

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ
кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

До захисту допущено:

Завідувач кафедри

_____ Наталія АУШЕВА

«__» _____ 2024 р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
“Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему: **Платформа дистанційного навчання за дисциплінами
кафедри**

Виконала:
студентка IV курсу, групи ТМ-01

_____ ТИЩЕНКО Анастасія Андріївна

(прізвище, ім’я, по батькові)

_____ (підпис)

Керівник: *доцент каф. цифрових технологій в енергетиці*

_____ доцент, к.е.н., Ірина СЕГЕДА

(посада, вчене звання, науковий ступінь, ім’я, ПРІЗВИЩЕ)

_____ (підпис)

Рецензент: *доцент каф. інженерії програмного забезпечення в енергетиці*

_____ доцент, к.т.н., Валерій КУЗЬМІНИХ

(посада, вчене звання, науковий ступінь, ім’я, ПРІЗВИЩЕ)

_____ (підпис)

Н.контроль: _____ доцент Андрій ДОНЕЦЬ

(посада, ім’я, ПРІЗВИЩЕ)

_____ (підпис)

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

Студент (-ка) _____
(підпис)

Київ – 2024

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – перший (бакалаврський)

спеціальність 122 “Комп’ютерні науки”

Освітньо-професійна “Цифрові технології в енергетиці”
програма _____

ЗАТВЕРДЖУЮ

Завідувач кафедри ЦТЕ

_____ Наталія АУШЕВА

«__» _____ 2024 р.

ЗАВДАННЯ
на дипломну роботу студенту

ГІЩЕНКО Анастасії Андріївни

(прізвище, ім’я, по батькові)

1. Тема роботи **Платформа дистанційного навчання за дисциплінами**
кафедри

керівник роботи **Сегеда Ірина Василівна, к.е.н., доцент**

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__» _____ 202__ р.

№ _____

2. Термін подання роботи студентом **7.06.2024 року**

3. Вихідні дані до роботи мови програмування HTML, CSS, JavaScript, PHP,
середовище розробки PhpStorm, MySQL Workbench, система керування базами
даних MySQL, інструмент контролювання дії окремих компонентів XAMPP
control panel, середовище побудови діаграм Visual paradigm online

4. Перелік питань, які потрібно розробити _____

1) провести аналіз серед наявних аналогів

2) обґрунтувати вибір програмних застосунків і шляхи розробки інформаційної

системи

3) розробити програмне забезпечення

4) розробити користувацький інтерфейс для візуалізації роботи системи

5. Перелік ілюстративного матеріалу Веб-додаток платформи для навчання Moodle, веб-додаток платформи для сервісу для дистанційного навчання Google classroom, веб-додаток освітньої міжнародної платформи Coursera, діаграма прецедентів, діаграма класів, розширена діаграма сутності та зв'язку, ХАМРР Control Panel, сторінка входу, особистий кабінет, сторінка новин, вкладка календар, вкладка курси, вкладка лекції, контрольна робота, вкладка після пройденої контрольної роботи, завантажений файл з тестом, вкладка оцінки, вкладка лабораторних робіт, вкладка завдання до лабораторної роботи, повідомлення у разі успішного завантаження файлу, вихід з системи.

6. Дата видачі завдання «19» вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Затвердження теми роботи		
2.	Вивчення та аналіз задачі	17–20.04.24	
3.	Розробка архітектури та загальної структури системи	21–25.04.24	
4.	Розробка частин окремих підсистем	25.04–02.05.24	
5.	Програмна реалізація системи	03–07.05.24	
6.	Оформлення пояснювальної записки	08–12.05.24	
7.	Захист програмного продукту	13–17.05.24	
8.	Передзахист	03–06.06.24	
9.	Подання готової роботи на кафедру	07.06.2024	
10.	Захист	17–21.06.2024	

Студент

(підпис)

Анастасія ТИЩЕНКО

(ім'я, ПРІЗВИЩЕ)

Керівник роботи

(підпис)

Ірина СЕГЕДА

(ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Метою дипломної роботи є створення онлайн платформи дистанційного навчання за дисциплінами кафедри, основними складовими якої є інтерфейс для студентів, доступ до лекцій, новин, календарю, можливість здачі лабораторних робіт та перегляд оцінок. Платформа написана за допомогою редактора вихідного коду PhpStorm. Для розробки системи використано технології: HTML, CSS, Javascript, php, MySQL.

Дипломна записка складається зі вступу, 5 розділів і висновків. Записка містить 71 сторінку, 20 рисунків, 19 посилань і 1 додаток.

Ключові слова: дистанційне навчання, навчальні матеріали, освіта, студенти.

ABSTRACT

The aim of the thesis is to create an online platform for distance learning in the disciplines of the department, the main components of which are an interface for students, access to lectures, news, calendar, the possibility of submitting laboratory work and viewing grades. The platform is written using the PhpStorm source code editor. The following technologies were used to develop the system: HTML, CSS, Javascript, php, MySQL.

The dissertation consists of an introduction, 5 chapters and conclusions. The note contains 71 pages, 20 figures, 19 references and 1 appendix.

Keywords: distance learning, educational materials, education, students.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП.....	8
1 ЗАДАЧА РЕАЛІЗАЦІЇ ДИСТАНЦІЙНОЇ ПЛАТФОРМИ ДЛЯ НАВЧАННЯ ЗА ДИСЦИПЛІНАМИ КАФЕДРИ.....	10
1.1 Призначення дистанційної платформи	11
1.2 Потенційні користувачі дистанційної платформи	11
1.3 Аналіз вимог для реалізації дистанційної платформи	12
2 АНАЛІЗ ІСНУЮЧИХ ПЛАТФОРМ ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ.....	15
2.1 Платформа для навчання Moodle.....	15
2.2 Сервіс для дистанційного навчання Google classroom.....	17
2.3 Освітня міжнародна платформа Coursera	18
3 ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	21
3.1 Мова розмітки гіпертексту HTML.....	21
3.2 Мова стилів CSS	22
3.3 Високорівнева скриптова мова Javascript	23
3.4 Скриптова мова php.....	25
3.5 СКБД MySQL.....	26
3.6 Середовище розробки phpStorm	27
3.7 Середовище розробки MySQL Workbench	28
3.8 Інструмент керування XAMPP Control Panel	29
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	30
4.1 Реалізовані функції.....	30
4.2 UML–діаграми	31
4.3 Таблиці баз даних	34

4.4 Зв'язки між таблицями баз даних.....	35
5 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ІНФОРМАЦІЙНОЮ СИСТЕМОЮ ...	37
5.1 Системні вимоги та встановлення інформаційної системи.....	37
5.1.1 Вимоги до комп'ютера.....	37
5.1.2 Вимоги до програмного забезпечення	38
5.2 Взаємодія користувача з платформою дистанційного навчання	38
ВИСНОВКИ	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК А	51

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

HTML (HyperText Markup Language) – мова розмітки гіпертексту

CSS (Cascading Style Sheets) – каскадні таблиці стилів

UI (User Interface) – інтерфейс користувача

PDF (Portable Document Format) – портативний формат документів

ВСТУП

У зв'язку зі стрімким розвитком технологій і зростанням інтересу до дистанційного навчання сучасні навчальні заклади повинні створити платформи для дистанційного навчання за дисциплінами кафедри. Віддалене навчання стає все більш популярним серед студентів, оскільки воно зручне та гнучке. Ця платформа дозволить отримувати доступ до навчального матеріалу в будь-який час і з будь-якого місця, що покращить якість освіти та дозволить краще контролювати навчальний процес.

Мета створення платформи дистанційного навчання за дисциплінами кафедри полягає в тому, щоб надати студентам доступ до навчальної літератури, організувати навчальний процес і ефективно контролювати свої знання за допомогою сучасних інформаційних технологій. Багато платформ, які вже існують, пропонують різні способи навчання, такі як зберігання інформації, організація курсів, тестування, відео-лекції, та багато іншого.

Тим не менш, багато існуючих платформ не відповідають вимогам кафедри. Наприклад, вони можуть бути складними для студентів у використанні, не забезпечувати можливості для студентів використовувати індивідуальний підхід, або вони можуть не працювати разом з іншими навчальними системами, які вже використовуються на кафедрі. Отже, потрібна спеціалізована платформа, яка задовольнить всі вимоги кафедри та забезпечить зручний і ефективний навчальний процес, щоб впровадити дистанційне навчання.

Для реалізації такої інформаційної системи слід використовувати новітні технології для легшого подальшого впровадження у сферу освіти, а результати роботи такої системи можуть бути використані різними навчальними закладами вищої освіти.

У першому розділі пояснювальної записки сформульовано постановку задачі, описано загальну мету й задачі які розв'язуються розробленою системою. У другому – проаналізовано наявні системи аналогічні до створеної. У третьому

розділі обґрунтовано вибір засобів й технологій для створення інформаційної системи. Реалізовані функції, UML–діаграми, таблиці баз даних та зв'язки між таблицями баз даних описано в четвертому розділі. У п'ятому розділі описано процес розгортання системи та подано приклади взаємодії з нею.

1 ЗАДАЧА РЕАЛІЗАЦІЇ ДИСТАНЦІЙНОЇ ПЛАТФОРМИ ДЛЯ НАВЧАННЯ ЗА ДИСЦИПЛІНАМИ КАФЕДРИ

Зокрема у сфері освіти, ІТ–технології пропонують щодня нові та інноваційні інструменти. Одним із найважливіших інструментів є платформи дистанційного навчання. Від шкіл до закладів вищої освіти ця технологія зарекомендувала себе як революційна, і її використовують у всьому світі.

Завдяки поширенню дистанційного навчання в повсякденному житті зростає попит на зручні, зрозумілі та доступні платформи для отримання високоякісної освіти. Дистанційні платформи дають студентам нові можливості, дозволяючи краще організувати навчальний процес і надавати доступ до ресурсів у будь-який час, незалежно від місця перебування.

Такі платформи допомагають створити інтерактивне середовище навчання, де студенти можуть не лише виконувати завдання та переглядати лекції, але й брати участь у віртуальних класах і спілкуватися з однокурсниками в реальному часі. Це призводить до активнішої залученості та підвищує бажання навчатися.

Крім того, важливо відзначити, що завдяки прогресу технологій інтеграція з різними додатковим сервісами стає можливою. До таких сервісів належать електронні бібліотеки, платформи для проведення онлайн–іспитів і оцінювання, інструменти для автоматизації адміністративних процесів і багато іншого. Зважаючи на це, дистанційні платформи є незамінним інструментом для сучасної освіти.

Таким чином, дистанційні освітні платформи відповідають сучасним потребам у навчанні та значно покращують доступність і якість навчання. Вони забезпечують можливість постійного розвитку та вдосконалення знань, що є важливим для успіху в сучасному світі, який швидко змінюється.

1.1 Призначення дистанційної платформи

Дистанційна платформа за дисциплінами кафедри створена для оптимізації навчального процесу, забезпечення зручного доступу до матеріалів і підтримки комунікації між студентами. Вона не тільки надає студентам можливість переглядати лекції, презентації, статті, підручники та інші навчальні матеріали, але й надає їм можливість завантажувати матеріали для офлайн використання.

Платформа дозволяє планувати навчальний процес та архівувати заняття для перегляду. Онлайн-тести, контрольні роботи та лабораторні роботи використовуються для оцінки знань.

Платформа також дозволяє адаптувати навчання, відстежувати прогрес студентів і адаптувати матеріали відповідно до їх потреб.

Можливість користуватися платформою з будь-якого пристрою, підключеного до Інтернету, підтримка різних форматів файлів і інтеграція з іншими онлайн-інструментами роблять її доступною і зручною у використанні. Завдяки цьому студенти можуть навчатись у будь-який час та у будь-якому місці, що сприяє підвищенню ефективності та гнучкості навчального процесу.

1.2 Потенційні користувачі дистанційної платформи

Студенти, які прагнуть отримати високоякісну освіту в зручному для них форматі, є потенційними користувачами дистанційної платформи. Ці студенти можуть бути з різних курсів і спеціальностей, від першокурсників, які тільки починають свій шлях, до старшокурсників, які готуються до випуску та написання дипломних робіт.

Платформа дозволяє першокурсникам легко перейти від шкільного до університетського навчання, збираючи всі необхідні матеріали та ресурси в одному місці. Вони можуть використовувати платформу для ознайомлення з курсами, участі в лекціях і планування навчального процесу.

Старшокурсники з більш специфічними потребами можуть використовувати

платформу, щоб отримати доступ до більш детальних матеріалів, проведених наукових досліджень. Крім того, вони отримують можливість здавати онлайн-тести та контрольні роботи, що допомагає їм ефективно планувати час і готуватися до сесій.

Платформа дозволяє іноземним студентам або тим, хто тимчасово за межами країни безпечно навчатися незалежно від місцезнаходження. Вони можуть брати участь у всіх навчальних заходах, як і їх колеги.

Окрім академічних переваг, платформа навчає самостійно вчитися та керувати часом, що важливо для успішної кар'єри. Студенти мають можливість активно керувати своїм навчальним процесом, вибираючи, коли їм зручно вивчати матеріали та в який темп.

1.3 Аналіз вимог для реалізації дистанційної платформи

Розробка дистанційної платформи вимагає ретельного аналізу. Це включає розгляд функціональних і нефункціональних вимог, а також технічних аспектів, необхідних для забезпечення ефективної та стабільної роботи системи. Основною метою платформи є надання користувачам можливості отримувати знання та навички онлайн, що дозволяє зменшити перешкоди для доступу до навчання та зробити його більш гнучким.

Функціональні вимоги визначають, які конкретні функції та можливості система повинна виконувати, наприклад автентифікація користувачів. Платформа повинна мати надійні функції для автентифікації існуючих користувачів. Це включає підтримку різних посад, таких як адміністратор або студент.

Управління контентом. Платформа повинна підтримувати широкий спектр навчального контенту, такого як тексти, презентації, лекції, контрольні роботи, лабораторні роботи, тести та інші.

Навчальні курси та програми. Платформа повинна мати можливість створювати та керувати освітніми програмами. Налаштування структури курсу,

додавання матеріалів і встановлення графіків навчального процесу – усе це відноситься до цієї категорії.

Система здачі лабораторних робіт та тестів. Основним елементом є можливість здачі лабораторних робіт, тестів та контрольних робіт. Це включає можливість ручної перевірки лабораторних робіт і автоматичне оцінювання тестів та контрольних робіт.

Також є вимоги, які не є функціональними, наприклад продуктивність і масштабованість. Платформа повинна витримати велику кількість роботи, особливо під час пікових періодів, таких як початок навчального року. Система повинна мати здатність адаптуватися до зростаючої кількості користувачів і контенту.

Безпека. Гарантування захисту даних є життєво важливим. Це включає безпечне зберігання навчальних матеріалів, захист персональних даних користувачів і запобігання несанкціонованому доступу та кіберзагрозам.

Інтерфейс користувача та зручність використання. Інтерфейс платформи повинен бути простим для розуміння та зручним для використання. Важливо, щоб користувачі легко знаходили та швидко навчалися використовувати систему.

Сумісність і доступність. Платформа повинна працювати з багатьма браузерами та пристроями, включаючи мобільні пристрої. Забезпечуючи відповідність стандартам доступності.

Для впровадження дистанційної платформи необхідно ретельно вибрати технології та архітектуру системи.

Архітектура системи. Вибір мікросервісної або монолітної архітектури залежить від потреби в масштабованості та гнучкості. Мікросервісна архітектура може забезпечити кращу масштабованість і можливість розгортання різних елементів системи незалежно.

Технологічний стек. При виборі мови програмування, фреймворків і баз даних необхідно враховувати продуктивність, безпеку та підтримку. Популярними варіантами можуть бути мови програмування, такі як HTML, CSS, JavaScript, PHP,

середовище розробки PhpStorm, MySQL Workbench, система керування базами даних MySQL, інструмент контролювання дії окремих компонентів XAMPP control panel.

Розробка дистанційної платформи є складним і різноманітним завданням, що вимагає врахування різноманітних вимог. Важливо створити систему, яка є надійною та зручною для користувача та адаптується до змінюваних умов і вимог.

