

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки
на тему:

**РОЗРОБКА СИСТЕМИ ДЛЯ АВТОМАТИЧНОГО ОЦІНЮВАННЯ ТЕСТІВ
З ВІДКРИТИМИ ПИТАННЯМИ**

Виконала студентка 4-го курсу
Ксенія ГРИЩЕНКО



(підпис)

Науковий керівник:
доцент, кандидат технічних наук
Олексій ТКАЧЕНКО



(підпис)

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

Студент



(підпис)

Роботу розглянуто й допущено до захисту на засіданні
кафедри теорії та технології програмування

«5» червня 2023 р.,

протокол № 18

Завідувач кафедри

Микола НІКІТЧЕНКО

(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи 59 сторінок, 32 ілюстрації, 1 таблиця, 18 джерел посилань.

АНАЛІЗ ТЕКСТУ, БАЗА ДАНИХ, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, КУРС, ОЦІНЮВАННЯ ВІДКРИТИХ ВІДПОВІДЕЙ, ПРЕДМЕТ, РОЗПІЗНАВАННЯ МОВЛЕННЯ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єктом дослідження є автоматизоване тестування відкритих відповідей. Предметом дослідження є застосування технологій штучного інтелекту в розробці системи оцінювання відкритих відповідей.

Метою дослідження є створення системи для автоматичного оцінювання тестів з відкритими питання з можливостями ведення обліку робіт студентів, призначених завдань, управління курсами.

Завдання дослідження включають в себе опис проблеми оцінювання відкритих питань та можливостей використання засобів штучного інтелекту для її розв'язання, огляд існуючих методів аналізу тексту, створення системи з використанням технологій штучного інтелекту, опис роботи з нею, а також її порівняння із вже існуючими конкурентами.

Методи розробки: комп'ютерне моделювання, об'єктно-орієнтовний підхід, аналіз і порівняння існуючих систем. Інструменти розробки: мови програмування Python, JavaScript, IDE PyCharm, текстовий редактор Visual Studio Code, фреймворки Django, Django Rest Framework, Vue.js.

Результати роботи: досліджено проблему відкритих питань й застосування засобів штучного інтелекту для її розв'язання, здійснено огляд методів аналізу тексту, розроблено систему автоматичного оцінювання відповідей, описано роботу з нею, а також проведено порівняння функціоналу з конкурентами.

Програмний продукт Sigma School може використовуватись в навчальному процесі будь-яких навчальних курсів (університетів).

Можливий подальший розвиток системи з додаванням нового функціоналу, який би покращив досвід використання даного продукту й зробив його корисним

засобом для використання в процесі навчання, а саме додавання інших видів питань в тести й призначення завдань на окремих студентів, а також можливість тестування й оцінювання математичних дисциплін, що містять формули й символи.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	6
ВСТУП	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ	10
1.1 Проблема оцінювання відкритих питань	10
1.2 Підходи обробки природної мови	10
1.3 Використання моделі GPT в оцінюванні відкритих відповідей	12
1.4 Принципи роботи GPT	12
1.4.1 Поняття GPT	12
1.4.2 Архітектура трансформаторів	13
1.4.3 Особливості роботи GPT	14
1.5 Розпізнавання мовлення	15
1.5.1 Опис процесу розпізнавання мовлення	15
1.5.2 Засоби розпізнавання мовлення	17
1.6 Сучасні підходи та методи аналізу тексту	17
РОЗДІЛ 2 ОГЛЯД НАЯВНИХ НА РИНКУ СИСТЕМ	20
РОЗДІЛ 3 ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ	24
3.1 Огляд технологій серверної частини	24
3.2 Огляд технологій клієнтської частини	24
3.3 Огляд БД	25
РОЗДІЛ 4 ПРИЗНАЧЕННЯ ТА АРХІТЕКТУРА СИСТЕМИ	26
4.1 Призначення системи	26
4.2 Архітектура системи	26
РОЗДІЛ 5 ОСОБЛИВОСТІ РОЗРОБКИ	28
5.1 Сутності системи	28

	5
5.2 Опис функціоналу	29
5.3 Деталі проходження тесту з відкритими відповідями	32
РОЗДІЛ 6 РОБОТА З СИСТЕМОЮ	34
РОЗДІЛ 7 ПОРІВНЯННЯ РЕАЛІЗОВАНОГО ФУНКЦІОНАЛУ З КОНКУРЕНТАМИ	52
ВИСНОВКИ	54
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТОК А Діаграма класів	58
ДОДАТОК Б Діаграма прецедентів для ролі вчителя (власника курсу)	59

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

DRF - Django REST Framework;

GPT - Generative Pre-trained Transformers;

GUI - Graphical user interface, графічний інтерфейс;

IDE – Integrated Design Environment, інтегроване середовище розробки;

NLP - Natural Language Processing, обробка природної мови;

БД - База даних;

МН - Машинне навчання;

ШІ - Штучний інтелект;

ВСТУП

Оцінка сучасного стану об'єкта дослідження або розробки. Оцінювання правильності відповідей студентів та їхнього рівня знань зазвичай залежить від викладачів, які проводять тести або екзамени. Такий підхід може призвести до відмінностей в оцінюванні, залежно від суб'єктивного судження викладача.

З розвитком технологій, зокрема штучного інтелекту та машинного навчання, з'явилися можливості для автоматизації процесу оцінювання відповідей студентів. Однак більшість існуючих систем зосереджена на тестах з кількома варіантами відповідей або коротких письмових відповідях.

Що стосується усних відповідей, сучасні системи зазвичай використовують прості алгоритми порівняння текстів або базуються на певних ключових словах. Це може призвести до помилок та необ'єктивності оцінки через обмеженість таких підходів.

Використання GPT[1] для аналізу правильності відповідей є інноваційним підходом, що дозволяє генерувати більш гнучкі та точні еталонні відповіді. Однак цей підхід ще не став широко використовуваним у системах оцінювання, і розробка такої системи на основі штучного інтелекту може привести до значних поліпшень в якості оцінювання усних відповідей студентів та ефективності навчального процесу.

Актуальність роботи та підстави для її виконання. Актуальність роботи визначається потребою в автоматизованому та об'єктивному оцінюванні усних відповідей студентів, що сприяє підвищенню якості навчання та ефективності освітнього процесу. Застосування моделі GPT для оцінювання відкритих відповідей дозволяє отримати більш точні та гнучкі оцінки, а також сприяє індивідуалізації навчання.

Підстави для виконання роботи включають розвиток технологій штучного інтелекту та машинного навчання, зокрема моделей, як GPT, що надають нові можливості для аналізу текстів та генерації відповідей. Також актуальність роботи

зумовлена зростанням потреби в дистанційній освіті та бажанням використовувати новітні технології для покращення якості та ефективності навчального процесу.

Мета й завдання роботи. Метою дослідження є створення системи для автоматичного оцінювання тестів з відкритими питаннями на основі моделі GPT для підвищення якості навчання та ефективності освітнього процесу, що також передбачатиме можливість ведення обліку робіт студентів, призначених завдань, управління курсами.

Завдання даної роботи включають:

- Описати проблему оцінювання відкритих питань та можливості використання засобів штучного інтелекту для її розв'язання.
- Вивчити сучасні методи аналізу тексту.
- Розробити систему, що використовує модель GPT для оцінювання відповідей.
- Описати роботу з системою.
- Порівняти розроблену систему з існуючими конкурентами, визначити відмінні риси та функції, що можна додати.

Об'єкт, предмет, методи й засоби розробки. Об'єктом дослідження є автоматизоване тестування відкритих відповідей.

Предметом дослідження є застосування технологій штучного інтелекту в розробці системи оцінювання відкритих відповідей.

Серед методів розробки можна виділити комп'ютерне моделювання, об'єктно-орієнтовний підхід, аналіз і порівняння існуючих систем.

Засоби розробки включають в себе IDE PyCharm[2] для програмування мовою Python[3], текстовий редактор Visual Studio Code[4] для використання мови JavaScript[5], фреймворки Django[6], DRF[7] для серверної частини, Vue[8] для клієнтської частини, бібліотеку Element UI[9] для GUI, а також PostgreSQL[10] в якості бази даних.

Можливі сфери застосування. Дана система може бути інтегрована в навчальний процес будь-яких навчальних курсів (або навіть університетів) і

використовуватись безпосередньо в якості застосунку, який би спрощував проведення усних робіт і вів облік робіт студентів окремих груп.

