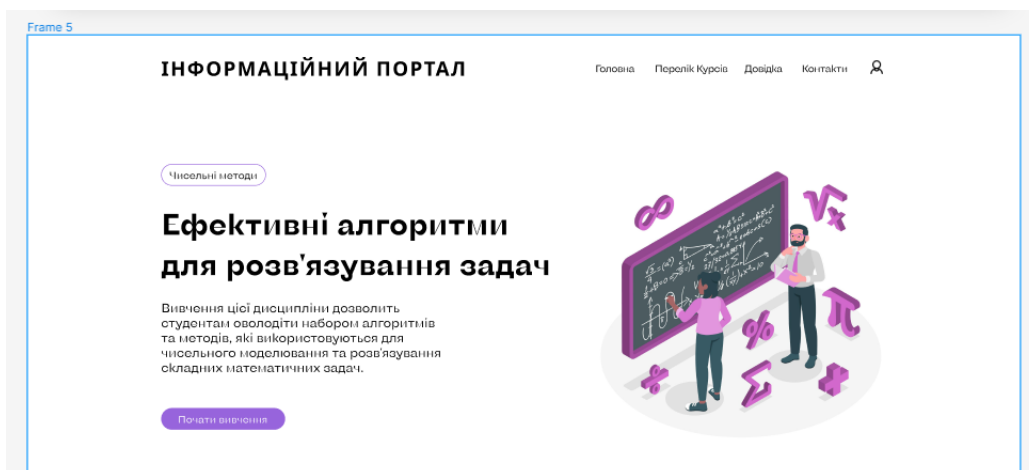


Рис.3.8. Frame 5. Частина основної сторінки з описом предмету

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.



Доцент кафедри комп'ютерних наук

КОРОТУН Ольга Володимирівна

Науковий ступінь: кандидат педагогічних наук
Alma mater: Національного університету імені Василя Стуса, Житомирський державний університет імені Івана Франка
Освіта: магістр з прикладної математики, спеціаліст з інформатики
Наукова діяльність: автор (співавтор) понад 50 наукових та навчально-методичних праць
Наукові інтереси: бази даних, інтелектуальний аналіз даних, чисельні методи в комп'ютерних науках, системи управління навчанням
Профілі у наукометричних базах:

І міжнародні стажування: в університетах Польщі, Туреччини
Сертифікати та дипломи:
 *Certificate of Kutahya Dumlupinar Universitesi (Turkiye) Certificate of Humanities and Natural Sciences University in Sandomierz (Poland) сертифікат від IT Ukraine Associate in Education, сертифікат з проблем модернізації педагогічної освіти в Україні та в країнах ЄС, Certificate of Development of Digital and Didactic Competences of High School Teachers (Ukraine)
Членство в радах, дорадчих та консультативних органах:
 *Імені Вченої Ради ФІКТ
Email: korotun-olga@stu.od.ua
Кабінет: 332

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 продемонстрували чітке розуміння структури та функціональності розробленої веб-орієнтованої системи, що є важливим етапом в процесі розробки програмного коду інтерфейсу.

3.3 Тестування роботи системи

У цьому розділі детально розглянемо процес тестування розробленої веб-орієнтованої системи та 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 дозволять 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.

