

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

червня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавр

зі спеціальності 122 – Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «Інформаційна система керування онлайн-навчанням»

здобувача групи ІН-91 Пономаренка Гліба Віталійовича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Гліб ПОНОМАРЕНКО

(підпис)

Керівник,

старший викладач кафедри комп'ютерних

наук, кандидат фізико-математичних наук

Анна БАДАЛЯН

(підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН-91 Пономаренка Гліба Віталійовича

1. Тема роботи: «Інформаційна технологія прогнозування курсу валют»
затверджую наказом по СумДУ від «01» червня 2023 р. № 0475-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 09 червня 2023 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Розробка інформаційної системи керування онлайн-навчанням. 4) Аналіз результатів.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «___» _____ 20__ р.

Завдання прийняв до виконання _____
(підпис)

Керівник _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд платформ, що використовують для онлайн-навчання</i>		
3	<i>Розробка інформаційної системи керування онлайн-навчанням</i>		
4	<i>Аналіз отриманих результатів</i>		
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

АНОТАЦІЯ

Записка: 79 стр., 90 рис., 1 додаток, 20 джерел.

Об'єкт дослідження: процес проведення освітніх занять онлайн.

Мета: створити інформаційну систему керування онлайн-навчанням для можливості отримувати освіту онлайн та спрощення процедури проведення занять.

Предмет дослідження: об'єднання та розробка етапів взаємодії між викладачем та студентом.

Результати: результатом проведеної роботи є реалізованна та викладенна на хостинг інформаційна система керування онлайн-навчанням з можливістю створювати власні курси та завантажувати в них освітні матеріали, виконувати пошук потрібних курсів, формувати групи, створювати курси та календар онлайн-занять.

ІНФОРМАЦІЙНА СИСТЕМА, КЕРУВАННЯ, ОНЛАЙН-НАВЧАННЯ,
ОСВІТА, СТУДЕНТ, ВИКЛАДАЧ, КУРС, РОЗКЛАД,

ЗМІСТ

ВСТУП	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Аналіз предметної області.....	7
1.2 Аналіз інформаційних систем для керування онлайн навчанням	9
1.3. Мета та задачі дослідження	26
2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ СИСТЕМИ.....	29
2.1. Структура та функції інформаційної системи керування онлайн навчанням	29
2.2. Вибір засобів для програмної реалізації.....	31
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	34
3.1 Розробка дизайну інформаційної системи керування онлайн навчанням ...	34
3.2 Програмна реалізація інформаційної системи керування онлайн навчанням	36
3.3 Використання інформаційної системи керування онлайн навчанням зі	38
сторони користувача	38
3.4 Використання інформаційної системи керування онлайн навчанням зі	61
сторони адміністратора	61
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	65
ДОДАТОК А. ЛІСТИНГ КОДУ	67

ВСТУП

Кожного дня сучасне життя стає комфортнішим завдяки новітнім технологіям. Головним рушієм цього є наука, адже вона не стоїть на місці. Щодня виконуються дослідження, розробітки та винаходять нові рішення для тих чи інших проблем. Більшість досліджень направлені на те, щоб покращити наше життя.

Завдяки новітнім технологіям у людей з'являється багато нових можливостей. Одна з таких можливостей – отримання знань. Школи і університети – є основними місцями для навчання. Книги та вчителі – були єдиним джерелом знань, які були доступні для учнів, тому треба було чимало часу приділяти на пошук саме того, що вам потрібно.

З моменту, коли з'явився інтернет книги почали переноситись туди у вільний доступ, щоб люди мали можливість знайти їх у найкоротший час і могли шукати потрібну їм інформацію набагато швидше. До того ж, багато людей почали ділитися власним досвідом на просторах мережі. Форуми набрали велику популярність за короткі терміни, так як люди мали можливість ділитися своєю проблемою одразу на весь світ і отримувати відповідь від великої кількості людей з потрібним їм рішенням.

Так сформувався новий термін «онлайн-навчання». Коли кожен може знайти будь-яку необхідну для нього інформацію лише у своєму чпристрої, який має доступ до інтернету.

Гіпотеза: використання створеної інформаційної системи та виконання керування онлайн-навчанням у освітніх цілях.

Практичне значення: практичне значення у цьому проєкті є створення інформаційної системи по онлайн-навчанню, яка задовольняє всі поставлені потреби зі спрощення організації освітнього процесу онлайн.

При використанні даної інформаційної системи реалізовані такі функціональні можливості, як:

- оцінювання робіт в межах інформаційної системи;
- проходження існуючих курсів;
- створювання власних курсів;
- полегшення пошуку та комунікації із викладачами;
- надсилання виконаних завдань в межах інформаційної системи;
- оцінювання студентських робіт;
- планування розкладу занять;
- формування груп;

Отже, було прийнято рішення по розробці інформаційно-комунікаційної керування онлайн навчанням.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз предметної області

За останні роки чималу популярність набули сервіси для проведення онлайн-зустрічей. Зараз такою функцією користуються для багатьох цілей. Одна з таких – онлайн навчання.

Всі звикли, що для отримання знань необхідно щодня відвідувати навчальні заклади і проводити там купу часу, для того щоб засвоювати знання, після чого ще працювати з матеріалом вдома та у вільний час. Але тепер багато чого змінилося.

Онлайн-навчання є частиною курсів багатьох навчальних закладів у всьому світі. З курсами, доступними майже з усіх предметів, і гнучкими розкладами, які відповідають майже будь-якому стилю життя, студенти все частіше звертаються до онлайн-навчання як до життєздатної альтернативи навчання на кампусі[1]. Використання додаткових платформ дає здатність людям навчатися багатьом різним речам не покидаючи свого дому. Подібно до відвідування фізичного кампусу та особистого навчання, є переваги відвідування віртуального класу та навчання онлайн.

Серед багатьох переваг онлайн-навчання ви побачите, що віртуальна освіта дозволяє вам насолоджуватися більш гнучким графіком та може дозволити вам легше розвивати свою кар'єру разом із продовженням освіти [2]. У традиційній аудиторії лекції будуть заплановані на певний час дня, і ваш розклад буде сформовано відповідно до наявності занять.

Якщо ви зараз працюєте, а курси недоступні в неробочий час, може бути важко жонглювати курсовим навантаженням на додаток до ваших робочих обов'язків. Відвідуючи віртуальний кампус, онлайн-навчання дає набагато більше автономії у вирішенні власного розкладу. Це означає, що ви можете

навчатися, коли вам зручно. Багато аспірантів не можуть дозволити собі відпочивати на роботі. Або вони хочуть продовжувати працювати над кар'єрним зростанням, залишаючись на своїй поточній посаді, покращуючи свої облікові дані та застосовуючи отримані знання на роботі. Отже, якщо ви хочете продовжувати працювати на поточній роботі, здобуваючи ступінь, онлайн-навчання – це чудовий шлях. Вам не потрібно вибирати між навчанням і роботою – ви можете робити обидва[3].

Пристосування до моделі онлайн-навчання спочатку може бути складним завданням, але як тільки ви адаптуєтесь до формату, ви отримаєте численні переваги. Незалежно від причини, з якої ви вирішите продовжити онлайн-навчання, отримання онлайн-ступеню може допомогти вам підготуватися до кар'єрного зростання та продемонструвати ключові навички потенційним роботодавцям[4]. Також офіційні заклади освіти почали масово переходити на онлайн-формат, що надає їм та для учнів(студентів) купу додаткових переваг, одним з яких: безпека для власного життя та здоров'я. Окрім того, навчання у вашому власному темпі, участь у більш комфортних способах і відсутність важкої поїздки на роботу можуть допомогти вам обмежити стрес.

Фактично, протягом останніх років люди побачили, що таке практична, стійка модель онлайн-навчання. Це забезпечує доступність освіти під час кризи в галузі охорони здоров'я, стихійного лиха чи інших обставин, коли студенти та викладачі не можуть подорожувати. Він може обслуговувати студентів у ширшому географічному діапазоні, залучаючи голоси з усієї країни та за її межами.

Також, онлайн навчання розділяють на певні категорії: асинхронні онлайн-курси - ці типи курсів не відбуваються в режимі реального часу. Студентам надається зміст і завдання, а також час для виконання курсової роботи та іспитів. Взаємодія зазвичай відбувається через дошки обговорень, блоги та вікі. У результаті немає часу для зборів класу. Асинхронне онлайн-навчальне

середовище ефективно для студентів з обмеженим часом або зайнятим графіком. Синхронні онлайн-курси - ці типи курсів вимагають одночасної взаємодії викладача та всіх зареєстрованих студентів онлайн.

Подібно до певної міри вебінару, учасники взаємодіють за допомогою текстового, відео чи аудіочату. Синхронні навчальні середовища дозволяють студентам брати участь у курсі на відстані в режимі реального часу. Змішаний тип курсів - це навчальні середовища, які дозволяють як особисто, так і онлайн-взаємодію. Як правило, гібридні курси проводяться особисто кілька разів протягом семестру та передбачають комп'ютерне спілкування між сесіями. Відповідно до вимог різних типів онлайн-курсів, були створені спеціальні інформаційні системи та веб-додатки, які надають змогу проводити подібні заняття[5].

1.2 Аналіз інформаційних систем для керування онлайн навчанням

З часом онлайн-навчання набуло великої популярності, а отже кількість веб-додатків для нього зростає. Багато освітніх закладів створили власні інформаційні системи, але найпопулярніші відкриті веб-ресурси зберігають велику популярність.

Для аналізу веб-додатків для онлайн-курсів/навчання було обрано «Classroom.google.com»[6], «Udemy.com»[7], «Prometheus.org.ua»[8].

Розглянемо веб-додаток «Classroom.google.com». Даний додаток являє собою платформу для створення онлайн-курсів.

На рисунку 1.1 зображено головну сторінку «Classroom.google.com», на якій буде представлено список ваших курсів. На рисунку список пустий, відповідно вам буде запропоновано дві функції: «створити курс» та «приєднатися до курсу».

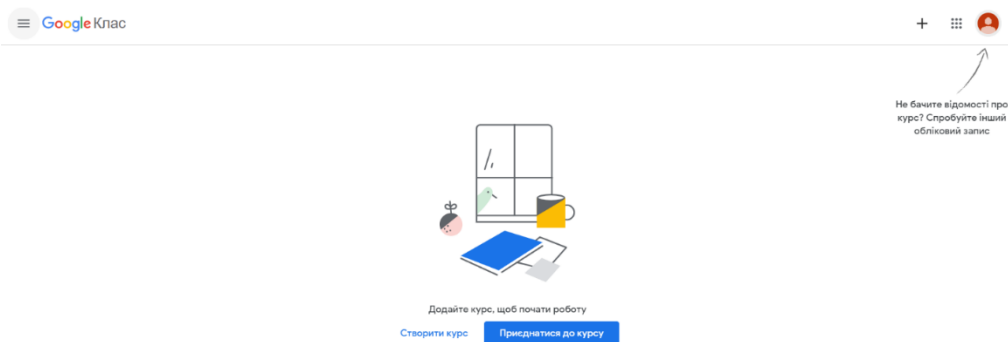


Рисунок 1.1 – «Classroom.google.com» - Головна сторінка без курсів

Коли ви приєднаєтесь до потрібного курсу, він з'явиться у списку, де буде відображено коротку інформацію про нього: назва курсу, автор курсу, як на рисунку 1.2.

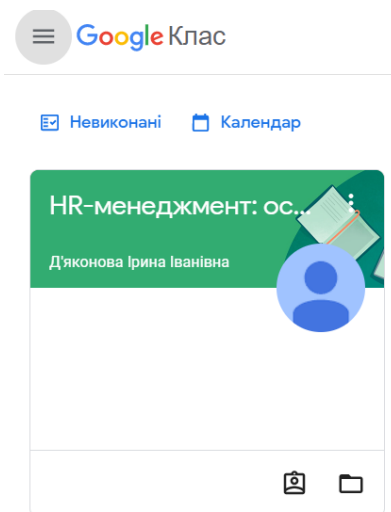


Рисунок 1.2 – «Classroom.google.com» - Доступний курс на головній сторінці

Додатково додаються дві кнопки «невиконані» та «календар». У вкладці «невиконані» будуть відображатися завдання, які ви не здали із курсів, як зображено на рисунку 1.3.

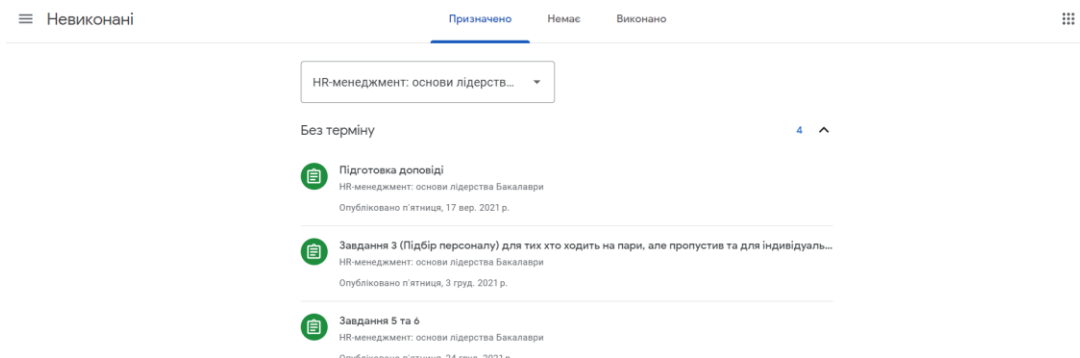


Рисунок 1.3 – «Classroom.google.com» - Невиконані завдання з курсів

У календарі можна переглянути заплановані заняття та заходи, які пов'язані з курсами (рис.1.4). Однією з головних переваг такої функції є відображення конфліктів занять, коли вони заплановані на приблизно на один і той же час, що дає можливість студентам одразу планувати свій графік.

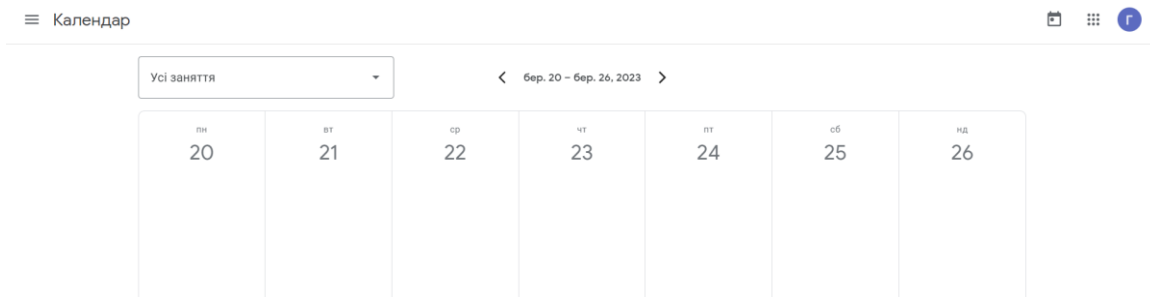


Рисунок 1.4 – «Classroom.google.com» - Календар занять

Можливість створити власний курс або приєднатися до існуючого переноситься у додаткову кнопку у верхньому правому кутку сторінки, як на рисунку 1.5.

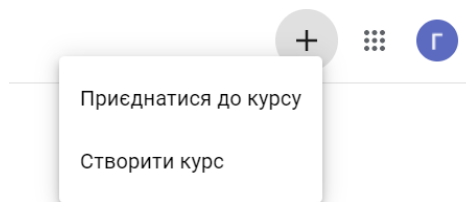


Рисунок 1.5 – «Classroom.google.com» - Дії з курсами

Також реалізовано додаткову панель зі списком доступних функцій, як на рисунку 1.6, де також є зручна додаткова функція перейти до архіву неактуальних курсів.

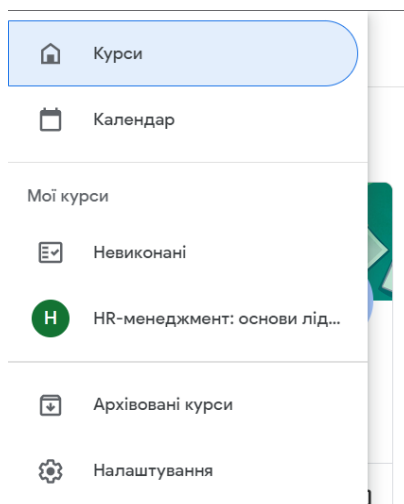


Рисунок 1.6 – «Classroom.google.com» - Додаткова бокова панель

На панелі доступних курсів є дві додаткові кнопки «завдання», «папка», які зображені на рисунках 1.7, 1.8.