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ

1.1 Проблема оцінювання відкритих питань

Відкриті запитання становлять значну складність з точки зору оцінювання через притаманну їм гнучкість і складність, адже вони передбачають широкий діапазон можливих правильних відповідей. Це створює проблему для традиційних методів автоматизованого оцінювання, які працюють шляхом порівняння відповіді із заздалегідь визначеним ключем відповіді.

Подальші ускладнення виникають через неоднозначну природу людської мови. Ту саму відповідь можна сформулювати багатьма різними способами, і семантичну еквівалентність важко вловити. Відповіді можуть бути концептуально точними, але значно відрізнятися з точки зору граматики, синтаксису та лексики, що ускладнює визначення правильності.

Крім того, суб'єктивність відкритих питань також ускладнює їх оцінювання. Одне питання може бути інтерпретоване різними людьми по-різному, що призводить до різноманітних дійсних відповідей. Таким чином, оцінка достовірності відповіді стає суб'єктивною, залежно від розуміння та інтерпретації індивідуальної оцінки відповіді.

1.2 Підходи обробки природної мови

Використання обробки природної мови (NLP) може значно поліпшити процес оцінювання відкритих питань. А саме, допомогти ефективніше зрозуміти, інтерпретувати та класифікувати питання. Ці можливості стають ще більш важливими зі збільшенням кількості складних даних і складних мовних структур.

Підходи NLP можна описати за допомогою трьох основних стратегій.

- **Методи, засновані на правилах (rule-based)**, є найпершими типами підходів NLP. Вони все ще використовуються сьогодні, незважаючи на давнину появи, завдяки своїй доведеній ефективності. Методи, засновані на

правилах, можуть забезпечити глибоке розуміння, якщо застосувати їх до тексту. Наприклад, вони можуть визначити іменники в тексті або навіть розпізнати шаблон, який може бути кодом якоїсь мови програмування. Підходи, засновані на правилах, які включають регулярні вирази та контекстно-вільну граматику, насамперед роблять акцент на розпізнаванні шаблонів або аналізі. Ці методи часто сприймаються як підхід «заповнення прогалін». Вони мають низьку точність, але високу запам'ятовуваність, що свідчить про те, що вони можуть бути дуже ефективними в певних сценаріях, але можуть зазнати погіршення продуктивності, якщо застосовувати їх у більш загальній формі.

- **Класичні методи машинного навчання.** До них належать імовірнісне моделювання, максимізація ймовірності та лінійні класифікатори. Важливо відзначити, що ці методи не передбачають використання моделей нейронної мережі. Вони характеризуються використанням навчальних даних. Ці методи також включають конструювання ознак (feature engineering), яке включає такі елементи, як тип слова, контекст, використання великих літер, множина тощо. Процес, що виконується в цих методах, зазвичай передбачає навчання моделі на параметрах і подальшу адаптацію її до тестових даних. Висновок, який передбачає застосування моделі до тестових даних, характеризується передбаченням найбільш імовірних слів, наступного слова, найкращої категорії тощо. Іншою примітною особливістю цих методів є «семантичне заповнення слотів».
- **Нейронні мережі** пропонують подібний підхід до традиційного машинного навчання, але з кількома ключовими відмінностями. По-перше, зазвичай немає потреби в конструюванні ознак, оскільки нейронні мережі здатні самостійно виділяти їх. Ця характеристика часто вважається основною перевагою використання нейронних мереж у NLP. Замість самостійно розроблених функцій у нейронні мережі вводяться необроблені потоки параметрів, тобто векторні представлення слів. Однак ці мережі вимагають значної кількості тренувальних даних.

1.3 Використання моделі GPT в оцінюванні відкритих відповідей

Усі згадані раніше методи NLP: методи на основі правил, класичне машинне навчання та нейронні мережі – мають своє місце в аналізі та розумінні мови. Однак, коли справа доходить до оцінювання відкритих питань, один метод має значну перевагу: нейронні мережі, зокрема моделі на основі трансформаторів, такі як GPT (Generative Pretrained Transformer).

GPT, продукт розвитку нейронних мереж, має перевагу вивчення важливих функцій з даних без потреби в явному конструюванні ознак, таким чином перевершуючи традиційні методи машинного навчання на основі правил для певних завдань. Його архітектура дозволяє йому розуміти контекст довгих фрагментів тексту, що робить його особливо підходящим для відкритих запитань, які часто вимагають розуміння тонкого контексту для отримання відповідних відповідей. Крім того, GPT попередньо навчений на великій кількості даних, що дає змогу охоплювати широкий спектр мовних моделей і нюансів. Це широке розуміння мови та контексту в поєднанні з його здатністю генерувати зв'язні та контекстуально правильні відповіді робить GPT чудовим вибором для оцінювання відкритих запитань.

1.4 Принципи роботи GPT

1.4.1 Поняття GPT

Generative Pre-trained Transformers (GPT) являють собою серію моделей штучного інтелекту, заснованих на архітектурі трансформатора. Ці моделі відіграють важливу роль у створенні генеративних додатків ШІ, таких як ChatGPT. Вони надають програмам можливість генерувати текст та інший вміст (включаючи зображення, музику тощо), який точно імітує людське спілкування, і відповідати на запити в розмовному стилі.

GPT також може узагальнювати такий вміст, як розмови, статті, електронні листи та веб-сторінки. Він також використовується як персональний помічник для таких завдань, як керування розкладами та написання електронних листів. Загалом він має широкий спектр використання і може бути застосований до багатьох задач.

1.4.2 Архітектура трансформаторів

Архітектура трансформатора (рис. 1) - це інноваційний підхід до обробки послідовностей, запропонований у 2017 році у статті "Attention Is All You Need"[11] від Васвані та інших. Вона відрізняється від попередніх методів, таких як рекурентні нейронні мережі (RNN) та довгострокові короткострокові пам'яті (LSTM) тим, що в основному використовує **механізм уваги** для встановлення залежностей між словами у послідовностях.

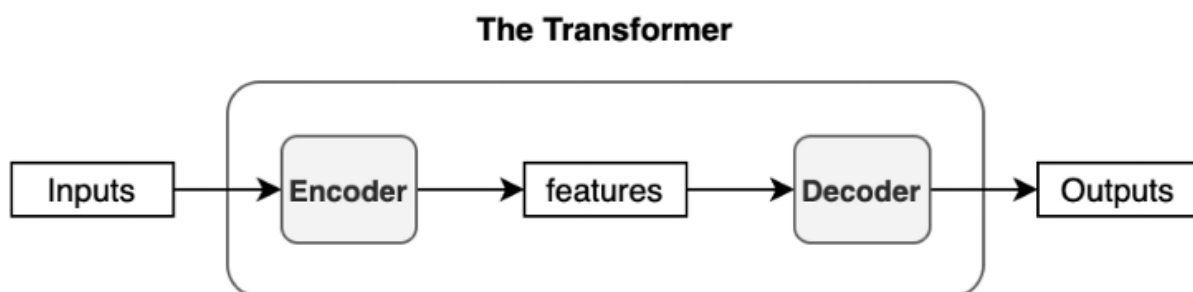


Рисунок 1 - Загальна модель архітектури трансформатора

Даний механізм уваги для зважування взаємозв'язків між різними словами у послідовності допомагає моделям краще виявляти довгострокові залежності у тексті. Вони також можуть обробляти всі слова в одному проході, що забезпечує більш швидке навчання та ефективніший процес обчислення порівняно з рекурентними моделями.

Трансформатори можуть бути легко масштабовані до різних рівнів складності, що дозволяє створювати моделі для різних завдань та рівнів

потужності. Завдяки цим перевагам та гнучкості, архітектура трансформатора може бути використана для широкого спектру завдань обробки природної мови, таких як машинний переклад, аналіз емоцій, створення текстів та багато іншого.

Більшість конкурентних нейронних моделей послідовної трансдукції мають структуру **кодера-декодера**. Тут кодер відображає вхідну послідовність символічних представлень (x_1, \dots, x_n) на послідовність безперервних представлень $z = (z_1, \dots, z_n)$. Враховуючи z , декодер потім генерує вихідну послідовність символів (y_1, \dots, y_m) по одному елементу за раз. На кожному кроці модель авторегресує, споживаючи раніше згенеровані символи як додатковий вхід при генеруванні наступного. Трансформатор слідує цій загальній архітектурі, використовуючи механізм самоуваги та точково пов'язані шари, для обох кодера та декодера.

1.4.3 Особливості роботи GPT

Процес навчання GPT поділяється на два етапи: попереднє навчання без нагляду та контрольне налаштування. На етапі попереднього навчання модель вчиться передбачати наступне слово в реченні, використовуючи великі обсяги немаркованих текстових даних з Інтернету. Це дає змогу моделі вивчати граматику, факти про світ та здібності міркувати.