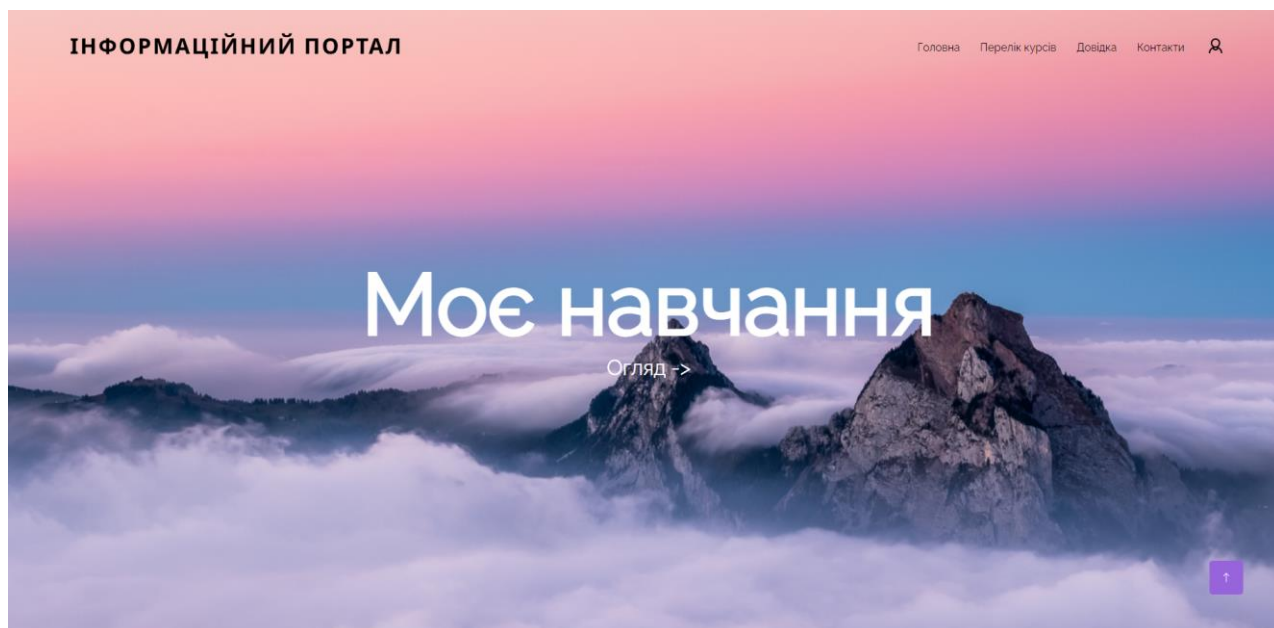


Рис.3.10. Вигляд головної сторінки

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.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

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.

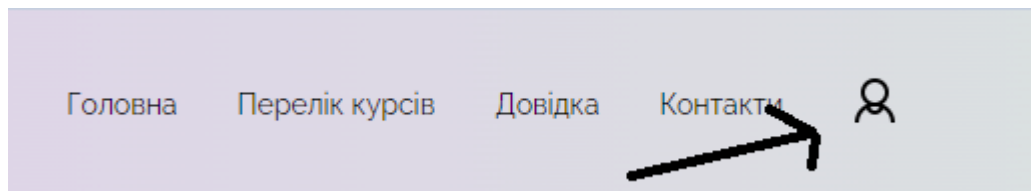


Рис.3.14. Кнопка особистого кабінету

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.

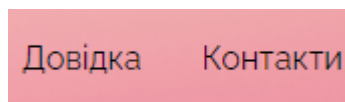


Рис.3.16. Елементи "Довідка" та "Контакти"

