

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
«ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ»
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ В ЕКОНОМІЦІ

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Освітньо-професійна програма

«Інформаційні управляючі системи і технології»

Форма навчання: ДЕННА

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ
ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ

здобувача Ігнат'єва Іллі Сергійовича _____

Науковий керівник:

Доктор економічних наук, професор

_____ Устенко С.В.

Кваліфікаційна магістерська робота

допущений до захисту в

Екзаменаційній комісії з атестації

здобувачів вищої освіти

в.о. завідувача кафедри к.е.н., доцент.

_____ Тішков Б.О.

Київ 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА»

Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці
галузь знань 12 «Інформаційні технології»
спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма
«Інформаційні управляючі системи і технології»

Затверджую:

в.о. завідувача кафедри _____ Тішков Б.О.

“ _____ ” _____ 2023 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
здобувача Ігнат'єва Іллі Сергійовича

денної форми навчання

на підготовку кваліфікаційної магістерської роботи

на тему: «Розроблення інформаційної системи для дистанційного навчання»

Тему затверджено наказом ректора Університету від «20» лютого 2023 р.
№ 335-ст

Кваліфікаційна магістерська робота виконується на матеріалах
систем для дистанційного навчання

ВІДГУК

на кваліфікаційну магістерську роботу здобувача
Навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
ОП «Інформаційні управляючі системи і технології»
спеціальності 122 «Комп'ютерні науки»

Ігнат'єва Іллі Сергійовича, виконаного на тему:
«Розроблення інформаційної системи для дистанційного навчання»

Кваліфікаційна магістерська робота Ігнат'єва І.С. присвячена вирішенню задач з розроблення інформаційної системи для дистанційного навчання у ВНЗ.

Для створення інформаційної системи здобувачем розв'язано такі задачі: досліджено загальний процес розробки інтернет-ресурсу для проектування інформаційної системи з використання сучасних методологій проектування; досліджено можливості удосконалення процесу навчання студентів ЗВО при використанні сучасних технологій; створено інформаційну систему та її компонентів проведення учбового процесу для дистанційної форми навчання.

В першому розділі кваліфікаційної магістерської роботи проведено аналіз сучасних інформаційних систем дистанційного навчання та визначення їх переваг, обґрунтовано методи та підходи до створення цільової системи. Здійснено розробку концептуальної схеми інформаційної системи дистанційного навчання, яка включає визначення функціональної структури та опис окремих компонентів системи. Концептуальна модель системи складається з трьох компонентів: фронтенд (HTML, CSS, JS), бекенд (Python) та бази даних (SQLite), використання яких дозволяє створити простий і в той же час мати багатофункціональний веб-сайт для навчання.

Другий розділ присвячений опису структури та характеристики системи: сервер на якому розташовано ядро системи дистанційного навчання, необхідні підтримуючі файли, БД, сховище даних, інтерфейсні модулі. Структура проекту включає 2 частини серверну та клієнтську. Для виконання постановки та розроблення алгоритму розв'язання задач, а також візуалізації структури інформаційної системи було взято за основу еРС діаграми, нотації BPMN, що допомагає уникнути невідповідності між вимогами користувачів та функціональністю системи та забезпечує ефективну комунікацію між розробниками та користувачами. В розроблюваній інформаційній системі для заохочення заданих необхідних курсів для користувачів був використаний алгоритм колаборативного фільтрування. Цей алгоритм базується на розрахунку схожості між користувачами та курсами і використовує цю інформацію для рекомендації та підтримки заданих курсів.

В розділі III здобувачем висвітлені питання організації інформаційного забезпечення, програмного та технічного дослідження. В роботі розглянуті

питання вибору СКБД SQLite, створено перелік необхідних масивів та їх структуру, екранні форми і створено модель БД. Клієнт-серверна архітектура та інструменти розробки, що були використані у проєкті дозволили створити надійний та функціональний додаток для дистанційного навчання з необхідними функціями, такими як реєстрація користувачів, створення курсів та лекцій, додавання матеріалів для навчання, проходження тестів та отримання сертифікатів про успішне завершення курсів. Клієнт-серверна архітектура дозволила забезпечити стабільну та безперебійну роботу додатку, а використання Python та Flask зробило розробку серверної частини швидкою та зручною. Використання HTML, CSS та JavaScript дозволило розробити привабливий та інтуїтивно зрозумілий інтерфейс користувача. Використання SQLite та SQLite3 для зберігання даних забезпечило надійність та ефективність роботи додатку. Також розроблено методичне забезпечення у вигляді «Керівництва користувача» у вигляді наданих опцій та рекомендацій для практичної роботи.

Результатом реалізації даної системи є розроблена інформаційна система для дистанційного навчання під **авторським брендом «Ignatyev»**. Розроблена інформаційна система виконує всі визначені в процесі дослідження функціональні вимоги та задачі проєктування.

При виконанні кваліфікаційної магістерської роботи здобувач Ігнат'єв І.С. показав високий рівень знань, умінь узагальнювати практичний досвід використання сучасних комп'ютерних технологій і самостійно розробляти новітні проєктні рішення. Кваліфікаційна магістерська робота відповідає нормативним вимогам спеціальності 122 «Комп'ютерні науки» ОП «Інформаційні управляючі системи і технології», виконана на високому професійному рівні і може бути допущена до захисту в екзаменаційній комісії.

Рекомендації наукового керівника:

Ця тематика вкрай необхідна для нашого Університету, тому бажано продовжити, як її впроваджувати для нашого Інформаційного Центру та навчального процесу в цілому.

Науковий керівник, д.е.н., професор
кафедри ICE

С.В. Устенко

РЕЦЕНЗІЯ

**на кваліфікаційну магістерську роботу здобувача
спеціальності 122 «Комп'ютерні науки»**

Ігнат'єва Іллі Сергійовича,

виконаного на тему:

«Розроблення інформаційної системи для дистанційного навчання»

Кваліфікаційна магістерська робота(КМР) здобувача Ігнат'єва І.С. є професійним науковим дослідженням, присвяченим розробці сучасної інформаційної системи для дистанційного навчання закладів вищої освіти України.

Як свідчать матеріали КМР результатом реалізації даної системи є авторський бренд «Ignatyev», незважаючи на велику кількість інформаційних систем і виконує всі визначені в процесі дослідження функціональні вимоги та задачі проектування.

В першому та другому розділах кваліфікаційної магістерської роботи проведено аналіз сучасних інформаційних систем дистанційного навчання та визначені їх переваг та недоліки, проведено загальний опис структури та характеристики системи, наведено алгоритм розв'язання задач. Запропонована концептуальна модель системи, яка складається з трьох компонентів: фронтенд (HTML, CSS, JS), бекенд (Python) та бази даних (SQLite), використання яких дозволяє створити простий і в той же час мати багатофункціональний веб-сайт для навчання студентів.

В конструктивному розділі висвітлені питання організації інформаційного забезпечення, програмного та технічного дослідження. В роботі розглянуті питання вибору СКБД SQLite, створено перелік необхідних масивів та їх структуру, екранні форми і створено модель БД. Клієнт-серверна архітектура та інструменти розробки дозволили створити надійний додаток для дистанційного навчання з необхідними функціями, такими як реєстрація користувачів, створення курсів та лекцій, додавання матеріалів для навчання, проходження тестів та отримання сертифікатів про успішне завершення навчальних курсів. Розроблено також інтерфейс користувача та проведено тестування дистанційної системи навчання.

Кваліфікаційна магістерська робота Ігнат'єва І. С. виконана на високому професійному рівні з урахуванням навчально-методичного забезпечення до розробки магістерських робіт у ЗВО зі спеціальності 122 «Комп'ютерні науки» і заслуговує на оцінку «Відмінно».

Рецензент:

Начальник відділу електронних компонентів

Державного підприємства «Антонов»

Луганський С.В.

“ ____ ” _____ 2023 р.

План кваліфікаційної магістерської роботи

Розділ I Характеристика та аналіз предметної галузі _____

Розділ II Розробка вимог і моделювання інформаційної системи

Розділ III Проектування та реалізація компонентів системи

Об'єкт дослідження - інформаційна система дистанційного навчання _____

Предмет дослідження - методології проектування дистанційної форми навчання _____

Мета кваліфікаційної магістерської роботи розробка та реалізація інформаційної системи дистанційного навчання, яка забезпечить ефективну організацію процесу навчання та надання якісної освіти з використанням сучасних технологій. _____

Конкретні завдання, які студент повинен виконати для досягнення поставленої мети:

У розділі I навести характеристику предметної галузі та об'єкта дослідження, проаналізувати літературні джерела та практичний досвід використання інформаційних систем і технологій для задач дистанційного навчання _____

У розділі II Характеристика інформаційних систем та моделі і методи вимог до інформаційної системи дистанційного навчання, її архітектурну структуру _____

У розділі III описати проектні рішення з компонентів інформаційної системи – інформаційного, технічного та програмного забезпечень _____

Завдання підготував

науковий керівник _____ Устенко Станіслав Веніамінович

(підпис) (прізвище, ім'я, по-батькові)

“ _____ ” _____ 2023 р.

Завдання одержав

здобувач _____ Ігнат'єв Ілля Сергійович

(підпис) (прізвище, ім'я, по-батькові)

“ _____ ” _____ 2023 р.

АНОТАЦІЯ

Кваліфікаційна магістерська робота студента 6 курсу

Навчально-наукового інституту «Інститут інформаційних технологій в економіці»

Ігнат'єва Іллі Сергійовича, виконаного на тему:

«Інформаційна система для дистанційного навчання»

Київ: кафедра інформаційних систем в економіці, 2023 р.

Кваліфікаційна магістерська робота присвячена темі в сучасному світі, де зростає потреба в ефективному інтернет-навчанні. У зв'язку з поширенням технологій та зростанням кількості користувачів Інтернету, все більше людей обирають дистанційну форму навчання, яка дозволяє здобувати знання у зручний для них час та місце.

Кваліфікаційна магістерська робота складається з трьох розділів, логічно пов'язаних між собою.

Перший розділ являється теоретичним. В ньому наведено аналіз задачі, розробки та функціонування інформаційної системи для дистанційного навчання, а також наводяться переваги та недоліки існуючих програмних засобів для вирішення поставленої задачі.

Другий розділ є аналітичним і присвячений дослідженню об'єкта, опису структури інформаційної системи для дистанційного навчання.

Третій розділ – конструктивний. Тут наведено організаційне, інформаційне, програмне забезпечення для функціонування інформаційної системи дистанційного навчання. Висвітлені питання організації інформаційного забезпечення: розроблена структура інформаційного забезпечення, описані та наведені в додатках форми вхідних та вихідних документів.

Висновки містять рекомендації щодо доцільності розробки та впровадження інформаційної системи для дистанційного навчання.

В додатках розміщені програмні коди реалізації даної системи та інтерфейси для користувача та адміністратора.

РЕФЕРАТ

Кваліфікаційна магістерська робота містить 85 сторінок, 9 таблиць, 33 рисунки, список літератури з 19 найменуваннями, 3 додатки.

“ Інформаційна система для дистанційного навчання ”

Перелік ключових слів: дистанційне навчання, python, SQL, інтерфейс, система, інформаційна система, Flask, Дистанційне навчання, Онлайн навчання, Віртуальне навчальне середовище, Система управління навчанням, Змішане навчання, Синхронне навчання, Асинхронне навчання.

Об’єктом дослідження є інформаційна система дистанційного навчання, що об’єднує в собі компоненти апаратно-програмного забезпечення, зокрема інтерфейсний модуль, база даних та інші складові, необхідні для реалізації процесу дистанційного навчання.

Предметом дослідження для інформаційної системи дистанційного є сам процес навчання на віддаленій платформі. Дослідження може охоплювати різні аспекти такої системи, включаючи:

1. Ефективність навчання: Дослідження спрямовується на визначення ефективності дистанційного навчання порівняно з традиційними методами навчання. Вивчаються показники успішності, рівень засвоєння матеріалу та залученість студентів до навчального процесу.
2. Взаємодія та спілкування: Дослідження фокусується на аналізі взаємодії між здобувачами та викладачами на дистанційній платформі.
3. Технологічні аспекти: Дослідження охоплює технологічні аспекти розробки та функціонування інформаційної системи для дистанційного навчання.
4. Вплив дистанційного навчання: Дослідження спрямовується на вивчення впливу дистанційного навчання на студентів та їхні результати.

Мета кваліфікаційної магістерської роботи є розробка та реалізація інформаційної системи дистанційного навчання, яка забезпечить ефективну організацію процесу навчання та надання якісної освіти з використанням сучасних технологій.

Результати, досягнуті в процесі роботи: розроблена інформаційна система для дистанційного навчання.

Одержані результати можуть бути використані в закладах для навчання.

Рік виконання магістерської кваліфікаційної роботи: 2023

Рік захисту магістерської кваліфікаційної роботи: 2023

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ.....	4
Вступ.....	5
I. ТЕОРЕТИЧНИЙ РОЗДІЛ. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Аналіз існуючих інформаційних систем (предметної області).....	9
1.1.2 Основні концепти дистанційного навчання	12
1.1.3 Організаційні питання при проектуванні системи	15
1.1.4 Фактори що впливають на тематику та розробку ІС.....	16
1.2 Обґрунтування вибору підходів і технологій для створення інформаційних систем.	20
Розділ II. Аналітичний розділ. Характеристика інформаційних управляючих систем та методи і моделі.....	28
2.1 Структура і характеристика системи	28
2.1.1 Дерево задач Клієнтської частини	31
2.1.2 Функціональні та нефункціональні вимоги	33
2.1.3 Клієнт-серверна архітектура.....	35
2.1.4 Діаграми всіх рівнів декомпозиції	38
2.1.5 Діаграма прецедентів.....	42
2.2. Методи та моделі в інформаційних управляючих системах та технологіях.....	50
РОЗДІЛ III. КОНСТРУКТИВНИЙ РОЗДІЛ.....	54
3.1. Проектування бази даних для інформаційних систем	54
3.1.1. Проектування інфологічної(логічної) моделі БД	54

3.1.2. Обґрунтування вибору СКБД та проектування фізичної моделі бази даних	56
3.2. Інтелектуальний аналіз даних.....	66
3.3 Програмне забезпечення	68
3.3.1 Архітектура програмного забезпечення	69
3.3.2 Керівництво користувача	70
3.4 Технічне забезпечення.....	79
3.5 Результати реалізації інформаційної системи.....	81
Висновки	83
Список використаної літератури	84
Додатки	

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ ВЕЛИЧИН І ТЕРМІНІВ

БД – База даних

БЗ – База знань

ООП – Об’єктно орієнтоване програмування

ПЗ – Програмне забезпечення

СКБД – Система керування базами даних

ІС – Інформаційна система

ІТ – Інформаційні технології

ІМБД - Інформаційна модель бази даних

ВСТУП

Розробка інформаційної системи дистанційного навчання має багато переваг і актуальна в сучасному світі з кількох причин.

По-перше, дистанційна форма навчання стала необхідною у зв'язку з пандемією COVID-19 та воєнного часу. Інформаційні системи дистанційного навчання допомогли забезпечити безперебійний процес навчання в умовах зовнішніх ризиків.

По-друге, інформаційні системи дистанційного навчання стали більш популярними не тільки через зовнішні ризики, але й як зручний та доступний спосіб навчання для здобувачів з різних куточків світу. Це забезпечує більш широку доступність до освіти та можливість здобуття якісної освіти в будь-якому місці.

По-третьє, розробка інформаційної системи дистанційного навчання може забезпечити більш ефективне та якісне навчання. Інформаційні системи дистанційного навчання можуть містити в собі багато різноманітних матеріалів, відео-уроків, інтерактивних завдань, тестів та інших електронних продуктів, що допомагає здобувачам засвоювати матеріал більш ефективно та зрозуміло. Також, можливість взаємодії з викладачами та здобувачами у режимі онлайн сприяє більш активному та якісному навчанню.

По-четверте, Сучасні інформаційні системи які функціонують в режимі дистанційного навчання мають широкі можливості використання електронних платформ, мереж та програмних продуктів і систем.

По-п'яте, як показав ковідний досвід, використання інформаційних систем дистанційного навчання забезпечує стабільну роботу навчальних закладів, тому в сучасних воєнних умовах цей досвід може бути врахований з урахуванням використання інноваційних технологій і інформаційних систем.

Отже, розробка інформаційної системи дистанційного навчання є дуже актуальною темою в сучасному світі. Світ дедалі більше переходить до цифрової ери, а навчання не є винятком. Завдяки інформаційним технологіям, сьогодні учні

та студенти можуть отримувати знання не виходячи з дому, просто за допомогою комп'ютера та Інтернету.

Однак розробка ІС дистанційного навчання має свої виклики та складнощі. Основні проблеми пов'язані з організацією процесу навчання, контролю за роботою студентів, створенням та збереженням цифрових матеріалів та даних, забезпечення безпеки та захисту інформації тощо.

Для розробки ІС дистанційного навчання необхідно мати знання з області програмування, баз даних, створення веб-сторінок та їх оптимізації. Також необхідно розуміти процеси, що відбуваються в процесі навчання та мати досвід у розробці та реалізації педагогічних методик та технологій.

Отже, розробка ІС дистанційного навчання є складним та важливим завданням, яке потребує великої кількості знань та досвіду. Але зробити це можливо, і в результаті можна отримати ефективну та зручну систему, яка допоможе в навчанні та розвитку людей в будь-якому куточку світу.

Мета і задачі дослідження

Метою дослідження є розробка та реалізація інформаційної системи дистанційного навчання, яка забезпечить ефективну організацію процесу навчання та надання якісної освіти з використанням сучасних технологій.

Задачі дослідження:

- Аналіз сучасних інформаційних систем дистанційного навчання та визначення їх переваг та недоліків.
- Розробка концептуальної схеми інформаційної системи дистанційного навчання, включаючи визначення функціональної структури та опис окремих компонентів системи.
- Розробка бази даних для системи дистанційного навчання, яка забезпечить збереження та швидкий доступ до різноманітних даних та матеріалів.
- Розробка алгоритмів та програмного забезпечення для реалізації функціональної структури інформаційної системи дистанційного навчання.

- Розробка інтерфейсу користувача та забезпечення його зручності та простоти використання.
- Тестування та валідація розробленої інформаційної системи дистанційного навчання.
- Оцінка ефективності та користуваності розробленої системи дистанційного навчання.

Отже, мета дослідження є створення ефективної та зручної інформаційної системи дистанційного навчання, яка забезпечить якісну освіту та допоможе підвищити рівень знань та навичок студентів та учнів. Для цього необхідно вирішити ряд завдань, пов'язаних з аналізом, проектуванням, розробкою та тестуванням системи

Об'єкт дослідження

Об'єктом дослідження є інформаційна система дистанційного навчання, яка є складною інформаційною системою, що об'єднує в собі елементи програмного забезпечення, апаратне забезпечення, бази даних та інші складові, необхідні для реалізації процесу дистанційного навчання. Об'єктом дослідження можуть бути як вже наявні інформаційні системи дистанційного навчання, так і процес їх створення та вдосконалення.

Особистий внесок магістранта

Особистий внесок магістранта може включати в себе:

- Пошук та аналіз наукової літератури з питань розробки та впровадження інформаційних систем дистанційного навчання.
- Розробка та вибір методів та технологій, які будуть використовуватися для реалізації інформаційної системи дистанційного навчання.
- Участь у процесі розробки та впровадження інформаційної системи дистанційного навчання, зокрема, у розробці та тестуванні функціональних вимог, розробці дизайну та інтерфейсу користувача, участі у виборі технічних рішень та інших аспектів проекту.

- Проведення аналізу результатів впровадження інформаційної системи дистанційного навчання та внесення пропозицій щодо її подальшого вдосконалення.
- Аналіз та узагальнення досвіду використання інформаційних систем дистанційного навчання в інших навчальних закладах, виявлення їх переваг та недоліків та врахування цього досвіду у процесі розробки та впровадження власної інформаційної системи.

I. ТЕОРЕТИЧНИЙ РОЗДІЛ. ДОСЛІДЖЕННЯ ТА АНАЛІЗ ПІДХОДІВ ДО СТВОРЕННЯ ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ (ПРЕДМЕТНОЇ ОБЛАСТІ).

Розробка інформаційної системи для дистанційного навчання є складним і трудомістким процесом, який включає низку потенційних проблем і перешкод, з якими користувачі можуть зіткнутися, починаючи цей тип проекту.

Серед найпоширеніших проблем визначимо:

- *Технічні проблеми:* розробка інформаційної системи для дистанційного навчання потребує досвіду в ряді технічних областей, включаючи розробку програмного забезпечення, адміністрування мережі та управління базами даних, тестування системи тощо. Залежно від завдань проекту, це може вимагати значних інвестицій в апаратне забезпечення, програмне забезпечення та іншу інфраструктуру.

- *Проблеми взаємодії з користувачем:* щоб бути успішною, ІС для дистанційного навчання має бути простою у використанні, інтуїтивно зрозумілою та привабливою як для викладачів, так і для здобувачів. Для цього потрібно реалізувати принципи веб-дизайну взаємодії з користувачем (UX) і відданість безперервному тестуванню та вдосконаленню.

- *Проблеми розробки контенту.* Розробка високоякісного освітнього контенту для ІС дистанційного навчання потребує знання предмета, сильних навичок письма та глибокого розуміння того, як надавати контент привабливим та ефективним способом. Залежно від обсягу проекту це може вимагати значних інвестицій у розробку та виробництво контенту.

- *Проблеми доступності:* щоб бути ефективною, ІС для дистанційного навчання має бути доступною для користувачів із будь-якими здібностями. Це вимагає відданості відповідності стандартам проектування та можливості інвестувати в інструменти та технології систем дистанційного навчання, які можуть допомогти користувачам з обмеженими можливостями чи іншими

особливими потребами, особливо для користувачів, які постраждали від дій воєнного часу.

- *Фінансові труднощі:* розробка ІС для дистанційного навчання може бути дорогим заходом, і можуть бути значні початкові витрати, пов'язані з обладнанням, програмним забезпеченням та іншою інфраструктурою. Крім того, поточні витрати на обслуговування та підтримку можуть бути значними, і може знадобитися забезпечити постійне фінансування, щоб забезпечити довгострокову життєздатність системи.

Таким чином, розробка інформаційної системи для дистанційного навчання є складною роботою, яка потребує експертних знань у низці технічних, освітніх та користувацьких областей. Щоб успішно подолати ці виклики, потрібне ретельне планування, сильні навички управління проектами та відданість постійній ітерації та вдосконаленню на основі відгуків користувачів і зацікавлених сторін.

Дослідження щодо розробки інформаційних систем для дистанційного навчання тривають і охоплюють ряд дисциплін, включаючи інформатику, освіту, психологію та взаємодію людини з комп'ютером, використання сучасних методологій користування ІС. Деякі з поточних напрямків у цій галузі включають:

1. *Архітектурне забезпечення ІС для дистанційного навчання:*

- *Клієнтські додатки та інтерфейси користувача:* включають в себе веб-браузери, мобільні додатки, програмне забезпечення для відеоконференцій та інші додатки, які використовуються для отримання доступу до системи дистанційного навчання.

- *Серверне програмне забезпечення:* включає в себе сервери для збереження та обробки даних, сервери для відеоконференцій та інші сервери, що забезпечують роботу системи дистанційного навчання.

- *База даних:* включає в себе всі дані, які використовуються в системі дистанційного навчання, включаючи інформацію про користувачів, курси, завдання, результати тестів, оцінки та ін.

2. *Дизайн, орієнтований на користувача*: зростає визнання важливості проектування систем дистанційного навчання з урахуванням потреб і вподобань користувачів. Дослідники досліджують низку методів включення відгуків і переваг користувачів у проектування системи, включаючи тестування користувачів, фокус-групи та опитування.