2 АНАЛІЗ ІСНУЮЧИХ ПЛАТФОРМ ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ

Зі зростанням кількості нових онлайн–навчальних ресурсів на ринку постійно з'являються нові типи інтерфейсів. Це може створити проблеми для користувачів. Основними недоліками цих інтерфейсів є складність для студентів і обмежені можливості, такі як відсутність можливості переглядати архівовані курси або завантажити результати пройдених тестів. Також вони не інтегровані з навчальними закладами. Крім того, багато платформ вимагають оплати за доступ до курсів, що може перешкоджати користувачам.

Однак, незважаючи на ці недоліки, існує низка платформ дистанційного навчання, які варто розглянути та вивчити як переваги, так і недоліки. Перш за все, необхідно виділити платформи, які мають простий інтерфейс, який легко зрозуміти для користувачів і робить їх зручними для навчання та викладання. Далі важливо врахувати можливості платформи щодо зберігання та доступу до навчальних матеріалів, а також можливостей для проведення тестів і оцінювання знань.

Важливо також оцінити рівень інтеграції платформи з іншими освітніми установами та послугами, оскільки це може допомогти їй використовуватися більш ефективно під час навчання. Фінансові питання, такі як доступність безкоштовних курсів і оплата за додаткові функціональні можливості, також повинні бути враховані.

Таким чином, вивчення різних онлайн–ресурсів дозволить краще зрозуміти їх переваги та недоліки, а також зробити більш обґрунтовані рішення щодо того, як їх найкраще використовувати під час навчання.

2.1 Платформа для навчання Moodle

(Modular Object–Oriented Dynamic Learning Environment)– це модульне об'єктно–орієнтоване динамічне навчальне середовище, яке також називають

системою управління навчанням (LMS), системою управління курсами (CMS), віртуальним навчальним середовищем (VLE) або просто платформою навчання, яка надає викладачам, студентам і адміністраторам надзвичайно інноваційний набір інструментів для комп'ютеризованого навчання, в тому числі дистанційного. Moodle можна використовувати для навчання студентів, школярів, підвищення кваліфікації та бізнес-навчання як в комп'ютерних класах, так і для самостійної роботи вдома.

Moodle – це система, яка є безкоштовною та відкритою для використання. Вона працює безкоштовно і не потребує жодного платного програмного забезпечення. Таким чином, будь-який навчальний заклад може створити безкоштовну, найкращу і абсолютно ліцензійну систему, не витративши жодної копійки на придбання програмного забезпечення. Таким чином він має можливість вносити зміни до коду відповідно до своїх потреб.

Moodle є найкращою системою для такого типу навчання в Україні та по всьому світу. На даний момент Moodle має приблизно 400 мільйонів користувачів у всьому світі та продовжує розвиватися значно швидше, ніж його конкуренти. У 2018 році відбулася важлива подія. Статистика показує, що Moodle має більше користувачів у всьому світі, ніж будь-яка інша платформа.

Moodle є стандартом для 2/3 вищої освіти в Європі. Moodle практично не має конкурентів в Україні, де платні платформи дуже обмежені. Покладатися на Moodle як на іноземну мову – це приблизно те саме, що використовувати англійську як іноземну мову в навчальному закладі [1]. Веб-додаток платформи для навчання Moodle наведено на рисунку 2.1.

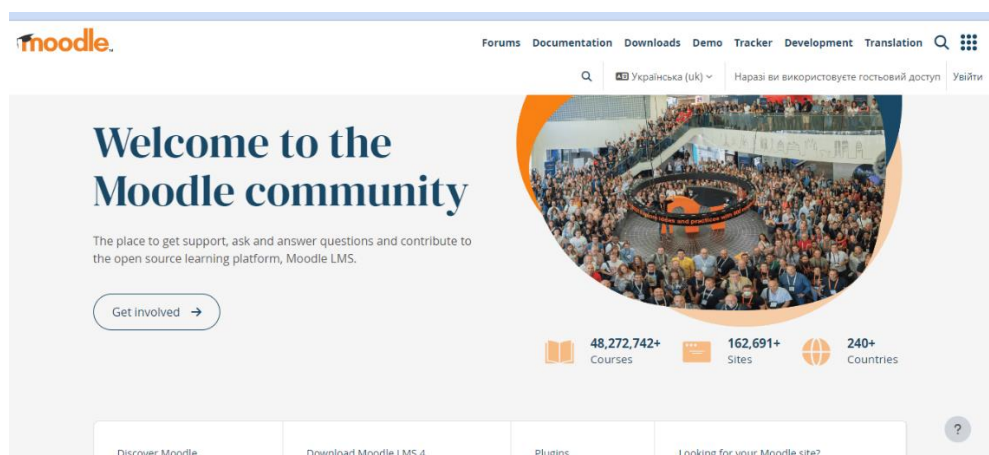


Рисунок 2.1 – Веб-додаток платформи для навчання Moodle

Переваги:

- Програмне забезпечення, яке є відкритим.
- Широкий спектр функцій для планування навчального процесу
- Велика спільнота розробників і користувачів

Недоліки:

- Інтерфейс для студентів складний.
- Мало можливостей для індивідуального спілкування між студентами
- Відсутність можливості завантажити пройдений тест
- Відсутність можливості переглянути архівовані курси

2.2 Сервіс для дистанційного навчання Google classroom

Google Classroom – це безкоштовна онлайн-платформа для дистанційного навчання, заснована ще в 2014 році. Однак він довго не користувався великою популярністю. За даними сайту Android Police, до березня цього року Classroom навіть не був включений до списку 100 найкращих освітніх додатків. Проте сьогодні, його функціонал виявився актуальним.

Classroom – це комбінація сервісів Google, таких як Google Disc, Google Docs та інші, розроблених спеціально для навчання. Для роботи з ним необхідний акаунт Google. Користування доступне як на комп'ютері, так і на мобільному пристрої.

У класі вчителі можуть створювати віртуальні класи та окремі курси за допомогою кодів, які присвоюються студентам. На сторінці кожного курсу вчитель може публікувати лекції, проводити опитування, проводити тести, тощо. Можна встановити термін виконання для кожного завдання. Крім того, викладач може спостерігати за тим, як студенти виконують завдання в режимі реального часу, бачити список робіт, які ще не завершені, і виставляти оцінки за кожною шкалою. Крім того, викладач публікує оголошення в стрічці класу, включаючи не лише текст, але й зображення або відео з YouTube, які використовуються для спілкування зі студентами в чаті. Студенти можуть побачити список завдань курсу, як виконаних, так і тих, які ще потрібно виконати. При цьому вчитель може дати

кожному студенту унікальне завдання [2]. Веб–додаток платформи для сервісу для дистанційного навчання Google classroom наведено на рисунку 2.2.

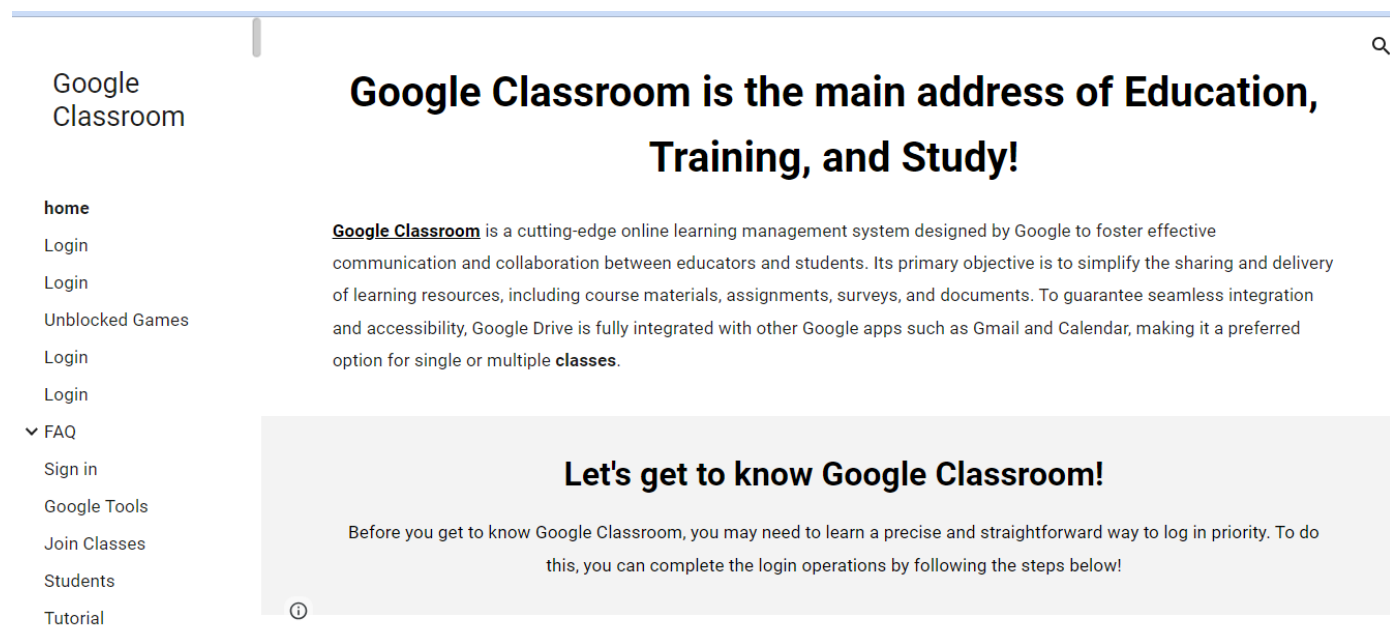


Рисунок 2.2 – Веб–додаток платформи для сервісу для дистанційного навчання Google classroom

Переваги:

- Зв’язок з іншими сервісами Google.
- Простий інтерфейс, який легко зрозуміти.
- Здатність створювати завдання та тести.

Недоліки:

- Недостатня продуктивність порівняно з іншими системами.
- Не завжди підходить для великих курсів або закладів вищої освіти.

2.3 Освітня міжнародна платформа Coursera

Освітня платформа Coursera відома як MOOC (масові відкриті онлайн–курси або відкриті курси онлайн від найкращих університетів світу). У 2012 році двоє професорів комп’ютерних наук Стенфордського університету заснували Coursera,

яка зараз є однією з найвідоміших ініціатив MOOC.

На платформі Coursera курси займають від шести до десяти тижнів. Вивчення включає консультації, завдання, обговорення на форумах, відеолекції та тести. Слухачі отримують сертифікати після завершення курсу. Навчання можна проводити як на мобільному пристрої, так і на персональному комп'ютері.

Під час карантину Coursera вирішила надати свої навчальні програми безкоштовно безробітним у різних країнах. Мета ініціативи полягає в тому, щоб допомогти працівникам, які працюють, розвинути додаткові знання та навички, які їм потрібні для працевлаштування.

З п'яти країн Україна була визнана такою. Служба зайнятості дозволила безкоштовно отримати доступ до 3800 онлайн-курсів із 400 спеціалізаціями, які пропонує міжнародна платформа Coursera. Безробітні отримують сертифікати за умови успішного завершення навчання. Зареєстрований безробітний повинен звернутися до свого кар'єрного радника, якщо він хоче отримати безкоштовну освіту.

Навчальні програми включають комп'ютерні технології, природничі, технічні, соціальні та гуманітарні науки, а також навчання іноземних мов. Крім того, доступні курси з охорони здоров'я, особистісного розвитку, побудови та розвитку бізнесу та ін [3]. Веб-додаток освітньої міжнародної платформи Coursera наведено на рисунку 2.3.

The image shows the Coursera Plus website interface. At the top, there are navigation links: "для фізичних осіб", "для компаній", "для університетів", and "для державних установ". The main header includes the Coursera logo, a "Вивчити" dropdown menu, a search bar with the text "Чого ви хочете навчитися?", and links for "Онлайн-ступеня", "Знайти кар'єру", "Увійти", and "Приєднатися безкоштовно".

The main content area features a large blue banner for "coursera PLUS". The text on the banner reads: "Необмежений доступ до більш ніж 7000 курсів світового рівня, практичних проектів та програм сертифікації за необхідними для роботи навичками в одній підписці 46 € на місяць, можна скасувати у будь-який час". Below this, there is a blue button that says "Почніть 7-денний безкоштовний пробний період" and text indicating "або 315 € /рік з гарантією повернення коштів протягом 14 днів". To the right of the text is a photograph of a smiling man with glasses and a watch, wearing a green shirt.

At the bottom of the banner, it says "Отримуйте знання від 325+ провідних університетів та компаній з Coursera Plus".

Рисунок 2.3 – Веб-додаток освітньої міжнародної платформи Coursera

Переваги:

- Зв'язок з іншими сервісами Google.
- Простий інтерфейс, який легко зрозуміти.
- Здатність створювати завдання та тести.

Недоліки:

- Недостатня продуктивність порівняно з іншими системами.
- Не завжди підходить для великих курсів або закладів вищої освіти.

Кожна з популярних платформ для організації дистанційного навчання, таких як Coursera, Moodle і Google Classroom, має свої особливості, переваги та недоліки. Вибір платформи залежить від цілей користувача, бюджету, можливостей техніки та освітніх цілей. Кожна з цих платформ пропонує унікальні можливості та, якщо їх правильно використовувати, може допомогти студентам досягти успіху в навчанні.

3 ЗАСОБИ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

У даному проєкті для розробки клієнтської частини платформи, тобто для створення веб-інтерфейсу користувача було використано такі мови програмування як: HTML, CSS, JavaScript. PHP використовувалось для розробки серверної частини платформи, обробки даних та взаємодії з базою даних. Для бази даних використовувався MySQL як система управління базою даних для зберігання інформації про користувачів, навчальний матеріал, результати тестувань тощо. Інтегрованим середовищем розробки (IDE) являється phpStorm. У даному проєкті використовується для розробки серверної частини платформи на мові PHP. Забезпечує зручне редагування коду, автодоповнення, відладку тощо. Середовищем для роботи з базою даних є MySQL Workbench. У даному проєкті використовується для проєктування баз даних, створення та редагування схеми бази даних, виконання SQL-запитів тощо. XAMPP Control Panel використовується для локального тестування та виконання веб-додатків на основі Apache, MySQL, PHP та інших сервісів.

3.1 Мова розмітки гіпертексту HTML

HTML було обрано для створення структури веб-сторінок та веб-додатків. HTML дозволяє визначати заголовки, параграфи, списки, таблиці тощо, що є основою будь-якого веб-документа.

HTML – це скорочення від «HyperText Mark-up Language», що означає «мова розмітки гіпертексту». Гіпертекст – це текст, який не послідовно пов'язаний з іншими документами, тобто можна перейти з першої сторінки документа на останню. Іншими словами, HTML є мовою розмітки та засобом зберігання даних. За допомогою HTML позначається текст, щоб веб-переглядач міг зрозуміти його так само, як на жорсткому диску. Інформація зберігається в блоках, кластерах,

секторах і доріжках, і комп'ютер розуміє, що треба, а що не треба зчитувати завдяки цій структурі [4].

Основною мовою для створення веб-сторінок є HTML, який використовується разом із CSS (Cascading Style Sheets) і JavaScript для створення інтерактивних веб-сайтів, які виконують різноманітні завдання. Використовуючи CSS для надання стилів і зовнішнього вигляду елементів HTML, JavaScript надає користувачам можливість постійно змінювати контент сторінки та взаємодіяти з нею.

HTML використовує просту синтаксичну структуру з тегами, атрибутами та значеннями. Коли використовуються символи "<" і ">", можна побачити вміст між ними на веб-сторінці. Наприклад, теги "p" вказують на параграф, "img" вставляють зображення, а "a" створюють посилання.

Основна ідея HTML полягає в тому, що вона створює структуру документа, яку веб-браузери можуть прочитати та відобразити користувачу. Він дозволяє розміщати текст, зображення, відео, звук і інші мультимедійні компоненти на веб-сторінці та гарантує, що вони відображаються належним чином [15].

3.2 Мова стилів CSS

CSS було обрано для опису зовнішнього вигляду HTML-документів. CSS забезпечує відокремлення структури документа від його представлення, надаючи гнучкість у дизайні та полегшуючи підтримку та оновлення стилів.

CSS (аббревіатура від Cascading Style Sheets, що в перекладі означає каскадні таблиці стилів) – це спеціальна мова (мова стилів), за допомогою якої описують вигляду документів (як і де відображати елементи веб-сторінки), написаних мовами розмітки даних. Найчастіше CSS використовується для документів, котрі розмічені мовою HTML, XHTML та XML.

