

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА
Навчально-науковий інститут енергетичної, інформаційної
та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

Пояснювальна записка
до кваліфікаційної роботи бакалавра

на тему: «Розробка інформаційної системи управління навчальними курсами з
використанням програмного фреймворку Django»

Виконав: студент 4 курсу, групи КН 2022-1
спеціальності

122 Комп'ютерні науки
(шифр і назва спеціальності)

Анна МУКОВОЗ
(ім'я та прізвище)

Керівник: д.ф-м.н., проф. Наталія СІЗОВА
(ім'я та прізвище)

Рецензент:
к.т.н., доц. Пахомов Ю.В.
(прізвище та ініціали)

м. Харків – 2026 рік

Харківський національний університет міського господарства імені О. М. Бекетова

(повне найменування закладу вищої освіти)

Навчально-науковий Інститут енергетичної, інформаційної

та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Спеціальність F3(122) Комп'ютерні науки

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КНтаІТ



Марина НОВОЖИЛОВА

«22» червня 2026 року

З А В Д А Н Н Я

НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Муковоз Анна Євгеніївна

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка інформаційної системи управління навчальними курсами з використанням програмного фреймворку Django»

керівник роботи д.ф-м.н., проф. Сізова Н.Д.

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

затверджені наказом закладу вищої освіти від 22» травня 2026 р. № 440-03

2. Термін подання студентом роботи 22.06.2026

3. Вихідні дані до роботи Рекомендації для розробки інформаційної системи управління навчальними курсами з використанням програмного фреймворку Django.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати предметну область та сучасні системи дистанційного навчання; виконати порівняльний аналіз LMS-платформ; обґрунтувати вибір програмних засобів розробки; визначити функціональні та нефункціональні вимоги до системи; розробити архітектуру програмного забезпечення та структуру бази даних; реалізувати механізми реєстрації, автентифікації та керування ролями користувачів; створити функціонал управління курсами, навчальними матеріалами та тестуванням; реалізувати засоби комунікації між учасниками освітнього процесу; розробити користувацький інтерфейс системи; провести тестування та розгортання програмного продукту.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
16 слайдів

6. Консультанти розділів роботи

Розділ	Ім'я та Прізвище, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	 Наталія СІЗОВА	11.05.2026	10.05.2026
Розділ II	 Наталія СІЗОВА	17.05.2026	15.05.2026
Розділ III	 Наталія СІЗОВА	21.05.2026	30.05.2026
Розділ IV	 Вікторія МАЛИШЕВА	27.05.2026	15.06.2026

7. Дата видачі завдання 10.05.2026

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми дипломної роботи	03.05.2026	Викон.
2	Затвердження тем, наукових керівників, завдань та календарного плану підготовки дипломної роботи	05.05.2026	Викон.
3	Написання I розділу	10.05.2026	Викон.
4	Написання II розділу	15.05.2026	Викон.
5	Написання III розділу	20.05.2026	Викон.
6	Написання IV розділу	30.05.2026	Викон.
7	Подання дипломної роботи керівнику	05.06.2026	Викон.
8	Робота по усуненню зауважень керівника, уточнення і доповнення практичного матеріалу, оформлення додатків до роботи	10.06.2026	Викон.
9	Подання доопрацьованого варіанту роботи керівнику	15.06.2026	Викон.
10	Захист матеріалів дипломної роботи на засіданні кафедри	18.06.2026	Викон.
11	Офіційний захист матеріалів дипломної роботи на засіданні екзаменаційної комісії	22.06.2026	Викон.

Студент


(підпис)

Анна МУКОВОЗ

(прізвище та ініціали)

Керівник роботи


(підпис)

Наталія СІЗОВА

(прізвище та ініціали)

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка кваліфікаційної роботи бакалавра здобувача вищої освіти групи КН 2022-1 спеціальності 122 «Комп'ютерні науки» Муковоз Анни Євгеніївни за темою «Розробка інформаційної системи управління навчальними курсами з використанням програмного фреймворку Django» складається з 4 розділів, містить 76 сторінок тексту, 10 рисунків, 3 таблиці, 29 джерел.

Дана робота присвячена розробці інформаційної системи управління навчальними курсами з використанням програмного фреймворку Django для організації дистанційного навчання та управління освітнім процесом.

Актуальність дослідження. Сучасний освітній процес потребує ефективних інформаційних систем, які забезпечують дистанційне навчання, централізоване зберігання навчальних матеріалів, контроль успішності студентів та комунікацію між учасниками освітнього процесу. Існуючі платформи дистанційного навчання часто є надто складними для впровадження або мають обмежені можливості адаптації до потреб конкретного навчального закладу. Тому актуальним є створення власної інформаційної системи управління навчальними курсами, яка поєднує необхідний функціонал із простотою використання.

Метою роботи є розробка веборієнтованої інформаційної системи управління навчальними курсами на базі програмного фреймворку Django, яка забезпечує створення та управління навчальними курсами, проведення тестування, оцінювання результатів навчання, комунікацію між користувачами та контроль освітнього процесу.

Завдання полягає у проведенні аналізу предметної області та існуючих систем дистанційного навчання, визначенні функціональних та нефункціональних вимог до системи, проєктуванні архітектури програмного забезпечення та структури бази даних, реалізації механізмів реєстрації та автентифікації користувачів, розробці модулів управління курсами, уроками та навчальними матеріалами, створенні підсистем тестування, оцінювання та

обліку успішності, реалізації чату для взаємодії між учасниками навчального процесу, а також проведенні тестування програмного забезпечення та підготовці технічної документації.

Об'єктом дослідження є процес організації та управління навчальними курсами в електронному освітньому середовищі.

Предметом дослідження є методи, технології та програмні засоби розробки інформаційних систем управління навчальними курсами з використанням вебтехнологій та програмного фреймворку Django.

Методами дослідження є аналіз предметної області та існуючих LMS-платформ, методи проєктування інформаційних систем, моделювання структури даних, програмна реалізація вебдодатків, тестування програмного забезпечення та аналіз результатів його функціонування.

Наукова новизна одержаних результатів полягає у розробці інформаційної системи, яка об'єднує функції управління навчальними курсами, тестування, оцінювання знань, контролю прогресу та комунікації користувачів у межах єдиного вебсередовища, що забезпечує комплексну підтримку дистанційного навчання.

Практична значимість роботи полягає у створенні працездатного програмного продукту, який може бути використаний закладами освіти для організації дистанційного та змішаного навчання, а також слугувати основою для подальшого розвитку повноцінної LMS-платформи.

У першому розділі виконано аналіз предметної області, розглянуто особливості сучасних систем дистанційного навчання, проведено огляд існуючих аналогів та обґрунтовано вибір технологій розробки.

Другий розділ присвячено проєктуванню інформаційної системи. Виконано аналіз предметної області, розроблено структуру бази даних, визначено архітектуру системи та описано алгоритми функціонування основних підсистем.

У третьому розділі представлено програмну реалізацію інформаційної системи управління навчальними курсами. Обґрунтовано вибір засобів

розробки, описано реалізацію основних програмних модулів, наведено інтерфейси користувача та результати тестування системи.

У четвертому розділі розглядаються питання охорони праці.

Ключові слова: інформаційна система, дистанційне навчання, LMS, Django, SQLite, вебдодаток, навчальний курс, тестування, оцінювання, електронне навчання.

ANNOTATION

Structure and Scope of the Thesis. Explanatory Note for the Bachelor's Thesis by a student in the KN 2022-1 group, majoring in 122 "Computer Science" by Anna Yevgeniivna Mukovoz on the topic "Development of an Information System for Managing Educational Courses Using the Django Software Framework" consists of 4 chapters, contains 76 pages of text, 10 figures, 3 tables, and 29 sources.

This work is dedicated to the development of an information system for managing educational courses using the Django software framework to organize distance learning and manage the educational process.

Relevance of the research. The modern educational process requires effective information systems that support distance learning, centralized storage of educational materials, monitoring of student performance, and communication among participants in the educational process. Existing distance learning platforms are often too complex to implement or have limited capabilities for adaptation to the needs of a specific educational institution. Therefore, it is relevant to create a proprietary information system for managing educational courses that combines the necessary functionality with ease of use.

The goal of this project is to develop a web-based information system for managing educational courses using the Django software framework, which enables the creation and management of educational courses, the administration of tests, the assessment of learning outcomes, communication between users, and the monitoring of the educational process.

The tasks include analyzing the subject domain and existing distance learning systems, defining functional and non-functional requirements for the system, designing the software architecture and database structure, implementing user registration and authentication mechanisms, developing course management modules, lessons, and learning materials, creating subsystems for testing, assessment, and tracking academic performance, implementing a chat feature for interaction among participants in the educational process, as well as conducting software testing and preparing technical documentation.

The object of this study is the process of organizing and managing educational courses in an e-learning environment.

The subject of this study is the methods, technologies, and software tools used to develop information systems for managing educational courses using web technologies and the Django software framework.

The research methods include analysis of the subject area and existing LMS platforms, methods of information system design, data structure modeling, software implementation of web applications, software testing, and analysis of its performance results.

The scientific novelty of the obtained results lies in the development of an information system that combines the functions of course management, testing, knowledge assessment, progress monitoring, and user communication within a single web environment, providing comprehensive support for distance learning.

The practical significance of the work lies in the creation of a functional software product that can be used by educational institutions to organize distance and blended learning, as well as serve as a basis for the further development of a full-fledged LMS platform.

The first chapter analyzes the subject area, examines the features of modern distance learning systems, reviews existing analogues, and justifies the choice of development technologies.

The second chapter is devoted to the design of the information system. A domain analysis was performed, a database structure was developed, the system architecture was defined, and the algorithms for the operation of the main subsystems were described.

The third chapter presents the software implementation of the information system for managing training courses. The choice of development tools is justified, the implementation of the main software modules is described, and the user interfaces and system testing results are presented.

The fourth chapter addresses occupational safety issues.

Keywords: information system, distance learning, LMS, Django, SQLite, web application, training course, testing, assessment, e-learning.

ЗМІСТ

ВСТУП.....	13
РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	16
1.1 Опис предметного середовища	16
1.2 Огляд наявних аналогів	17
1.3 Постановка задачі	20
Висновки до першого розділу	22
РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ	23
2.1 Аналіз предметної області.....	23
2.1.1 Вхідні дані системи.....	23
2.1.2 Вихідні дані системи	25
2.2 Проєктування системи	27
2.2.1 Архітектура системи.....	27
2.2.2 Проєктування моделі даних	28
2.2.3 Проєктування взаємодії користувачів із системою	29
2.3 Математичне та алгоритмічне забезпечення	30
2.3.1 Алгоритм автентифікації та авторизації користувачів	31
2.3.2 Алгоритм роботи з навчальними курсами.....	31
2.3.3 Алгоритм тестування та оцінювання	31
2.3.4 Алгоритм обліку прогресу навчання	32
2.3.5 Алгоритм обміну повідомленнями	32
Висновки до другого розділу	33
РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	34
3.1 Засоби розробки	34

	12
3.2 Вимоги до технічного та програмного забезпечення.....	36
3.2.1 Вимоги до середовища розробки	36
3.2.2 Вимоги до середовища експлуатації.....	36
3.2.3 Вимоги до користувацького середовища	37
3.3 Опис програмної реалізації	37
3.3.1 Ініціалізація та запуск системи.....	38
3.3.2 Реалізація системи керування користувачами.....	38
3.3.3 Реалізація модуля навчальних курсів.....	38
3.3.4 Реалізація контролю знань та комунікації	39
3.4 Керівництво користувача	40
3.4.1 Початок роботи	40
3.4.2 Перегляд каталогу курсів	42
3.4.3 Робота з навчальним курсом	43
3.4.4 Кабінет викладача.....	45
3.4.5 Завершення роботи із системою.....	45
Висновки до третього розділу	46
РОЗДІЛ 4 ОХОРОНА ПРАЦІ.....	48
4.1 Організаційно-правові основи забезпечення безпеки праці.....	48
4.2 Характеристика об'єкта та виявлення потенційних небезпек.....	49
4.3 Дослідження ризику реалізації потенційних небезпек та розробка заходів щодо їх попередження	50
Висновки до четвертого розділу	52
ЗАГАЛЬНІ ВИСНОВКИ.....	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ЛІСТИНГ КОДУ	58

ВСТУП

Сучасний етап розвитку суспільства характеризується активним впровадженням інформаційних технологій у всі сфери людської діяльності. Однією з галузей, яка зазнала суттєвих змін під впливом цифрової трансформації, є освіта. Використання інформаційно-комунікаційних технологій дозволяє підвищити ефективність навчального процесу, забезпечити оперативний доступ до освітніх ресурсів та створити умови для організації дистанційного й змішаного навчання.

Зростання потреби в електронних освітніх ресурсах зумовлює необхідність створення спеціалізованих інформаційних систем, які дозволяють централізовано керувати навчальними матеріалами, контролювати успішність студентів, здійснювати оцінювання результатів навчання та підтримувати взаємодію між учасниками освітнього процесу. Особливо актуальним це питання стало в умовах поширення дистанційної форми навчання [23], коли значна частина освітньої діяльності здійснюється за допомогою мережі Інтернет [22].

На сьогодні існує велика кількість систем управління навчанням, серед яких Moodle, Google Classroom, Canvas LMS та інші. Незважаючи на широкий функціонал, багато з них є складними для налаштування, потребують значних ресурсів для впровадження або містять надлишкові можливості, які не завжди необхідні для конкретного закладу освіти. У зв'язку з цим виникає потреба у створенні власних інформаційних систем, адаптованих до конкретних вимог користувачів та особливостей організації навчального процесу.

