

## КВАЛІФІКАЦІЙНА РОБОТА

Інформаційна система управління навчальними завданнями студентів  
Назва теми

Рівень вищої освіти перший (бакалаврський)

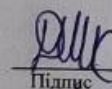
Галузь знань 12 «Інформаційні технології»  
Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»  
Шифр, назва

Освітня програма «Інформаційні системи та технології»  
Назва

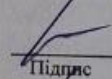
Шифр КвРІСТ 220180.22.01.06 ПЗ

Виконав здобувач IV курсу, група ІСТ-22-1

  
Підпис

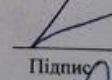
Денис СІМЧУК  
Ініціали, прізвище

Керівник канд.фіз.-мат.наук, доц.  
Науковий ступінь, учене звання

  
Підпис

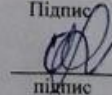
Тетяна КИСІЛЬ  
Ініціали, прізвище

Нормоконтролер канд.фіз.-мат.наук, доц.  
Науковий ступінь, учене звання

  
Підпис

Тетяна КИСІЛЬ  
Ініціали, прізвище

До захисту допускаю:  
завідувач кафедри КПС  
« 14 » червня 2026 р.

  
Підпис

Ольга ПАВЛОВА  
Ініціали, прізвище

дата

# ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

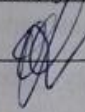
Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІС



Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

## ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Сімчук Денис Олегович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система управління навчальними завданнями студентів

Керівник проекту (роботи) Кисіль Тетяна Миколаївна, к.ф-м.н., доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 20.01.2026 р. № 7

2. Термін подання здобувачем роботи на кафедру 10.01.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) \_\_\_\_\_

Аналіз предметної області управління навчальними завданнями студентів і постановка задачі розроблення інформаційної системи

Проектування інформаційної системи управління навчальними завданнями студентів, визначення її функціональної структури, інформаційної моделі та алгоритмів основних процесів роботи

Програмна реалізація інформаційної системи управління навчальними завданнями студентів, демонстрація її функціонування та перевірка працездатності розробленого рішення

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) \_\_\_\_\_

Алгоритм роботи програми

Загальна архітектура програми

Практичне використання програми

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання

« 10 » 01 2026 р.

**КАЛЕНДАРНИЙ ПЛАН**

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю управління навчальними завданнями студентів, формулювання мети та завдань дослідження, визначення об'єкта і предмета роботи	01.02.2026	виконано
3	Робота над розділом 1 – аналіз сучасних підходів до організації навчальних завдань та постановка задачі розроблення інформаційної системи	01.03.2026	виконано
4	Робота над розділом 2 – проектування структури інформаційної системи управління навчальними завданнями, її функціональних підсистем та алгоритмів основних процесів	01.04.2026	виконано
5	Робота над розділом 3 – програмна реалізація інформаційної системи та перевірка працездатності розробленого рішення	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач

Підпис

Денис СІМЧУК

Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи

Підпис

Тетяна КИСІЛЬ

Імя, ПРІЗВИЩЕ



## АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система управління навчальними завданнями студентів».

Автор роботи: Денис СІМЧУК.

Керівник роботи: Тетяна КИСІЛЬ.

Пояснювальна записка: 60 с., 19 рис., 3 табл., 3 дод., 41 джерел.

Графічна частина: 3 креслення.

АНАЛІТИКА, АВТОРИЗАЦІЯ, БАЗА ДАНИХ, ІНФОРМАЦІЙНА СИСТЕМА, НАВЧАЛЬНІ ЗАВДАННЯ, ОЦІНЮВАННЯ, СПОВІЩЕННЯ.

Кваліфікаційна робота бакалавра присвячена розробленню інформаційної системи управління навчальними завданнями студентів, призначеної для впорядкування основних процесів освітньої взаємодії між викладачами, студентами та адміністраторами. Актуальність теми зумовлена потребою у зручному цифровому середовищі, яке дозволяє організовувати курси, створювати завдання, контролювати строки їх виконання, приймати студентські роботи, фіксувати результати перевірки та відображати навчальний прогрес. Використання такої системи дає змогу зменшити розрізненість навчальної інформації, спростити контроль дедлайнів і зробити процес обліку виконаних робіт більш прозорим.

Метою роботи є проектування, програмна реалізація та перевірка інформаційної системи для керування навчальними завданнями студентів. Для досягнення поставленої мети проаналізовано предметну область і сучасні освітні платформи, визначено логіку роботи користувачів за ролями, спроектовано функціональну структуру та інформаційну модель системи, розроблено алгоритми подання й оцінювання студентських робіт. Практична частина роботи охоплює реалізацію засобів керування курсами й завданнями, контролю дедлайнів, завантаження матеріалів, перевірки робіт, формування сповіщень, перегляду прогресу та підготовки звітної інформації для викладача.







Підпис здобувача

30.05.2026

Дата

## ЗМІСТ

Вступ .....	4
1 Аналіз предметної області управління навчальними завданнями студентів та постановка задачі щодо розроблення інформаційної системи .....	6
1.1 Актуальність розроблення інформаційної системи управління навчальними завданнями студентів .....	6
1.2 Аналіз сучасних інформаційних систем і підходів до управління навчальними завданнями студентів .....	8
1.3 Постановка задачі .....	13
1.4 Висновки до першого розділу .....	14
2 Проектування інформаційної системи управління навчальними завданнями студентів .....	16
2.1 Проектування логіки організації роботи з навчальними завданнями .....	16
2.2 Проектування функціональної структури інформаційної системи .....	20
2.3 Проектування інформаційної моделі обліку навчальних завдань і результатів їх виконання .....	25
2.4 Проектування алгоритмів обробки навчальних подій у системі .....	29
2.5 Висновки до другого розділу .....	34
3 Програмна реалізація та перевірка працездатності інформаційної системи управління навчальними завданнями студентів .....	35
3.1 Реалізація серверної частини та механізмів доступу користувачів .....	35
3.2 Реалізація бази даних і збереження навчальної інформації .....	38
3.3 Реалізація клієнтського інтерфейсу інформаційної системи .....	41
3.4 Демонстрація основних сценаріїв роботи та перевірка працездатності системи .....	47
3.5 Висновки до третього розділу .....	51
Висновки .....	53
Перелік джерел посилань .....	55
Додаток А (обов'язковий) Копія креслення «алгоритм роботи програми» ....	60

КвРІСТ 220180.22.01.06_ПЗ				
Зм.	Арк.	№ док.ум.	Підпис	Дата
Виконав		Денис СІМЧУК		
Перевір.		Тетяна КИСІЛЬ		
Н.контр.		Тетяна КИСІЛЬ		
Затвер.		Ольга ПАВЛОВА		06.03
Інформаційна система управління навчальними завданнями студентів			Літера	Аркуш
			у	2
ХНУ ІСТ-22-1			Аркушів	66

Додаток Б (обов'язковий) копія креслення «загальна архітектура інформаційної системи» .....	61
Додаток В (обов'язковий) копія креслення «практичне використання програми» .....	62

					КвРІСТ 220180.22.01.06 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		3

## ВСТУП

Навчальний процес у закладах вищої освіти пов'язаний із постійним рухом великої кількості інформації, де викладачі формують курси, створюють завдання, визначають строки їх виконання, перевіряють подані роботи та фіксують результати оцінювання, а студенти мають своєчасно відстежувати власне навчальне навантаження, виконувати роботи й реагувати на коментарі викладача. За відсутності єдиного цифрового середовища ці дії часто розподіляються між різними каналами комунікації, електронною поштою, месенджерами, окремими файлами та локальними таблицями. Унаслідок цього ускладнюється контроль дедлайнів, відстеження статусів робіт, збереження історії перевірок і формування цілісного уявлення про навчальний прогрес студентів.

Для студента важливо бачити не лише перелік робіт, а й їхній поточний стан, а саме що потрібно виконати, що вже подано, які завдання перевірено, де виставлено оцінку, а які матеріали повернуто на доопрацювання. Для викладача необхідним є впорядкований перегляд поданих робіт, контроль прострочених здач, швидке оцінювання, надання коментарів і підготовка звітної інформації за курсом. Якщо ці процеси не об'єднані в одну систему, частина навчальних даних втрачає зв'язок між собою, а облік результатів стає менш прозорим і більш залежним від ручної роботи.

Саме тому розроблення інформаційної системи управління навчальними завданнями студентів є актуальним завданням, спрямованим на впорядкування основних процесів навчальної взаємодії та автоматизацію обліку результатів виконання робіт.

Метою дипломної роботи є розроблення інформаційної системи управління навчальними завданнями студентів, яка забезпечує організацію курсів, створення та супровід завдань, подання студентських робіт, контроль строків виконання, перевірку й оцінювання результатів, формування сповіщень,

					КвРІСТ 220180.22.01.06 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

відображення прогресу та підготовку звітної інформації для викладача. Для досягнення поставленої мети необхідно виконати такі завдання:

- 1) проаналізувати предметну область і сучасні підходи до цифрової організації навчальних завдань;
- 2) визначити вимоги до інформаційної системи та ролі її користувачів;
- 3) спроектувати логіку роботи системи, її функціональну структуру та інформаційну модель;
- 4) розробити алгоритми подання, перевірки й оцінювання робіт;
- 5) реалізувати програмні засоби керування курсами, завданнями, сповіщеннями, прогресом і звітністю;
- 6) провести перевірку працездатності створеної інформаційної системи.

Об'єктом дослідження є процес організації та контролю виконання навчальних завдань студентів у цифровому освітньому середовищі.

Предметом дослідження є моделі, методи та програмні засоби побудови інформаційної системи управління навчальними завданнями студентів.

Практичне значення одержаних результатів полягає у створенні інформаційної системи, що відтворює повний цикл роботи з навчальними завданнями, від формування курсу й публікації роботи до подання результату студентом, перевірки викладачем, виставлення оцінки, надсилання повідомлення та формування підсумкових показників.

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ УПРАВЛІННЯ НАВЧАЛЬНИМИ ЗАВДАННЯМИ СТУДЕНТІВ ТА ПОСТАНОВКА ЗАДАЧІ ЩОДО РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Актуальність розроблення інформаційної системи управління навчальними завданнями студентів

Організація роботи з навчальними завданнями є однією з основних складових сучасного освітнього процесу. У закладах вищої освіти студент одночасно працює з кількома дисциплінами, отримує різні типи завдань, дотримується встановлених термінів і взаємодіє з викладачами щодо перевірки та оцінювання результатів. За умов активного використання цифрових технологій ці процеси потребують упорядкованого інформаційного супроводу, а не розрізненого обміну файлами, повідомленнями та таблицями. Розвиток цифрового освітнього середовища також передбачає створення систем, які не лише надають доступ до матеріалів, а й підтримують керування навчальною діяльністю. Значення таких рішень підкреслюється в сучасних підходах до цифрової трансформації освіти.

Навчальне завдання у межах інформаційної системи виступає не просто текстовим описом роботи, а структурованим об'єктом із власними параметрами: назвою, дедлайном, максимальною оцінкою, вимогами до файлів, статусом виконання та результатом перевірки. Саме тому системи управління навчанням розглядаються як важлива основа для організації курсів, взаємодії студентів і викладачів та контролю виконання робіт. У теоретичних підходах до LMS наголошується, що такі платформи мають підтримувати не лише доступ до матеріалів, а й логіку освітнього процесу. Практика використання сучасних навчальних систем підтверджує важливість функцій створення завдань, подання робіт, перевірки результатів і моніторингу прогресу. Проблеми виникають тоді, коли робота із завданнями не об'єднана в єдиному цифровому середовищі. У такому випадку частина інформації може надсилатися електронною поштою,

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						6
Зм.	Арк.	№ докум.	Підпис	Дата		

частина через месенджери, а оцінки та коментарі зберігатися окремо. Це ускладнює контроль за дедлайнами, статусами задач і загальною картиною виконання курсу.

Для студентів особливо важливими є інструменти планування та відстеження власного прогресу, оскільки здатність організувати навчальну діяльність прямо пов'язана з результативністю навчання. Проблема своєчасного виконання робіт також посилюється через академічну прокрастинацію, зокрема в умовах дистанційного або змішаного навчання. Наявність чітких термінів і візуального контролю дедлайнів впливає на дисципліну подання завдань. Окрему роль відіграють сповіщення та зворотний зв'язок. Повідомлення про наближення дедлайну, зміну статусу роботи або виставлення оцінки допомагають студенту орієнтуватися у поточному стані навчання. Push-сповіщення й автоматичні нагадування можуть підтримувати залученість і зменшувати відкладання навчальних дій. Не менш важливою є можливість отримувати коментарі викладача після перевірки роботи. Якісний feedback підвищує ефективність навчання, підтримує формувальне оцінювання і допомагає студенту правильно використовувати зауваження для покращення результату.

Онлайн-зворотний зв'язок також сприяє розвитку самоконтролю та навчальної саморегуляції. Інформаційна система управління навчальними завданнями корисна не лише для окремого студента, а й для викладача та закладу освіти загалом. Вона може накопичувати дані про кількість виданих завдань, подані роботи, середні оцінки, прострочені задачі та загальний рівень виконання курсу. Такі можливості пов'язані з підходами навчальної аналітики, що використовуються для моніторингу освітнього процесу й підтримки прийняття рішень. Аналітичні інструменти дозволяють відстежувати прогрес студентів і виявляти ризики відставання. Систематизація таких даних підвищує обґрунтованість управління навчальним процесом, хоча ефективність аналітики залежить від її практичного зв'язку з реальними освітніми діями.

					КвРІСТ 220180.22.01.06 ПЗ	Арк.
						7
Зм.	Арк.	№ докум.	Підпис	Дата		

Отже, розроблення інформаційної системи управління навчальними завданнями студентів є актуальним завданням. Така система має об'єднати роботу з курсами, завданнями, дедлайнами, здачею файлів, перевіркою, оцінюванням, сповіщеннями та аналітикою в єдиному цифровому середовищі. Це дозволяє зробити навчальний процес більш організованим, прозорим і зручним як для студентів, так і для викладачів.

## 1.2 Аналіз сучасних інформаційних систем і підходів до управління навчальними завданнями студентів

Сучасні інформаційні системи в освіті дедалі частіше використовуються не лише для розміщення навчальних матеріалів, а й для повноцінної організації роботи із завданнями студентів. Такі платформи забезпечують створення курсів, призначення робіт, встановлення дедлайнів, подання файлів, перевірку результатів, виставлення оцінок і надання зворотного зв'язку. Завдяки цьому навчальні завдання переходять із розрізненого формату листування, файлів і таблиць у впорядковану цифрову структуру, де кожен етап виконання має власний статус. Однією з найвідоміших платформ цього типу є Moodle. У ній завдання створюються як окремі навчальні елементи курсу. Студент може подати відповідь у текстовому форматі або завантажити файли, а викладач отримує доступ до перевірки, оцінювання та коментування результатів. Для завдання можна встановити строки подання, обмеження доступності, формат оцінювання, можливість повторної здачі та інші параметри, що дозволяє адаптувати його до різних навчальних сценаріїв. Moodle також підтримує рубрики, сповіщення та журнал оцінок, що робить систему зручною для контролю виконання робіт і ведення результатів курсу (рисунок 1.1).