На етапі контрольного налаштування попередньо підготовлена модель додатково навчається на певному завданні з використанням позначених даних. Це спеціальне навчання для конкретного завдання дозволяє моделі адаптувати свої загальні можливості розуміння мови до конкретних вимог даного завдання, наприклад перекладу, відповідей на запитання або аналізу настроїв.

Ще одна характеристика, яка вирізняє моделі GPT, особливо останні версії, це їхній надзвичайний розмір. Наприклад, модель GPT-4 має трильйон параметрів. Для порівняння її попередник, GPT-3, має 175 мільярдів параметрів. Сама величина цих моделей надає їм надзвичайну здатність інкапсулювати широкий

спектр мовних шаблонів, що дозволяє створювати більш узгоджені та контекстуально релевантні результати.

Моделі GPT також відрізняються своїми можливостями навчання передачі. Вони створені для вмілого виконання широкого спектру завдань, не вимагаючи конфігурації моделі для конкретного завдання. Ця риса можливості передачі, коли знання з одного завдання можна застосувати до інших, робить моделі GPT особливо універсальними.

Однак ця технологічна майстерність має деякі застереження. Оскільки моделі GPT навчаються на величезних обсягах текстових даних Інтернету, вони чутливі до успадкування та навіть посилення упереджень, присутніх у цих наборах даних. Крім того, їхні навички створення тексту, схожого на людський, становлять потенційний ризик неправомірного використання, зокрема створення оманливого вмісту або поширення дезінформації.

1.5 Розпізнавання мовлення

Дана система передбачає ввід даних, використовуючи мікрофон, для того, щоб зробити прозорим процес тестування і унеможливити копіювання відповіді з сторонніх ресурсів. Мовлення має бути оброблене відповідно і перетворене у текст задля подальшої обробки. Саме тому важливо розглянути, як саме відбувається процес розпізнавання мовлення і які засоби пропонуються для розв'язання цієї задачі.

1.5.1 Опис процесу розпізнавання мовлення

Система розпізнавання мовлення працює за допомогою ШІ, який ретельно перевіряє мову та діалект. Вона навчається розпізнавати слова, які вимовляє людина, і згодом відображає ці слова з високою точністю транскрипції як текстові дані або вміст моделі на дисплеї.

Розпізнавання мовлення здійснюється за допомогою методу, відомого як мовне та акустичне моделювання, яке використовує спеціальні алгоритми. Акустичне моделювання втілює зв'язок між звуковими хвилями та елементами мови. З іншого боку, мовне моделювання зіставляє послідовності слів із відповідними звуками, щоб розрізнити слова чи фрази, які звучать однаково (рис. 2).

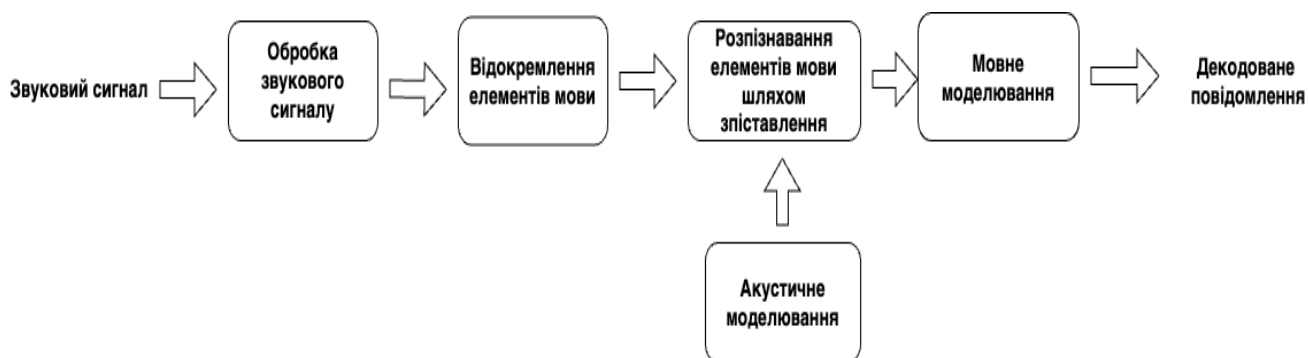


Рисунок 2 - Загальна схема розпізнавання мовлення

Приховані моделі Маркова (ПММ) також часто використовуються для точного визначення певних шаблонів мовлення, прив'язаних до часу, тим самим підвищуючи точність системи. ПММ — це статистична модель, яка описує систему, що випадково змінюється з часом, де майбутні зміни не залежать від минулих.

Інший підхід, який використовується для розпізнавання мовлення, включає N-грами та обробку природної мови (NLP). Процедура розпізнавання мовлення стає більш ефективною та швидшою за допомогою NLP. Тим часом N-грами забезпечують простіший метод для мовних моделей, створюючи розподіл ймовірностей для певної послідовності.

1.5.2 Засоби розпізнавання мовлення

Web Speech API[12], випущений наприкінці 2012 року групою спільнотою Speech API, пропонує веб-розробникам можливість інтегрувати функції мовного введення та виведення тексту у мовлення у веб-браузерах – функції, які зазвичай недоступні за допомогою звичайного програмного забезпечення для розпізнавання мовлення чи зчитування з екрана. Цей API також надає пріоритет конфіденційності користувача, вимагаючи явного дозволу від користувача для використання мікрофону, перш ніж веб-сайт зможе отримати доступ до його голосу через нього.

Web Speech API надає інтерфейс під назвою SpeechRecognition, що використовується для перетворення усної мови в письмовий текст. Ця функція працює шляхом розпізнавання голосового контексту з аудіовходу, як правило, через стандартну службу розпізнавання мовлення пристрою. Зазвичай це досягається шляхом створення нового об'єкта SpeechRecognition, який оснащений декількома обробниками подій для виявлення, коли мова вводиться через мікрофон пристрою.

Інтерфейс SpeechGrammar представляє набір граматичних правил, які має розпізнавати програма. Коли система успішно розпізнає слово чи фразу, вона повертає результат у вигляді текстового рядка. Потім цей текст можна використовувати для запуску інших дій у програмі. Більшість сучасних операційних систем мають систему розпізнавання мовлення за замовчуванням (наприклад, Dictation у macOS, Siri в iOS, Cortana у Windows 10 і Android Speech), і Web Speech API зазвичай використовує ці системи для розпізнавання мовлення.

1.6 Сучасні підходи та методи аналізу тексту

Оцінювання відкритих питань може бути складним завданням, оскільки вони не обмежуються одною відповіддю, а можуть допускати безліч варіацій. Саме тому дуже важливо мати надійні методи аналізу тексту, особливо в контексті

освіти. Ось кілька поширених методів і підходів, які можна поділити на окремі категорії:

- **Ручне кодування та тематичний аналіз.** Це передбачає ручне читання кожної відповіді та визначення повторюваних тем або категорій, що називається тематичним аналізом. Після визначення цих тем відповіді можна закодувати відповідно до відповідної теми. Ручне кодування займає багато часу та вимагає глибокого розуміння предмета, але воно часто дає цінну інформацію. Тут дуже важливим є критерій узгодженості між різними кодувальниками, оскільки це гарантує валідність кодування.
- **Контент-аналіз.** Контент-аналіз є більш систематичним і кількісним підходом до оцінювання відкритих запитань. Це передбачає створення схеми кодування для кількісного визначення присутності певних слів, тем або понять у відповідях. Цей метод дозволяє аналізувати великі обсяги даних і може виявити тенденції та закономірності у відповідях.
- **Інтелектуальний аналіз тексту (text mining) та обробка природної мови (NLP).** З розвитком ШІ і МН це методи стають все більш популярними. Вони передбачають використання алгоритмів для отримання значущої інформації з текстових даних. Такі алгоритми можуть швидко аналізувати великі набори даних і визначати закономірності, тенденції та зв'язки, які можуть бути неочевидними під час ручного кодування. Однак ці методи можуть не повністю розуміти нюанси й тонкощі людської мови.
- **Сентиментальний аналіз.** Сентиментальний аналіз, підгалузь NLP, передбачає визначення настроїв або емоційного тону, що стоїть за серією слів. Це можна використовувати, щоб зрозуміти ставлення, думки та емоції, висловлені у відповідь на відкриті запитання. Однак аналіз настроїв іноді може не враховувати такі речі, як сарказм, заперечення та інші складні аспекти людської мови.
- **Змішаний підхід.** Змішаний підхід поєднує кількісні та якісні методи для аналізу відповідей на відкриті запитання. Цей підхід може забезпечити більш цілісне уявлення про дані, охоплюючи як широту (за допомогою

кількісних методів), так і глибину (за допомогою якісних методів) відповідей.