Актуальність теми дипломної роботи полягає у необхідності розробки сучасної веборієнтованої інформаційної системи управління навчальними курсами, яка забезпечує комплексну підтримку освітнього процесу та об'єднує в межах єдиного програмного середовища функції керування курсами, навчальними матеріалами, тестуванням, оцінюванням і комунікацією між користувачами.

Метою дипломної роботи є розробка інформаційної системи управління навчальними курсами з використанням програмного фреймворку Django, яка забезпечує створення, наповнення та супровід навчальних курсів, проведення контролю знань і підтримку взаємодії між учасниками освітнього процесу.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналіз предметної області та дослідити сучасні системи дистанційного навчання;
- виконати порівняльний аналіз існуючих LMS-платформ та визначити їхні основні переваги і недоліки;
- обґрунтувати вибір технологій та інструментальних засобів розробки програмного забезпечення;
- спроектувати архітектуру інформаційної системи та структуру бази даних;
- реалізувати механізми реєстрації, автентифікації та розмежування прав доступу користувачів
- розробити функціонал створення і керування навчальними курсами, модулями та уроками;
- реалізувати підсистему тестування, контролю знань та оцінювання результатів навчання;
- забезпечити можливість комунікації між учасниками освітнього процесу за допомогою вбудованих засобів обміну повідомленнями;
- провести тестування інформаційної системи та оцінити результати її функціонування.

Об'єктом дослідження є процес організації та управління навчальними курсами в електронному освітньому середовищі.

Предметом дослідження є методи, технології та програмні засоби розробки веборієнтованих інформаційних систем управління навчальними курсами з використанням програмного фреймворку Django.

Методами дослідження є аналіз наукової та технічної літератури, методи проєктування інформаційних систем, моделювання структури даних,

об'єктно-орієнтоване програмування, тестування програмного забезпечення та аналіз результатів його функціонування.

Практичне значення роботи полягає у створенні працездатної інформаційної системи, яка може бути використана для організації дистанційного та змішаного навчання, підтримки навчального процесу в закладах освіти, а також для подальшого розвитку й удосконалення функціоналу систем електронного навчання.

У першому розділі виконано аналіз предметної області, розглянуто особливості організації дистанційного навчання, проведено огляд існуючих аналогів та обґрунтовано вибір технологій розробки.

У другому розділі досліджено інформаційне та математичне забезпечення системи, розроблено структуру бази даних, описано архітектуру програмного рішення та алгоритми роботи основних підсистем.

У третьому розділі наведено опис програмної реалізації інформаційної системи, обґрунтовано вибір програмних засобів, представлено інтерфейс користувача та результати тестування програмного продукту.

У четвертому розділі розглянуто питання охорони праці та безпеки життєдіяльності під час роботи з комп'ютерною технікою.

РОЗДІЛ 1

ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Предметним середовищем даної дипломної роботи є організація та підтримка освітнього процесу за допомогою сучасних інформаційних технологій. Основна увага приділяється створенню інформаційної системи управління навчальними курсами, яка забезпечує взаємодію між викладачами та студентами в єдиному цифровому середовищі.

Сучасний освітній процес характеризується активним впровадженням дистанційних та змішаних форм навчання. У таких умовах виникає необхідність використання програмних засобів, які дозволяють централізовано зберігати навчальні матеріали, організувати доступ до них, здійснювати контроль знань та забезпечувати комунікацію між учасниками освітнього процесу. Саме тому системи управління навчанням (Learning Management System, LMS) стали невід'ємною складовою цифрової освіти.

Основними учасниками предметного середовища є студенти, викладачі та адміністратори системи. Викладачі відповідають за створення та наповнення навчальних курсів, формування структури модулів і уроків, розміщення навчальних матеріалів, проведення тестування та оцінювання результатів навчання. Студенти отримують доступ до навчального контенту, виконують завдання, проходять тестування, переглядають власні результати та взаємодіють з викладачами через засоби комунікації. Адміністратори здійснюють контроль роботи системи, управління користувачами та підтримку її функціонування.

Центральним елементом предметного середовища є інформаційна система управління навчальними курсами. Вона об'єднує функції створення курсів, організації навчальних матеріалів, проведення контролю знань, оцінювання навчальних досягнень та забезпечення взаємодії між користувачами. Навчальний контент у системі представлений у вигляді курсів,

які можуть містити модулі, уроки, текстові матеріали, відеофайли, презентації, PDF-документи та інші освітні ресурси.

Важливою складовою предметного середовища є контроль навчальної діяльності. Для цього система повинна забезпечувати проведення тестування, автоматичне або напівавтоматичне оцінювання результатів, збереження статистики успішності та відображення прогресу проходження навчальних курсів. Це дозволяє викладачам оперативно аналізувати результати навчання, а студентам – відстежувати власні досягнення.

Окрему роль відіграють засоби комунікації між учасниками освітнього процесу. У межах інформаційної системи користувачі повинні мати можливість обмінюватися повідомленнями, ставити запитання щодо навчальних матеріалів та отримувати зворотний зв'язок. Наявність таких механізмів сприяє підвищенню ефективності навчання та забезпечує постійну взаємодію між студентами і викладачами.

З технічної точки зору предметне середовище охоплює вебтехнології, засоби серверної обробки даних та системи керування базами даних. Для реалізації інформаційної системи використовується програмний фреймворк Django, який забезпечує створення надійної архітектури вебдодатка, підтримує механізми автентифікації користувачів, роботу з базами даних та реалізацію бізнес-логіки системи.

Таким чином, предметне середовище даної дипломної роботи являє собою комплекс взаємопов'язаних процесів, що забезпечують створення, управління та використання навчальних курсів у цифровому освітньому середовищі. Розроблювана система спрямована на підвищення ефективності організації навчання шляхом інтеграції навчального контенту, контролю знань та комунікації користувачів у межах єдиної вебплатформи.

1.2 Огляд наявних аналогів

Перед розробкою інформаційної системи управління навчальними курсами доцільно виконати аналіз існуючих програмних рішень, які використовуються для організації дистанційного навчання. Таке дослідження

дозволяє визначити найбільш ефективні підходи до реалізації функціоналу, виявити сильні та слабкі сторони наявних платформ, а також сформулювати перелік вимог до розроблюваної системи.

Однією з найпоширеніших систем дистанційного навчання є Moodle [19]. Дана платформа належить до систем класу Learning Management System (LMS) та використовується в закладах освіти різного рівня. Moodle надає широкі можливості для створення навчальних курсів, розміщення навчальних матеріалів, проведення тестування, організації форумів та оцінювання результатів навчання. До переваг системи належать відкритий програмний код, гнучкість налаштування та велика кількість додаткових модулів. Водночас складність адміністрування та перевантаженість інтерфейсу можуть створювати труднощі для користувачів, які не мають достатнього досвіду роботи з LMS-платформами.

Іншим популярним рішенням є Google Classroom [20]. Платформа інтегрована з сервісами Google Workspace та дозволяє організовувати навчальні курси, обмінюватися матеріалами, видавати завдання та здійснювати оцінювання студентів. Основною перевагою Google Classroom є простота використання та швидке налаштування освітнього середовища. Проте система має обмежені можливості щодо модифікації функціоналу та налаштування структури навчального процесу порівняно з повноцінними LMS-платформами.

Серед сучасних систем дистанційного навчання також варто виділити Canvas LMS [21]. Дана платформа підтримує управління курсами, тестування, контроль успішності, аналітику навчального процесу та різноманітні засоби комунікації між учасниками освітнього процесу. Canvas LMS відзначається сучасним інтерфейсом та високою продуктивністю, однак окремі функції можуть бути доступними лише в комерційних версіях продукту.

Окремої уваги заслуговує платформа Edmodo, яка орієнтована на організацію взаємодії між викладачами та студентами у форматі освітньої соціальної мережі. Система дозволяє публікувати навчальні матеріали,

виконувати завдання та підтримувати комунікацію між користувачами. Незважаючи на зручність використання, функціональні можливості Edmodo поступаються більш розвиненим LMS-рішенням.

Проведений аналіз показав, що більшість сучасних систем дистанційного навчання забезпечують базові функції управління навчальним процесом, проте часто характеризуються або надлишковою складністю налаштування, або обмеженими можливостями адаптації до потреб конкретного навчального закладу. Крім того, деякі рішення потребують використання додаткових сервісів для організації комунікації, тестування чи зберігання навчальних матеріалів.

Таблиця 1.1 – Порівняльна характеристика сучасних систем дистанційного навчання

Критерій порівняння	Moodle	Google Classroom	Canvas LMS	Розроблена система
Створення курсів	+	+	+	+
Модулі та уроки	+	±	+	+
Завантаження навчальних матеріалів	+	+	+	+
Відеоматеріали та презентації	+	+	+	+
Тестування	+	±	+	+
Автоматичне оцінювання	+	+	+	+
Контроль прогресу навчання	+	±	+	+
Чат та повідомлення	+	±	+	+
Домашні завдання	+	+	+	+
Практичні та лабораторні роботи	+	±	+	+
Сертифікати	+	–	+	+
Ролі користувачів	+	+	+	+
Власне розгортання системи	+	–	±	+
Модифікація програмного коду	+	–	±	+
Вартість використання	Безкоштовна	Безкоштовна	Частково платна	Безкоштовна

З огляду на результати аналізу було прийнято рішення щодо розробки власної інформаційної системи управління навчальними курсами. Розроблювана система повинна поєднувати основні переваги сучасних LMS-платформ, забезпечуючи створення та наповнення курсів, проведення

тестування, оцінювання результатів навчання, контроль прогресу студентів, організацію комунікації між користувачами та централізоване управління освітнім контентом у межах єдиного вебсередовища.

1.3 Постановка задачі

На основі проведеного аналізу предметної області та існуючих систем дистанційного навчання було встановлено необхідність створення власної інформаційної системи управління навчальними курсами, яка забезпечить ефективну взаємодію між викладачами та студентами в межах єдиного освітнього середовища.

Метою дипломної роботи є розробка інформаційної системи управління навчальними курсами з використанням програмного фреймворку Django, яка забезпечить створення, організацію та супровід навчального процесу в дистанційному форматі.

Для досягнення поставленої мети необхідно вирішити ряд взаємопов'язаних завдань.

Першим завданням є проектування архітектури програмної системи. На цьому етапі необхідно визначити структуру програмного забезпечення, розробити модель взаємодії між основними компонентами системи, сформулювати структуру бази даних та визначити ролі користувачів. Особливу увагу слід приділити забезпеченню безпечної роботи системи та можливості її подальшого розширення.

Наступним завданням є реалізація механізму реєстрації, автентифікації та авторизації користувачів. Система повинна підтримувати розмежування прав доступу між студентами, викладачами та адміністраторами, що дозволить забезпечити безпечний доступ до навчальних ресурсів та функціональних можливостей платформи.

Важливим етапом є створення підсистеми управління навчальними курсами. Викладачі повинні мати можливість створювати нові курси, додавати модулі та уроки, завантажувати навчальні матеріали, редагувати структуру курсу та здійснювати супровід освітнього процесу. Студенти, у свою чергу,

повинні отримувати доступ до навчального контенту відповідно до наданих прав доступу.

Окремим завданням є реалізація механізму проведення тестування та контролю знань. Система повинна забезпечувати створення тестів, проходження тестування студентами, автоматичну перевірку результатів та збереження отриманих оцінок у базі даних. Це дозволить автоматизувати процес контролю знань та підвищити ефективність оцінювання.

Наступним завданням є розробка підсистеми моніторингу успішності студентів. Вона повинна забезпечувати збереження інформації про результати навчання, формування статистики успішності та відображення прогресу проходження навчальних курсів.

Важливою складовою системи є реалізація засобів комунікації між учасниками освітнього процесу. Для цього необхідно створити механізм обміну повідомленнями, який дозволить студентам і викладачам підтримувати зворотний зв'язок під час навчання.

Окрему увагу необхідно приділити розробці графічного інтерфейсу користувача. Інтерфейс повинен бути інтуїтивно зрозумілим, забезпечувати зручну навігацію між розділами системи та надавати швидкий доступ до основних функцій платформи.

Завершальним етапом є тестування програмного забезпечення. У процесі тестування необхідно перевірити коректність роботи всіх функціональних модулів, оцінити продуктивність системи, перевірити правильність взаємодії з базою даних та усунути виявлені недоліки.

У результаті виконання поставлених завдань повинна бути створена повнофункціональна інформаційна система управління навчальними курсами, яка забезпечуватиме організацію дистанційного навчання, управління освітнім контентом, контроль знань та ефективну взаємодію між учасниками навчального процесу.

Висновки до першого розділу

У першому розділі дипломної роботи було проведено аналіз предметної області дистанційного навчання та сучасних інформаційних систем, що використовуються для підтримки освітнього процесу. Встановлено, що в умовах активного розвитку цифрових технологій особливої актуальності набуває використання систем управління навчанням, які забезпечують ефективну взаємодію між викладачами та студентами, централізоване зберігання навчальних матеріалів та автоматизацію контролю знань.

У процесі дослідження було визначено основні складові предметного середовища, до яких належать користувачі системи, навчальні курси, модулі, уроки, навчальні матеріали, засоби контролю знань та механізми комунікації між учасниками освітнього процесу. Встановлено, що сучасна інформаційна система управління навчальними курсами повинна забезпечувати комплексне управління освітнім контентом та підтримувати основні процеси дистанційного навчання.

Також було проведено аналіз існуючих аналогів, зокрема Moodle, Google Classroom, Canvas LMS та інших платформ дистанційного навчання. Результати дослідження показали, що зазначені системи мають широкий функціонал, однак окремі рішення характеризуються складністю налаштування, обмеженими можливостями модифікації або залежністю від сторонніх сервісів. Це підтвердило доцільність розробки власної інформаційної системи, орієнтованої на потреби конкретного навчального закладу.