За допомогою CSS можна відображати одну й ту ж ідею в різних стилях; зробити багато різних дизайнів сторінок для різних пристроїв (наприклад, коли ви друкете на екрані, який має велику ширину, меню не виводитиметься, однак на

смартфоні меню буде розташоване внизу, під вмістом); переносити правила відображення в інший CSS-файл дозволяє скоротити час завантаження сторінок сайту(в цьому випадку браузер завантажує лише структуру документа та дані на сторінці, а стильові правила даних завантажуються і кешуються лише один раз); простий процес подальшої модифікації дизайну– не потрібно змінювати кожен сторінку, достатньо просто змінити CSS-файл; дозволяє створювати складні та продумані методи дизайну [5].

Для того, щоб застосувати таблицю стилів до HTML-документу, можна обрати один з трьох способів, або комбінувати їх:

–застосувати зовнішні стилі (у вигляді окремого текстового .css – файлу) за допомогою елемента link;

–вбудувати стилі безпосередньо в HTML-документ (у вигляді блоку css-тексту) за допомогою елемента style;

–застосувати inline-стиль , тобто призначити стиль конкретному HTML-елементу безпосередньо в документі, за допомогою HTML-атрибуту style [16].

3.3 Високорівнева скриптова мова JavaScript

JavaScript було обрано для додавання інтерактивності до веб-сторінок через те що вона дозволяє здійснювати динамічні зміни на сторінці без перезавантаження, що покращує користувацький досвід.

JavaScript – це високорівнева скриптова мова, яка широко використовується для створення інтерактивних програм і сайтів. JS дозволяє додавати анімацію та спливаючі повідомлення, валідувати форми, оновлювати контент і змінювати інтерфейс відповідно до дій користувача. Наприклад, коли наводиться курсор на кнопку, вона змінює колір або зникає, коли користувач натискає на неї. Сторінка є функціональною завдяки використанню мови JavaScript. У неї є певні характеристики, наприклад: інтерпретація, підключення до HTML і CSS, динамічна типізація, багато бібліотек і фреймворків JavaScript, кросплатформеність, імперативність, функціональності та ефективності ООП,

безперервність, вбудована підтримка DOM (Document Object Model), велика громада, відкриті коди [6].

Наприклад, JavaScript може:

–Додавати новий HTML-код на сторінку, змінювати наявний вміст, змінювати стилі.

–Реагувати на дії користувача, опрацьовувати натискання миші, переміщення вказівника, натискання на клавіші клавіатури.

–Надсилати запити мережею до віддалених серверів, надсилати файли (так звані технології AJAX і COMET).

–Отримувати та надсилати куки, ставити запитання відвідувачам, показувати повідомлення.

–Запам'ятовувати дані на стороні клієнта (“local storage”), які будуть доступні в майбутніх сесіях на цьому веб-сайті.

Можливості JavaScript у браузері обмежені для безпеки користувача. Мета полягає в тому, щоб небезпечні вебсторінки не мали доступу до приватної інформації та не могли пошкодити інформацію на комп'ютері користувача. Приклади таких обмежень:

–JavaScript на вебсторінці не може читати/записувати довільні файли на жорсткому диску, копіювати їх чи виконувати програми. Скрипт не має прямого доступу до функцій ОС. Сучасні браузери дозволяють працювати з файлами, але доступ до них обмежений і надається тільки тоді, коли користувач виконав відповідні дії, наприклад, перетягнув файл у вікно браузера чи вибрав його через тег `<input>`. Є шляхи взаємодії з камерою/мікрофоном та іншими пристроями, але для цього потрібен явний дозвіл користувача. Тому сторінка з JavaScript не може нишком увімкнути вебкамеру, спостерігати за оточенням і надсилати інформацію до СБУ.

–Різні вкладки/вікна зазвичай не знають одне про одного. Іноді це можливо, наприклад, коли одне вікно використовує JavaScript, щоб відкрити інше. Але навіть у такому випадку JavaScript з однієї сторінки не має доступу до іншої, якщо вони з різних сайтів (мають різні домени, протоколи чи порти). Це називається “Політикою того ж походження (Same Origin Policy)”. Щоб обійти це

обмеження, обидві сторінки мають погодитися на обмін даними та містити спеціальний JavaScript-код, який здійснюватиме це. Ми розглянемо цю тему в посібнику. Знову-таки, це обмеження існує задля безпеки користувача. Сторінка за адресою `http://anysite.com`, яку відкрив користувач, не повинна мати доступ до іншої вкладки браузера з URL-адресою `http://gmail.com` і викрадати звідти інформацію.

–JavaScript може легко спілкуватися мережею з сервером, від якого отримана поточна сторінка. Але здатність скрипту отримувати дані з інших сайтів/доменів обмежена. Такі запити можливі, але потребують спеціальної згоди (вираженої в HTTP-заголовках) від віддаленого сервера. Це також зроблено задля безпеки [17].

3.4 Скриптова мова PHP

Для розробки серверної частини платформи було обрано PHP через те що він підтримується багатьма хостинг-провайдерами і має велику спільноту розробників, легко інтегрується з HTML і має простий синтаксис, що прискорює процес розробки та має вбудовані функції для взаємодії з різними базами даних, включаючи MySQL.

PHP (англ. PHP: Hypertext Preprocessor – "PHP: препроцесор гіпертексту") – це скриптова мова програмування, яка використовується для розробки програм і сайтів. Майже кожна хостинг-компанія зараз використовує цю мову програмування для створення динамічних веб-сторінок. Саме PHP відповідає за оновлення сайту. Ця мова підтримує сценарії, які є спеціалізованими програмами на стороні сервера, які запускаються веб-браузером користувача у відповідь на запит. Таким чином, PHP – це інтерпретатор сценаріїв, а сам процес – це інтерпретація. PHP швидко створює професійні програми та сайти. PHP постійно оновлюється, і незважаючи на наявність і популярність інших мов, він продовжує бути найпопулярнішим у списку затребуваних. Він простий у використанні та пропонує багато переваг веб-розробникам [7].

Популярність PHP обумовлена його такими перевагами:

–Простий та інтуїтивно зрозумілий синтаксис. PHP швидко освоюють навіть програмісти–новачки. Він увібрав усі найкращі особливості таких популярних мов, як C, Java та Perl. PHP–код легко читається незалежно від способу використання (для написання невеликих скриптів або створення потужних додатків з використанням об'єктно–орієнтованого підходу до реалізації програми).

–Кросплатформність та гнучкість. PHP сумісний із усіма популярними платформами (Linux, Windows, MacOS). Написані на ньому програми успішно працюють на різному серверному ПЗ (IIS, Nginx, Apache та багатьох інших).

–Відмінна масштабованість. PHP дозволяє досягти максимальної продуктивності додатків, написаних на ньому, зі зростанням апаратних ресурсів. Веб–програми, розподілені на кілька серверів, здатні справлятися із суттєвими навантаженнями (великим трафіком).

–Вбудовуваність у HTML–документи. На просту HTML–сторінку можна легко додати контент, що динамічно змінюється шляхом вставки блоків коду PHP. Вони додаються подібно до HTML–тегів, не порушують структуру документа.

–Активний розвиток та вдосконалення. Спільнота розробників постійно працює над впровадженням додаткового функціоналу, що розширює можливості мови, спрощенням синтаксису та покращенням захисту від можливих атак.

–Детальна документація. На офіційному сайті проекту представлені повні відомості про кожну функціональну одиницю мови з прикладами використання.

–Простий пошук рішень проблем, що виникають. В інтернеті існує безліч форумів, присвячених програмуванню на PHP.

–Широкі перспективи її подальшого розвитку. Більшість CMS були створені на чистому PHP та фреймворках. Цим зумовлені популярність та затребуваність PHP програмістів [18].

3.5 СКБД MySQL

Для управління базою даних було обрано MySQL, яка є популярною системою управління базами даних (СУБД) через те що MySQL забезпечує високу

швидкість обробки запитів і є надійною системою для зберігання великих обсягів даних, є відкритим програмним забезпеченням, що робить її доступною для використання без додаткових витрат та підтримується багатьма мовами програмування та платформами, що робить її універсальною у використанні.

MySQL – вільна система керування реляційними базами даних, розроблена для підвищення швидкості обробки великих баз даних. Система керування базами даних (СКБД) з відкритим кодом була розроблена для заміни комерційних систем. Оскільки вона чудово працює з багатьма мовами програмування, вона в основному використовується для створення динамічних веб-сторінок. Основні характеристики: MySQL – це система керування базами даних: реляційна база збільшує швидкість і гнучкість, оскільки дані зберігаються в різних таблицях; програмне забезпечення MySQL – це програмне забезпечення програмування з відкритим кодом; технічні можливості СУБД MySQL: ПО MySQL є системою клієнт-сервер, яка містить багато-SQL-сервер і підтримує різні обчислювальні машини баз даних, а також пропонує різноманітні клієнтські програми та бібліотеки, засоби адміністрування та різноманітні програмні інтерфейси (API); безпека: система безпеки забезпечує гнучкість і безпеку завдяки використанню привілеїв і паролів для верифікації з віддаленого комп'ютера, паролі, які передаються через мережу, коли вони з'єднуються з сервером, шифруються; розмір даних: з версії MySQL 3.23, яка використовує новий тип таблиць, розмір таблиці може досягати 8 мільйонів терабайт, або 2⁶³ байти [8].

3.6 Середовище розробки phpStorm

Інтегроване середовище розробки phpStorm було обрано для розробки серверної частини платформи на мові PHP через те що він забезпечує зручне редагування коду, автодоповнення, відладку та інші корисні функції, що підвищують продуктивність розробки.

PhpStorm – це інструмент для веб-розробки, призначений для створення веб-додатків і інших типів програм. Рішення phpStorm дозволяє розгортати та

синхронізувати проекти за допомогою протоколу FTP. Автоматичне завершення мовних конструкцій PHP в коді, перевірка коду, кілька алгоритмів рефакторингу та швидка навігація по коду – це функції, які пропонує середовище PhpStorm. Умовні точки зупину, відстеження цінностей і автоматизоване налагодження процедур є функціями, які підтримує графічний налагоджувач PHP. Для тестування програм підтримується модуль тестування PHPUnit, а також графічний інтерфейс запуску тестів. Редагування коду покращує синтаксисні конструкції, надає розширене форматування конфігурації, використовує виявлення помилок у реальному часі та завершує код. PhpStorm–редактор автоматично вирішує проблеми, враховуючи коментарі до коду, коли він завершується. PhpStorm дозволяє візуалізувати код ієрархічно та швидко навігувати по всіх елементах [9].

3.7 Середовище розробки MySQL Workbench

Середовище для роботи з базою даних MySQL Workbench було обрано для проектування баз даних, створення та редагування схем бази даних, виконання SQL–запитів та іншої роботи з даними.

MySQL Workbench– це уніфікований візуальний інструмент для архітекторів баз даних, розробників і адміністраторів баз даних. MySQL Workbench забезпечує моделювання даних, розробку SQL і комплексні засоби адміністрування для налаштування сервера, адміністрування користувачів, резервного копіювання та багато іншого. MySQL Workbench доступний у Windows, Linux і Mac OS X. MySQL Workbench дозволяє адміністратору бази даних, розробнику або архітектору даних візуально проектувати, моделювати, створювати та керувати базами даних. Він містить усе, що потрібно розробнику даних для створення складних моделей ER, прямого та зворотного проектування, а також надає ключові функції для виконання складних завдань із керування змінами та документування, які зазвичай потребують багато часу та зусиль. MySQL Workbench надає візуальні інструменти для створення, виконання та оптимізації запитів SQL. Редактор SQL забезпечує підсвічування синтаксису кольором, автозаповнення, повторне використання

фрагментів SQL і історію виконання SQL. Розробники та адміністратори баз даних можуть використовувати візуальні інструменти для налаштування серверів, адміністрування користувачів, резервного копіювання та відновлення, перевірки даних аудиту та перегляду стану бази даних [10].

3.8 Інструмент керування XAMPP Control Panel

Програмний пакет XAMPP Control Panel було обрано для локального тестування та виконання веб-додатків на основі Apache, MySQL, PHP та інших сервісів. XAMPP забезпечує легкий спосіб налаштування локального серверного середовища, що дозволяє швидко тестувати та відлагоджувати веб-додаток.

XAMPP Control Panel – це інструмент керування, який пропонує контролювати дії окремих компонентів XAMPP. Він є безкоштовним пакетом, який поєднує в собі кілька програм і інструментів, необхідних для запуску та роботи веб-сервера на локальному комп'ютері. Він керує кожним компонентом текстового сервера. Користувач може ініціювати або зупиняти окремі модулі, натискаючи кнопки під колонкою «Дії». Панелі керування ефективно керують усіма компонентами пакета XAMPP.

Основною функцією XAMPP є можливість створювати локальний сервер. Це дозволяє дизайнерам і розробникам тестувати та перевіряти свої веб-проекти без необхідності завантажувати їх на віддалений сервер. Оскільки всі зміни та виправлення можна вносити безпосередньо на локальному сервері, це значно спрощує та прискорює процес розробки [19].

За допомогою панелі керування можна визначити, чи працюють Apache, MySQL, Mercury тощо. Середовище розробки можна використовувати, лише коли принаймні Apache перебуває у запущеному стані [11].

Ці засоби дозволили ефективно розробляти, тестувати та впроваджувати платформу дистанційного навчання за допомогою відкритих та широко використовуваних технологій.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Основною ціллю розробки є створення дистанційної платформи за предметами кафедри, який надалі може бути використаним як стандартна дистанційна платформа для навчання в різних закладах освіти. Також, було створено веб– додаток для демонстрації взаємодії користувача з ним.

4.1 Реалізовані функції

Система реалізує онлайн платформу для навчання, яка дозволяє користувачам пройти курси з відповідного навчального предмету. Основні можливості системи включають:

–Автентифікація користувача.

Користувачі можуть авторизуватися в системі з використанням наданого електронної адреси та пароля.

–Навчальні матеріали.

Кожен курс містить навчальні матеріали, такі як текстові матеріали, лекції, презентації та навчальні посібники.

–Лабораторні роботи.

Деякі курси можуть включати в себе виконання лабораторних робіт. Система надає можливість завантаження завдань, завантаження та видалення файлів для лабораторних робіт.

–Тести та контрольні роботи.

Користувачі можуть проходити тести з відповідного курсу. Тести складаються з питань та варіантів відповідей. Після проходження тесту, користувач може побачити оцінку за пройдений тест та завантажити документ пройденого тесту для само–аналізу. В завантаженому документі користувач побачить питання, свої відповіді та правильні відповіді, у випадку якщо користувач відповів на запитання не вірно.

–Оцінки.

У розділі «Оцінки» користувач зможе переглянути усі свої оцінки, а також за пошуком знайти оцінки для конкретного курсу.

–Новини та календарні події.

Система надає можливість перегляду оголошень та новин щодо курсів та інших подій, а також календар подій, в якому наявний календарний графік навчального процесу.

–Керування користувачами та курсами.

Адміністратори системи можуть додавати та видаляти курси, користувачів, події, лабораторні роботи, навчальні матеріали та контролювати доступ користувачів до курсів.

–Інтерактивний інтерфейс.

Система має зручний інтерфейс користувача, який дозволяє легко орієнтуватися та використовувати всі її функції.

–Зручне зберігання та відображення даних.

Дані про користувачів, курси, навчальні матеріали, результати тестів та лабораторних робіт зберігаються в базі даних і легко доступні для аналізу та використання.

4.2 UML– діаграми

Нижче наведені діаграми є важливими інструментами в розробці програмного забезпечення та баз даних. Кожна з них має свою специфічну роль і допомагає в різних аспектах проектування системи.

Діаграма Use Case (діаграма прецедентів використання) – це графічне виявлення функціональності системи чи програми, яка показує, як різні актори (користувачі) взаємодіють із системою для досягнення певних цілей. Вона є одним із ключових інструментів у розробці програмного забезпечення, що дозволяє описати сценарії використання системи з точки зору кінцевих користувачів.

Діаграма прецедентів використання включає учасників, прецедентів та

зв'язки між ними. Актори є ентитетами, які взаємодіють із системою; це можуть бути окремі користувачі або зовнішні системи та інші частини. Прецеденти є конкретними функціональними можливостями системи, які актори можуть реалізувати. Актори та прецеденти пов'язані, щоб показати, які прецеденти вони можуть виконати.