3. *Штучний інтелект (ШІ) і машинне навчання*: зростає інтерес до використання ШІ та алгоритмів машинного навчання для підвищення ефективності систем дистанційного навчання. Дослідники досліджують ряд застосувань цих технологій, включаючи персоналізоване навчання, адаптивне оцінювання та автоматичний зворотний зв'язок.

4. *Гейміфікація та залучення*: Дослідники досліджують використання принципів дизайну ігор та інших стратегій залучення, щоб зробити системи дистанційного навчання більш цікавими та мотивуючими для учнів. Це включає експерименти з такими функціями, як винагороди, індикатори прогресу та соціальні функції.

5. *Доступність та інклюзивність*: дослідники досліджують способи зробити системи дистанційного навчання більш доступними та інклюзивними для учнів з обмеженими можливостями чи іншими особливими потребами. Це включає в себе вивчення використання допоміжних технологій, таких як зчитувачі з екрану та програмне забезпечення для розпізнавання голосу, а також розробку контенту, який розроблений таким чином, щоб бути інклюзивним і доступним для всіх учнів.

6. *Педагогіка та дизайн навчання*: Дослідники досліджують низку педагогічних підходів і стратегій дизайну навчання, які можуть допомогти зробити дистанційне навчання ефективнішим. Це включає дослідження використання інтерактивних мультимедіа, спільного навчання та проектного навчання.

Дослідження в галузі дистанційного навчання є дуже міждисциплінарними та охоплюють широкий спектр тем. Оскільки попит на дистанційне навчання продовжує зростати, цілком ймовірно, що ця сфера й надалі привертатиме інтерес та інвестиції з боку дослідників, педагогів і технологічних компаній.

1.1.2 ОСНОВНІ КОНЦЕПТИ ДИСТАНЦІЙНОГО НАВЧАННЯ

В останні роки використання технологій в освіті стрімко зросло, особливо у сфері дистанційного навчання. Дистанційне навчання стало можливим завдяки розробці інформаційних систем, які дозволяють здобувачам отримувати доступ до навчальних матеріалів і віддалено взаємодіяти зі своїми вчителями та однолітками. Далі будуть наведені приклади розробки інформаційної системи дистанційного навчання та будуть наведені реальні факти їх впровадження.[5]

Розробка інформаційної системи для дистанційного навчання передбачає інтеграцію різних технологій, таких як системи управління навчанням (LMS), програмне забезпечення для відеоконференцій, онлайн-інструменти для співпраці та системи керування контентом (CMS). Ці технології використовуються для створення онлайн-навчального середовища, доступного для студентів з будь-якої точки світу.

Розглянемо деякі сучасні ІС для дистанційного навчання:

1. Blackboard Learning Management System. Blackboard — це веб-система, яка надяє викладачам платформу для створення та керування онлайн-курсами. Система містить інструменти для створення та обміну матеріалами курсу, спілкування зі студентами та моніторингу прогресу студентів. Blackboard використовується багатьма університетами та коледжами по всьому світу, і він відіграв важливу роль у розвитку програм дистанційного навчання.[17]

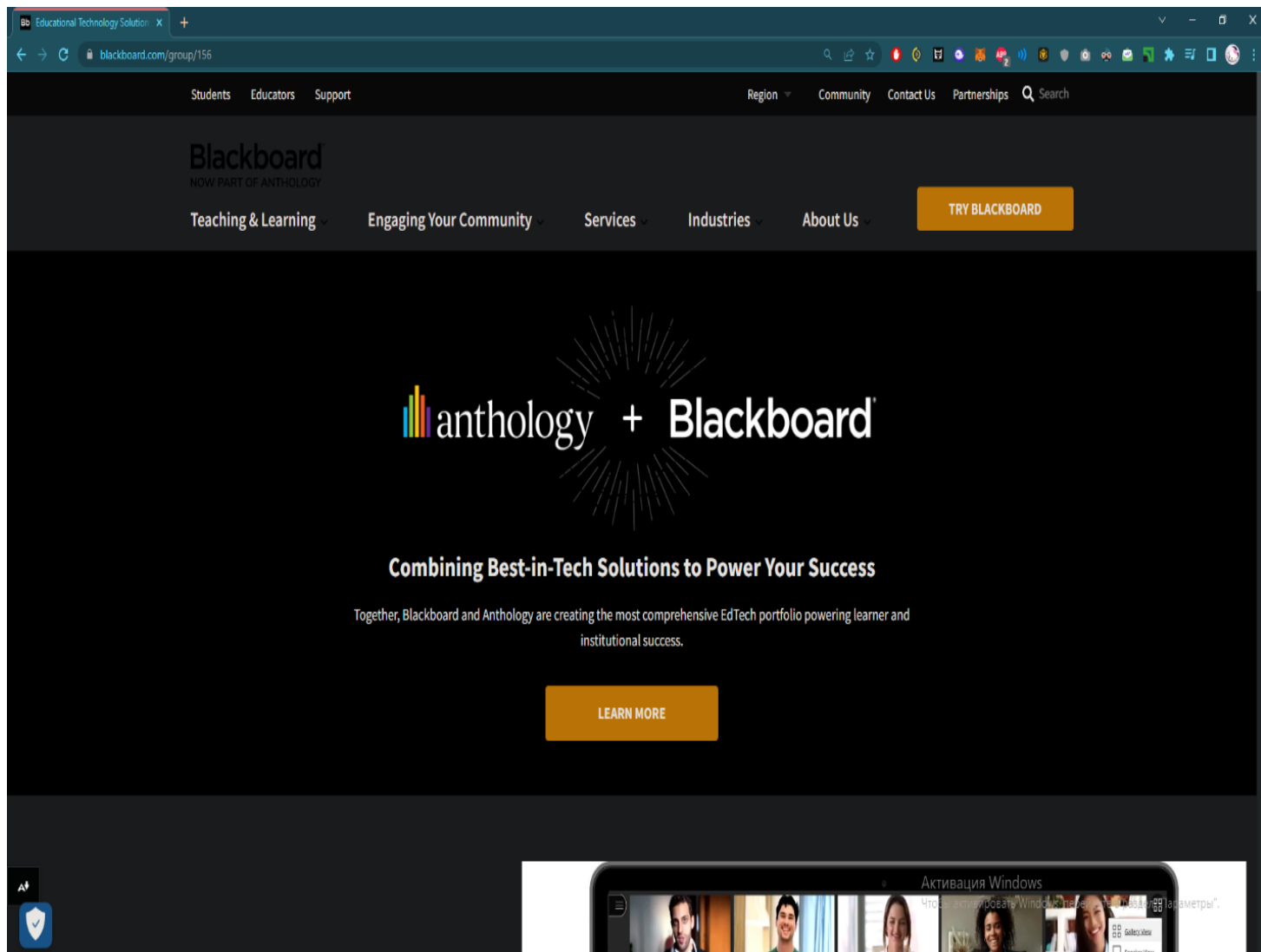


Рис. 1.1 Головна сторінка BlackBoard Learning [17]

2. Coursera

Coursera — це навчальна онлайн-платформа, яка пропонує курси від провідних університетів і організацій з усього світу. Платформа містить відеолекції, тести та завдання, які оцінюються іншими студентами або автоматизованими системами. Coursera використовується мільйонами студентів у всьому світі, і це допомогло демократизувати доступ до освіти.[18]

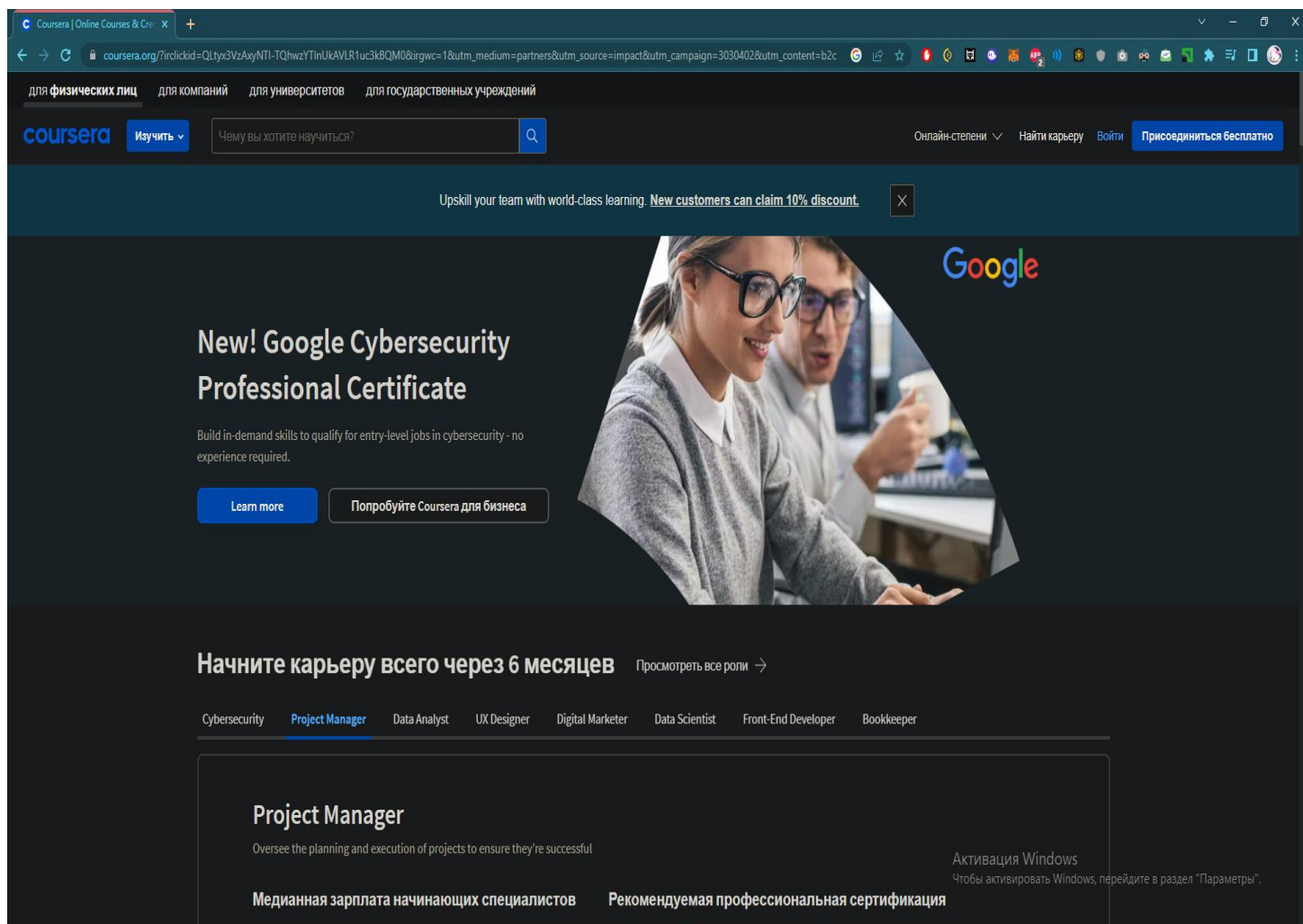


Рис. 1.2 Головна сторінка Coursera. [19]

На додаток до цих прикладів, було розроблено багато інших інформаційних систем для дистанційного навчання, кожна зі своїми унікальними функціями та можливостями. Однак розробка ефективної інформаційної системи дистанційного навчання потребує не лише технологій. Це також вимагає залучення кваліфікованих викладачів, здатних розробити та провести високоякісні онлайн-курси.[19]

3. Moodle

Moodle - система управління курсами, також відома як система управління навчанням або віртуальне середовище. Є аббревіатурою від англ. Modular Object-Oriented Dynamic Learning Environment. Є вільним веб-додатком, що надає можливість створювати сайти для онлайн-навчання. На рисунку 1.3 зображено головну сторінку Moodle.[16]

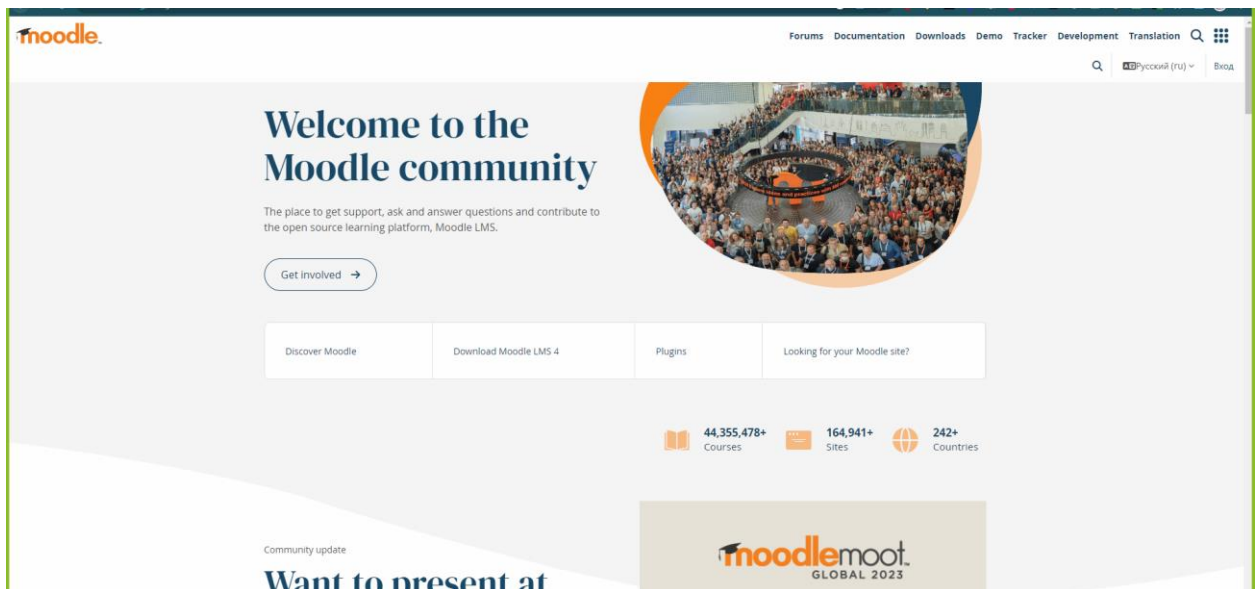


Рис. 1.3 Головна сторінка Moodle. [18]

Підсумовуючи, розвиток інформаційних систем для дистанційного навчання змінив спосіб надання освіти. Ці системи дали можливість студентам отримати доступ до високоякісної освіти з будь-якої точки світу та надали можливості для навчання протягом усього життя. Постійний розвиток цих систем матиме вирішальне значення для того, щоб освіта залишалася доступною та актуальною в епоху цифрових технологій.[18]

1.1.3 ОРГАНІЗАЦІЙНІ ПИТАННЯ ПРИ ПРОЕКТУВАННІ СИСТЕМИ

Розробка інформаційної системи дистанційного навчання передбачає низку рішень, які необхідно прийняти, щоб забезпечити успіх проекту. Ці рішення можна розділити на кілька категорій, зокрема:

1. *Технологія*: Перше рішення, яке необхідно прийняти, це вибір технологічної платформи для інформаційної системи. Це включає рішення щодо обладнання, програмного забезпечення та мережевої інфраструктури.

2. *Зміст контенту*: Ще одним важливим рішенням є вибір і розробка контенту для системи дистанційного навчання. Це включає рішення щодо матеріалів курсу, мультимедійних ресурсів та інструментів оцінювання.

3. *Педагогіка*: розробка ефективних педагогічних стратегій також має вирішальне значення для успіху системи дистанційного навчання. Це включає рішення щодо дизайну навчання, залучення студентів і методів оцінювання.

4. *Адміністрування*: адміністрування системи дистанційного навчання вимагає ретельного планування та прийняття рішень щодо таких питань, як управління набором, служби підтримки студентів і навчання викладачів.

Сутність проблемних ситуацій у розробці інформаційної системи для дистанційного навчання може включати такі проблеми, як недостатнє фінансування чи ресурси, відсутність інституційної підтримки чи участі зацікавлених сторін, технічні труднощі з апаратними чи програмними системами та проблеми, пов'язані із залученням студентів. і мотивація.[13]

1.1.4 ФАКТОРИ ЩО ВПЛИВАЮТЬ НА ТЕМАТИКУ ТА РОЗРОБКУ ІС

Цілі вирішення цих проблемних ситуацій зазвичай зосереджені на покращенні доступу до освіти для здобувачів, які можуть не мати доступу до традиційних умов навчання через географічні чи інші обмеження. Інші цілі можуть включати покращення результатів навчання студентів за рахунок більш персоналізованого навчання та підвищення інституційної ефективності за рахунок зменшення витрат, пов'язаних із фізичними засобами.

Організаційний рівень прийняття системи дистанційного навчання може змінюватися залежно від таких факторів, як інституційна культура, ставлення викладачів до технологічного навчання та попит студентів на онлайн-курси. Успішне впровадження вимагає ретельного розгляду цих факторів, щоб забезпечити широку підтримку з боку всіх зацікавлених сторін. [10]

Розробка інформаційної системи для дистанційного навчання є складним завданням, яке включає ряд факторів, які можуть впливати на середовище прийняття рішень. Ці фактори можна охарактеризувати за ступенем їх впливу і можуть включати:

Технологічні фактори: технологічний ландшафт постійно розвивається, і рішення, пов'язані з розробкою інформаційної системи для дистанційного навчання, повинні враховувати нові технології та тенденції. Ступінь впливу технологічних факторів на прийняття рішень може бути високим, оскільки вони можуть мати значний вплив на ефективність і зручність використання системи.

Освітні чинники: природа предмета, що викладається, і потреби учнів є важливими міркуваннями при розробці інформаційної системи для дистанційного навчання. Ступінь впливу освітніх факторів може бути високим, оскільки вони можуть мати значний вплив на проектування та доставку системи.

Організаційні фактори: структура та культура організації, відповідальної за розробку інформаційної системи для дистанційного навчання, також можуть впливати на прийняття рішень. Ступінь впливу організаційних факторів може бути високим, оскільки вони можуть впливати на розподіл ресурсів і здатність впроваджувати зміни.

Економічні фактори: Витрати, пов'язані з розробкою та підтримкою інформаційної системи для дистанційного навчання, є важливим фактором. Ступінь впливу економічних факторів може бути високим, оскільки вони можуть впливати на рівень інвестицій у систему та ресурси, доступні для розвитку.

Щоб приймати оптимальні рішення, пов'язані з розробкою інформаційної системи для дистанційного навчання, необхідно добре розуміти фактори, які

впливають на середовище прийняття рішень. Це вимагає глибокого розуміння предмета, а також знання передового досвіду та нових тенденцій у цій галузі.

Щоб ефективно орієнтуватися в ситуаціях прийняття рішень, пов'язаних із розробкою інформаційної системи дистанційного навчання, важливо враховувати унікальні особливості цих ситуацій. Вони можуть включати такі фактори, як невизначеність, складність і потреба у співпраці та залученні зацікавлених сторін.

Враховуючи складність прийняття рішень у цій сфері, існує очевидна потреба в системі управління інформацією для підтримки процесу прийняття рішень. Така система може містити такі функції, як аналітика даних, інструменти для спільної роботи та алгоритми підтримки прийняття рішень, щоб допомогти зацікавленим сторонам приймати обґрунтовані та оптимальні рішення, пов'язані з розробкою інформаційної системи для дистанційного навчання. Вибір відповідних сценаріїв підтримки для прийняття рішень залежатиме від ряду факторів, включаючи організаційну культуру, технічну інфраструктуру та ресурси, доступні для впровадження.[6]

Підсумовуючи, розробка інформаційної системи для дистанційного навчання є складним і складним завданням, яке вимагає ретельного розгляду ряду факторів, які можуть впливати на середовище прийняття рішень. Щоб приймати оптимальні рішення в цій галузі, необхідно мати глибоке розуміння предмета, а також знання передового досвіду та нових тенденцій у цій галузі. Надійна система управління інформацією може сприяти прийняттю рішень у цій сфері та покращити загальну ефективність системи.

Вирішення проблемних ситуацій при розробці інформаційної системи дистанційного навчання може включати низку альтернативних рішень, які необхідно оцінити на основі їх ефективності у вирішенні конкретних завдань. Ці рішення можуть бути періодичними або постійними, залежно від характеру проблеми та ресурсів, доступних для її вирішення.

Обмеження та критерії прийняття рішень щодо розробки інформаційної системи для дистанційного навчання можуть включати такі фактори, як бюджетні обмеження, технічна здійсненність, педагогічна ефективність та інституційні пріоритети. Ці критерії необхідно ретельно розглянути, щоб переконатися, що рішення приймаються таким чином, щоб максимізувати переваги системи при мінімізації потенційних ризиків або негативних наслідків.

1.2 ОБГРУНТУВАННЯ ВИБОРУ ПІДХОДІВ І ТЕХНОЛОГІЙ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.

Для аналізу готових систем дистанційного навчання за критеріями розглянемо деякі популярні системи: Blackboard, Moodle і Canvas.