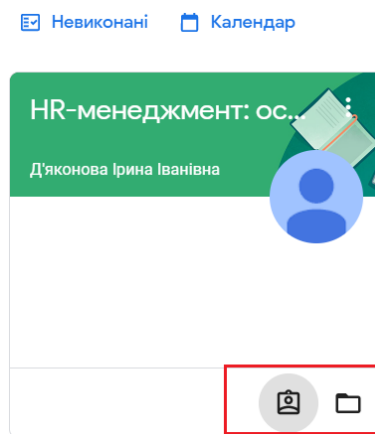
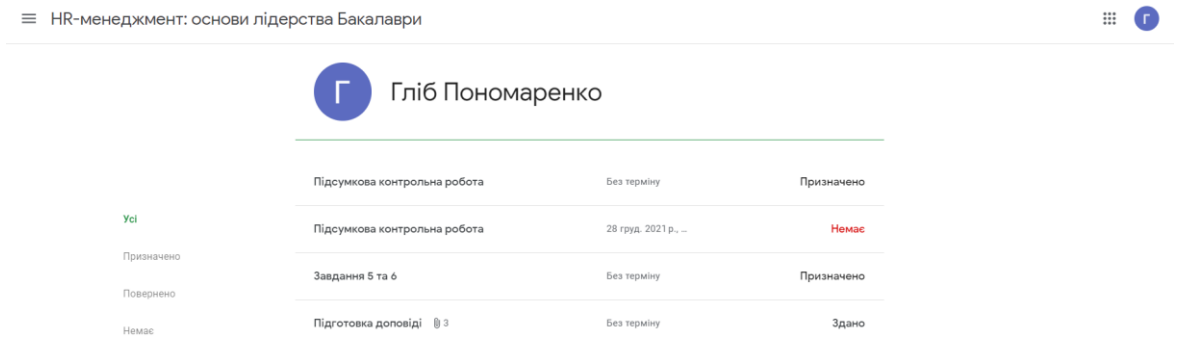


Рисунок 1.7 – «Classroom.google.com» - Додаткові вкладки курсу

Перейшовши за кнопкою «завдання» вам буде доступно список завдань, які було викладено саме у цьому курсі та відображено їх статус (рис.1.8).



	Підсумкова контрольна робота	Без терміну	Призначено
Усі	Підсумкова контрольна робота	28 груд. 2021 р., ...	Немає
Призначено	Завдання 5 та 6	Без терміну	Призначено
Повернено	Підготовка доповіді @ 3	Без терміну	Здано
Немає			

Рисунок 1.8 – «Classroom.google.com» - Сторінка «завдання» для обраного курсу

Кнопка «папка» переносе користувача на сторінку гугл диску, де буде відображено загальний список файлів – звітів, які користувач надсилав для здачі завдань (рис.1.9). Дана функція є корисною тим, що дозволяє студентам зберігати їх роботи у хмарному середовищі та мати доступ до них будучи віддаленим від пристрою на якому фізично було збережено файли.

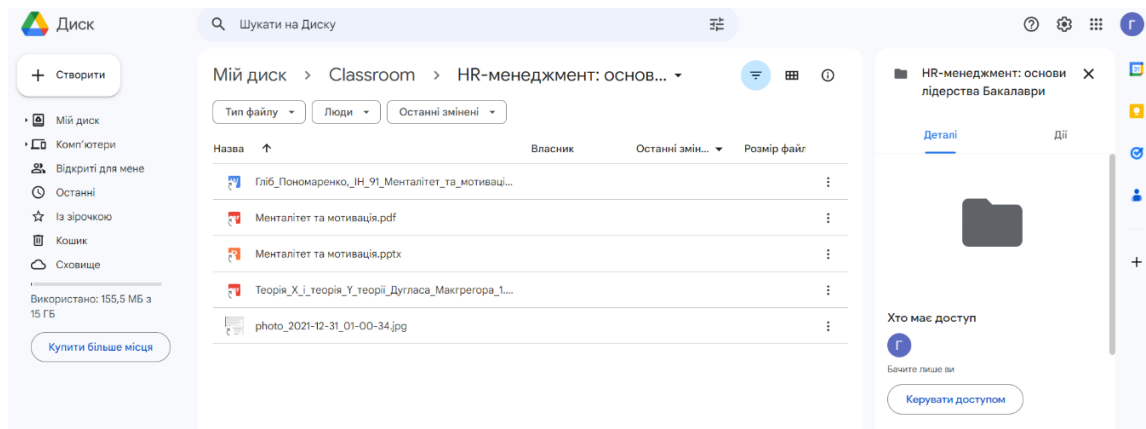


Рисунок 1.9 – «Classroom.google.com» - Гугл диск з файлами-звітами

На головній сторінці обраного курсу (рис.1.10) відображається вся загальна інформація про курс, основна дошка подій, завдання та доступний список учасників.

Користувачеві доступна функція надсилати повідомлення на головну дошку подій, а також коментувати оголошення інших користувачів. Додатково до курсу інтегровано окрему кнопку з посиланням на зустріч у google meet, що є дуже зручним рішенням і дозволяє уникнути «втрати посилання» та економить власний час, який було б затрачено на пошук цього посилання. Окремо створене поле-нагадування для відображення найближчих подій.

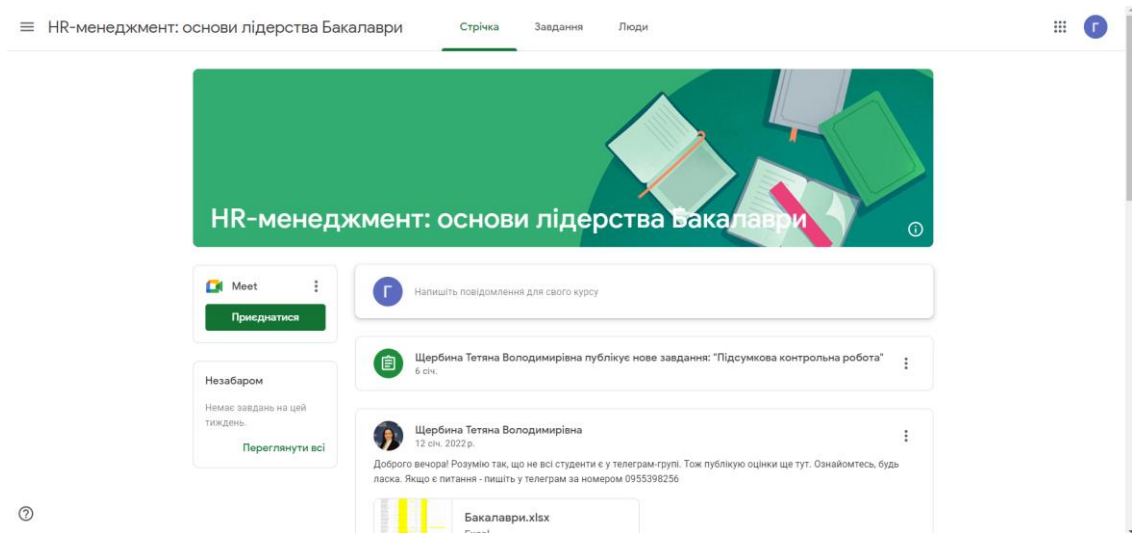


Рисунок 1.10 – «Classroom.google.com» - Головна сторінка курсу

На сторінці «завдання» відображається увесь список завдань, їх назви та термін на їх виконання, якщо така опція необхідна (рис 1.11). Додатково винесені кнопку-посилання на календар та зустріч.

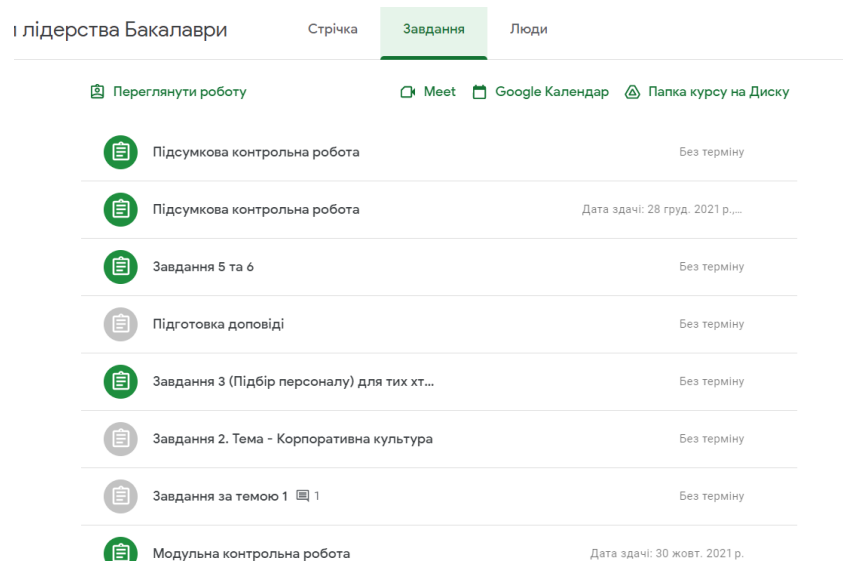


Рисунок 1.11– «Classroom.google.com» - Вкладка «завдання» обраного курсу

Натиснувши на обране завдання, відобразиться детальна інформація від викладача про те, що саме необхідно виконати. До опису можна додавати посилання на файл – інструкцію, як зображено на рисунку 1.12.

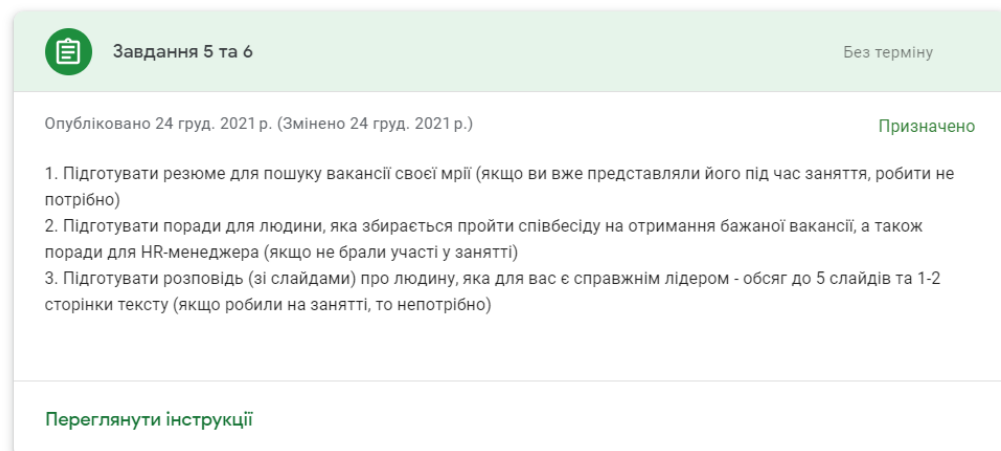


Рисунок 1.12 – «Classroom.google.com» - Детальний опис завдання

Перейшовши за завданням, відкриється його сторінка (рис.1.13), де буде надано додаткові функції, такі як: надіслати файл-звіт виконаної роботи,

відмітити завдання, як виконане та написання приватних повідомлень викладачу курсу.

The screenshot shows a Classroom assignment interface. On the left, the assignment title is 'Завдання 5 та 6' with a green icon. Below it, the instructor's name 'Щербина Тетяна Володимирівна' and the date '24 груд. 2021 р. (Змінено 24 груд. 2021 р.)' are displayed. The assignment description contains three numbered tasks: 1. Prepare a resume for a job search, 2. Prepare advice for an interview, and 3. Prepare a speech with slides. On the right, there are two panels. The top panel, titled 'Ваша робота' (Your work), shows the status 'Призначено' (Assigned) and buttons for '+ Додати або створити' (Add or create) and 'Позначити як виконане' (Mark as completed). The bottom panel, titled 'Приватні коментарі' (Private comments), has a button 'Додати коментар для викладача (Щербина Тетяна Володимирівна)' (Add comment for teacher).

Рисунок 1.13 – «Classroom.google.com» - Головна сторінка завдання

Розглянемо веб-додаток «Www.udemy.com». Даний додаток є більш розширеним, оскільки на ньому безпосередньо доступний загальний каталог існуючих курсів від різних викладачів, компаній і навчальних закладів. На рисунках 1.14-1.15 представлена головна сторінка додатку, на якому розміщено банер сайту та рекламні оголошення популярних курсів зі знижками. На верхній панелі сайту зручно розміщено кнопку каталогу.

The screenshot shows the Udeemy.com homepage. At the top, there is a navigation bar with the Udeemy logo, a search bar, and links for 'Udeemy Business', 'Teach on Udeemy', 'Log in', and 'Sign up'. The main banner features a woman in a red top with the text 'It's the last day to save' and 'Expand your horizons with learning that's worldwide. Log in now to shop the sale.' Below the banner, there is a section titled 'A broad selection of courses' with the text 'Choose from 213,000 online video courses with new additions published every month' and a list of course categories: Python, Excel, Web Development, JavaScript, Data Science, Amazon AWS, and Drawing.

Рисунок 1.14 – «Www.udemy.com» - Головна сторінка додатку

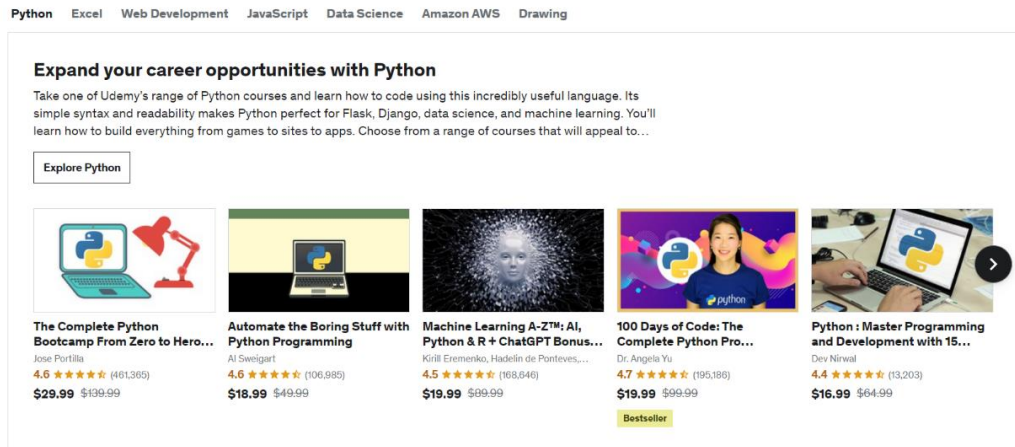


Рисунок 1.15 - «Www.udemy.com» Головна сторінка додатку

На рисунку 1.16 відображено розгорнутий каталог усіх категорій курсів та їх підкатегорій для зручного пошуку необхідних користувачеві курсів.

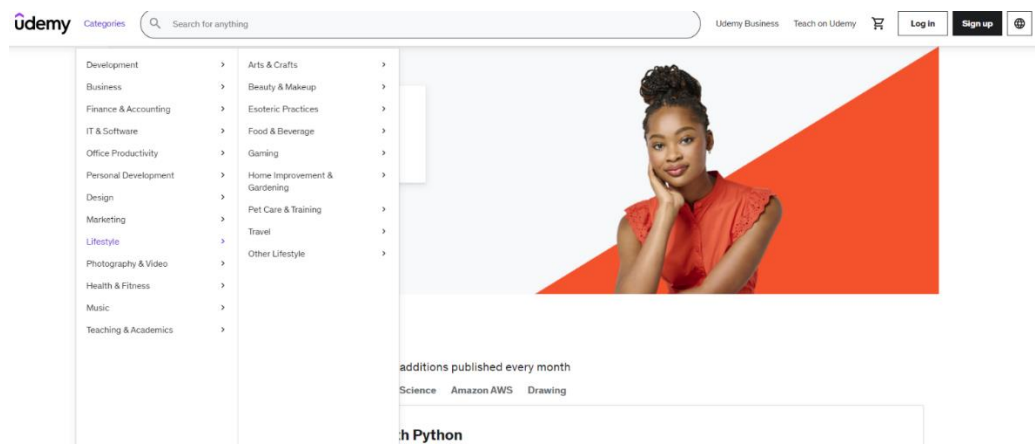


Рисунок 1.16 – «Www.udemy.com» - Розгорнутий каталог категорій курсів

Для того, щоб користувач відчував себе комфортніше, на головній сторінці автори розмістили дошку, на якій написані розміщені відгуки та поради від студентів, які вже пройшли якісь курси і мають досягнення (рис.1.17).

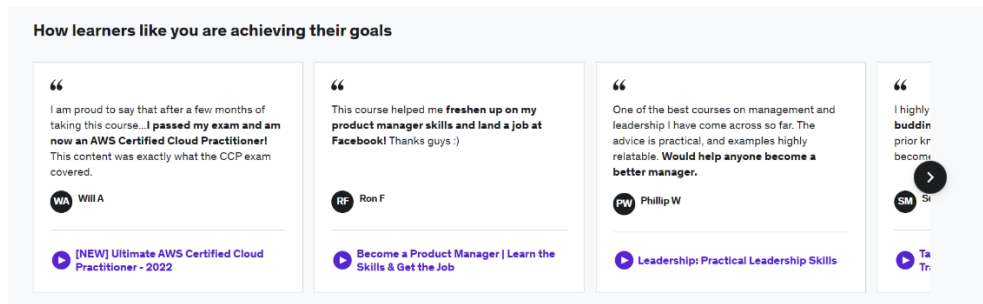


Рисунок 1.17 - «Www.udemy.com» - Відгуки студентів

Нижче реалізовано каталог популярних категорій курсів із застосуванням графічних зображень, що робить каталог візуально спрощеним та зручнішим (рис.1.18.)