Розробка та замовлення діаграми Use Case дозволяє розробникам програмного забезпечення візуально визначити та зрозуміти функції системи, її можливості та потреби користувачів. Крім того, вона використовує сценарії використання системи для визначення важливих акторів і прецедентів. Крім того, вона виявить проблеми та недоліки в роботі системи [12]. Діаграма прецедентів наведена на рисунку 4.1.

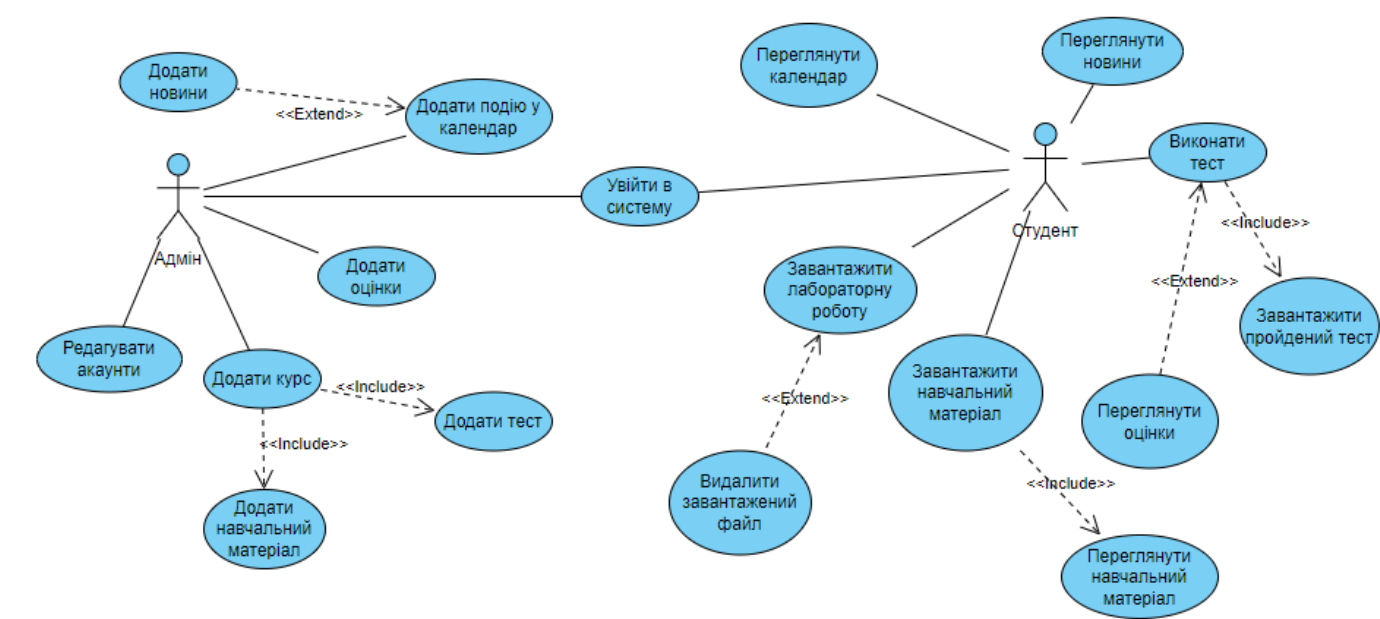


Рисунок 4.1 – Діаграма прецедентів

Діаграма класів є типом структурної діаграми, які використовуються для моделювання об'єктно-орієнтованих систем. Вони показують класи, їх атрибути, методи та зв'язки між класами. Діаграми класів дозволяють розібрати систему на окремі компоненти і зрозуміти, як вони взаємодіють між собою.

Класи діаграм допомагають розділити складну систему на окремі частини та розробити, як взаємозв'язані їхні частини. Це дозволяє розробникам отримати

загальне розуміння того, як побудована система.

Для проектування об'єктно–орієнтованих систем використовують діаграми класів. Перед початком фази реалізації вони допомагають вирішити проблему класу, їх характеристики та методи.

Діаграми класів допомагають усвідомити, як працює система, і виявити певні проблеми або невідповідності в її структурі [13]. Діаграма класів наведена на рисунку 4.2.

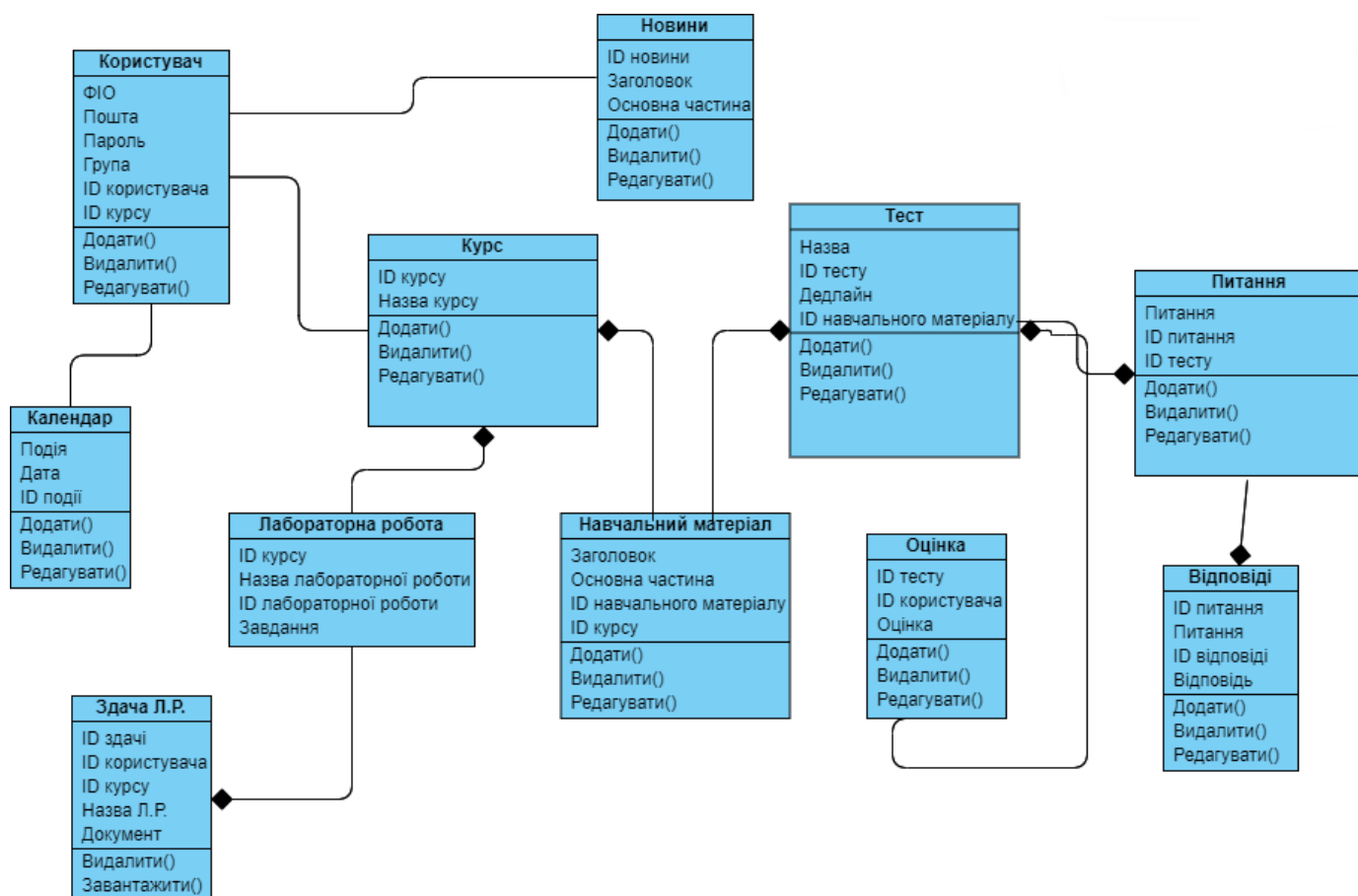


Рисунок 4.2 – Діаграма класів

Розширена діаграма сутності–зв'язку (EER) є важливою частиною інтерфейсу моделювання в MySQL Workbench. Діаграма EER надає візуальне представлення зв'язків між таблицями у моделі. Зміни, внесені за допомогою редактора моделей, показано на відповідній діаграмі. Подібним чином зміни, які вносяться в діаграму, легко реєструються в редакторі моделей. MySQL Workbench застосовує стандартні концепції та символи для відображення зв'язків між

таблицями [14]. Розширена діаграма сутності та зв'язку наведена на рисунку 4.3

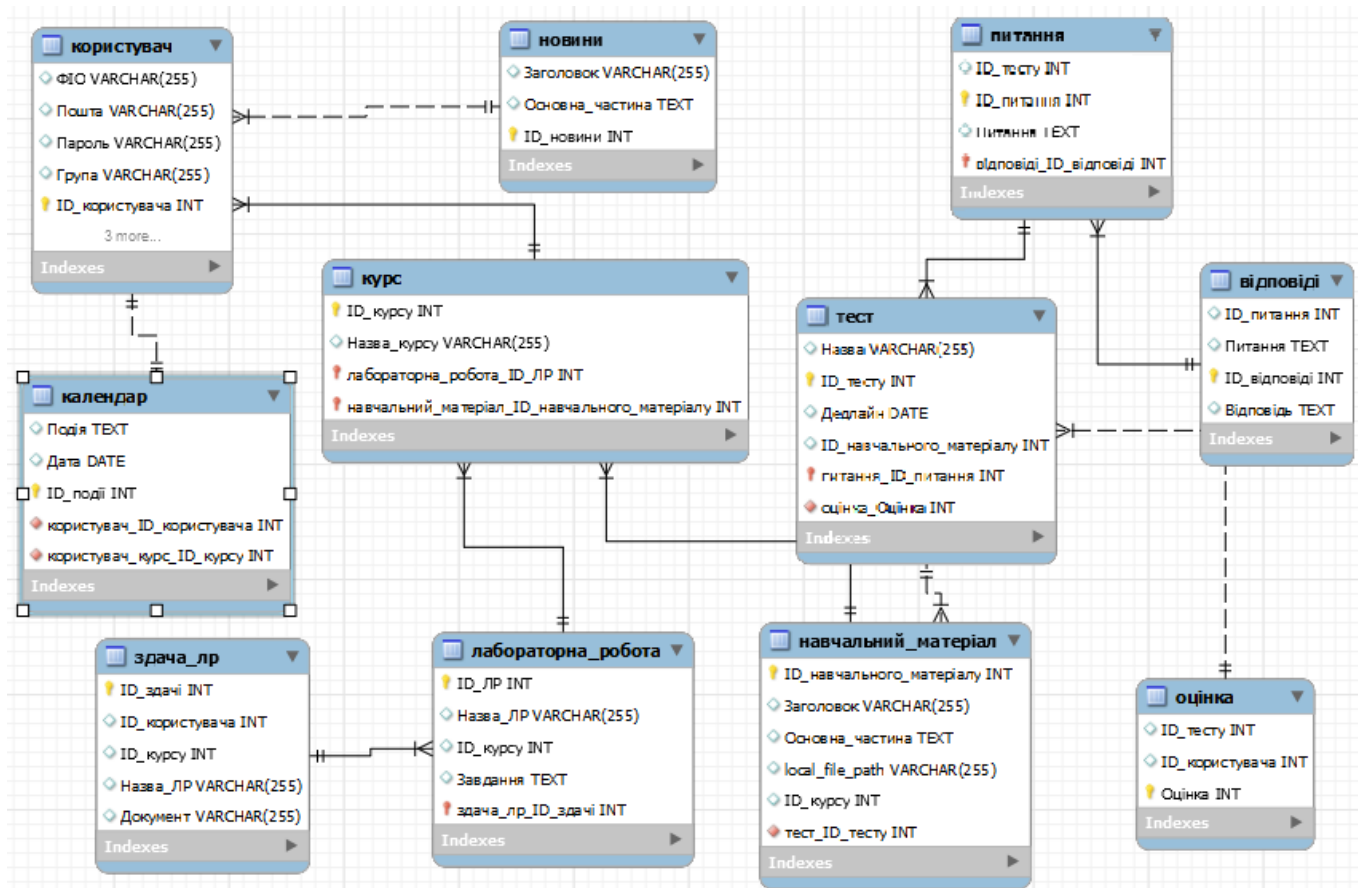


Рисунок 4.3 – Розширена діаграма сутності та зв'язку (EER)

4.3 Таблиці баз даних

Основні таблиці бази даних включають «відповіді», «здача_лр», «лабораторна_робота», «питання», «оцінка», «тест», «навчальний_матеріал», «курс», «новини», «користувач» та «календар».

- Таблиця «відповіді» зберігає варіанти відповідей на питання тестів.
- Таблиця «здача_лр» зберігає файли та іншу інформацію про здачу лабораторних робіт.
- Таблиця «лабораторна_робота» містить інформацію про лабораторні роботи.
- Таблиця «питання» зберігає питання для тестів.
- Таблиця «оцінка» містить оцінки, отримані користувачами за тести.

- Таблиця «тест» містить інформацію про тести.
 - Таблиця «навчальний_матеріал» зберігає лекції, презентації, навчальні посібники та інший навчальний матеріал для навчання.
 - Таблиця «курс» визначає належність відповідного користувача до курсу та містить інформацію про курс.
 - Таблиця «новини» містить інформацію про новини.
 - Таблиця «користувач» зберігає користувачів системи та їх дані.
 - Таблиця «календар» відстежує календарні події для користувачів системи.
- Ця структура бази даних дозволяє взаємодіяти з курсами, лабораторними роботами, тестами, навчальними матеріалами, новинами та календарними подіями для навчальної програми. Були використані зовнішні ключі для забезпечення цілісності даних та зв'язків між таблицями.

4.4 Зв'язки між таблицями баз даних

Зв'язки між таблицями в системі виглядають наступним чином:

- Таблиця «відповіді»: зв'язана з таблицею «питання» через зовнішній ключ «ID_питання».
- Таблиця «здача_лр»: зв'язана з таблицею «лабораторна_робота» через зовнішній ключ «ID_ЛР».
- Таблиця «лабораторна_робота»: зв'язана з таблицею «курс» через зовнішній ключ «ID_курсу».
- Зв'язана з таблицею «здача_лр» через зовнішній ключ «ID_ЛР».
- Таблиця «питання»: зв'язана з таблицею «тест» через зовнішній ключ «ID_тесту».
- Зв'язана з таблицею «відповіді» через зовнішній ключ «ID_питання».
- Таблиця «оцінка»: зв'язана з таблицею «тест» через зовнішній ключ «ID_тесту».
- Таблиця «тест»: зв'язана з таблицею «питання» через зовнішній ключ «ID_тесту».

Зв'язана з таблицею «оцінка» через зовнішній ключ «ID_тесту».

Зв'язана з таблицею «навчальний_матеріал» через зовнішній ключ «ID_навчального_матеріалу».

–Таблиця «навчальний_матеріал»: зв'язана з таблицею «курс» через зовнішній ключ «ID_курсу».

Зв'язана з таблицею «тест» через зовнішній ключ «ID_навчального_матеріалу».

–Таблиця «курс»: зв'язана з таблицею «лабораторна_робота» через зовнішній ключ «ID_курсу».

Зв'язана з таблицею «навчальний_матеріал» через зовнішній ключ «ID_курсу».

Зв'язана з таблицею «користувач» через зовнішній ключ «ID_курсу».

–Таблиця «новини»: зв'язана з таблицею «користувач» через зовнішній ключ «новини_ID_новини».

–Таблиця «користувач»: зв'язана з таблицею «курс» через зовнішній ключ «ID_курсу».

Зв'язана з таблицею «новини» через зовнішній ключ «новини_ID_новини».

Зв'язана з таблицею «календар» через зовнішній ключ «ID_події».

–Таблиця «календар»: зв'язана з таблицею «користувач» через зовнішній ключ «ID_події».

5 МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ІНФОРМАЦІЙНОЮ СИСТЕМОЮ

З метою використання розробленої системи дистанційної платформи для навчання за предметами кафедри було розроблено зручний інтуїтивно зрозумілий інтерфейс користувача. Графічний інтерфейс є стандартним і використовується для базової взаємодії користувача з програмним забезпеченням.