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

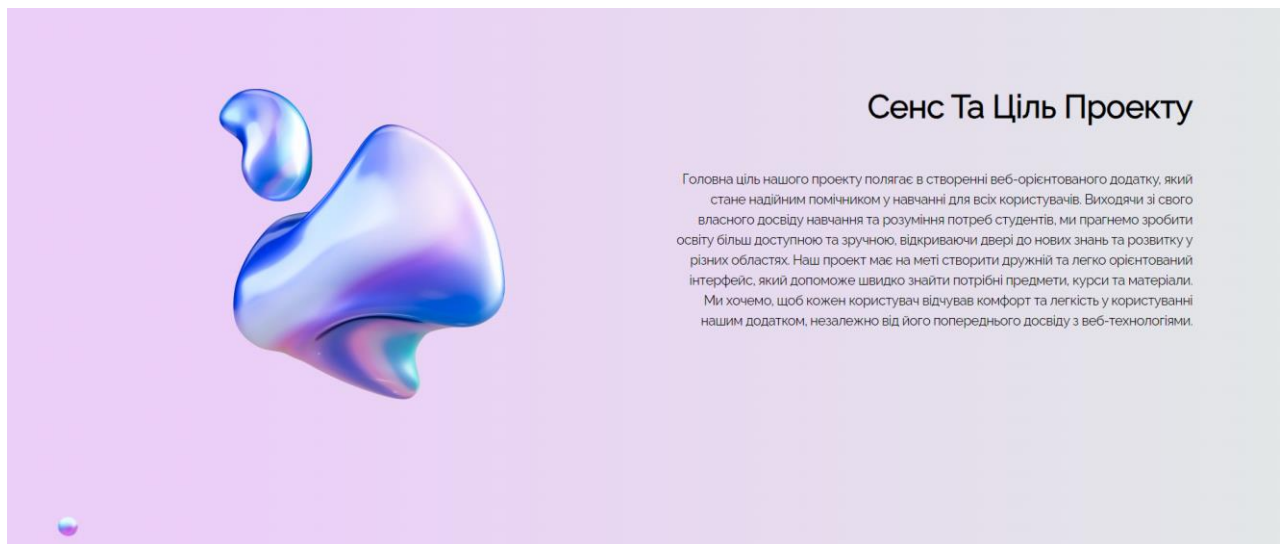


Рис.3.17. Довідка сайту

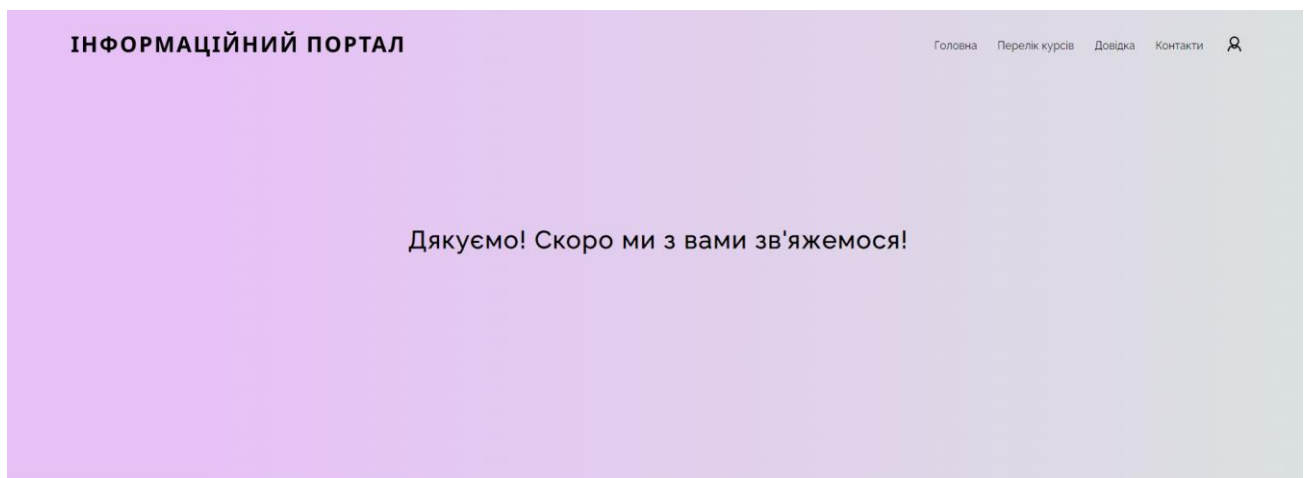
"Контакти" перенаправляють користувача на форму зворотнього зв'язку, де він може залишити свою електронну пошту для адміністратора проекту.

ГОТОВИЙ ДО НАВЧАННЯ З НАМИ?

Введіть свій email



Рис.3.18. Форма зворотнього зв'язку



ГОТОВИЙ ДО НАВЧАННЯ З НАМИ?

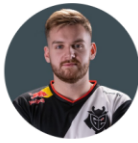
Введіть свій email



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.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		



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

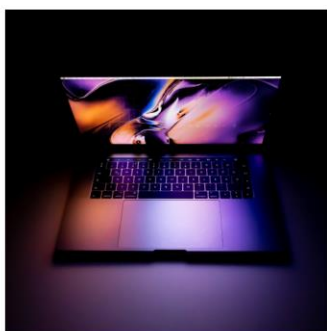
На першому курсі нашого університету ми пропонуємо вам розпочати свою навчальну подорож з ряду захоплюючих предметів, які допоможуть вам розкрити свій потенціал і набути необхідних знань у вашій обраній галузі.



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 дисциплінами, 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 дисциплін

На основній сторінці можна переглянути загальну інформацію про обрану дисципліну, викладача, який його веде, та відгуки студентів про цього викладача. Вона допомагає користувачам отримати повну картину про дисципліну та викладача перед початком та протягом 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.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

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



ст.викладач

**ЛЕВКІВСЬКИЙ Віталій
Леонідович**

Освіта: спеціаліст. спеціальність «Інформаційні управлюючі 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 базис:

ORCID

Google Scholar

Scopus

WEB OF SCIENCE

Email: levkivskyi@ztu.edu.ua

Кабінет: 102а

Відгуки молодих спеціалістів Житомирської Політехніки:

Літовченко М.А.

"Годний препод. Хз що сказать"

Мурович В.Ю.

"Нормальний"

Копилов В.Р.

"Молодий та дикий, житомирянин"

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 повний контроль над системою.

Головна Додавання предметів Керування користувачами Відправка смс Оцінки Заявки Вихід

Рис.3.26. Панель керування адміністратора

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

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Додавання предмета

Системний аналіз та теорія прийняття рішень

Системний аналіз та прийняття рішень: Розкриття потенціалу ефективного у

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.