GPT є мовною моделлю з області NLP, тому вона швидко й ефективно аналізує великі набори даних. З усіх існуючих систем аналізу тексту вона є найбільш поширеною і потужною, що її виділяє. Використання GPT у системі оцінювання усних відповідей студентів дозволяє реалізувати гнучкіше та адаптивніше оцінювання. Модель може враховувати контекст, семантику, структуру та інші важливі аспекти відповідей, що забезпечує більш точне та об'єктивне оцінювання знань студентів.

РОЗДІЛ 2 ОГЛЯД НАЯВНИХ НА РИНКУ СИСТЕМ

На ринку існує декілька основних конкурентів системи оцінювання відповідей і ведення обліку робіт студентів, а саме Google Classroom[13] в комбінації з Google Forms[14], а також Microsoft Teams[15] з Microsoft Forms[16]. Для порівняння було обрано саме ті сервіси, що містять в собі функції, близькі до можливостей системи, що розробляється. Опишемо кожен з них з коротким переліком функціоналу.

- Опис функціоналу Google Classroom і Google Forms

У поєднанні **Google Classroom і Google Forms** пропонують високоінтерактивну та ефективну освітню платформу, яка полегшує як навчання, так і оцінювання.

Google Classroom є основою цієї комбінації, забезпечуючи цифрове середовище, де вчителі та учні можуть без проблем взаємодіяти між собою. Викладачі можуть створювати класи, запрошувати студентів, розподіляти завдання та спілкуватися за допомогою публікацій.

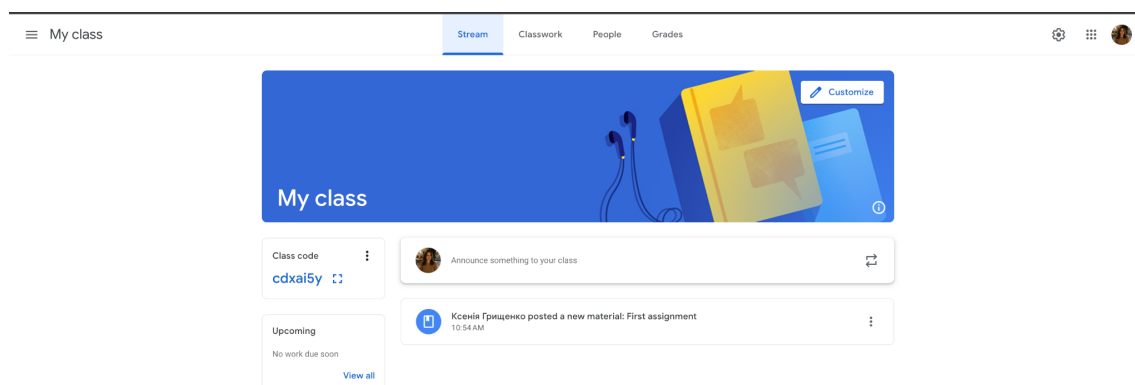
Система оцінювання в Google Classroom є гнучкою та легкою в управлінні. Викладачі можуть оцінювати роботу студентів та надавати відгуки по ній. Оцінки автоматично збираються в журнал оцінок, що забезпечує організований огляд успішності студентів.

Google Forms доповнює Google Classroom, пропонуючи можливість створювати власні тести та опитування. Цей інструмент особливо корисний для проведення оцінювання, оскільки вчителі можуть створювати тести з кількома варіантами відповідей, запитання з короткою або навіть обширною відповіддю. Google Forms підтримують автоматичне оцінювання для певних типів запитань, як-от множинний вибір або правда/неправда, що може заощадити вчителям значний час.

В комбінації з Google Classroom Google Forms стає ще потужнішим. Викладачі можуть поширювати опитування, створені в Google Forms, як завдання

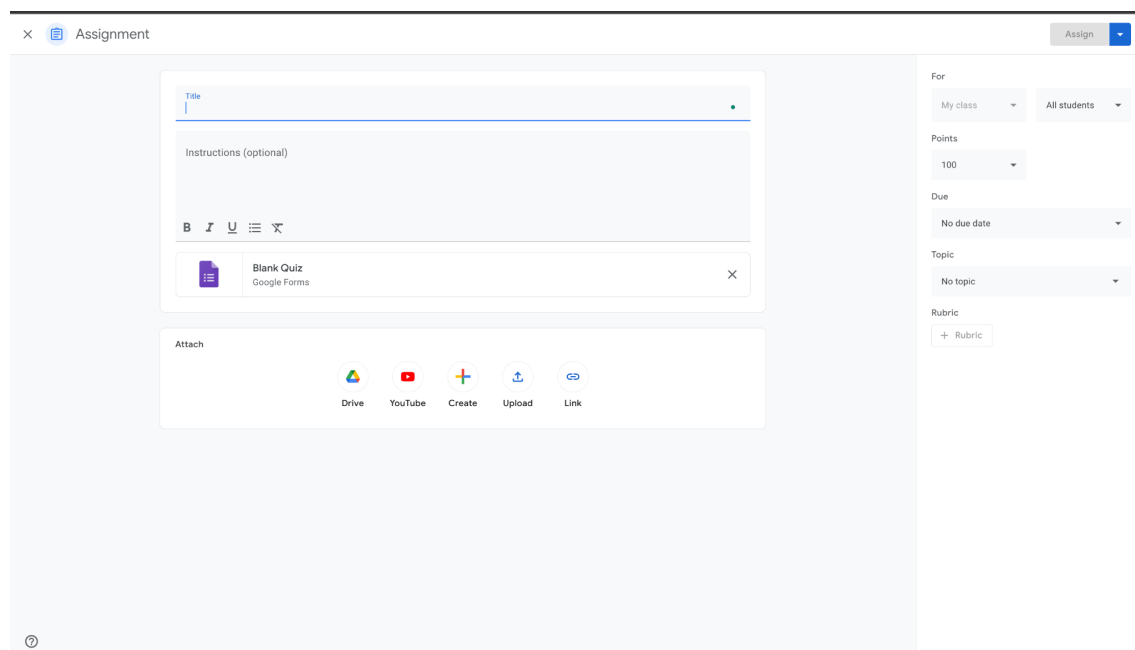
в Classroom. Після того, як учні пройдуть тест, їхні оцінки можна буде імпортувати безпосередньо в журнал заліків Classroom.

Отже, Google Classroom і Google Forms забезпечують комплексне цифрове освітнє середовище. Приклади інтерфейсу Google Classroom, та його взаємодію з Google Forms можна побачити на рисунках 3 і 4 відповідно.



©

Рисунок 3 - Вигляд інтерфейсу Google Classroom



©

Рисунок 4 - Приклад створення завдання в класі з прикріпленою формою Google Forms

- Опис функціоналу Microsoft Teams і Microsoft Forms

Поєднання **Microsoft Teams** і **Microsoft Forms** пропонує інтегровану, продуктивну та інтерактивну освітню платформу, яка спрощує як навчання, так і оцінювання.

Microsoft Teams об'єднує чат, відеозустрічі та спільну роботу над файлами, що робить його потужним інструментом у контексті освіти. Це дозволяє вчителям створювати спеціальні команди для своїх класів, де вони можуть розподіляти завдання, ділитися ресурсами та сприяти обговоренню.

Функція командних завдань є ефективним способом розподілу, відстеження та оцінювання роботи студентів. Викладачі можуть надавати відгуки безпосередньо про роботу студентів, виставляти оцінки для кожного студента, які потім автоматично організуються в журналі оцінок.

Microsoft Forms, подібно до Google Forms, є інструментом, який дозволяє викладачам створювати власні тести та опитування. Він підтримує різні типи запитань, включаючи кілька варіантів відповіді, текстові, рейтингові тощо. Аналогічно до Google Forms доступне автоматичне виставлення оцінок для певних типів питань.

Загалом Microsoft Teams і Microsoft Forms забезпечують надійне цифрове середовище для викладання та навчання. Приклад інтерфейсу Microsoft Teams представлений на рисунку 5, а Microsoft Forms на рисунку 6.