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						8
Зм.	Арк.	№ докум.	Підпис	Дата		



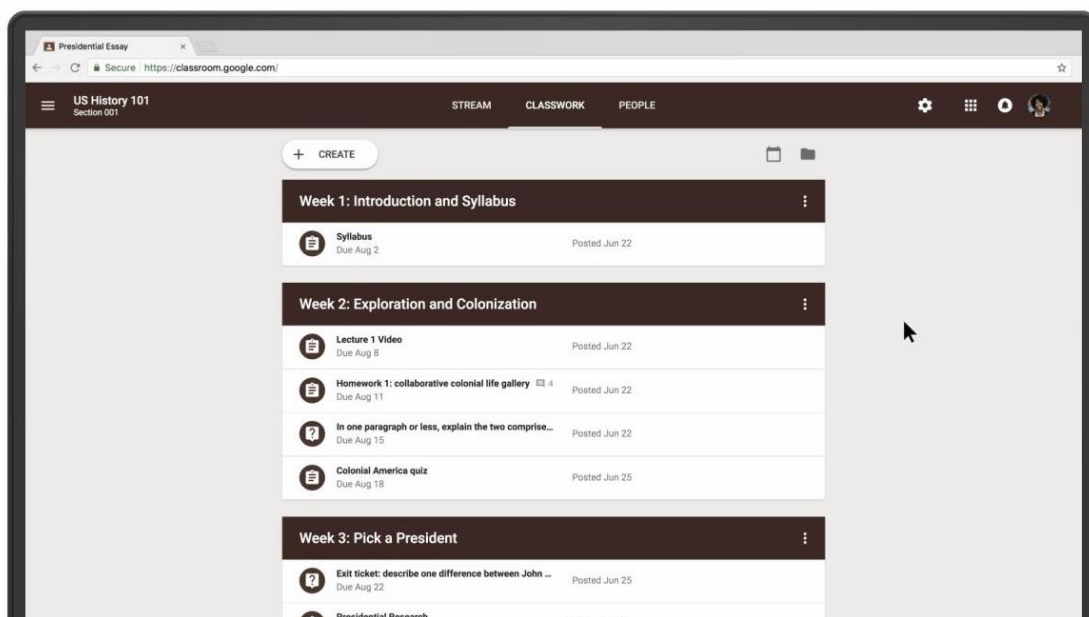


Рисунок 1.2 – Приклад організації навчальних завдань на сторінці Classwork у Google Classroom [2]

Canvas LMS є більш розвинутою системою з погляду оцінювання та аналітики. Платформа підтримує створення завдань, гнучке налаштування параметрів курсу, журнал оцінок і системи перевірки студентських робіт. Особливе значення має інструмент SpeedGrader, який дозволяє викладачу переглядати подання студентів, виставляти бали, залишати коментарі й працювати з рубриками в одному інтерфейсі. Система також має налаштування сповіщень і аналітичні засоби, що допомагають відстежувати активність студентів та їхній прогрес у межах курсу (рисунок 1.3).

Microsoft Teams for Education поєднує управління навчальними завданнями з комунікаційним середовищем. Викладач може створювати завдання, додавати інструкції, прикріплювати ресурси, задавати дедлайни, налаштовувати оцінювання й призначати роботи окремим студентам або групам. У модулі Grades передбачено перегляд статусів студентських робіт, оцінок і динаміки виконання завдань. Додаткові аналітичні інструменти дозволяють бачити загальну картину навчальної активності та підтримують контроль освітнього процесу. Створення навчально завдання зображено на рисунку 1.4.

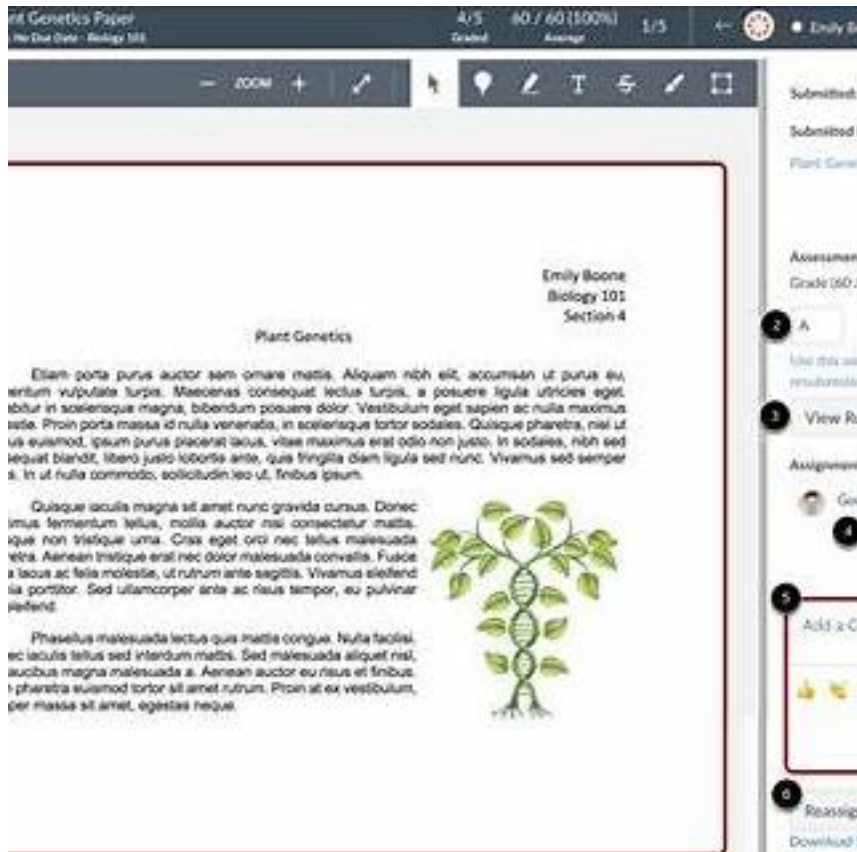


Рисунок 1.3 – Приклад перевірки студентської роботи у SpeedGrader системи Canvas [3]

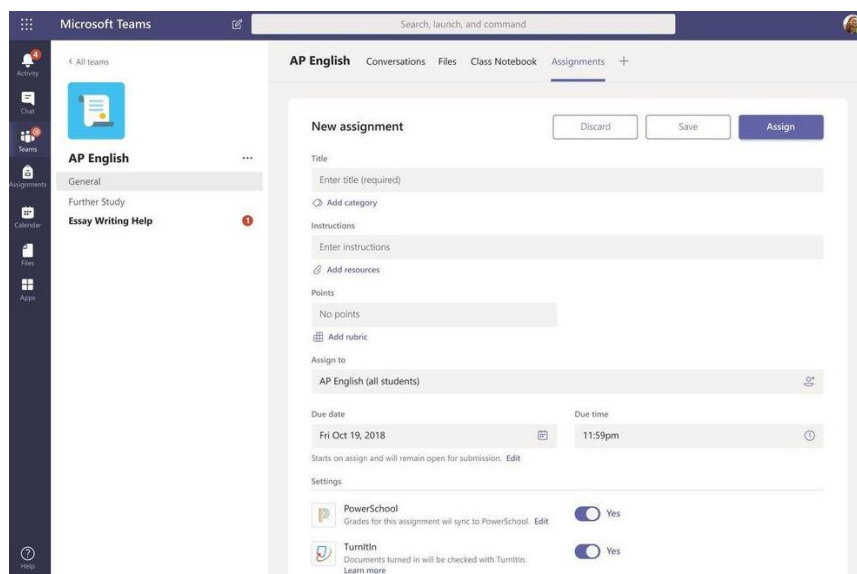


Рисунок 1.4 – Приклад створення навчального завдання у Microsoft Teams for Education [4]

Зм.	Арк.	№ докум.	Підпис	Дата

Розглянуті платформи демонструють спільну логіку побудови інформаційної систем для управління навчальними завданнями. У кожній із них підтримуються курси, завдання, дедлайни, подання робіт, оцінювання й сповіщення. Водночас підходи до організації цих процесів відрізняються. Moodle орієнтований на глибоку гнучкість налаштувань, Google Classroom забезпечує простоту й швидкий старт, Canvas сильніше акцентує увагу на перевірці та аналітиці, а Microsoft Teams поєднує навчальні інструменти з комунікацією та командною роботою. Порівняльні дослідження підтверджують, що ефективність LMS залежить від поєднання функціональності, зручності інтерфейсу та відповідності платформи реальним освітнім сценаріям.

Для розроблення інформаційної системи доцільно виділити саме ті можливості, які безпосередньо підтримують основний навчальний цикл. До них належать створення курсів, призначення завдань, встановлення строків виконання, подання файлів студентами, фіксація статусу здачі, перевірка й оцінювання робіт, повернення на доопрацювання, а також інформування учасників процесу про зміни.

Саме узгодженість цих механізмів визначає практичну цінність системи, а не лише кількість окремих функцій. Окреме значення має зручність користування платформою. Надто складна структура може уповільнювати виконання типових дій, тоді як надмірно спрощений інтерфейс обмежує можливості викладача. Порівняльні оцінки різних навчальних систем показують, що на сприйняття платформи впливають зрозумілість навігації, доступність основних функцій і відповідність інтерфейсу потребам користувачів.

Важливою складовою сучасних систем є аналітика навчальної діяльності, що дозволяє відображати кількість виконаних завдань, частку прострочених робіт, середній бал, прогрес студента й загальний стан курсу. Подібні інструменти допомагають викладачу швидше оцінювати ситуацію в групі, а студенту бачити власні результати та проблемні зони. Саме тому в

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 12
Зм.	Арк.	№ докум.	Підпис	Дата		

інформаційній системі управління навчальними завданнями доцільно передбачити формування статистики, звітів і показників особистого прогресу.

Аналіз сучасних рішень показує, що управління навчальними завданнями є окремим важливим напрямом розвитку освітніх інформаційних систем. Більшість відомих платформ підтримує повний цикл роботи із завданнями, проте кожна реалізує його з різним рівнем складності та акцентами, тому доцільно створити спеціалізовану інформаційну систему, зосереджену саме на керуванні навчальними завданнями студентів, обліку дедлайнів, статусів, оцінок, сповіщень і результатів прогресу.

### 1.3 Постановка задачі

Головною метою кваліфікаційної роботи є розробка інформаційної системи управління навчальними завданнями студентів, призначеної для впорядкування взаємодії між викладачами, студентами та адміністраторами в межах освітнього процесу. Система має забезпечувати створення курсів і завдань, подання студентських робіт, контроль дедлайнів, перевірку результатів, виставлення оцінок, формування сповіщень і відображення навчального прогресу.

Розроблюване рішення повинно моделювати типовий цикл роботи з навчальними завданнями: створення курсу, призначення завдання, його виконання студентом, здачу роботи, перевірку викладачем, оцінювання та подальше узагальнення результатів. Такий підхід дозволяє сформувавши цілісне інформаційне середовище для організації навчальної діяльності та контролю її основних етапів. Для досягнення поставленої мети необхідно виконати такі завдання:

- 1) проаналізувати предметну область і наявні системи управління навчальними завданнями;
- 2) визначити ролі користувачів і основні вимоги до системи;

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 13
Зм.	Арк.	№ докум.	Підпис	Дата		

- 3) спроектувати архітектуру інформаційної системи та її основні модулі;
- 4) реалізувати функції створення курсів, завдань, подання і перевірки робіт;
- 5) забезпечити контроль дедлайнів, статусів, оцінок і сповіщень;
- 6) реалізувати відображення прогресу студентів і формування звітності для викладача;
- 7) провести тестування розробленої системи.

Поставлена задача передбачає створення інформаційної системи, яка об'єднує основні процеси роботи з навчальними завданнями в єдиному цифровому середовищі. У результаті виконання роботи має бути отриманий функціональний програмний продукт, що забезпечує автоматизацію обліку завдань, контроль їх виконання, оцінювання студентських робіт і подання узагальненої інформації про результати навчальної діяльності.

#### 1.4 Висновки до першого розділу

Проведений аналіз показав, що розроблення інформаційної системи управління навчальними завданнями студентів є актуальним. У сучасному навчальному процесі важливо зручно організувати створення завдань, контроль термінів виконання, подання робіт, їх перевірку та виставлення оцінок. Коли ці процеси зібрані в одному цифровому середовищі, студентам легше орієнтуватися у власних завданнях, а викладачам простіше контролювати перебіг навчальної роботи.

Було розглянуто сучасні освітні платформи, які використовуються для керування курсами й завданнями. Вони дають змогу публікувати навчальні роботи, встановлювати дедлайни, приймати файли студентів, оцінювати результати та надсилати сповіщення. Водночас такі системи відрізняються рівнем складності, гнучкістю налаштувань і набором аналітичних можливостей.

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

Це показує доцільність створення окремої інформаційної системи, зосередженої саме на повному циклі роботи з навчальними завданнями.

У процесі аналізу визначено основні функції майбутньої системи. Вона має підтримувати ролі користувачів, роботу з курсами та завданнями, контроль дедлайнів і статусів здачі, оцінювання робіт, повернення їх на доопрацювання, сповіщення, перегляд прогресу студентів і формування звітів для викладача.

Поєднання цих можливостей дає змогу зробити навчальну взаємодію більш зрозумілою, впорядкованою та зручною. Постановка задачі передбачає розроблення інформаційної системи управління навчальними завданнями студентів, яка допоможе автоматизувати облік робіт, контроль їх виконання та подання узагальненої інформації про навчальну активність. Така система має бути корисною як для студентів, так і для викладачів, оскільки спрощує організацію навчального процесу та робить його більш прозорим.

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						15
Зм.	Арк.	№ докум.	Підпис	Дата		

## 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМИ ЗАВДАННЯМИ СТУДЕНТІВ

### 2.1 Проєктування логіки організації роботи з навчальними завданнями

Інформаційна система управління навчальними завданнями студентів має відтворювати послідовність дій, характерну для реального освітнього процесу. Її основне призначення полягає не лише у зберіганні переліку робіт, а в організації повного циклу взаємодії між викладачем і студентом, починаючи від створення курсу та призначення завдання і завершуючи перевіркою поданої роботи, виставленням оцінки та відображенням результатів навчальної діяльності. Логіка функціонування системи побудована навколо навчального курсу.

Курс є основною організаційною одиницею, у межах якої викладач формує перелік завдань, визначає їх зміст, дедлайни, максимальну кількість балів і правила подання. До курсу приєднуються студенти, які отримують доступ до опублікованих навчальних робіт, переглядають вимоги до них і виконують завдання у встановлений строк. Така структура дозволяє поєднати дисципліну, викладача, студентську групу та результати виконання завдань в єдиному інформаційному просторі. Після створення завдання система переходить до етапу його виконання студентами. Кожне завдання має власний стан, який змінюється залежно від дій користувачів. Спочатку воно доступне для перегляду та виконання. Після завантаження файлів студентом формується подання роботи. У цей момент система фіксує дату й час здачі, порівнює їх із установленим дедлайном і визначає, чи робота подана вчасно.

Якщо строк виконання ще не минув, подання отримує статус звичайної задачі. Якщо дедлайн уже завершився, робота позначається як прострочена. Такий механізм дозволяє автоматизувати контроль термінів і зменшує кількість ручних перевірок з боку викладача. Наступним етапом є опрацювання роботи викладачем. Після перегляду поданих матеріалів викладач може виставити оцінку та залишити текстовий коментар. Якщо результат потребує виправлення,

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

робота повертається студенту на доопрацювання. За умови, що завдання дозволяє повторну здачу, студент отримує можливість завантажити оновлений варіант.