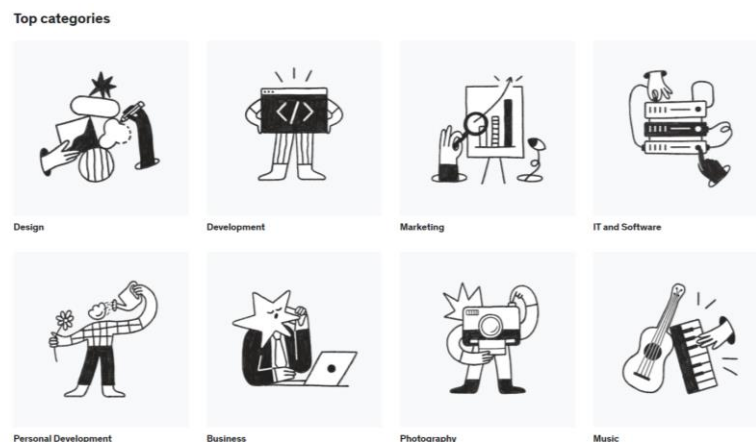


Рисунок 1.18 – «Www.udemy.com» - Каталог з графічною реалізацією

Обравши одну із категорій, користувачу відкривається сторінка, де буде відображено всі курси за даною категорією. У даному додатку реалізовано рекомендації курсів за «популярністю», «нові» та «для початківців», що може допомогти користувачеві звужити його коло пошуку, як зображено на рисунку 1.19

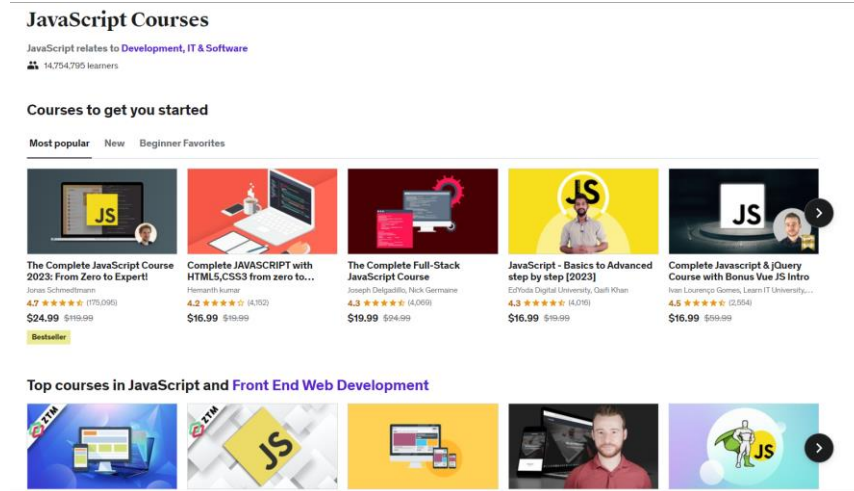


Рисунок 1.19 – «Www.udemy.com» - Сторінка курсів за обраною категорією

Нижче розгорнуто загальний список усіх курсів, до яких можна застосувати фільтри, які розташовані у полі зліва. Однією з переваг такого фільтру є те, що курси можна сортувати не тільки за їх популярністю, але й за довжиною уроків і рейтингом оцінок (рис.1.20).

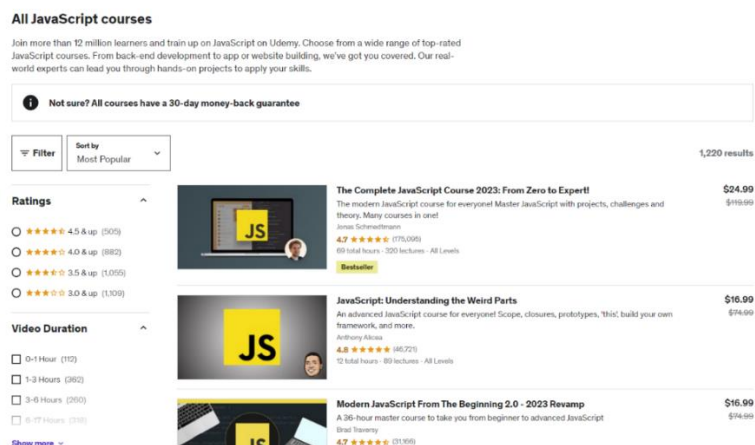


Рисунок 1.20 – «Www.udemy.com» - Загальний список з фільтром

Коли користувач відкриває курс, який, на його думку, міг би йому підійти, перед ним відкривається сторінка (рис.1.21) на якій знаходиться загальна інформація про курс, його вміст, може бути розміщено прев'ю-відео. Доступні

додаткові вкладки для перегляду матеріалів, що використовуються, відгуки та дані про викладачів. На сторінці є кнопка яка реєструє користувача на проходження курсу.

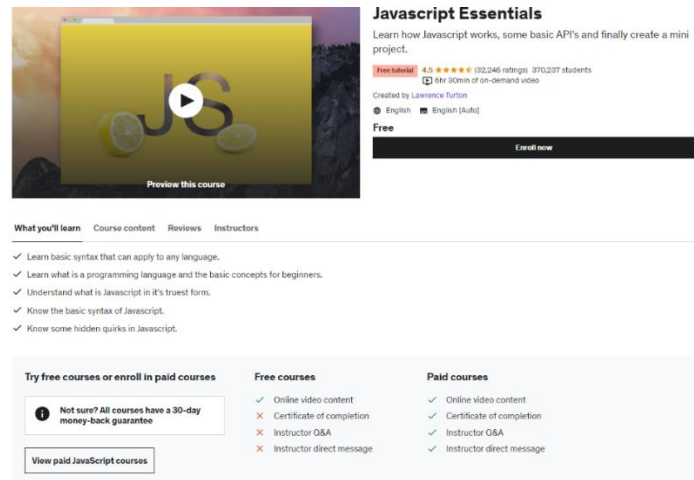


Рисунок 1.21 – «Www.udemy.com» - Сторінка інформації про курс

Наступний на розгляді веб-додаток «prometheus.org.ua». За своєю структурою та концепцією додаток має схожість з додатком, який був попереднім. На головній сторінці також є банер, який одразу пропонує зареєструватися, та короткий опис додатку. Одразу видно першу перевагу: наявна українська локалізація сторінок (рис.1.22). На даній сторінці користувач також можемо бачити верхню панель користувача.

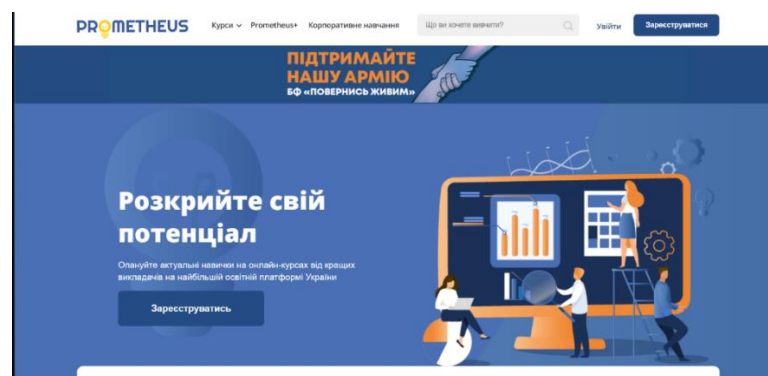


Рисунок 1.22 – «Prometheus.org.ua» - Головна сторінка

Нижче на головній сторінці відображено загальну статистику, де відображено кількість слухачів, курсів, сертифікатів та кількість років роботи сервісу у сфері освіти. Нижче бачимо рекламу нових та доступних курсів (рис.1.23.)

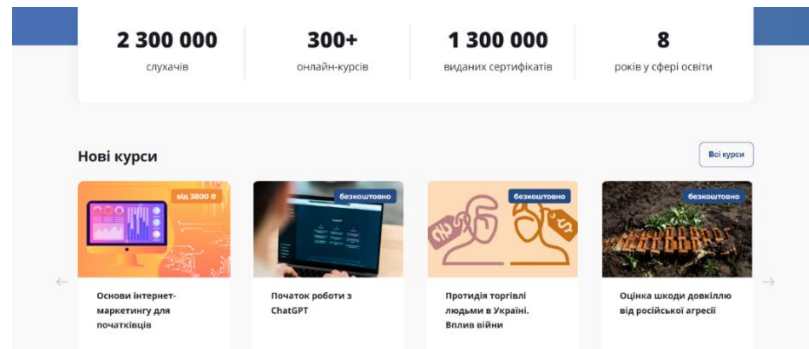


Рисунок 1.23 – «Prometheus.org.ua» - Головна сторінка

Ще нижче користувач може бачити також каталог категорій курсів з графічною реалізацією, як була у попереднього сервісу. На відміну від попереднього веб-додатку, у графічній реалізації категорій використано кольорову палітру, що робить картинку більш насиченою (рис.1.24).

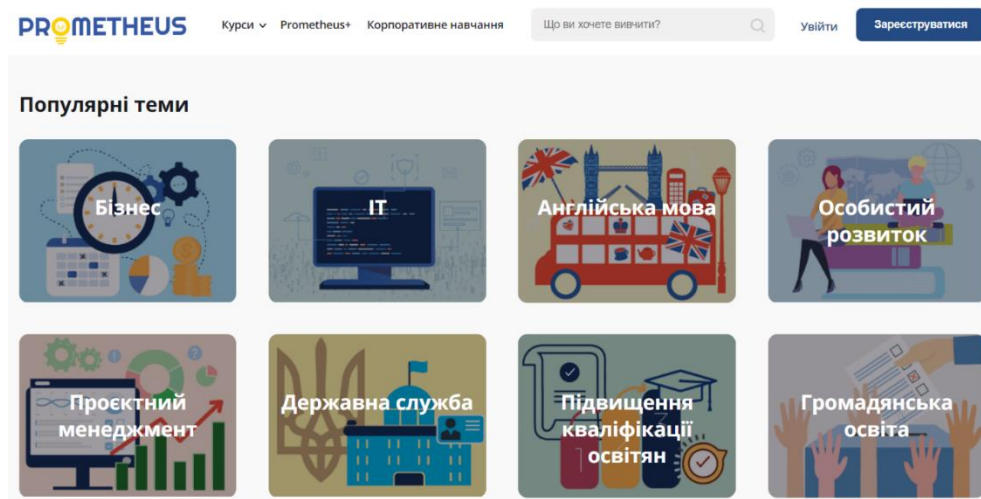


Рисунок 1.24 – «Prometheus.org.ua» - Категорії курсів з граф. реалізацією

На продовженні головної сторінки користувачу, також, пропонують дві додаткові кнопки, завдяки одній з яких користувач може створити власний курс. Нижче на рисунку 1.25 користувач бачить детальний опис платних курсів та їх переваги.

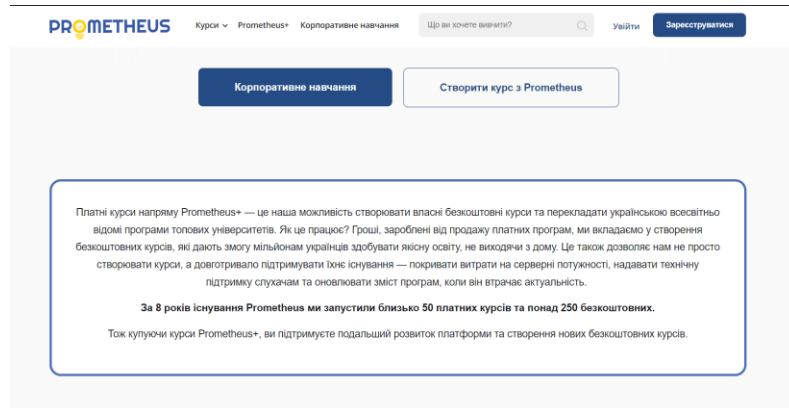


Рисунок 1.25 – «Prometheus.org.ua» - Головна сторінка

Ще нижче на головній сторінці користувач може бачити панель з відгуками від учнів та студентів, які проходили курси від даного сервісу (рис.1.26). Подібну панель вже було зображено у веб-додатку від «Udemy».

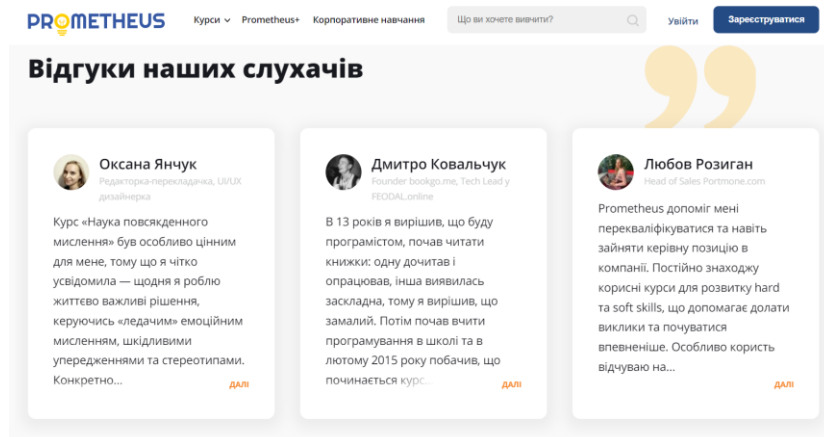


Рисунок 1.26 – «Prometheus.org.ua» - Панель відгуків

У кінці головної сторінки розміщено мікроблог, що також розбавляє інформаційний простір веб-додатку (рис.1.27). Там користувач може бачити посилання на сам блог та різні оголошення, пов'язанні з роботою та життям сервісу.

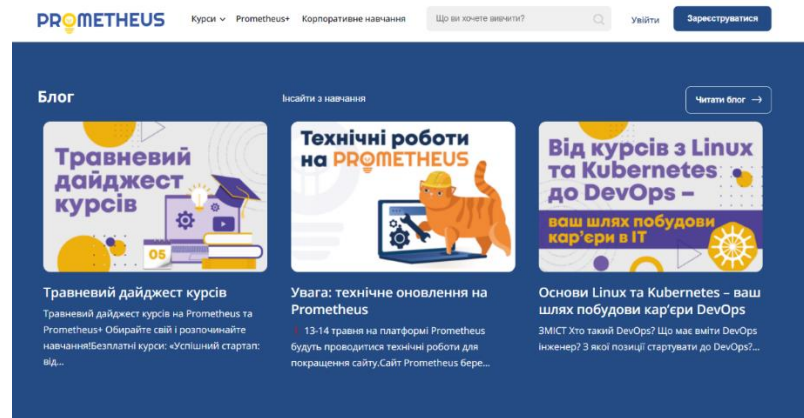


Рисунок 1.27 – «Prometheus.org.ua» - Мікроблог на головній сторінці

Відкривши каталог курсів на верхній панелі користувача, можна побачити запропоновані категорії курсів, та додатково глобальні фільтри «Prometheus+» та «Безкоштовні курси». Після перегляду каталогу від «Udemy» (рис.1.28), одразу видно, що у даному сервісі каталог не такий детальний, та розширений.

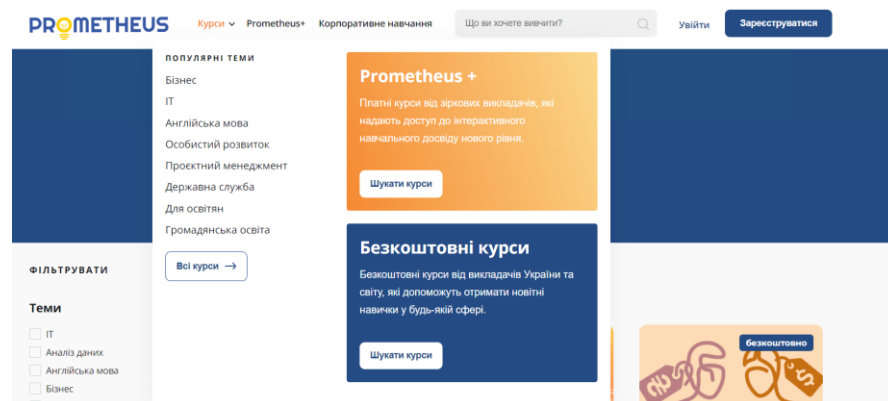


Рисунок 1.28 – «Prometheus.org.ua» - Каталог категорій курсів

Перейшовши за однією з категорій користувач переходить на сторінку пошуку курсів. Зручним фільтром є сортуванням за критерієм ціни та часу публікації курсу (рис.1.29). У даному сервісі відсутній пошук курсів за підкатегоріями, наприклад, користувач хоче пройти ІТ-курси по програмуванню на мові Python, але такої категорії немає, тому йому доведеться витратити більше часу для пошуку, прописуючи свій запит у пошуковій системі.

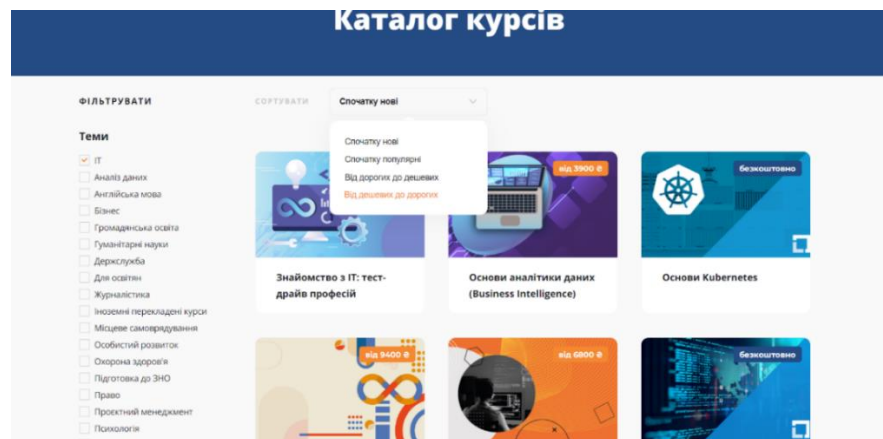


Рисунок 1.29 – «Prometheus.org.ua» - Каталог курсів

Перейшовши на сторінку одного з курсів, користувач може спостерігати детальний опис курсу та додаткову кнопку для реєстрації на курс (рис.1.30). Коли курс безкоштовний, процедура реєстрації відбувається миттєво.