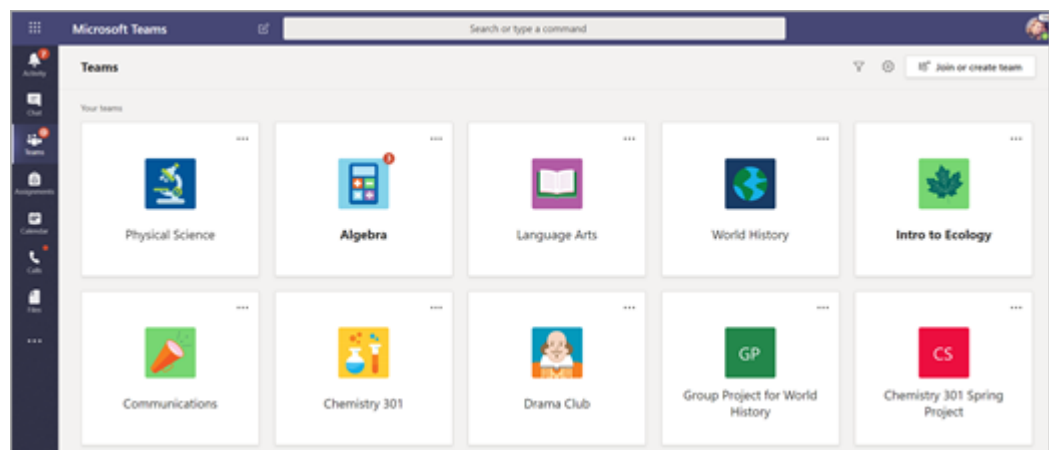


Рисунок 5 - Вигляд інтерфейсу Microsoft Teams

The image shows a screenshot of a Microsoft Forms page titled "Conference Schedule Sign Up". The page is displayed within a browser window with the address bar showing "Boston DXM > Office 365". The form content includes a green header with the title, a message: "Hi David, when you submit this form, the owner will be able to see your name and email address.", a "Required" indicator, and two questions:

- 1. Mobile Number ***
To be used during the conference for immediate communications
Input field: "Enter your answer"
- 2. Ideal time on Monday ***
Select all ideal times
List of time slots with checkboxes:
 - 8a-10a
 - 10a-12p
 - 12p-2p
 - 2p-4p

Рисунок 6 - Приклад форми в Microsoft Forms

На основі аналізу конкурентів було зроблено висновки, що на момент розробки жоден з них не пропонує готове рішення для автоматичного оцінювання відкритих відповідей. Така можливість суттєво скоротила би час на перевірку завдань студентів, що значно поліпшило і оптимізувало учбовий процес. Саме тому було прийнято рішення реалізувати цей функціонал в системі, що була розроблена в ході даної роботи із запозиченням деяких функцій від конкуруючих систем, таких як ведення обліку робіт студентів різних груп, призначення завдань, можливість надавати коментарі до роботи та управляти балами.

РОЗДІЛ 3 ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

3.1 Огляд технологій серверної частини

Як вже було зазначено раніше, для серверної частини було використано мову Python. Ця мова програмування є досить простою і зрозумілою у використанні, дозволяючи економити час на освоєнні її програмних засобів, при цьому пропонуючи потужні можливості в багатьох напрямках. Більше того, оскільки застосунок передбачає взаємодію з ШІ у вигляді GPT, то вибір цієї мови є дуже доречним, оскільки розробники даної моделі, OpenAI[17], пропонують офіційну бібліотеку для взаємодії з сервісом, яка називається openai[18]. Дана робота передбачає використання моделі GPT-4 четвертого покоління.

В якості IDE виступає PyCharm. Вона є однією з найбільш інтегрованих серед Python IDE, оскільки містить багато модулів та інструментів, які допомагають програмістам швидше та простіше писати код.

Мова Python пропонує багато фреймворків, які пристосовані відповідно до певних потреб та пришвидшують й спрощують реалізацію продукту. Оскільки дана система передбачає створення насамперед веб-застосунку, то для цього було обрано використання DRF. DRF (Django REST framework) - це потужний та гнучкий інструментарій для створення веб-застосунків з використанням фреймворку Django. Він дозволяє легко створювати API, що робить його популярним вибором для розробки високопродуктивних веб-застосунків з використанням Python та Django.

3.2 Огляд технологій клієнтської частини

Клієнтська частина передбачала використання широко відомої мови JavaScript. JavaScript - це мова програмування, яка використовується для створення динамічних веб-сторінок та взаємодії з користувачем на стороні

клієнта. Вона дозволяє змінювати зовнішній вигляд веб-сторінки, взаємодіяти з користувачем та виконувати різноманітні завдання на стороні клієнта.

Інструментом для програмування мовою JavaScript виступив Visual Studio Code. Він є дуже легким та швидким текстовим редактором з багатьма розширеннями, які дозволяють підлаштувати його під свої потреби.

Аналогічно існує багато фреймворків для мови JavaScript. В даній роботі було використано фреймворк Vue. Vue - це прогресивний фреймворк для створення інтерфейсів користувача з підтримкою двостороннього зв'язку даних, який дозволяє легко розширювати та налаштовувати компоненти і додатки. До переваг Vue можна віднести швидкодію, простоту використання та велику кількість додаткових плагінів та розширень.

В якості GUI виступала бібліотека Element UI. Element UI - це бібліотека готових компонентів інтерфейсу користувача для Vue.js, яка дозволяє легко створювати стильні та інтерактивні веб-додатки з мінімальними зусиллями. Бібліотека містить багато готових компонентів, таких як кнопки, таблиці, форми тощо, що робить розробку веб-інтерфейсу швидшою та простішою. Серед переваг можна виділити широкий вибір готових компонентів, гнучкість та простоту налаштування.

3.3 Огляд БД

В якості БД в даній роботі виступає PostgreSQL. Це потужна та надійна система управління базами даних (СУБД), яка має високий рівень безпеки та забезпечує ефективну роботу з великими обсягами даних. Вона має багато корисних функцій, таких як транзакції, гнучке керування доступом, розширюваність та масштабованість, що робить її популярним вибором для багатьох веб-додатків та програмних проектів.

РОЗДІЛ 4 ПРИЗНАЧЕННЯ ТА АРХІТЕКТУРА СИСТЕМИ

4.1 Призначення системи

Дана система призначена для автоматизації процесу перевірки усних відповідей студентів, що дає змогу економити час, відведений на перевірку з метою оптимізації учбового процесу і перерозподілу цього ресурсу на інші задачі, що більше потребують безпосередньої уваги та участі викладачів, а також для надання зручного інтерфейсу для ведення навчального процесу, управління завданнями та роботами студентів.

4.2 Архітектура системи

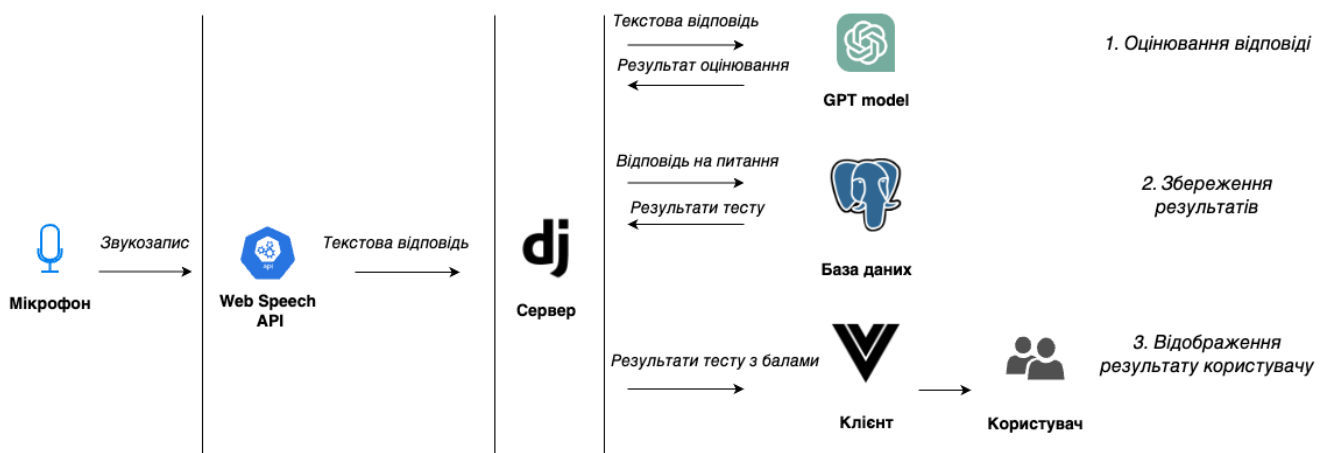


Рисунок 7 - Архітектура системи

На рис. 7 можна побачити з чого складається дана система та в якому порядку відбувається взаємодія окремих її частин. А саме:

- Ввід відповіді за допомогою мікрофону.
- Перетворення мовлення в текст за допомогою Web Speech API.
- Відправка текстової відповіді на сервер.
- Запит серверу до моделі GPT для отримання результатів оцінювання наданої відповіді.