5.1 Системні вимоги та встановлення інформаційної системи

Для забезпечення стабільної роботи створеної програмної системи необхідно дотримуватися рекомендацій щодо її використання, а також дотримуватися вимог її встановлення. Для коректної роботи веб-додатку системи необхідно враховувати потужність комп'ютера, на якому воно буде працювати, відповідати вимогам до версії операційної системи.

5.1.1 Вимоги до комп'ютера

1. Операційна система:

- Windows 10 або новіша версія
- macOS 10.13 (High Sierra) або новіша версія
- Linux (наприклад, Ubuntu 18.04 або новіша версія)

2. Місце на диску:

- Мінімум 10 ГБ SSD, рекомендовано 20 ГБ або більше.
- SSD диски рекомендуються для швидшого доступу до файлів і швидшого завантаження системи.

3. Додаткові пристрої:

- Монітор з роздільною здатністю 1080p або вищою.
- Стабільне інтернет-з'єднання для завантаження програмного забезпечення

та бібліотек.

5.1.2 Вимоги до програмного забезпечення

1. HTML, CSS, JavaScript:

– Підтримуються будь-яким сучасним текстовим редактором або IDE (наприклад, PhpStorm).

2. PHP:

– PHP 7.4 або новіша версія (рекомендовано PHP 8.x для останніх функцій і покращень безпеки).

3. MySQL:

– MySQL 5.7 або новіша версія (рекомендовано MySQL 8.x).

4. PhpStorm:

– Остання версія PhpStorm (JetBrains). Це комерційний продукт, тому потрібна ліцензія або підписка.

5. XAMPP:

– Остання версія XAMPP, яка включає Apache, MariaDB/MySQL, PHP та Perl.

6. MySQL Workbench:

– Остання версія MySQL Workbench для моделювання баз даних і управління MySQL сервером.

7. Веб-браузери:

– Сучасні веб-браузери для тестування веб-сторінок, наприклад, Google Chrome, Mozilla Firefox, Microsoft Edge.

5.2 Взаємодія користувача з платформою дистанційного навчання

Важливо забезпечити, щоб усі ці компоненти були налаштовані правильно та працювали коректно, що дозволить ефективно використовувати платформу дистанційного навчання. Для інсталяції системи, додатково потрібно, за

допомогою пошти, завантажити архів файлу “diploma.rar”.

Після завантаження усіх необхідних файлів, користувачу потрібно відкрити “XAMPP Control Panel” та натиснути в стовпці Actions кнопку Start навпроти модуля Apache. Після виконання всього вище зазначеного XAMPP Control Panel повинен виглядати наступним чином. XAMPP Control Panel наведено на рисунку 5.1.

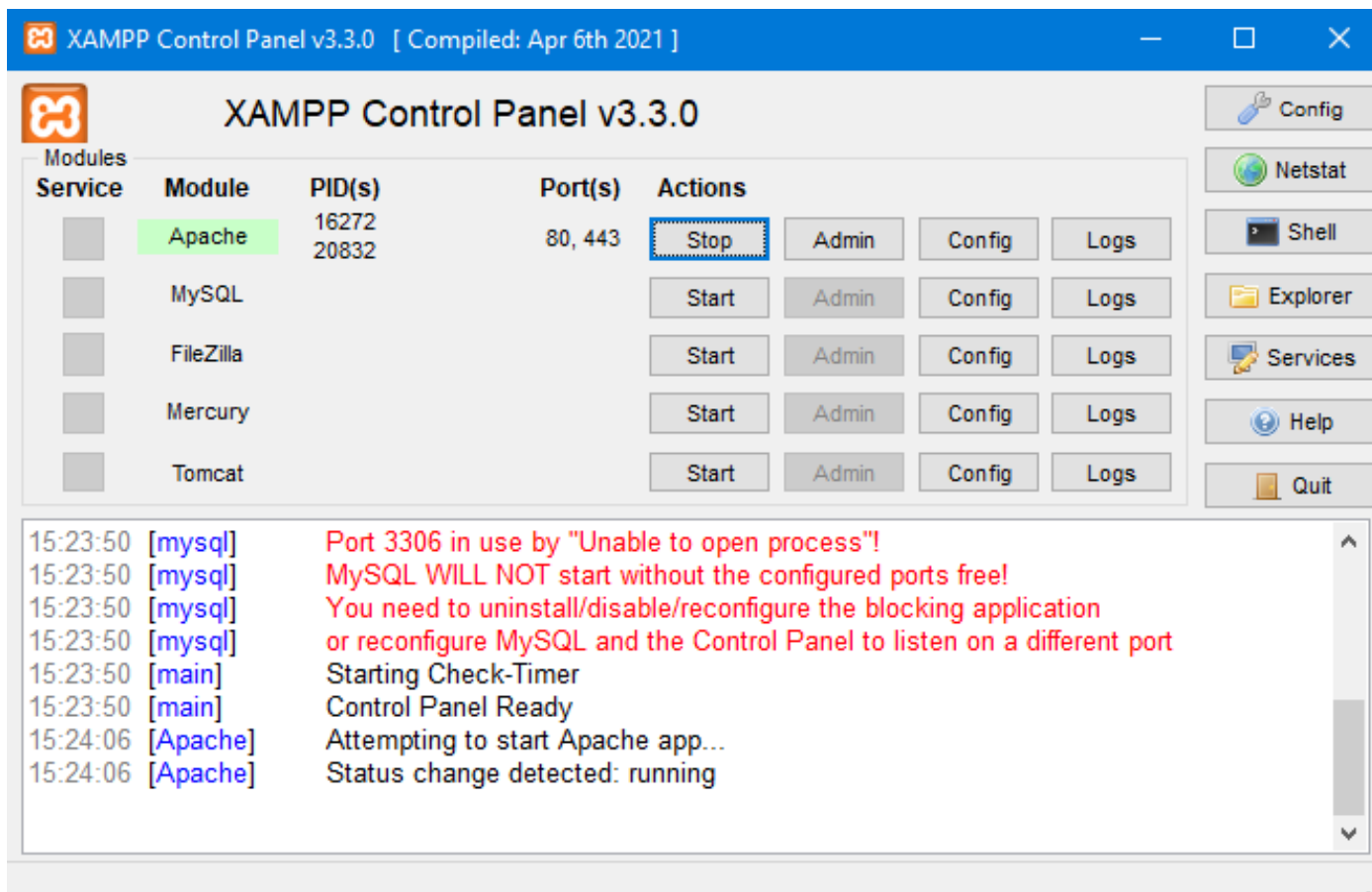


Рисунок 5.1 – XAMPP Control Panel

Далі потрібно перейти до веб-браузеру Google Chrome та в пошуковій строчці ввести: “http://localhost/diploma/login.php”, що приведе користувача на сторінку входу, приклад якого показано нижче. Сторінка входу наведена на рисунку 5.2.

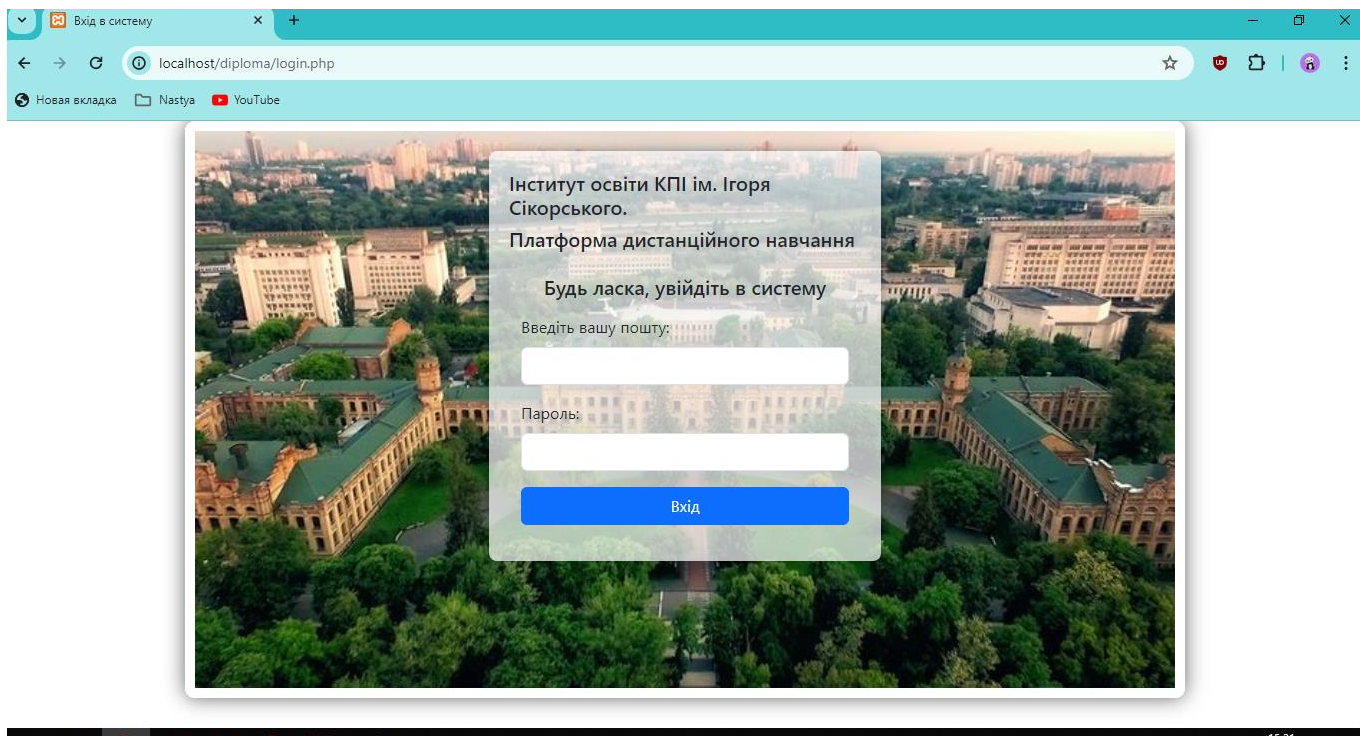


Рисунок 5.2 – Сторінка входу

Користувач вводить пошту та пароль, які надають адміністратори курсів та після успішного входу в систему користувач побачить особистий кабінет. Приклад особистого кабінету наведено на рисунку 5.3.

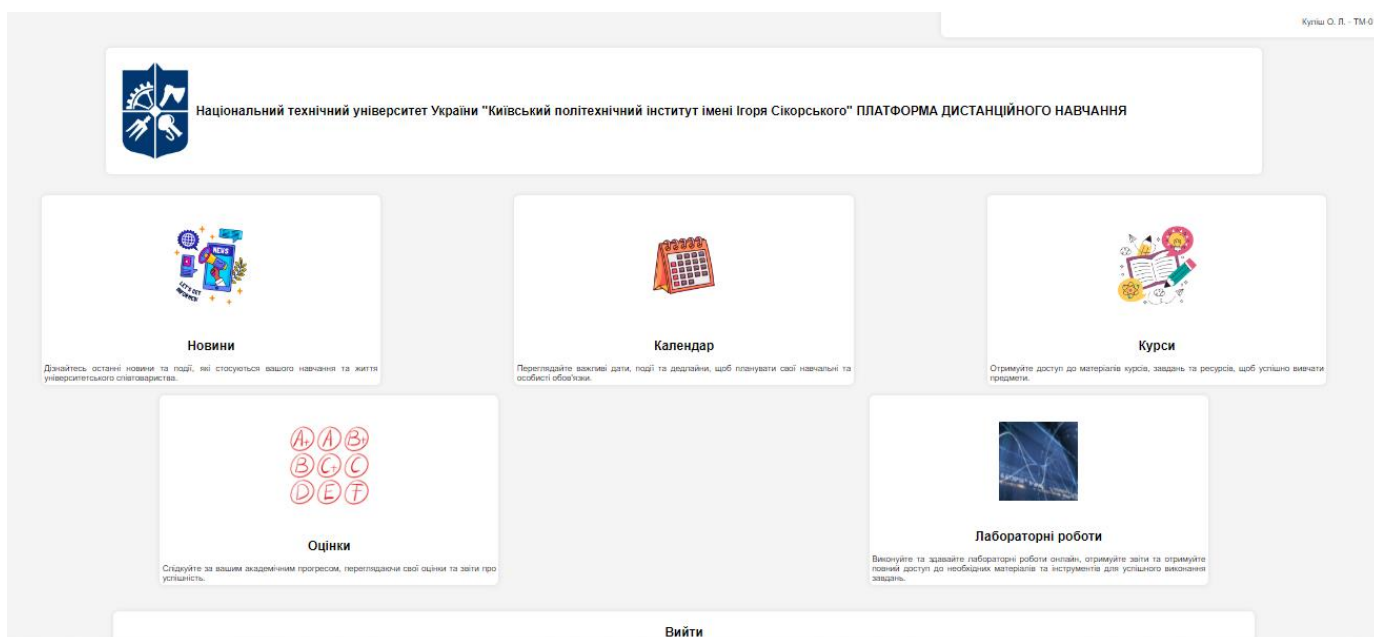


Рисунок 5.3 – Особистий кабінет

Відкривши вкладку новини, користувач побачить усі новини, які стосуються кафедри та університету. Приклад сторінки «Новини» наведено на рисунку 5.4.



Рисунок 5.4 – Сторінка Новини

Відкривши вкладку календар, користувач побачить графік навчального процесу, ключові дати першого та другого семестрів, важливі події та календар з ключовими датами першого та другого семестрів. Приклад вкладки календар наведено на рисунку 5.5.

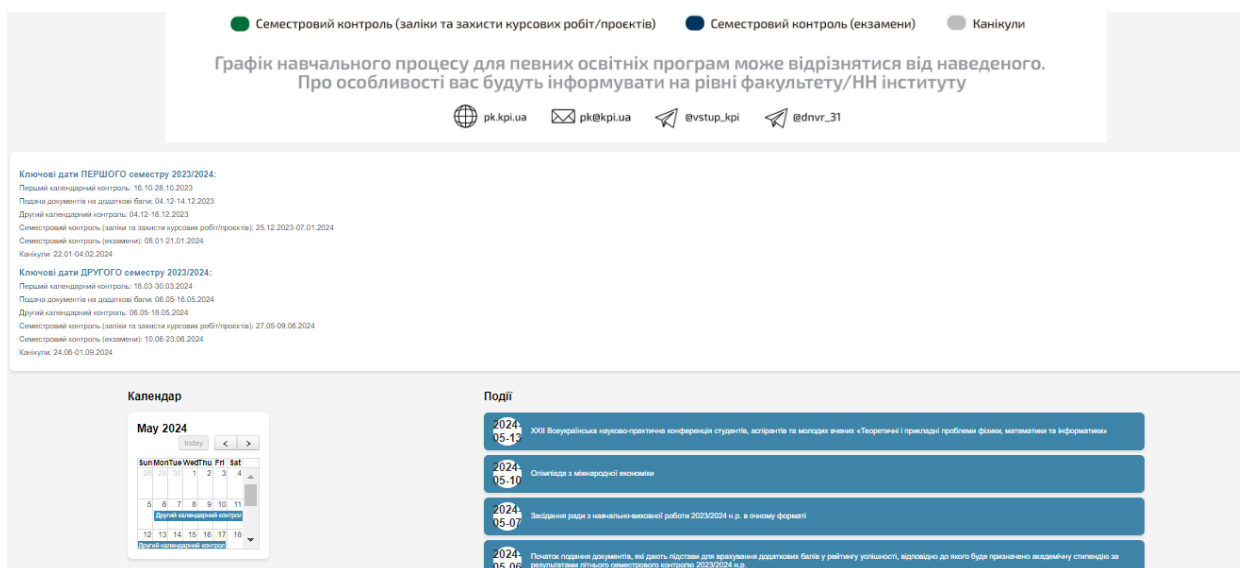


Рисунок 5.5 – Вкладка Календар

Відкривши вкладку Курси, користувач побачить перелік курсів, тестів, контрольних робіт та лекцій. Приклад вкладки, де зображено перелік курсів, тестів, контрольних робіт та лекцій наведено на рисунку 5.6.