Рисунок 1.30 – «Prometheus.org.ua» - Сторінка обраного курсу

Нижче на головній сторінці курсу, додатково до опису, додано відео для перегляду, де окрім тексту процес візуалізовано (рис.1.31). У правій частині сторінки відображено рекламу твіттеру сервісу, що створює зайвий інформаційний шум.

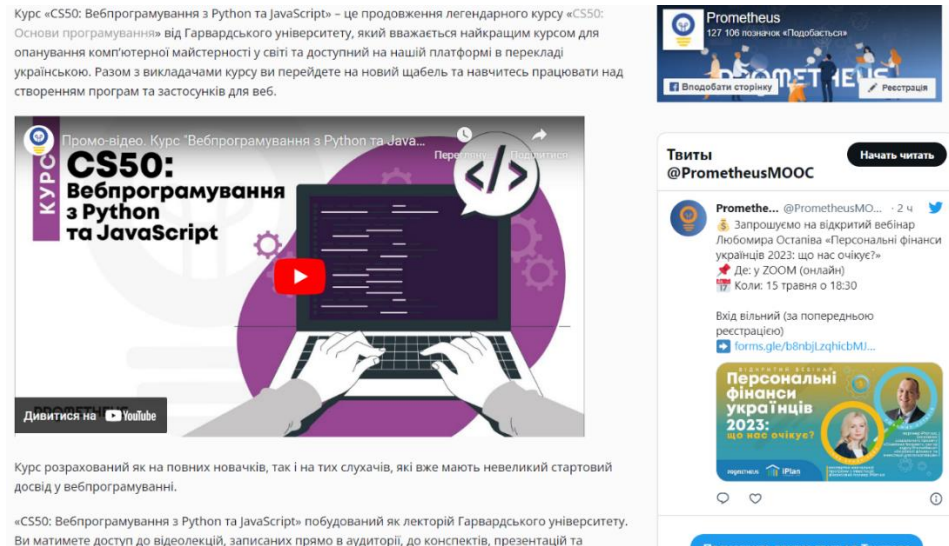


Рисунок – 1.31 – «Prometheus.org.ua» - Опис курсу з відеоматеріалом

У кінці сторінки продовжено детальний опис вмісту курсу та його переваги. Також можна побачити детальну інформацію про викладачів, а у кінці є відповіді на одні з найпопулярніших питань. Дане поле є важливим у інформуванні людей, так як там може міститись відповідь на їх питання, що може зберегти їх власний час. Також користувач бачить ліцензію курсу та термін, на який розраховано програму (рис.1.32).

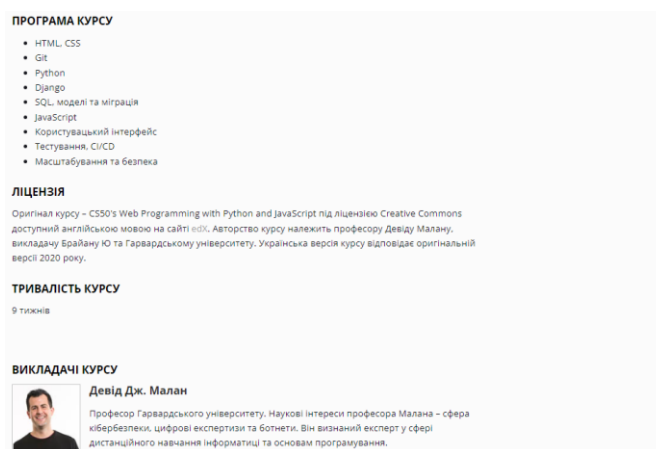


Рисунок 1.32 – «Prometheus.org.ua» - Детальний опис курсу

Отже, було проведено аналіз продуктів-аналогів, які використовуються для онлайн-навчання. Проаналізовано і у подробицях досліджено їх структуру та функціонал для подальшого планування розробки функціональної моделі та структури для кваліфікаційно роботи бакалавра.

1.3. Мета та задачі дослідження

За останні роки у сучасному світі робота та навчання онлайн повністю увійшли у наше життя у зв'язку з подіями, які відбуваються навколо нас. Для отримання нових знань та навичок не обов'язково особисто відвідувати заклади освіти.

Онлайн платформи з можливістю ведення курсів, занять та тренінгів набули великої популярності. Ними користуються не тільки спеціалісти, які вирішили ділитися власним досвідом, а і заклади освіти, що значно спрощує освітній процес.

Головна мета – створення інформаційної системи керування онлайн освітою. За допомогою даного сервісу можна буде створювати власні курси, а також проходити вже існуючі.

Для формування мети та задачі інформаційної системи керування онлайн освітою потрібно створити вимоги до функціоналу. Основний функціонал базується на тому, що за допомогою даного сервісу можна буде створювати власні курси, а також проходити вже існуючі. До переліку функцій системи відносяться:

1. Реєстрація та авторизація користувачів;
2. Створення дисципліни або курсу;
3. Видалення дисципліни або курсу;
4. Додавання учасників курсу;
5. Видалення учасників курсу;
6. Створення розкладу;
7. Редагування розкладу;
8. Додавання навчальних матеріалів;
9. Надсилання виконаних робіт;
10. Оцінювання робіт;
11. Автоматична генерація посилання на заняття через платформу ZOOM;
12. Редагування персональних даних;

Окрім того, при розробці інформаційної системи керування онлайн освітою буде виконуватися розподілення на користувачів, а саме викладачі, студентів та адміністраторів. Виконаємо розподіл функцій за групами користувачів.

Варіації функцій для викладачі:

- Додавання студентів до курсу;
- Видалення студентів з курсу;
- Створення розкладу проведення занять;
- Редагування розкладу проведення занять;
- Оцінювання надісланих робіт;
- Редагування створених дисциплін чи курсів;
- Видалення дисциплін чи курсів;

Варіації функцій для студентів:

- Перегляд дисциплін;
- Перегляд розкладу;
- Перегляд загального списку завдань з курсів та дисциплін;
- Надсилання виконаних робіт на оцінку;

Варіації функцій для адміністраторів:

- Блокування користувачів;
- Редагування загальної інформації;
- Редагування всіх дисциплін;

Отже, результатом роботи є розроблена структура інформаційної системи керування онлайн навчанням. Також був сформований список із типів користувачів та переліком відкритих функцій кожного.

2 ПОСТАНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ СИСТЕМИ

2.1. Структура та функції інформаційної системи керування онлайн навчанням

Наступним етапом є розроблення структури інформаційної системи керування онлайн навчанням. На рисунку 1.1 представлена мапа веб-сайту. Детальний опис кожної із сторінок представлений в таблиці 1.1.

Таблиця 1.1 – Таблиця із описом структури інформаційної системи

Назва сторінки	Опис
Форма реєстрації	<p>Форма складається з полів:</p> <ul style="list-style-type: none"> - Прізвище - Ім'я - Ім'я По-батькові - Дата народження - Контактний номер телефону - Роль - Email - Пароль користувача - Підтвердження паролю - Кнопка «Реєструватися»
Форма авторизації	<p>Форма складається з полів:</p> <ul style="list-style-type: none"> - Email - Пароль - Підтвердження паролю - Кнопка «Вхід» - Кнопка «Відновлення паролю»

Назва сторінки	Опис
Форма відновлення паролю	Форма складається з: <ul style="list-style-type: none"> - ПІБ користувача - Email на який зареєстровано користувача
Головна сторінка	Містить: <ul style="list-style-type: none"> Стрічка оновлень (для кожної ролі своя)
Дисципліни	Містить: <ul style="list-style-type: none"> - Список дисциплін у яких користувач бере участь - Створення нових дисциплін (для викладачів) - Вибір шкали оцінювання робіт - Кожен викладач сам вирішує якою шкалу оцінювання буде користуватися для зручності.
Створити дисципліну	Містить: <ul style="list-style-type: none"> - Назва дисципліни - Опис - Кнопка «Додати матеріали»
Сторінка обраної дисципліни	Містить: <ul style="list-style-type: none"> - Опис дисципліни - Завдання - Навчальні матеріали - Посилання на заняття за допомогою платформи ZOOM
Календар	Містить: <ul style="list-style-type: none"> - Візуальний інтерфейс з відображенням занять

Назва сторінки	Опис
	<ul style="list-style-type: none"> - Кнопка «Додати заняття» - Можливість видалити заняття/змінити час
Додати Заняття	<p>Містить:</p> <ul style="list-style-type: none"> - Обрати дисципліну для якої буде заняття - Дату заняття - Час проведення
Завдання	<p>Містить:</p> <ul style="list-style-type: none"> - Список всіх завдань з усіх дисциплін - Фільтр : по дисципліні/по даті - Кнопка «відправити роботу» (для студентів) - Кнопка «оцінити» для викладачів
Особиста сторінка	<p>Містить:</p> <ul style="list-style-type: none"> - Зазначені при реєстрації ПІБ - Номер телефону - Email - Фото профілю

Отже, провівши детальний аналіз структур продуктів-аналогів було розроблено загальну функціональну структуру інформаційної системи керування онлайн навчанням з детальним описом кожного розділу, на основі опису якого буде проводитися реалізація проекту.

2.2. Вибір засобів для програмної реалізації

Для створення фронт-енд частини проекту та оформлення його зовнішнього вигляду були використані наступні технології: HTML, CSS, JavaScript , зокрема, фреймворк Vue.js.

Нам відомо, що HTML є стандартизованим стилем гіпертекстової розмітки[9]. Вона застосовується для перегляду веб-сторінок у браузерях, які інтерпретують код файлу і відображають його у візуальному інтерфейсі на пристроях. За допомогою HTML можна створити відносно простий, але естетично оформлений документ. Дану технологію було обрано, як основний інструмент для створення основи та макету інформаційної системи керування онлайн навчанням.

Крім того разом із HTML використовується CSS для опису дизайну сторінки. Даний інструмент дозволяє налаштовувати стилі шрифтів, кольори елементів, розташування об'єктів та інше[10].

JavaScript є об'єктно-орієнтованою прототипною мовою програмування. Ця мова використовується для реалізації сценаріїв взаємодії користувача з серверною та інтерфейсною частинами веб-сторінки.

Vue.js є JavaScript-фреймворком, призначеним для створення інтерфейсів користувача[11]. Основною метою Vue.js є спрощення та надання готових рішень для швидкої розробки складних інтерфейсів за допомогою використання компонентів. Його легко налаштовувати та використовувати у проєктів, що дає можливість швидко отримати бажані результати.

Даний фреймворк виділяється швидкістю своєї роботи, що позитивно впливає на загальну роботу проєкту. Не менш важливою властивістю Vue.js є легкість у розгортанні та написанні невеликих додатків. Даний фреймворк ідеально підходить для розробки інформаційної системи керування онлайн навчання. Адже, код фреймворку має гарну читабельну здатність і однофайлові компоненти, що робить зручним можливість швидко та просто розібратися у коді та спрощує процес демонстрації його використання[12-13].

Для написання бекенд-частини проєкту були використані PHP та фреймворк Laravel.

Мову було обрано, як основу функціональної частини проекту. Даний вибір був сформований на основі того, що PHP є однією з найнадійніших для створення інформаційних систем[14]. Крім того, робота з базами даних на мові PHP є однією із найпростіших у порівняннях з іншими мовами, так як PHP має власні вбудовані модулі[15].

Laravel представляє PHP фреймворк з відкритим кодом. Головне його призначення - це розробка веб-сторінок за шаблоном Model-View-Controller[16]. Обраний фреймворк має найрізноманітніші плагіни та додатки. У сукупності доповнення допомагають у розгортанні додатків, технічному обслуговуванні, а також використанні різних баз даних. Даний фреймворк використовує велику варіативність шаблонів, які значно скорочують час написання коду[17]. Про це свідчить його використання при написанні різноманітних методів для реалізації функціоналу інформаційних систем, наприклад функціонал авторизації та реєстрації користувачів.

Крім того, Laravel сповнений значною кількістю об'єктно-орієнтованих бібліотек, які вагомо розширюють функціонал PHP. Дане розширення також дає можливість розширити функціонал веб-додатку. Також завдяки даному фреймворку можна реалізувати системну роботу черг повідомлень з метою підтримки балансу навантаження без втрати даних[18-19].

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Розробка дизайну інформаційної системи керування онлайн навчанням

Дизайн був розроблений в додатку Figma. Макети представленні у додатку на рисунку 1.2-5.

Figma використовує як інструмен в найновіших та найкращих практиках дизайну, таких як інтерфейсу користувача та дизайну користувацького досвіду чи іншого. У додатку Figma було розроблено дизайни сторінок інформаційної системи керування онлайн навчанням.

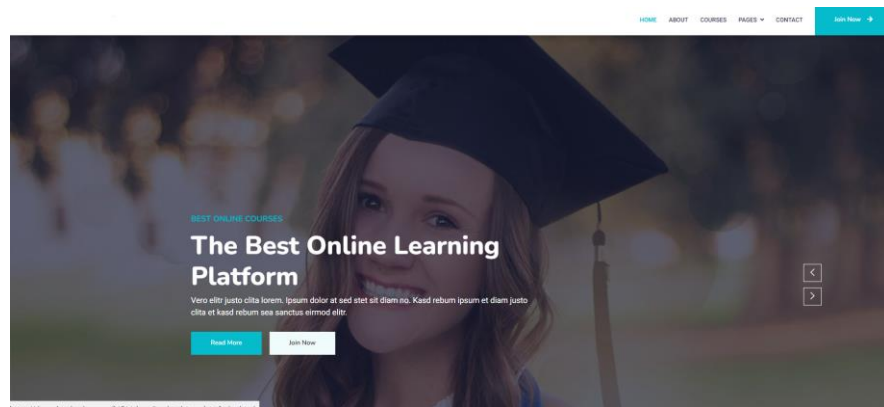


Рисунок 3.1 – Головна сторінка

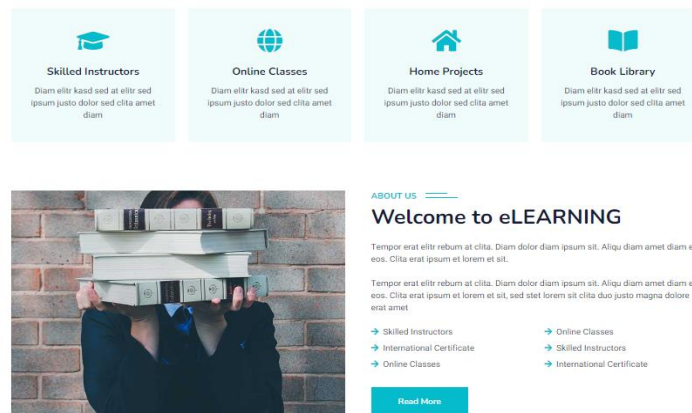


Рисунок 3.2 – Сторінка із додатковою інформацією



Рисунок 3.3 – Сторінка із курсами



Рисунок 3.4 – Сторінка із викладачами

Отже, завдяки інструментарію Figma було розроблено власний унікальний дизайн головних частин головної сторінки інформаційної системи керування онлайн навчанням відповідно до розробленого функціоналу для взаємодії з користувачем.

3.2 Програмна реалізація інформаційної системи керування онлайн навчанням

Програмна реалізація інформаційної системи керування онлайн-навчанням реалізовувалась із використання фреймворків Laravel та Vue.js. На рисунку 1.1 зображено пакети та модулі, які було створено в процесі реалізації проекту. Детальний код реалізації знаходиться у Додатку А.

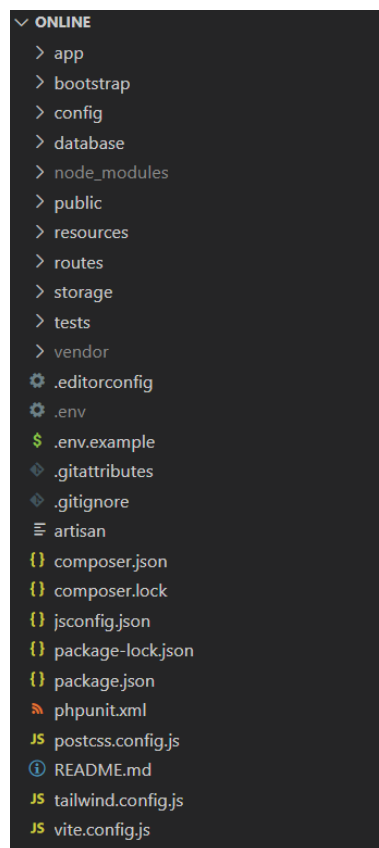


Рисунок 3.5 – Пакети та модулі інформаційної системи керування онлайн-навчанням

Також для інформаційної системи, згідно до розробленої моделі функціоналу було розроблено відповідну базу даних[20]. У кожній з таблиць будуть зберігатися дані про студентів, викладачів та курси. Ці дані будуть надалі

оброблюватися відповідно до використання функцій інформаційної системи керування онлайн-навчанням.