На основі проведеного аналізу було сформульовано мету та основні завдання дипломної роботи. Визначено необхідність створення інформаційної системи управління навчальними курсами з використанням фреймворку Django, яка забезпечуватиме керування курсами, організацію навчальних матеріалів, проведення тестування, оцінювання результатів навчання, моніторинг успішності студентів та підтримку комунікації між користувачами.

РОЗДІЛ 2

ІНФОРМАЦІЙНЕ ТА АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз предметної області

Для проектування інформаційної системи управління навчальними курсами необхідно виконати аналіз предметної області та визначити основні сутності, що беруть участь в організації освітнього процесу. Предметна область охоплює процеси створення та супроводу навчальних курсів, розміщення навчальних матеріалів, проведення тестування, оцінювання результатів навчання та забезпечення взаємодії між учасниками освітнього процесу.

2.1.1 Вхідні дані системи

Для коректного функціонування інформаційної системи управління навчальними курсами необхідна обробка різних типів вхідних даних, які надходять від користувачів та зберігаються в базі даних. Дані можна класифікувати за джерелом їх надходження та функціональним призначенням.

1. Дані користувачів системи:

- Реєстраційні дані:
 - ім'я користувача;
 - адреса електронної пошти;
 - пароль;
 - роль користувача (студент або викладач).
- Дані профілю:
 - прізвище, ім'я та по батькові;
 - фотографія профілю;
 - контактна інформація.
- Дані авторизації:
 - логін;
 - пароль;
 - службова інформація про сеанс користувача.

2. Дані навчальних курсів:

Дані навчальних курсів формуються викладачами під час створення та наповнення курсів та зберігаються в базі даних системи.

- Інформація про курс:
 - назва курсу;
 - опис курсу;
 - автор курсу;
 - дата створення.
- Дані модулів:
 - назва модуля;
 - опис модуля;
 - порядок розташування в курсі.
- Дані уроків:
 - назва уроку;
 - текстовий матеріал;
 - посилання на відеоматеріали;
 - прикріплені файли.

3. Дані контролю знань:

- Дані тестів:
 - назва тесту;
 - перелік запитань;
 - варіанти відповідей;
 - правильні відповіді.
- Дані результатів тестування:
 - кількість набраних балів;
 - дата проходження тесту;
 - відсоток правильних відповідей.
- Дані оцінювання:
 - оцінки за завдання;
 - оцінки за тести;

- підсумкові результати навчання.

4. Дані бази даних

Усі зазначені категорії інформації зберігаються в реляційній базі даних SQLite [*]. Основними сутностями бази даних є користувачі, курси, модулі, уроки, тести, результати тестування, повідомлення та дані про прогрес навчання.

Коректна обробка наведених типів вхідних даних забезпечує стабільне функціонування інформаційної системи управління навчальними курсами та дозволяє реалізувати всі функціональні можливості платформи дистанційного навчання.

2.1.2 Вихідні дані системи

Вихідними даними системи є інформація, яка формується в результаті обробки вхідних даних та надається користувачам відповідно до їхніх прав доступу. Основною метою формування вихідних даних є забезпечення користувачів актуальною інформацією щодо навчального процесу, результатів навчання та взаємодії між учасниками освітнього середовища.

1. Дані навчальних курсів

- перелік доступних курсів;
- структура курсу, що включає модулі та уроки;
- опис курсів та навчальних матеріалів;
- інформація про викладача курсу;
- перелік доступних навчальних ресурсів.

2. Навчальні матеріали

- текстові навчальні матеріали;
- презентації;
- PDF-документи;
- відеоматеріали;
- методичні рекомендації;
- додаткові файли для виконання навчальних завдань.

3. Результати тестування та оцінювання

- отримані бали за тестування;
 - оцінки за виконані завдання;
 - результати перевірки навчальних робіт;
 - статистика проходження тестів;
 - інформація про кількість правильних та неправильних відповідей.
4. Дані про успішність та прогрес навчання
- відсоток проходження курсу;
 - кількість завершених модулів та уроків;
 - середній бал студента;
 - рейтинг успішності;
 - інформація про виконані та невиконані завдання.
5. Дані комунікації
- повідомлення чату;
 - повідомлення від викладачів;
 - повідомлення від студентів;
 - системні сповіщення;
 - інформаційні повідомлення щодо навчального процесу.
6. Адміністративна інформація
- інформацію про зареєстрованих користувачів;
 - перелік створених курсів;
 - статистику проходження курсів;
 - звіти про успішність студентів;
 - інформацію про активність користувачів.

Таким чином, вихідні дані системи забезпечують користувачів повною інформацією щодо навчального процесу, результатів навчання та взаємодії між учасниками освітнього середовища, що дозволяє ефективно організовувати дистанційне навчання та контролювати його результати.

2.2 Проєктування системи

На основі проведеного аналізу предметної області та сформульованих функціональних вимог було виконано проєктування інформаційної системи управління навчальними курсами. Основною метою даного етапу є визначення архітектури програмного забезпечення, структури його основних модулів, взаємозв'язків між компонентами та організації зберігання даних.

2.2.1 Архітектура системи

Для розробки інформаційної системи управління навчальними курсами було обрано модульну архітектуру [16]. Такий підхід дозволяє розділити функціональність системи на окремі незалежні компоненти, кожен з яких відповідає за виконання конкретних завдань. Використання модульної архітектури спрощує процес розробки, тестування, супроводу та подальшого розширення програмного забезпечення.

Система складається з таких основних модулів:

1. Модуль керування користувачами (users): Відповідає за реєстрацію, авторизацію та автентифікацію користувачів, а також за розмежування прав доступу між студентами та викладачами. Модуль забезпечує роботу з профілями користувачів та збереження персональної інформації.
2. Модуль управління курсами (courses): Є центральним компонентом системи та забезпечує створення, редагування й видалення навчальних курсів. До його функцій належить організація структури курсів, управління модулями, уроками та навчальними матеріалами.
3. Модуль тестування та оцінювання: Забезпечує створення тестових завдань, проведення контролю знань студентів, автоматичну перевірку результатів та формування оцінок. Модуль дозволяє зберігати результати тестування та аналізувати успішність користувачів.

4. Модуль комунікації: Призначений для обміну повідомленнями між учасниками освітнього процесу. Реалізує функції чату та передачі інформаційних повідомлень між студентами та викладачами.
5. Модуль бази даних: Відповідає за зберігання та обробку інформації про користувачів, курси, навчальні матеріали, результати тестування та повідомлення. Для забезпечення цілісності та надійності зберігання даних використовується система керування базами даних SQLite.
6. Адміністративний модуль: Забезпечує керування користувачами, контроль роботи системи та адміністрування навчального середовища. За допомогою адміністративної панелі здійснюється моніторинг функціонування системи та управління її основними компонентами.

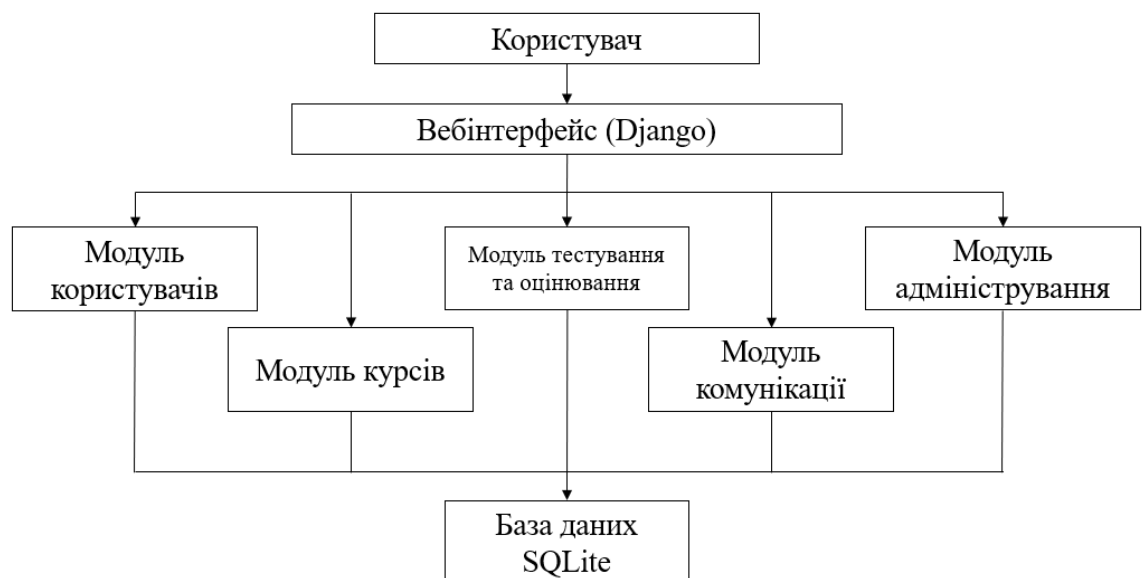


Рисунок 2.1 – Архітектура інформаційної системи управління навчальними курсами

2.2.2 Проєктування моделі даних

Для забезпечення зберігання та обробки інформації в інформаційній системі управління навчальними курсами було спроектовано реляційну базу даних [4]. У процесі розробки використано систему керування базами даних SQLite [5], яка інтегрована з фреймворком Django та забезпечує надійне

зберігання даних, підтримку зв'язків між сутностями та виконання основних операцій над даними.

Таблиця 2.1 - Основні сутності моделі даних

Сутність	Призначення
User	Зберігання інформації про користувачів системи та їх ролі
Course	Зберігання інформації про навчальні курси
Category	Категоризація навчальних курсів
Tag	Формування тематичних тегів курсів
CourseModule	Структурування курсу за окремими модулями
Lesson	Зберігання навчальних матеріалів та уроків
LessonProgress	Відстеження прогресу проходження уроків
Assignment	Зберігання домашніх, практичних та лабораторних робіт
Submission	Зберігання відповідей студентів та результатів перевірки
Quiz	Організація тестування користувачів
QuizQuestion	Зберігання питань тестів
QuizAttempt	Зберігання результатів проходження тестування
Message	Реалізація обміну повідомленнями між користувачами
FavoriteCourse	Зберігання обраних курсів користувачів
StudentGroup	Організація студентських груп у межах курсів

2.2.3 Проектування взаємодії користувачів із системою

Одним із важливих етапів проектування інформаційної системи управління навчальними курсами є організація взаємодії користувачів із функціональними компонентами системи. Для забезпечення зручності

використання було реалізовано механізм розмежування прав доступу відповідно до ролей користувачів.

У системі передбачено дві основні категорії користувачів: викладачі та студенти. Кожна категорія має власний набір функціональних можливостей, що визначає доступ до окремих модулів та операцій системи.

Після проходження процедури автентифікації користувач отримує доступ до персоналізованого інтерфейсу. Викладачі можуть створювати навчальні курси, додавати модулі та уроки, завантажувати навчальні матеріали, формувати завдання та тести, перевіряти роботи студентів і виставляти оцінки. Студенти мають можливість переглядати доступні курси, вивчати навчальні матеріали, виконувати завдання, проходити тестування та переглядати результати навчання.

Взаємодія користувачів із системою здійснюється через вебінтерфейс, реалізований засобами фреймворку Django. Усі дії користувача обробляються серверною частиною застосунку та супроводжуються зверненням до бази даних для отримання або збереження необхідної інформації.

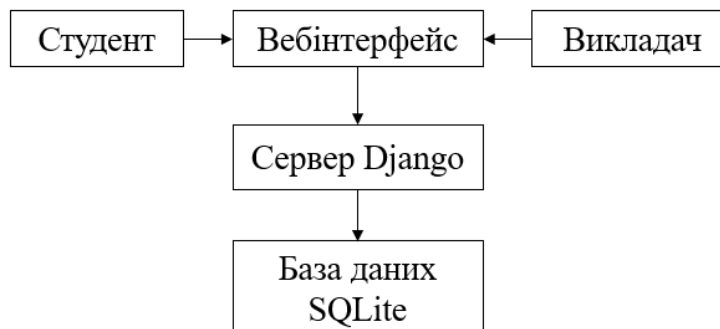


Рисунок 2.2 – Схема взаємодії користувачів із системою

2.3 Математичне та алгоритмічне забезпечення

Для реалізації функціональних можливостей інформаційної системи управління навчальними курсами було використано сукупність алгоритмів обробки даних, механізмів контролю доступу, оцінювання результатів навчання та моніторингу прогресу користувачів. Застосовані алгоритми

забезпечують ефективну взаємодію між учасниками освітнього процесу та підтримують основні функції дистанційного навчання.

2.3.1 Алгоритм автентифікації та авторизації користувачів

В основі роботи системи лежить алгоритм автентифікації користувачів, реалізований засобами фреймворку Django [3]. Процес входу до системи складається з таких етапів:

1. Користувач вводить логін та пароль.
2. Система перевіряє наявність облікового запису в базі даних.
3. Виконується перевірка правильності введеного пароля.
4. У разі успішної перевірки створюється активна сесія користувача.
5. Користувач отримує доступ до функціоналу відповідно до своєї ролі.

Застосування механізму розмежування прав доступу забезпечує безпечну роботу системи та захист інформації від несанкціонованого доступу.

2.3.2 Алгоритм роботи з навчальними курсами

Для організації навчального процесу використовується алгоритм управління курсами.

1. Викладач створює новий курс.
2. До курсу додаються модулі та уроки.
3. Завантажуються навчальні матеріали.
4. Студенти отримують доступ до курсу.
5. Система зберігає інформацію про проходження навчального матеріалу.

Алгоритм забезпечує структуроване зберігання навчального контенту та підтримує його подальше редагування.

2.3.3 Алгоритм тестування та оцінювання

Контроль знань студентів реалізується за допомогою алгоритму автоматизованого тестування.

1. Викладач формує тестові завдання.
2. Студент проходить тестування.
3. Система порівнює відповіді студента з правильними варіантами.