Таким чином, система підтримує не лише одноразове подання файлу, а повноцінний навчальний цикл із перевіркою, зворотним зв'язком і коригуванням результату. Окреме місце в логіці системи займають сповіщення. Вони мають супроводжувати головні події, важливі для користувача. Студент повинен отримувати повідомлення про нове завдання, наближення дедлайну, виставлення оцінки або повернення роботи на доопрацювання. Викладачу, зі свого боку, корисно бачити появу нових подань від студентів і загальний стан виконання завдань у межах курсу. Завдяки цьому інформаційна система не лише накопичує дані, а й підтримує своєчасну взаємодію між учасниками навчального процесу. Завершальним елементом цієї логіки є формування показників прогресу. На основі даних про завдання, подання робіт, оцінки та прострочені задачі система може відображати для студента частку виконаних робіт, середній бал і поточний стан його навчальної активності. Для викладача можуть формуватися узагальнені показники за курсом, зокрема кількість завдань, кількість отриманих і перевірених робіт, середній результат студентів і список робіт, які потребують уваги. Такий підхід розширює систему від звичайного електронного журналу до інструмента підтримки організації навчальної діяльності (рис. 2.1.):

У межах розроблюваної системи передбачено три основні ролі користувачів: адміністратор, викладач і студент. Кожна роль має власне призначення та обсяг доступних дій. Розмежування прав є важливим, оскільки воно забезпечує впорядковану організацію роботи системи та запобігає виконанню користувачами операцій, які не відповідають їхнім функціям. Основні можливості ролей наведено в таблиці 2.1.

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 17
Зм.	Арк.	№ докум.	Підпис	Дата		

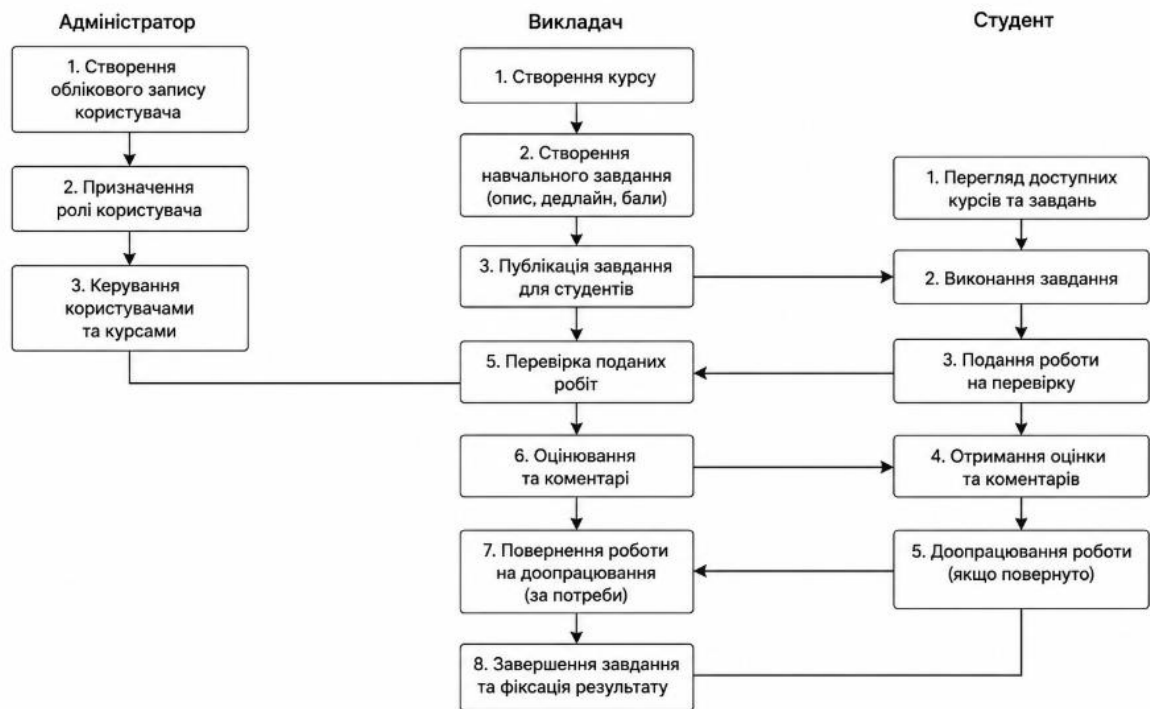


Рисунок 2.1 – Загальна логіка роботи з навчальним завданням у системі

Таблиця 2.1 – Ролі користувачів і їхні основні функції в системі

Роль користувача	Основні функції
Адміністратор	Створення користувачів, призначення ролей, перегляд загальної структури облікових записів
Викладач	Створення курсів і завдань, керування дедлайнами, перегляд поданих робіт, оцінювання, повернення на доопрацювання, формування звітів
Студент	Перегляд курсів і завдань, подання робіт, отримання оцінок і коментарів, перегляд сповіщень та власного прогресу

З таблиці 2.1, кожна роль у системі має власну зону відповідальності та виконує чітко визначені дії. Адміністратор відповідає за організаційний рівень роботи системи, оскільки саме він створює облікові записи та визначає

належність користувача до певної ролі. Це забезпечує правильний розподіл доступу до функцій системи й унеможлиблює виконання дій, які не відповідають повноваженням конкретного користувача.

Викладач є основним учасником формування навчального процесу в межах системи. Він створює курси, визначає зміст і параметри завдань, установлює дедлайни, перевіряє подані роботи та фіксує результати оцінювання. Крім цього, викладач може повертати роботу на доопрацювання, що дозволяє підтримати не лише контрольну, а й навчальну функцію системи. Завдяки цьому процес оцінювання стає більш прозорим, а студент отримує можливість зрозуміти недоліки виконаної роботи та покращити результат.

Студент, у свою чергу, взаємодіє із системою на рівні виконання навчальних дій. Він переглядає доступні курси й завдання, орієнтується у строках здачі, завантажує результати роботи, отримує оцінки, коментарі та повідомлення про зміни статусу. Наявність особистого кабінету й упорядкованого переліку завдань дає змогу студенту краще контролювати власне навчальне навантаження та бачити поточний прогрес.

Дії адміністратора створюють організаційну основу, викладач наповнює систему навчальним змістом і керує процесом перевірки, а студент виконує завдання та отримує результат. Усі ці дії пов'язані між собою, тому система повинна не просто зберігати окремі записи, а підтримувати узгоджений рух інформації між учасниками освітнього процесу. Логіка роботи з навчальними завданнями в розроблюваній системі охоплює повний послідовний цикл: створення курсу, формування завдання, його виконання студентом, подання роботи, перевірку, оцінювання, надсилання сповіщень і відображення оновленого прогресу.

Отже, така організація дозволяє чітко впорядкувати дії користувачів, зафіксувати зміни стану кожної роботи та забезпечити прозорий контроль навчальної діяльності. Саме ця модель визначає загальну логіку системи й дає

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

змогу надалі обґрунтовано проєктувати її функціональні модулі та інформаційні зв'язки.

## 2.2 Проєктування функціональної структури інформаційної системи

Інформаційна система управління навчальними завданнями студентів охоплює кілька різних процесів, які в реальному освітньому середовищі часто виконуються окремо. Викладач створює курси й завдання, студент стежить за строками виконання, готує роботу та надсилає її на перевірку, після чого відбувається оцінювання, повернення результатів і формування загальної успішності. Якщо ці дії не об'єднані в межах одного середовища, частина інформації зберігається в листуванні, частина в окремих файлах, а частина залишається лише у викладача або студента. Що ускладнює контроль за ходом навчальної роботи й робить сам процес менш прозорим. Під час проєктування системи важливо згрупувати їх відповідно до логіки навчального процесу.

Тому функціональна структура формується навколо основних дій, які постійно повторюються в роботі з навчальними завданнями: створення та ведення курсів, підготовка завдань, приймання робіт, перевірка результатів, контроль термінів, інформування користувачів і накопичення даних для подальшого аналізу. Такий поділ дозволяє уникнути змішування різних функцій і зробити систему зрозумілішою як на етапі проєктування, так і під час її практичного використання. Функціональну структуру зображено на рисунку 2.2.

Від ролі користувача залежить, які дані бачить людина і які дії може виконувати. Адміністратор підтримує загальну користувацьку структуру, створює облікові записи та призначає ролі. Викладач працює з навчальним змістом, тобто створює курси, готує завдання, перевіряє роботи та переглядає результати студентів. Студент отримує доступ до власних курсів, завдань, дедлайнів, оцінок і сповіщень. Такий розподіл робить функціонування системи

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		



матеріал для перевірки. Коли студент надсилає виконану роботу, система має зафіксувати не лише сам факт подання, а й контекст цієї дії. Важливо зберегти, до якого завдання належить робота, хто її подав, коли саме це відбулося і чи не був перевищений установлений строк. Ці дані впливають на подальший статус роботи, якщо подання виконано своєчасно, воно переходить до перевірки у звичайному порядку. Якщо строк минув, система відзначає це окремо, що дає викладачу можливість врахувати порушення дедлайну під час оцінювання.

Перевірка робіт пов'язана не лише з виставленням бала, адже у процесі часто виникає потреба залишити пояснення, вказати на помилки або повернути матеріал на доопрацювання. Тому функції оцінювання повинні охоплювати не тільки фінальний результат, а й коментар викладача, зміну статусу роботи та можливість повторного подання, якщо така дія дозволена умовами завдання. У такому випадку система підтримує повніший освітній сценарій, а не обмежується простим механічним обліком оцінок.

Дедлайн потрібен не тільки для того, щоб показати дату в картці роботи, він впливає на статус подання, на нагадування студенту і на виділення завдань, які потребують особливої уваги. Завдяки цьому система допомагає не лише зберігати інформацію, а й орієнтувати користувачів у поточній навчальній ситуації, а сповіщення роблять взаємодію менш пасивною. Студент може своєчасно дізнатися про нове завдання, наближення строку виконання, оцінку або повернення роботи.

Після накопичення даних про курси, завдання, подання й оцінки система може формувати узагальнену інформацію про хід навчальної діяльності. Для студента це може бути прогрес за виконаними завданнями, середній бал і кількість прострочених здач. Для викладача є більш загальні показники, зокрема кількість отриманих робіт, частка перевірених подань, загальна активність студентів і підготовка підсумкової відомості. Завдяки цьому інформаційна система виконує не лише облікову, а й аналітичну функцію. Основні функціональні частини системи та їх призначення подано в таблиці 2.2.

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						22
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.2 – Основні функціональні частини інформаційної системи

Функціональна частина	Призначення
Користувачі та ролі	Розмежування доступу для адміністратора, викладача і студента
Курси	Об'єднання викладача, студентів і навчальних матеріалів у межах дисципліни
Завдання	Формування робіт, які потрібно виконати, із зазначенням строків і вимог
Подання робіт	Приймання результатів від студентів і фіксація стану здачі
Оцінювання	Перевірка роботи, виставлення бала, коментарів і повернення на доопрацювання
Сповіщення	Інформування користувачів про важливі події
Контроль дедлайнів	Визначення прострочених подань і підготовка
Аналітика та звітність	Відображення прогресу, статистики й підготовка звітних даних

Функціональні частини системи утворюють єдину логіку роботи, у якій кожен наступний процес спирається на результати попереднього. Після створення та публікації завдання воно стає доступним студенту, а система може додатково інформувати його про встановлений строк виконання. Коли робота подається на перевірку, відомості про неї передаються до оцінювання, де викладач переглядає результат, виставляє бал, залишає коментар або повертає матеріал на доопрацювання. Зміна статусу роботи впливає одразу на кілька напрямів роботи системи, на формування сповіщення для користувача, оновлення навчального прогресу для студента, а дані враховуються під час підготовки аналітичної та звітної інформації. Взаємодії основних функціональних частин наведено на рисунку 2.3.



Отже, функціональна структура системи охоплює основні процеси, необхідні для організації роботи з навчальними завданнями студентів. Вона поєднує керування користувачами, курсами, завданнями, поданнями, оцінюванням, дедлайнами, сповіщеннями та звітністю в єдину логічну модель. Такий підхід дозволяє впорядкувати навчальну взаємодію, чітко розмежувати призначення окремих частин системи та забезпечити послідовний рух інформації від створення завдання до фіксації підсумкового результату.

### 2.3 Проектування інформаційної моделі обліку навчальних завдань і результатів їх виконання

Під час роботи інформаційної системи одні дані з'являються раніше, а інші виникають як наслідок уже виконаних дій, адже спочатку формується навчальний курс, після цього в ньому з'являються завдання, далі фіксуються подання робіт, оцінки, зміни статусів і повідомлення для користувачів. Звітність та показники прогресу не створюються окремо, а обчислюються на основі всієї накопиченої інформації. Тому при проектуванні важливо визначити не лише перелік даних, а й порядок їх появи, залежності між ними та роль кожної групи відомостей у загальній структурі системи.

Інформаційна модель визначає порядок формування та взаємозв'язку даних, що супроводжують роботу з навчальними завданнями. Вона поєднує відомості про користувачів, курси, завдання, подані роботи, результати перевірки, сповіщення та підсумкові показники в єдину логічну структуру. Завдяки цьому кожен етап навчальної діяльності зберігає зв'язок із попередніми подіями, а накопичені дані можуть коректно використовуватися для контролю виконання завдань, відображення прогресу студентів і формування звітної інформації (таблиця 2.3).

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 25
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.3 – Групи даних інформаційної моделі системи та їх використання

Групи даних	Основні інформаційні об'єкти	Значення для роботи системи
Організаційні дані	Користувач, роль, курс, участь у курсі	Визначають, хто працює в системі та з якими курсами пов'язаний
Навчальні дані	Завдання, строк виконання, максимальний бал	Описують зміст роботи, яку потрібно виконати студенту
Дані про виконання	Подання роботи, дата здачі, стан подання	Фіксують результат дій студента щодо конкретного завдання
Дані перевірки	Оцінка, коментар, рішення про доопрацювання	Відображають результат опрацювання роботи викладачем
Узагальнені дані	Сповіщення, прогрес, звітність	Формуються на основі змін у системі та накопичених результатів

Такий поділ дає змогу чітко відрізнити дані, що задають структуру навчального середовища, від даних, які з'являються вже під час виконання конкретних робіт. Користувач і курс описують стабільні відомості, без яких неможливо визначити, хто навчається, хто викладає і до якої дисципліни належить певне завдання. Натомість подання роботи, зміна статусу, оцінка або сповіщення відображають перебіг навчальної діяльності в часі та залежать від дій учасників системи.

Ключовим зв'язком організаційної частини моделі є участь студента в курсі, адже один студент може навчатися на кількох курсах, а один курс може містити значну кількість студентів, тому цей зв'язок не можна зводити лише до



Для однієї навчальної роботи може бути створено багато подань, по одному від кожного студента, а за потреби ще й повторні варіанти після повернення на доопрацювання. У цьому об'єкті важливо зафіксувати не лише завантажений файл, а й дату надсилання, факт дотримання або порушення дедлайну, а також актуальний стан роботи. Завдяки цьому система здатна показати не просто кінцевий результат, а послідовність змін, які відбувалися із завданням після його виконання студентом. Оцінювання пов'язується саме з поданням, а не із завданням загалом.

Така логіка є принциповою, оскільки викладач перевіряє конкретну роботу окремого студента, а не сам шаблон завдання. Бал, коментар і рішення щодо повернення на доопрацювання мають належати до відповідного подання. Це дозволяє коректно відобразити результати кожного студента, зберегти зворотний зв'язок і не змішувати між собою дані різних учасників одного курсу. На основі змін, що відбуваються із завданнями, поданнями та оцінками, з'являються похідні дані. Сповіднення інформують студента про нову роботу, наближення дедлайну, оцінювання чи повернення на доопрацювання. Прогрес формується через співвідношення виконаних і невиконаних завдань, отримані бали та кількість прострочених здач. Звітність для викладача узагальнює відомості за курсом, дає змогу бачити стан перевірки робіт і підсумкові результати студентів.