На рисунку 3.6 зображено ERD-діаграму, де зображено модель бази даних і її таблиці.

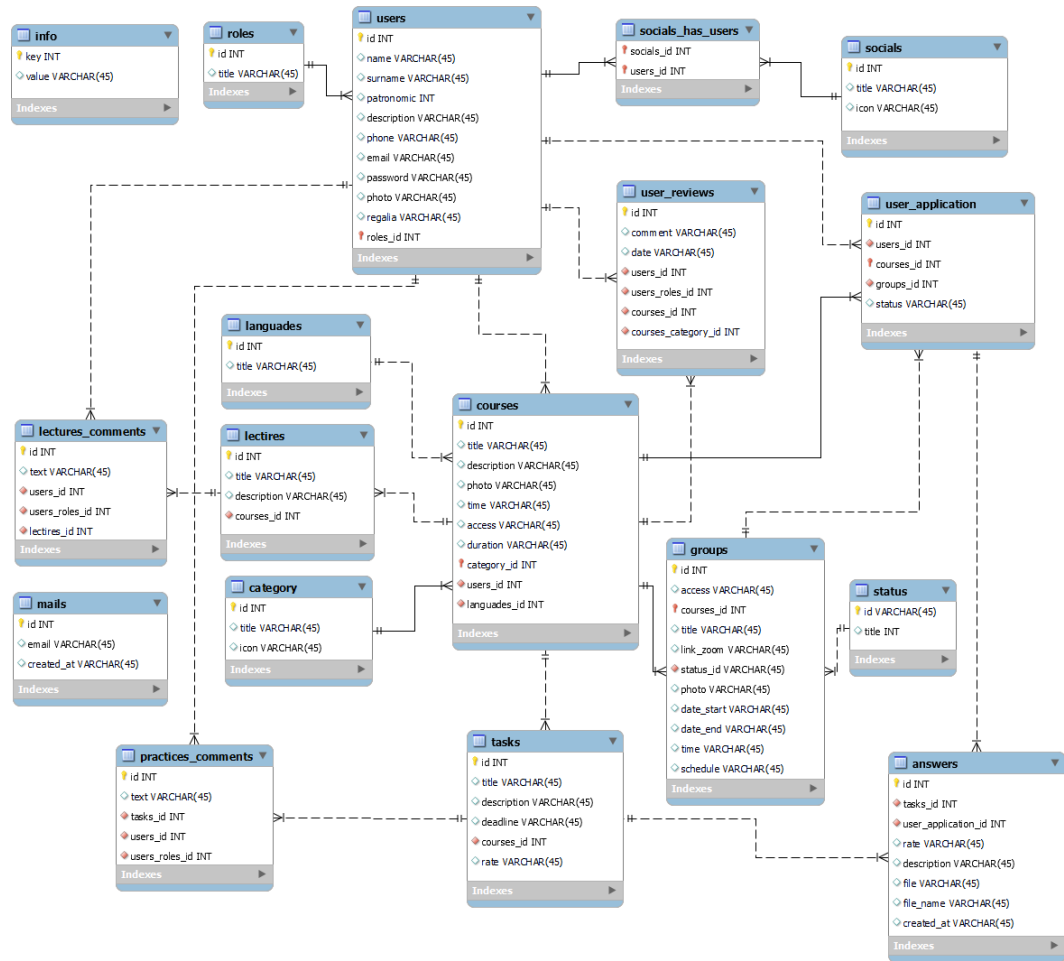


Рисунок 3.6 – ERD-діаграма бази даних інформаційної системи керування онлайн-навчанням

Отже, у реалізації роботи було розроблено відповідну до вимог та функціоналу базу даних, яка містить в собі 18 таблиць. Надалі ці таблиці будуть використовуватися для збереження в собі необхідних даних про користувачів, курси та викладачів.

3.3 Використання інформаційної системи керування онлайн навчанням зі сторони незареєстрованого користувача

Головна сторінка інформаційної системи являється інформативною зі здатністю попереднього перегляду певних сторінок, для цього у хедері активні додаткові кнопки для навігації для ознайомлення з окремими пунктами. Користувач одразу бачить банер з зображенням студентів-випускників навчальних закладів. Зображення з інформацією на банері можна гортати (рис.3.7).

Дана сторінка є основною для всіх незареєстрованих користувачів.

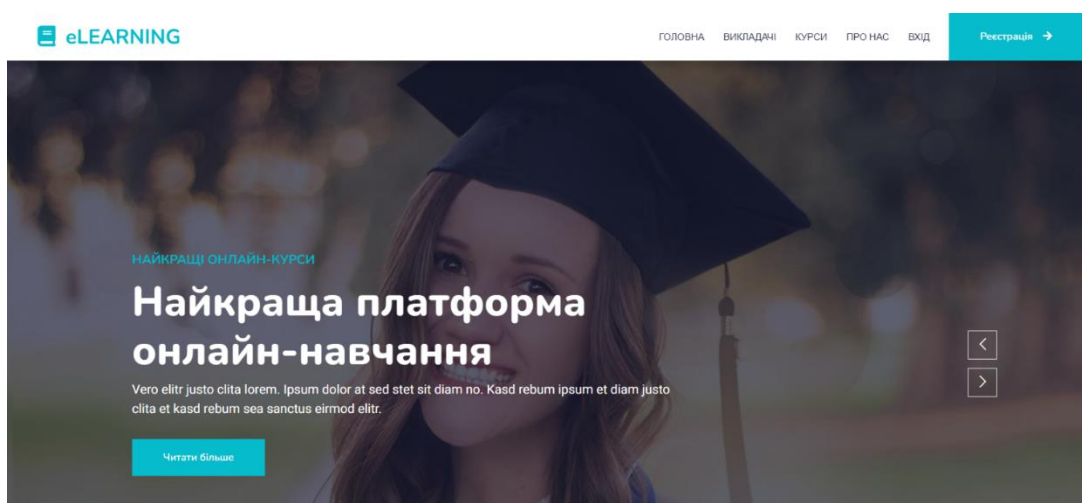


Рисунок 3.7 – Головна сторінка незареєстрованого користувача

Нижче на головній сторінці користувач бачить візуалізований фільтр основних категорій курсів, який зображено на рисунку 3.8. Також, коли користувач опускається по сторінці вниз, для зручної навігації у правому нижньому кутку екрану з'являється спеціальна кнопка, натиснувши на яку користувач знову опиниться на верхівці сторінки.



Рисунок 3.8 – Список категорій курсів на гол. Сторінці

На наступному секторі головної сторінки користувачу представлено візуалізований список курсів, де користувачу одразу відображено рейтинг даного курсу, його викладача та тривалість курсу. Також зазначено категорію, до якої відноситься курс (рис.3.9).

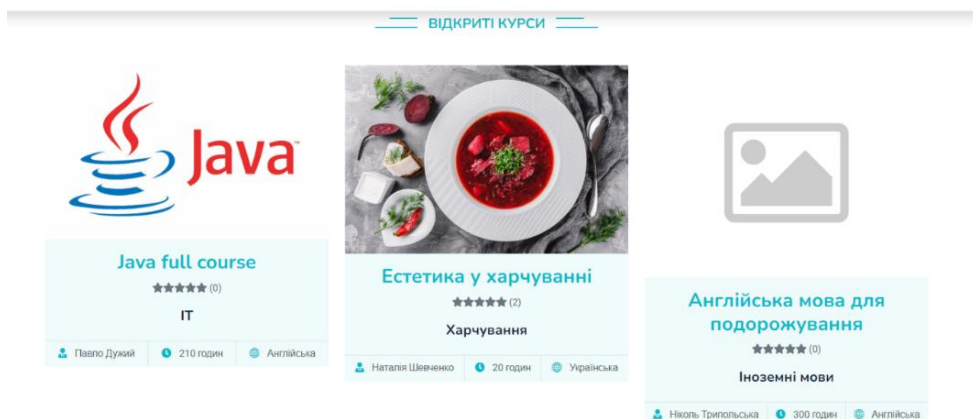


Рисунок 3.9 – Список доступних курсів

Незареєстровані користувачі мають змогу переходити на початкову сторінку переглядів курсів для ознайомлення з їх описом та відгуками про них (рис.3.10).



Рисунок 3.10 – Сторінка перегляду основної інформації про курси

Повернувшись назад на головну сторінку інформаційної системи і опустившись нижче, незареєстровані користувачі можуть бачити список викладів, які працюють на платформі, як зображено на рисунку 3.11, та можуть ознайомитися з їх персональними даними (рис.3.12).



Рисунок 3.11 – Перегляд викладачів на головній сторінці.

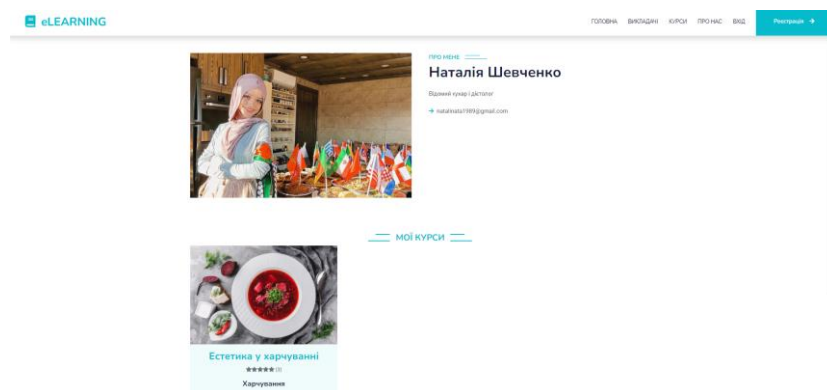


Рисунок 3.12 – Персональна інформація викладача

Повернувшись знову на головну сторінку далі незареєстрований користувач може спостерігати панель відгуків зареєстрованих учнів інформаційної системи для керування онлайн навчанням (рис.3.13).

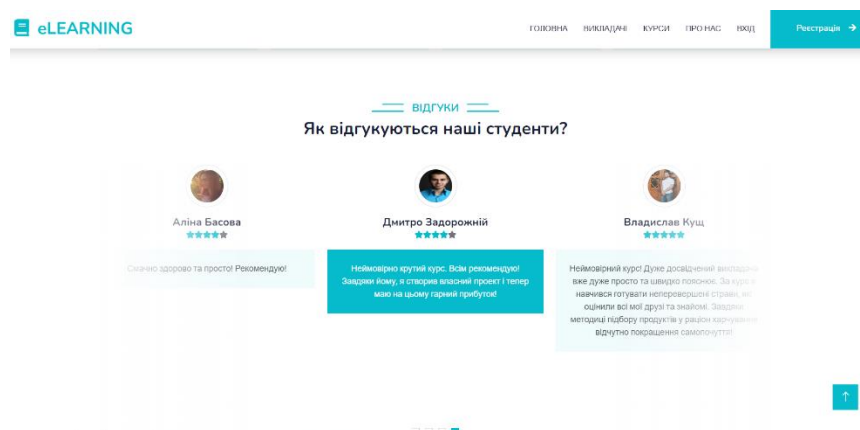


Рисунок 3.13 – Панель відгуків зареєстрованих студентів на головній сторінці

Додатково незареєстрований користувач може перейти по розділам з хедеру, який зображено на рисунку 3.14 інформаційної системи і перегляну всю інформацію з головної сторінки окремо.



Рисунок 3.14 – Панель навігації інформаційної системи

Перейшовши у розділ з навігації «викладачі» користувач бачить весь список викладачів на окремій сторінці, що робить пошук необхідного викладача спрощеним (рис.3.15).



Рисунок 3.15 – Розділ інформаційної системи «Викладачі»

Наступний розділ з навігації: «курси». Потрапивши у цей розділ користувач може бачити повний список доступних курсів і має змогу скористуватися пошуком за потрібним йому критерієм, у разі великої кількості курсів (рис.3.16).

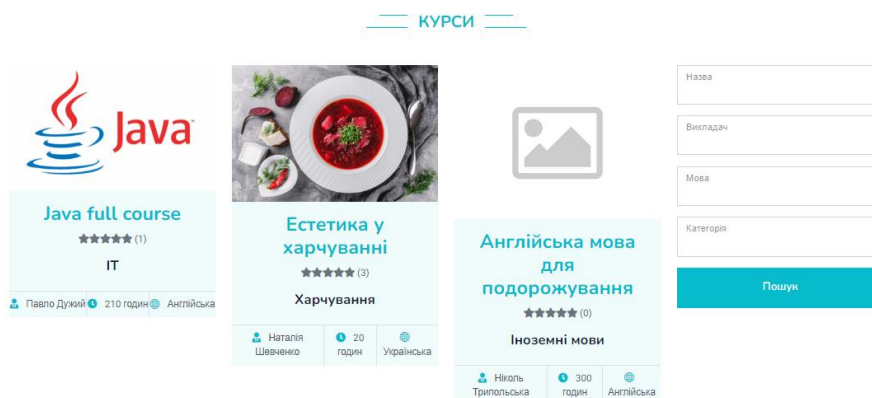



Рисунок 3.16 – Розділ інформаційної системи «Курси»


Останнім розділом з навігації є «про нас», який зображено на рисунку 3.17. Потрапляючи туди. Незареєстрований користувач має здатність прочитати інформацію про інформаційну систему та ознайомитись з її перевагами.

— ПРО НАС —


Ви щойно до нас завітали і хочете дізнатися у чому ж перевага у навчанні на нашій платформі? Тоді ось тобi перелік наших переваг:

-  **Доступність:** Наша платформа eLEARNING забезпечує доступ до навчання для широкого кола людей.


Вам потрібен лише комп'ютер або смартфон і доступ до Інтернету.

Немає обмежень щодо місця проживання або фізичних обмежень - ви можете навчатися з будь-якої точки світу.
 -  **Різноманітність матеріалів:** Тут ви знайдете широкий вибір навчальних матеріалів - від відеороликів і презентацій до інтерактивних модулів і тестів.

Ви можете вибрати формат, який найкраще підходить для вашого стилю навчання.

Ви можете вибрати формат, який найкраще підходить для вашого стилю навчання.
 -  **Індивідуалізоване навчання:** Наша платформа пропонує індивідуальний підхід до навчання.

Ви можете пройти тестування або оцінити свої знання, щоб отримати персоналізовані рекомендації щодо навчальних матеріалів і шляху навчання.

Таким чином, ви можете зосередитися на саме потрібних для вас темах і зміцнити свої знання.
 -  **Технологічна підтримка:** Наша команда завжди готова допомогти вам у разі виникнення проблем з платформою або технічних питань.


Ви можете звертатися до нас з будь-якими запитаннями або необхідністю додаткової допомоги.
- З eLEARNING ви отримуєте можливість навчатися з комфортом, розвивати свої навички та вдосконалювати знання. 

Рисунок 3.17 – Розділ інформаційної системи «про нас»

Далі користувач має змогу здійснити авторизацію (якщо у нього вже є створений власний профіль викладача або студента), де йому необхідно ввести тільки власні логін та пароль у формі авторизації, яку зображено на рисунку 3.18.

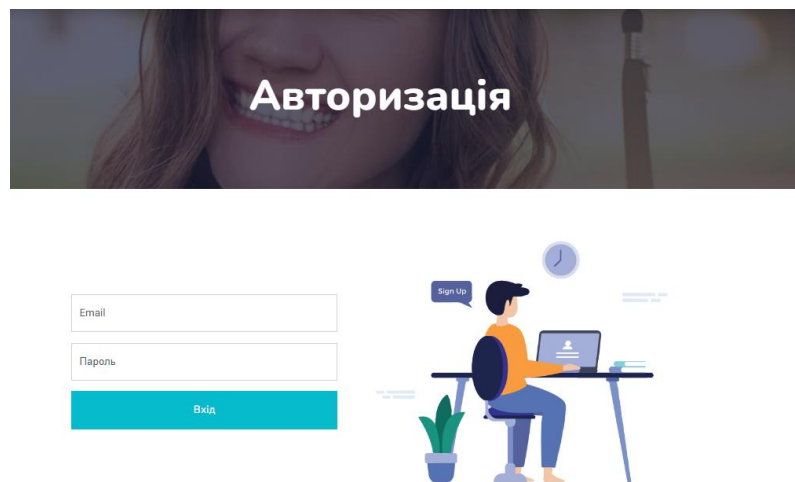
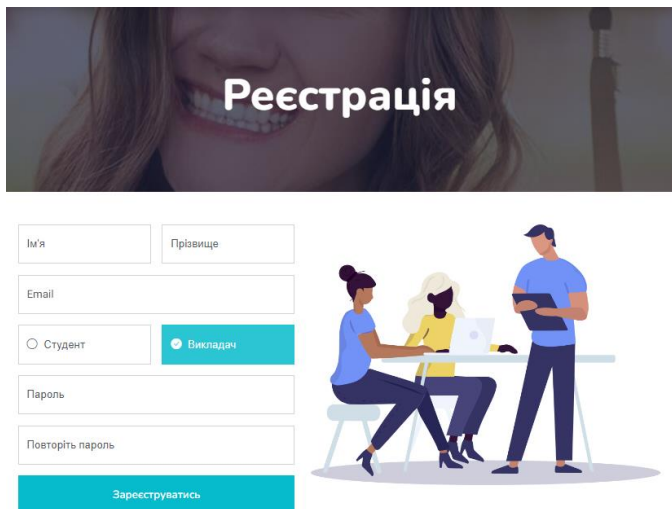


Рисунок 3.18 – Форма авторизації для усіх користувачів.