4. Обчислюється кількість правильних відповідей.
5. Формується підсумкова оцінка та зберігається у базі даних.
6. Результат тестування визначається за формулою:

Оцінка = (Кількість правильних відповідей / Загальна кількість питань) × 100%.

Даний підхід забезпечує об'єктивність оцінювання та автоматизує процес контролю знань.

2.3.4 Алгоритм обліку прогресу навчання

Для контролю навчальної діяльності студентів використовується алгоритм моніторингу прогресу.

1. Після завершення уроку система фіксує факт його проходження.
2. Дані зберігаються у таблиці прогресу.
3. Для кожного курсу обчислюється відсоток виконаних навчальних матеріалів.
4. Отримані результати відображаються користувачеві у вигляді статистики.
5. Відсоток проходження курсу визначається за формулою:
6. Прогрес = (Кількість завершених уроків / Загальна кількість уроків курсу) × 100%.

2.3.5 Алгоритм обміну повідомленнями

Для забезпечення взаємодії між учасниками освітнього процесу використовується алгоритм обміну повідомленнями.

1. Для забезпечення комунікації між учасниками освітнього процесу реалізовано алгоритм обміну повідомленнями.
2. Користувач створює повідомлення.
3. Повідомлення зберігається у базі даних.
4. Інший користувач отримує доступ до повідомлення.
5. За необхідності формується відповідь або надсилається вкладений файл.

Висновки до другого розділу

У другому розділі дипломної роботи було виконано проектування інформаційної системи управління навчальними курсами та розроблено її математичне й алгоритмічне забезпечення.

По-перше, було проведено аналіз предметної області, у результаті якого визначено основні сутності системи, зокрема користувачів, навчальні курси, модулі, уроки, завдання, тести, результати навчання та повідомлення. Також було визначено вхідні та вихідні дані системи, необхідні для забезпечення функціонування освітнього середовища та організації дистанційного навчання.

По-друге, на основі проведеного аналізу було виконано проектування інформаційної системи. Розроблено модульну архітектуру програмного забезпечення, що складається з модулів керування користувачами, управління курсами, тестування та оцінювання, комунікації, адміністрування та зберігання даних. Також було спроектовано модель даних, яка забезпечує зберігання інформації про користувачів, навчальні курси, навчальні матеріали, результати тестування, прогрес навчання та повідомлення.

По-третє, було розроблено математичне та алгоритмічне забезпечення системи. Описано алгоритми автентифікації та авторизації користувачів, управління навчальними курсами, проведення тестування та оцінювання результатів навчання, обліку прогресу студентів і забезпечення комунікації між учасниками освітнього процесу. Застосування зазначених алгоритмів дозволяє автоматизувати основні процеси дистанційного навчання та забезпечує ефективне функціонування інформаційної системи.

Таким чином, у другому розділі було сформовано теоретичне та технічне підґрунтя для подальшої програмної реалізації інформаційної системи управління навчальними курсами. Розроблені архітектурні рішення, модель даних та алгоритми роботи системи забезпечують можливість створення сучасного програмного продукту для організації та підтримки освітнього процесу.

РОЗДІЛ 3

ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Засоби розробки

Для створення інформаційної системи управління навчальними курсами було використано сучасні програмні засоби та технології, що забезпечують ефективність розробки, надійність роботи системи та можливість її подальшого масштабування.

1. Мова програмування Python

Основною мовою програмування при розробці системи було обрано Python [2]. Дана мова характеризується простим синтаксисом, високою продуктивністю розробки та широкими можливостями інтеграції з сучасними вебтехнологіями. Python є однією з найпопулярніших мов програмування для створення вебзастосунків, інформаційних систем та освітніх платформ [15].

2. Програмний фреймворк Django

Основою програмної реалізації системи став фреймворк Django [1]. Він реалізує архітектурний шаблон Model–View–Template (MVT), що забезпечує логічний поділ програмного коду та спрощує підтримку програмного продукту.

Основними перевагами Django є:

- наявність вбудованої системи автентифікації та авторизації користувачів;
- підтримка ORM для роботи з базами даних;
- вбудована адміністративна панель;
- високий рівень безпеки;
- можливість швидкої розробки вебзастосунків.

3. Середовище розробки:

Розробка програмного забезпечення здійснювалася в середовищі Visual Studio Code [12]. Даний редактор коду є одним із найпоширеніших інструментів для веброзробки та підтримує роботу з великою кількістю мов програмування.

Visual Studio Code забезпечує зручне редагування програмного коду, автоматичне доповнення конструкцій мов програмування, підсвічування синтаксису, інтеграцію з системою контролю версій Git, а також можливість використання розширень для розробки вебзастосунків на базі фреймворку Django. Завдяки цьому процес розробки, тестування та супроводу інформаційної системи став більш ефективним і зручним.

4. Технології веброботи

Для створення користувацького інтерфейсу використовувалися такі технології:

- HTML5 для формування структури вебсторінок [9];
- CSS3 для оформлення інтерфейсу користувача [10];
- JavaScript для реалізації інтерактивних елементів [11];
- шаблонізатор Django Templates для динамічного формування вебсторінок.

5. Система управління базами даних

Для зберігання даних було використано реляційну систему керування базами даних SQLite. Дана СУБД інтегрується з Django за замовчуванням та забезпечує надійне зберігання інформації про користувачів, курси, навчальні матеріали, результати тестування та повідомлення.

Взаємодія із базою даних здійснюється за допомогою ORM-механізму Django, який дозволяє працювати з даними через об'єкти Python без необхідності написання SQL-запитів.

6. Система контролю версій Git та платформа GitHub

Для зберігання програмного коду та контролю змін використовувалася система керування версіями Git [6]. Репозиторій проєкту розміщувався на платформі GitHub [7], що забезпечувало резервне зберігання коду, відстеження внесених змін та зручне керування процесом розробки.

6. Хмарна платформа Render

Платформа Render забезпечує автоматичне оновлення системи після внесення змін до репозиторію GitHub та надає можливість цілодобового доступу користувачів до інформаційної системи через мережу Інтернет.

3.2 Вимоги до технічного та програмного забезпечення

Для коректної розробки, розгортання та функціонування інформаційної системи управління навчальними курсами необхідно забезпечити відповідність технічного та програмного забезпечення встановленим вимогам. Дані вимоги поділяються на вимоги до середовища розробки, середовища експлуатації та користувацького середовища.

3.2.1 Вимоги до середовища розробки

Середовище розробки являє собою сукупність апаратних та програмних засобів, що використовуються для створення, тестування та налагодження інформаційної системи.

Мінімальні апаратні вимоги:

- Процесор: двоядерний процесор із тактовою частотою не менше 2 ГГц.
- Оперативна пам'ять: 4 ГБ;
- Вільне місце на диску: 5 ГБ;
- Доступ до мережі Інтернет.

Програмне забезпечення:

- Операційна система: Windows 10/11;
- Мова програмування Python версії 3.10 або вище;
- Програмний фреймворк Django;
- Середовище розробки Visual Studio Code;
- Система контролю версій Git;
- Веббраузер Google Chrome або Microsoft Edge.

3.2.2 Вимоги до середовища експлуатації

Середовище експлуатації призначене для розміщення та забезпечення безперервної роботи вебзастосунку.

Мінімальні апаратні вимоги:

- процесор: 2 vCPU;
- оперативна пам'ять: 2 ГБ;
- дисковий простір: 10 ГБ;
- стабільне підключення до мережі Інтернет.

Програмне забезпечення:

- операційна система Windows або Linux;
- Python версії 3.10 або вище;
- Django;
- SQLite;
- вебсервер, що підтримується платформою Render.

3.2.3 Вимоги до користувацького середовища

Для роботи з інформаційною системою користувачеві не потрібно встановлювати спеціалізоване програмне забезпечення. Доступ до системи здійснюється через веббраузер.

Апаратні вимоги:

- Персональний комп'ютер, ноутбук, планшет або смартфон;
- Доступ до мережі Інтернет.

Програмні вимоги:

- Сучасний веббраузер (Google Chrome, Microsoft Edge, Safari);
- Підтримка JavaScript та Cookies;
- Стабільне підключення до мережі Інтернет.

3.3 Опис програмної реалізації

Програмна реалізація інформаційної системи управління навчальними курсами виконана мовою програмування Python із використанням фреймворку Django. Система побудована відповідно до архітектури Model–View–Template (MVT) [13], що дозволяє розділити логіку обробки даних, інтерфейс користувача та механізми взаємодії з базою даних [18].

3.3.1 Ініціалізація та запуск системи

Точкою входу до програмного проєкту є файл `manage.py`, який використовується для запуску вебзастосунку, виконання міграцій бази даних та інших адміністративних команд Django.

Під час запуску системи виконується така послідовність дій:

1. Завантаження конфігураційних параметрів із файлу `settings.py`.
2. Ініціалізація встановлених застосунків проєкту.
3. Підключення до бази даних SQLite.
4. Завантаження маршрутів URL та налаштування механізму обробки HTTP-запитів.
5. Ініціалізація системи автентифікації та авторизації користувачів.
6. Запуск вебсервера та обробка запитів користувачів.

3.3.2 Реалізація системи керування користувачами

Для організації роботи користувачів використовується застосунок `users`. У ньому реалізовано власну модель користувача `User`, що успадковує функціональність стандартної моделі `AbstractUser`.

Додатково модель містить поле `role`, яке визначає роль користувача в системі, та поле `avatar`, призначене для зберігання зображення профілю.

У системі передбачено дві основні ролі:

- студент;
- викладач.

Для кожної ролі реалізовано окремий набір функціональних можливостей. Викладачі можуть створювати навчальні курси, додавати навчальні матеріали, формувати завдання та перевіряти роботи студентів. Студенти мають можливість проходити навчання, виконувати завдання та переглядати результати своєї успішності.

3.3.3 Реалізація модуля навчальних курсів

Основний функціонал системи реалізовано в застосунку `courses`. Для зберігання інформації про навчальні курси використовується модель `Course`,

яка містить назву курсу, опис, категорію, викладача, список студентів та службову інформацію щодо стану курсу.

Для структуризації навчального матеріалу використовуються моделі CourseModule та Lesson. Такий підхід дозволяє організовувати навчальний контент у вигляді модулів та окремих уроків, забезпечуючи логічну послідовність вивчення матеріалу.

Система також підтримує категоризацію та тегування курсів за допомогою моделей Category та Tag, що спрощує пошук необхідних навчальних матеріалів.

Для організації навчальних груп реалізовано модель StudentGroup, яка забезпечує об'єднання студентів у межах конкретного курсу та спрощує керування навчальним процесом.

3.3.4 Реалізація контролю знань та комунікації

Контроль знань реалізовано за допомогою моделей Assignment, Submission, Quiz, QuizQuestion та QuizAttempt. Дані моделі забезпечують створення завдань і тестів, збереження відповідей студентів та автоматичний облік результатів навчання.

Для відстеження прогресу проходження навчальних матеріалів використовується модель LessonProgress, яка дозволяє фіксувати завершення уроків та формувати статистику успішності студентів.

Обмін повідомленнями між користувачами реалізовано за допомогою моделі Message. Це забезпечує взаємодію між студентами та викладачами безпосередньо в межах інформаційної системи [17].

Взаємодія з базою даних здійснюється за допомогою ORM-механізму Django, який дозволяє працювати з даними через об'єкти Python без необхідності написання SQL-запитів. Такий підхід спрощує розробку програмного забезпечення та підвищує безпеку роботи з даними [14].

Таким чином, програмна реалізація системи забезпечує повний цикл управління навчальним процесом: від реєстрації користувачів і створення

курсів до контролю знань, відстеження прогресу навчання та організації комунікації між учасниками освітнього процесу.

3.4 Керівництво користувача

Для ефективної взаємодії кінцевого користувача з розробленим програмним продуктом було спроектовано зручний та інтуїтивно зрозумілий вебінтерфейс. Система дистанційного навчання забезпечує доступ до навчальних курсів, уроків, завдань, тестів, журналу оцінок, повідомлень, календаря дедлайнів та інструментів для викладача. Цей підрозділ описує основні сценарії використання системи та надає послідовність дій для доступу до її ключових функціональних можливостей.

3.4.1 Початок роботи

Для початку роботи з системою користувачу необхідно перейти на головну сторінку вебзастосунку EduCourse. На головній сторінці (рис. 3.1) відображаються основні елементи навігації: вхід до системи, реєстрація нового користувача, каталог курсів та календар дедлайнів.