- Запис результату в базу даних.
- Отримання результатів всього тесту після обробки всіх відповідей з відповідно розрахованими балами.
- Відправка результатів клієнту.
- Показ результатів користувачу.

РОЗДІЛ 5 ОСОБЛИВОСТІ РОЗРОБКИ

5.1 Сутності системи

Для того, щоб розуміти специфіку системи варто розуміти, з яких сутностей (класів) вона складається. Діаграму класів наведено в додатку А.

Для зручності оперування поняттями випишемо їх словник:

- Користувач - User;
- Предмет - Subject;
- Шаблонний тест - Quiz;
- Питання тесту - Question;
- Курс - Course;
- Завдання - Task;
- Сесія завдання - Task Session;
- Відповідь користувача - User Answer.

Одна з ключових одиниць системи - користувач. Він характеризується такими рисами як: ім'я, прізвище, електронна пошта.

Оскільки передбачається робота зі студентами певного навчального середовища, то інші важливі сутності це предмет та курс. Опишемо детальніше кожну з них та їх взаємозв'язок.

Предмет - одиниця навчального процесу, що характеризується певною назвою, та пов'язаними викладачами. Предмет може містити набір шаблонних тестів, які можуть перевикористовуватись для різних курсів. При створенні предмету генерується запрошувальне посилання, по якому викладачі можуть приєднатись до нього, щоб мати права вчителя в курсах, пов'язаних з ним та можливість створювати або користуватись тестами, що прив'язані до нього.

Шаблонний тест - набір питань на пов'язану тему, що відноситься до певного предмету. Описується певною назвою та максимально можливою тривалістю проходження.

Питання тесту - функціональна одиниця, що характеризується заголовком, очікуваною відповіддю та кількістю балів, в яке воно оцінюється.

Курс - одиниця навчального процесу, що має певну назву, та обов'язково прив'язана до певного існуючого предмету. Курс містить користувачів, що є його учасниками та може містити набір завдань, призначених для них. При створенні курсу генерується запрошувальне посилання, по якому будь-який користувач може приєднатись до нього і автоматично отримати роль студента. За потребою, власник цього курсу може змінювати права учасників та надавати їм роль вчителя.

Завдання - тест для курсу, створений з певного шаблонного тесту, що має певну назву. Воно призначається для всіх студентів курсу і стає доступним з моменту його створення і до закінчення можливого терміну виконання, що задається при створенні. Оскільки завдання створюється з шаблонного тесту, то воно містить множину питань, на які студент має дати відповіді.

Сесія завдання - факт проходження студентом певного завдання. Містить дату початку і дату кінця.

Відповідь користувача - відповідь на певне питання завдання, що характеризується текстом, з боку студента й результатом перевірки, а саме відсотком правильності й поясненням до відповіді. Також вона містить дані про те, чи була відповідь редагована вчителем.

5.2 Опис функціоналу

Система передбачає систему доступів з трьох ролей: власник курсу, вчитель, студент і доступна для використання тільки авторизованим користувачам і обмежена функціями, відповідно до ролі, яку має користувач. Розглянемо детальніше можливості кожної з них.

- Опис функціоналу вчителя (власника курсу)

Роль вчителя передбачає доступ до всіх можливостей системи. Перерахуємо підсистеми, що доступні користувачу:

а) Особистий профіль:

- 1) зміна особистої інформації - пароллю, електронної пошти, імені, фото профілю

б) Підсистема предметів:

- 1) приєднання за запрошувальним посиланням
- 2) створення предмету
- 3) отримання запрошувального посилання на предмет
- 4) доступ до інформації про предмет - викладачів, назви
- 5) можливість редагувати назву
- 6) створення шаблонних тестів, прив'язаних до предмету
- 7) видалення предмету

в) Підсистема шаблонних тестів:

- 1) створення тесту
- 2) видалення тесту
- 3) редагування інформації про тест - назви та максимальної тривалості проходження
- 4) додавання питань в тест

г) Підсистема питань тесту:

- 1) створення питання
- 2) редагування питання - назви, очікуваної відповіді, оцінки за нього
- 3) видалення питання

д) Підсистема курсів:

- 1) приєднання за запрошувальним посиланням
- 2) створення курсу
- 3) отримання запрошувального посилання на курс
- 4) доступ до інформації про курс - назви, пов'язаного предмету, переліку учасників курсу з їх ролями
- 5) можливість змінити роль користувача, у випадку ролі власника курсу
- 6) видалення курсу

е) Підсистема завдань:

- 1) створення завдання з переліку шаблонних тестів предмету, що пов'язаний з курсом
 - 2) редагування інформації про завдання - назви та кінцевої дати виконання
 - 3) видалення завдання
 - 4) перегляд результатів виконання завдання студентами
- ж) Підсистема відповідей студентів:
- 1) перегляд відповіді
 - 2) редагування оцінки студента та коментарю до оцінки

З перерахованим функціоналом можна також ознайомитись в додатку Б на діаграмі прецедентів.

- Опис функціоналу студента

Підсистеми, що доступні студенту:

а) Особистий профіль:

- 1) зміна особистої інформації - паролю, електронної пошти, імені, фото профілю

б) Підсистема курсів:

- 1) приєднання до курсу за запрошувальним посиланням
- 2) перегляд доступних курсів
- 3) перегляд інформації про курс
- 4) перегляд призначених завдань

в) Підсистема завдань:

- 1) початок проходження завдання
- 2) запис відповідей
- 3) закінчення проходження завдання
- 4) перегляд результату

г) Підсистема відповідей студента:

- 1) ввід відповіді за допомогою мікрофону
- 2) редагування відповіді не більше, ніж на 20% у випадку неточностей розпізнавання мовлення

3) перезапис відповіді

Всі перераховані можливості також зображено на діаграмі прецедентів, зображеної на рисунку 8.

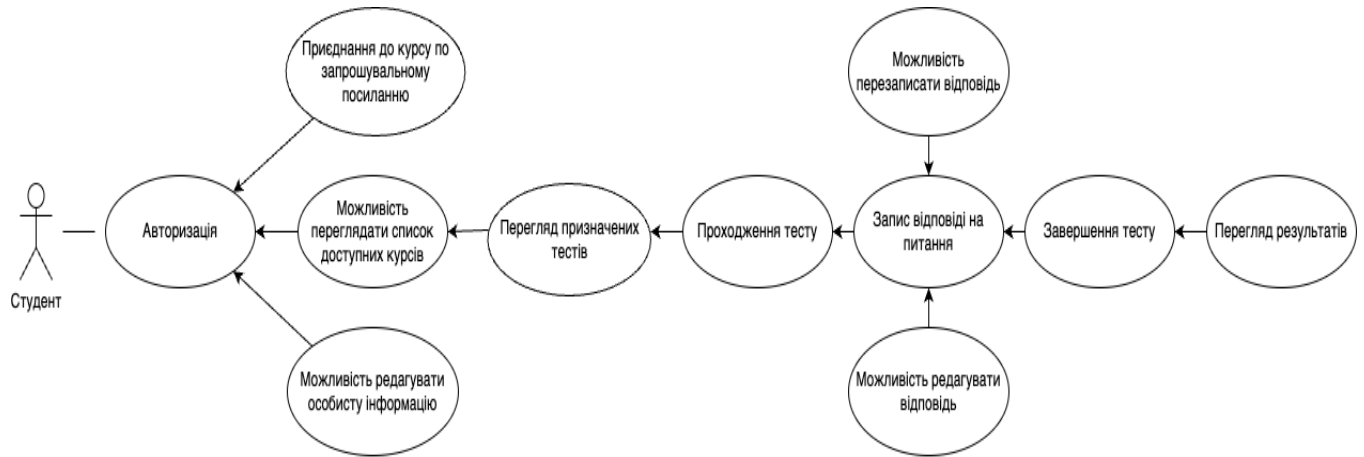


Рисунок 8 - Use-case діаграма для ролі студента

5.3 Деталі проходження тесту з відкритими відповідями

На момент розробки система Sigma School передбачає тести лише з відкритими питаннями. Для того, щоб зробити процес проходження тесту максимально прозорим, для запису відповіді можливий лише голосовий ввід за допомогою мікрофона. Перейшовши на сторінку певного питання студент може ввімкнути звукозапис й продиктувати свою відповідь. Далі вона розпізнається за допомогою Web Speech API й зображується на екрані.

Студенту надається можливість відредагувати текст, що був перетворений з усного мовлення, але не більше ніж на 20%. Це пов'язано з тим, що можуть бути певні неточності в ході трансформації мовлення, тому таким чином доступна мінімальне коректування, щоб унеможливити вставку відповіді з якимось письмового джерела. Схожість редагованого тексту перевіряється за допомогою відстані Левенштейна[19].