Фото викладача

Выберите файл 2110.png

доцент кафедри комп'ютерної інженерії та кібербезпеки

БРОДСЬКИЙ Юрій Борисович

Навковий ступінь: кандидат технічних наук

Вчене звання: доцент

Alma mater: Житомирське вище училище радіоелектроніки ППО

Освіта: Житомирське вище училище радіоелектроніки ППО, 1985, спеціальність - радіотехнічні засоби, кваліфікація - радіоінженер

Наукова діяльність: Автор більше 70 наукових праць.

Виберіть курс

3

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 курси. Під час тестування переконались, що зміни вносяться коректно, а інформація про курс оновлюється на всіх відповідних сторінках.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

Редагування предмета

Системний аналіз та теорія прийняття рішень

Системний аналіз та прийняття рішень: Розкриття потенціалу ефективного у

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.

доцент кафедри комп'ютерної інженерії та кібербезпеки

БРОДСЬКИЙ Юрій Борисович

Науковий ступінь: кандидат технічних наук

Вчене звання: доцент

Alma mater: Житомирське вище училище радіоелектроніки ППО

Освіта: Житомирське вище училище радіоелектроніки ППО, 1985, спеціальність - радіотехнічні засоби, кваліфікація - радіоінженер

Наукова діяльність: Автор більше 70 наукових праць.

Підтвердити

Рис.3.29. Модуль редагування предмету

Адміністратор має можливість переглядати базу даних користувачів та видаляти їх облікові записи. 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 перевірили, що повідомлення досягають користувачів, та студенти мають змогу переглядати

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

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

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

Рис.3.32. Результат відображення повідомлення в кабінеті користувача

Адміністратор може виставляти оцінки певним користувачам за певні дисципліни. 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

Також, адміністратор має можливість переглядати контакти, які надійшли через форму зворотнього зв'язку від відвідувачів та користувачів. Під час тестування 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 переконались, що всі вони працюють належним чином, забезпечують потрібний функціонал та безпроблемне управління розробленої веб-системою.

ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ

У цьому розділі детально розглянуті кілька важливих аспектів веб-орієнтованої системи – архітектура та її розгортання, проектування загального інтерфейсу та тестування її роботи. Описані ключові етапи процесу розробки та впровадження веб-орієнтованої системи, що забезпечують її ефективність та якість функціонування.

Висвітлені важливі аспекти структури розробленої веб-орієнтованої системи, використовуючи модульний підхід, який сприяє зручності розробки, підтримці та розширенню. Також процес розгортання системи, враховуючи вибір хостингової платформи та конфігурацію серверного середовища.

Для користувацько-орієнтованого інтерфейсу використані мінімалістичний стиль з глассморфізмом, створені зручні та привабливі фрейми, враховуючи сучасні тенденції дизайну. Також впровадили функціональні елементи, які забезпечують зручну навігацію для користувачів та забезпечують ефективність використання додатку.

Зроблена перевірка функціональності та взаємодії користувачів з веб-орієнтованою системою. Провели тестування головної сторінки, кабінету користувача та окремих функцій, включаючи адміністрування. Цей процес дозволив виявити та усунути помилки, забезпечити безперебійну роботу та зручність використання додатку користувачами.

Загалом, завдяки архітектурі додатку, розробленому інтуїтивному інтерфейсу та тестуванню функціоналу, забезпечили створення веб-орієнтованої системи освітнього процесу університету. Дотримуючись передових методів розробки та використовуючи найкращі практики, створили надійний та ефективний програмний продукт, який готовий задовольнити потреби наших користувачів.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

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

При проектуванні веб-орієнтованої системи ретельно вивчено структуру та варіанти її використання. Розроблена архітектура системи та встановлено залежності між її компонентами. Також спроектована структура бази даних та файлової системи, що забезпечують ефективне зберігання та організацію даних.

Порядок роботи та тестування системи чітко визначені та виконані. Розроблені загальний та функціональний алгоритми веб-системи, а також проведені тестування їх роботи. Цей процес дозволив перевірити функціональність, надійність та зручність її використання.

У підсумку, результатом нашої кваліфікаційної роботи є готовий програмний продукт, який задовольняє мінімальні потреби користувача. Має приємний та зручний інтерфейс, що сприяє його легкому використанню. Крім того, заздалегідь визначений вектор розвитку дозволить покращити функціонал веб-орієнтованої системи та привернути нову аудиторію в майбутньому.

В процесі роботи над кваліфікаційним проектом, здобули значний досвід у розробці веб-систем та керуванні проектами. Вибір відповідних технологій та архітектури, а також ретельне аналізування вимог користувача дозволили досягти успішного результату. Цей проект став вагомим внеском в нашу професійну підготовку та вирішення реальних завдань в галузі ІТ.