Запропонована модель дає змогу відстежувати не тільки фінальний результат, а й усю історію роботи із завданням. Для конкретного студента можна визначити курс, до якого належить робота, встановлений строк здачі, факт подання, ознаку запізнення, рішення викладача та вплив цієї роботи на загальний прогрес. Саме така зв'язаність даних потрібна для коректного відображення статусів, формування особистого кабінету студента, підготовки відомості викладача та подальшої аналітики.

Отже, інформаційна модель системи управління навчальними завданнями студентів визначає не лише склад даних, а й порядок їх взаємозв'язку. Вона

					КвРІСТ 220180.22.01.06 ПЗ	Арк.
						28
Зм.	Арк.	№ докум.	Підпис	Дата		

об'єднує організацію курсів, виконання завдань, подання робіт, перевірку результатів, формування сповіщень і підсумкових показників в одну узгоджену структуру. Завдяки цьому система може коректно зберігати історію навчальної діяльності, точно відобразити стан кожної роботи та надавати користувачам повну й непротивічну інформацію про перебіг навчального процесу.

## 2.4 Проектування алгоритмів обробки навчальних подій у системі

Робота інформаційної системи управління навчальними завданнями визначається не лише складом її об'єктів і модулів, а й правилами, за якими обробляються події, що виникають під час навчального процесу. Найважливішими серед них є подання роботи студентом, перевірка результату викладачем, зміна статусу завдання, повернення матеріалу на доопрацювання та формування сповіщень. Саме ці дії створюють основний рух інформації в системі та впливають на відображення дедлайнів, оцінок, прогресу та звітності.

Під час проектування алгоритмів важливо врахувати, що одна й та сама навчальна подія може мати різні наслідки залежно від умов, за яких вона відбулася. Наприклад, подання роботи до завершення дедлайну й після його спливу має фіксуватися по-різному, так само перевірка викладачем може завершитися виставленням оцінки або поверненням завдання на повторне виконання. Тому алгоритмічна логіка системи повинна не просто послідовно виконувати окремі кроки, а й коректно реагувати на зміну станів, дотримуючись визначених правил навчального процесу.

Одним із головних сценаріїв є подання роботи студентом, спочатку користувач переходить до навчального завдання та переглядає його умови: опис, строк виконання, максимальний бал і вимоги до матеріалів, які можна надсилати. Якщо завдання вже закрито або студент не має доступу до курсу, тому подання не повинно створюватися. Якщо робота доступна, система переходить до перевірки файлів, які студент додає до відповіді. Після чого враховуються

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						29
Зм.	Арк.	№ докум.	Підпис	Дата		

дозволені типи документів і допустимий розмір вкладень, оскільки результат має відповідати правилам, установленим для конкретного завдання.

Після успішної перевірки матеріалів система фіксує дату й час надсилання роботи. Це значення порівнюється з дедлайном, вказаним у завданні. Якщо відповідь подано вчасно, їй призначається стандартний статус прийнятого подання. Якщо строк уже минув, робота все одно може бути зареєстрована, але з позначкою про запізнення, що надалі враховується викладачем під час перевірки. У разі повторної здачі необхідно також врахувати, чи дозволено оновлене подання після доопрацювання. Якщо така можливість передбачена, новий результат пов'язується з попередньою історією виконання, не руйнуючи вже зафіксованих даних. Після завершення подання система оновлює поточний стан завдання для студента, передає інформацію про нову роботу викладачу та враховує факт здачі під час формування показників прогресу. Таким чином, один алгоритм охоплює одразу кілька аспектів роботи системи, від контролю доступу, перевірки результатів, дотримання дедлайну, зміни статусу, виникнення сповіщення та оновлення аналітичних даних (рисунок 2.5).

Другим основним процесом є перевірка та оцінювання студентської роботи викладачем. Після появи нового подання викладач відкриває перелік робіт, які очікують опрацювання, обирає потрібний результат і переглядає надіслані матеріали. Далі він приймає рішення щодо якості виконання. Якщо робота відповідає вимогам, викладач виставляє бал і за потреби додає коментар. Після цього система змінює статус результату на оцінений, фіксує підсумкові відомості та передає студенту повідомлення про перевірку. Інший сценарій виникає тоді, коли робота не може бути прийнята остаточно. У такому випадку викладач залишає коментар із поясненням і повертає результат на доопрацювання. Для системи це означає не завершення процесу, а перехід до нового циклу взаємодії зі студентом. Статус подання змінюється, а студент отримує сповіщення, після чого завдання знову відображається як таке, що потребує уваги.

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						30
Зм.	Арк.	№ докум.	Підпис	Дата		

Якщо правила дозволяють повторну здачу, користувач може сформувати нове подання, після чого робота ще раз надходить на перевірку. У процесі оцінювання система повинна не лише зберегти результат рішення викладача, а й оновити пов'язані інформаційні показники. Виставлений бал впливає на середнє значення успішності студента, на прогрес за курсом і на підсумкові звіти викладача. Повернення на доопрацювання не збільшує кількість завершених робіт, але змінює поточний стан завдання й формує нову подію для повідомлення користувача. Завдяки чому оцінювання розглядається не як окрема дія, а як центральний етап, що визначає подальший рух навчальної інформації. Алгоритм перевірки та оцінювання роботи викладачем зображено на рисунку 2.6.

Алгоритми охоплюють два центральні процеси, від яких залежить коректність роботи всієї системи. Подання роботи визначає, як саме фіксується навчальна активність студента, а перевірка й оцінювання встановлюють підсумковий стан цієї діяльності. Разом вони забезпечують керований перехід завдання від етапу виконання до етапу результату, підтримують контроль строків, дозволяють враховувати повторні спроби й створюють підставу для формування повідомлень та аналітичних показників.

Отже, алгоритмічна логіка системи повинна забезпечити не просто механічне виконання дій, а правильний порядок обробки кожної навчальної події. Завдяки цьому стає можливим надійно фіксувати стан завдань, відрізнити своєчасні та прострочені подання, підтримувати повторне доопрацювання робіт, передавати користувачам актуальні повідомлення й формувати достовірну картину навчального прогресу. Крім того, чітко налаштовані алгоритми дозволяють мінімізувати суб'єктивний фактор під час технічної перевірки робіт, автоматизуючи рутинні операції та вивільняючи час для якісного фідбеку. Це створює стійке цифрове середовище, де кожен крок студента від завантаження файлу до отримання фінального балу є задокументованим, прозорим та логічно завершеним.

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 31
Зм.	Арк.	№ докум.	Підпис	Дата		

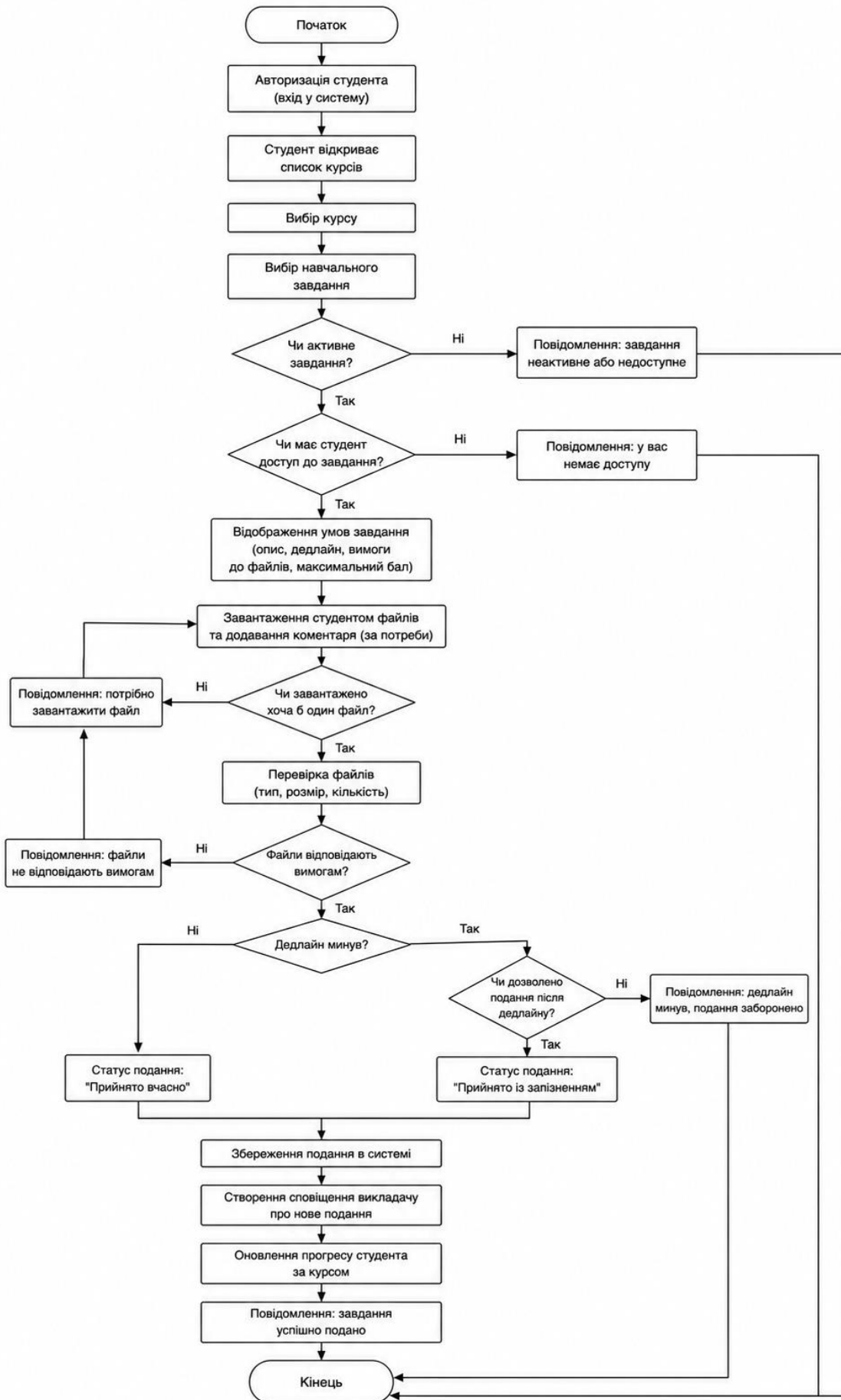


Рисунок 2.5 – Алгоритм подання студентської роботи

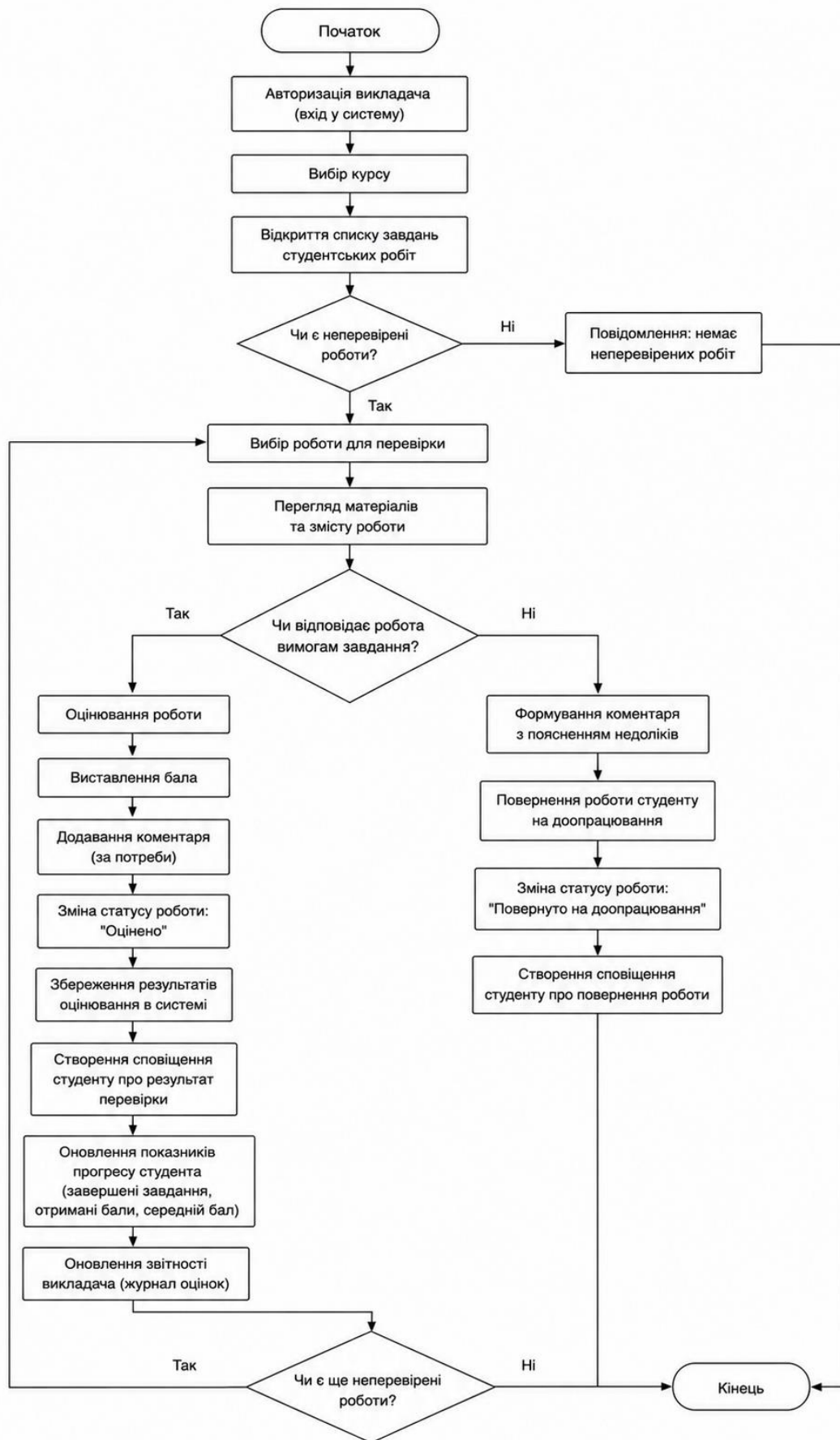


Рисунок 2.6 – Алгоритм перевірки та оцінювання студентської роботи викладачем

## 2.5 Висновки до другого розділу

У другому розділі було спроектовано логіку інформаційної системи управління навчальними завданнями студентів. Визначено послідовність основних дій, які відображають реальний порядок навчальної роботи, від створення курсу, формування завдання, його виконання студентом, подання результату, перевірки викладачем, оцінювання, надсилання сповіщень і оновлення показників прогресу.

Було виділено частини, що відповідають за користувачів і ролі, курси, навчальні завдання, подання робіт, оцінювання, контроль дедлайнів, сповіщення та звітність. Такий поділ дозволяє чітко розмежувати призначення кожної складової й водночас зберегти їхню взаємодію в одному навчальному циклі. Також розроблено інформаційну модель, яка описує порядок зв'язку між основними даними системи. Вона враховує участь студентів у курсах, належність завдань до дисциплін, появу подань, результати оцінювання, формування повідомлень і підсумкових показників. Завдяки цьому дані не розглядаються окремими фрагментами, а утворюють цілісну картину роботи із навчальними завданнями.