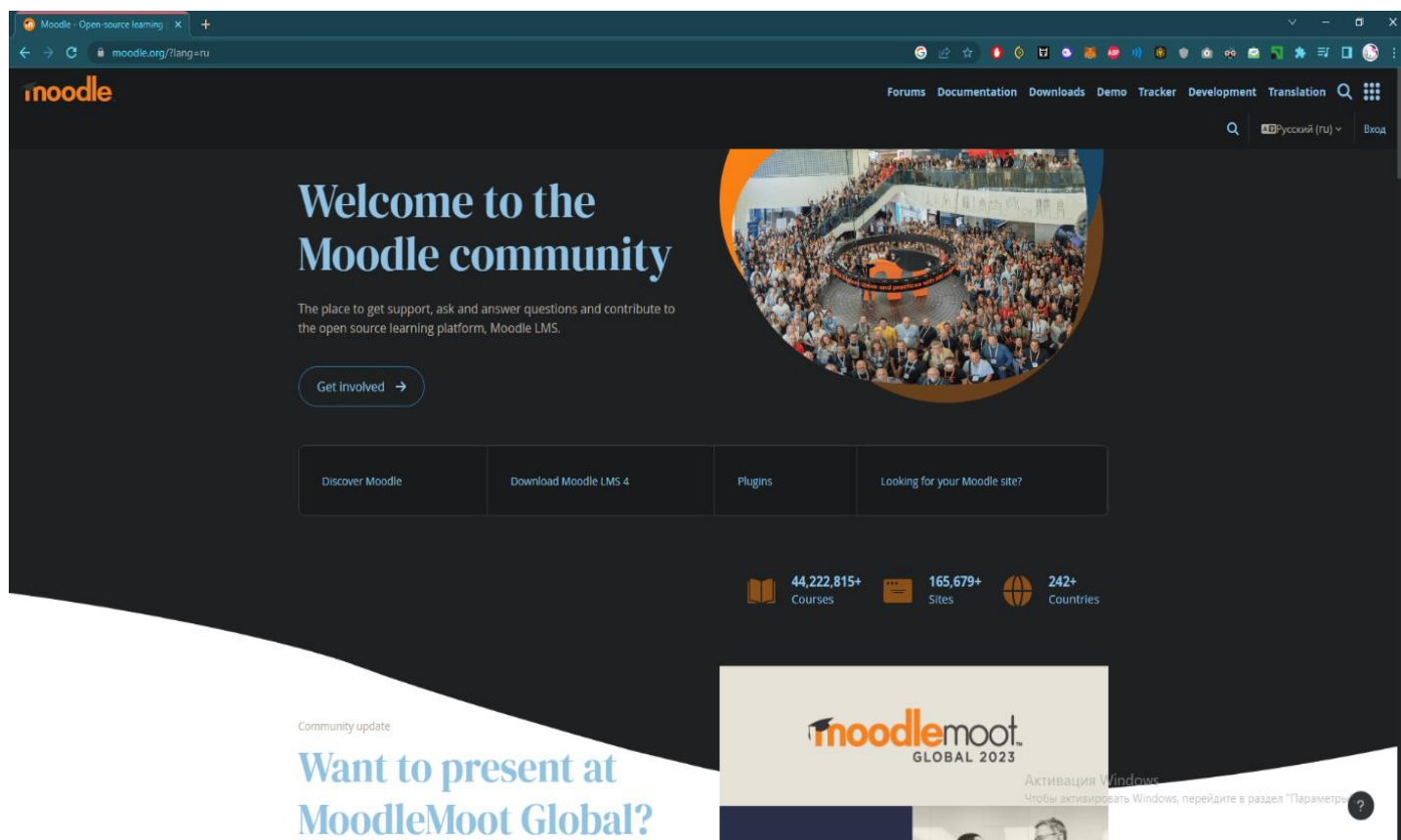


Рис. 1.3 Головна сторінка Moodle [18]

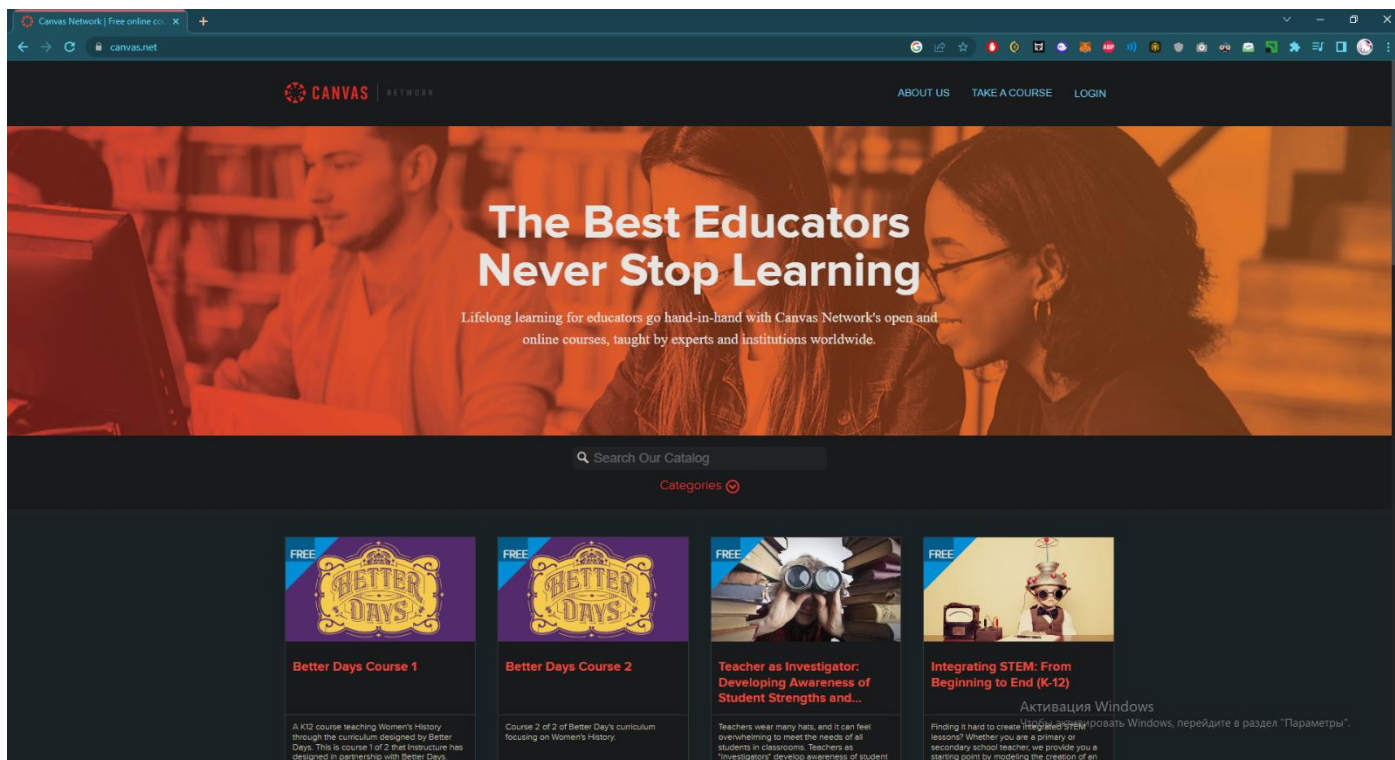


Рис. 1.4 Головна сторінка Canvas [19]

- Критерії за якими буде проводитись порівняння:
Повний функціонал інформаційних систем управління
- Концептуальна схема управління системою
- Ситуація, пов'язана з прийняттям рішень, функціями та виконанням завдань, на які орієнтована інформаційна система управління
- Заплановані підходи до організації бази даних/бази даних, бази знань, програмного забезпечення
- Технологічні та організаційні аспекти
- Особливості використання природного інтерфейсу
- Склад і форма результатів, які можна отримати з їх допомогою використання системи
- Переваги від використання інформаційних систем управління, сертифікованих її користувачами

1. Повний функціонал інформаційних систем управління:

Усі три системи пропонують повний набір інструментів для керування навчальним контентом, полегшення спілкування між викладачами та студентами та відстеження прогресу студентів. Вони надають такі функції, як дискусійні форуми, відеоконференції, інструменти для створення вмісту та аналіз студентів.

2. Концептуальна схема управління системою:

Blackboard і Canvas використовують централізований підхід до управління, коли системою керує центральний адміністратор, який керує встановленням, оновленнями та обслуговуванням системи. Moodle, з іншого боку, дотримується децентралізованого підходу, коли кожна установа чи користувач може встановлювати та керувати власним екземпляром програмного забезпечення.

3. Ситуація, пов'язана з прийняттям рішень, функціями та виконанням завдань, на які орієнтована інформаційна система управління:

Усі три системи розроблені для підтримки дистанційного навчання, а їхні можливості прийняття рішень орієнтовані на такі завдання, як створення курсу, керування контентом і відстеження прогресу студентів. Вони надають функції аналітики та звітності, які дозволяють інструкторам приймати рішення на основі даних.

4. Заплановані підходи до організації бази даних/бази даних, бази знань, програмного забезпечення:

Blackboard і Canvas використовують власне програмне забезпечення та системи баз даних, а Moodle — це платформа з відкритим кодом, яка дозволяє користувачам налаштовувати програмне забезпечення та базу даних відповідно до своїх потреб.

5. Технологічні та організаційні аспекти:

Усі три системи є хмарними та пропонують масштабовані рішення для організацій різного розміру. Вони забезпечують інтеграцію з іншими системами керування навчанням, інструментами веб-конференцій і мобільними додатками.

6. Особливості використання природного інтерфейсу:

Усі три системи мають інтуїтивно зрозумілий і зручний інтерфейс, який полегшує навігацію та створення вмісту. Вони пропонують такі функції, як перетягування та скидання, адаптивний дизайн та мультимедійна інтеграція.

7. Склад і форма результатів, які можна отримати з їх допомогою використання системи:

Усі три системи забезпечують комплексні функції звітності та аналітики, які дозволяють інструкторам контролювати прогрес студентів, відстежувати результати навчання та визначати сфери, які потрібно вдосконалити. Вони також надають інструменти для відгуків та оцінювання студентів.

8. Переваги від використання інформаційних систем управління, сертифікованих її користувачами:

Дослідження показали, що використання систем управління інформацією для дистанційного навчання може призвести до кращого залучення студентів, збільшення показників утримання та кращих результатів навчання. Blackboard, Moodle і Canvas отримали позитивні відгуки від своїх користувачів і широко використовуються в освітньому секторі.

Підсумовуючи, усі проаналізовані системи пропонують комплексні рішення для дистанційного навчання та надають низку функцій для підтримки потреб інструкторів і користувачів. Хоча кожна система має свої унікальні функції та підхід, усі вони забезпечують надійні можливості керування інформацією, які можуть підтримувати прийняття рішень у контексті дистанційного навчання.[10]

Компоненти sqlite3, Python, HTML, CSS та JS були обрані для реалізації проекту з кількох причин:

- Sqlite3 - це компактна, швидка та надійна СУБД, яка може бути легко інтегрована в проекти на Python, не потребуючи налаштування окремого сервера баз даних. Вона підходить для малих та середніх проектів, де не потрібна велика кількість одночасних підключень. Sqlite3 є зручним рішенням для зберігання інформації про курси та користувачів у нашому проекті.
- Python - це проста та ефективна мова програмування, яка має безліч бібліотек та фреймворків, що дозволяють розробляти веб-додатки. Ми використовували Python для розробки серверної частини програми та обробки даних у базі даних.
- HTML – це стандартна мова розмітки для створення веб-сторінок. Він використовується для структурування контенту на сторінці та визначення його зовнішнього вигляду. Ми використовували HTML для створення основних елементів інтерфейсу сторінок.
- CSS - це мова таблиць стилів, яка використовується для оформлення HTML-документів. Він дозволяє визначити кольори, шрифти, розміри, розташування елементів на сторінці та багато іншого. Ми використовували CSS для створення красивого та привабливого зовнішнього вигляду наших сторінок.
- JS - це мова програмування, яка використовується для створення інтерактивних елементів на сторінці, таких як меню, форми та анімації. Ми використовували JS для створення простої системи тестування та перевірки завдань у нашому проекті.

Таким чином, всі ці компоненти були використані в проекті з певною метою та враховані їхні переваги та недоліки для вирішення завдань проекту.

Є кілька способів покращити проект та додати функціональність. Наприклад, можна додати такі можливості:

- Аутентифікація користувачів: додавання можливості реєстрації користувачів та захист сторінки з курсами паролем або токеном аутентифікації.
- Оцінки та звіти: додавання можливості створювати тести з безліччю питань, надання користувачеві зворотного зв'язку за їх відповідями та створення звітів про пройдені курси.
- Автоматизація: додавання автоматичного створення курсів та завдань на основі готових шаблонів, а також автоматичної перевірки результатів тестів.

Для реалізації цих функцій можна використовувати різні інструменти та технології, наприклад:

- Flask або Django для створення веб-програми з реєстрацією та автентифікацією користувачів.
- Бібліотеки для створення тестів, наприклад, pytest або unittest.
- Django-написані пакети, такі як django-comments та django-forum, для додавання коментарів та форумів.
- Інтеграція з сервісами для автоматичного створення курсів та перевірки результатів тестів, наприклад Coursera або edX.

Також можна використовувати бібліотеки та фреймворки для покращення зовнішнього вигляду веб-застосунку, наприклад, Bootstrap для створення адаптивного дизайну, jQuery для додавання інтерактивності та Vue.js для створення компонентного підходу до розробки веб-додатків.

Концептуальна модель проекту системи може виглядати так:

Сутності:

1. Користувач: має ім'я користувача, зберігає дані автентифікації
2. Курс: має назву, короткий опис, список завдань, оцінки за кожне завдання
3. Завдання: має номер, текст завдання, правильну відповідь, оцінку за завдання
4. Рішення: пов'язує користувача із завданням, зберігає відповідь користувача, дату відправлення, оцінку за завдання

Зв'язки:

1. Користувач може бути зареєстрований на кількох курсах, а на один курс може бути зареєстровано декілька користувачів
2. Курс може містити декілька завдань, а кожне завдання може бути включене до кількох курсів
3. Рішення пов'язує користувача та завдання, кожне рішення має лише одне завдання та одного користувача
4. Дана модель дозволяє зручно зберігати дані та зв'язки між ними, а також забезпечує можливість додавання нових сутностей у майбутньому, наприклад додавання нових типів завдань або можливість коментування рішень іншими користувачами.

Схематичний опис проекту:

Проект складається з трьох компонентів: фронтенд (HTML, CSS, JS), бекенд (Python) та бази даних (SQLite). Користувач взаємодіє з фронтендом, який надсилає запити на бекенд. Бекенд обробляє запити, взаємодіє з базою даних для отримання та збереження даних та надсилає відповіді на фронтенд.

При запуску проекту, бекенд підключається до бази SQLite і створює необхідні таблиці, якщо вони ще не існують. Адміністратор може додавати нові курси на сайт, використовуючи форму на сторінці адміністратора. Користувач

може вибрати курс, який йому потрібно пройти, та розпочати роботу над ним. Система тестування оцінює правильність заданого завдання. Адміністратор може переглядати завдання, які користувачі здали, і має доступ до оцінок.[7]

Кожен запит від користувача обробляється на сервері, де бекенд використовує шаблони Django для створення динамічних сторінок на основі даних з бази даних. Фронтенд використовує HTML, CSS та JavaScript для відображення сторінок на стороні клієнта.

Зокрема, використання вказаних вище додаткових компонентів у проєкті дозволить створити простий і в той же час мати багатофункціональний веб-сайт для навчання.

РОЗДІЛ II. АНАЛІТИЧНИЙ РОЗДІЛ. ХАРАКТЕРИСТИКА ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМ ТА МЕТОДИ І МОДЕЛІ

2.1 СТРУКТУРА І ХАРАКТЕРИСТИКА СИСТЕМИ

Архітектура проекрованої системи включає в себе:

1. Сервер на якому розташовано ядро системи дистанційного навчання. Необхідні підтримуючі файли, БД, сховище даних, інтерфейсні модулі.

Структура проекту включає:

1. Серверна частина
2. Клієнтська частина

За умови наявності доступу до мережі інтернет, та при встановленні на пристрої браузера та **власної системи дистанційного навчання “Ignatyev”**.

Для візуалізації структури ІС було взято за основу еРС діаграми, нотації BPMN.

BPMN (Business Process Model and Notation) – це нотація для моделювання бізнес-процесів, яка використовується для візуалізації та документування бізнес-процесів, їх послідовності та взаємодії між учасниками процесу. BPMN дозволяє учасникам процесу легко розуміти його структуру та дії, а також допомагає автоматизувати бізнес-процеси.

BPMN-діаграми складаються з елементів, що представляють дії, події та потоки даних. Вони також містять елементи для опису потоків управління та інформаційних потоків між учасниками процесу. Діаграми BPMN можуть бути простими або складними, і вони можуть використовуватися для моделювання будь-яких бізнес-процесів, включаючи процеси замовлення, управління проектами, виробництва тощо.[9]

BPMN-діаграми використовуються для визначення бізнес-процесів та їх автоматизації у системах управління бізнес-процесами (BPM-системах). Ці системи використовують діаграми BPMN для визначення бізнес-правил та

автоматизації бізнес-процесів, зменшуючи час та вартість виконання бізнес-операцій, збільшуючи ефективність та точність процесів та покращуючи керованість бізнесу.

BPMN-діаграми можуть використовуватися для моделювання та управління бізнес-процесами на всіх рівнях організації, від окремих бізнес-процесів до комплексних систем управління бізнес-процесами. Вони можуть допомогти організаціям збільшити ефективність, скоротити час та вартість виконання бізнес-операцій, покращити якість продукції чи послуг та підвищити задоволеність клієнтів.

Далі буде наведена BPMN діаграма в якій присутні дії як користувача, так і адміністратора.

Дії користувача:

1. Користувач може записатись на курс необхідний йому.
2. Користувач може переглянути свої оцінки.
3. Користувач може надіслати завдання

Дії адміністратора:

1. Адміністратор може передивитись завдання користувача.
2. Адміністратор може додати новий курс для впровадження його на платформі.

Автоматизований процес перевірки роботи студента, проте для правильності видачі оцінки, завдання буде повторно переглянуте Адміністратором.

Також присутня БД курсів, в якій зберігається вся необхідна інформація.

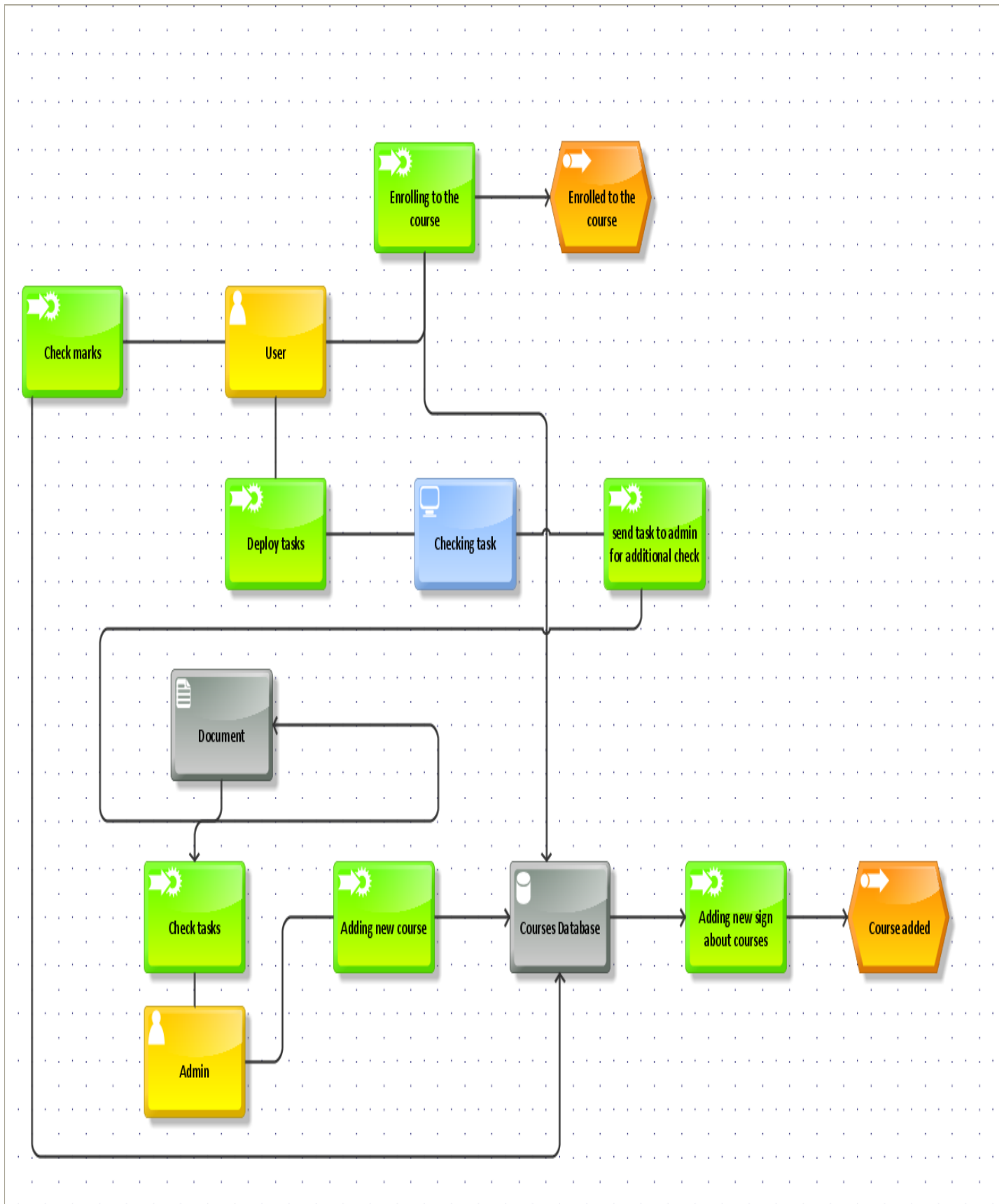


Рис. 2.1 BPMN діаграма. Джерело автора.

2.1.1 ДЕРЕВО ЗАДАЧ КЛІЄНТСЬКОЇ ЧАСТИНИ

Дерево завдань (Task Tree) – це інструмент для управління проектами, який дозволяє візуалізувати завдання та їх залежності у вигляді деревоподібної структури. Воно дозволяє розбити проект на дрібніші підзавдання та визначити їх залежності один від одного, щоб спростити планування та управління проектом.

Дерево завдань може включати різні рівні, починаючи від великих етапів проекту і закінчуючи дрібними завданнями, які можуть бути виконані за кілька годин. Кожне завдання може мати свої дочірні завдання, які залежать від її виконання, а також попередні завдання, від виконання яких це завдання залежить.

У дереві завдань можна задавати терміни виконання, призначати відповідальних виконання кожної завдання, відстежувати прогрес і оцінювати ризики. Це допомагає спростити планування проекту та управління його виконанням, а також зробити процес більш прозорим та передбачуваним.

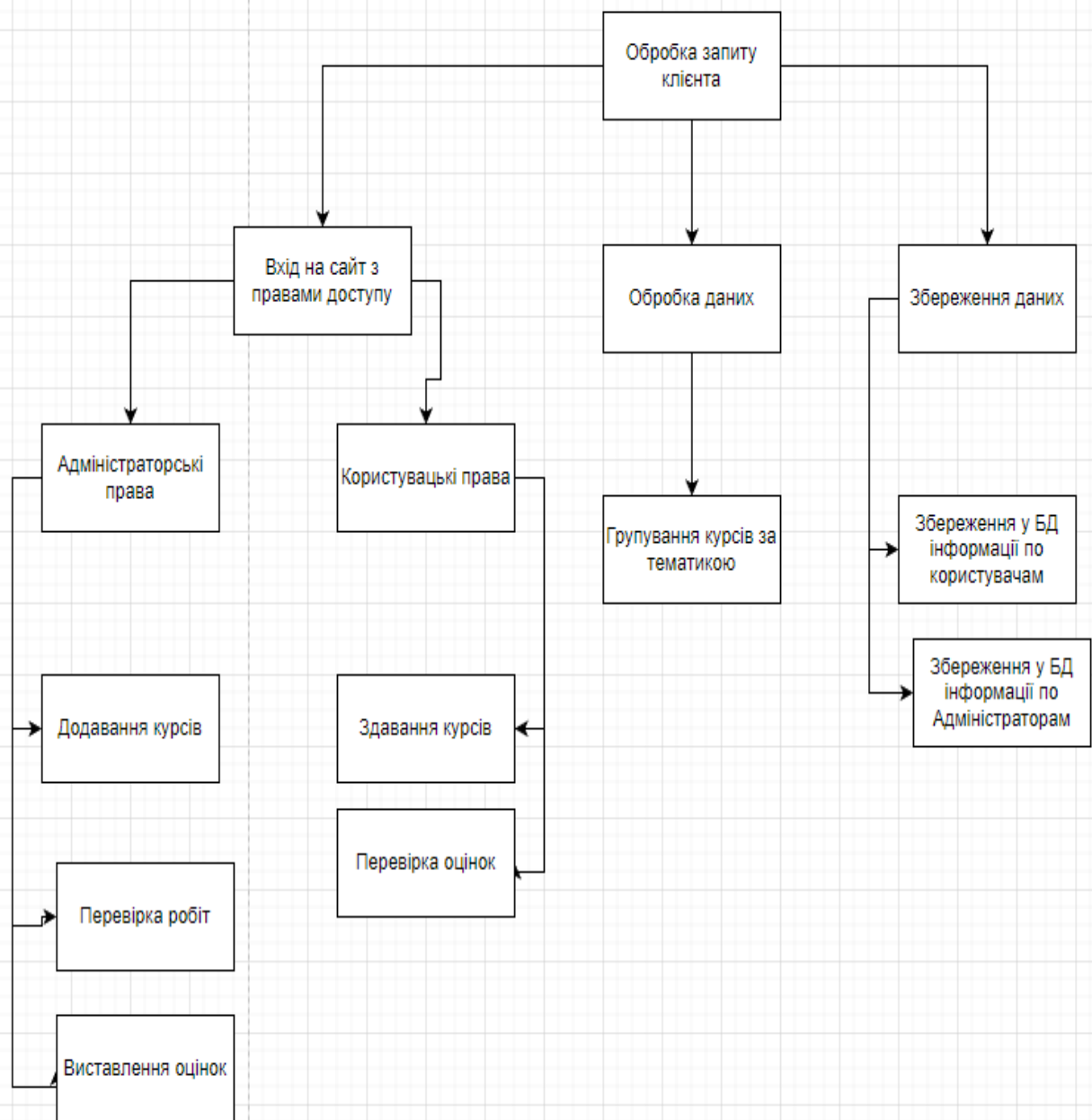


Рис. 2.2 Дерево задач Клієнтської частини. Джерело автора.