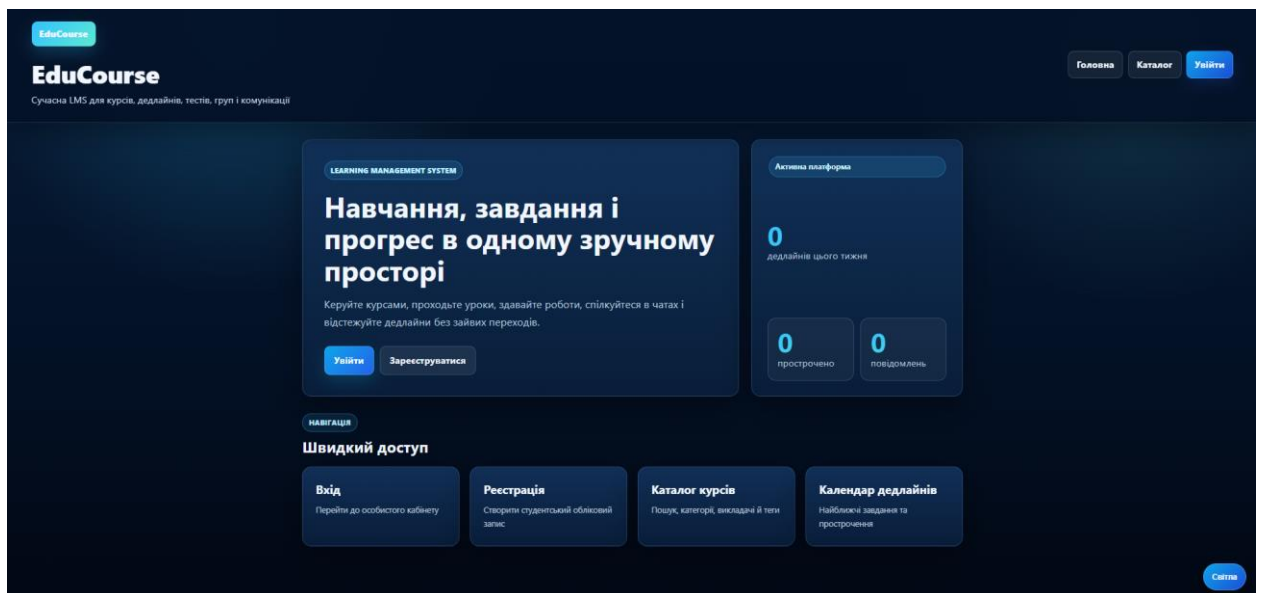


Рисунок 3.1 – Головна сторінка вебзастосунку

Якщо користувач ще не має облікового запису, необхідно натиснути кнопку «Зареєструватися» та заповнити реєстраційну форму (рис. 3.2). Після

створення акаунта користувач може увійти до системи за допомогою кнопки «Увійти», вказавши логін та пароль.

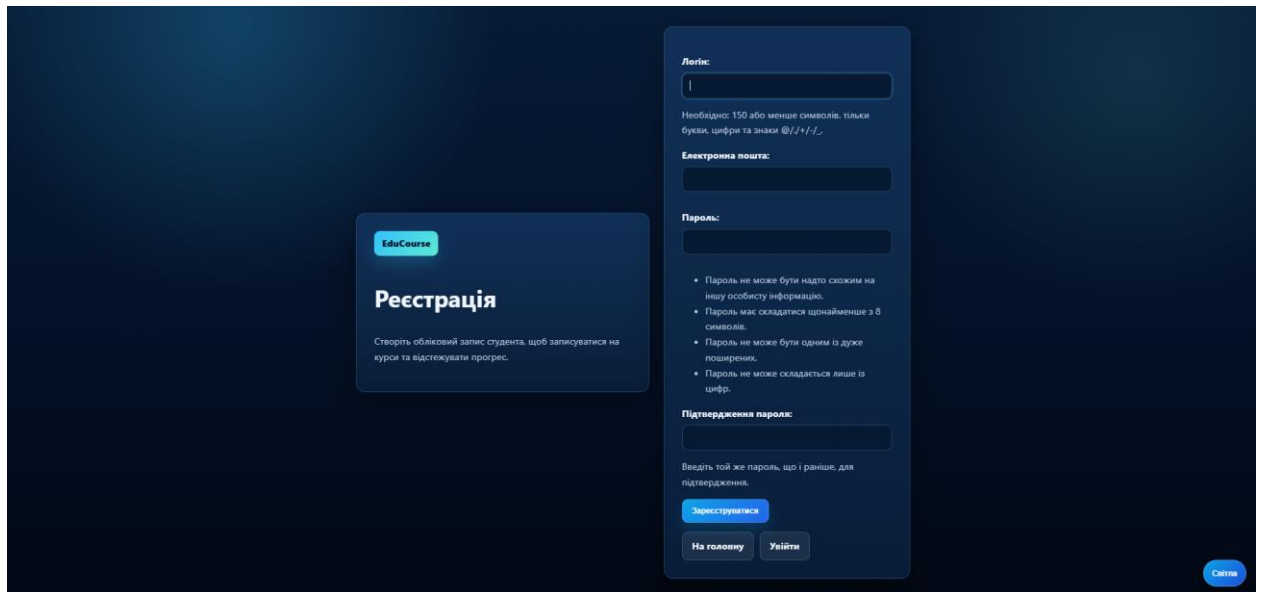


Рисунок 3.2 – Форма реєстрації

Після авторизації користувачу відкривається доступ до особистого кабінету (рис. 3.3), у якому доступні такі основні розділи:

- Мій кабінет – перегляд персональної інформації, навчальних курсів та загального прогресу.
- Каталог курсів – перегляд доступних курсів, пошук і фільтрація за категоріями, викладачами та тегами.
- Журнал оцінок – перегляд результатів виконаних завдань і тестів.
- Повідомлення – перегляд системних сповіщень про завдання, оцінки та події.
- Календар дедлайнів – перегляд найближчих термінів здачі робіт.

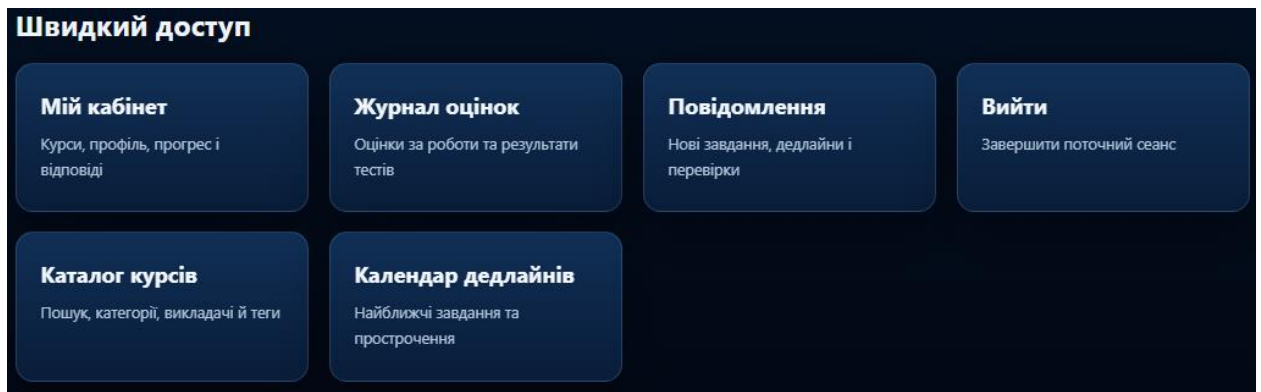


Рисунок 3.3 – Розділи, які доступні користувачам

Кабінет викладача (рис. 3.4) – доступний користувачам з роллю викладача або адміністратора.

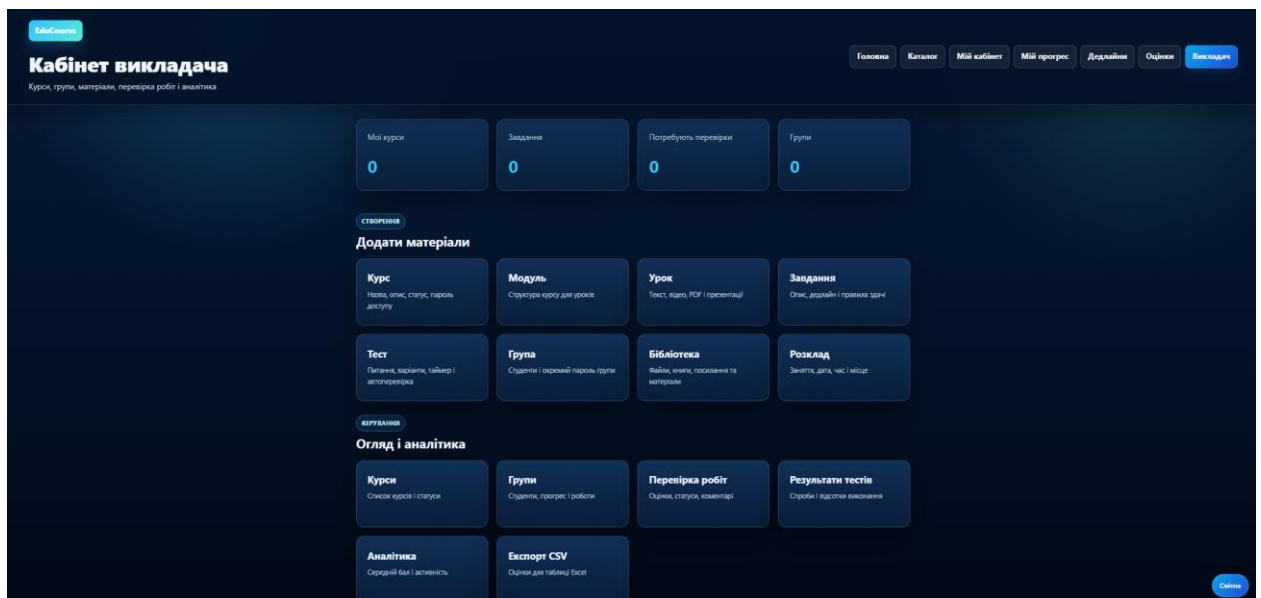


Рисунок 3.4 – Кабінет викладача

3.4.2 Перегляд каталогу курсів

Для перегляду доступних навчальних матеріалів користувач переходить до розділу «Каталог курсів» (рис. 3.5). У цьому розділі відображається список курсів із коротким описом, категорією, викладачем та іншими характеристиками.

Користувач може скористатися пошуком або фільтрами, щоб швидко знайти потрібний курс. Система підтримує фільтрацію за категоріями, викладачами, тегами, а також перегляд курсів, до яких користувач уже приєднався або які додав до обраного.

Для відкриття курсу необхідно натиснути на його назву або кнопку переходу до курсу. Якщо курс має обмежений доступ, користувачу може знадобитися ввести пароль для запису на курс.

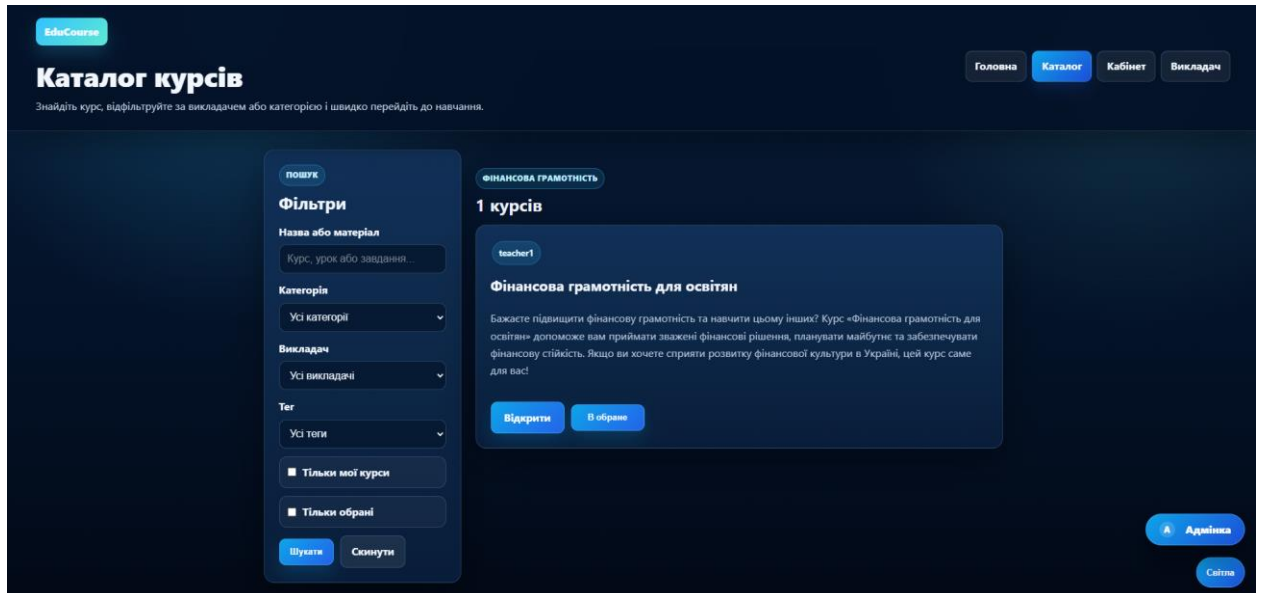


Рисунок 3.5 – Каталог курсів

3.4.3 Робота з навчальним курсом

Після відкриття сторінки курсу користувач отримує доступ до його навчальних матеріалів. Курс може містити уроки, модулі, завдання, тести, бібліотеку матеріалів, розклад подій та чат для комунікації (рис. 3.6, 3.7, 3.8).

У межах курсу студент може:

- переглядати уроки;
- відкривати текстові матеріали, відео, PDF-файли та презентації;
- позначати уроки як завершені;
- виконувати завдання;
- проходити тести;
- спілкуватися в чаті курсу;
- переглядати власний прогрес.