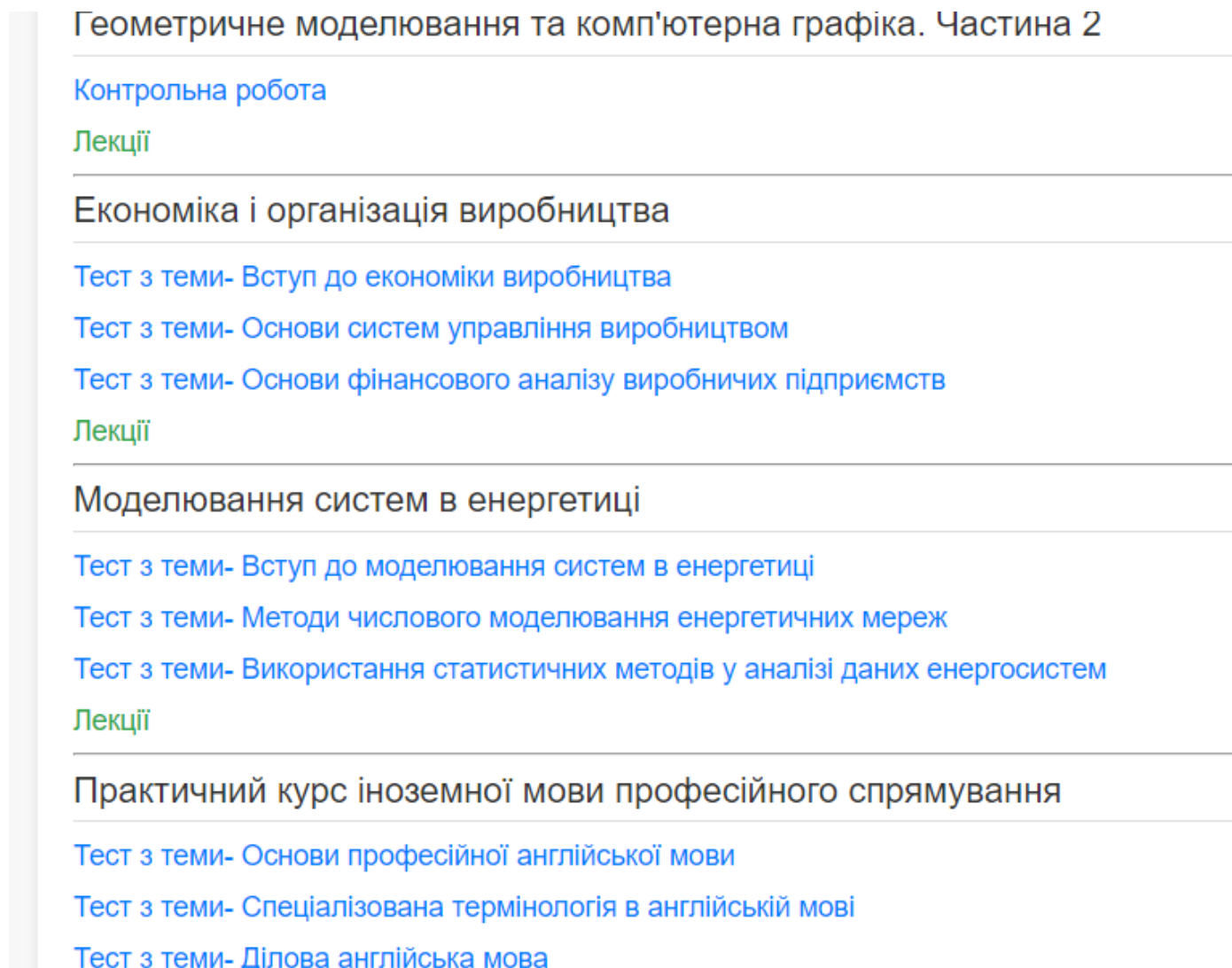


Рисунок 5.6 – Вкладка Курси

Перейшовши за посиланням лекцій, користувач побачить назви лекцій, та натиснувши на назву лекції один раз, користувач побачить текст лекції або документ, який потрібно завантажити, щоб переглянути лекцію. Приклад вкладки лекції наведено на рисунку 5.7.

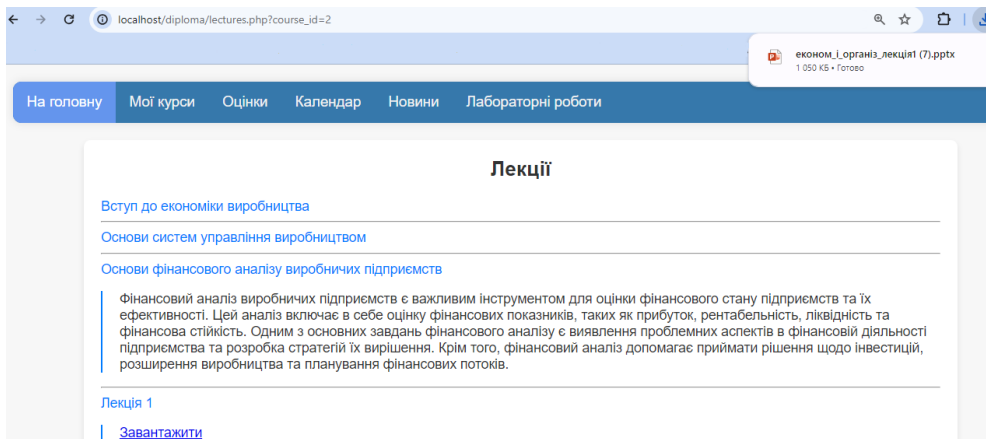


Рисунок 5.7 – Вкладка Лекції

Відкривши вкладку Курси, якщо користувач натисне на контрольну роботу або тест, то він перейде за посиланням, де побачить відповідні запитання та відповіді для контрольної роботи або тесту. Обравши відповіді на кожне питання та натиснувши кнопку завершити тест, користувач побачить свою оцінку, можливість завантажити пройдений тест та кнопку повернутись до курсів. В завантаженому файлі, користувач побачить питання, свою відповідь та правильну відповідь, у разі якщо користувач відповів на питання не вірно.

Приклад вкладок, де зображено контрольну роботу наведено на рисунку 5.8, вкладку після завершення контрольної роботи наведено на рисунку 5.9 та завантажений файл з тестом наведено на рисунку 5.10.

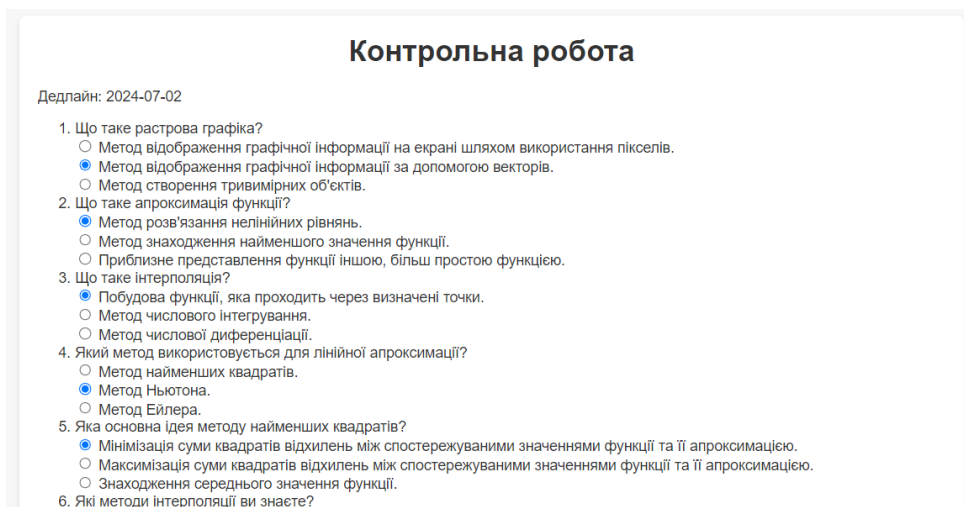


Рисунок 5.8 – Контрольна робота

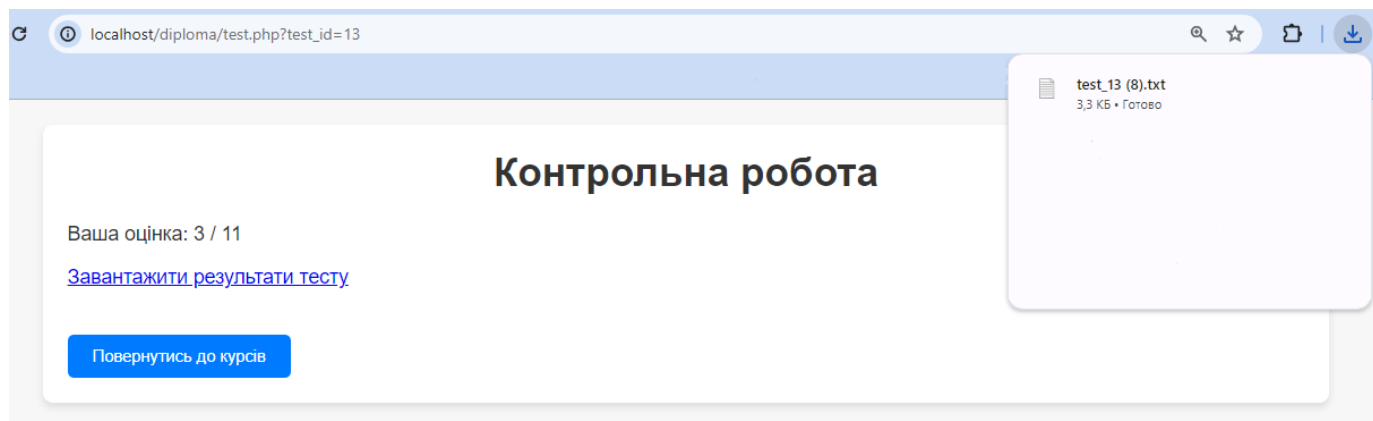


Рисунок 5.9 – Вкладка після пройденної контрольної роботи

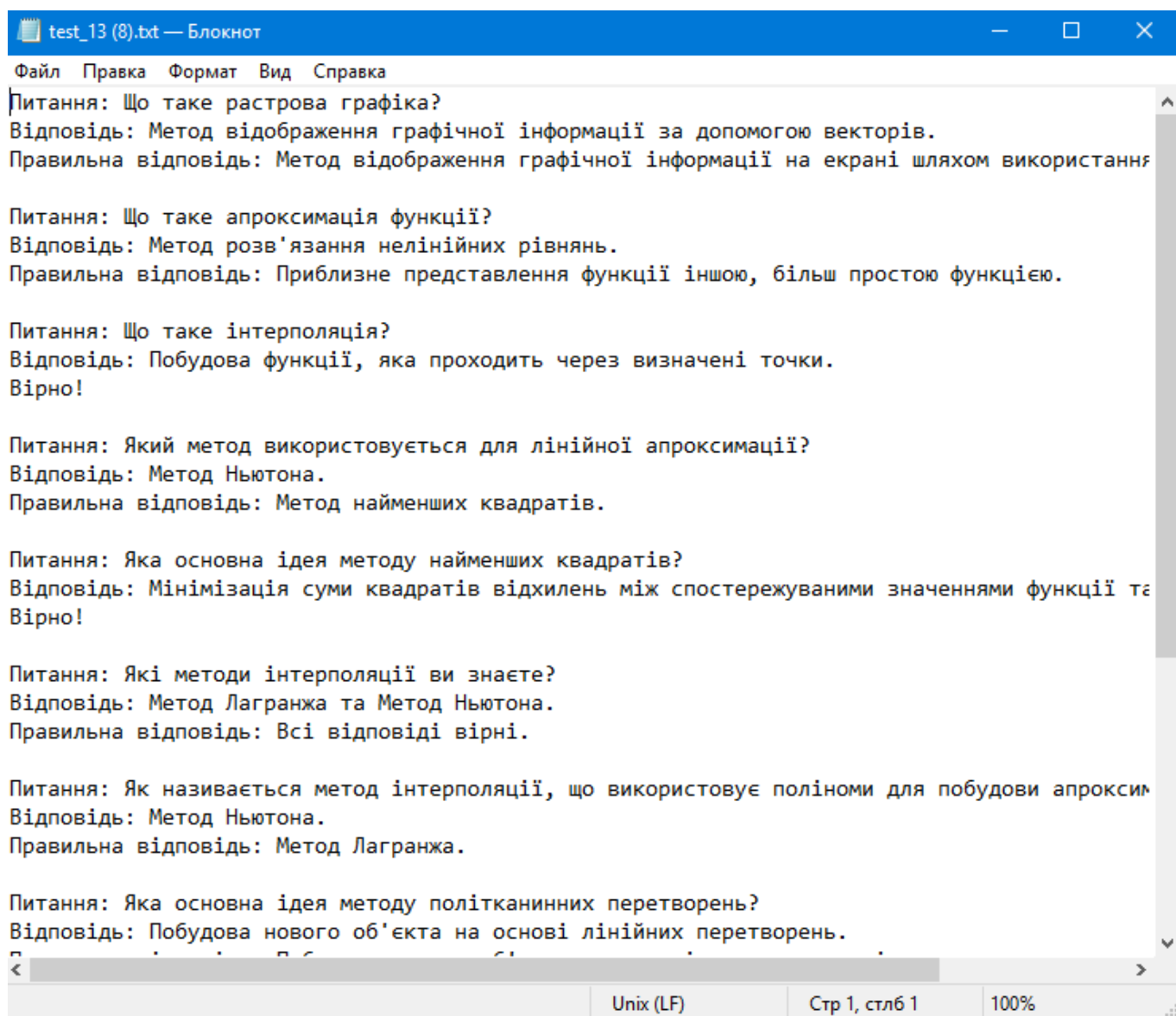


Рисунок 5.10 – Завантажений файл з тестом

Відкривши вкладку оцінки, користувач побачить усі свої оцінки, а також за пошуком має змогу знайти оцінки для конкретного курсу. Приклад вкладки оцінки наведено на рисунку 5.11.

Назва курсу	Назва тесту	Оцінка
Економіка і організація виробництва		
	Тест з теми- Вступ до економіки виробництва	3
	Тест з теми- Основи систем управління виробництвом	4
	Тест з теми- Основи фінансового аналізу виробничих підприємств	3
Моделювання систем в енергетиці		
	Тест з теми- Використання статистичних методів у аналізі даних енергосистем	3
	Тест з теми- Вступ до моделювання систем в енергетиці	4

Рисунок 5.11 – Вкладка оцінки

Відкривши вкладку Лабораторні роботи, користувач побачить список лабораторних робіт до відповідних курсів. Натиснувши на назву лабораторної роботи, користувач перейде за посиланням, де побачить завдання до лабораторної роботи, можливість завантажити або видалити файл, в якому буде міститись зроблена лабораторна робота. У разі успішного завантаження файлу, користувач побачить відповідне повідомлення.

Приклад вкладки списку лабораторних робіт до відповідних курсів наведено на рисунку 5.12, завдання до лабораторної роботи наведено на рисунку 5.13 та повідомлення у разі успішного завантаження файлу наведено на рисунку 5.14.