2.1.2 ФУНКЦІОНАЛЬНІ ТА НЕФУНКЦІОНАЛЬНІ ВИМОГИ

Функціональні та нефункціональні вимоги - це набір вимог, які необхідні для розробки та впровадження проекту.

Функціональні вимоги визначають, якою має бути поведінка системи та що вона повинна робити, тоді як нефункціональні вимоги визначають, якими мають бути характеристики системи, такі як продуктивність, надійність, безпека, зручність використання тощо.[4]

Функціональні вимоги для проекту ІС для дистанційного навчання:

- Реєстрація нових користувачів
- Вхід для зареєстрованих користувачів
- Створення курсу адміністратором
- Оцінка робіт користувачів адміністратором
- Можливість пошуку курсів за ключовими словами
- Перегляд інформації про курс, такий як опис, рівень складності, вартість та ін.
- Проходження курсів та отримання сертифікатів

Нефункціональні вимоги для проекту ІС для дистанційного навчання:

- Продуктивність: система повинна бути швидкою та чуйною, щоб користувачі не чекали надто довго при завантаженні сторінок та виконанні запитів
- Надійність: система має бути стабільною та надійною, щоб користувачі могли отримувати доступ до курсів та не втрачали свого прогресу під час проходження курсів.

- **Безпека:** система має бути захищена від несанкціонованого доступу та зловживань, щоб користувачі могли бути впевнені у збереженні своїх особистих даних та фінансової інформації
- **Зручність використання:** система має бути простою та інтуїтивно зрозумілою, щоб користувачі могли легко знаходити потрібну інформацію та використовувати функціонал системи без необхідності одержання спеціального навчання.

Всі ці вимоги допомагають визначити, якими мають бути основні функції та характеристики проекту ІС для дистанційного навчання, щоб він був зручним та корисним для користувачів та адміністраторів.

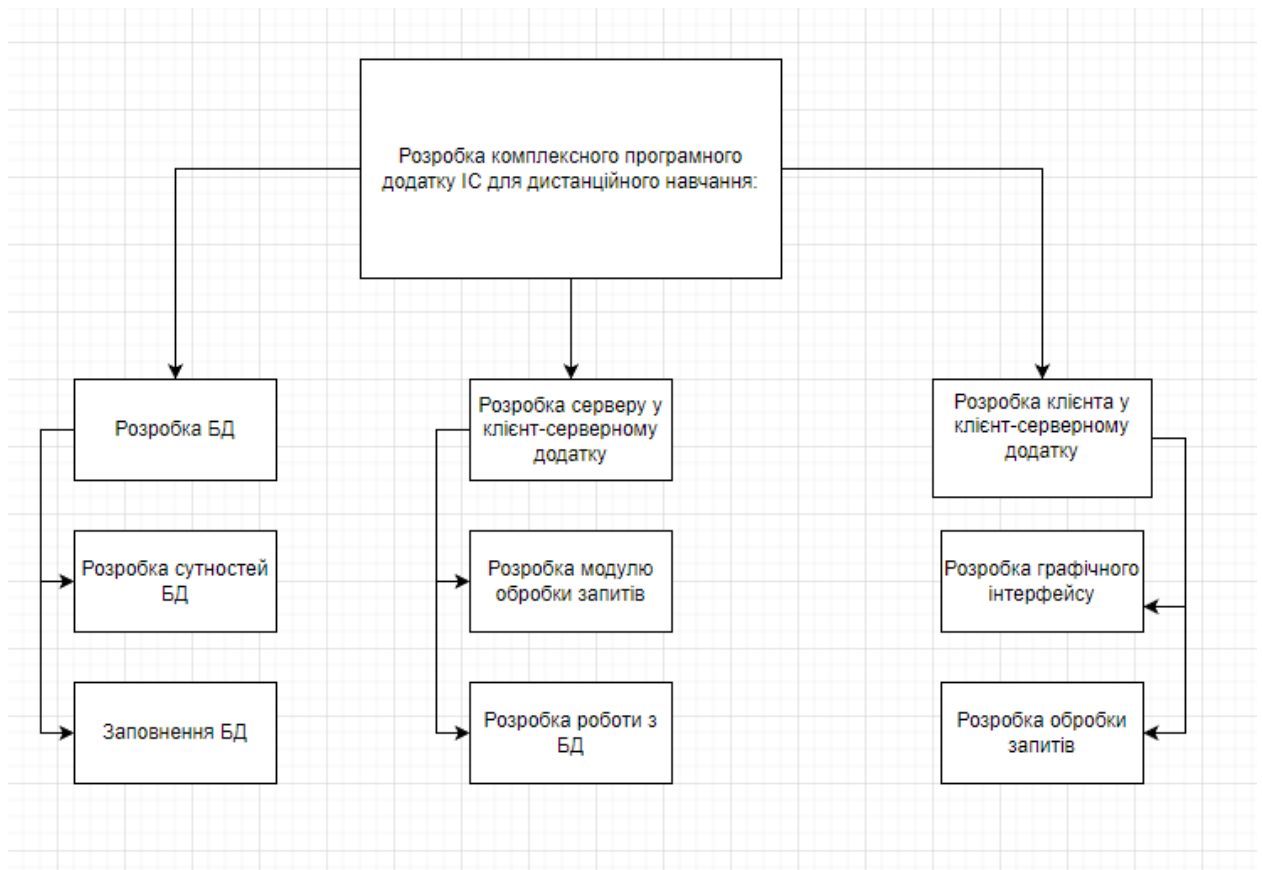


Рис. 2.3 «Дерево» функцій дипломного проекту. Джерело автора.

2.1.3 КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА

Клієнт-серверна архітектура - це розподілена архітектура, в якій програма розділена на дві основні частини: клієнтську та серверну. Клієнт – це інтерфейс користувача, через який користувач взаємодіє з додатком. Сервер – це машина, яка надає ресурси та послуги клієнту. Клієнт та сервер взаємодіють між собою за допомогою мережі.

Клієнт та сервер можуть бути розміщені на різних машинах та працювати на різних платформах. Клієнт зазвичай надає інтерфейс користувача і управляє логікою відображення інформації. Він надсилає запити на сервер, який обробляє їх та відправляє назад клієнту необхідні дані.[13]

Сервер надає клієнту дані та ресурси, які він запитує, такі як бази даних, файли, послуги мережі тощо. Він також обробляє запити, надає безпеку та керує доступом до даних та ресурсів.

Клієнт-серверна архітектура має кілька переваг, у тому числі:

Розширюваність: клієнт та сервер можуть бути розміщені на різних машинах, що дозволяє легко масштабувати систему за потреби.

Безпека: сервер може надавати захист даних та авторизацію користувачів, що забезпечує безпечнішу роботу програми.

Легкість супроводу: логіка програми розподілена між клієнтом і сервером, що дозволяє покращувати і супроводжувати програму частинами, а не всього.

Покращення продуктивності: сервер може надавати оптимізовані та швидкі запити до даних, що підвищує продуктивність програми.

Однак клієнт-серверна архітектура має свої недоліки, наприклад:

Залежність від мережі: програма працює лише за наявності зв'язку між клієнтом і сервером, тому збої в мережі можуть призвести до неполадок у роботі програми.[6]

Складність розробки: розробка програми, заснованої на клієнт-серверній архітектурі, може бути складнішою, ніж розробка програми, що працює на одній машині.

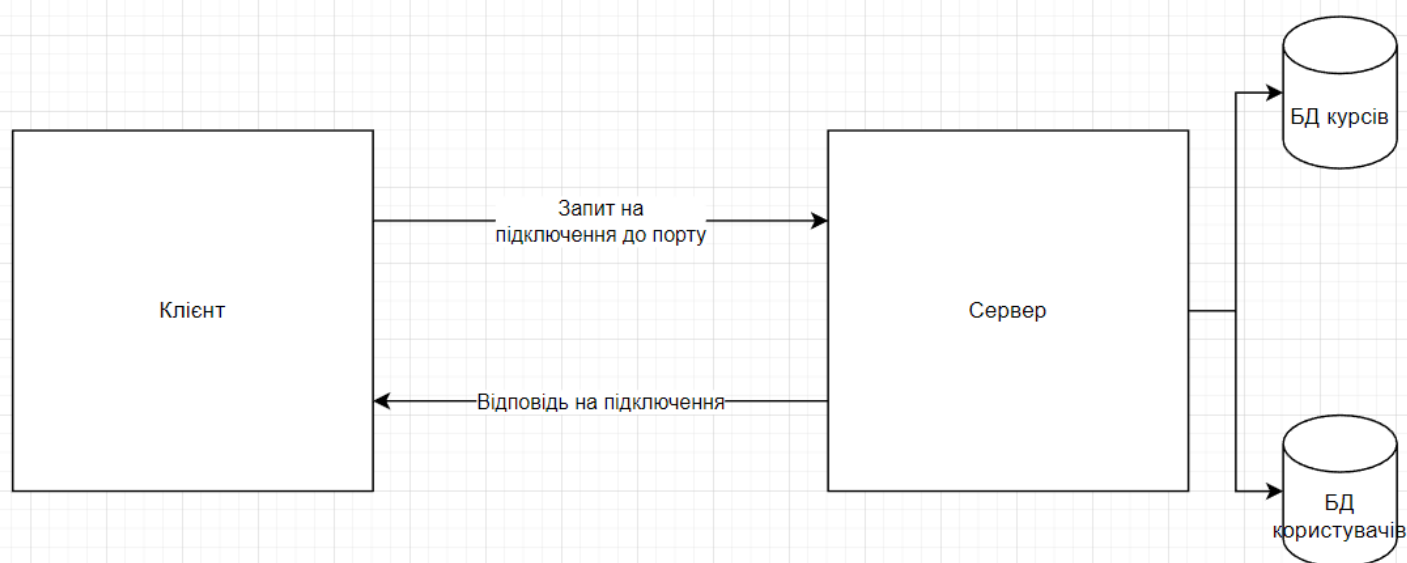


Рис. 2.4 Клієнт-серверна архітектура. Джерело автора.

За сутність клієнта відповідає форма, якою користується користувач, що взаємодіє з фізичним сервером, а саме відправкою та прийняттям запитів.

За базу даних з користувачами відповідає сховище інформації, де зберігаються дані для авторизації користувачів.

За базу даних з курсами відповідає сховище інформації, де зберігаються усі необхідні дані стосовно курсів

Вхідні дані:

- Відео заняття
- Типи користувачів

Вихідні дані:

- Web-платформа для online навчання

Буде проходити під управлінням:

- Технічне завдання
- Бюджет

Механізмом створення буде:

- Графічний макет
- Розробник
- CMS WordPress[15]

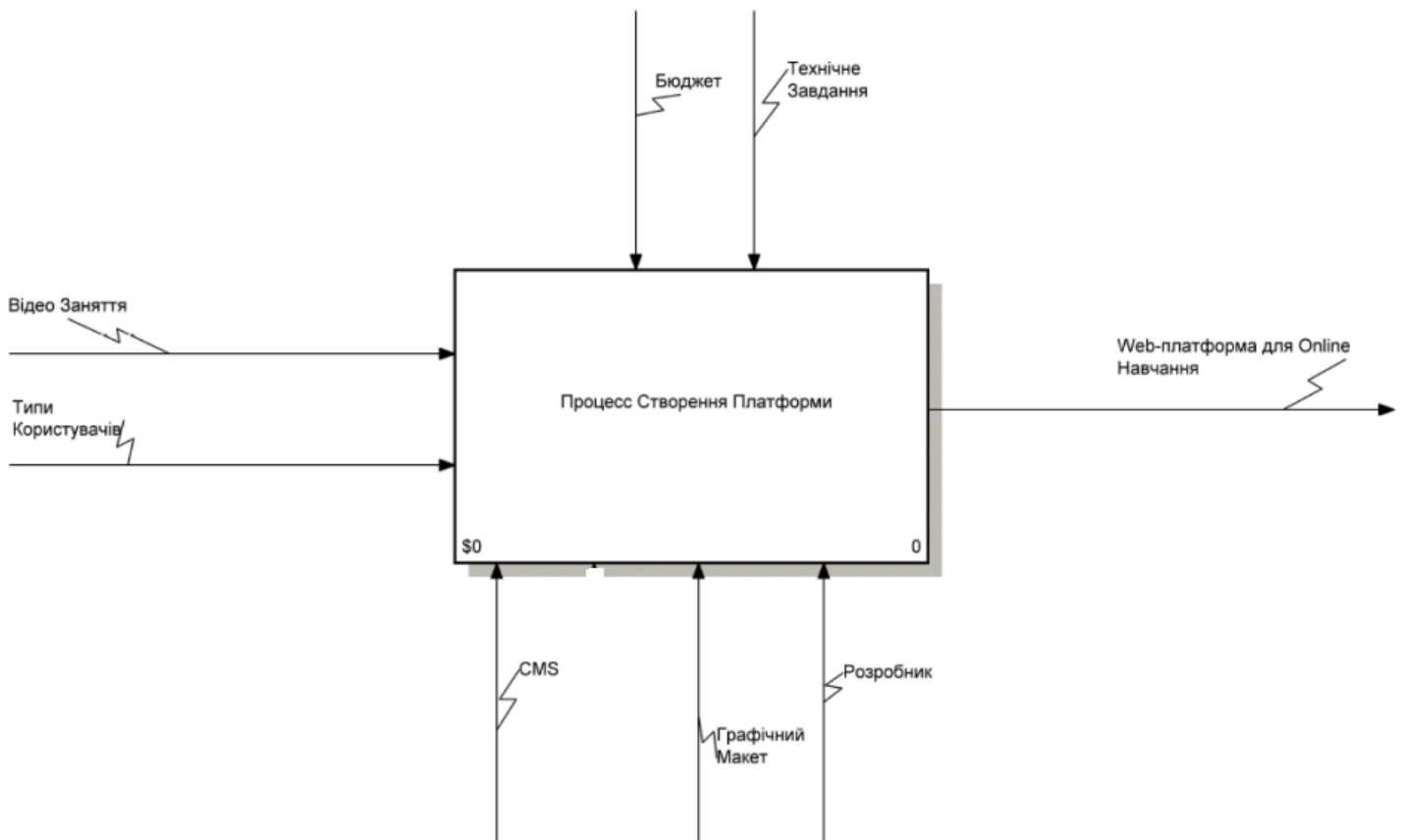


Рис. 2.5 Контекстна діаграма. Джерело автора.

2.1.4 ДІАГРАМИ ВСІХ РІВНІВ ДЕКОМПОЗИЦІЇ

Декомпозиція першого рівня

На початковому етапі створення веб-платформи необхідно провести відбір системи, яка буде використовуватися для розробки проекту. При цьому ми керуємося вимогами технічного завдання та встановленим бюджетом проекту. Для виконання цієї задачі можна застосовувати різноманітні механізми, одним з них є система управління вмістом (CMS). У нашому випадку ми звернемо увагу на можливість використання CMS, оскільки це може дозволити зменшити час та витрати на розробку проекту. Крім того, розробник, який займатиметься аналізом конкурентних платформ та вибором оптимальної, є необхідним членом команди на цій стадії розробки проекту.[8]

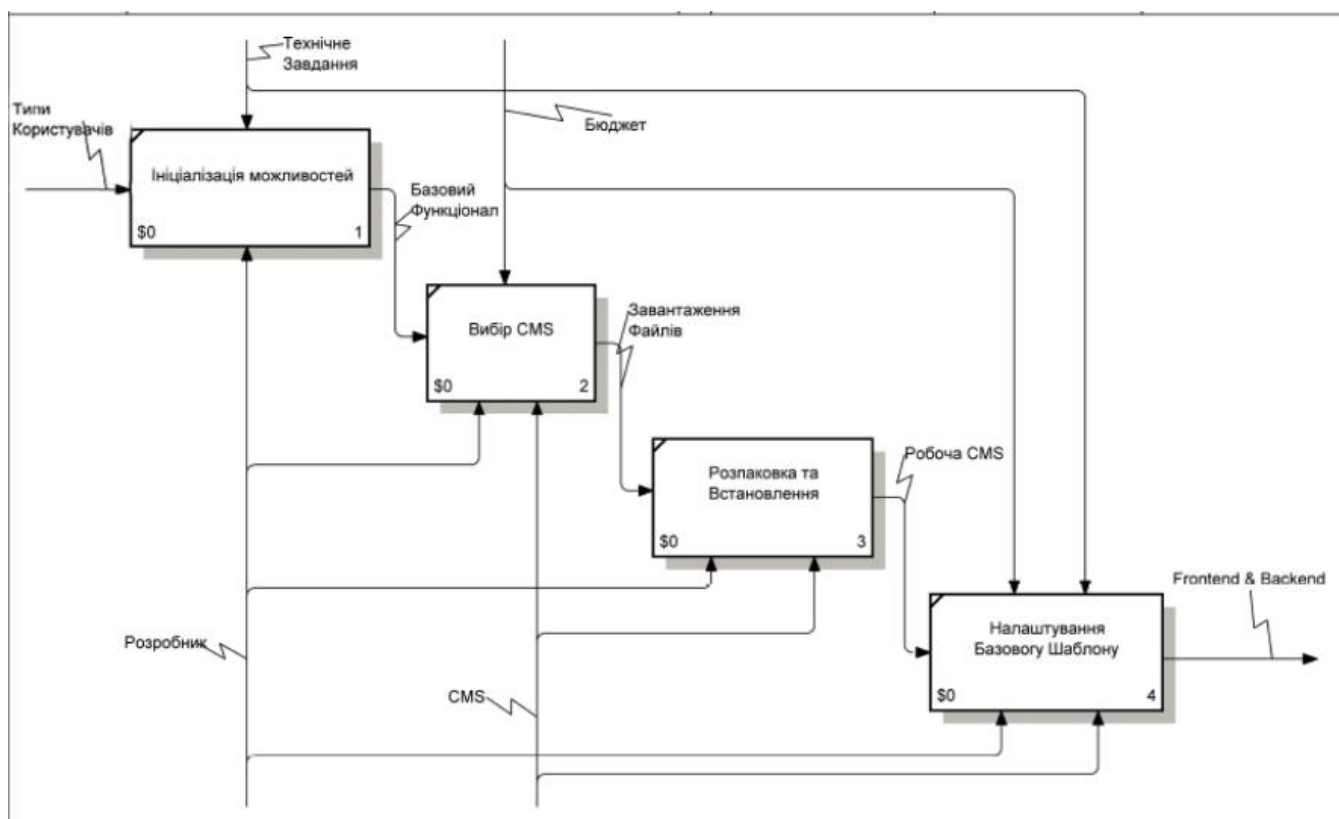


Рис. 2.6 Декомпозиція першого рівня. Джерело автора.

На другому етапі проекту розробляється сам веб-сайт. Для початку потрібно визначитись зі стилем майбутньої платформи і створити макет в форматі PSD. Для цього використовується програма для дизайну Adobe Photoshop 2020. Окрім цього, на цій стадії потрібно реалізувати адаптивний дизайн, щоб сайт коректно відображався на різних пристроях, а також зробити певні налаштування і запустити майже робочу платформу.

Для розробки веб-сайту потрібно враховувати потенціал викладачів та студентів курсів, які використовуватимуть платформу. Також необхідно враховувати можливість викладання відео-занять. Діяти на цій стадії проекту будемо під управлінням бюджету, щоб не вийти за рамки задачі, та технічного завдання для виконання необхідних базових умов.

Весь тягар по реалізації платформи ляже на розробника, який буде задіяний на цій стадії. Він повинен мати достатній рівень кваліфікації та досвід веб-розробки, щоб забезпечити успішну реалізацію проекту. Крім того, необхідна систематичність, уважність до деталей та готовність працювати в команді, щоб успішно виконати поставлені завдання.

Далі наведено декомпозицію другого рівня.

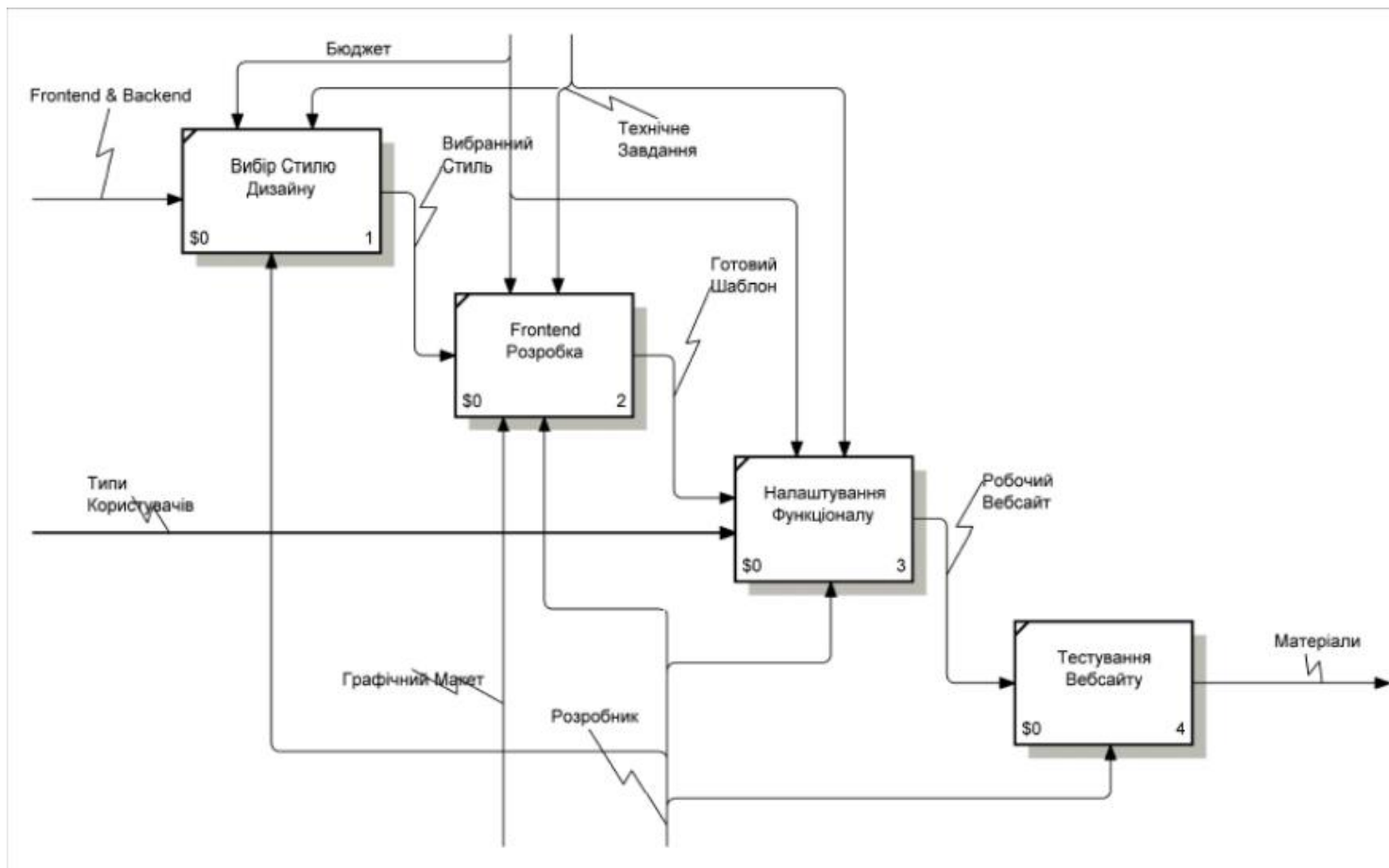


Рис. 2.7 Декомпозиція другого рівня. Джерело автора.