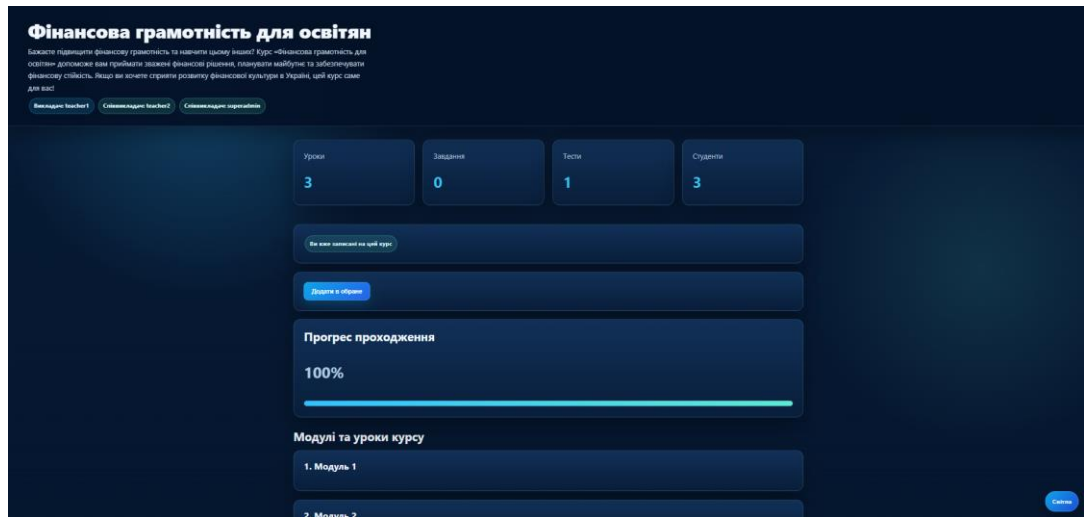


Рисунок 3.6 – Перегляд курсу

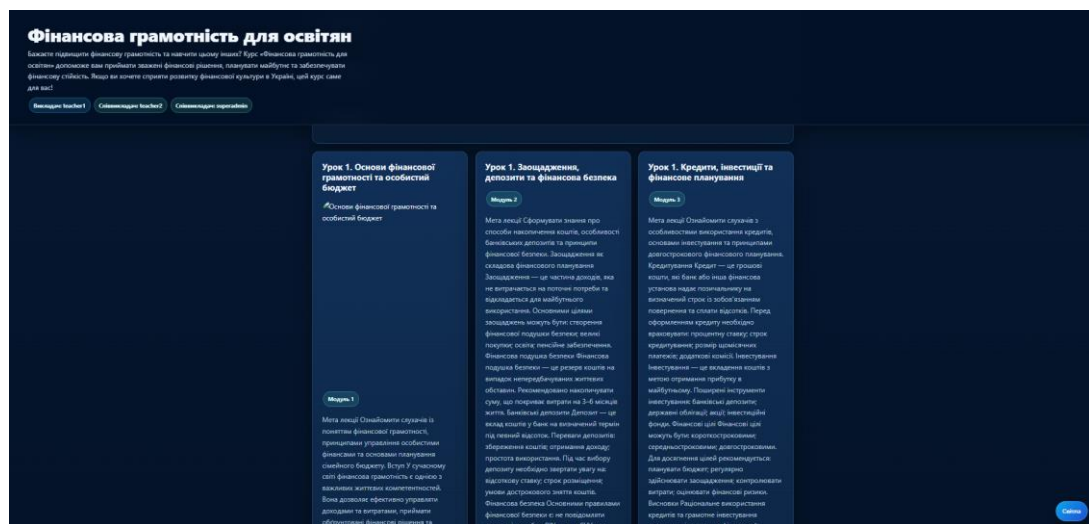


Рисунок 3.7 – Перегляд матеріалів курсу

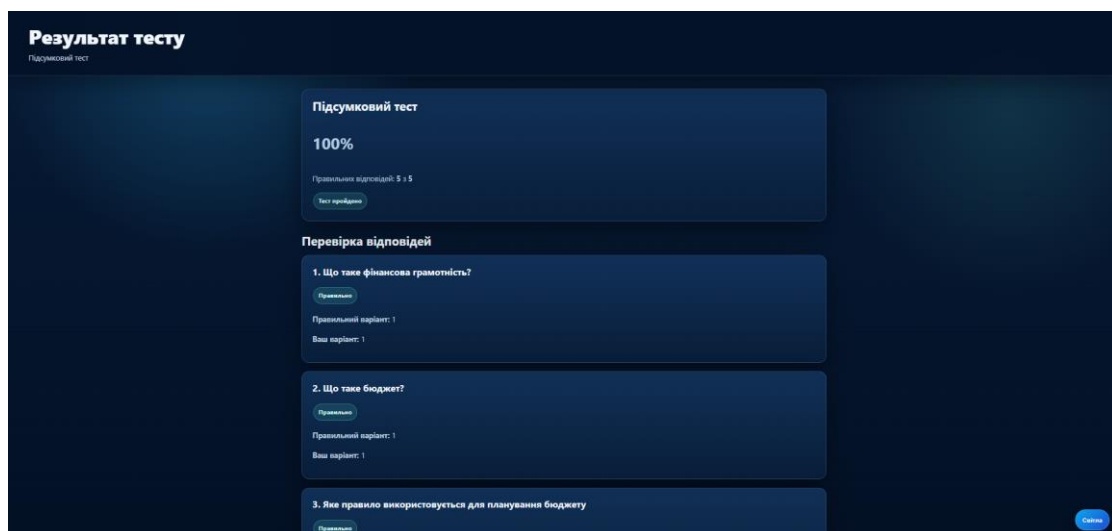


Рисунок 3.8 – Проходження тесту у межах курсу

Після проходження уроків та виконання завдань система автоматично оновлює прогрес користувача в межах курсу.

3.4.4 Кабінет викладача

Користувачі з роллю викладача мають доступ до окремого розділу «Кабінет викладача». У цьому розділі викладач може керувати навчальним процесом та створювати навчальні матеріали.

Основні можливості кабінету викладача (див. рис. 3.4):

- створення та редагування курсів;
- створення модулів і уроків;
- додавання навчальних матеріалів;
- створення завдань, практичних робіт і тестів;
- формування навчальних груп;
- перегляд зданих робіт студентів;
- оцінювання відповідей;
- перегляд аналітики за курсами;
- експорт оцінок;
- імпорт студентів;
- формування звітів.

Для створення нового курсу викладач переходить до розділу «Створити курс», заповнює назву, опис, статус, пароль доступу та інші параметри. Після створення курсу викладач може додавати до нього уроки, завдання, тести та матеріали.

3.4.5 Завершення роботи із системою

Після завершення роботи користувач може натиснути кнопку «Вийти». Система завершить поточний сеанс і поверне користувача до сторінки входу або головної сторінки вебзастосунку.

Висновки до третього розділу

У третьому розділі дипломної роботи було виконано практичну реалізацію програмного продукту – вебсистеми дистанційного навчання EduCourse, а також описано технічні та програмні аспекти, пов'язані з її створенням, розгортанням і функціонуванням.

По-перше, було обґрунтовано вибір засобів розробки, що сформували технологічний стек проекту. Як основні інструменти було обрано мову програмування Python, вебфреймворк Django, систему керування базами даних SQLite, шаблонізатор Django, HTML, CSS та JavaScript для реалізації користувацького інтерфейсу. Такий вибір дозволив створити функціональний вебзастосунок із підтримкою авторизації користувачів, розмежуванням ролей, обробкою навчальних матеріалів, завдань, тестів і результатів навчання.

По-друге, було сформульовано вимоги до програмного та технічного забезпечення, необхідного для розгортання та експлуатації системи. Визначено основні залежності проекту, структуру серверного середовища, вимоги до бази даних та компоненти, потрібні для стабільної роботи вебзастосунку.

По-третє, було детально описано програмну реалізацію ключових модулів системи. Розглянуто структуру Django-проекту, процес ініціалізації та запуску застосунку, моделі даних, маршрутизацію, шаблони сторінок і механізми взаємодії користувача із системою. Особливу увагу приділено реалізації функціональних блоків: реєстрації та авторизації користувачів, роботи з курсами, уроками, завданнями, тестами, прогресом навчання та кабінетом викладача.

На завершення було розроблено керівництво користувача, яке в доступній формі описує основні сценарії взаємодії з вебсистемою: від реєстрації та входу до перегляду курсів, виконання завдань, проходження тестів, контролю навчального прогресу та використання інструментів викладача для організації навчального процесу.

Таким чином, у ході виконання робіт над третім розділом було успішно створено програмний продукт, що відповідає поставленим завданням, має зрозумілий інтерфейс, підтримує основні процеси дистанційного навчання та є готовим до практичного використання.

РОЗДІЛ 4

ОХОРОНА ПРАЦІ

4.1 Організаційно-правові основи забезпечення безпеки праці

Охорона праці є важливою складовою організації трудової діяльності та спрямована на збереження життя, здоров'я і працездатності працівників під час виконання ними професійних обов'язків. В умовах широкого використання інформаційних технологій особливого значення набуває забезпечення безпечних умов праці для користувачів комп'ютерної техніки та інформаційних систем. Дотримання вимог охорони праці дозволяє зменшити вплив шкідливих факторів виробничого середовища, запобігти професійним захворюванням та підвищити ефективність роботи персоналу.

Правове регулювання питань охорони праці в Україні здійснюється на основі Конституції України, Закону України «Про охорону праці» [25], Кодексу законів про працю України та інших нормативно-правових актів, які встановлюють вимоги щодо створення безпечних і нешкідливих умов праці [24]. Законодавство визначає права працівників на належні умови праці, а також обов'язки роботодавця щодо забезпечення безпеки працівників під час виконання трудових функцій.

Особливого значення питання охорони праці набувають при роботі з персональними комп'ютерами та інформаційними системами. Тривале перебування за монітором, статичне навантаження на опорно-руховий апарат, підвищене навантаження на органи зору та психоемоційне напруження можуть негативно впливати на стан здоров'я працівників. Тому роботодавець повинен забезпечити належну організацію робочого місця, відповідний рівень освітлення, оптимальні параметри мікроклімату та дотримання режимів праці й відпочинку.

Важливим елементом системи охорони праці є проведення інструктажів, навчання працівників безпечним методам роботи та контроль за дотриманням встановлених вимог. Працівники, у свою чергу, зобов'язані виконувати

правила безпеки, дбати про власне здоров'я та не допускати дій, які можуть створити небезпеку для оточуючих.

Державний контроль за дотриманням законодавства у сфері охорони праці здійснюють уповноважені органи державної влади, зокрема Державна служба України з питань праці. Їх діяльність спрямована на забезпечення належного рівня безпеки праці, профілактику виробничого травматизму та попередження професійних захворювань.

Таким чином, дотримання вимог охорони праці є необхідною умовою безпечної експлуатації інформаційної системи управління навчальними курсами та забезпечення комфортних умов роботи користувачів, насамперед викладачів і працівників закладів освіти, які щоденно використовують комп'ютерну техніку для виконання своїх професійних обов'язків.

4.2 Характеристика об'єкта та виявлення потенційних небезпек

Об'єктом дослідження в даному розділі є робоче місце викладача закладу освіти, який використовує інформаційну систему управління навчальними курсами для організації та супроводу освітнього процесу.

Робоче місце розташоване в навчальному кабінеті або службовому приміщенні закладу освіти та обладнане персональним комп'ютером або ноутбуком, периферійними пристроями, мережевим обладнанням та засобами доступу до мережі Інтернет. Приміщення повинно відповідати вимогам щодо освітлення, мікроклімату, електробезпеки та пожежної безпеки [27].

Під час роботи з інформаційною системою викладач виконує завдання, пов'язані зі створенням та редагуванням навчальних курсів, завантаженням навчальних матеріалів, формуванням тестових завдань, перевіркою результатів навчання студентів, веденням електронної документації та обміном інформацією з учасниками освітнього процесу. Значна частина робочого часу проходить за персональним комп'ютером, що обумовлює вплив певних шкідливих та небезпечних факторів [26].

У процесі експлуатації інформаційної системи можуть виникати небезпечні ситуації та фактори, здатні негативно впливати на здоров'я та

працездатність користувача. До них належать фізичні, психофізіологічні та загальні безпеки.

До фізичних небезпечних факторів відносяться недостатня або надмірна освітленість робочого місця, тривале навантаження на органи зору під час роботи з монітором, несприятливі параметри мікроклімату приміщення, а також небезпека ураження електричним струмом у разі несправності обладнання чи електромережі.

До психофізіологічних факторів належать значне інформаційне навантаження, необхідність тривалої концентрації уваги, емоційне напруження під час організації освітнього процесу, а також статичне навантаження на опорно-руховий апарат внаслідок тривалого перебування в сидячому положенні.

До небезпек загального характеру належать пожежна небезпека, що може виникнути внаслідок короткого замикання електромережі або несправності електрообладнання, а також ризики, пов'язані з надзвичайними ситуаціями воєнного характеру, зокрема повітряними тривогами та ракетними обстрілами.

4.3 Дослідження ризику реалізації потенційних небезпек та розробка заходів щодо їх попередження

Після виявлення потенційних небезпек необхідно визначити рівень ризику їх реалізації. Оцінка ризику являє собою процес визначення ймовірності виникнення небезпечної події та можливих наслідків її впливу на працівника. Основною метою оцінки ризиків є виявлення найбільш небезпечних факторів виробничого середовища, визначення пріоритетності заходів захисту та зниження ймовірності виникнення нещасних випадків і професійних захворювань.

Процедура оцінювання ризиків дозволяє своєчасно виявляти небезпечні ситуації, прогнозувати їх наслідки та розробляти ефективні заходи щодо їх попередження. Для оцінки ризиків використовується матричний метод, який

базується на визначенні двох основних показників: категорії серйозності наслідків та ймовірності виникнення небезпечної події [28].

Для аналізу було обрано дві найбільш характерні небезпеки для робочого місця викладача, який використовує інформаційну систему управління навчальними курсами: підвищене напруження зору та пожежна небезпека.

1. Оцінка ризику "Підвищене напруження зору":

Під час тривалої роботи за комп'ютером працівник постійно сприймає інформацію з екрана монітора, що призводить до втоми органів зору та може спричинити погіршення зорових функцій.

Категорія серйозності наслідків – II (критична).

Ймовірність виникнення – A (часта).

Відповідно до матриці ризиків отримано індекс ризику 2A, який належить до категорії неприпустимих ризиків та потребує впровадження заходів щодо його зниження.

2. Оцінка ризику "Пожежна небезпека":

Пожежа може виникнути внаслідок короткого замикання електромережі, несправності комп'ютерної техніки або порушення правил експлуатації електрообладнання [29].

Категорія серйозності наслідків – I (катастрофічна).

Ймовірність виникнення – D (віддалена).