Для головних процесів сформовано алгоритмічну логіку та описано порядок подання студентської роботи з перевіркою доступності завдання, контролем файлів і визначенням статусу задачі. Також спроектовано алгоритм перевірки та оцінювання, який враховує виставлення бала, повернення на доопрацювання, зміну статусів, сповіщення студентів і оновлення звітних даних.

Отже, спроектована система охоплює не лише облік навчальних завдань, а повний порядок роботи з ними від появи завдання до фіксації кінцевого результату. Такий підхід відповідає темі дипломної роботи й дозволяє розглядати інформаційну систему як цілісне середовище для організації, контролю та аналізу навчальної діяльності студентів.

					КвРІСТ 220180.22.01.06 ПЗ	Арк.
						34
Зм.	Арк.	№ докум.	Підпис	Дата		

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ПЕРЕВІРКА ПРАЦЕЗДАТНОСТІ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ НАВЧАЛЬНИМИ ЗАВДАННЯМИ СТУДЕНТІВ

#### 3.1 Реалізація серверної частини та механізмів доступу користувачів

Практична реалізація інформаційної системи управління навчальними завданнями побудована навколо серверного застосунку, який приймає запити від інтерфейсу, перевіряє їх коректність і виконує дії відповідно до ролі користувача. На цьому рівні визначається, хто може створювати курс, публікувати завдання, надсилати роботу, виставляти оцінку або керувати обліковими записами. Такий підхід дозволив не переносити важливі правила роботи системи до клієнтської частини, а зосередити їх у backend, де вони виконуються незалежно від того, з якої сторінки або форми надійшов запит.

У результаті всі основні процеси, описані на етапі проектування, отримали програмну реалізацію в єдиному керованому середовищі. Сервер створено мовою Python із використанням FastAPI. Для запуску застосунку застосовано Uvicorn, для роботи з базою даних використано SQLAlchemy, а для перевірки переданих структур даних залучено Pydantic. Внутрішня будова проекту розділена на кілька рівнів. А у блоці api/v1 зібрані маршрути для авторизації, курсів, завдань, подань, оцінювання, сповіщень, користувачів і звітності.

Сервісний шар відповідає за прикладну логіку, а саме перевіряє доступ до курсу, визначає можливість здачі завдання, обробляє оцінювання та формує показники для звітів. Моделі описують структуру даних, схеми задають формат обміну з frontend, а окремі модулі відповідають за файлове сховище, WebSocket-з'єднання та фонові нагадування про дедлайни. Така організація зробила серверну частину не громіздким набором окремих функцій, а впорядкованою програмною основою системи (рисунок 3.1).

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 35
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.1 – Структура серверної частини інформаційної системи

Під час входу користувач надсилає email і пароль, після чого сервер перевіряє облікові дані, формує токен доступу та передає дані про поточний профіль. Паролі не зберігаються у відкритому вигляді, оскільки перед записом до бази вони проходять хешування. У проєкті передбачено також refresh token у захищеному cookie, що дозволяє підтримувати користувацьку сесію без повторного введення пароля після кожної короткочасної втрати access token. Така схема є достатньою для навчальної інформаційної системи, де потрібно поєднати зрозумілий механізм входу з коректним захистом закритих маршрутів.

У системі використано три ролі: студент, викладач і адміністратор. Права для кожної ролі не задаються лише в інтерфейсі, а перевіряються безпосередньо на сервері через залежності FastAPI. Це означає, що навіть у разі прямого звернення до API користувач не зможе виконати дію, яка не відповідає його ролі. Адміністратор має доступ до створення користувачів і перегляду їхнього переліку, викладач керує власними курсами, завданнями й результатами подань,

а студент працює лише з доступними йому курсами та власними роботами. Ця модель повторює реальний розподіл відповідальності в навчальному процесі й водночас унеможлиблює змішування адміністративних, викладацьких і студентських функцій.

Для користувачів список курсів формується по-різному: адміністратор бачить усі записи, викладач лише власні курси, а студент лише ті, до яких його було додано через enrollment. Створення завдання також виконується не напряму, а через перевірку належності курсу конкретному викладачу. Новий запис спочатку отримує статус draft, тому студент його ще не бачить. Після публікації статус змінюється на published, а після завершення періоду приймання робіт завдання може бути закрите. Такий порядок реалізує зрозумілий цикл підготовки, відкриття та завершення роботи з навчальним матеріалом.

Також реалізовано правила подання студентських робіт, адже цей процес пов'язує навчальне завдання з фактичним результатом студента. Перед створенням подання сервер перевіряє, чи завдання опубліковане, чи студент зарахований на відповідний курс, чи дозволене повторне надсилання, а також чи не порушено встановлені обмеження щодо формату та розміру файлів. Якщо роботу подано до дедлайну, вона отримує статус submitted. Якщо строк уже минув, система фіксує статус late, не втрачаючи сам результат. Якщо викладач повернув попередню спробу на доопрацювання і в налаштуваннях завдання дозволено повторну здачу, студент може завантажити оновлену версію. Саме така логіка забезпечує відповідність реалізації тому навчальному циклу, який було спроектовано в другому розділі. Приймання файлів винесено в окремий механізм локального сховища.

Документи зберігаються в директорії uploaded\_files, а база даних містить відомості про належність файлу до конкретного подання. Це дозволяє не перевантажувати структуру бази великими двійковими об'єктами й водночас зберігати точний зв'язок між студентом, завданням, поданням і прикріпленим матеріалом. Окремий сервіс сховища також робить роботу з файлами більш

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 37
Зм.	Арк.	№ докум.	Підпис	Дата		

керованою, оскільки правила перевірки вкладень не змішуються з маршрутами або логікою оцінювання. Серверна частина підтримує й оперативне інформування користувачів. Сповіщення зберігаються в базі, а для їх передавання без ручного оновлення сторінки використано WebSocket. Через нього до інтерфейсу можуть надходити повідомлення про виставлення оцінки, повернення роботи на доопрацювання або інші важливі події.

Також, у системі підключено APScheduler, який виконує фонові перевірки строків і формує нагадування про наближення дедлайну. Що важливо для практичного використання розробки, оскільки сервер не лише реагує на вже виконані дії, а й підтримує своєчасну комунікацію між системою та користувачем.

Отже, серверну частину реалізовано як основу всієї інформаційної системи, що поєднує автентифікацію, рольовий доступ, роботу з навчальними об'єктами, контроль подань, файлові операції та механізми сповіщення. Що задає правила, за якими система зберігає коректний перебіг навчального процесу. Завдяки цьому реалізований застосунок підтримує повний ланцюг дій від входу користувача до створення курсу, призначення завдання, здачі роботи та її подальшого опрацювання.

### 3.2 Реалізація бази даних і збереження навчальної інформації

База даних у розробленій інформаційній системі реалізована як основа для збереження всіх відомостей, що виникають у процесі роботи студентів, викладачів і адміністратора. Для збереження даних використано локальну базу SQLite, а робота з нею організована засобами SQLAlchemy. Це дало змогу створити повноцінне сховище без окремого серверного налаштування СУБД, зберігши при цьому чітку структуру зв'язків між об'єктами системи. У центральній частині реалізованої структури розміщено користувачів, курси й записи студентів на навчальні дисципліни. Таблиця користувачів зберігає

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 38
Зм.	Арк.	№ докум.	Підпис	Дата		

ідентифікатор, електронну адресу, ім'я, хешований пароль і роль, яка визначає статус користувача в системі. Курс пов'язується з викладачем, що дозволяє відрізнити навчальні простори різних авторів і формувати перелік дисциплін залежно від облікового запису.

Для фіксації участі студентів у курсах використано окрему сутність enrollment, завдяки якій один студент може навчатися на кількох дисциплінах, а один курс може містити багато учасників. У структурі також передбачено студентські групи, що створює можливість організовувати підключення користувачів не лише індивідуально, а й через навчальні об'єднання. На наступному рівні бази даних розміщено завдання. Кожне завдання належить до конкретного курсу й містить параметри, потрібні для фактичної організації навчальної роботи: назву, опис, стан публікації, тип, дедлайн, максимальний бал, допустимий розмір файла, дозволені формати вкладень і ознаку повторної здачі.

Таким чином, у базі зберігається не лише текст навчальної роботи, а повний набір умов, які далі використовуються серверною логікою під час доступу студента до завдання й перевірки можливості подання результату. Саме ця структура дає змогу підтримувати стани draft, published і closed, описані в попередньому підрозділі на рівні backend-реалізації. Загальну схему зв'язків зображено на рисунку 3.2.

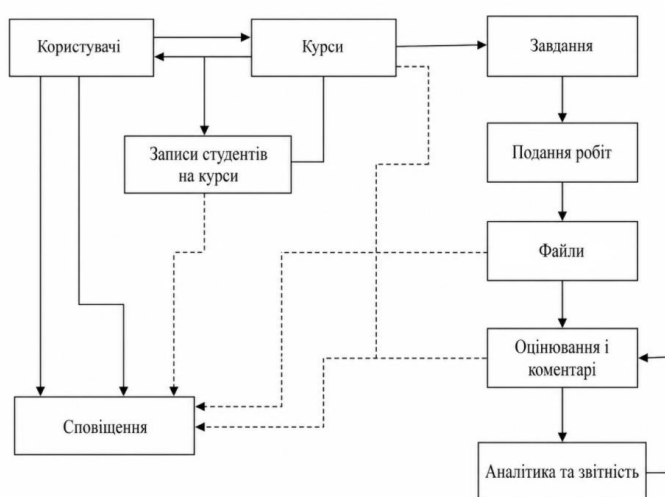


Рисунок. 3.2 – Схема зв'язків між основними сутностями бази даних інформаційної системи

Робочий ланцюг у базі даних формується через зв'язок завдання, студентського подання та результату оцінювання. Для кожної виконаної роботи створюється запис у таблиці submissions, де фіксуються завдання, студент, статус подання, час надсилання та номер спроби. Це дозволяє зберігати не лише факт наявності файлу, а повний стан виконання роботи, включно з випадками запізнення або повторного подання після доопрацювання.

До подання можуть бути прикріплені окремі файли, для яких у базі зберігаються службові відомості, зокрема назва, шлях до локального сховища, тип вмісту та розмір. Самі бінарні файли не записуються безпосередньо в базу, що зменшує її навантаження й дозволяє працювати з вкладеннями більш раціонально. Оцінювання подання реалізовано через окрему сутність grade. Вона пов'язується з конкретною студентською роботою та містить набраний бал, максимальне значення, текстовий відгук, ознаку повернення на доопрацювання й час перевірки.

Такий спосіб збереження добре відповідає реальній роботі викладача, оскільки оцінюється не абстрактне завдання, а конкретний результат окремого студента. Для розширеного оцінювання в структурі передбачено рубрики, критерії та оцінки за цими критеріями, що дає можливість деталізувати перевірку й пов'язати підсумковий бал із більш конкретними складовими виконання. Додаткові коментарі викладача також виділені в окремий рівень, що зберігає історію пояснень і не зводить весь feedback до одного короткого поля.

Реалізована схема даних охоплює не тільки основні навчальні об'єкти, а й інформацію, яка потрібна для аналітики та підтримки користувача. На основі зв'язків між курсами, завданнями, поданнями й оцінками система може визначати кількість опублікованих робіт, відсоток виконання, середній бал і стан перевірки матеріалів курсу. Це показує, що база даних використовується не лише як пасивне місце зберігання записів, а як основа для побудови аналітичних представлень у клієнтській частині системи.

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 40
Зм.	Арк.	№ докум.	Підпис	Дата		

Для забезпечення стабільної роботи найчастіше використовуваних операцій у структурі передбачено індекси й обмеження унікальності. Зокрема, індексуються адреси користувачів, ролі, ідентифікатори викладачів у курсах, статуси й дедлайни завдань, студенти й стани подань. Окреме обмеження за парою `course_id` і `student_id` не допускає повторного запису одного студента на той самий курс. Подібні рішення є важливими не лише з погляду продуктивності, а й для збереження цілісності навчальної інформації, оскільки вони запобігають появі дубльованих зв'язків і спрощують подальший пошук даних під час роботи інтерфейсу.

У підсумку база даних інформаційної системи реалізована як пов'язана структура, що відтворює повний цикл роботи з навчальними завданнями від створення користувача й курсу до подання роботи, її перевірки, виставлення оцінки та формування аналітичних показників. Використання SQLite і SQLAlchemy дозволило поєднати простоту локального запуску з достатньо гнучкою моделлю даних, а винесення файлів у окреме сховище зберегло базу компактною та придатною до подальшого розвитку. Така реалізація створює надійну основу для клієнтського інтерфейсу, який використовує ці дані для відображення курсів, завдань, подань, журналу оцінок, прогресу та сповіщень.

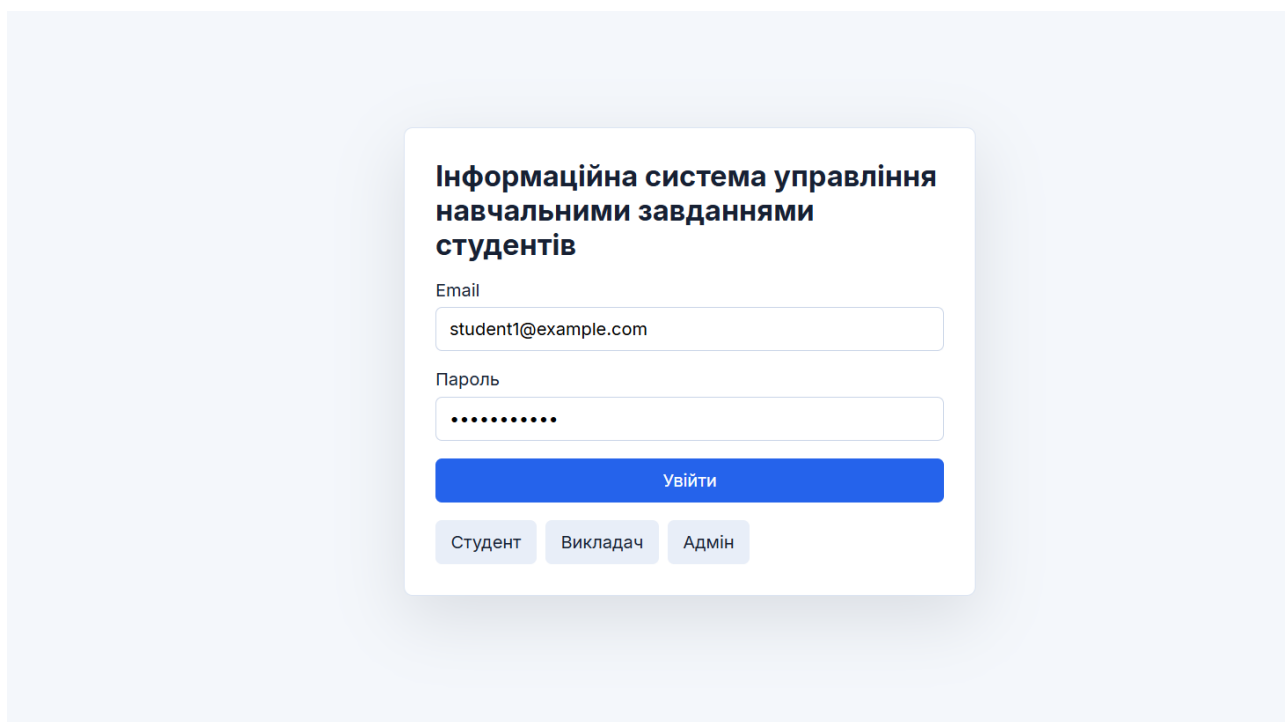
### 3.3 Реалізація клієнтського інтерфейсу інформаційної системи

Клієнтську частину інформаційної системи реалізовано як вебзастосунок на основі React 18 і TypeScript. Її призначення полягає не лише у відображенні окремих даних, а в організації повного робочого простору для трьох категорій користувачів: студента, викладача й адміністратора. Після входу до системи користувач отримує інтерфейс, склад якого змінюється відповідно до його ролі.