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						66
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Brown M. HTML. 2nd ed. Paris : Simon & Schuster Macmillan, 1996.
2. Institute Of Electrical and Electronics Engineers. HTML Fundamentals. Institute of Electrical & Electronics Enginee, 1997.
3. Rule J. Dynamic HTML: The HTML Developer's Guide. Addison-Wesley Professional, 1998.
4. Kaur G. Css: What Is CSS?. Independently Published, 2022.
5. McFarland D. S. CSS. O'Reilly Media, Incorporated, 2006.
6. Jansch I. PHP/Architect's Guide to Enterprise PHP Development (PHP Arthitect Nanobooks.). musketeers.me, LLC, 2008.
7. Gibbs P. PHP Tutorials : Programming with PHP and MySQL: Learn PHP 7. Gibbs, Paul, 2022.
8. Mitra S. Php for Beginners: Learn PHP MySQL. Independently Published, 2018.
9. Vickler A. Javascript: Javascript Basics for Beginners. Independently Published, 2021.
10. Goddard N. Javascript: Beginners Guide on Javascript Programming. CreateSpace Independent Publishing Platform, 2016.
11. W3Schools [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com/>
12. Stack Overflow [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com/>
13. Документація MySQL [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>
14. Документація HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/HTML>
15. Документація CSS [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/CSS>

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

- 16.Документація JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- 17.GitHub [Електронний ресурс] – Режим доступу до ресурсу: <https://github.com/>
- 18.Веб-сайт університету [Електронний ресурс] – Режим доступу до ресурсу: <https://ztu.edu.ua/>
- 19.Документація CSS Grid [Електронний ресурс] – Режим доступу до ресурсу: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout
- 20.Документація PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/docs.php>
- 21.Платформа вивчення PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/ru/tutorial.php>
- 22.Офіційний сайт MAMP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mamp.info/>
- 23.Документація MAMP [Електронний ресурс] – Режим доступу до ресурсу: <https://documentation.mamp.info/>
- 24.Офіційний сайт Visual Studio [Електронний ресурс] - Режим доступу до ресурсу: <https://visualstudio.microsoft.com/>
- 25.Всесвітня організація охорони здоров'я (ВООЗ) [Електронний ресурс] - Режим доступу до ресурсу: <https://www.who.int/>
- 26.Офіційний сайт "PostgreSQL" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.postgresql.org/>
- 27.Офіційний сайт "Oracle Database" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.oracle.com/database/>
- 28.Офіційний сайт "Microsoft SQL Server" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.microsoft.com/en-us/sql-server/>
29. Веб-орієнтована система "Blackboard" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.blackboard.com/>

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

- 30.Веб-орієнтована система "Moodle" [Електронний ресурс] - Режим доступу до ресурсу: <https://moodle.org/>
- 31.Веб-орієнтована система "Canvas" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.canvaslms.com/>
- 32.Веб-орієнтована система "Schoology" [Електронний ресурс] - Режим доступу до ресурсу: <https://www.schoology.com/>
- 33.Веб-орієнтована система "Google Classroom" [Електронний ресурс] - Режим доступу до ресурсу: <https://classroom.google.com/>
- 34.Онлайн-інструмент для моделювання баз даних [Електронний ресурс] - Режим доступу до ресурсу: <https://dbdiagram.io/home>
- 35.Онлайн-інструмент для створення діаграм [Електронний ресурс] - Режим доступу до ресурсу: <https://www.lucidchart.com>

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

Файл thank.php

```

<!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-wrapper">
    <?php include './temp/header.php'; ?>
    <h2 class="thanks-text">Дякуємо! Скоро ми з вами зв'яжемося!</h2>
    <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>
  </div>

</body>

</html>

```

Файл exit.php

```

<?php

session_start();
session_destroy();

header('Location: ./index.php');

exit;
?>

```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						71
Змн.	Арк.	№ докум.	Підпис	Дата		

Файл index.js

```
let blockCourse = document.querySelectorAll('.block-course__text'),
    btnBlockCourse = document.querySelectorAll('.pagination button');

function hiddenBlock(){
    blockCourse.forEach(item => {
        item.style.display = 'none';
    });
}

function showBlock(i){
    blockCourse[i].style.display = 'grid';
}

btnBlockCourse.forEach((item, i) => {
    item.addEventListener('click', (e) => {
        for(let j = 0 ; j < btnBlockCourse.length ; j++){
            btnBlockCourse[j].classList.remove('active');
        }
        item.classList.add('active');
        hiddenBlock();
        showBlock(i);
    })
});

hiddenBlock();
showBlock(i = 0);
```

					КН.КР.Б – 122 – 23 – ПЗ	Арк.
						72
Змн.	Арк.	№ докум.	Підпис	Дата		