Згідно з матрицею ризиків отримано індекс ризику 1D, який належить до категорії небажаних ризиків та потребує впровадження профілактичних заходів.

За результатами проведеної оцінки ризиків розроблено комплекс заходів щодо підвищення рівня безпеки праці на досліджуваному об'єкті.

Таблиця 4.1 – Заходи щодо зниження рівня ризику

№	Небезпека	Запропоновані заходи	Очікуваний результат
1	Підвищене напруження зору	Організація регламентованих перерв під час роботи за комп'ютером, використання сучасних моніторів, забезпечення належного освітлення робочого місця	Зменшення навантаження на органи зору та профілактика погіршення зору
2	Статичне навантаження на опорно-руховий апарат	Використання ергономічних меблів, дотримання правильної робочої пози, виконання виробничої гімнастики	Зменшення ризику захворювань опорно-рухового апарату
3	Психоемоційне перенапруження	Раціональний розподіл робочого часу, дотримання режиму праці та відпочинку	Зниження рівня стресу та підвищення працездатності
4	Ураження електричним струмом	Регулярний технічний огляд електрообладнання та електромережі, використання справного обладнання	Підвищення рівня електробезпеки
5	Пожежна безпека	Оснащення приміщення вогнегасниками, проведення інструктажів, перевірка стану електромережі	Зниження ймовірності виникнення пожежі
6	Воєнна загроза та повітряна тривога	Ознайомлення працівників із маршрутами евакуації та правилами поведінки під час сигналу тривоги	Зменшення ризику травмування людей під час надзвичайних ситуацій

Таким чином, проведена оцінка ризиків дозволила визначити найбільш небезпечні фактори, характерні для робочого місця викладача, який використовує інформаційну систему управління навчальними курсами, та розробити комплекс організаційних і технічних заходів щодо підвищення рівня безпеки праці.

Висновки до четвертого розділу

У четвертому розділі було розглянуто питання охорони праці та безпеки життєдіяльності, пов'язані з використанням інформаційної системи управління навчальними курсами в закладах освіти.

По-перше, було проаналізовано організаційно-правові основи забезпечення безпеки праці. Розглянуто основні нормативно-правові акти України у сфері охорони праці, визначено права та обов'язки працівників і

роботодавців, а також охарактеризовано систему державного контролю за дотриманням вимог безпеки праці.

По-друге, було проведено характеристику об'єкта дослідження – робочого місця викладача закладу освіти, який використовує інформаційну систему управління навчальними курсами для організації освітнього процесу. У результаті аналізу виявлено основні потенційні небезпеки та шкідливі фактори, що можуть виникати під час роботи з комп'ютерною технікою, зокрема підвищене навантаження на органи зору, статичне навантаження на опорно-руховий апарат, психоемоційне перенапруження, ризик ураження електричним струмом, пожежну небезпеку та небезпеки, пов'язані з надзвичайними ситуаціями.

По-третє, було виконано оцінювання ризиків реалізації найбільш характерних небезпек із використанням матричного методу. Для ризиків, пов'язаних із підвищеним напруженням зору та пожежною небезпекою, визначено рівень серйозності наслідків і ймовірність виникнення, що дозволило встановити ступінь їх небезпечності та необхідність впровадження профілактичних заходів.

На основі проведеного аналізу розроблено комплекс організаційних і технічних заходів, спрямованих на зниження рівня професійних ризиків. Запропоновані рекомендації передбачають раціональну організацію робочого місця, дотримання режимів праці та відпочинку, забезпечення належного освітлення, використання справного комп'ютерного обладнання, виконання вимог електро- та пожежної безпеки, а також дотримання правил поведінки під час надзвичайних ситуацій.

Таким чином, реалізація запропонованих заходів сприятиме створенню безпечних та комфортних умов праці для викладача закладу освіти, який використовує інформаційну систему управління навчальними курсами у своїй професійній діяльності.

ЗАГАЛЬНІ ВИСНОВКИ

У ході виконання даної дипломної роботи було досягнуто поставленої мети – розроблено інформаційну систему управління навчальними курсами, призначену для організації та підтримки освітнього процесу в закладах освіти.

У процесі виконання роботи було проведено аналіз предметної області та досліджено сучасні підходи до організації дистанційного навчання. Розглянуто особливості функціонування систем управління навчанням, визначено основні вимоги до програмного продукту та сформовано перелік функціональних можливостей, необхідних для забезпечення ефективної взаємодії між викладачами та здобувачами освіти.

На основі проведеного аналізу було спроектовано архітектуру інформаційної системи, визначено структуру даних, логіку взаємодії користувачів із системою та основні програмні модулі. Обґрунтовано вибір засобів розробки, серед яких мова програмування Python, середовище розробки Visual Studio Code, система керування версіями Git, платформа GitHub для зберігання вихідного коду та хмарна платформа Render [8] для розгортання програмного продукту.

У рамках практичної реалізації було створено функціональну інформаційну систему, що забезпечує керування навчальними курсами, розміщення навчальних матеріалів, організацію тестування, зберігання результатів навчання та взаємодію між учасниками освітнього процесу. Реалізовані функціональні можливості дозволяють автоматизувати значну частину процесів, пов'язаних з організацією навчання, що сприяє підвищенню ефективності роботи викладачів та покращенню доступності навчальних ресурсів для студентів.

Окрему увагу було приділено питанням охорони праці та безпеки життєдіяльності. Проведено аналіз умов праці користувача системи, виявлено потенційні небезпеки, виконано оцінку ризиків та розроблено рекомендації щодо підвищення рівня безпеки під час роботи з комп'ютерною технікою та інформаційною системою.

Таким чином, у результаті виконання дипломної роботи створено працездатну інформаційну систему управління навчальними курсами, яка відповідає поставленим вимогам та може бути використана для підтримки освітнього процесу в закладах освіти. *Переваги розробленої системи у порівнянні з існуючими ****

*** Якщо система вже впроваджена в деякий освітній заклад, можна про це написати.

Розроблений програмний продукт має перспективи подальшого розвитку, зокрема шляхом розширення функціональних можливостей, удосконалення механізмів оцінювання знань, інтеграції додаткових сервісів та підвищення рівня інформаційної безпеки.

АПРОБАЦІЯ РОБОТИ.

Надруковано тези доповіді "Штучний інтелект у технології доповненої реальності для навчальних цілей" на III (IX) міжнародній науково-практичній конференції здобувачів вищої освіти і молодих учених «Інформаційні технології: теорія і практика». Харків: ХНУМГ ім. О.М. Бекетова, 2026 .с.591-598.

<https://ir.nmu.org.ua/entities/publication/caa62181-7105-4cd6-9694-7219ab164b3>

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. <https://docs.djangoproject.com/en/>
2. <https://docs.python.org/3/>
3. <https://docs.djangoproject.com/en/topics/auth/>
4. <https://docs.djangoproject.com/en/topics/db/models/>
5. <https://www.sqlite.org/docs.html>
6. <https://git-scm.com/docs>
7. <https://docs.github.com>
8. <https://render.com/docs>
9. <https://developer.mozilla.org/en-US/docs/Web/HTML>
10. <https://developer.mozilla.org/en-US/docs/Web/CSS>
11. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
12. <https://code.visualstudio.com/docs>
13. Vincent W. Django for Beginners / W. Vincent. – Leanpub, 2022. – 320 с.
<https://i.twirpx.link/file/3756890/>
14. Shaw A. Django in Action / A. Shaw. – Manning, 2023. – 450 с.
<https://prom.ua/m-8685980014873394424-django-action-christopher.html>
15. Lutz M. Learning Python / M. Lutz. – 5-те вид. – O'Reilly Media, 2013. – 1594 с. <https://www.yakaboo.ua/ua/learning-python.html?srsltid=AfmBOor7tzLx7ll0SOgCho1ehqyyX9QfjbWPE5haagZNQqxWm9VkDTNr>
16. Greenfeld D. Two Scoops of Django 3.x / D. Greenfeld, A. Roy. – Two Scoops Press, 2021. – 520 с.
https://www.reddit.com/r/djangolearning/comments/yak3ap/is_two_scoops_of_django_3x_still_alright_for/?tl=
17. Mele A. Django By Example /A. Mele. – Packt Publishing, 2022. – 766 с.
<https://balka-book.com/ua/python-70/django-4-by-example-build-powerful-and-reliable-python-web-applications-from-scratch-4th-edition-245903>
18. Vincent W. Django for APIs / W. Vincent. – Leanpub, 2022. – 210 с.
https://play.google.com/store/books/details/Django_for_APIs_Build_web_APIs_with_Python_and_Dja?id=0VxwDwAAQBAJ&hl=uk
19. <https://docs.moodle.org>
20. <https://support.google.com/edu>
21. <https://www.instructure.com/canvas>
22. Биков В. Ю. Моделі організаційних систем відкритої освіти / В. Ю. Биков : монографія. – К. : Атіка, 2009. – 684 с.
https://lib.iitta.gov.ua/id/eprint/845/1/Bykov_Models.pdf
23. Кухаренко В. М. Система підготовки кадрів дистанційного навчання / В. М. Кухаренко // Проблеми впровадження дистанційного навчання в освітньому процесі вищих військових навчальних закладів та можливі

шляхи їх вирішення : матеріали I міжнародної науково-практичної конференції. – Харків, 2018. – С. 98–104.

<https://nuou.org.ua/assets/documents/Zbirnyk-ADL-2020.pdf>

24. Піскунова Л.Е. Безпека життєдіяльності / Л.Е. Піскунова, В.А. Прилипко, Т.О. Зубок : підручн. – К. : Академія, 2014. – 224 с.

[https://www.yakaboo.ua/ua/bezpeka-zhittedijal-nosti-](https://www.yakaboo.ua/ua/bezpeka-zhittedijal-nosti-1228775.html?srsltid=AfmBOoriAHCi2WIyoWIANzWE83EvrXPAi_vl99cczbTraVnV7vvNOJ0Z)

[1228775.html?srsltid=AfmBOoriAHCi2WIyoWIANzWE83EvrXPAi_vl99cczbTraVnV7vvNOJ0Z](https://www.yakaboo.ua/ua/bezpeka-zhittedijal-nosti-1228775.html?srsltid=AfmBOoriAHCi2WIyoWIANzWE83EvrXPAi_vl99cczbTraVnV7vvNOJ0Z)

25. Закон України «Про охорону праці» [Електронний ресурс] / Офіційний сайт Верховної Ради України. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2694-12> . – Назва з екрану.

26. Бедрій Я.І. Основи охорони праці користувачів персональних комп'ютерів / Я.І. Бедрій : навч. посіб. – Тернопіль. : Навчальна книга – Богдан, 2014. – 144 с.

<https://www.google.com/search?q=%D0%B1%D0%B5%D0%B4%D1%80%D1%96%D0%B9+%D1%8F.%D1%96.+%D0%BE%D1%81%D0%BD%03qAIIsAIB8QVnpcGkmsjJrA&sourceid=chrome&ie=UTF-8>

27. Санітарні норми мікроклімату виробничих приміщень : ДСН 3.3.6.042-99. – [Чинний від 1999-12-01] . – Київ : МОЗ, 1999. – 21 с.

28. Коцан І.Я. Безпека життєдіяльності / І.Я. Коцан, О.Ю.Дмитрук, Є.П. Желібо : підручн. – Х. :Фоліо, 2014. – 463 с.

[https://folio.com.ua/books/bezpeka-](https://folio.com.ua/books/bezpeka-zhittediyalnosti?srsltid=AfmBOoosRf7NaMsGU-dYd01YtiZeJ-jX33pdKVtp5394xALCnfj3jbB2)

[zhittediyalnosti?srsltid=AfmBOoosRf7NaMsGU-dYd01YtiZeJ-jX33pdKVtp5394xALCnfj3jbB2](https://folio.com.ua/books/bezpeka-zhittediyalnosti?srsltid=AfmBOoosRf7NaMsGU-dYd01YtiZeJ-jX33pdKVtp5394xALCnfj3jbB2)

29. Бедрій Я.І. Охорона праці та пожежна безпека / Я.І. Бедрій : навч. посіб. – Тернопіль. : Навчальна книга – Богдан, 2013. – 184 с

<https://bohdan->

[books.com/catalog/book/105269/?srsltid=AfmBOooBfa7srg_xM4i12j3FiWN35UO6phQKA51ICT6T0-EhISamtq5k](https://bohdan-books.com/catalog/book/105269/?srsltid=AfmBOooBfa7srg_xM4i12j3FiWN35UO6phQKA51ICT6T0-EhISamtq5k)

ЛІСТИНГ КОДУ

1. Код файлу `course_system/settings.py`

```
INSTALLED_APPS = [  
    'users',  
    'courses',  
  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]  
  
MIDDLEWARE = [  
    'django.middleware.security.SecurityMiddleware',  
    'whitenoise.middleware.WhiteNoiseMiddleware',  
    'django.contrib.sessions.middleware.SessionMiddleware',  
    'django.middleware.common.CommonMiddleware',  
    'django.middleware.csrf.CsrfViewMiddleware',  
    'django.contrib.auth.middleware.AuthenticationMiddleware',  
    'course_system.middleware.AdminShortcutMiddleware',  
    'django.contrib.messages.middleware.MessageMiddleware',  
    'django.middleware.clickjacking.XFrameOptionsMiddleware',  
]  
  
ROOT_URLCONF = 'course_system.urls'  
  
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [],  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',
```

```

        ],
    },
},
]

WSGI_APPLICATION = 'course_system.wsgi.application'

# Database
# https://docs.djangoproject.com/en/6.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

DATABASE_URL = os.environ.get('DATABASE_URL')

if DATABASE_URL:
    DATABASES['default'] = dj_database_url.parse(
        DATABASE_URL,
        conn_max_age=600,
        ssl_require=not DEBUG
    )

# Password validation
# https://docs.djangoproject.com/en/6.0/ref/settings/#auth-password-
validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },

```

```

    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

```

Internationalization

<https://docs.djangoproject.com/en/6.0/topics/i18n/>

2. Код файла users/models.py

```

from django.contrib.auth.models import AbstractUser
from django.db import models

```

```

class User(AbstractUser):

```

```

    ROLE_CHOICES = (
        ('student', 'Студент'),
        ('teacher', 'Викладач'),
    )

```

```

    role = models.CharField(
        max_length=20,
        choices=ROLE_CHOICES,
        default='student',
        verbose_name='Роль'
    )

```

```

    avatar = models.FileField(
        upload_to='avatars/',
        null=True,
        blank=True,
        verbose_name='Аватар'
    )

```

```

class Meta:

```

```
verbose_name = 'Користувач'  
verbose_name_plural = 'Користувачі'
```

3. Код файлу courses/models.py

```
class Course(models.Model):  
  
    STATUS_CHOICES = (  
        ('published', 'Опубліковано'),  
        ('hidden', 'Приховано'),  
        ('archived', 'Архівовано'),  
    )  
  
    title = models.CharField(  
        max_length=200,  
        verbose_name='Назва курсу'  
    )  
  
    description = models.TextField(  
        verbose_name='Опис курсу'  
    )  
  
    category = models.ForeignKey(  
        Category,  
        on_delete=models.SET_NULL,  
        null=True,  
        blank=True,  
        related_name='courses',  
        verbose_name='Категорія'  
    )  
  
    teacher = models.ForeignKey(  
        User,  
        on_delete=models.CASCADE,  
        related_name='courses',  
        verbose_name='Викладач'  
    )  
  
    co_teachers = models.ManyToManyField(  
        User,
```

```
        related_name='assisted_courses',
        blank=True,
        verbose_name='Додаткові викладачі'
    )

students = models.ManyToManyField(
    User,
    related_name='enrolled_courses',
    blank=True,
    verbose_name='Студенти'
)

tags = models.ManyToManyField(
    Tag,
    related_name='courses',
    blank=True,
    verbose_name='Теги'
)

is_visible = models.BooleanField(
    default=True,
    verbose_name='Показувати курс'
)

enrollment_password = models.CharField(
    max_length=100,
    blank=True,
    verbose_name='Пароль для запису на курс'
)

status = models.CharField(
    max_length=20,
    choices=STATUS_CHOICES,
    default='published',
    verbose_name='Статус курсу'
)

created_at = models.DateTimeField(
    auto_now_add=True,
```

```
        verbose_name='Дата створення'  
    )  
  
    class Meta:  
        verbose_name = 'Курс'  
        verbose_name_plural = 'Курси'  
  
    def __str__(self):  
        return self.title  
  
    def is_teacher(self, user):  
        return (  
            user.is_authenticated  
            and (  
                user == self.teacher  
                or user.is_superuser  
                or self.co_teachers.filter(id=user.id).exists()  
            )  
        )  
    )
```

```
class CourseModule(models.Model):
```

```
    course = models.ForeignKey(  
        Course,  
        on_delete=models.CASCADE,  
        related_name='modules',  
        verbose_name='Курс'  
    )  
  
    title = models.CharField(  
        max_length=180,  
        verbose_name='Назва модуля'  
    )  
  
    description = models.TextField(  
        blank=True,  
        verbose_name='Опис модуля'  
    )
```

```
order = models.PositiveIntegerField(
    default=1,
    verbose_name='Порядок'
)

class Meta:
    ordering = ['order', 'id']
    verbose_name = 'Модуль курсу'
    verbose_name_plural = 'Модулі курсу'

def __str__(self):
    return f'{self.course.title} - {self.title}'

class Lesson(models.Model):

    course = models.ForeignKey(
        Course,
        on_delete=models.CASCADE,
        related_name='lessons',
        verbose_name='Курс'
    )

    module = models.ForeignKey(
        CourseModule,
        on_delete=models.SET_NULL,
        related_name='lessons',
        null=True,
        blank=True,
        verbose_name='Модуль'
    )

    title = models.CharField(
        max_length=200,
        verbose_name='Назва уроку'
    )

    content = models.TextField(
```

```
    verbose_name='Матеріал уроку'
)

category = models.CharField(
    max_length=120,
    blank=True,
    verbose_name='Категорія уроку'
)

order = models.PositiveIntegerField(
    default=1,
    verbose_name='Номер уроку'
)

banner_image = models.ImageField(
    upload_to='lesson_banners/',
    null=True,
    blank=True,
    verbose_name='Банер уроку'
)

video = models.FileField(
    upload_to='lesson_videos/',
    null=True,
    blank=True,
    verbose_name='Відеоурок'
)

pdf_material = models.FileField(
    upload_to='lesson_materials/',
    null=True,
    blank=True,
    verbose_name='PDF-матеріал'
)

presentation = models.FileField(
    upload_to='lesson_presentations/',
    null=True,
    blank=True,
```

```
    verbose_name='Презентація'  
)
```

```
class Meta:  
    ordering = ['order', 'id']  
    verbose_name = 'Урок'  
    verbose_name_plural = 'Уроки'  
  
    def __str__(self):  
        return self.title
```

```
class LessonProgress(models.Model):
```

```
    lesson = models.ForeignKey(  
        Lesson,  
        on_delete=models.CASCADE,  
        related_name='progress',  
        verbose_name='Урок'  
    )
```

```
    student = models.ForeignKey(  
        User,  
        on_delete=models.CASCADE,  
        related_name='lesson_progress',  
        verbose_name='Студент'  
    )
```

```
    completed = models.BooleanField(  
        default=True,  
        verbose_name='Пройдено'  
    )
```

```
    completed_at = models.DateTimeField(  
        auto_now_add=True,  
        verbose_name='Дата проходження'  
    )
```

```
class Meta:
```

```

verbose_name = 'Прогрес уроку'
verbose_name_plural = 'Прогрес уроків'

```

```

def __str__(self):
    return f'{self.student.username} - {self.lesson.title}'

```

```

class Assignment(models.Model):

```

```

    ASSIGNMENT_TYPES = (
        ('homework', 'Домашнє завдання'),
        ('practice', 'Практична робота'),
        ('lab', 'Лабораторна робота'),
    )

```

```

    course = models.ForeignKey(
        Course,
        on_delete=models.CASCADE,
        related_name='assignments',
        verbose_name='Курс'
    )

```

```

    title = models.CharField(
        max_length=200,
        verbose_name='Назва завдання'
    )

```

```

    description = models.TextField(
        verbose_name='Опис завдання'
    )

```

```

    assignment_type = models.CharField(
        max_length=20,
        choices=ASSIGNMENT_TYPES,
        default='homework',
        verbose_name='Тип роботи'
    )

```

```

    deadline = models.DateField(

```

```
        verbose_name='Дедлайн',
        null=True,
        blank=True
    )

    allow_late_submissions = models.BooleanField(
        default=True,
        verbose_name='Дозволити задачу після дедлайну'
    )

    class Meta:
        verbose_name = 'Завдання'
        verbose_name_plural = 'Завдання'

    def __str__(self):
        return self.title

class Submission(models.Model):

    STATUS_CHOICES = (
        ('pending', 'Очікує перевірки'),
        ('checked', 'Перевірено'),
    )

    assignment = models.ForeignKey(
        Assignment,
        on_delete=models.CASCADE,
        related_name='submissions',
        verbose_name='Завдання'
    )

    student = models.ForeignKey(
        User,
        on_delete=models.CASCADE,
        related_name='submissions',
        verbose_name='Студент'
    )
```

```
answer = models.TextField(
    verbose_name='Відповідь студента'
)

file = models.FileField(
    upload_to='submissions/',
    verbose_name='Файл',
    null=True,
    blank=True
)

grade = models.IntegerField(
    verbose_name='Оцінка',
    null=True,
    blank=True
)

status = models.CharField(
    max_length=20,
    choices=STATUS_CHOICES,
    default='pending',
    verbose_name='Статус'
)

teacher_comment = models.TextField(
    verbose_name='Коментар викладача',
    blank=True
)

is_late = models.BooleanField(
    default=False,
    verbose_name='Здано після дедлайну'
)

submitted_at = models.DateTimeField(
    auto_now_add=True,
    verbose_name='Дата відправлення'
)
```

```
class Meta:
    verbose_name = 'Відповідь студента'
    verbose_name_plural = 'Відповіді студентів'

def __str__(self):
    return f'{self.student.username} - {self.assignment.title}'
```

```
class FavoriteCourse(models.Model):
```

```
    student = models.ForeignKey(
        User,
        on_delete=models.CASCADE,
        related_name='favorite_courses',
        verbose_name='Студент'
    )
```

```
    course = models.ForeignKey(
        Course,
        on_delete=models.CASCADE,
        related_name='favorite_marks',
        verbose_name='Курс'
    )
```

```
    created_at = models.DateTimeField(
        auto_now_add=True,
        verbose_name='Дата додавання'
    )
```

```
class Meta:
    unique_together = ('student', 'course')
    verbose_name = 'Обраний курс'
    verbose_name_plural = 'Обрані курси'
```

```
def __str__(self):
    return f'{self.student.username} - {self.course.title}'
```

```
class Quiz(models.Model):
```

```
course = models.ForeignKey(
    Course,
    on_delete=models.CASCADE,
    related_name='quizzes',
    verbose_name='Курс'
)

title = models.CharField(
    max_length=200,
    verbose_name='Назва тесту'
)

description = models.TextField(
    blank=True,
    verbose_name='Опис тесту'
)

time_limit = models.PositiveIntegerField(
    default=10,
    verbose_name='Час на виконання, хв'
)

max_attempts = models.PositiveIntegerField(
    default=3,
    verbose_name='Максимальна кількість спроб'
)

show_correct_answers = models.BooleanField(
    default=True,
    verbose_name='Показувати правильні відповіді після тесту'
)

created_at = models.DateTimeField(
    auto_now_add=True,
    verbose_name='Дата створення'
)

class Meta:
```

```
verbose_name = 'Тест'  
verbose_name_plural = 'Тести'
```

```
def __str__(self):  
    return self.title
```

```
class QuizQuestion(models.Model):
```

```
    quiz = models.ForeignKey(  
        Quiz,  
        on_delete=models.CASCADE,  
        related_name='questions',  
        verbose_name='Тест'  
    )
```

```
    text = models.TextField(  
        verbose_name='Питання'  
    )
```

```
    order = models.PositiveIntegerField(  
        default=1,  
        verbose_name='Номер питання'  
    )
```

```
    option_1 = models.CharField(  
        max_length=255,  
        blank=True,  
        verbose_name='Варіант 1'  
    )
```

```
    option_2 = models.CharField(  
        max_length=255,  
        blank=True,  
        verbose_name='Варіант 2'  
    )
```

```
    option_3 = models.CharField(  
        max_length=255,
```

```
        blank=True,
        verbose_name='Варіант 3'
    )

    option_4 = models.CharField(
        max_length=255,
        blank=True,
        verbose_name='Варіант 4'
    )

    correct_option = models.PositiveIntegerField(
        default=1,
        verbose_name='Номер правильної відповіді'
    )

    class Meta:
        ordering = ['order', 'id']
        verbose_name = 'Питання тесту'
        verbose_name_plural = 'Питання тесту'

    def __str__(self):
        return self.text[:80]

class QuizAttempt(models.Model):

    quiz = models.ForeignKey(
        Quiz,
        on_delete=models.CASCADE,
        related_name='attempts',
        verbose_name='Тест'
    )

    student = models.ForeignKey(
        User,
        on_delete=models.CASCADE,
        related_name='quiz_attempts',
        verbose_name='Студент'
    )
```

```
score = models.PositiveIntegerField(
    default=0,
    verbose_name='Правильних відповідей'
)

total_questions = models.PositiveIntegerField(
    default=0,
    verbose_name='Усього питань'
)

percent = models.PositiveIntegerField(
    default=0,
    verbose_name='Результат, %'
)

started_at = models.DateTimeField(
    auto_now_add=True,
    verbose_name='Початок'
)

finished_at = models.DateTimeField(
    auto_now_add=True,
    verbose_name='Завершення'
)

time_spent_seconds = models.PositiveIntegerField(
    default=0,
    verbose_name='Витрачено секунд'
)

class Meta:
    ordering = ['-finished_at']
    verbose_name = 'Спроба тесту'
    verbose_name_plural = 'Спроби тестів'

def __str__(self):
    return f'{self.student.username} - {self.quiz.title} - {self.percent}%'
```

```
class Notification(models.Model):

    NOTIFICATION_TYPES = (
        ('info', 'Інформація'),
        ('assignment', 'Завдання'),
        ('grade', 'Оцінка'),
        ('message', 'Повідомлення'),
        ('course', 'Курс'),
    )

    user = models.ForeignKey(
        User,
        on_delete=models.CASCADE,
        related_name='notifications',
        verbose_name='Користувач'
    )

    title = models.CharField(
        max_length=160,
        verbose_name='Заголовок'
    )

    text = models.TextField(
        verbose_name='Текст'
    )

    notification_type = models.CharField(
        max_length=20,
        choices=NOTIFICATION_TYPES,
        default='info',
        verbose_name='Тип'
    )

    url = models.CharField(
        max_length=255,
        blank=True,
        verbose_name='Посилання'
    )
```

```
is_read = models.BooleanField(
    default=False,
    verbose_name='Прочитано'
)

created_at = models.DateTimeField(
    auto_now_add=True,
    verbose_name='Дата створення'
)

class Meta:
    ordering = ['-created_at']
    verbose_name = 'Повідомлення'
    verbose_name_plural = 'Повідомлення'

def __str__(self):
    return f'{self.user.username} - {self.title}'
```