Такий підхід дозволив не створювати ізольовані програми для кожного учасника навчального процесу, а реалізувати єдину клієнтську частину з адаптованим набором доступних дій, сторінок і інформаційних блоків. Для

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

зв'язку з backend використовується окремий API-клієнт, який додає JWT-токен до захищених запитів, а для оперативного отримання сповіщень підключається WebSocket-канал. Початкова взаємодія користувача із системою відбувається через сторінку автентифікації, що містить поля для електронної пошти та пароля, а після успішного входу переводить користувача до відповідного робочого середовища. У конфігурації передбачено тестові облікові записи адміністратора, викладача і студента, що дозволяє показати різницю між режимами роботи системи під час перевірки функціональності. Сторінка входу є важливою не лише як технічний елемент авторизації, а і як межа доступу до закритих частин інформаційної системи, оскільки без підтвердженої сесії користувач не може перейти до курсів, завдань, подань чи аналітичних сторінок (рисунок 3.3).



The image shows a login form titled "Інформаційна система управління навчальними завданнями студентів". It contains two input fields: "Email" with the value "student1@example.com" and "Пароль" (Password) with masked characters. Below the fields is a blue "Увійти" (Login) button. At the bottom, there are three buttons for user roles: "Студент" (Student), "Викладач" (Teacher), and "Адмін" (Admin).

Рисунок. 3.3 – Сторінка входу до інформаційної системи управління навчальними завданнями студентів

Після автентифікації відкривається головна сторінка Dashboard, яка виконує роль центральної оболонки інтерфейсу. Структура побудована на поєднанні бічної навігаційної панелі, верхнього інформаційного блоку та

основної робочої області. У лівій частині користувач обирає курс або переходить до потрібного розділу, тоді як центральна частина екрану відображає поточний зміст роботи. Для студента це можуть бути завдання курсу, форма подання файлів, календар дедлайнів, дошка станів виконання, журнал оцінок і графіки прогресу. Для викладача в тому самому макеті відкриваються інструменти створення курсів і завдань, перегляд студентських подань, оцінювання за рубрикою, звітність та контроль активності в курсі. Завдяки такій організації інтерфейс зберігає однакову логіку навігації, але не навантажує кожну роль функціями, які їй не потрібні (рисунок 3.4).

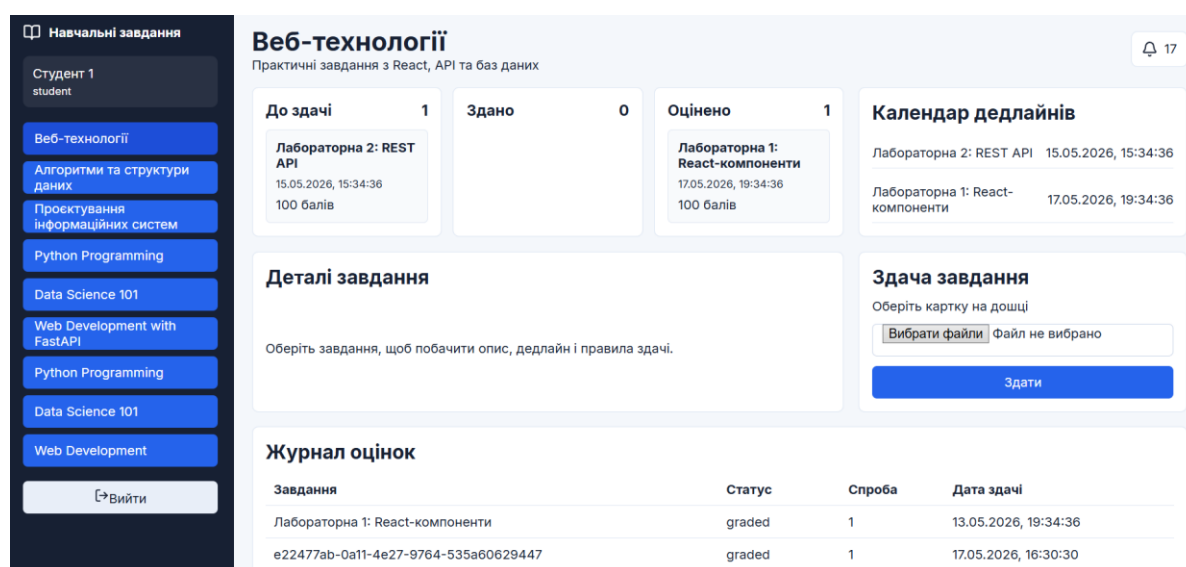


Рисунок. 3.4 – Робоче середовище студента в інформаційній системі

Студентський інтерфейс побудовано навколо вибраного курсу й актуальних навчальних робіт. Компонент CourseView виводить перелік опублікованих завдань, показує дедлайн, максимальний бал і статус власного подання, якщо воно вже існує. Після вибору певного завдання відкривається TaskDetail, де подано повний опис, дозволені формати файлів, вимоги до розміру вкладення та інші параметри, необхідні перед виконанням роботи.

Для надсилання результату використовується окрема форма SubmitTask, що підтримує завантаження файлів і перевіряє відповідність вкладень правилам завдання ще до відправлення запиту на сервер. Після успішної здачі список подань оновлюється, а студент бачить новий стан роботи в особистому інтерфейсі, що продовжує алгоритм подання студентської роботи, описаний у попередньому розділі, але вже у вигляді завершеного програмного сценарію.

Крім базової роботи із завданнями, студентський режим містить елементи, які допомагають контролювати навчальне навантаження. Календар дедлайнів виділяє дні, на які припадають строки виконання, а дошка Kanban розподіляє роботи за станами: ще не розпочаті, у процесі, подані та оцінені. Окремий блок прогресу показує кількість виконаних завдань, середній бал, число прострочених подань і позицію студента в групі, якщо така аналітика доступна для певного курсу. Ці показники доповнюються графіками, які візуалізують поточний стан успішності. У сукупності такі елементи переводять інтерфейс від простого списку навчальних робіт до персонального робочого простору, у якому студент бачить не тільки окрему задачу, а і власну динаміку виконання курсу.

Режим викладача реалізований у тій самій клієнтській структурі, викладач може створити курс, сформулювати нове завдання, задати його тип, дедлайн, максимальний бал, допустимі формати файлів і можливість повторної здачі. Окремий редактор завдання забезпечує підготовку навчального матеріалу до публікації, після чого робота стає видимою студентам. Для перевірки результатів використовується компонент SubmissionReview, який показує подання за завданнями та надає доступ до завантажених файлів. Оцінювання реалізовано через форму рубрики, де викладач може виставити бали, надати відгук і за потреби повернути роботу на доопрацювання. Це дає змогу відтворити не лише факт виставлення оцінки, а повний сценарій педагогічного опрацювання студентського результату (рисунок 3.5).

Для узагальнення результатів навчальної діяльності у frontend передбачено журнал оцінок, блок прогресу й панель звітності. Gradebook відображає

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						44
Зм.	Арк.	№ докум.	Підпис	Дата		

результати студентів у форматі відомості, де можна переглянути статуси подань і виставлені бали за окремими завданнями. Progress для викладача показує кількість студентів у курсі, число завдань, подань і перевірених робіт, а ReportsPanel формує підсумок курсу та надає доступ до CSV-експорту журналу.

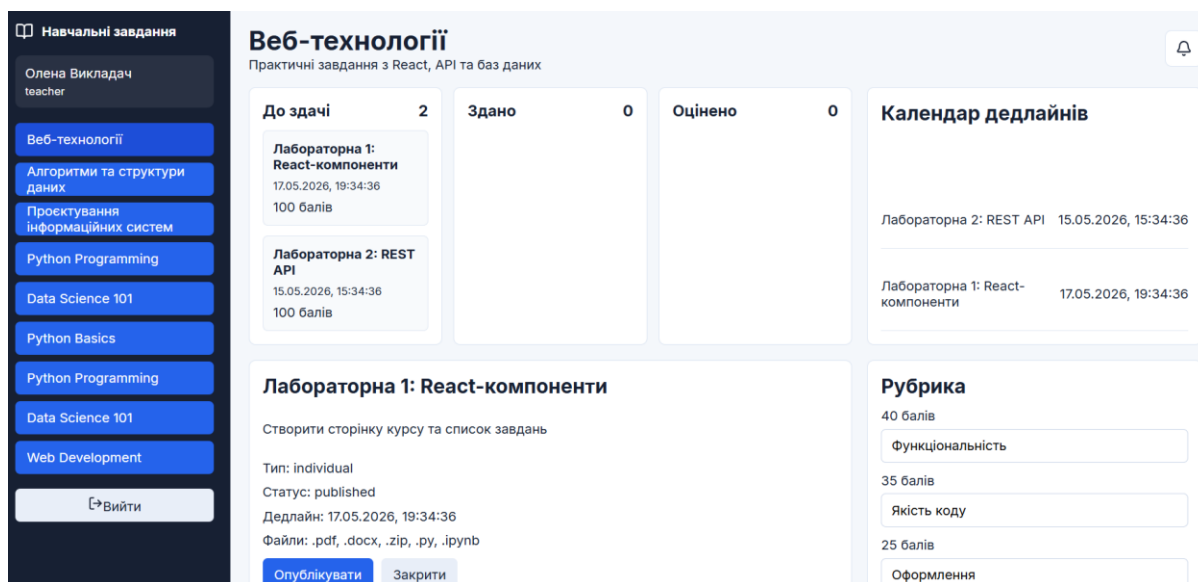
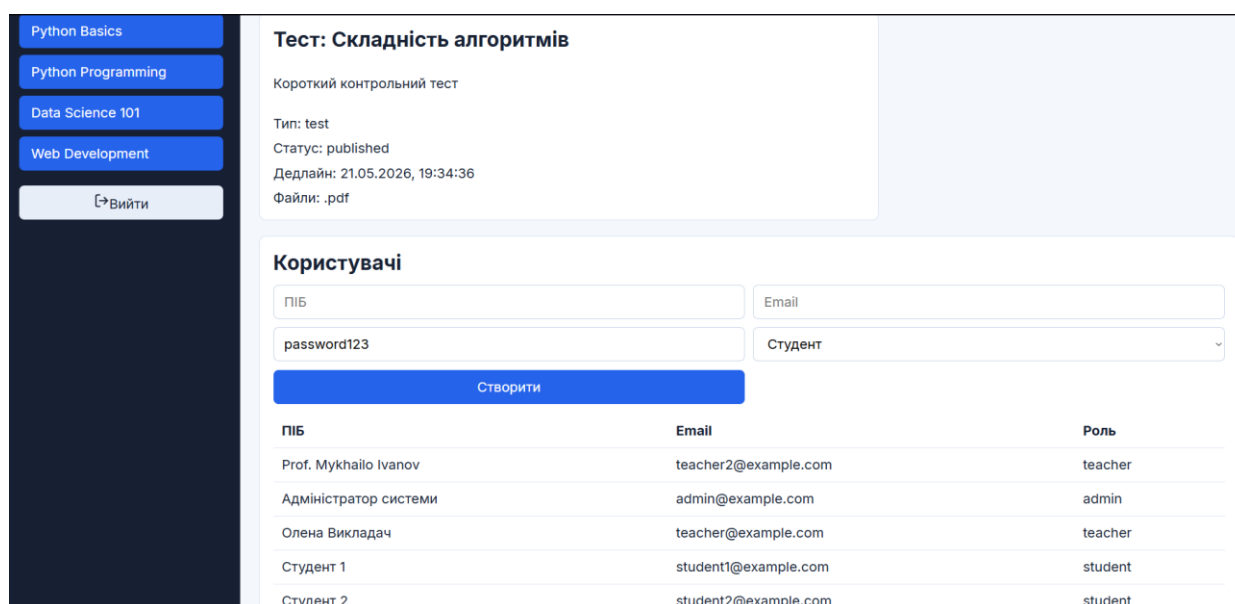


Рисунок. 3.5 – Інтерфейс викладача для роботи з курсами, завданнями та перевіркою подань

Завдяки цьому інтерфейс викладача охоплює не лише створення й перевірку завдань, а й оцінку загального стану навчальної роботи в межах дисципліни. Адміністративний режим клієнтської частини зосереджений на роботі з користувачами. У панелі адміністратора відображається перелік облікових записів, їхні ролі та контактні дані, а також доступна форма створення нового користувача зі статусом студента, викладача або адміністратора.

Така реалізація підтримує загальну логіку системи, у якій саме адміністратор формує початкову користувацьку структуру, після чого викладач і студент виконують уже прикладні навчальні дії. Панель адміністратора не втручається у процес оцінювання чи подання робіт, але створює організаційну основу для коректної роботи всіх інших режимів (рисунок 3.6).

Окремим елементом клієнтського інтерфейсу є сповіщення, що відображають повідомлення про наближення дедлайнів, виставлення оцінок, повернення робіт на доопрацювання та інші події, пов'язані з навчальними діями користувача. Для оновлення використано хук useWebSocket, який відкриває з'єднання після входу до системи та передає нові повідомлення до Zustand-сховища сповіщень. Завдяки цьому лічильник у дзвіночку повідомлень і сам список подій можуть змінюватися без перезавантаження сторінки. Такий підхід підвищує зв'язність інтерфейсу з реальними подіями backend-рівня та робить сповіщення не статичним архівом, а діючою частиною користувацького середовища.



Рисунк. 3.6 – Панель адміністратора для керування користувачами системи

Отже, клієнтську частину інформаційної системи реалізовано як рольово орієнтований вебінтерфейс, що поєднує роботу з курсами, завданнями, студентськими поданнями, оцінюванням, аналітикою, звітністю та сповіщеннями. Вона не обмежується відображенням даних, а забезпечує повноцінну взаємодію користувача із системою в межах його функцій. React-компоненти, кастомні hooks, Zustand-сховища та WebSocket-оновлення

утворюють цілісну клієнтську логіку, яка узгоджується із серверною реалізацією й структурою бази даних.

### 3.4 Демонстрація основних сценаріїв роботи та перевірка працездатності системи

Працездатність розробленої інформаційної системи розглядається через послідовний сценарій, який відтворює реальний порядок навчальної взаємодії. У межах такого сценарію викладач створює навчальний курс, формує завдання з дедлайном і правилами подання, студент отримує доступ до опублікованої роботи та надсилає власний результат, після чого викладач перевіряє подання, виставляє оцінку або повертає його на доопрацювання. Далі система оновлює статуси, журнал оцінок, аналітичні показники й сповіщення. Така послідовність була закладена в проектну логіку другого розділу та реалізована в програмній частині системи.