Третій етап розробки веб-платформи передбачає важливий процес додавання занять. Цей етап є ключовим, оскільки саме заняття надають користувачам можливість навчатися віддалено. Щоб реалізувати цей етап, необхідно створити курс та наповнити його матеріалами, які можуть бути у форматі відео, текстових приміток та описів. Крім того, необхідно налаштувати відображення за часом та категоріями, щоб користувачі могли знайти потрібні заняття з легкістю. Важливим етапом є також тестування перегляду курсу, щоб користувач не зіштовхнувся з проблемою на початку навчання.

Під час розробки цього етапу необхідно дотримуватися бюджету та виконувати технічне завдання. Для забезпечення якісної розробки необхідна підтримка викладача, який зможе створювати та додавати матеріали на платформу. Механізмом реалізації додавання занять виступатиме розробник, який буде відповідати за створення курсу та додавання матеріалів на платформу.

Далі наведено декомпозицію третього рівня

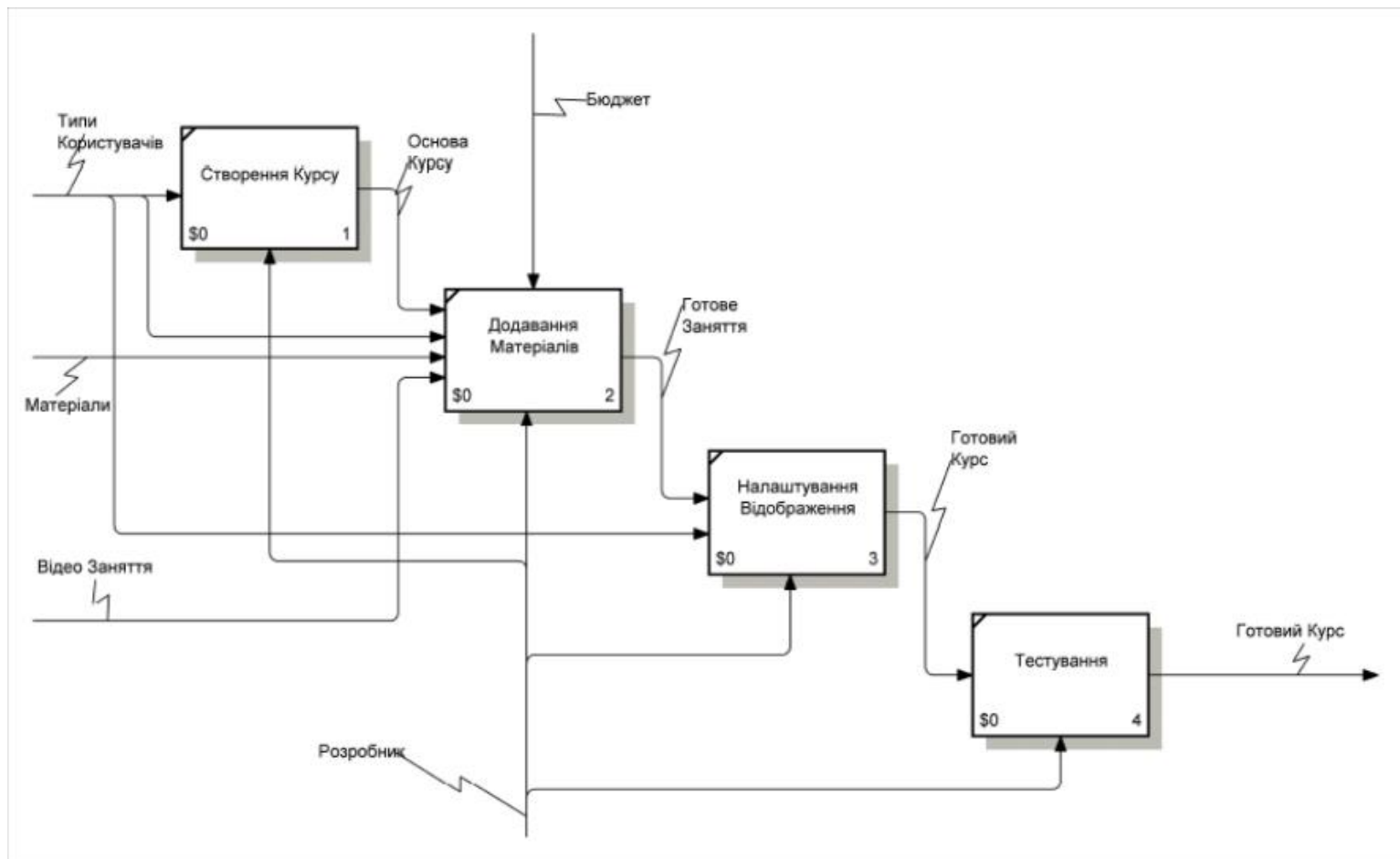


Рис. 2.8 Декомпозиція третього рівня. Джерело автора.

2.1.5 ДІАГРАМА ПРЕЦЕДЕНТІВ

Діаграма прецедентів - це графічне зображення взаємодії між акторами та системою, що відображає функціональність системи з точки зору користувача. Прецеденти описують поведінку системи та різні дії користувачів в контексті використання системи. Діаграма прецедентів допомагає розробникам зрозуміти вимоги користувачів та створити функціональну систему, яка задовольняє їх потреби.

У проєкті веб-платформи для навчання було обрано таких акторів:

1. Викладач – користувач, котрий має доступ до бази даних, володіє можливістю створювати нові курси, заняття та опитування, а також налаштовувати особливості відображення курсу.

2. Адмін – має повний функціонал можливостей викладача, але має можливість редагувати дані користувачів, а також робити зміни на сайті в візуальному редакторі.

3. Користувач – кінцевий користувач, котрий прагне придбати курс та добути нових знань.

Після завершення визначення акторів необхідно визначитись з варіантами використання. Вони повинні повністю відповідати технічному завданню.

Варіанти використання Web-платформи:

- авторизація;
- реєстрація;
- редагування даних;
- додавання та редагування користувачів;
- створювати курс;
- створювати заняття;
- створювати опитування;
- дивитися курс;
- дивитися план;
- дивитися заняття;
- дивитися викладача;

Після налаштування зв'язків між акторами та варіантами використання було побудовано Use Case Diagram.

Діаграма прецедентів відображає ці прецеденти у вигляді овалів, які містять назви прецедентів. Актори, які взаємодіють з системою, відображаються вище прецедентів. Взаємодія між акторами та системою відображається лініями, які вказують на те, який прецедент виконується. Діаграма прецедентів також може містити додаткову інформацію, таку як опис прецедентів та список вимог до системи.

Діаграма прецедентів є важливим інструментом для розробки функціональності системи та забезпечення відповідності вимогам користувачів. Вона допомагає уникнути невідповідності між вимогами користувачів та функціональністю системи та забезпечує ефективну комунікацію між розробниками та користувачами.

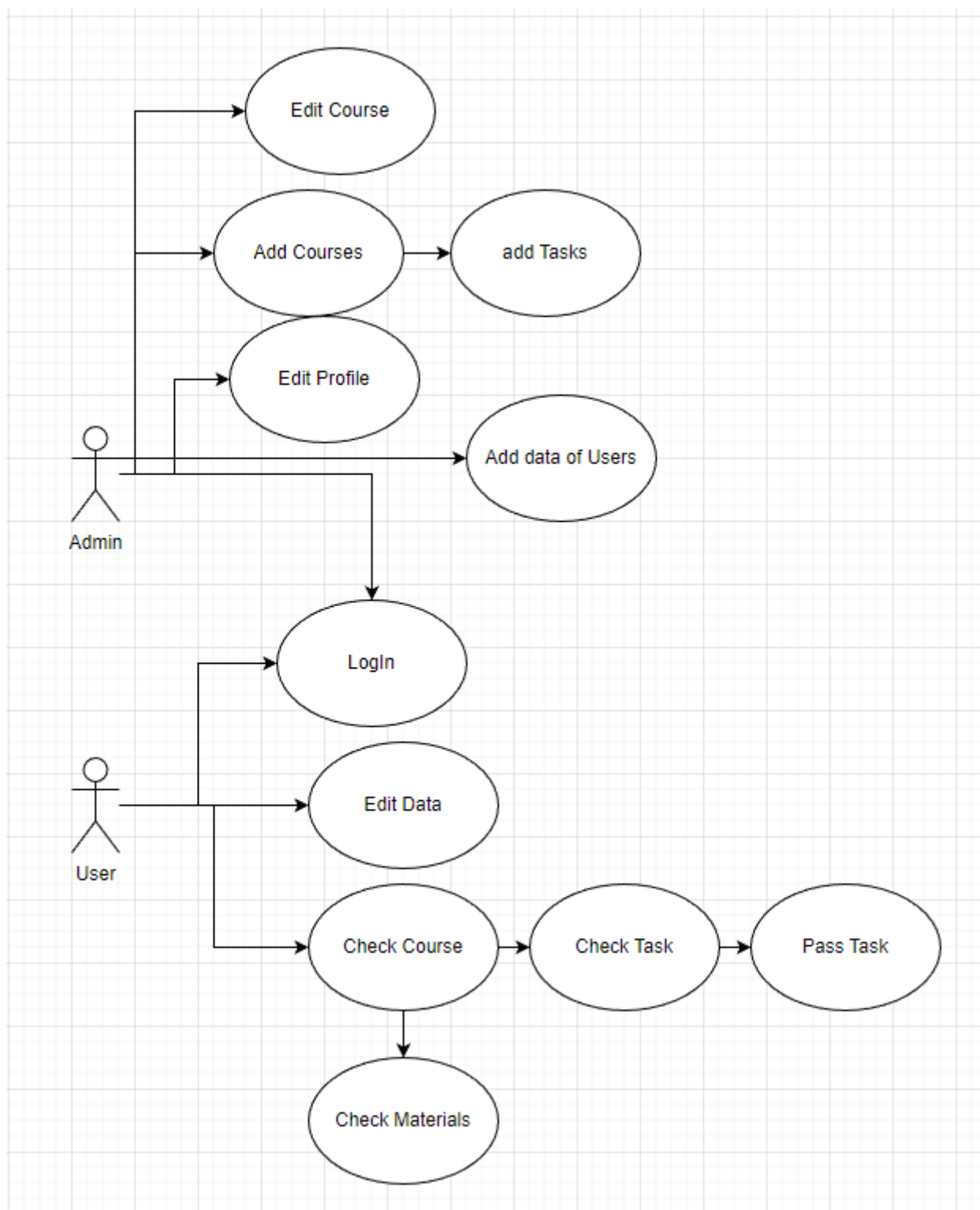


Рис. 2.9 Діаграма прецедентів. Джерело автора.

Для більш детального аналізу прецедентів, було створено діаграми послідовності, що допомагають описати послідовність дій, що відбуваються в системі. Крім того, для кожного типу користувачів були визначені окремі прецеденти. Для зручності, в діаграмах було введено загальний тип "Користувач", щоб уникнути зайвого дублювання і забезпечити більшу чіткість при визначенні прецедентів.

Діаграми послідовності - це потужний інструмент для моделювання взаємодії між об'єктами в системі, що допомагає краще зрозуміти логіку роботи системи і виявити можливі проблеми в процесі розробки. Зокрема, за допомогою діаграм послідовності можна визначити послідовність взаємодії між користувачем та системою, яка складається з запитів, відповідей та повідомлень.

В результаті, діаграми прецедентів та діаграми послідовності разом надають повну картину процесу взаємодії між користувачем та системою, що є ключовим для успішної розробки програмного продукту.

На рисунку – 2.9 зображено послідовність виконання дій користувача для задачі завдань.

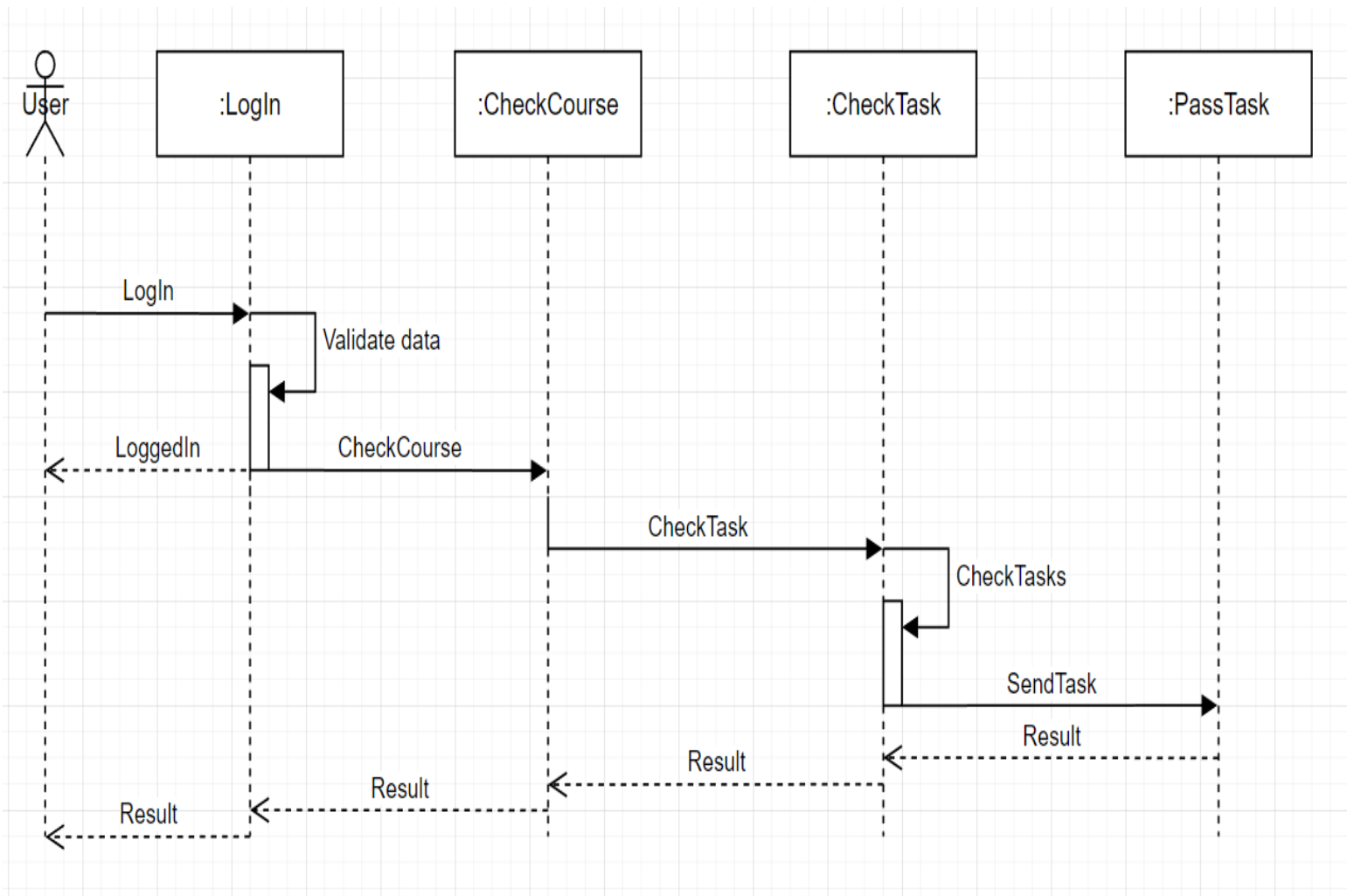


Рис. 2.10 Діаграма послідовності користувача для здачі завдань. Джерело автора.

На рисунку 2.10 зображено послідовність дій адміністратора для додавання нового курсу.

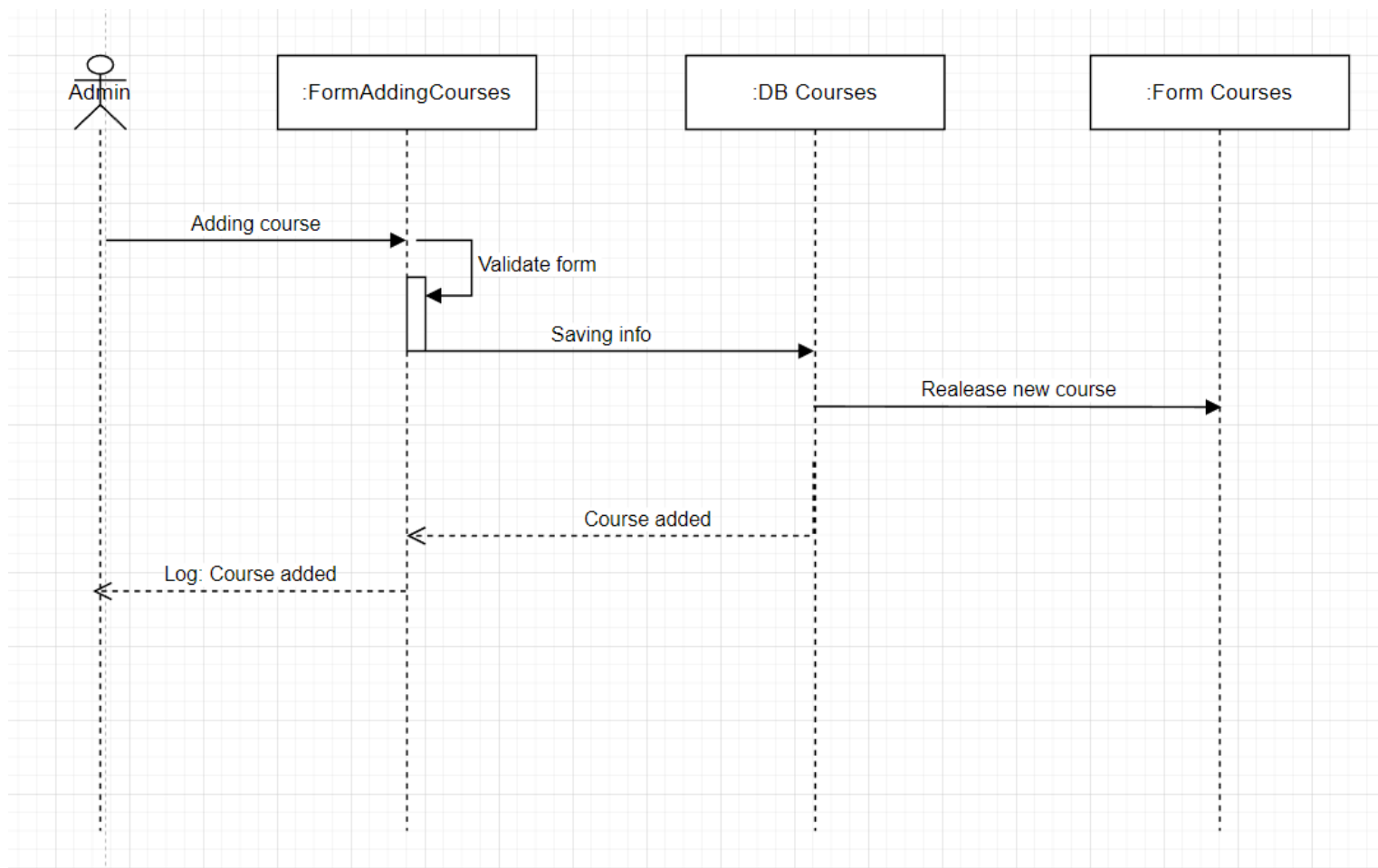


Рис. 2.11 Діаграма послідовності адміністратора для додавання нового курсу.

Джерело автора.

На рисунку 16 зображено послідовність дій вчителя для редагування інформації курсів та своєї особистої.

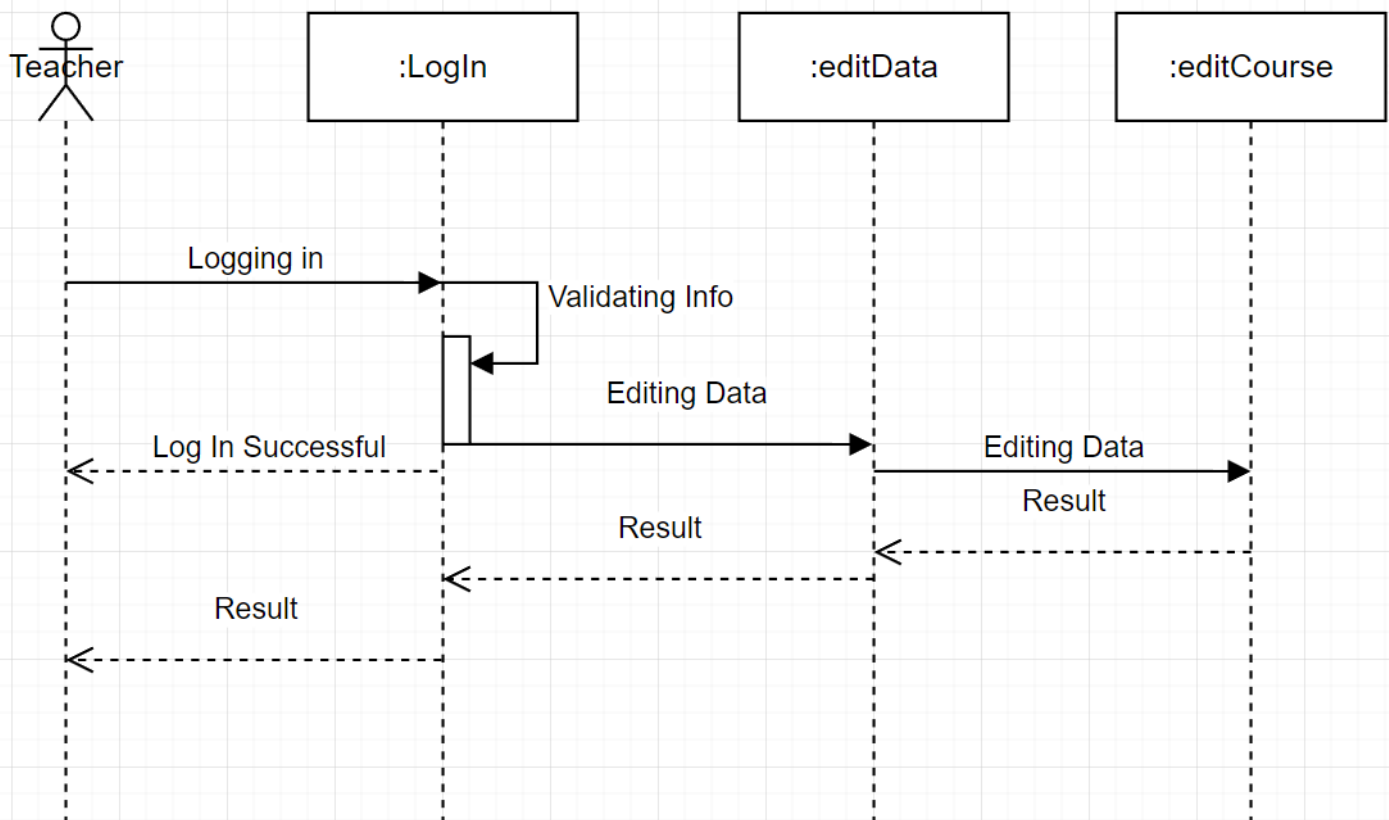


Рис. 2.12 Діаграма послідовності вчителя для редагування інформації курсів та своєї особистої. . Джерело автора.