Для здійснення реєстрації власного профілю на інформаційній системі керування онлайн – навчанням користувачу необхідно перейти до розділу «реєстрація», після чого відкриється форма реєстрації, як на рисунку 3.19, і заповнити поля з особистою інформацією: ім'я, прізвище, e-mail, тип користувача (студент\викладач), пароль і підтвердження паролю.



The image shows a registration form titled "Реєстрація" (Registration) overlaid on a background image of a smiling woman. The form includes the following fields and elements:

- Имя (Name) and Прізвище (Surname) input fields.
- Email input field.
- Radio buttons for user type: Студент (Student) and Викладач (Lecturer).
- Пароль (Password) input field.
- Повторіть пароль (Repeat password) input field.
- A blue button labeled "Зареєструватись" (Register).

To the right of the form is an illustration of three people (two women and one man) sitting at a table, looking at a laptop screen.

Рисунок 3.19 – Форма реєстрації користувача

Отже, незареєстрований користувач має доступ до повного функціоналу, який передбачає собою перегляд основних даних про викладачів та курси, для того щоб мати можливість одразу підібрати йому необхідний. Також продемонстровано роботу авторизації та реєстрації користувача.

3.4 Використання інформаційної системи керування онлайн навчанням зі сторони студента

Користувач, який вже попередньо зареєстрував власний профіль та провів авторизацію у інформаційній системі, то він автоматично потрапляє у розділ «мої групи», де відображено список груп курсів, на які користувач зареєструвався (рис.3.20). У випадку, якщо користувач ще не обрав собі курсів,

список буде відповідно порожнім, як на рисунку 3.21. Також, для зручності, користувачу доступна додаткова бокова панель навігації.

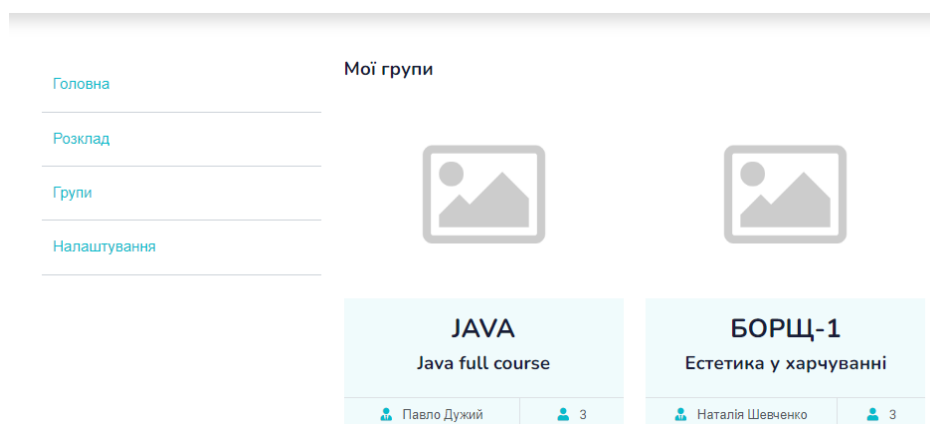


Рисунок 3.20 – Головна сторінка авторизованого студента

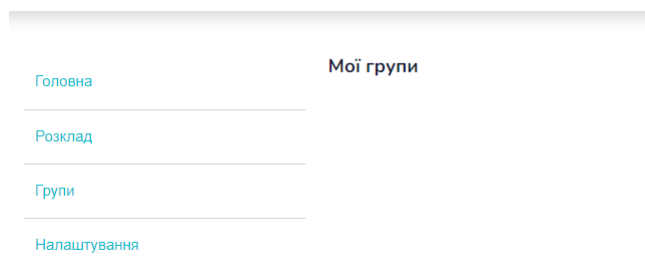


Рисунок 3.21 – Сторінка «Мої групи» для авторизованого студента без груп

Кнопка «розклад» відкриває студенту сторінку з розкладом на візуальному активному календарі, який зображено на рисунку 3.22. Користувач бачить на яку дату, день неділі та групу у якій назначено заняття.

Головна **червень 2023 р.** Сьогодні < >

нд	пн	вт	ср	чт	пт	сб
28	29	30	31	1	2	3
		18:40 - JAVA			18:10 - БОРЦІ-1	
4	5	6	7	8	9	10
		18:40 - JAVA	18:40 - JAVA	18:40 - JAVA	18:10 - БОРЦІ-1	
11	12	13	14	15	16	17
	18:40 - JAVA	18:40 - JAVA		18:40 - JAVA	18:10 - БОРЦІ-1	
		18:40 - JAVA				
18	19	20	21	22	23	24
18:40 - JAVA		18:40 - JAVA		18:40 - JAVA	18:10 - БОРЦІ-1	
		18:40 - JAVA				
25	26	27	28	29	30	1
		18:40 - JAVA		18:40 - JAVA		
2	3	4	5	6	7	8
	18:40 - JAVA	18:40 - JAVA	18:40 - JAVA	18:40 - JAVA		
		18:40 - JAVA	18:40 - JAVA			

Рисунок 3.22 – Панель розкладу для студента

Натиснувши на іконку заняття, з'явиться окрема панель, до користувачу буде детально представлено інформацію та кнопка «підключитись» яка автоматично пересилає користувача на платформу «ZOOM» де буде проводитися заняття (рис.3.23-24).

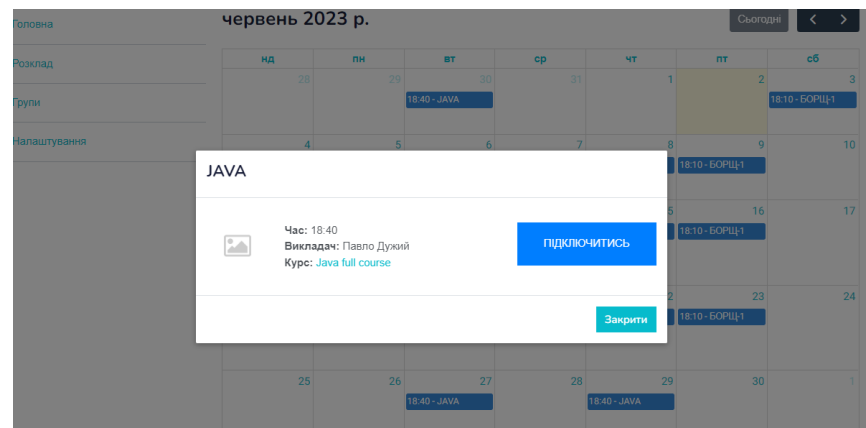


Рисунок 3.23 – Панель з інформацією та посиланням на заняття групи.

Click **Open Zoom Meetings** on the dialog shown by your browser

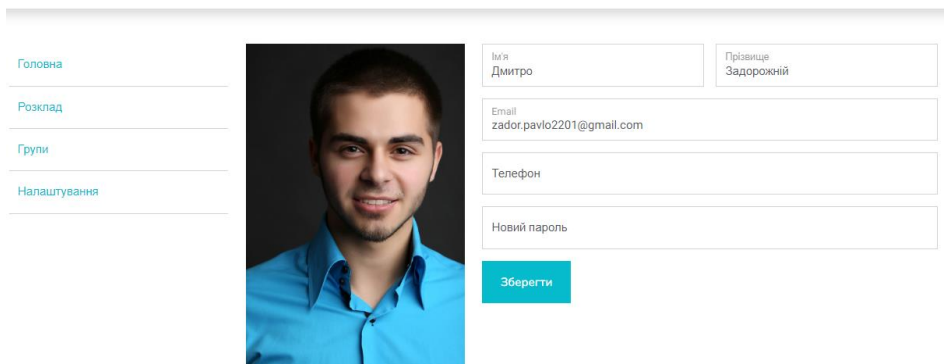
If you don't see a dialog, click **Launch Meeting** below

By joining a meeting, you agree to our [Terms of Service](#) and [Privacy Statement](#)

Launch Meeting

Рисунок 3.24 - Запуск конференції на платформі «ZOOM» за посиланням

Кнопка «налаштування» відкриває користувачу панель налаштувань профілю, де він може змінити власні персональні дані, такі як: фото профілю; ім'я; прізвище; e-mail; номер телефону; пароль, як на рисунку 3.25.



Головна	Ім'я Дмитро	Прізвище Задорожній
Розклад	Email zador.pavlo2201@gmail.com	
Групи	Телефон	
Налаштування	Новий пароль	
	Зберегти	

Рисунок 3.25 – Панель «Налаштування» профілю користувача

Для авторизованих користувачів у хедері інформаційної системи керування онлайн – навчанням з'являється іконка її профілю, де вони також можуть скористуватися навігацією та здійснити вихід зі свого профілю (рис.3.26).

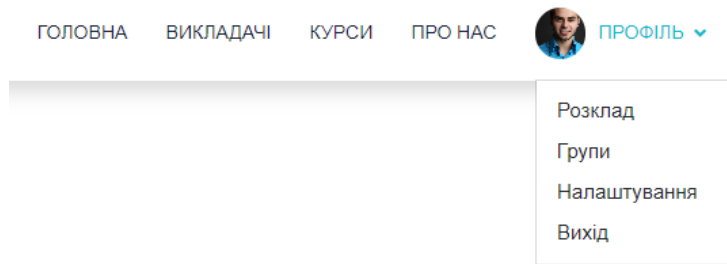


Рисунок 3.26 - Іконка профілю користувача у хедері інформаційно системи

Якщо користувач хоче обрати собі курс, він має перейти у розділ «курси» з навігації у хедері і обрати курс, який йому підходить. На відміну від неавторизованого користувача, у студента з'являється кнопка «Записатись», яку зображено на рисунку 3.27, яка в свою чергу відправить викладачу заявку на реєстрацію на даний курс.

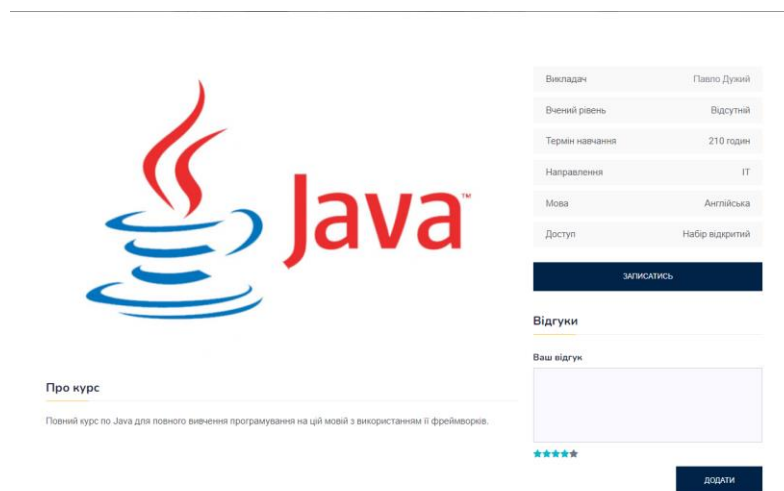


Рисунок 3.27 – Кнопка «Записатись» на курс для користувача – студента

Коли користувач вже зареєстрований на курс, і він переходить на нього з головної сторінки після авторизації, він потрапляє на робочу сторінку курсу,

яку зображено на рисунку 3.28, де користувач бачить доступні лекції, практичні завдання та членів його групи.

Рисунок 3.28 – Робоча сторінка курсу - лекції

Перейшовши до розділу «практичні», студент бачить список доступних практичних завдання: їх назва та статус перевірки. Обравши одне із них відкриється опис завдання, та кнопка «перейти до завдання» (рис.3.29).

Рисунок 3.29 – Робоча сторінка курсу – практичні завдання

Перейшовши до завдання користувач потрапляє на сторінку відповідної практичної роботи, як зображено на рисунку 3.30, де буде також розписано опис завдання, час його здачі та поле для заповнення відповіді у текстовому

форматі, кнопка для прикріплення додаткового файлу, якщо такий необхідний та кнопка «здати» або «перездати» в залежності від того який раз студент надсилає роботу. Також, якщо викладач вже перевірів роботу, оцінка студента буде відображатися над кнопкою «Прикріпити файл».

Практична 1
Павло Дужий 25.05.2023

Оцінка: 10

Час виконання: 20:30
Створити власний простий проект за прикладом у лекції

Відповідь

```
class Animal():
    def __init__(self, name):
        self.name=name
    def breathe(self):
        print(self.name, 'дихає')
    def eat_food(self,food):
        print(self.name, 'ємь', food)
class Dog (Animal):
    def guard(self):
        print(self.name, 'охороняє')
class Cat (Animal):
    def catchmouse(self):
        print(self.name, 'ловить мишу')
dog1 = Dog("Орко ")
cat1 = Cat("Мурчик")
cat1.breathe() #
cat1.eat_food (" рибку")
dog1.eat_food (" кістку")
dog1.guard() # відправляємо собаку охороняти
cat1.catchmouse() # відправляємо kota на полювання
```

Коментарі

Рисунок 3.30 – Розширена сторінка практичного завдання

Кожен студент може переглянути список студентів, які також проходять з ним курс у одній групі (рис.3.31).

JAVA
Статус: Йде набір
3 Викладач: Павло Дужий
Курс: Java full course

Лекції Практичні Група

#	ПІБ	Email	Телефон
1	Владислав Куц	v.kushch@gmail.com	
2	Yehor Ivashchenko	ivasthenko.egor109bbc@gmail.com	
3	Аліна Басова	alinka.bas2004@gmail.com	

Рисунок 3.31 – Робоча сторінка курсу – група

Отже, авторизований користувач може використовувати всі реалізовані функції інформаційної системи керування онлайн-навчанням і може здійснювати, пошук курсів, записуватися на них та виконувати завдання від викладачів, які викладено у курсі.

3.5 Використання інформаційної системи керування онлайн навчанням зі сторони викладача

Авторизований викладач потрапляє на головну сторінку (рис.3.32) після авторизації, на якій відображено його групи та список надісланих звітів його студентів, для загального моніторингу звітності, де також відображено, що звіт не перевірено. Також до панелі навігації додається розділ «заявки» у яких викладача може бачити нові заявки від студентів на реєстрацію на курс та «Мої курси» для їх керування.

The screenshot shows a web interface for an instructor. On the left is a navigation menu with items: Головна, Розклад, Заявки, Групи, Мої курси, and Налаштування. The main content area is titled 'Мої групи' and features a profile card for a course named 'JAVA' (Java full course) by 'Павло Духий' with 3 participants. Below this is a table titled 'Звіти студентів' with the following data:

#	ПІБ	Група	Завдання	Дата	Оцінка
1	Владислав Куц	JAVA	Практична 1	25.05.2023	10
2	Аліна Басова	JAVA	Практична 1	01.06.2023	Не перевірено

Рисунок 3.32 – Головна сторінка авторизованого викладача

Перейшовши у групу з головної сторінки, викладач бачить основну інформацію про курс: його статус (йде набір\закритий), його викладач та курс, до якого відноситься група, список студентів, які проходять курс, та звіти, які вони

надсилали (рис.3.33-34) з можливістю перевірки звіту та виставлення оцінки за роботу (рис.3.35). Додатково викладач може видаляти студентів із групи.

The screenshot shows the 'JAVA' course page. On the left is a navigation menu with items: Головна, Розклад, Заявки, Групи, Мої курси, and Налаштування. The main content area has a header for 'JAVA' with status 'Йде набір', instructor 'Павло Дужий', and course 'Java full course'. Below the header are tabs for 'Студенти' and 'Звіти'. A table lists three students with their IDs, names, emails, and phone numbers. Each row has a red trash icon for deletion.

#	ПІБ	Email	Телефон
1	Владислав Куц	v.kushch@gmail.com	
2	Yehor Ivashchenko	ivasthenko.egor109bbc@gmail.com	
3	Аліна Басова	alinka.bas2004@gmail.com	

Рисунок 3.33 – Головна сторінка групи – список студентів

This screenshot shows the 'Звіти' (Reports) tab selected. The table lists reports submitted by students, including student ID, name, task name, submission date, and score.

#	ПІБ	Завдання	Дата	Оцінка
1	Владислав Куц	Практична 1	25.05.2023	10
2	Аліна Басова	Практична 1	01.06.2023	0

Рисунок 3.34 – Головна сторінка групи – список звітів студентів

The screenshot shows the 'Практична 1' (Practical 1) report review page for student 'Аліна Басова'. The page displays the score 'Оцінка: 5' with a progress bar. Below the score, there is a 'Відповідь' (Answer) section with the text 'TEST' and a 'Коментарі' (Comments) section with a text input field and a 'Відправити' (Send) button.

Рисунок 3.35 – Перевірка звіту студента

Повернувшись на головну сторінку викладача та перейшовши до розділу «Заявки» викладач бачить список заявок які надходили до нього на реєстрацію

для проходження курсу. Викладач має змогу прийняти заявку та визначити групу, до якої буде зараховано студента, як зображено на рисунках 3.36 - 37. Також заявку можна відмінити.