Початковим етапом циклу є робота викладача з навчальним курсом, після входу до системи користувач із роллю викладача переходить до відповідного робочого середовища, де може створити новий курс і задати його основні характеристики. Реалізована логіка передбачає, що курс автоматично пов'язується з поточним викладачем, а в разі адміністративного створення може бути призначений іншому користувачу з відповідною роллю. Після формування курсу викладач створює нове завдання, указуючи назву, опис, тип роботи, строк виконання, максимальний бал, допустимі формати файлів і можливість повторної задачі. На серверному рівні така робота не стає доступною студентам одразу, оскільки нове завдання отримує стан чернетки й лише після публікації переходить до активного навчального процесу (рисунк 3.7).

На цьому етапі система показує важливу відмінність між простим збереженням записів і повноцінною реалізацією навчальної логіки. Викладач не лише додає нову позицію в список, а формує об'єкт, який надалі визначає умови

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 47
Зм.	Арк.	№ докум.	Підпис	Дата		

роботи студента. Дедлайн впливає на подальший статус подання, ознака повторної здачі задає можливість доопрацювання, а перелік дозволених типів файлів використовується під час контролю результату перед відправленням.

The screenshot shows three main sections in the instructor interface:

- Новий курс (New Course):** Two input fields containing 'Test', a blue 'Створити курс' (Create Course) button, and a 'Test' label.
- Нове завдання (New Assignment):** Two input fields with 'Test', a dropdown menu set to 'Індивідуальне' (Individual), another dropdown set to 'Чернетка' (Draft), a date field '18.05.2026 18:53', a score field '100', a file type input field with '.pdf,.docx,.zip,.py,.ipynb', a checked checkbox 'Повторна здача' (Repeatable), and a blue 'Додати завдання' (Add Assignment) button.
- Перевірка робіт (Check Assignments):** A list item 'fbbbdc21graded' with a score of '100', a 'Відгук' (Feedback) input field, and 'Оцінити' (Evaluate) and 'Повернути' (Return) buttons.

Рисунок. 3.7 – Створення курсу та навчального завдання в інтерфейсі викладача

Завдяки чому на етапі створення завдання система накопичує всю інформацію, потрібну для наступних дій, і не вимагає ручного доопрацювання умов під час приймання робіт. Після публікації завдання воно стає доступним студенту в межах відповідного курсу.

У студентському інтерфейсі відображаються назва роботи, встановлений строк, максимальна оцінка й поточний стан виконання, після відкриття деталей студент бачить повний опис завдання та вимоги до матеріалів, які дозволено подати. На backend-рівні під час створення подання перевіряється, чи записаний студент на курс, чи завдання опубліковане, чи відповідає файл установленим обмеженням і чи не порушено правила повторної здачі. Якщо роботу надіслано вчасно, вона отримує статус *submitted*, а якщо строк уже минув, результат зберігається зі статусом *late*, що дозволяє не втратити факт подання, але водночас зафіксувати порушення дедлайну (рисунок 3.8).

Студент обирає завдання у власному курсі, додає файл, надсилає його через форму, а система одночасно створює запис подання, зберігає відомості про вкладення, оновлює стан роботи й робить її доступною викладачу для подальшої перевірки. Якщо викладач у майбутньому поверне роботу на доопрацювання, а параметри завдання дозволяють повторне надсилання, студент зможе подати оновлену версію як наступну спробу, не втрачаючи попередньої історії взаємодії.

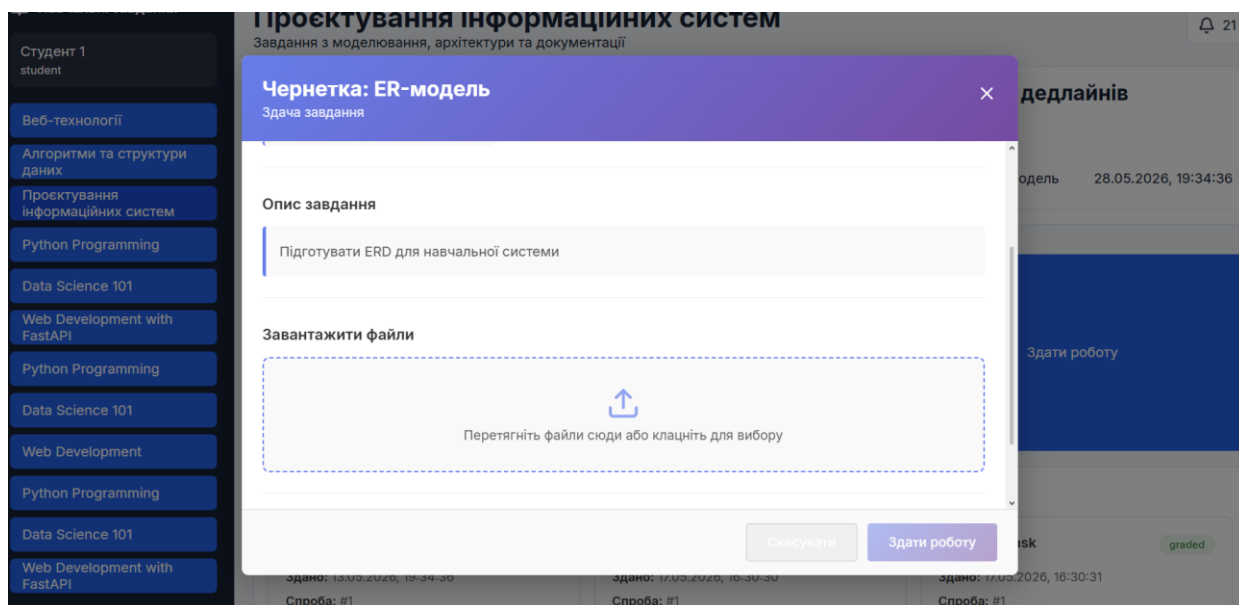


Рисунок. 3.8 – Подання студентської роботи через клієнтський інтерфейс

Наступним етапом є перевірка робіт викладачем, у робочій області викладача відображаються студентські результати, які очікують опрацювання. Після вибору конкретної роботи він може переглянути вкладені файли, оцінити їх за підготовленою рубрикою, зазначити підсумковий бал, додати текстовий відгук і прийняти одне з двох рішень. Якщо робота відповідає вимогам, вона отримує оцінку й переходить до завершеного стану. Якщо результат потребує виправлень, викладач повертає його на доопрацювання, а студент отримує відповідне повідомлення й можливість повторного подання за наявності дозволу на таку дію.

Стани `graded` і `returned` реалізовані безпосередньо в моделі подання, тому рішення викладача впливає не лише на текстовий коментар, а на весь подальший перебіг навчального процесу (рисунок 3.9).

Після оцінювання зміни автоматично відображаються в суміжних частинах системи, а у журналі оцінок фіксується новий результат, а у прогрес студента враховується виконане завдання й отриманий бал, а через механізм сповіщень користувач може дізнатися про виставлення оцінки або повернення роботи. На рівні звітності викладач отримує узагальнені показники за курсом, зокрема кількість студентів, завдань, подань, оцінених робіт і середній бал. Така поведінка системи підтверджує, що подання, оцінювання, аналітика й сповіщення не є окремими екранами без взаємозв'язку, а працюють як частини єдиного інформаційного процесу.

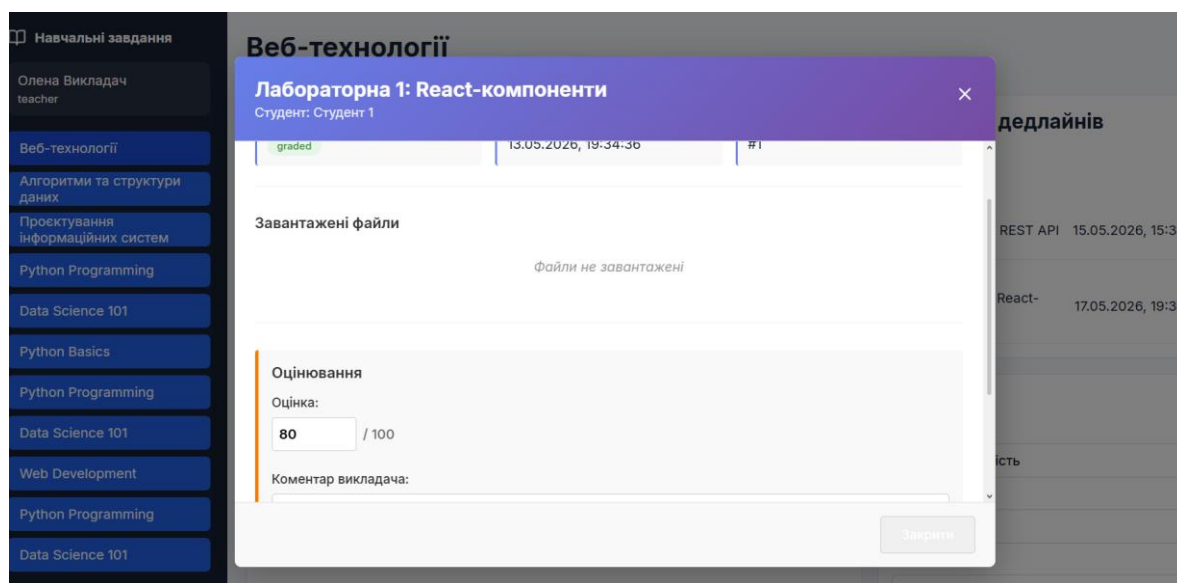


Рисунок. 3.9 – Перевірка студентської роботи та формування результату оцінювання

Перевірка працездатності системи здійснювалася також на рівні окремих правил оброблення даних, у проєкті реалізовано умови, за якими викладач може працювати лише з власними курсами, студент бачить лише ті курси, до яких його зараховано, завдання після створення перебуває в стані чернетки, а приймання

					КвРІСТ 220180.22.01.06 ПЗ	Арк. 50
Зм.	Арк.	№ докум.	Підпис	Дата		

робіт можливе лише після публікації. Для завдань, які здає студент контролюються дедлайни, статуси, допустимі файли й повторні спроби. Такий набір правил дозволяє перевірити не тільки відображення сторінок, а й коректність самої прикладної логіки, яка лежить в основі навчального циклу.

Сценарію перевірки показав, що система підтримує всі передбачені кроки без розриву між ними. Викладач створює навчальний контекст і публікує завдання, студент бачить його в курсі й надсилає роботу, викладач переглядає результат і приймає рішення, після чого дані оновлюються в журналі, аналітиці й повідомленнях. У цьому процесі одночасно перевіряється робота авторизації, рольового доступу, системи курсів, механізму завдань, подання файлів, оцінювання та обчислення підсумкової інформації.

Отже, демонстрація основних сценаріїв роботи підтвердила, що реалізована система відтворює повний порядок управління навчальними завданнями. Забезпечує створення курсів і завдань, контроль їхнього життєвого циклу, здачу студентських робіт, перевірку результатів, повернення на доопрацювання, фіксацію оцінок, формування сповіщень і оновлення аналітичних показників. Усі ці можливості свідчать про відповідність програмної реалізації завданням, сформульованим у попередніх розділах.

### 3.5 Висновки до третього розділу

У третьому розділі розглянуто практичну реалізацію інформаційної системи управління навчальними завданнями студентів. Серверну частину побудовано на основі FastAPI із чітким розмежуванням маршрутів, сервісної логіки, механізмів авторизації, роботи з файлами та сповіщеннями.

Реалізована рольова модель дозволяє коректно розподілити доступ між адміністратором, викладачем і студентом, а прикладні правила системи забезпечують правильний порядок виконання основних навчальних дій:

створення курсу, публікацію завдання, подання роботи, оцінювання та повернення результату на доопрацювання.

Описана структура бази даних дає змогу зберігати всі відомості, необхідні для підтримки повного циклу роботи із завданнями, що дозволяє не лише фіксувати окремі операції, а й відтворювати послідовність навчальної взаємодії, зберігати історію виконання робіт і формувати підсумкові показники для журналу оцінок, аналітики та звітності.

Клієнтську частину реалізовано як рольово орієнтований вебінтерфейс, у якому кожен користувач отримує ті інструменти, що відповідають його функціям у системі. Студент працює з курсами, завданнями, дедлайнами, поданнями, оцінками та власним прогресом. Викладач створює навчальний контент, переглядає студентські роботи, виконує перевірку та аналізує результати курсу. Адміністратор підтримує користувацьку структуру системи.

Демонстрація підтвердила, що реалізовані компоненти працюють узгоджено, де викладач може створити курс і нове завдання, студент бачить його в доступному навчальному просторі та надсилає файл із результатом, після чого викладач перевіряє подання, виставляє оцінку або повертає роботу на доопрацювання. Після цього система оновлює відповідні статуси, журнал оцінок, аналітичні показники й сповіщення. Така послідовність показує, що створений програмний продукт не обмежується окремими сторінками чи формами, а реалізує завершений цикл керування навчальними завданнями.

Отже, підтверджено, що спроектовані раніше рішення були реалізовані у вигляді працездатної інформаційної системи та відповідають до поставленої мети роботи. Вона забезпечує авторизований доступ, підтримує ролі користувачів, дає змогу керувати курсами й завданнями, приймати та перевіряти студентські роботи, формувати оцінки, сповіщення, аналітику та звітні дані.

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						52
Зм.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

У роботі за результатами виконаних теоретичних і практичних досліджень було розроблено інформаційну систему управління навчальними завданнями студентів, призначену для впорядкування основних процесів взаємодії між студентами, викладачами й адміністраторами в цифровому освітньому середовищі. Створене рішення охоплює повний цикл роботи з навчальними завданнями, починаючи від формування курсу та публікації роботи й завершуючи здечею результату студентом, перевіркою викладачем, виставленням оцінки та відображенням підсумкових показників.

Реалізована система забезпечує не лише облік навчальних робіт, а й підтримує послідовний порядок їхнього проходження через усі головні стани, що робить процес організації та контролю навчальної діяльності більш зрозумілим і керованим.

У першому розділі проведено аналіз предметної області управління навчальними завданнями студентів і визначено чинники, які обґрунтовують актуальність розроблення спеціалізованої інформаційної системи. Було розглянуто особливості цифрової організації навчального процесу, значення контролю дедлайнів, статусів подання, оцінювання, зворотного зв'язку та навчальної аналітики. Досліджено сучасні програмні рішення, що дозволило встановити спільні підходи до побудови таких платформ і виявити функції, найбільш важливі для розроблюваної системи.

На основі проведеного аналізу сформульовано мету, завдання, об'єкт і предмет дослідження, а також обґрунтовано доцільність створення окремого веборієнтованого середовища, зосередженого саме на керуванні навчальними завданнями, поданнями, перевіркою й результатами прогресу.

У другому розділі проведено проєктування інформаційної системи, визначено логіку організації роботи з навчальними завданнями, описано взаємодію адміністратора, викладача і студента, сформовано функціональну

					КвРІСТ 220180.22.01.06 ПЗ	Арк.
						53
Зм.	Арк.	№ докум.	Підпис	Дата		

структуру системи та встановлено призначення її основних частин. Також розроблено інформаційну модель обліку навчальних завдань і результатів їх виконання, у якій пов'язані користувачі, курси, завдання, подання робіт, оцінювання, сповіщення, прогрес і звітність. Для найважливіших подій створено алгоритми подання студентської роботи та перевірки й оцінювання подання викладачем. Це дозволило визначити послідовність зміни статусів, умови перевірки дедлайнів, порядок повернення на доопрацювання та логіку подальшого оновлення аналітичних показників. Таким чином, другий розділ сформував змістову основу для подальшої програмної реалізації системи.