2.2. МЕТОДИ ТА МОДЕЛІ В ІНФОРМАЦІЙНИХ УПРАВЛЯЮЧИХ СИСТЕМАХ ТА ТЕХНОЛОГІЯХ

В розроблюваній інформаційній системі для рекомендації курсів для користувачів можна використати алгоритм колаборативного фільтрування. Цей алгоритм базується на розрахунку схожості між користувачами та курсами, і використовує цю інформацію для рекомендації курсів.

$$P(course|user) = \prod_{u \in similar_users} \max(r_{u, course} + 1) \quad (2.1)$$

Де

- $P(course|user)$ - ймовірність того, що користувач оцінить курс;
- $r_{\{u, course\}}$ - оцінка користувача u для курсу $course$;
- $similar_users$ - безліч користувачів, у яких збігаються інтереси з поточним користувачем;
- $\max(r_u)$ - максимальна оцінка користувача u ;
- $|C|$ - кількість курсів.

Далі для знаходження користувачів зі схожими інтересами:

$$similar_users = \{u: I_u = I_{user}\} \setminus \{user\} \quad (2.2)$$

Де

- I_u - вектор інтересів користувача u ;
- $I_{\{user\}}$ - вектор інтересів поточного користувача.

Приклад: якщо інтереси користувачів дорівнюють:

	Курс 1	Курс 2	Курс 3	Курс 4
Користувач 1	1	0	1	0

2	Користувач	1	1	0	0
3	Користувач	0	1	1	1
4	Користувач	0	0	1	0

Таблиця 2.1 Приклад інтересів користувачів

а інтереси поточного користувача дорівнюють [1, 0, 1, 0], то користувачі зі схожими інтересами будуть:

$$similar_{users} = \{user1, user3\} \quad (2.3)$$

Далі ми будемо обчислювати ймовірності того, що користувач оцінить кожен курс. Для цього ми використовуємо формулу на основі байєсівської теореми:

$$P(C_i|U) = P(U|C_i)P(C_i) / \sum_j P(U|C_j)P(C_j) \quad (2.4)$$

де

- $P(C_i | U)$ - ймовірність того, що користувач оцінить курс C_i в залежності від його інтересів U
- $P(U | C_i)$ - ймовірність того, що користувач з інтересом U оцінить курс C_i
- $P(C_i)$ - апіорна ймовірність курсу C_i
- $\sum_j P(U | C_j)P(C_j)$ - нормалізаційна константа, щоб забезпечити, що сума всіх ймовірностей дорівнює 1.

В нашому коді ми використовуємо апіорну ймовірність кожного курсу рівну $1/\text{len}(\text{courses})$, тому що ми не маємо жодних апіорних знань про популярність кожного курсу серед користувачів.

Тоді для кожного курсу C_i ми обчислюємо $P(U | C_i)$ наступним чином:

$$P(U|C_i) = \prod_{u \in U_{\text{sim}}} r_{u, \max + \text{len}(\text{courses}) - i + 1} \quad (2.5)$$

Де:

- U_{sim} - множина користувачів, які мають інтереси, які співпадають з U
- $r_{u,i}$ - оцінка користувача u для курсу C_i $r_{u,\max}$ - максимальна оцінка, яку користувач u виставив для будь-якого курсу.

Далі вже для рекомендацій курсів користувачам використовується один з методів колаборативного фільтрування

У цьому методі схожість між користувачами обчислюється за допомогою кореляції Пірсона, його формула:

$$similarity(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)(r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \mu_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \mu_v)^2}} \quad (2.6)$$

Де:

- $similarity(u, v)$ - схожість між користувачами u та v
- r_{ui} - оцінка, яку користувач u виставив курсу i
- μ_u - середнє значення оцінок користувача u за всі курси
- I_{uv} - множина курсів, які оцінювали як користувач u , так і користувач v .

Далі, коли ми визначили схожість між користувачами, ми можемо використовувати його для прогнозування того, яку оцінку користувач u поставить курсу i , якого він ще не оцінював. Формула для прогнозування оцінки:

$$r_{ui} = \mu_u + \frac{\sum_{v \in N(u)} similarity(u, v)(r_{vi} - \mu_v)}{\sum_{v \in N(u)} similarity(u, v)} \quad (2.7)$$

Де:

- $\hat{r}_{ui}$ - прогнозована оцінка користувача u для курсу i
- $N(u)$ - множина користувачів, які мають схожість з користувачем u .

Застосування алгоритмів рекомендацій дозволяє пропонувати студентам курси, які можуть відповідати їхнім інтересам та потребам навчання. Це може збільшити мотивацію студентів до вивчення матеріалу, що позитивно впливає на їхній успіх та результативність. Також, модуль для рекомендацій по курсах здобувачам допомагає підтримувати високу активність та інтерес користувачів до навчання на платформі, що може збільшити їхню лояльність та залученість до продукту.

РОЗДІЛ III. КОНСТРУКТИВНИЙ РОЗДІЛ

3.1. ПРОЕКТУВАННЯ БАЗИ ДАНИХ ДЛЯ ІНФОРМАЦІЙНИХ СИСТЕМ

3.1.1. ПРОЕКТУВАННЯ ІНФОЛОГІЧНОЇ(ЛОГІЧНОЇ) МОДЕЛІ БД

Інформаційна модель бази даних (ІМБД) є візуальним зображенням структури даних, які зберігаються в базі даних. Вона описує, як дані пов'язані між собою, як вони організовані, та як вони можуть бути використані в різних аспектах діяльності бізнесу або організації.

ІМБД допомагає вирішувати проблеми, пов'язані з розробкою та управлінням базами даних, такі як:

- Визначення необхідних даних для відображення бізнес-процесів;
- Організація структури даних та взаємозв'язків між ними;
- Визначення правил доступу до даних;
- Вирішення проблем інтеграції даних;
- Планування розширення бази даних та збільшення її продуктивності.

Інформаційна модель дозволяє описати бізнес-процеси та взаємозв'язки між ними, що дозволяє забезпечити правильність зберігання даних та швидкий доступ до них. Крім того, ІМБД допомагає уникнути помилок при розробці бази даних та підвищує рівень зрозумілості та ефективності взаємодії з базою даних для різних зацікавлених сторін.

На рисунку 3.1 зображено інфологічну модель БД

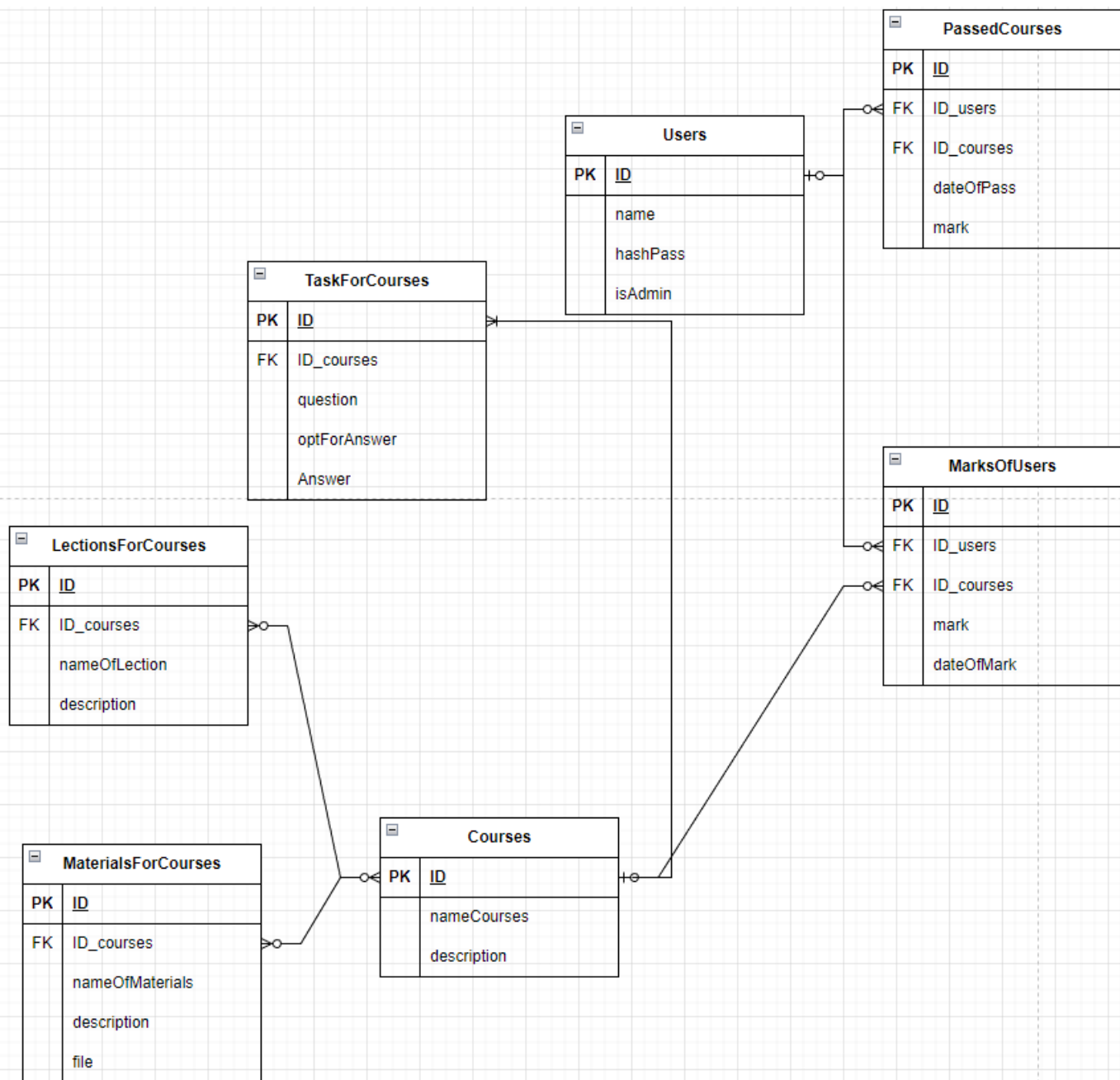


Рис. 3.1 Інфологічна модель БД. Джерело автора.

3.1.2. ОБҐРУНТУВАННЯ ВИБОРУ СКБД ТА ПРОЕКТУВАННЯ ФІЗИЧНОЇ МОДЕЛІ БАЗИ ДАНИХ

Для проекту дистанційного навчання можна використовувати різні СУБД, такі як MySQL, PostgreSQL, Oracle, Microsoft SQL Server та багато інших. Різні СУБД мають різні особливості, переваги та недоліки, і вибір конкретної СУБД залежить від потреб проекту, вимог до продуктивності, масштабування, доступності технічної підтримки та багатьох інших факторів.

SQLite є дуже популярною СУБД, яка має наступні переваги:

- Легка встановлюється та налаштовується, що робить її ідеальним вибором для невеликих проектів.
- Не вимагає окремого сервера, тому що вона є локальною файловою СУБД, що забезпечує її простоту у використанні та підтримці.
- Має маленький розмір, що забезпечує високу продуктивність в порівнянні з іншими СУБД.
- Має хорошу підтримку стандарту SQL та розширенням SQLite, які дозволяють використовувати більшість функцій, які доступні в більш складних СУБД.
- Може бути використана на різних платформах, включаючи Windows, MacOS, Linux та багато інших.
- Має безкоштовну ліцензію, що робить її привабливим варіантом для невеликих проектів та підприємств з обмеженим бюджетом.

З огляду на всі переваги, SQLite є хорошим вибором для проекту дистанційного навчання, зокрема для зберігання інформації про користувачів, курси, завдання та інші сутності. Однак, якщо проект розрахований на велику кількість користувачів та вимагає високої продуктивності та масштабованості, можуть бути розглянуті інші СУБД, наприклад, MySQL або PostgreSQL. Обидві ці СУБД є потужними інструментами для зберігання великих обсягів даних та забезпечують високу швидкодію завдяки оптимізації запитів та можливості

горизонтального масштабування. Однак, в порівнянні з SQLite вони можуть вимагати більш складного налаштування та адміністрування.

Обравши SQLite для проекту дистанційного навчання, можна отримати декілька переваг. По-перше, це легковага СУБД, що не вимагає встановлення окремого серверу баз даних та може бути вбудована в додаток. По-друге, SQLite забезпечує швидкий доступ до даних завдяки своїй архітектурі та оптимізованим запитам. По-третє, SQLite має простий та зрозумілий синтаксис SQL-запитів, що спрощує розробку та підтримку проекту.

Далі на рисунку 3.2 буде наведено даталогічну модель БД.

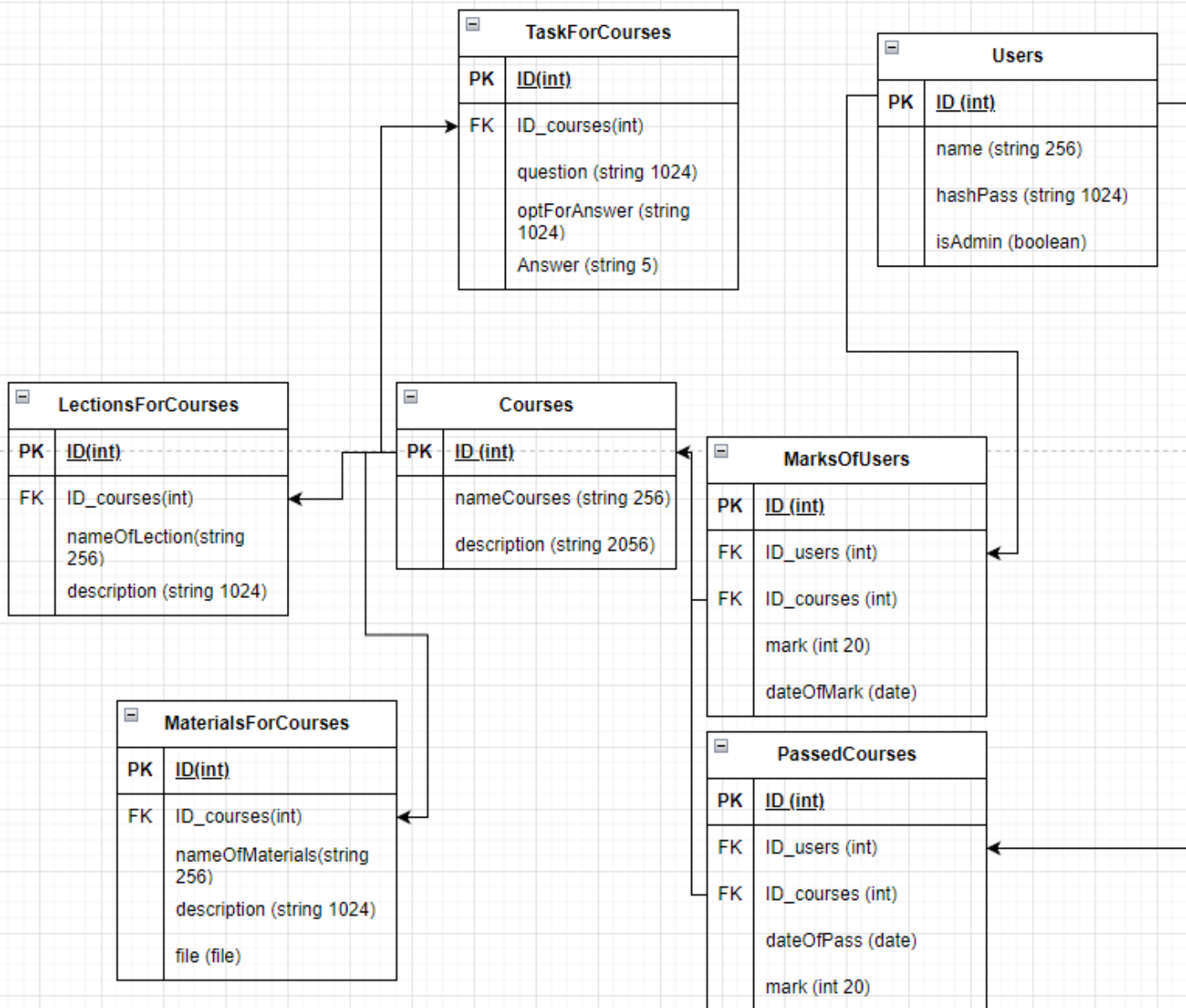


Рис. 3.2 Дatalogічна модель БД. Джерело автора.

Детальний опис масивів даних, що зберігаються в БД розписано в таблицях 3.1 - 3.7

Опис масиву «Users»

Найменування масиву – Users

Ідентифікатор масиву – ID

Найменування носія інформації – твердий магнітний диск

Максимальний об'єм масиву – 10000000 записів.

Довжина запису – 61 символ (або байт).

Метод організації – Послідовний.

Ключі упорядкування – Код ідентифікатора масиву.

Найменування	Ідентифікатор у програмі	Умовне позначення	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Код ідентифікатора	id	--	int	PK	-	так	ІНД	--
Ім'я користувача	name	--	string	-	-	так	-	--
Хеш пароля	hashPass		string			так		
isAdmin	isAdmin		boolean			так		

Опис масиву «LectonsForCourses»

Найменування масиву – LectonsForCourses

Ідентифікатор масиву – ID

Найменування носія інформації – твердий магнітний диск

Максимальний об'єм масиву – 10000000 записів.

Довжина запису – 61 символ (або байт).

Метод організації – Послідовний.

Ключі упорядкування – Код ідентифікатора масиву, FK Код ідентифікатора курсів.

Найменування	Ідентифікатор у програмі	Умове позначення	Формат	Бізнес-правила				Логічні чи семантичні
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Код ідентифікатора	id	--	int	PK	-	так	ІНД	
Код ідентифікатора курсів	ID_courses	--	int	FK	-	так	ІНД	courses
Назва лекції	nameOfLectio n	--	string		-	так	ІНД	
Опис лекції	desc		string			так		

Опис масиву «Courses»

Найменування масиву – Courses

Ідентифікатор масиву – ID

Найменування носія інформації – твердий магнітний диск

Максимальний об'єм масиву – 10000000 записів.

Довжина запису – 61 символ (або байт).

Метод організації – Послідовний.

Ключі упорядкування – Код ідентифікатора масиву.

Найменування	Ідентифікатор у програмі	Умове позначення	Формат	Бізнес-правила				Логічні чи семантичні
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Код ідентифікатора	id	--	int	ПК	-	так	ІНД	--
Назва курсів	nameCourses	--	string	-	-	так	-	--
Опис курсів	descCourses		string			так		

Опис масиву «MaterialsForCourses»Найменування масиву – **MaterialsForCourses**

Ідентифікатор масиву – ID

Найменування носія інформації – твердий магнітний диск

Максимальний об'єм масиву – 10000000 записів.

Довжина запису – 61 символ (або байт).

Метод організації – Послідовний.

Ключі упорядкування – Код ідентифікатора масиву, FK Код ідентифікатора курсів.

Найменування	Ідентифікатор у програмі	Умове позначення	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Код ідентифікатора	id	--	int	PK	-	так	ІНД	
Код ідентифікатора курсів	ID_courses	--	int	FK	-	так	ІНД	courses
Назва матеріалів	nameOfMaterials	--	string		-	так		
Опис матеріалів	descMaterials		string			так		
Файл	File		file			ні		

Опис масиву «TaskForCourses»Найменування масиву – **TaskForCourses**

Ідентифікатор масиву – ID

Найменування носія інформації – твердий магнітний диск

Максимальний об'єм масиву – 10000000 записів.

Довжина запису – 61 символ (або байт).

Метод організації – Послідовний.

Ключі упорядкування – Код ідентифікатора масиву, FK Код ідентифікатора курсів.

Найменування	Ідентифікатор у програмі	Умове позначення	Формат	Бізнес-правила				Логічні чи семантичні зв'язки
				Первинний ключ	Умова на значення	Обов'язкове поле	Ідексне поле	
Код ідентифікатора	id	--	int	PK	-	так	ІНД	
Код ідентифікатора курсів	ID_courses	--	int	FK	-	так	ІНД	courses
Питання	question	--	string		-	так		
Варіанти відповідей	optAnser		string			так		
Відповіді	answer		string			так		

Опис масиву «PassedCourses»Найменування масиву – **PassedCourses**

Ідентифікатор масиву – ID

Найменування носія інформації – твердий магнітний диск

Максимальний об'єм масиву – 10000000 записів.

Довжина запису – 61 символ (або байт).

Метод організації – Послідовний.

Ключі упорядкування – Код ідентифікатора масиву, FK Код ідентифікатора курсів, FK Код ідентифікатора користувачів.

Найменування	Ідентифікатор у програмі	Умове позначення	Формат	Бізнес-правила				Логічні чи семантичні
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Код ідентифікатора	id	--	int	PK	-	так	ІНД	
Код ідентифікатора курсів	ID_courses	--	int	FK	-	так	ІНД	courses
Код ідентифікатора користувачів	ID_users	--	string		-	так		users
Дата здачі	dateOfPass		date			так		
оцінка	Mark		string			так		

Опис масиву «MarksOfUsers»Найменування масиву – **MarksOfUsers**

Ідентифікатор масиву – ID

Найменування носія інформації – твердий магнітний диск

Максимальний об'єм масиву – 10000000 записів.

Довжина запису – 61 символ (або байт).

Метод організації – Послідовний.

Ключі упорядкування – Код ідентифікатора масиву, FK Код ідентифікатора курсів, FK Код ідентифікатора користувачів.

Найменування	Ідентифікатор у програмі	Умовне позначення	Формат	Бізнес-правила				Логічні чи семантичні
				Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле	
Код ідентифікатора	id	--	int	PK	-	так	ІНД	
Код ідентифікатора курсів	ID_courses	--	int	FK	-	так	ІНД	courses
Код ідентифікатора користувачів	ID_users	--	string		-	так		users
Дата оцінки	dateOfMark		date			так		
оцінка	Mark		string			так		

3.2. ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ДАНИХ

Відповідно до досліджених методів в розділі 2.2, для аналізу і оцінки ефективності алгоритмів схожості між користувачами та курсами, можна використати косинусну схожість. Ця міра схожості базується на куті між векторами, що представляють користувачів та курси у просторі оцінок. Чим більше кут між векторами, тим менша схожість між ними.

3.1 Приклад програмного коду реалізації. Джерело автора.

```
# Определение интересов пользователя
user_interests = interests.loc[user_id]

# Нахождение пользователей с похожими интересами
similar_users = interests[interests.eq(user_interests).all(1)].index
similar_users = similar_users.drop(user_id)

# Нахождение вероятности того, что пользователь выберет каждый курс
p_course_given_user = {}
for course in all_courses:
    ratings_given_course = ratings[course].loc[similar_users]
    p_course_given_user[course] = np.prod(
        ratings_given_course.apply(lambda x: (x + 1) / (x.max() + len(courses)),
axis='index'))

# Отсортировать курсы по убыванию активности и колебанию
recommended_courses = sorted(p_course_given_user, key=p_course_given_user.get,
reverse=True)[:n_recommendations]
```

У цьому коді використовується формула для знаходження ймовірності того, що користувач вибере кожен курс, виходячи з рейтингів курсів, які були оцінені користувачами зі схожими інтересами. Формула виглядає так:

```
p_course_given_user[course] = np.prod(
    ratings_given_course.apply(lambda x: (x + 1) / (x.max() + len(courses)),
axis='index'))
```

Тут "ratings_given_course" є рейтингами курсу, отриманими від користувачів зі схожими інтересами, "x.max()" - максимальне значення рейтингу, а "len(courses)" - загальна кількість курсів. В результаті "p_course_given_user[course]" - це ймовірність того, що користувач вибере певний курс, ґрунтуючись на рейтингах курсів, які були оцінені користувачами зі схожими інтересами. Функція "np.prod()" використовується для перемноження значень ймовірності кожного курсу.