Після проходження всього тесту відповідь надсилається на сервер, де потім перевіряється GPT моделлю. По закінченню перевірки студент може побачити свої результати.

РОЗДІЛ 6 РОБОТА З СИСТЕМОЮ

Для того, щоб продемонструвати роботу з системою Sigma School опишемо кожен модуль, що вона вміщує.

- Авторизація та реєстрація

Перший екран, що бачить користувач при вході в систему - сторінка реєстрації. Для реєстрації потрібно заповнити поля email-у, імені, прізвища, та паролю. На всі поля передбачена валідація вводу, а саме на довжину та формат для email-у. На рисунку 9 зображено вигляд сторінки реєстрації з заповненими полями.

SIGMA SCHOOL

Зареєструйтесь, щоб продовжити

Email
teacher1@test.com

Ім'я
Іван

Прізвище
Петренко

Пароль
...
Мінімальна довжина паролю - 5 символів

Зареєструватись

Вже зареєстровані? Увійти

Рисунок 9 - Сторінка реєстрації

У випадку, якщо користувач уже є в системі, він може перейти на сторінку авторизації натиснувши на кнопку “Увійти” (рис. 10).

Зареєструватись

Вже зареєстровані? Увійти

Рисунок 10 - Перехід на сторінку авторизації

Для входу в систему потрібно заповнити поля email-у та паролю. Аналогічно, можна перейти на сторінку реєстрації натиснувши на кнопку “Зареєструватись” (рис. 11).

Рисунок 11 - Сторінка авторизації

- Навігація

Доступна бокова навігація для переходу між підсистемами предметів та курсів (рис. 12).

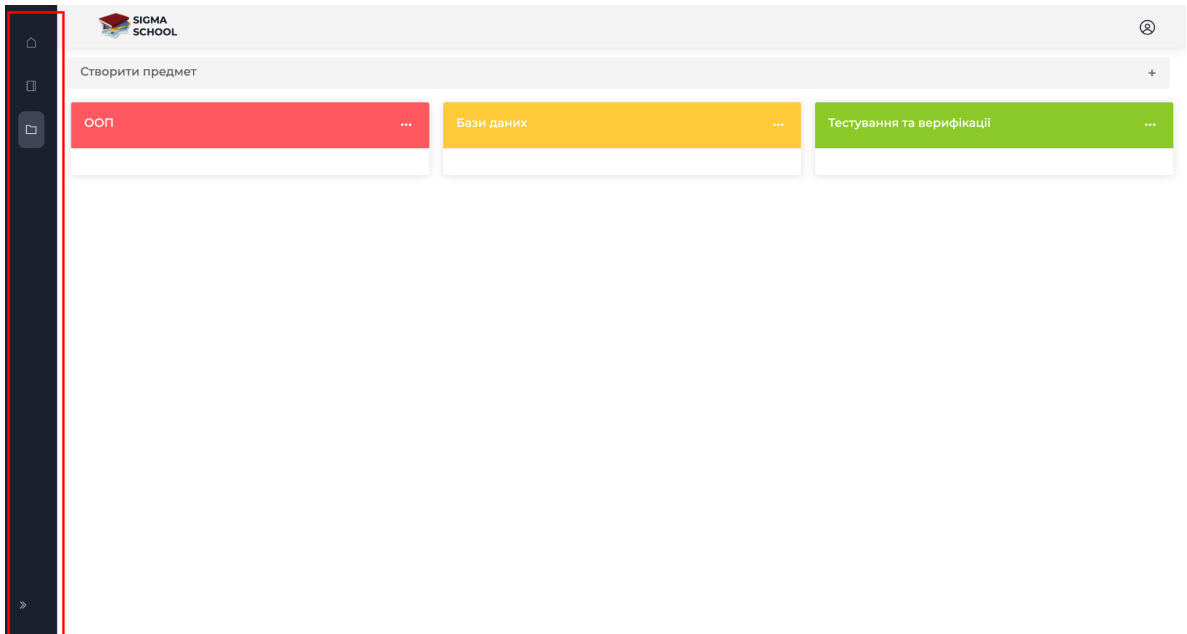


Рисунок 12 - Бокова навігація

- Сторінка предметів

На цій сторінці можна побачити перелік доступних предметів у вигляді карток. Також, можна створити предмет, ввівши його назву (рис. 13).

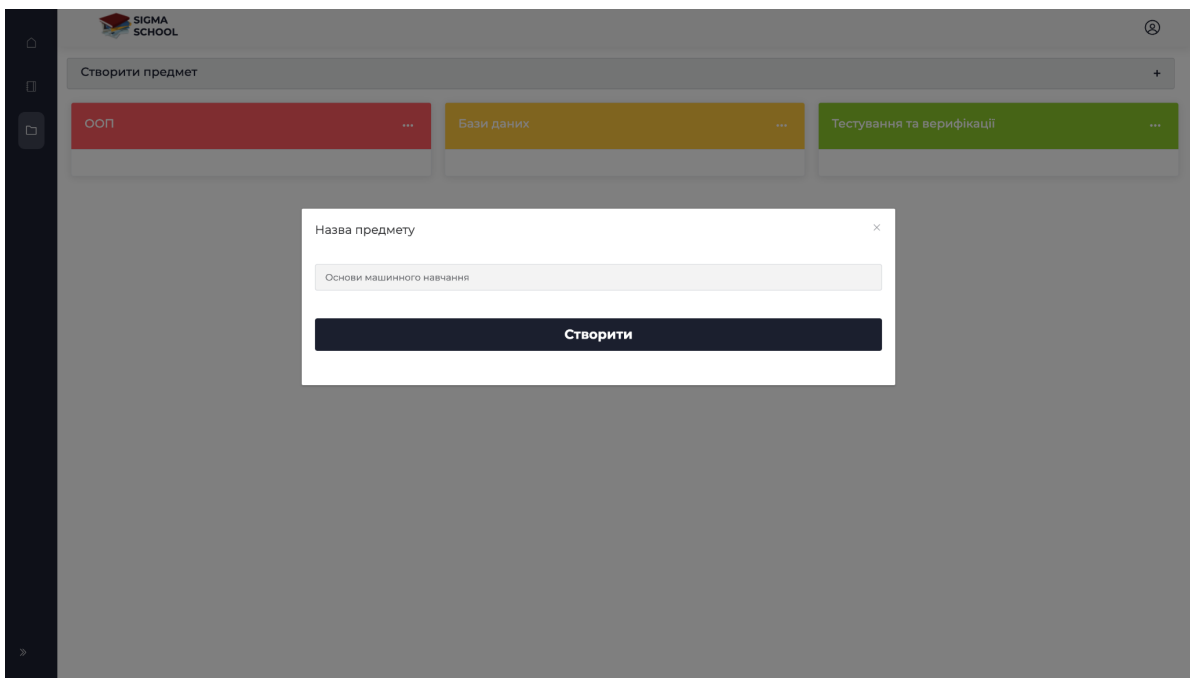


Рисунок 13 - Модальне вікно для створення предмету

При створенні одразу відкривається модальне вікно із запрошувальним посиланням для приєднання інших вчителів (рис. 14).