#	ПІБ	Курс	Статус	Дата	Група	Дія
1	Владислав Куц	Java full course	Прийнято	25.05.2023	JAVA	✖
2	Yehor Ivashchenko	Java full course	Прийнято	25.05.2023	JAVA	✖
3	Аліна Басова	Java full course	Прийнято	25.05.2023	JAVA	✖
4	Дмитро Задорожній	Java full course	Відхилено	01.06.2023		✔

Рисунок 3.36 – Розділ «Заявки» - список заявок

#	ПІБ	Курс	Статус	Дата	Група	Дія
1	Владислав Куц	Java full course	Прийнято	25.05.2023	JAVA	✖
2	Yehor Ivashchenko	Java full course	Прийнято	25.05.2023	JAVA	✖
3	Аліна Басова	Java full course	Прийнято	25.05.2023	JAVA	✖
4	Дмитро Задорожній	Java full course	Прийнято	01.06.2023	Оберть групу	✖

Рисунок 3.37 – Розділ «Заявки» - прийняття заявки студента та вибір групи

Перейшовши до розділу групи, викладачу представлено список його груп, де відображено основну інформацію про них: назва, курс, статус, кількість студентів, дата створення. Викладачу надаються такі функції: додати нову групу, редагувати існуючу, видалити існуючу (рис.3.38).

#	Назва	Курс	Статус	Студентів	Дата створення
1	JAVA	Java full course	Йде набір	3	25.05.2023

Рисунок 3.38 – Розділ «Групи»

Натиснувши на «Додати групу», відкриється форма створення нової групи, яку зображено на рисунку 3.39, де викладач буде вносити всі необхідні дані у відповідні поля для формування групи. Необхідна інформація для створення групи: назва групи, за бажанням – фото для групи, обрати курс до якого буде належати група, за допомогою календаря обрати тривалість курсу, час проведення занять, за допомогою календаря відмітити дні у які будуть проводитися заняття.

Рисунок 3.39 – Форма створення нової групи

Перейшовши до редагування існуючої групи, викладач може змінити всі параметри та поля, що заповнюються при створенні групи, як на рисунку 3.40.

Рисунок 3.40 – Редагування існуючої групи

Перейшовши у розділ «Мої курси», що зображено на рисунку 3.41, перед викладом розгортається список його власних курсів з інформацією про них: назва, категорія, мова, доступ, тривалість. Доступний функціонал: створити новий курс, редагувати існуючий курс, видалити існуючий курс, заповнення курсу.



#	Назва	Категорія	Мова	Доступ	Тривалість	
1	Java full course	ІТ	Англійська	Відкритий	210 годин	 

Рисунок 3.41 – Розділ «Мої курси»

Натиснувши «Додати курс» викладач переходить до форми створення нового курсу, де йому треба заповнити відповідні поля, такі як: назва, категорія, мова курсу, тривалість курсу, відкритий/закритий курс, опис курсу, за бажанням фото для курсу (рис.3.42).

Головна

Розклад

Заявки

Групи

Мої курси

Налаштування

Заголовок
TEST

Категорія
ІТ

Мова
Українська

Тривалість
TEST

Закритий

Відкритий

Paragraph

TEST

Зберегти

Рисунок 3.42 – Форма створення нового курсу

Перейшовши до редагування існуючого курсу, викладач може змінити всі дані про курс, які зазначалися при його створенні (рис.3.43).

Головна

Розклад

Заявки

Групи

Мої курси

Налаштування

Java

Заголовок
Java full course

Категорія
ІТ

Мова
Англійська

Тривалість
210 годин

Закритий

Відкритий

Paragraph

Повний курс по Java для повного вивчення програмування на цій мові з використанням її фреймворків.

Зберегти

Рисунок 3.43 – Форма редагування існуючого курсу

Перейшовши за самим курсом, відкриється робоча сторінка курсу для викладача, як на рисунку 3.44, де буде відображено групи, які підключені до даного курсу, лекції та практичні завдання.

The screenshot shows a course management interface for 'Java full course'. On the left is a navigation menu with items: Головна, Розклад, Заявки, Групи, Мої курси, and Налаштування. The main header includes the Java logo and course details: Назва: Java full course, Доступ: Відкритий, and Тривалість: 210 годин. Below the header are three tabs: Групи (selected), Лекції, and Практичні. A table displays the 'Groups' section with the following data:

#	Назва	Студентів
1	JAVA	0

A red trash icon is visible at the end of the first row in the table.

Рисунок 3.44 – Робоча сторінка курсу для викладача – групи

У розділі «Лекції» викладачу представлено список існуючих лекцій з можливістю їх редагування, видалення, перегляду та додавання нових лекцій (рис.3.45-47).

The screenshot shows the same course management interface, but with the 'Lectures' tab selected. The table now displays the 'Lectures' section with the following data:

#	Назва	Дата створення
1	Лекція 1	25.05.2023

At the bottom of the table, there is a blue 'Додати' (Add) button and a red trash icon.

Рисунок 3.45 – Робоча сторінка курсу викладача – список лекцій

Головна	Лекція 1	Павло Дужий	25.05.2023
Розклад	Поняття класу та об'єкту є основою об'єктно-орієнтованого програмування (ООП). Програма створюється як деяка сукупність об'єктів, які взаємодіють між собою. ООП базується на трьох основних принципах: інкапсуляція , наслідування та поліморфізм . Розглянемо сутність цих принципів.		
Заявки	Об'єкти дозволяють об'єднувати функції (методи класу) та дані (змінні, що зберігають значення атрибутів) всередині одного елемента. Атрибути являють собою інформацію, а методи – дії.		
Групи	ОБ'ЄКТ = АТРИБУТИ + МЕТОДИ		
Мої курси	Наприклад, ми описали клас Balloons, що представляє повітряні кульки, який містить дані про кульку: колір, висота, «надутість», і методи: змінити значення атрибутів в разі протикання кульки, збільшити розміри кульки тощо. При цьому клас стає новим типом даних в програмі, що створюється, і ми можемо створювати змінні цього нового типу.		
Налаштування	Розглянемо такі поняття, як людина та учень. Будь-яка людина має ім'я, прізвище, вік та інші загальні характеристики. Учень також має ім'я, прізвище, вік, але, крім цього, він має додаткові характеристики: навчальний заклад, клас, середній бал успішності. З точки зору наслідування, учень є нащадком поняття людина. Таким чином, клас-нащадок містить ті самі атрибути та методи, що й базовий клас, але при цьому його можна розширити через додавання нових методів та атрибутів. Це позбавляє необхідності починати опис класу-нащадку «з нуля» – описувати атрибути й методи, успадковані від базового класу.		

Приклад:

На основі базового класу Тварина створити 2 класи: Собака і Кіт, що мають власні методи.

```
class Animal():
    def __init__(self, name):
        self.name=name
    def breathe(self):
        print(self.name+'Avvae')
```

Рисунок 3.46 – Перегляд лекції

The screenshot shows a web interface for editing a lecture. On the left is a sidebar with navigation links: Головна, Розклад, Заявки, Групи, Мої курси, and Налаштування. The main content area is a rich text editor with a toolbar containing options for Paragraph, text formatting (bold, italic, underline), alignment, link, list, indent, and media insertion. The editor content is identical to the text in Figure 3.46, including the definition of class Animal and the example code.



Рисунок 3.47 – Редагування існуючої лекції

Натиснувши на кнопку для додавання лекції, викладачу відкривається форма для створення лекції(рис.3.48): заповнюється назва лекції та її вміст.

Викладач має змогу представити лекцію у текстовому форматі з використанням мультимедійних додатків у виглядів фото, відео. Для зручності поле з заповненням тексту для лекції підтримує розширене форматування тексту.

Рисунок 3.48 – Створення нової лекції.

У розділі «практики» викладачу представлено список існуючих практик з можливістю їх редагування, видалення, перегляду та додавання нових лекцій (рис.3.49-51).

#	Назва	Дата створення	
1	Практична 1	25.05.2023	 

[Додати](#)

Рисунок 3.49 – Робоча сторінка курсу викладача – список практик

Рисунок 3.50 – Перегляд практичного завдання

Рисунок 3.51 – Редагування існуючого практичного завдання

Натиснувши на кнопку для додавання практик, викладачу відкривається форма для створення практичного завдання (рис.3.52). Для створення практичного завдання, викладачу необхідно заповнити відповідні поля: назва практики, опис її завдання, максимальний бал за виконання, час для здачі завдання.

Рисунок 3.52 – Створення нового практичного завдання

Перейшовши до розділу «Налаштування» (рис.3.53) викладач може редагувати дані свого профілю а також додати посилання на свої соціальні мережі, щоб у розділі «викладачі» студенти та неавторизовані користувачі мали змогу перейти за цим посиланням.

Рисунок 3.53 – Редагування особистих даних викладача

Отже, авторизований викладач може використовувати повний реалізований функціонал інформаційної системи керування онлайн-навчанням і створювати курси, додавати у них матеріали для навчання та завдання, а також назначати онлайн-зустрічі для проведення занять зі студентами та оцінювати їх роботи.

3.6 Використання інформаційної системи керування онлайн навчанням зі сторони адміністратора

Адміністратор переходить за окремим посиланням для авторизації у панель адміністратора. Для входу йому необхідно ввести необхідні дані у вигляді електронної пошти та паролю у формі авторизації, яку зображено на рисунку 3.54.

Рисунок 3.54 – Форма авторизації у панелі адміністратора

Після здійснення авторизації, адміністратору стає доступна панель редагування даних інформаційної системи керування онлайн-навчанням. Головна сторінка – редагування головного тексту з розділу інформаційної системи «Про нас» (рис.3.55).

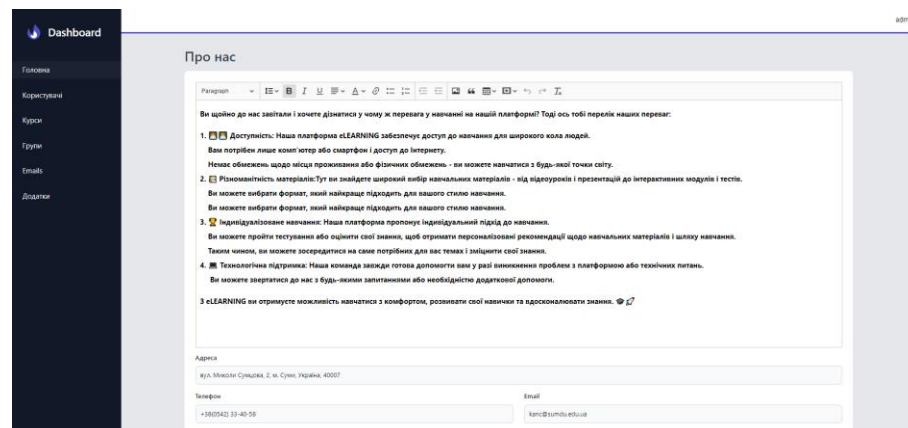


Рисунок 3.55 – Редагування розділу інформаційної системи «Про нас»

Адміністратору додатково доступна функція перегляду користувачів та можливість їх блокування, як на рисунку 3.56.

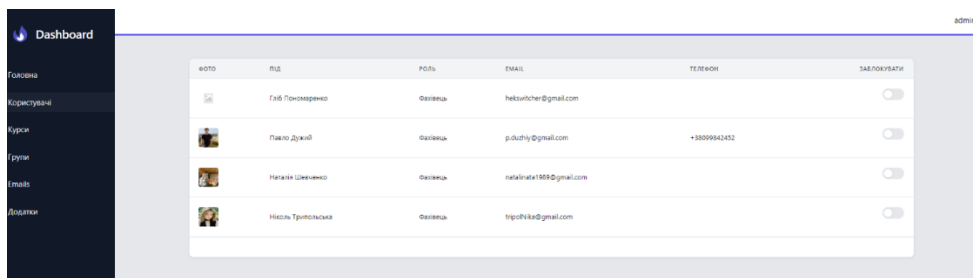


Рисунок 3.56 – список користувачів у панелі адміністратора

Також, як зображено на рисунку 3.57, адміністратор має змогу видаляти курси.

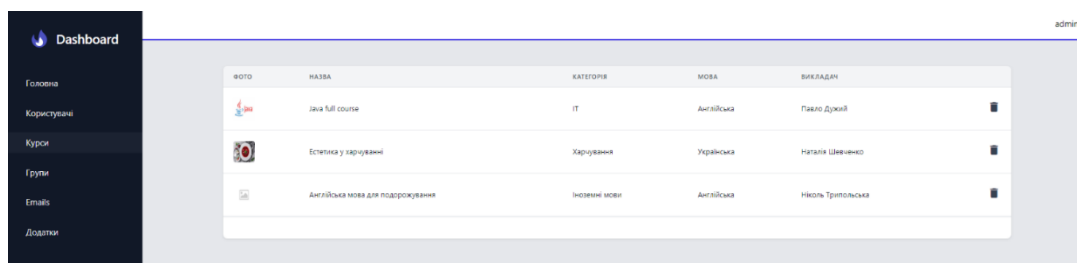


Рисунок 3.57 – Перегляд курсів у панелі адміністратора

З часом може виникнути необхідність у створенні додаткових категорій курсів, з можливістю редагування або додавання візуальних іконок для вже існуючих. Такий функціонал доступний у панелі адміністратора у розділі «додатки», який зображено на рисунку 3.58.

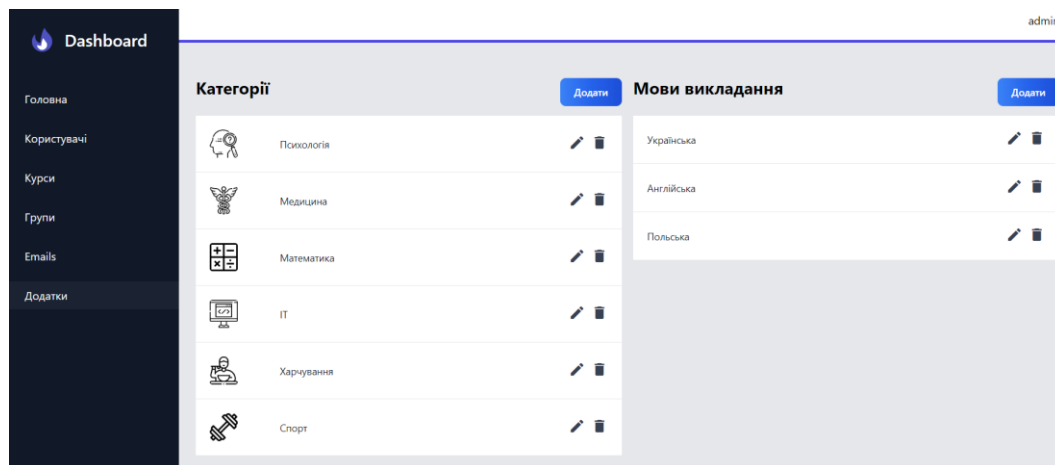


Рисунок 3.58 – Розділ «додатки»

ВИСНОВКИ

Під час виконання та розробки інформаційної системи керування онлайн-навчанням було проведено дослідження та аналіз актуальних веб-додатків та сайтів у освітньому напрямку. Завдяки аналізу було виявлено переваги та недоліки.

Реалізовано функціонал, який дозволяє використовувати одну інформаційну систему як для пошуку курсів, та їх проходження з можливістю виконання завдань та організації зустрічей без використання сторонніх застосунків.

При виконанні кваліфікаційної роботи було поставлено чітку мету створити інформаційну систему для керування онлайн освітою. При виконанні аналізу було підібрано програмне забезпечення та інструменти завдяки яким буде відбуватися реалізація.

Було продемонстровано роботу інформаційної системи керування онлайн-навчанням у повному обсязі у ролі неавторизованого користувача, викладача та адміністратора.

Результатом кваліфікаційної роботи бакалавра є розроблена інформаційна система керування онлайн-навчанням. Практичне призначення даної інформаційної системи являє собою спрощення організації та навчання онлайн. Це дає змогу користувачам створювати власні курси і ділитися своїми знаннями зі студентами, або набувати нових знань та навичок.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Online Learning in 2023 - What Is It And How Does It Work? [Електронний ресурс] – <https://www-educations-com.google.com/study-guides/study-online/>
2. THE BENEFITS OF ONLINE EDUCATION IN A VIRTUAL CLASSROOM [Електронний ресурс] – <https://drexel.edu/soe/resources/student-teaching/advice/benefits-of-online-and-virtual-learning/>
3. The Benefits of Online Learning: 7 Advantages of Online Degrees [Електронний ресурс] – <https://www.northeastern.edu/graduate/blog/benefits-of-online-learning/>
4. Benefits of Online Learning – 16 Advantages to Learning Online [Електронний ресурс] – <https://www.usa.edu/blog/benefits-of-online-learning/>
5. Types of Online Learning [Електронний ресурс] – <https://www.fordham.edu/about/leadership-and-administration/administrative-offices/office-of-the-provost/provost-office-units/online-learning/types-of-online-learning/>
6. Classroom [Електронний ресурс] – <https://classroom.google.com/>
7. Udemy [Електронний ресурс] – <https://www.udemy.com/>
8. Prometheus [Електронний ресурс] – <https://prometheus.org.ua/>
9. Advantages of HTML [Електронний ресурс] – <https://www.scaler.com/topics/advantages-of-html/>
10. Advantages of CSS [Електронний ресурс] – <https://www.tutorialspoint.com/What-are-the-advantages-of-CSS>
11. Why Vue.js [Електронний ресурс] – <https://www.sam-solutions.com/blog/why-vue-js/>
12. Benefits of Vue.js [Електронний ресурс] – <https://www.tatvasoft.com/outsourcing/2021/10/what-is-vue-js-and-its-benefits.html>

13. BrainMobi, Advantages of Using Vue.JS For Your Web Applications. Medium (2018).
14. Advantages of PHP [Электронный ресурс] - <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-php/>
15. PHP benefits [Электронный ресурс] - <https://anywhere.epam.com/business/pros-and-cons-of-php>
16. Laravel framework benefits [Электронный ресурс]- <https://www.netsolutions.com/insights/laravel-framework-benefits/>
17. M. I. Kausar Bagwan, P. D. Swati Ghule, A Modern Review on Laravel-PHP Framework. IRE Journals. 2, 1–3 (2019).
18. Why Laravel [Электронный ресурс] - <https://www.fastfwd.com/why-choose-laravel-for-your-next-web-project/>
19. R.Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, W. Ramadhan, Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website. EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi. 11, 48 (2021).
20. J. Shute et al., F1: A distributed SQL database that scales. Proceedings of the VLDB Endowment. 6, 1068–1079 (2013).