На рисунку 3.3 зображено приклад роботи програмного коду

```
127.0.0.1 - - [12/May/2023 17:01:38] "GET /static/pngs/pngtree-mental-health-problems-flat-illustration-female-psychologist-png-image_6855449.png HTTP/1.1" 304 -  
127.0.0.1 - - [12/May/2023 17:01:41] "POST /recommend HTTP/1.1" 200 -  
['Python', 'Java']
```

Рис. 3.3 Результат работы программного коду.

3.3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Клієнт-серверна архітектура та інструменти розробки, що були використані у проекті, дійсно дозволили створити надійний та функціональний додаток для дистанційного навчання. Нижче наведено більш детальний опис інструментів розробки, що були використані у проекті:

Python: Python - це високорівнева, інтерпретована мова програмування, яка була використана для розробки серверної частини додатку. Python має велику кількість корисних бібліотек, що спрощують розробку, наприклад, Flask.

Flask: Flask - це мікрофреймворк для розробки веб-додатків з використанням мови Python. Він дозволяє легко налаштувати маршрутизацію, шаблонізацію та інші функції. Flask був використаний для розробки серверної частини додатку.

HTML, CSS та JavaScript: HTML, CSS та JavaScript - це основні мови для розробки клієнтської частини веб-додатків. HTML використовується для створення структури веб-сторінок, CSS - для стилізації цих сторінок, а JavaScript - для динамічного взаємодії з користувачем. У проекті було використано ці три мови для розробки інтерфейсу користувача.

SQLite та SQLite3: SQLite - це легкий, вбудований реляційна база даних, яка була використана для зберігання даних у проекті. SQLite3 - це Python-бібліотека для роботи з SQLite у проекті на мові Python.

Використання цих інструментів розробки, включаючи клієнт-серверну архітектуру, дозволило ефективно розробити додаток для дистанційного навчання з необхідними функціями, такими як реєстрація користувачів, створення курсів та лекцій, додавання матеріалів для навчання, проходження тестів та отримання сертифікатів про успішне завершення курсів. Клієнт-серверна архітектура дозволила забезпечити стабільну та безперебійну роботу додатку, а використання Python та Flask зробило розробку серверної частини швидкою та зручною. Використання HTML, CSS та JavaScript дозволило розробити привабливий та інтуїтивно зрозумілий інтерфейс користувача. А використання SQLite та SQLite3 для зберігання даних забезпечило надійність та ефективність роботи додатку.

3.3.1 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Архітектура програмного забезпечення (АПЗ) інформаційної системи дистанційного навчання (ІСДН) описує структуру та організацію програмного забезпечення, яке використовується для забезпечення процесу дистанційного навчання.

Основні рівні архітектури програмного забезпечення ІСДН:

Рівень користувача (Presentation layer) - цей рівень відповідає за інтерфейс користувача та взаємодію з ним. Це можуть бути веб-сторінки, мобільні додатки, десктопні застосунки, які надають користувачеві можливість зареєструватися в системі, створювати акаунти, відслідковувати прогрес навчання, брати участь в онлайн-заняттях тощо.

Приклади програмного забезпечення для цього рівня: Moodle, Blackboard, Google Classroom.

Рівень додатку (Application layer) - цей рівень відповідає за логіку додатку та його функціональність. Це можуть бути різні компоненти, модулі та сервіси, які забезпечують роботу системи дистанційного навчання. Наприклад, система управління курсами, система відеоконференцій, система авторизації та аутентифікації користувачів, система сповіщень тощо.

Рівень даних (Data layer) - цей рівень містить базу даних, яка забезпечує збереження та доступ до даних, необхідних для роботи системи дистанційного навчання. Цей рівень може містити інформацію про користувачів, курси, завдання, результати тестувань та інші дані.

Всі ці рівні можуть бути розподілені на різних фізичних серверах або можуть бути зібрані на одному сервері. Для забезпечення взаємодії між різними рівнями використовуються різні протоколи, такі як HTTP, HTTPS, TCP/IP.

3.3.2 КЕРІВНИЦТВО КОРИСТУВАЧА

На рисунку 3.4 зображено вікно авторизації

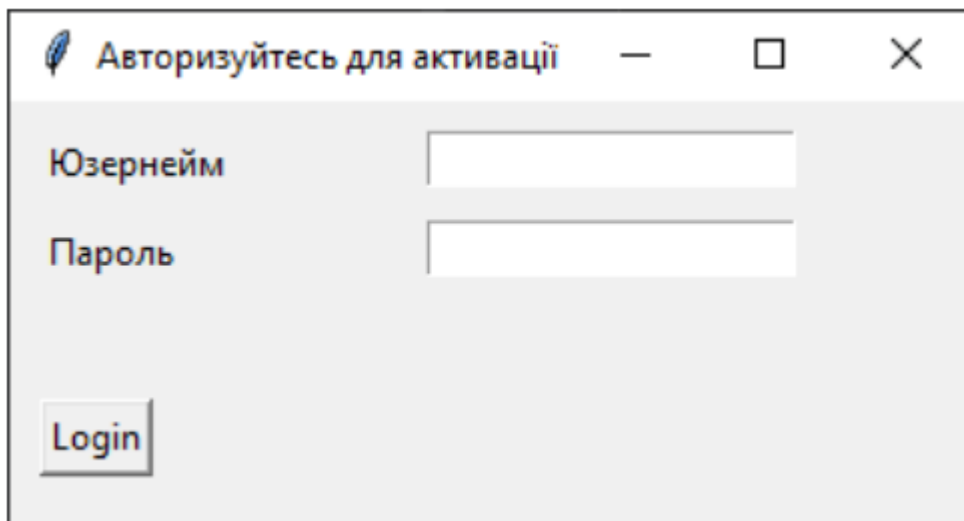


Рисунок 3.4 Вікно авторизації. Джерело автора.

У разі неправильного введення логіна чи пароля, система повідомить про помилку та не надасть доступу. Однак буде доступна ще одна спроба ввести коректні дані для логіну або пароля, кількість таких спроб необмежена.

На рисунку 3.5 зображено вікно помилкової авторизації

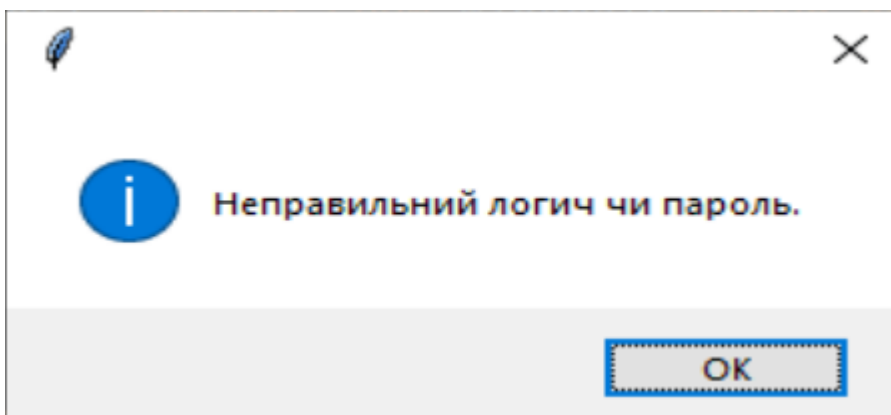


Рисунок 3.5 Помилка вводу логіну чи пароля. Джерело автора.

Якщо поля авторизації не будуть заповнені правильно, вікно помилки буде з'являтися постійно. Поки користувач не введе коректні дані, помилка буде повторюватися.

На рисунку 3.6 зображено вікно помилки

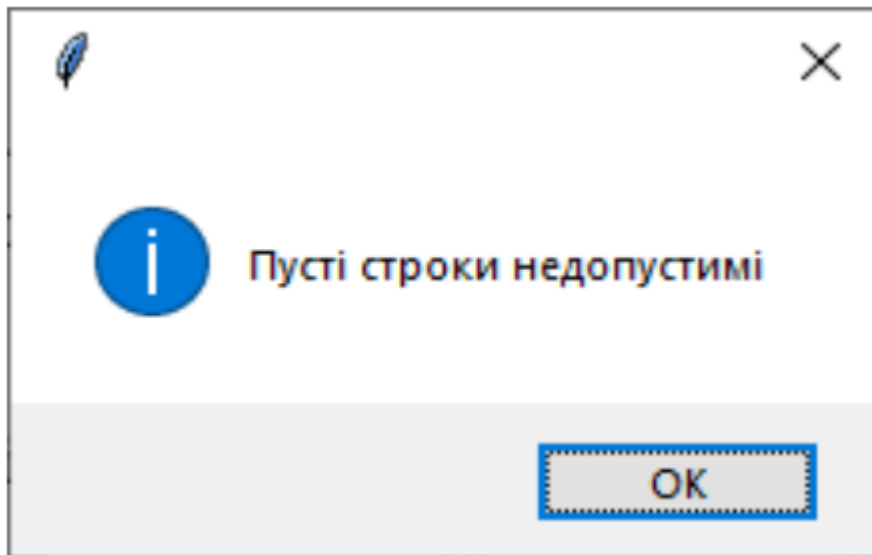


Рисунок 3.6 Неправильний ввід даних. Джерело автора.

Якщо користувач не заповнює поле, де має бути вказаний логін або пароль, система відображатиме повідомлення про помилку, що неможливо залишати це поле порожнім. Ця помилка буде відображатися кожен раз, коли користувач спробує увійти в систему, не заповнивши відповідне поле.

На рисунку 3.7 зображено вікно успішної авторизації

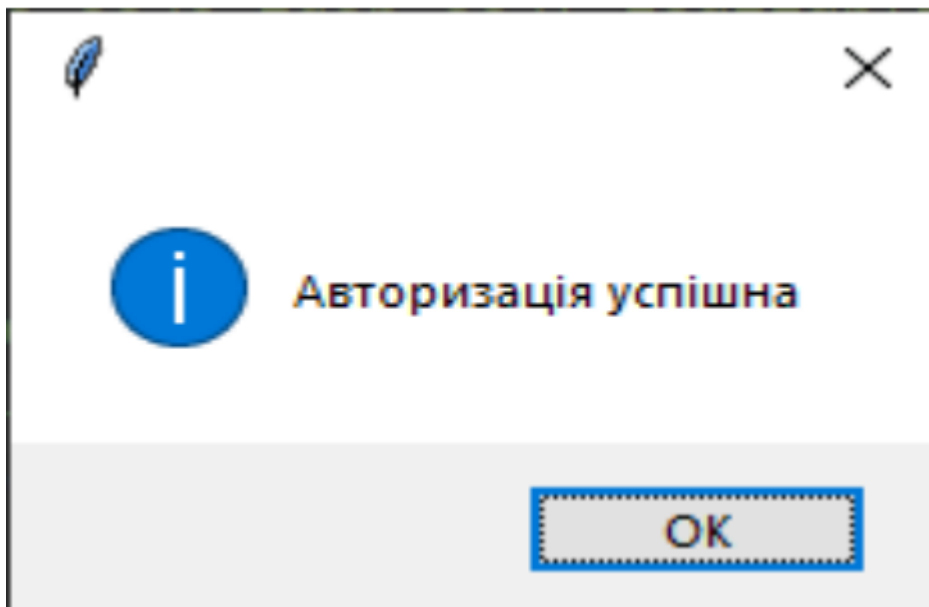


Рисунок 3.7 Авторизація успішна, запуск програми. Джерело автора.

Коли коректні дані будуть введені повністю, система підтвердить правильну авторизацію і відобразить наступне повідомлення. Застосунок розпочне відкриватися, що дозволить отримати доступ до системи.

Після того, як користувач увійшов до застосунку, він буде перенаправлений на першу форму графічного інтерфейсу. На цій сторінці буде доступний особистий кабінет користувача, де він зможе переглянути свої курси або додати нові.

На рисунку 3.8 зображено особистий кабінет користувача

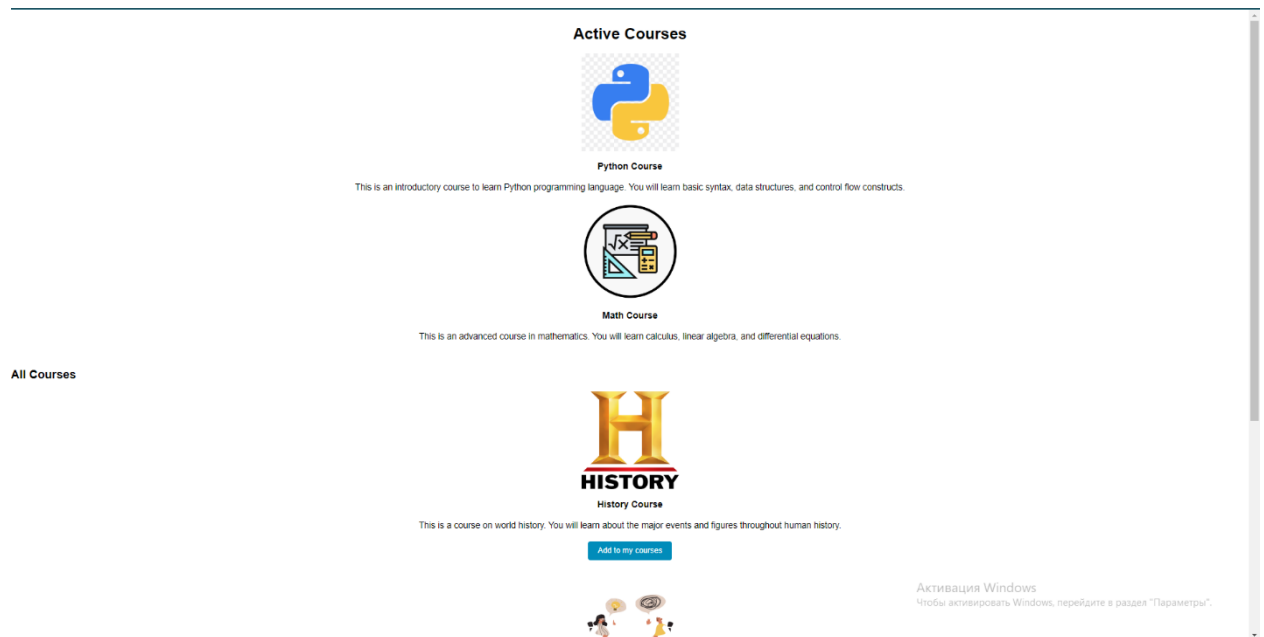


Рис. 3.8 Особистий кабінет користувача з курсами. Джерело автора.

На цьому етапі користувач може додати нові курси, які є наведені в списку нижче, або приступити до виконання своїх курсів.

На рисунку 3.9 буде зображено вікно, яке буде показуватись адміністратору при вдалому вході в систему. Також в даному вікні буде кнопка яка при натисканні буде обробляти запит до бази даних, та брати звідти значення оцінок та імен студентів які.

Add Course

Name:

Description:

Add Course

Student Name	Grade
Сидоров	A
Петров	F
Иванов	A
Сидоров	A
Петров	B

Load Data

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рис. 3.9 Вікно адміністратора. Джерело автора.

Також при натисканні на кнопку Add Course, курс буде додано та він буде відображатись вгорі, над написом Add Course.

ID	Название	Описание
1	Математика	Курс з математики для початківців
2	англійська мова	Курс англійської мови для середнього рівня
3	Программирование	Основи програмування на Python
4	a	a
5	s	s
6	sdf	sdf
7	as	asd

Add Course

Name:

Description:

Add Course

Student Name	Grade
--------------	-------

Load Data

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рис. 3.10 Відображення всіх нових доданих курсів. Джерело автора.

Рекомендації використовуючи алгоритм колаборативного фільтрування можуть бути корисні для користувачів, які хочуть отримати рекомендації з питань, що їх цікавлять. Наприклад, якщо користувач має певні інтереси, він може натиснути на кнопку "Отримати рекомендації" на сторінці курсів і отримати список курсів, які відповідають його інтересам.

Також рекомендації можуть допомогти користувачам розширити свій кругозір, надаючи їм курси, які вони не могли б знайти вручну. Користувачі можуть використовувати цю функцію, щоб знайти нові та цікаві курси, які відповідають їхнім захопленням та цілям.

Рекомендації можуть бути особливо корисними для нових користувачів, які ще не знайомі з усіма курсами, доступними на платформі. Це може допомогти їм швидко знайти курси, які відповідають їхнім інтересам, та розпочати навчання.

На рисунку 3.11 зображено приклад роботи рекомендацій.

Python

Java

Get Recommendations

Recommended courses

Active Courses

 Python Course

Python Course

This is an introductory course to learn Python programming language. You will learn basic syntax, data structures, and control flow constructs.

 Math Course

Math Course

This is an advanced course in mathematics. You will learn calculus, linear algebra, and differential equations.

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

Рис. 3.11 Приклад роботи рекомендацій. Джерело автора.

Також для користувача адміністратор присутня форма для заповнення оцінок, де потрібно ввести ім'я студента, та його оцінку

Зображено на малюнку 3.12 форма для заповнення оцінок.

Форма для вводу оцінок

ім'я студента:

Оцінка:

Отправить

Рис. 3.12 Форма для заповнення оцінок. Джерело автора.

Також присутня гістограма відображення оцінок, показується загальна успішність всіх студентів

На рисунку 3.13 зображено гістограму всіх поточних оцінок

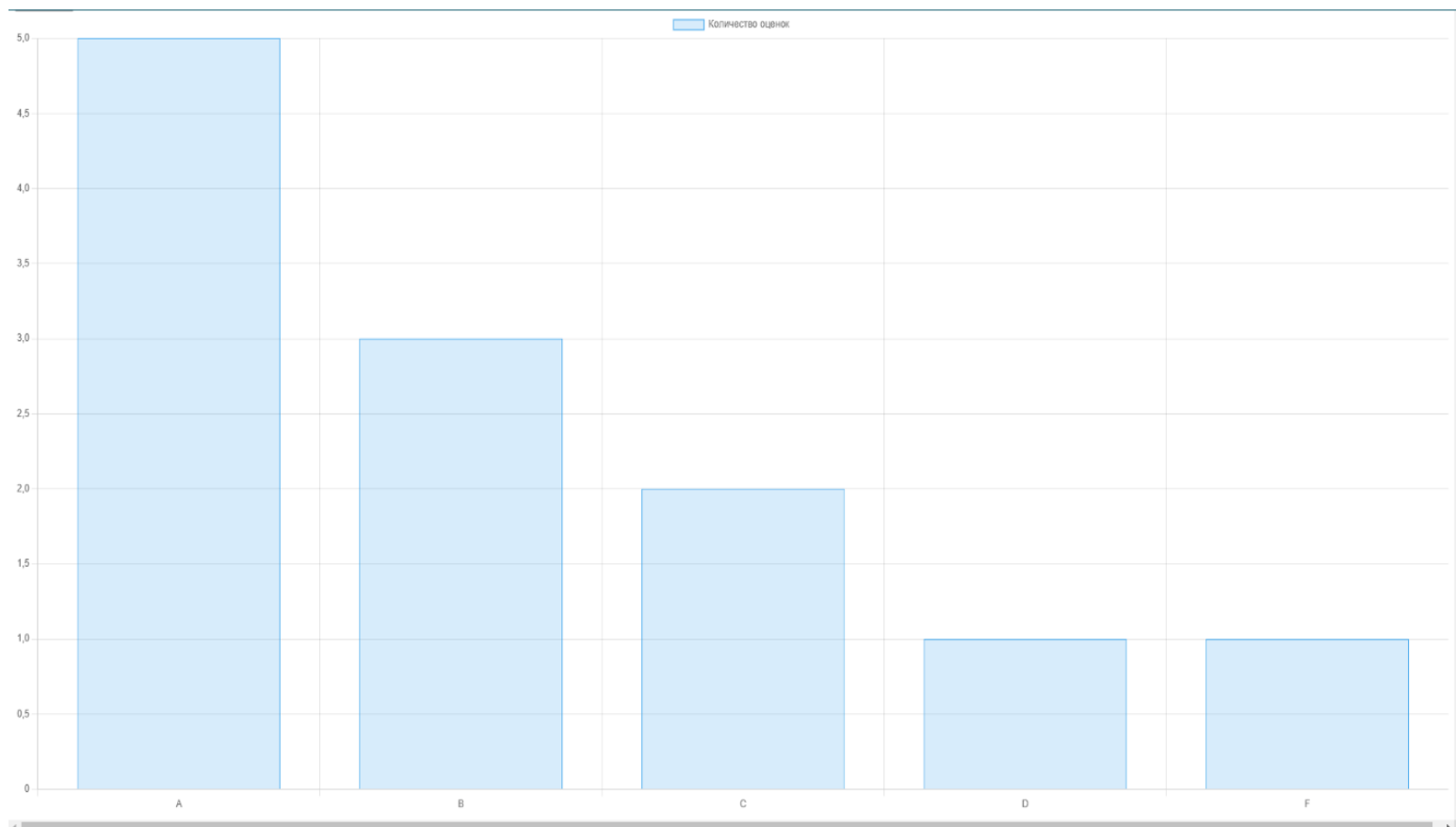


Рис. 3.13 Гістограма всіх поточних оцінок. Джерело автора.

Для користувача присутня таблиця з заборгованостями, де студент може подивитись на свої борги, а також здати завдання обравши з якого предмету йому необхідно його здати.

На рисунку 3.14 зображено заборгованості та форму для відправки завдань

Рис. 3.14 Заборгованості та форма для здачі. Джерело автора.

Заборгованості:

Курс	Номер завдання	Статус
Курс програмування	Завдання 1	Не виконано
Курс програмування	Завдання 2	Виконано частково
Курс математики	Завдання 1	Не виконано
Курс математики	Завдання 2	Не виконано

Submit Your Work

Choose an assignment: Programming Assignment 1 ▾

Upload your file: Виберіть файл Файл не вибран

Submit

3.4 ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

Інформаційна система для дистанційного навчання, яка базується на використанні індивідуального віддаленого серверу, може бути ефективним інструментом для навчання на відстані. Така система дозволяє здобувачам та викладачам мати доступ до матеріалів та інструментів навчання з будь-якої точки світу, де є доступ до Інтернету.

Ядро системи, база даних та необхідні файли зберігаються на індивідуальному віддаленому сервері, що забезпечує швидкий та безпечний доступ до навчальних ресурсів. Крім того, це зменшує навантаження на комп'ютери користувачів та викладачів, що дозволяє їм працювати зі складними додатками та важкими файлами, не стикаючись з обмеженнями апаратної потужності своїх комп'ютерів.

Застосування сайту з доступом до мережі Інтернет є зручним та простим способом доступу до системи дистанційного навчання. Це дозволяє користувачам отримувати доступ до навчальних матеріалів з будь-якого комп'ютера з доступом до Інтернету. Крім того, використання сайту дозволяє забезпечити безпеку та конфіденційність навчальних матеріалів та персональних даних користувачів.

На рисунку 3.15 зображено Загальна схема автоматизації.