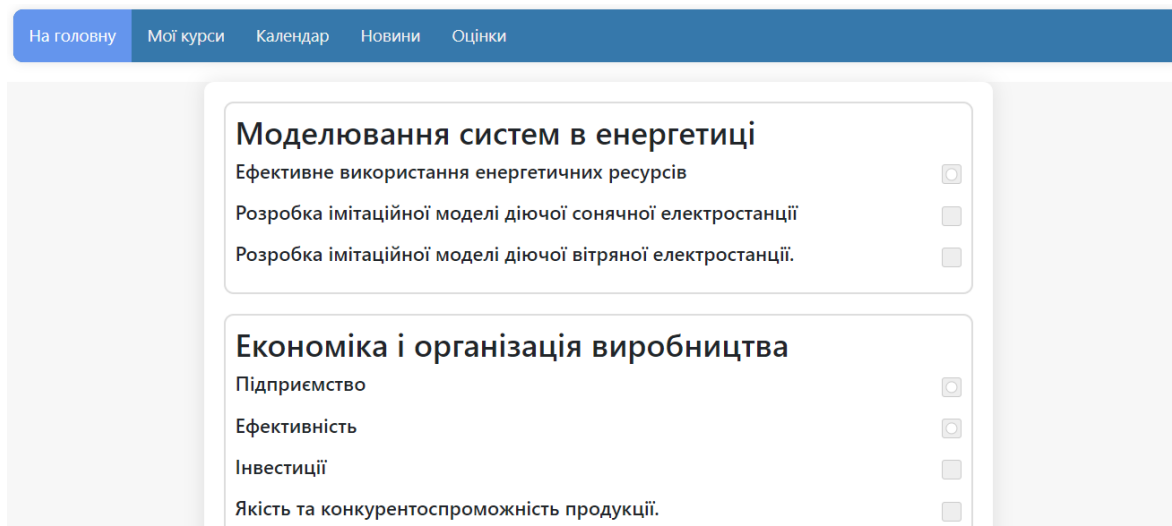


Рисунок 5.12 – Вкладка лабораторних робіт

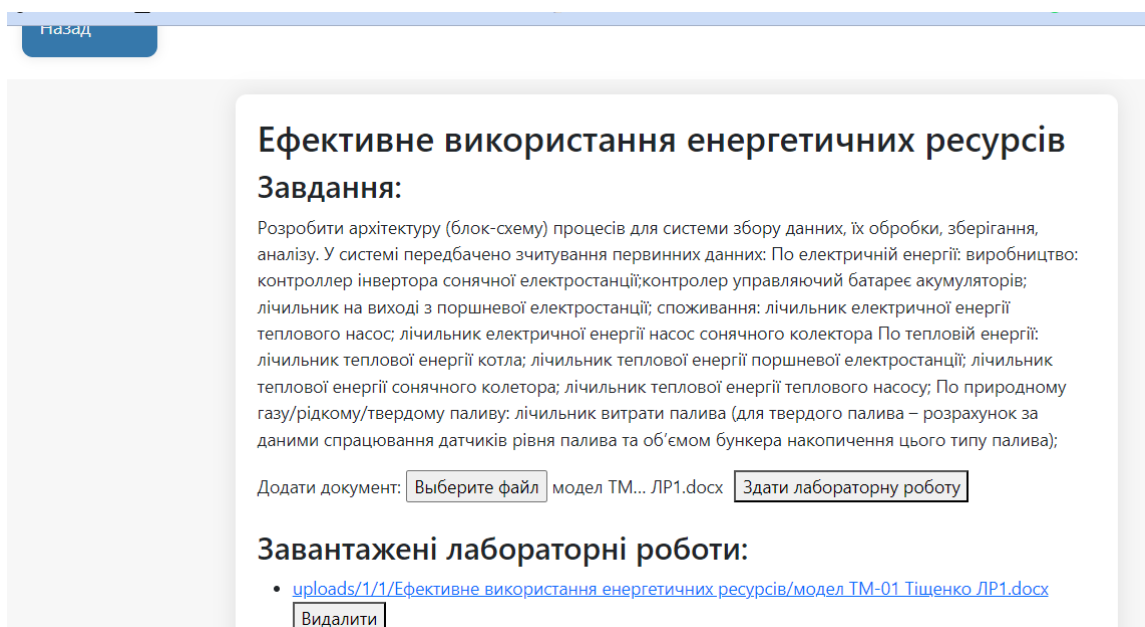
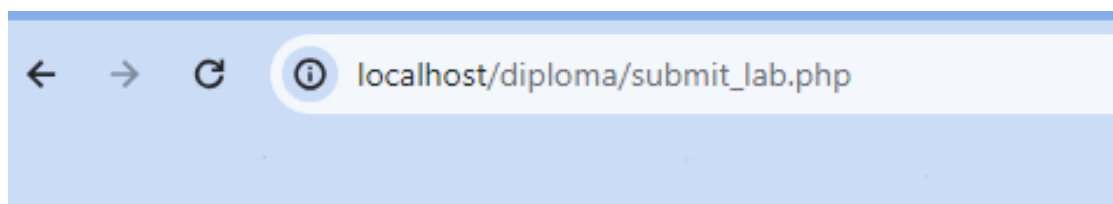


Рисунок 5.13 – Вкладка завдання до лабораторної роботи



Файл модел ТМ-01 Тіщенко ЛР1.docx успішно завантажено.

Рисунок 5.14 – Повідомлення у разі успішного завантаження файлу

Натиснувши кнопку вийти, користувач вийде зі системи. Приклад виходу з системи наведено на рисунку 5.15.

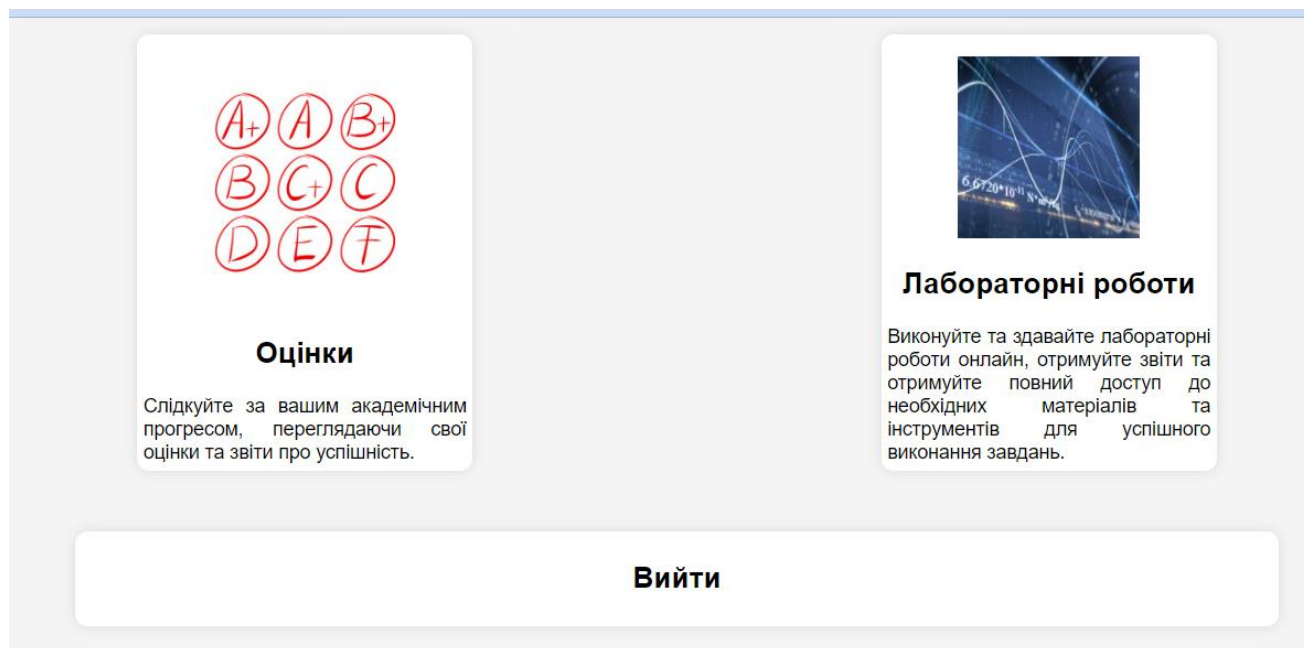


Рисунок 5.15 – Вихід з системи

Успішна взаємодія студента з платформою дистанційного навчання вимагає належної налаштування та коректної роботи всіх компонентів системи.

Платформа забезпечує доступ до особистого кабінету, новин, календаря навчального процесу, курсів, лекцій, тестів, контрольних та лабораторних робіт. Усі ці функції дозволяють студенту ефективно планувати та виконувати навчальні завдання. Система також надає можливість перегляду та аналізу оцінок, що сприяє моніторингу прогресу у навчанні.

Студент має змогу завантажувати матеріали, проходити тести та завантажувати результати своєї роботи, що робить процес навчання інтерактивним та зручним. Завдяки структурованому та зрозумілому інтерфейсу, платформа є ефективним інструментом для організації дистанційного навчання.

Завершуючи роботу з системою, студент може легко вийти з неї, що забезпечує безпеку та конфіденційність його даних. Усі ці аспекти свідчать про високу функціональність та зручність платформи дистанційного навчання.

ВИСНОВКИ

Під час виконання дипломної роботи було проведено аналіз та визначено переваги та недоліки наявних платформ для дистанційного навчання у сфері освіти. Було поставлено задачу на створення нової інформаційної системи для дистанційного навчання за предметами кафедри.

На сьогоднішній день доступні численні варіанти дистанційного навчання, але багато з них не відповідають особливим вимогам кафедр. Отже, було створено платформу дистанційного навчання, яка враховуватиме особливості кафедри, щоб зробити навчання простим і ефективним. Оцінка вимог кафедри та платформи дистанційного навчання. Визначено функціональні можливості платформи, забезпечено доступ до навчального матеріалу. Вибрано мови програмування та технологічні рішення для розробки платформи. Розроблено інтерфейс користувача платформи. Реалізовано функціональність платформи, включаючи створення системи для перевірки ідентифікації користувачів. Були проведені тестування та впровадження платформи.

Створена система дозволяє користувачам здобувати вищу освіту дистанційно та має усі функції дистанційної платформи для навчання, наприклад вхід в систему, перегляд та завантаження навчального матеріалу, здача лабораторних робіт та проходження контрольних робіт та тестів, а також оцінювання тестів, контрольних робіт та лабораторних робіт здійснюється автоматично та вручну.

Функціонал інформаційної системи розрахований на таку категорію користувачів як студенти. В розробленому веб-додатку є інтерфейс для студента, наприклад, звичайний користувач може переглядати новини, календар, графік навчального процесу, здавати лабораторні роботи, проходити тести на контрольні роботи, завантажувати пройдений тест або контрольну роботу після проходження для само-аналізу, переглядати свої оцінки та навчальний матеріал.

Створена інформаційна система є гнучкою в плані розширення і може бути використана як частина або основа для більш широко направлених програмних рішень у сфері освіти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке Moodle. URL: <https://moodle.org/mod/page/view.php?id=8174>.
2. Google Classroom: що це і як працює. URL: <https://www.bbc.com/ukrainian/news-52094706>.
3. Про онлайн навчання на платформі Coursera. URL: <https://ode.dcz.gov.ua/publikaciya/pro-onlayn-navchannya-na-platformi-coursera>
4. Що таке HTML. URL: https://css.in.ua/article/shcho-take-css_3
5. Що таке CSS. URL: https://css.in.ua/article/shcho-take-html_10
6. Що таке JavaScript. URL <https://goit.global/ua/articles/shcho-take-javascript-i-dlia-choho-vin-potriben/>
7. Що таке php. URL: <https://hyperhost.ua/uk/wiki/chto-takoe-php>
8. MySQL. URL: <https://promoter.net.ua/articles/baza-danix-mysql.html>
9. PhpStorm. URL: <https://itpro.ua/product/jetbrains-phpstorm/?tab=description>
10. MySQL Workbench. URL: <https://www.mysql.com/products/workbench/>
11. Xampp control panel. URL: <https://www.javatpoint.com/xampp-control-panel>
12. Use case diagram. URL: <https://zdorov.zapisi.cx.ua/ukraincyam/shho-take-diagrama-use-case-i-yak-vona-zastosovuietsya.html>
13. Діаграма класів. URL: <https://ua5.org/oop/392-diagrami-klasiv.html>
14. EER diagram. URL: <https://dev.mysql.com/doc/workbench/en/wb-eer-diagrams-section.html>
15. Що таке HTML. URL: <https://html.tutorial.in.ua/what-is-html/>
16. Що таке CSS. URL: http://www.ndu.edu.ua/liceum/html/teor_CSS.pdf
17. Вступ до JavaScript. URL: <https://uk.javascript.info/intro>
18. Php що це? Чим PHP краще за інші мови програмування? URL: <https://freehost.com.ua/ukr/faq/wiki/chto-takoe-php/>

19.Програма Хамрр: навіщо вона потрібна та які можливості надає. URL:
<https://shkola.zapisi.cx.ua/ukraincyam/programa-xampp-navishho-vona-potribna-ta-yaki-mozhlivosti-nadaie.html>

ДОДАТОК А

Платформа дистанційного навчання за дисциплінами кафедри

ТЕКСТ ПРОГРАМНОГО МОДУЛЮ

Аркушів 29

Київ – 2024

Текст програми

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Питання тесту</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f7f7f7;
      color: #333;
      margin: 0;
      padding: 0;
    }

    .container {
      width: 80%;
      margin: 20px auto;
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    }

    h2 {
      color: #333;
      text-align: center;
      margin-top: 0;
    }

    ol {
      padding-left: 20px;
    }

    li {
      margin-bottom: 10px;
    }

    input[type="submit"] {
      display: block;
      margin-top: 20px;
      padding: 10px 20px;
      background-color: #007bff;
      color: #fff;
```

```

        border: none;
        border-radius: 5px;
        cursor: pointer;
    }

    input[type="submit"]:hover {
        background-color: #0056b3;
    }
</style>
</head>
<body>

<div class="container">
    <h2>Питання тесту</h2>
</div>

<?php
session_start();
$connection = mysqli_connect("localhost", "root", "1234567890", "diploma");
if (!$connection)
    die("Нема підключення" . mysqli_connect_error());

if (isset($_SESSION['user']) && isset($_GET['test_id'])) {
    $testID = $_GET['test_id'];

    // Отримуємо запити для вибраного тесту
    $query = "SELECT * FROM Питання WHERE ID_тесту = ?";

    // Підготовлений запит
    $stmt = mysqli_prepare($connection, $query);
    mysqli_stmt_bind_param($stmt, "i", $testID);
    mysqli_stmt_execute($stmt);

    $result = mysqli_stmt_get_result($stmt);

    if (mysqli_num_rows($result) > 0) {
        // Виводимо питання тесту
        echo '<form method="post" action="test.php?test_id=' . $testID . "'>';
        echo '<ol>';
        while ($question = mysqli_fetch_assoc($result)) {
            echo '<li>' . $question['Питання'] . '</li>';
            // Отримуємо відповіді для кожного питання
            $answers_query = "SELECT * FROM Відповіді WHERE ID_питання = ?";
            $stmt_answers = mysqli_prepare($connection, $answers_query);
            mysqli_stmt_bind_param($stmt_answers, "i", $question['ID_питання']);
            mysqli_stmt_execute($stmt_answers);
        }
    }
}

```

```

    $answers_result = mysqli_stmt_get_result($stmt_answers);
    while ($answer = mysqli_fetch_assoc($answers_result)) {
        echo '<input type="radio" name="" . $question['ID_питання'] . "" value="" .
$answer['ID_відповіді'] . ""> ' . $answer['Відповідь'] . '<br>';
    }
}
echo '</ol>';
echo '<input type="submit" value="Закінчити тест">';
echo '</form>';
} else {
    echo 'Питання для цього тесту не знайдено.';
}
} else {
    echo 'Помилка: Тест не знайдено.';
}
?>

```

```
</body>
```

```
</html>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Тест</title>
```

```
<style>
```

```
body {
```

```
font-family: Arial, sans-serif;
```

```
background-color: #f7f7f7;
```

```
color: #333;
```

```
margin: 0;
```

```
padding: 0;
```

```
}
```

```
.container {
```

```
width: 80%;
```

```
margin: 20px auto;
```

```
background-color: #fff;
```

```
padding: 20px;
```

```
border-radius: 8px;
```

```
box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
```

```
}
```

```
h1 {
```

```
color: #333;
```

```
text-align: center;
```

```

    margin-top: 0;
    cursor: pointer;
}

form {
    display: none;
}

button {
    margin-top: 20px;
    padding: 10px 20px;
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
}

button:hover {
    background-color: #0056b3;
}
</style>
</head>
<body>

<div class="container">
    <?php
    session_start();
    $connection = mysqli_connect("localhost", "root", "1234567890", "diploma");
    if (!$connection) {
        die("Нема підключення" . mysqli_connect_error());
    }
    // Перевіряємо, переданий чи параметр ID_тесту
    if (isset($_GET['test_id'])) {
        $testID = $_GET['test_id'];

        // Запит для отримання інформації про тест
        $query = "SELECT * FROM Тест WHERE ID_тесту = $testID";
        $testResult = mysqli_query($connection, $query);
        $test = mysqli_fetch_assoc($testResult);

        // Перевіряємо, пройшов чи користувач тест
        if ($_SERVER['REQUEST_METHOD'] === 'POST') {
            // Обробка відповідей користувача та підрахунок балів
            $score = 0;
            $totalQuestions = 0;

```