ДОДАТОК А. ЛІСТИНГ КОДУ

AuthenticatedSessionController.php

Контролер авторизації.

```
<?php
```

```
namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Http\Requests\Auth\LoginRequest;
use App\Providers\RouteServiceProvider;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Route;
use Inertia\Inertia;
use Inertia\Response;

class AuthenticatedSessionController extends Controller
{
    /**
     * Display the login view.
     */
    public function create(): Response
    {
        return Inertia::render('Auth/Login', [
            'canResetPassword' => Route::has('password.request'),
            'status' => session('status'),
        ]);
    }

    /**
     * Handle an incoming authentication request.
     */
    public function store(LoginRequest $request)
    {
        if (Auth::attempt(['email' => $request->email, 'password' => $request->password, 'is_ban' => 1, 'roles_id' => 3])) {
            return response('error', 500);
        }

        $request->authenticate();

        $request->session()->regenerate();

        return redirect()->intended(RouteServiceProvider::HOME);
    }

    public function storeUser(LoginRequest $request)
    {
        if (Auth::attempt(['email' => $request->email, 'password' => $request->password, 'is_ban' => 1])) {
            return response('error', 500);
        }

        $request->authenticate();
    }
}
```

```

    $request->session()->regenerate();

    return redirect('/account');
}

/**
 * Destroy an authenticated session.
 */
public function destroyAdmin(Request $request): RedirectResponse
{
    Auth::guard('web')->logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

    return redirect('/login');
}

public function destroy(Request $request): RedirectResponse
{
    Auth::guard('web')->logout();

    $request->session()->invalidate();

    $request->session()->regenerateToken();

    return redirect('/');
}
}

```

UserController.php

Контролер користувачів.

```

<?php

namespace App\Http\Controllers;

use App\Models\User;
use App\Models\Groups;
use Inertia\Inertia;
use Illuminate\Support\Facades\Auth;
use Illuminate\Support\Facades\Hash;
use Illuminate\Http\Request;

class UserController extends Controller
{
    protected $publicStorage = '/uploads/';

    function profile()
    {
        return Inertia::render('Account/Settings', [
            'user' => Auth::user()
        ]);
    }
}

```

```

public function update(Request $request, $id)
{
    $data = $request->all();
    $data['is_ban'] = $request['is_ban'] == 'true' ? 1 : 0;
    User::find($id)->update($data);
}

function profileUpdate(Request $request)
{
    $user = User::find(Auth::id());
    $data = $request->all();
    $data['is_ban'] = $request['is_ban'] == 'true' ? 1 : 0;
    if (isset($request['new_photo']) && gettype($request['new_photo']) != 'string') {
        $path = $this->publicStorage . Auth::id();
        $name = $path . "/" . uniqid() . '.' . $request['new_photo']->getClientOriginalExtension();
        $request['new_photo']->move(public_path() . $path, $name);
        $data['photo'] = $name;
    }

    $user->update($data);
    return response()->json(User::find(Auth::id()));
}

function calendar()
{
    $userId = Auth::id();
    if (Auth::user()->roles_id == 1) {
        $model = Groups::whereHas('courses', function ($query) use ($userId) {
            $query->where('users_id', $userId);
        })->get();
    } else {
        $model = Groups::whereHas('applications', function ($query) use ($userId) {
            $query->where('users_id', $userId);
        })->get();
    }
    $events = [];
    foreach ($model as $value) {
        foreach (json_decode($value['schedule']) as $date) {
            array_push($events, [
                'id' => $value['id'],
                'title' => $value['time'] . " - " . $value['title'],
                'date' => $date
            ]);
        }
    }
    return Inertia::render('Account/Calendar', [
        'events' => $events,
        'groups' => $model
    ]);
}
}

```

CoursesController.php

Контролер курсів.

<?php

```

namespace App\Http\Controllers;

use App\Models\Courses;
use App\Models\Categories;
use App\Models\Languages;
use Illuminate\Http\Request;
use Inertia\Inertia;
use Illuminate\Database\Eloquent\Collection;
use Illuminate\Support\Facades\Auth;

class CoursesController extends Controller
{
    protected $publicStorage = '/uploads/';

    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        $courses = Courses::with('category', 'language')->where('users_id', Auth::id())->get();
        return Inertia::render('Account/Courses', [
            'courses' => $courses
        ]);
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        $course = Collection::make(new Courses);
        $course['photo'] = '/img/no-photo-available.png';
        return Inertia::render('Account/Teacher/CoursesEdit', [
            'categories' => Categories::get(),
            'languages' => Languages::get(),
            'course' => $course,
            'type' => 'create'
        ]);
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        $data = $request->all();
        $model = new Courses();
        if (isset($request['new_photo']) && $request['new_photo'] != "null" && $request['new_photo'] != "undefined") {
            $name = $this->publicStorage . uniqid() . '.' . $request['new_photo']->getClientOriginalExtension();
            $request['new_photo']->move(public_path() . $this->publicStorage, $name);
            $data['photo'] = $name;
        }
        $data['users_id'] = Auth::id();
        $model->create($data);
    }

    /**
     * Display the specified resource.
     */

```

```

public function show($id)
{
    $course = Courses::with(
        'category',
        'language',
        'groups',
        'lectures',
        'practices'
    )->find($id);
    if(Auth::user()->roles_id == 1) {
        return Inertia::render('Account/Teacher/Course', [
            'course' => $course
        ]);
    } else {
        return Inertia::render('Account/Course', [
            'course' => $course
        ]);
    }
}

/**
 * Show the form for editing the specified resource.
 */
public function edit($id)
{
    return Inertia::render('Account/Teacher/CoursesEdit', [
        'categories' => Categories::get(),
        'languages' => Languages::get(),
        'course' => Courses::find($id),
        'type' => 'edit'
    ]);
}

/**
 * Update the specified resource in storage.
 */
public function update(Request $request, $id)
{
    $model = Courses::find($id);
    $data = $request->all();
    if(isset($request['new_photo']) && gettype($request['new_photo']) != 'string') {
        $path = $this->publicStorage . Auth::id();
        $name = $path."/".uniqid().'.'.$request['new_photo']->getClientOriginalExtension();
        $request['new_photo']->move(public_path().$path, $name);
        $data['photo'] = $name;
    }
    $model->update($data);
}

/**
 * Remove the specified resource from storage.
 */
public function destroy($id)
{
    Courses::find($id)->delete();
}
}

```

Сторінка з календарем.

```

<script setup>
import { Head, Link } from '@inertiajs/vue3';
import { createToast } from 'mosha-vue-toastify';
defineProps({
  events: Array,
  groups: Array
});
</script>
<template>
  <Head title="Календар" />

  <Header></Header>
  <div class="container-xxl py-5">
    <div class="container">
      <div class="row">
        <div class="col-md-3 col-lg-3 col-xl-3 col-xs-12">
          <Navigation></Navigation>
        </div>
        <div class="col-md-9 col-lg-9 col-xl-9 col-xs-12">
          <FullCalendar :options="calendarOptions" />
        </div>
      </div>
    </div>
  </div>
  <Modal v-if="showModal" @close="showModal = false">
    <template v-slot:header>
      <h4>{{ group.title }}</h4>
    </template>
    <template v-slot:body>
      <div class="row">
        <div class="col-2">
          
        </div>
        <div class="col-6">
          <b>Час:</b> {{ group.time }}<br>
          <b>Викладач:</b> {{ group.courses.user.name }} {{ group.courses.user.surname }}<br>
          <b>Курс:</b>
          <Link :href="/account/courses/${group.courses.id}`">{{ group.courses.title }}</Link>
        </div>
        <div class="col-4">
          <a :href="group.link_zoom"
            class="primary-btn2 text-uppercase enroll rounded-0 text-white bg-blue link-zoom">Підключитись</a>
        </div>
      </div>
    </template>
  </Modal>

  <Modal v-if="showCreateModal" @close="showCreateModal = false">
    <template v-slot:header>
      <h4>Назначити заняття</h4>
    </template>
    <template v-slot:body>
      <div class="form-floating">
        <select class="form-control" id="groups_id" v-model="newEvent.groups_id">
          <option v-for="item in groups" :key="item.id" :value="item.id">{{
            item.title }}</option>
        </select>
      </div>
    </template>
  </Modal>

```



```

        </select>
        <label for="groups_id">Група</label>
    </div>
</template>
<template v-slot:footer>
    <button class="btn btn-primary" @click="addEvent()">
        Назначити
    </button>
    <button class="btn btn-primary" @click="showCreateModal = false">
        Закрити
    </button>
</template>
</Modal>

<Footer></Footer>
</template>

<script>
import Header from '@/Components/Site/Header.vue'
import Footer from '@/Components/Site/Footer.vue'
import Navigation from '@/Components/Site/Navigation.vue'
import Modal from '@/Components/Site/Modal.vue'

import FullCalendar from '@fullcalendar/vue3'
import dayGridPlugin from '@fullcalendar/daygrid'
import interactionPlugin from '@fullcalendar/interaction'
import allLocales from '@fullcalendar/core/locales-all';

export default {
  components: {
    Header,
    Footer,
    FullCalendar,
    Modal
  },
  data() {
    return {
      showModal: false,
      showCreateModal: false,
      group: null,
      newEvent: {
        groups_id: null,
        schedule: []
      },
      calendarOptions: {
        plugins: [dayGridPlugin, interactionPlugin],
        initialView: 'dayGridMonth',
        locale: 'ukr',
        locales: allLocales,
        buttonText: {
          today: 'Сьогодні'
        },
        display: '#DEDEDE',
        dateClick: this.handleDateClick,
        events: this.events,
        eventClick: this.handleEventClick
      }
    }
  },
},

```

```

methods: {
  addEvent() {
    axios.post(`/api/account/groups/${this.newEvent.groups_id}/schedule`, this.newEvent)
      .then((response) => {
        location.reload();
      }).catch(() => {
        createToast('Помилка!', {
          type: 'danger'
        })
      })
  },
  handleEventClick(info) {
    axios.get(`/api/account/groups/${info.event.id}`).then((response) => {
      this.group = response.data;
      this.showModal = true;
    }).catch(() => {
      createToast('Помилка!', {
        type: 'danger'
      })
    })
  },
  handleDateClick(arg) {
    if (this.$page.props.auth.user.roles_id == 1) {
      this.newEvent.schedule = [arg.dateStr];
      this.showCreateModal = true;
    }
  }
}
}
</script>
<style lang="css">
.link-zoom {
  width: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 20px 0;
}
</style>

```

Course.vue

Сторінка курсу.

```

<script setup>
import { Head, Link } from '@inertiajs/vue3';
import { createToast } from 'mosha-vue-toastify';
defineProps({
  course: Object
});
</script>
<template>
  <Head :title="course.title" />

  <Header></Header>

  <div class="container-fluid bg-primary py-5 mb-5 page-header">
    <div class="container py-5">
      <div class="row justify-content-center">

```

```

    <div class="col-lg-10 text-center">
      <h1 class="display-3 text-white animated slideInDown">{{ course.title }}</h1>
    </div>
  </div>
</div>
</div>
</div>
<div class="container-xxl py-5">
  <div class="container">
    <section class="course_details_area section_gap">
      <div class="container">
        <div class="row">
          <div class="col-lg-8 course_details_left">
            <div class="main_image">
              
            </div>
            <div class="content_wrapper">
              <h4 class="title">Про курс</h4>
              <div class="content" v-html="course.description"></div>
            </div>
          </div>
          <div class="col-lg-4 right-contents">
            <ul>
              <li>
                <div class="justify-content-between d-flex">
                  <p class="m-0">Викладач</p>
                  <span class="or">
                    <Link href="#">{{ course.user.name }} {{ course.user.surname }}</Link>
                  </span>
                </div>
              </li>
              <li>
                <div class="justify-content-between d-flex">
                  <p class="m-0">Вчений рівень</p>
                  <span>{{ course.regalia ? course.regalia : 'Відсутній' }}</span>
                </div>
              </li>
              <li>
                <div class="justify-content-between d-flex">
                  <p class="m-0">Термін навчання</p>
                  <span>{{ course.duration }}</span>
                </div>
              </li>
              <li>
                <div class="justify-content-between d-flex">
                  <p class="m-0">Напрявлення</p>
                  <span>{{ course.category.title }}</span>
                </div>
              </li>
              <li>
                <div class="justify-content-between d-flex">
                  <p class="m-0">Мова</p>
                  <span>{{ course.language.title }}</span>
                </div>
              </li>
              <li>
                <div class="justify-content-between d-flex">
                  <p class="m-0">Доступ</p>
                  <span>{{ course.access ? 'Набір відкритий' : 'Набір закрито' }}</span>
                </div>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </section>
  </div>
</div>

```



```

</template>

<script>
import Header from '../Components/Site/Header.vue'
import Footer from '../Components/Site/Footer.vue'
export default {
  components: {
    Header,
    Footer
  },
  data() {
    return {
      app: false,
      comment: {
        description: "",
        rate: 4
      }
    }
  },
  methods: {
    application() {
      axios.post('/account/applications', {
        courses_id: this.course.id
      }).then(() => {
        this.app = true;
        createToast("Заявку подано!")
      }).catch(() => {
        createToast('Помилка!', {
          type: 'danger'
        })
      })
    },
    addReview() {
      axios.post(`/account/courses/${this.course.id}/reviews`, this.comment).then((response) => {
        this.course.reviews.push(response.data);
        this.comment.description = "";
        createToast("Відгук додано!")
      }).catch(() => {
        createToast('Помилка!', {
          type: 'danger'
        })
      })
    }
  }
}
</script>

```

Teacher.vue

Профіль викладача.

```

<script setup>
import { Head, Link } from '@inertiajs/vue3';
defineProps({
  teacher: Object
});
</script>

<template>

```

```

<Head title="Головна" />

<!-- Navbar Start -->
<Header></Header>
<!-- Navbar End -->

<!-- About Start -->
<div class="container-xxl py-5">
  <div class="container">
    <div class="row g-5">
      <div class="col-lg-6 wow fadeInUp" data-wow-delay="0.1s" style="min-height: 400px;">
        <div class="position-relative h-100">
          
        </div>
      </div>
      <div class="col-lg-6 wow fadeInUp" data-wow-delay="0.3s">
        <h6 class="section-title bg-white text-start text-primary pe-3">Про мене</h6>
        <h1 class="mb-4">{{ teacher.data.name }} {{ teacher.data.surname }}</h1>
        <p class="mb-4" v-html="teacher.data.description"></p>
        <div class="row gy-2 gx-4 mb-4">
          <div class="col-sm-6">
            <p class="mb-0"><i class="fa fa-arrow-right text-primary me-2"></i>{{ teacher.data.email }}</p>
          </div>
          <div class="col-sm-6" v-if="teacher.data.phone">
            <p class="mb-0"><i class="fa fa-arrow-right text-primary me-2"></i>{{ teacher.data.phone }}</p>
          </div>
          <div class="col-sm-6" v-if="teacher.data.regalia">
            <p class="mb-0"><i class="fa fa-arrow-right text-primary me-2"></i>{{ teacher.data.regalia }}</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<!-- About End -->

<!-- Courses Start -->
<div class="container-xxl py-5">
  <div class="container">
    <div class="text-center wow fadeInUp" data-wow-delay="0.1s">
      <h4 class="section-title bg-white text-center text-primary px-3">Мої курси</h4>
    </div>
    <div class="row g-4">
      <div class="col-lg-4 col-md-6 wow fadeInUp" v-for="(item, index) in teacher.data.courses" :key="item.id"
        :data-wow-delay="`0.${index}s`">
        <Course :data="item"></Course>
      </div>
    </div>
  </div>
</div>
<!-- Courses End -->
<Footer></Footer>
</template>
<script>
import Header from './Components/Site/Header.vue'
import Footer from './Components/Site/Footer.vue'
import Course from './Components/Site/Course.vue'
export default {

```

```
    components: {  
      Header,  
      Footer,  
      Course  
    },  
  }  
</script>  
  
<style></style>
```