У третьому розділі реалізовано основні програмні компоненти інформаційної системи та показано їхню узгоджену роботу на практиці. Серверну частину створено на основі FastAPI з використанням рольового доступу, JWT-автентифікації, сервісного шару бізнес-логіки, механізмів роботи з файлами. Для збереження навчальної інформації реалізовано базу даних SQLite через SQLAlchemy, яка охоплює користувачів, курси, записи студентів, завдання, подання, файли, оцінки, рубрики, коментарі та повідомлення. Клієнтську частину створено засобами React і TypeScript, де кожна роль отримала відповідний робочий інтерфейс. У системі реалізовано сторінки для авторизації, керування курсами, створення завдань, подання робіт, перевірки результатів, перегляду журналу оцінок, аналітики, сповіщень, звітів і керування користувачами.

Отже, розроблена інформаційна система дає змогу централізовано організувати роботу з навчальними завданнями, автоматизувати контроль строків і статусів, спростити подання та перевірку студентських робіт, забезпечити своєчасне інформування користувачів і формувати узагальнену картину навчальної активності. Реалізована архітектура створює основу для подальшого розвитку системи, зокрема розширення механізмів звітності, удосконалення оцінювання, інтеграції додаткових каналів сповіщень і масштабування для ширшого кола користувачів.

					КвРІСТ 220180.22.01.06 ПЗ	Арк.
						54
Зм.	Арк.	№ докум.	Підпис	Дата		

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1) UNESCO. Global Education Monitoring Report 2023: Technology in education: A tool on whose terms? Paris : UNESCO, 2023. URL: <https://unesdoc.unesco.org/ark:/48223/pf0000385723> (дата звернення: 12.04.2026).

2) OECD. OECD Digital Education Outlook 2023: Towards an Effective Digital Education Ecosystem. Paris : OECD Publishing, 2023. DOI: <https://doi.org/10.1787/c74f03de-en>.

3) European Commission. Digital Education Action Plan 2021–2027. URL: <https://education.ec.europa.eu/focus-topics/digital-education/actions> (дата звернення: 12.04.2026).

4) EDUCAUSE. 2024 EDUCAUSE Horizon Report: Teaching and Learning Edition. Louisville : EDUCAUSE, 2024. URL: <https://library.educause.edu/resources/2024/5/2024-educause-horizon-report-teaching-and-learning-edition> (дата звернення: 12.04.2026).

5) Schoonenboom J. Using an adapted, task-level technology acceptance model to explain why instructors in higher education intend to use some learning management system tools more than others. Computers & Education. 2014. Vol. 71. P. 247–256. DOI: <https://doi.org/10.1016/j.compedu.2013.09.016> (дата звернення: 01.03.2026)..

6) Sánchez L., Peñarreta J., Soria Poma X. Learning management systems for higher education: A brief comparison. Discover Education. 2024. Vol. 3. Article 58. DOI: <https://doi.org/10.1007/s44217-024-00143-5> (дата звернення: 01.03.2026).

7) Rubin B., Fernandes R., Avgerinou M. D., Moore J. The effect of learning management systems on student and faculty outcomes. The Internet and Higher Education. 2010. Vol. 13, No. 1–2. P. 82–83. DOI: <https://doi.org/10.1016/j.iheduc.2009.10.008> (дата звернення: 03.03.2026).

8) Broadbent J., Poon W. L. Self-regulated learning strategies and academic achievement in online higher education learning environments: A systematic review.

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 55
Зм.	Арк.	№ докум.	Підпис	Дата		

The Internet and Higher Education. 2015. Vol. 27. P. 1–13. DOI: <https://doi.org/10.1016/j.iheduc.2015.04.007> (дата звернення: 03.03.2026).

9) Carless D., Boud D. The development of student feedback literacy: Enabling uptake of feedback. *Assessment & Evaluation in Higher Education*. 2018. Vol. 43, No. 8. P. 1315–1325. DOI: <https://doi.org/10.1080/02602938.2018.1463354> (дата звернення: 03.03.2026).

10) Ferguson R. Learning analytics: Drivers, developments and challenges. *International Journal of Technology Enhanced Learning*. 2012. Vol. 4, No. 5/6. P. 304–317. DOI: <https://doi.org/10.1504/IJTEL.2012.051816> (дата звернення: 12.03.2026).

11) Siemens G. Learning analytics: The emergence of a discipline. *American Behavioral Scientist*. 2013. Vol. 57, No. 10. P. 1380–1400. DOI: <https://doi.org/10.1177/0002764213498851> (дата звернення: 16.03.2026).

12) Viberg O., Hatakka M., Bälter O., Mavroudi A. The current landscape of learning analytics in higher education. *Computers in Human Behavior*. 2018. Vol. 89. P. 98–110. DOI: <https://doi.org/10.1016/j.chb.2018.07.027> (дата звернення: 17.03.2026).

13) Wong B. T. M. Learning analytics in higher education: An analysis of case studies. *Asian Association of Open Universities Journal*. 2017. Vol. 12, No. 1. P. 21–40. DOI: <https://doi.org/10.1108/AAOUJ-01-2017-0009> (дата звернення: 19.03.2026).

14) Ifenthaler D., Yau J. Y.-K. Utilising learning analytics to support study success in higher education: A systematic review. *Educational Technology Research and Development*. 2020. Vol. 68. P. 1961–1990. DOI: <https://doi.org/10.1007/s11423-020-09788-z> (дата звернення: 21.03.2026).

15) Bälter O., Jemstedt A., Abraham F. J., Osowski C. P., Mugisha R., Bälter K. Effect of personalized email-based reminders on participants' timeliness in an online education program: Randomized controlled trial. *JMIR Formative Research*. 2023. Vol. 7. Article e43977. DOI: <https://doi.org/10.2196/43977> (дата звернення: 25.03.2026).

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 56
Зм.	Арк.	№ докум.	Підпис	Дата		

16) Weijers R. J., de Koning B. B., Paas F. Nudging in education: From theory towards guidelines for successful implementation. *European Journal of Psychology of Education*. 2021. Vol. 36. P. 883–902. DOI: <https://doi.org/10.1007/s10212-020-00495-0> (дата звернення: 25.03.2026).

17) Castleman B. L., Page L. C. Summer nudging: Can personalized text messages and peer mentor outreach increase college going among low-income high school graduates? *Journal of Economic Behavior & Organization*. 2015. Vol. 115. P. 144–160. DOI: <https://doi.org/10.1016/j.jebo.2014.12.008> (дата звернення: 27.03.2026).

18) Theobald M., Bellhäuser H. How am I going and where to next? Elaborated online feedback improves university students' self-regulated learning and performance. *The Internet and Higher Education*. 2022. Vol. 55. Article 100872. DOI: <https://doi.org/10.1016/j.iheduc.2022.100872> (дата звернення: 28.03.2026).

19) Guzmán-Valenzuela C., Gómez-González C., Rojas-Murphy Tagle A., Lorca-Vyhmeister A. Learning analytics in higher education: A preponderance of analytics but very little learning? *International Journal of Educational Technology in Higher Education*. 2021. Vol. 18. Article 23. DOI: <https://doi.org/10.1186/s41239-021-00258-x> (дата звернення: 02.04.2026).

20) Google. Time for a refresh: Meet the new Google Classroom. 2018. URL: <https://blog.google/products-and-platforms/products/education/time-refresh-meet-new-google-classroom/> (дата звернення: 12.04.2026).

21) Cavus N., Zabadi T. A comparison of open source learning management systems. *Procedia - Social and Behavioral Sciences*. 2014. Vol. 143. P. 521–526. DOI: <https://doi.org/10.1016/j.sbspro.2014.07.430> (дата звернення: 05.04.2026).

22) Al-Fraihat D., Joy M., Masa'deh R., Sinclair J. Evaluating e-learning systems success: An empirical study. *Computers in Human Behavior*. 2020. Vol. 102. P. 67–86. DOI: <https://doi.org/10.1016/j.chb.2019.08.004> (дата звернення: 03.04.2026).

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 57
Зм.	Арк.	№ докум.	Підпис	Дата		

23) Al-Marouf R. A. S., Al-Emran M. Students acceptance of Google Classroom: An exploratory study using PLS-SEM approach. International Journal of Emerging Technologies in Learning. 2018. Vol. 13, No. 6. P. 112–123. DOI: <https://doi.org/10.3991/ijet.v13i06.8275> (дата звернення: 08.04.2026).

24) Schumacher C., Ifenthaler D. Features students really expect from learning analytics. Computers in Human Behavior. 2018. Vol. 78. P. 397–407. DOI: <https://doi.org/10.1016/j.chb.2017.06.030> (дата звернення: 08.04.2026).

25) Google. Grade and return an assignment. URL: <https://support.google.com/edu/classroom/answer/6020294> (дата звернення: 12.04.2026).

26) Google. Give feedback on assignments. URL: <https://support.google.com/edu/classroom/answer/9093530> (дата звернення: 12.04.2026).

27) Google. View or update your gradebook. URL: <https://support.google.com/edu/classroom/answer/9199710> (дата звернення: 12.04.2026).

28) Google. Manage notifications. URL: <https://support.google.com/edu/classroom/answer/6141557> (дата звернення: 12.04.2026).

29) Instructure. How do I use the Gradebook? URL: <https://community.instructure.com/en/kb/articles/660826-how-do-i-use-the-gradebook> (дата звернення: 12.04.2026).

30) Instructure. What is SpeedGrader? URL: <https://community.instructure.com/en/kb/articles/662775-what-is-speedgrader> (дата звернення: 12.04.2026).

31) Instructure. Canvas notifications. URL: <https://community.instructure.com/en/kb/articles/387041-canvas-notifications> (дата звернення: 12.04.2026).

32) Instructure. What is course analytics? URL: <https://community.instructure.com/en/kb/articles/662739-what-is-course-analytics> (дата звернення: 12.04.2026).

					КВРІСТ 220180.22.01.06 ПЗ	Арк. 58
Зм.	Арк.	№ докум.	Підпис	Дата		

33) Microsoft. Assignments in Teams for Education. URL: <https://learn.microsoft.com/en-us/microsoftteams/expand-teams-across-your-org/assignments-in-teams> (дата звернення: 12.04.2026).

34) Microsoft. Create an assignment in Microsoft Teams. URL: <https://support.microsoft.com/en-us/topic/create-an-assignment-in-microsoft-teams-23c128d0-ec34-4691-9511-661fba8599be> (дата звернення: 12.04.2026).

35) Microsoft. Track student progress in the Grades tab. URL: <https://support.microsoft.com/en-us/topic/track-student-progress-in-the-grades-tab-bc76ffde-bbf7-4db6-b443-735a07e32de0> (дата звернення: 12.04.2026).

36) Microsoft. Grades and Insights. URL: <https://support.microsoft.com/en-us/topic/grades-and-insights-cab0c8ac-0a25-49a6-829a-dfcd8d76c599> (дата звернення: 12.04.2026).

37) Moodle. Assignment activity. URL: [https://docs.moodle.org/en/Assignment\\_activity](https://docs.moodle.org/en/Assignment_activity) (дата звернення: 12.04.2026).

38) Moodle. Assignment settings. URL: [https://docs.moodle.org/en/Assignment\\_settings](https://docs.moodle.org/en/Assignment_settings) (дата звернення: 12.04.2026).

39) Moodle. Rubrics. URL: <https://docs.moodle.org/en/Rubrics> (дата звернення: 12.04.2026).

40) Moodle. Notifications. URL: <https://docs.moodle.org/en/Notifications> (дата звернення: 12.04.2026).

41) Moodle. Grades. URL: <https://docs.moodle.org/en/Grades> (дата звернення: 12.04.2026).

					КВРІСТ 220180.22.01.06 ПЗ	Арк.
						59
Зм.	Арк.	№ докум.	Підпис	Дата		







## Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

**Автор:** Денис СІМЧУК

**Співавтор:**

**Назва:** Інформаційна система управління навчальними завданнями студентів

**Експерт:** Тетяна КИСІЛЬ

**Підрозділ:** Кафедра комп'ютерної інженерії та інформаційних систем

**Коефіцієнт подібності 1:** 4.94%

**Коефіцієнт подібності 2:** 2.48%

**Мікропробіли:** 7

**Заміна букв:** 0

**Інтервали:** 0

**Білі знаки:** 0

**Дата створення звіту:** 2026-05-27 21:01:55.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедур. Таким чином робота не приймається.

Обґрунтування:

2026-05-28

Дата



Доцент Андрій Нічепорук

експерт

## Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 1.0%

Словники перевірки: en\_US, ru\_RU, ua\_UA. Помилки в документах: 7%

ID: 272548 Назва: БКР Інформаційна система управління навчальними завданнями студентів Додано в БД: 2026-05-28 Автора: Денис СІМЧУК Керівники: Тетяна КИСІЛЬ Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	83248	606	1684 (2%)	22 (4%)

### Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Сімчук Денис Олегович

Тема: Інформаційна система керування навчальними завданнями студентів

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень   3   Кількість сторінок записки   50  

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є проектування, програмна реалізація та перевірка інформаційної системи для керування навчальними завданнями студентів.

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі кваліфікаційної роботи проведено аналіз предметної області управління навчальними завданнями студентів, розглянуто сучасні підходи до організації освітнього процесу та проаналізовано можливості відомих систем дистанційного навчання. На основі проведеного аналізу визначено основні вимоги до інформаційної системи та сформульовано постановку задачі. У другому розділі виконано проектування інформаційної системи, а саме: розроблено логіку роботи системи, визначено ролі користувачів, спроектовано функціональну структуру, інформаційну модель та алгоритми основних процесів. Також сформовано модель взаємодії між користувачами, завданнями, оцінюванням, сповіщеннями та засобами аналітики. У третьому розділі виконано програмну реалізацію інформаційної системи, розроблено серверну та клієнтську частини, реалізовано базу даних, механізми авторизації, керування курсами й завданнями, оцінювання, сповіщення та формування звітності. Проведено перевірку працездатності системи та демонстрацію основних сценаріїв її використання.

4. Позитивні сторони роботи: висока практична цінність розробленої інформаційної системи, актуальність теми дослідження,

5. Негативні сторони роботи: недостатня увага приділена питанням масштабування системи та її інтеграції з іншими інформаційними ресурсами закладів освіти.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на належному науково-технічному рівні.

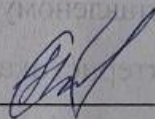
8. Інші зауваження: \_\_\_\_\_

– 9. Оцінка дипломної роботи: відмінно (А/93)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) \_\_\_\_\_

Гублек Наталія Серіївна, д-р філософії,  
доцент кафедри КБ

"04" 06 2026 р.

 (підпис)

Зав. кафедри КПС  
д-р. філософії Ользі ПАВЛОВІЙ

Денис СІМЧУК

---

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТ-22-1

### ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року



## РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

### КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Інформаційна система управління навчальними завданнями студентів  
 Автор Денис СІМЧУК  
 Освітня програма Інформаційні системи та технології  
 Рівень вищої освіти перший (бакалаврський)  
 Спеціальність 126 Інформаційні системи та технології  
 Науковий керівник: канд. фіз.-мат. наук, доц. Тетяна КИСІЛЬ

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

#### Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 4,94%; та системою Anti-Plagiarism складає 0,2%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

01.06.2026

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи

  
 Підпис  
  
 Підпис  
  
 Підпис

Ольга ПАВЛОВА  
Ім'я, ПРІЗВИЩЕ

Єлизавета ГНАТЧУК  
Ім'я, ПРІЗВИЩЕ

Тетяна КИСІЛЬ  
Ім'я, ПРІЗВИЩЕ