```

$answers = [];
foreach ($_POST as $questionID => $answerID) {
    $query = "SELECT * FROM Відповіді WHERE ID_питання = $questionID
AND ID_відповіді = $answerID";
    $result = mysqli_query($connection, $query);
    $answer = mysqli_fetch_assoc($result);
    if ($answer && $answer['ID_відповіді'] == 1) {
        $score++;
    }
    $totalQuestions++;
    // Зберігаємо відповіді користувача
    $answers[$questionID] = [
        'question' => $questionID,
        'answer' => $answerID,
        'text' => $answer['Відповідь']
    ];
}

// Створення директорії для збереження результатів
$resultsDir = "test_results";
if (!is_dir($resultsDir)) {
    mkdir($resultsDir);
}

// Виведення результату
echo "<h1 onclick='toggleTest(\"testContent\")'>{$stest['Назва']}</h1>";
echo "<p>Ваша оцінка: $score / $totalQuestions</p>";

// Зберігаємо результати тесту у текстовому файлі
$filename = "$resultsDir/test_{$stestID}.txt";
$file = fopen($filename, "w");
if ($file) {
    foreach ($answers as $answer) {
        $questionID = $answer['question'];
        $answerID = $answer['answer'];
        $answerText = $answer['text'];
        $question_query = "SELECT * FROM Питання WHERE ID_питання =
$questionID";
        $question_result = mysqli_query($connection, $question_query);
        $question = mysqli_fetch_assoc($question_result);
        $questionText = $question['Питання'];

        $text = "Питання: $questionText\n";
        $text .= "Відповідь: $answerText\n";

        $is_correct = (int)$answerID === 1;
    }
}

```

```

        if (!$is_correct) {
            $correctAnswer_query = "SELECT * FROM Відповіді WHERE
ID_питання = $questionID AND ID_відповіді = 1";
            $correctAnswer_result = mysqli_query($connection,
$correctAnswer_query);
            $correctAnswer = mysqli_fetch_assoc($correctAnswer_result);
            $correctAnswerText = $correctAnswer["Відповідь"];

            $text .= "Правильна відповідь: $correctAnswerText \n\n";
        }
        else {
            $text .= "Вірно! \n\n";
        }

        fwrite($file, $text);
    }

    foreach ($answers as $answer) {
        $question_query = "SELECT * FROM Питання WHERE ID_питання =
$questionID";
        $question_result = mysqli_query($connection, $question_query);
        $question = mysqli_fetch_assoc($question_result);

        if (!$is_correct) {
            $text = "Питання: $questionText\n";
        }
    }

    fclose($file);
    // Посилання на завантаження результатів тесту
    echo "<p><a href='$filename' download>Завантажити результати
тесту</a></p>";
    } else {
        echo "Помилка: Неможливо відкрити файл для запису результатів
тесту.";
    }

    echo '<button onclick="window.location.href = \'courses.php\'">Повернутись
до курсів</button>';
    } else {
        // Отримуємо питання для вибраного тесту
        $query = "SELECT * FROM Питання WHERE ID_тесту = ?";
        $stmt = mysqli_prepare($connection, $query);
        mysqli_stmt_bind_param($stmt, "i", $testID);
    }

```

```

mysqli_stmt_execute($stmt);
$result = mysqli_stmt_get_result($stmt);

if (mysqli_num_rows($result) > 0) {
    // Виводимо питання тесту та форму для його проходження
    echo "<h1 onclick='toggleTest(\"testContent\")'>{$test['Назва']}</h1>";
    echo "<p>Дедлайн: {$test['Дедлайн']}</p>";
    echo "<form method='post' id='testContent'>";
    echo '<ol>';
    while ($question = mysqli_fetch_assoc($result)) {
        echo '<li>' . $question['Питання'] . '</li>';
        // Отримуємо відповіді для кожного питання
        $answers_query = "SELECT * FROM Відповіді WHERE ID_питання =
?";

        $stmt_answers = mysqli_prepare($connection, $answers_query);
        mysqli_stmt_bind_param($stmt_answers, "i", $question['ID_питання']);
        mysqli_stmt_execute($stmt_answers);
        $answers_result = mysqli_stmt_get_result($stmt_answers);
        while ($answer = mysqli_fetch_assoc($answers_result)) {
            echo '<input type="radio" name="' . $question['ID_питання'] . "'
value="" . $answer['ID_відповіді'] . "'>' . $answer['Відповідь'] . '<br>';
        }
    }
    echo '</ol>';
    echo "<button type='submit'>Закінчити тест</button>";
    echo "</form>";
} else {
    echo 'Питання для цього тесту не знайдено!';
}
} else {
    echo "Тест не знайдено.";
}
?>
</div>

<script>
function toggleTest(testId) {
    var testContent = document.getElementById(testId);
    if (testContent.style.display === "none") {
        testContent.style.display = "block";
    } else {
        testContent.style.display = "none";
    }
}
}
</script>

```

```

</body>
</html>

<?php
if (!isset($_SESSION)) {
    session_start(); // Розпочинаємо сесію, якщо вона ще не була розпочата
}

function getSubmittedLabFiles($user_id, $courseId, $labTitle) {
    $dir = "uploads/{ $user_id }/{ $courseId }/{ $labTitle }/";
    if (file_exists($dir)) {
        $files = scandir($dir);
        $files = array_diff($files, array('.', '..'));
        $file_list = array();
        foreach ($files as $file) {
            $file_list[] = array('Документ' => $dir . $file);
        }
        return $file_list;
    } else {
        return array();
    }
}

if (isset($_SESSION['user']['ID_користувача'])) {
    $user_id = $_SESSION['user']['ID_користувача']; // Отримуємо ID користувача з сесії
    if (isset($_FILES['document'])) {
        $courseId = $_POST['course_id'];
        $labTitle = $_POST['lab_title'];

        $dir = "uploads/{ $user_id }/{ $courseId }/{ $labTitle }/";
        if (!file_exists($dir)) {
            mkdir($dir, 0777, true);
        }

        $target_file = $dir . basename($_FILES["document"]["name"]);
        if (move_uploaded_file($_FILES["document"]["tmp_name"], $target_file)) {
            echo "Файл " . basename($_FILES["document"]["name"]) . " успішно
завантажено.";
        } else {
            echo "Під час завантаження файлу сталася помилка.";
        }
    }
}
}

```

?>

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Вхід в систему</title>
  <!-- Підключаємо Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
  <style>
    .allbackgr {
      background-image: url("kpi_big.jpg");
      background-size: cover;
      background-position: center;
      border: 10px solid white;
      border-radius: 10px;
      box-shadow: 0 0 20px rgba(0, 0, 0, 0.6);
      max-width: 1000px;
      margin: auto;
      overflow: hidden;
      height: 95vh;
    }

    .loginall {
      background-color: rgba(255, 255, 255, 0.7);
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
      max-width: 300px;
      width: 40%;
      margin: 20px auto;
    }

    .bigpicture {
      background-image: url("kpi_big.jpg");
      background-size: cover;
      background-position: center;
      height: 40%;
      opacity: 0.5;
    }

    @media (min-width: 992px) {
      .bigpicture {
        max-width: 33%;
      }
    }
  </style>

```

```

        float: left;
    }
    .loginall {
        max-width: calc(66% - 20px);
        float: none;
    }
}
</style>
</head>
<body>

<div class="allbackgr">
    <div class="bigpicture"></div>
    <div class="loginall">
        <div class="login1">
            <h5> Інститут освіти КПІ ім. Ігоря Сікорського.</h5>
            <h5>Платформа дистанційного навчання</h5>
        </div>

        <div class="login2">
            <div class="container mt-4">
                <h5 class="text-center">Будь ласка, увійдіть в систему</h5>
            </div>

            <div class="container mt-3">
                <form method="post" action="login.php">
                    <div class="mb-3">
                        <label for="exampleInputEmail1" class="form-label">Введіть вашу
пошту:</label>
                        <input name="email" type="email" class="form-control"
id="exampleInputEmail1" aria-describedby="emailHelp" required>
                    </div>
                    <div class="mb-3">
                        <label for="exampleInputPassword1" class="form-
label">Пароль:</label>
                        <input name="password" type="password" class="form-control"
id="exampleInputPassword1" required>
                    </div>
                    <div class="mb-3">
                        <button type="submit" class="btn btn-primary w-100" name="button-
reg">Вхід</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

```

</div>

<!-- Bootstrap JS -->
<!--<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></scr
ipt-->

</body>
</html>
<?php
session_start();

if (!isset($_SESSION['user'])) {
    header("Location: login.php");
    exit();
}

$user = $_SESSION['user'];
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Особистий кабінет користувача</title>
    <link rel="stylesheet" href="diplomaCSS.css">
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }

        .simptext {
            width: 80%;
            margin: 20px auto;
            background-color: #fff;
            padding: 20px;
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
            text-align: center;
            display: flex;
            align-items: center;
        }
    </style>

```

```
.simpltext h2 {
  text-align: center;
}

.simpltext p {
  margin: 0;
}

.container {
  width: 200px;
  height: 350px;
  margin: 20px 20px auto;
  background-color: #fff;
  padding: 5px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  text-align: center;
  display: flex;
  flex-direction: column;
  justify-content: space-around;
}

.container:hover {
  cursor: pointer;
  background-color: #f0f0f0;
}

.container h2 {
  text-align: center;
  margin-top: 10px;
}

.container p {
  margin: 0;
  text-align: justify;
}

.container a {
  display: block;
  text-decoration: none;
  color: #007bff;
  margin-top: 10px;
}

.container a:hover {
  text-decoration: underline;
}

.flex-container {
  display: flex;
  justify-content: space-around;
```

```
    flex-wrap: wrap;
  }

.container img {
  width: 150px;
  height: 150px;
  margin: auto;
  display: block;
}

@media only screen and (min-width: 600px) {
  .container {
    width: calc(50% - 40px);
  }
}

@media only screen and (min-width: 900px) {
  .container {
    width: calc(25% - 40px);
  }
}

.exitbot{
  width: 80%;
  margin: 50px auto;
  background-color: #fff;
  padding: 5px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  text-align: center;
}

.exitbot h2 {
  text-align: center;
}

.exitbot:hover {
  cursor: pointer;
  background-color: #f0f0f0;
}

.toptext {
  width: 30%;
  margin: 10px 10px 10px auto;
  background-color: #fff;
  padding: 20px;
```

```

border-radius: 10px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
text-align: right;
display: flex;
flex-direction: column;
align-items: flex-end;
}

.toptext p {
margin: 0;
text-align: right;
}

</style>
</head>
<body>

<div class="toptext">
  <p><?php echo $user['ФІО']; ?> <?php echo "-"; ?> <?php echo $user['Група'];
?></p>
</div>

<div class="simptext">
  
  <h2>Національний технічний університет України "Київський політехнічний
інститут імені Ігоря Сікорського"
  ПЛАТФОРМА ДИСТАНЦІЙНОГО НАВЧАННЯ</h2>
</div>

<div class="flex-container">
  <div class="container" onclick="redirectToClick('news.php')">
    
    <h2>Новини</h2>
    <p>Дізнайтесь останні новини та події, які стосуються вашого навчання та
життя університетського співтовариства.</p>
  </div>

  <div class="container" onclick="redirectToClick('calendar.php')">
    
    <h2>Календар</h2>
    <p>Переглядайте важливі дати, події та дедлайни, щоб планувати свої
навчальні та особисті обов'язки.</p>
  </div>

  <div class="container" onclick="redirectToClick('courses.php')">
    

```

```

    <h2>Курси</h2>
    <p>Отримуйте доступ до матеріалів курсів, завдань та ресурсів, щоб
        успішно вивчати предмети.</p>
</div>

<div class="container" onclick="redirectToClick('grades.php')">
    
    <h2>Оцінки</h2>
    <p>Слідкуйте за вашим академічним прогресом, переглядаючи свої оцінки
        та звіти про успішність.</p>
</div>

<div class="container" onclick="redirectToClick('labas.php')">
    
    <h2>Лабораторні роботи</h2>
    <p>Виконуйте та здавайте лабораторні роботи онлайн, отримуйте звіти
        та отримуйте повний доступ до необхідних матеріалів та
        інструментів для успішного виконання завдань.</p>
</div>
</div>

<div class="exitbot" onclick="redirectToClick('login.php')">
    <h2>Вийти</h2>
</div>

<script>
    function redirectToClick(url) {
        window.location.href = url;
    }
</script>

</body>
</html>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Новини</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f4f4f4;
        }
    </style>

```

```
.container {
  display: flex;
  justify-content: center;
  width: 80%;
  margin: 20px auto;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  flex-direction: column;
  background-image: linear-gradient(to bottom, rgba(255, 255, 255, 0.9),
  rgba(200, 230, 255, 0.9));
}

.container h2 {
  text-align: center;
}

.container p {
  margin: 0;
}

/* Add a black background color to the top navigation */
.topnav {
  background-color: #3678ab;
  overflow: hidden;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  margin: 20px;
}

/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}

/* Change the color of links on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

```

/* Add a color to the active/current link */
.topnav a.active {
    background-color: cornflowerblue;
    color: white;
}

@media only screen and (max-width: 600px) {
    .container {
        width: 100%;
    }
}

</style>
</head>
<body>

<div class="topnav">
    <a class="active" href="personalpage.php">На головну</a>
    <a href="courses.php">Мої курси</a>
    <a href="calendar.php">Календар</a>
    <a href="grades.php">Оцінки</a>
    <a href="labas.php">Лабораторні роботи</a>
</div>

<div class="container">
    <h2>Новини<br></h2>
</div>

<?php
$connection = mysqli_connect("localhost", "root", "1234567890", "diploma");
if (!$connection)
    die("Нема підключення" . mysqli_connect_error());

$query = "SELECT * FROM Новини";
$stmt = mysqli_query($connection, $query);

$i = 0;

while ($row = mysqli_fetch_array($stmt, MYSQLI_ASSOC)) {
    $i++;
    if ($i % 2 == 0) {
        echo '<div class="container" style="background-image: linear-gradient(to bottom,
        rgba(200, 230, 255, 0.9), rgba(255, 255, 255, 0.9));">';
    } else {
        echo '<div class="container" style="background-image: linear-gradient(to bottom,

```

```

    rgba(255, 255, 255, 0.9), rgba(200, 230, 255, 0.9));">';
    }
    echo '<h1 style="font-size: 20px; color: #333;">' . $row['Заголовок'] . '</h1>';
    echo '<p>' . $row['Основна_частина'] . '</p>';
    echo '</div>';
}
?>

</body>
</html>

<?php
session_start();
include "DBconnect.php";
include "registerHTML.html";

global $connection;

if (isset($_POST['button-reg'])) {
    $email = $_POST['email'];
    $password = $_POST['password'];

    // Перевірка введених даних в базі
    $query = "SELECT * FROM Користувач WHERE Пошта='$email' AND
Пароль='$password'";
    $result = mysqli_query($connection, $query);

    if ($result && mysqli_num_rows($result) > 0) {
        $user = mysqli_fetch_assoc($result);
        $_SESSION['user'] = $user; // Зберігаємо користувача в сесії
        header("Location: personalpage.php");
        exit();
    } else {
        echo "Невірний логін або пароль";
    }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Лекції</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
        body {

```

```
font-family: Arial, sans-serif;
background-color: #f7f7f7;
color: #333;
margin: 0;
padding: 0;
}

.container {
  width: 80%;
  margin: 20px auto;
  background-color: #fff;
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

h2 {
  color: #333;
  text-align: center;
  margin-top: 0;
}

.lecture-link {
  display: block;
  margin-bottom: 10px;
  text-decoration: none;
  color: #007bff;
  transition: color 0.3s;
}

.lecture-link:hover {
  color: #0056b3;
}

.lecture-content {
  display: none;
  margin-top: 10px;
  padding-left: 20px;
  border-left: 2px solid #007bff;
}

.lecture-content.active {
  display: block;
}

/* Add a black background color to the top navigation */
```

```
.topnav {
  background-color: #3678ab;
  overflow: hidden;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  margin: 20px;
}

/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  color: #f2f2f2;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
  font-size: 17px;
}

/* Change the color of links on hover */
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Add a color to the active/current link */
.topnav a.active {
  background-color: cornflowerblue;
  color: white;
}

@media only screen and (max-width: 768px) {
  .container {
    width: 90%;
  }
}

@media only screen and (min-width: 768px) {
  .lecture-link {
    display: inline;
    margin-bottom: 0;
    margin-right: 20px;
    cursor: pointer;
  }
}
```