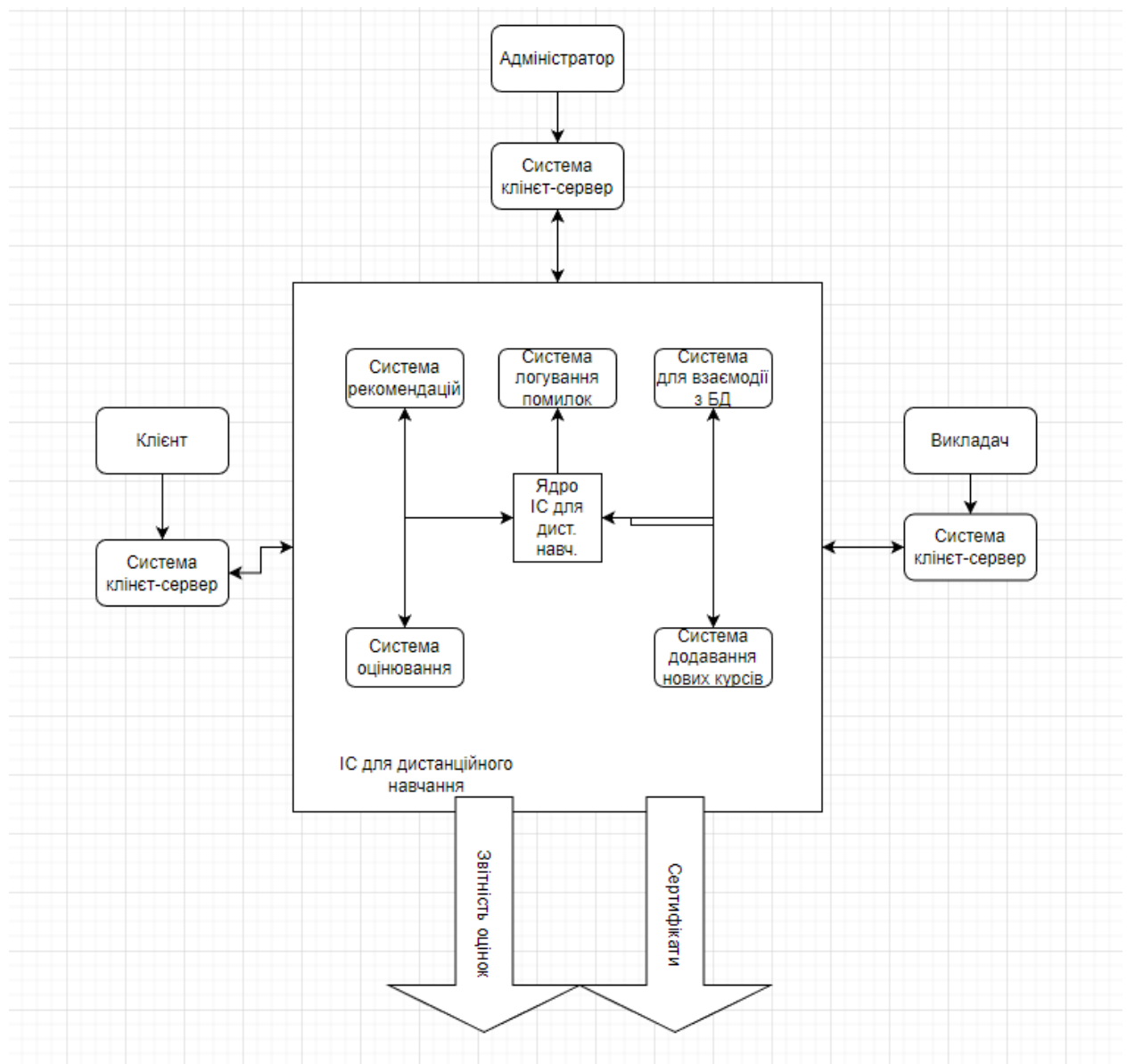


Рис 3.15 Загальна схема автоматизації. Джерело автора.

Для встановлення даної інформаційної системи необхідні такі характеристики сервера

Таблиця 3.8 Технічні характеристики сервера. Джерело автора.

Процесор	Intel Xeon E-2124
ОЗУ	8 ГБ оперативної пам'яті 2666 MT с DDR4 ECC UDIMM
Керування сервера	iDRAC8 Basic
Пам'ять	1 ТБ SSD
БЖ	350 Вт
Мережевий контролер	NIC – 2 порта 1GbE
Порти	USB 2.0 и IDRAC micro USB 2.0

3.5 РЕЗУЛЬТАТИ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Результатом реалізації даної системи є розроблена інформаційна система для дистанційного навчання «Ignatyev». Розроблена ІС виконує всі визначені в процесі дослідження функціональні вимоги.

Перерахувати вимоги:

Розроблені методи аналізу для рекомендованих курсів дають змогу користувачу отримати рекомендацію які ще курси йому можна освоїти, виходячи з даних отриманих іншими користувачами та попитом.

Розроблений проект системи включив:

1. Проведено системний аналіз предметної області та вибрано прототип ІС для дистанційного навчання.
2. Запропоновано архітектуру ІС дистанційного навчання як власну розробку за прізвищем автора «Ignatyev».
3. Розроблена БД та проведено демонстрація роботи її з реальними даними.
4. Проведено тестування ІС для дистанційного навчання «Ignatyev» в розділі 3.2.2.

Додатковим етапом розвитку ІС може бути перенесення функціоналу ІС на інші платформи.

Описані фактори будуть враховані під час модернізації ІС та розробці наступних версій.

Дана розроблена система має свої недоліки:

1. Ця система є прототипом, та має не дуже приємний оку вигляд.
2. В цю систему потрібно імпортувати ще декілька програмних модулів для ідеального відпрацювання системи.

Описані фактори будуть враховані під час модернізації ІС та розробці наступних версій.

ВИСНОВКИ

Метою розроблення інформаційної системи для дистанційного навчання є розробка та реалізація інформаційної системи дистанційного навчання авторського проекту «Ignatyev», яка забезпечує ефективну організацію процесу навчання та надання якісної освіти з використанням сучасних технологій.

Розроблений проєкт системи дозволяє зручно та ефективно навчатись та викладати начальну продукцію(навчальні дисципліни) в онлайн форматі.

У проєкті було реалізовано такі функції як:

- авторизація для користувачів та викладачів;
- роботу в авторській системі з можливістю створення нових курсів та додавання завдань до них, перегляду домашніх завдань та їх відправлення, оцінювання завдань та ліквідацію заборгованостей;
- візуалізація виконання, діалогу викладачів та здобувачів та оцінювання результатів навчання.
- надання сертифікатів навчання за рейтингом оцінювання результатів.

Також було використано HTML та Flask для розробки користувацького інтерфейсу та взаємодії з базою даних SQLite.

ПЕРЕЛІК ВЛАСНИХ ПУБЛІКАЦІЙ

Ігнат'єв І.С. Науковий керівник, д.е.н., професор Устенко Станіслав Веніамінович. «Авторська інформаційна система для дистанційного навчання» – III міжнародна студентська наукова конференція «Тренди та перспективи розвитку мультидисциплінарних досліджень» – МНЛ, 2023 р., Рівне. – Довідка про публікацію на 12.05.2023

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ секція 14.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Інформаційні управляючі системи та технології : міжнародна монографія / За заг. ред. докт. екон. наук, професора Устенко С. В. Київ : КНЕУ, 2019. 419 с [Електронне видання] Bialostockie Wydawnietwo Naukowe e-BWN — видавнича платформа www.e-bwn.com - 2020. <https://is.gd/sO7Fgk>
2. "Distance and E-Learning in Transition" by Olaf Zawacki-Richter and Terry Anderson
3. "Blended Learning: A Guide for the New Normal" by Terry Anderson, Jon Dron, and Chris Downey
4. "Designing Effective Online Instruction: A Handbook for Web-Based Courses" by Franklin R. Demana, Addison-Wesley
5. "E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning" by Ruth C. Clark, Richard E. Mayer, and John Wiley & Sons
6. "Engaging Students through Social Media: Evidence-Based Practices for Use in Student Affairs" by Reynol Junco, Jossey-Bass
7. "Innovative Assessment for the 21st Century: Supporting Educational Needs" by Valerie Shute and Betsy Jane Becker
8. "Online Education: Practical, Theory-Based Advice for the Instructor" by Rena M. Palloff and Keith Pratt, Jossey-Bass

9. "Teaching and Learning at a Distance: Foundations of Distance Education" by Michael Simonson, Sharon Smaldino, and Susan M. Zvacek, Information Age Publishing
10. "The Design of Learning Experience: Creating the Future of Educational Technology" by Brad Hokanson, EdTech Books
11. "The Theory and Practice of Online Learning" by Terry Anderson and Fathi Elloumi, Athabasca University Press
12. Reinmann-Rothmeier, G., Mandl, H., & Schneider, M. (2010). Distance education revisited: A review of the scientific literature. *Educational Research Review*, 5(1), 116-144.
13. Creelman, A., & Sherry, L. (2011). Assessment and accreditation issues in online higher education. *European Journal of Open, Distance and e-Learning*, 14(2), 1-9.
14. Gasevic, D., Kovanovic, V., Joksimovic, S., & Siemens, G. (2014). Where is research on massive open online courses headed? A data analysis of the MOOC research initiative. *The International Review of Research in Open and Distributed Learning*, 15(5), 134-176.
15. Hrastinski, S., Keller, C., & Carlsson, S. A. (2010). Design exemplars for synchronous e-learning: A design theory approach. *Computers & Education*, 55(3), 652-662.
16. Jansen, D., Schuwer, R., Teixeira, A., & Aydin, C. H. (2015). Comparing MOOC adoption strategies in Europe: Results from the HOME project survey. *International Review of Research in Open and Distributed Learning*, 16(6), 116-136.
17. BlackBoard learning - режим доступа - <https://www.blackboard.com/>
18. Moodle - режим доступа - <https://moodle.org/?lang=ru>
19. Canvas – режим доступа - <https://www.canvas.net/>

ДОДАТКИ

Додаток А

Програмний код реалізований на мові Python, фреймворк Flask.

```
import sqlite3, g
from time import sleep
import pandas as pd
import numpy as np
import random
import json

from flask import Flask, render_template, request, redirect, url_for, session,
jsonify, g

app = Flask(__name__)
app.secret_key = 'secret'
app.config['DATABASE'] = r'C:\Users\попытка
#5\PycharmProjects\flaskProject\Courses.db'

# Создание DataFrame с оценками пользователей для курсов
all_courses = ['Python', 'Java', 'SQL', 'Machine Learning']
users = ['user1', 'user2', 'user3', 'user4']
ratings = pd.DataFrame(np.random.randint(0, 5, size=(4, 4)), columns=all_courses,
index=users)

# Создание DataFrame с интересами пользователей к курсам
interests = pd.DataFrame(np.random.randint(0, 2, size=(4, 4)),
columns=all_courses, index=users)

registered = {
    "user1": "password1",
    "user2": "password2",
    "admin": "adminpassword"
}

def recommend_courses(user_id, ratings, interests, n_recommendations):
    # Определение интересов пользователя
    user_interests = interests.loc[user_id]

    # Нахождение пользователей с похожими интересами
    similar_users = interests[interests.eq(user_interests).all(1)].index
    similar_users = similar_users.drop(user_id)

    # Нахождение вероятности того, что пользователь выберет каждый курс
    p_course_given_user = {}
    for course in all_courses:
        ratings_given_course = ratings[course].loc[similar_users]
        p_course_given_user[course] = np.prod(
            ratings_given_course.apply(lambda x: (x + 1) / (x.max() +
len(all_courses)), axis='index'))

    # Отсортировать курсы по убыванию активности и колебанию
    recommended_courses = sorted(p_course_given_user, key=p_course_given_user.get,
reverse=True)[:n_recommendations]

    return recommended_courses

def get_db():
```

```

    if 'db' not in g:
        g.db = sqlite3.connect(app.config['DATABASE'])
    return g.db

def close_connection(exception):
    db = g.pop('db', None)
    if db is not None:
        db.close()

@app.teardown_appcontext
def close_connection(exception):
    dba = getattr(g, '_database', None)
    if dba is not None:
        dba.close()

# Роут для отображения формы логина
@app.route("/")
def login():
    return render_template("login.html")

# Роут для обработки запроса на авторизацию
@app.route("/auth", methods=["POST"])
def auth():
    username = request.form.get("username")
    password = request.form.get("password")

    # Проверяем, является ли пользователь администратором
    if username == "admin" and password == registered["admin"]:
        return redirect("/add_course")

    # Проверяем, является ли пользователь обычным пользователем
    if username in registered and password == registered[username]:
        return redirect("/home")

    # Если логин и пароль неверны, возвращаем сообщение об ошибке
    return render_template("login.html", error=True)

# Роут для страницы курсов
@app.route("/courses")
def courses():
    return render_template("courses.html")

@app.route("/home")
def home_link():
    return render_template("home.html")

@app.route('/get_grades')
def get_grades():
    conn = get_db()
    c = conn.cursor()
    c.execute("SELECT Users.name, grades.Grade FROM grades INNER JOIN Users ON grades.user = Users.id")
    rows = c.fetchall()
    conn.commit()
    conn.close()
    return jsonify(rows)

```

```

@app.route('/add_course', methods=['GET', 'POST'])
def add_course():
    if request.method == 'POST':
        name = request.form['name']
        description = request.form['description']
        conn = get_db()
        c = conn.cursor()
        sleep(3)
        c.execute("INSERT INTO lectures (courseName, description) VALUES (?, ?)",
(name, description))
        sleep(3)
        conn.commit()
        conn.close()
        return redirect(url_for('add_course'))
    else:
        conn = get_db()
        c = conn.cursor()
        c.execute("SELECT * FROM lectures")
        lectures = c.fetchall()
        conn.commit()
        conn.close()
        return render_template('add_cours.html', courses=lectures)

@app.route('/add_course', methods=['POST'])
def add_course_post():
    # Получаем данные из POST-запроса
    title = request.form['title']
    description = request.form['description']

    # Добавляем новый курс в базу данных
    conn = get_db()
    c = conn.cursor()
    c.execute("INSERT INTO letcures (courseName, description) VALUES (?, ?)",
(title, description))
    conn.commit()
    conn.close()

@app.route('/recommend', methods=['POST'])
def recommend():
    user_id = random.choice(list(ratings.index))
    n_recommendations = 2
    recommended_courses = recommend_courses(user_id, ratings, interests,
n_recommendations)
    return jsonify({'recommendations': recommended_courses})

@app.route('/redirect')
def home():
    # ...
    return redirect(url_for('courses'))

@app.route('/', methods=['GET', 'POST'])
def submit():
    if request.method == 'POST':
        name = request.form['name']
        score = request.form['score']
        # добавляем введенные данные в базу данных
        conn = get_db()
        c = conn.cursor()
        c.execute('INSERT INTO grades (user, Grade) VALUES (?, ?)', (name, score))

```

```

        conn.commit()
        conn.close()
        return 'Данные успешно отправлены в базу данных!'
    return render_template('add_cours.html')

@app.route('/')
def index():
    conn = get_db()
    c = conn.cursor()
    c.execute('SELECT COUNT(*) FROM grades WHERE grade="A"')
    countA = c.fetchone()[0]
    c.execute('SELECT COUNT(*) FROM grades WHERE grade="B"')
    countB = c.fetchone()[0]
    c.execute('SELECT COUNT(*) FROM grades WHERE grade="C"')
    countC = c.fetchone()[0]
    c.execute('SELECT COUNT(*) FROM grades WHERE grade="D"')
    countD = c.fetchone()[0]
    c.execute('SELECT COUNT(*) FROM grades WHERE grade="F"')
    countF = c.fetchone()[0]
    conn.close()
    return render_template('add_cours.html', counts=[countA, countB, countC,
countD, countF])

if __name__ == "__main__":
    app.run(debug=True)

```

Додаток Б

Код на HTML, CSS, JS для відображення інтерфейсу адміністратора

```

<!DOCTYPE html>
<html>
<head>
    <title>Add Course</title>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static',
filename='styles.css') }}">
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
</head>
<body>
<table>
    <thead>
        <tr>
            <th>ID</th>
            <th>Название</th>
            <th>Описание</th>
        </tr>
    </thead>
    <tbody>
        {% for course in courses %}
        <tr>
            <td>{{ course[0] }}</td>
            <td>{{ course[1] }}</td>
            <td>{{ course[2] }}</td>
        </tr>
        {% endfor %}
    </tbody>
</table>
<h1>Add Course</h1>
<form>
    <label for="name">Name:</label><br>
    <input type="text" id="name" name="name"><br>
    <label for="description">Description:</label><br>

```

```

        <textarea id="description" name="description"></textarea><br>
        <input type="submit" value="Add Course">
    </form>
<h1>Форма для ввода оценок</h1>
<form action="{ { url_for('submit') } }" method="post">
    <label for="name">имя студента:</label><br>
    <input type="text" id="name" name="name"><br>
    <label for="score">Оценка:</label><br>
    <input type="text" id="score" name="score"><br><br>
    <input type="submit" value="Отправить">
</form>
<table id="gradesTable">
    <thead>
        <tr>
            <th>Student Name</th>
            <th>Grade</th>
        </tr>
    </thead>
    <tbody>
    </tbody>
</table>
<button onclick="loadData()">Load Data</button>
<canvas id="myChart"></canvas>
<script>
    var ctx = document.getElementById('myChart').getContext('2d');
    var chartData = {
        labels: ['A', 'B', 'C', 'D', 'F'],
        datasets: [{
            label: 'Количество оценок',
            backgroundColor: 'rgba(54, 162, 235, 0.2)',
            borderColor: 'rgba(54, 162, 235, 1)',
            borderWidth: 1,
            data: [5, 3, 2, 1, 1] // здесь нужно заменить нули на количество оценок
из базы данных
        }]
    };
    var chartOptions = {
        scales: {
            yAxes: [{
                ticks: {
                    beginAtZero: true
                }
            }]
        }
    };
    var myChart = new Chart(ctx, {
        type: 'bar',
        data: chartData,
        options: chartOptions
    });
</script>

<script>
    function loadData() {
        $.get("/get_grades", function(data) {
            $("#gradesTable tbody").empty();
            data.forEach(function(row) {
                var tr = $("<tr>");
                tr.append($("<td>").text(row[0]));
                tr.append($("<td>").text(row[1]));
                $("#gradesTable tbody").append(tr);
            });
        });
    }
}

```

```
</script>
</body>
</html>
```

Додаток В

Код HTML, CSS, JS для відображення інтерфейсу користувача

```
<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
    }
    h1 {
      text-align: center;
      margin-top: 30px;
    }
    status {
      width: 200px; /* задаем ширину блока статуса */
      height: 20px; /* задаем высоту блока статуса */
      background-color: #f5f5f5; /* задаем фон блока статуса */
      border-radius: 10px; /* задаем скругление углов блока статуса */
      overflow: hidden; /* скрываем все, что не помещается в блоке */
    }
    .status-bar {
      height: 100%; /* задаем высоту полоски статуса */
      background-color: #4CAF50; /* задаем цвет полоски статуса */
      border-radius: 10px; /* задаем скругление углов полоски статуса */
      transition: width 0.5s ease-in-out; /* добавляем анимацию изменения ширины
полоски статуса */
    }
    h2 {
      margin-top: 50px;
    }
    ul {
      list-style: none;
      margin: 0;
      padding: 0;
    }
    li {
      margin-bottom: 50px;
    }
    img {
      display: block;
      margin: 0 auto;
      width: 200px;
      height: 200px;
    }
    h3 {
      text-align: center;
      margin-top: 20px;
```



```

        
        <h3>Python Course</h3>
        <p>This is an introductory course to learn Python programming language.
You will learn basic syntax, data structures, and control flow constructs.</p>
        <form action="/add_course" method="post">
            <input type="hidden" name="title" value="Python Course">
            <input type="hidden" name="description" value="This is an introductory
course to learn Python programming language. You will learn basic syntax, data
structures, and control flow constructs.">

        </form>
    </div>
    <div>
        
        <h3>Math Course</h3>
        <p>This is an advanced course in mathematics. You will learn calculus,
linear algebra, and differential equations.</p>
        <form action="/add_course" method="post">
            <input type="hidden" name="title" value="Math Course">
            <input type="hidden" name="description" value="This is an advanced
course in mathematics. You will learn calculus, linear algebra, and differential
equations.">

        </form>
    </div>
    <div>
        <h3>Заборогованності:</h3>
        <table>
            <thead>
                <tr>
                    <th>Курс</th>
                    <th>Номер завдання</th>
                    <th>Статус</th>
                    <div class="status">
                        <div class="status-bar" style="width: 40%;"></div>
                    </div>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>Курс програмування</td>
                    <td>Завдання 1</td>
                    <td>Не виконано</td>
                </tr>
                <tr>
                    <td>Курс програмування</td>
                    <td>Завдання 2</td>
                    <td>Виконано частково</td>
                </tr>
                <tr>
                    <td>Курс математики</td>
                    <td>Завдання 1</td>
                    <td>Не виконано</td>
                </tr>
                <tr>
                    <td>Курс математики</td>
                    <td>Завдання 2</td>
                    <td>Не виконано</td>
                </tr>
            </tbody>
        </table>
    </div>
    <h2>Submit Your Work</h2>

```

```

<form action="/submit" method="POST" enctype="multipart/form-data">
  <label for="assignment">Choose an assignment:</label>
  <select id="assignment" name="assignment">
    <option value="Programming Assignment 1">Programming Assignment 1</option>
    <option value="Math Assignment 1">Math Assignment 1</option>
  </select>
  <br><br>
  <label for="file">Upload your file:</label>
  <input type="file" id="file" name="file">
  <br><br>
  <input type="submit" value="Submit">
</form>
<h2>All Courses</h2>
<ul>
  <li>
    
    <h3>History Course</h3>
    <p>This is a course on world history. You will learn about the major events and figures throughout human history.</p>
    <form action="/add_course" method="post">
      <input type="hidden" name="title" value="History Course">
      <input type="hidden" name="description" value="This is a course on world history. You will learn about the major events and figures throughout human history.">
      <button type="submit">Add to my courses</button>
    </form>
  </li>
  <li>
    
    <h3>Psychology Course</h3>
    <p>This is an introductory course on psychology. You will learn about the basics of human behavior and mental processes.</p>
    <form action="/add_course" method="post">
      <input type="hidden" name="title" value="Psychology Course">
      <input type="hidden" name="description" value="This is an introductory course on psychology. You will learn about the basics of human behavior and mental processes.">
      <button type="submit">Add to my courses</button>
    </form>
  </li>
  <li>
    
    <h3>Biology Course</h3>
    <p>This is a course on biological science. You will learn about the diversity of life, cellular structure, genetics, and evolution.</p>
    <form action="/add_course" method="post">
      <input type="hidden" name="title" value="Biology Course">
      <input type="hidden" name="description" value="This is a course on biological science. You will learn about the diversity of life, cellular structure, genetics, and evolution.">
      <button type="submit">Add to my courses</button>
    </form>
  </li>
</ul>
</div>
</body>
</html>

```



Ім'я користувача:
Інформаційних систем в економіці Шкуратовська Те...

ID перевірки:
1015249244

Дата перевірки:
25.05.2023 14:21:02 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
25.05.2023 14:38:39 EEST

ID користувача:
100005745

Назва документа: ІУСТ_Ігнат'єв

Кількість сторінок: 82

Кількість слів: 9344

Кількість символів: 75891

Розмір файлу: 3.89 MB

ID файлу: 10

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

9.35%
Схожість

Найбільша схожість: 3.75% з джерелом з Бібліотеки (ID файлу: 1007948190)



0%
Цитат

Вилучення цитат виключене

Вилучення списку бібліографічних посилань виключене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

13

Підозріле форматування

18 сторінок

Назва документа: ІУСТ_Ігнат'єв

ID файлу:
1014924615

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
«ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ЕКОНОМІЦІ»
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ В ЕКОНОМІЦІ
галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма
«Інформаційні управлінські системи і технології»

Форма навчання: ДЕННА

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА
на тему:
РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ
ДЛЯ ДИСТАНЦІЙНОГО НАВЧАННЯ

здобувача Ігнат'єва Іллі Сергійовича

Науковий керівник:
Доктор економічних наук, професор
Устенко С.В.

Кваліфікаційна магістерська робота
дopusнений до захисту в
Екзаменаційній комісії з атестації
здобувачів вищої освіти
в.о. завідувача кафедри к.е.н., доцент
Тішкова Б.О.

Київ 2023