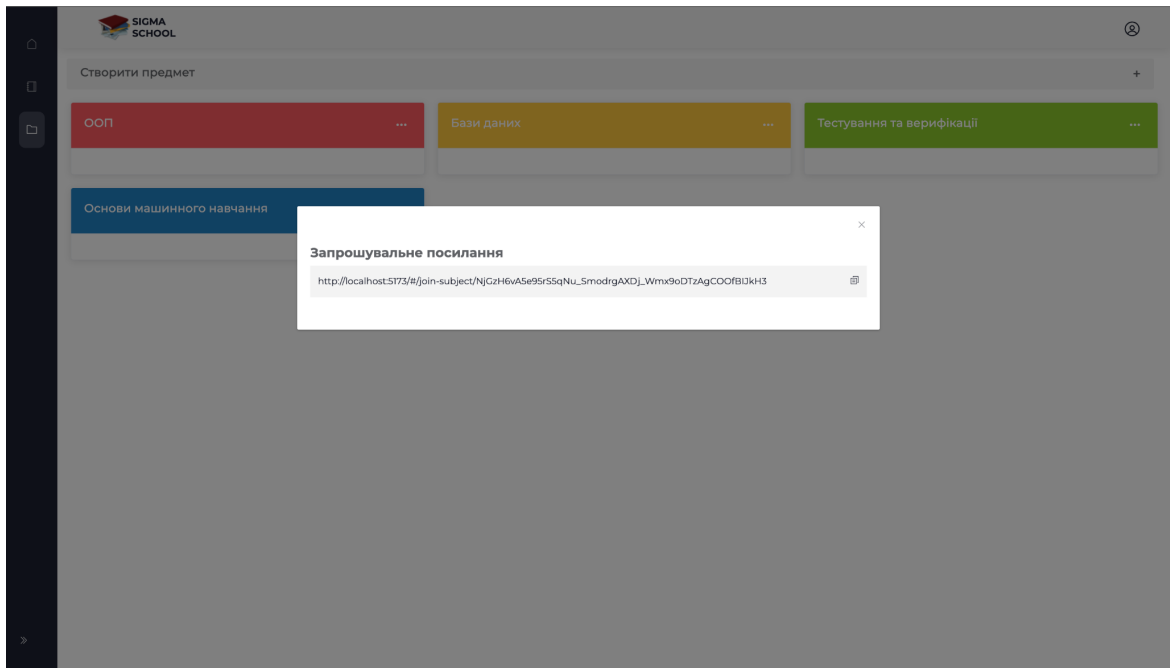


Рисунок 14 - Модальне вікно із запрошувальним посиланням

При натисканні на три крапки в правому верхньому куті картки предмету відкривається модальне вікно з детальною інформацією - назвою і запрошувальним посиланням, яке можна скопіювати по кліку на нього. Також передбачена можливість видалення предмету (рис. 15).

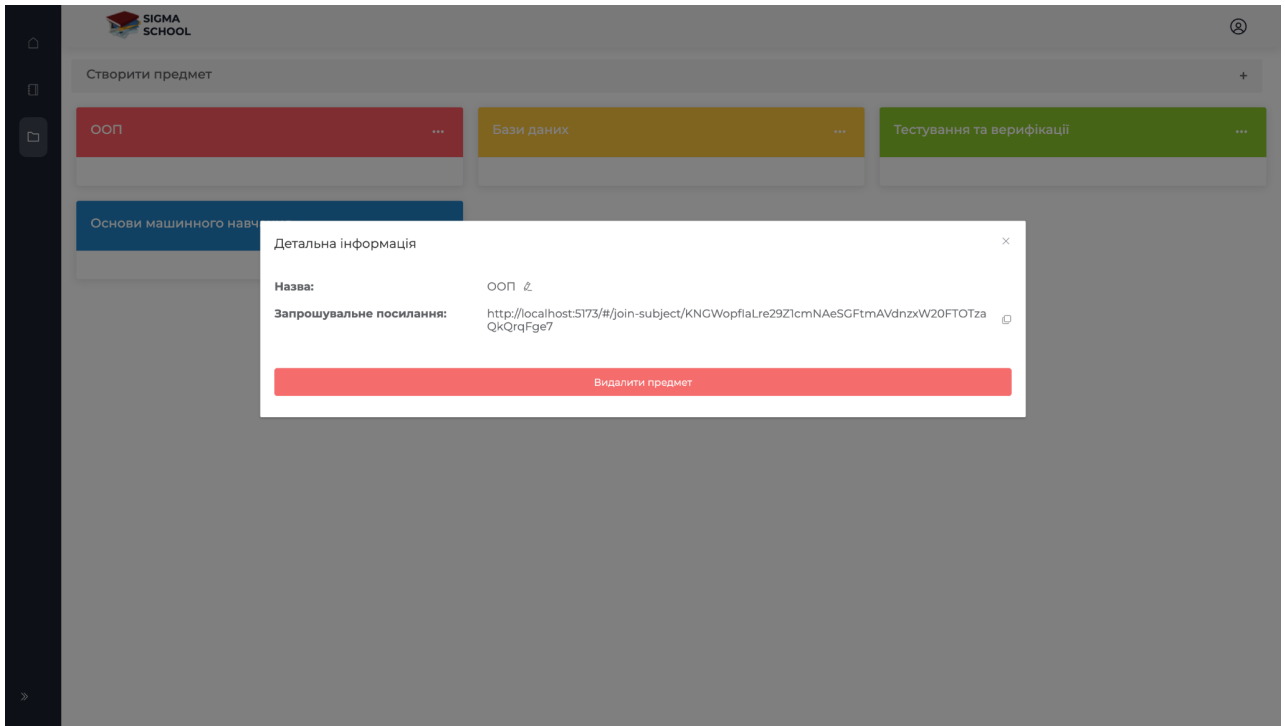


Рисунок 15 - Модальне вікно з детальною інформацією про предмет

При натисканні на картку предмету відбувається перехід на його сторінку з детальною інформацією та тестами.

- Сторінка окремого предмету

При переході на сторінку предмету в верхньому меню навігації з'являється дві вкладки - "Тести" та "Інформація", а також назва предмету в лівому куті.

Першою автоматично відкривається сторінка тестів предмету. Тут можна створити новий тест, ввівши його назву та максимальну тривалість проходження (рис. 16).

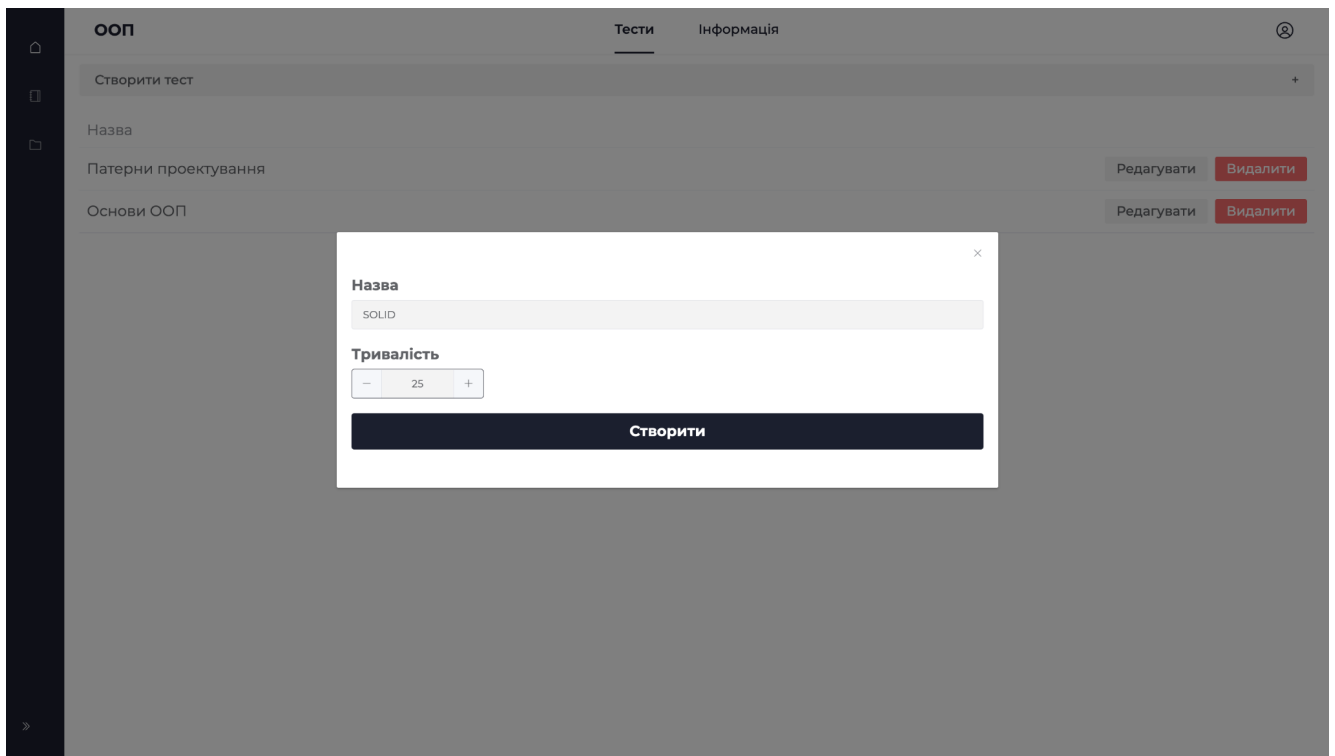


Рисунок 16 - Модальне вікно для створення тесту

Для кожного тесту доступно дві можливості - редагування інформації та видалення, що можна здійснити використовуючи кнопки “Редагувати” та “Видалити”.

Редагування тесту передбачає зміну назви або тривалості проходження. Відкривається модальне вікно, що аналогічне тому, що при створенні тесту.

При видаленні з’являється модальне вікно з підтвердженням операції (рис. 17).

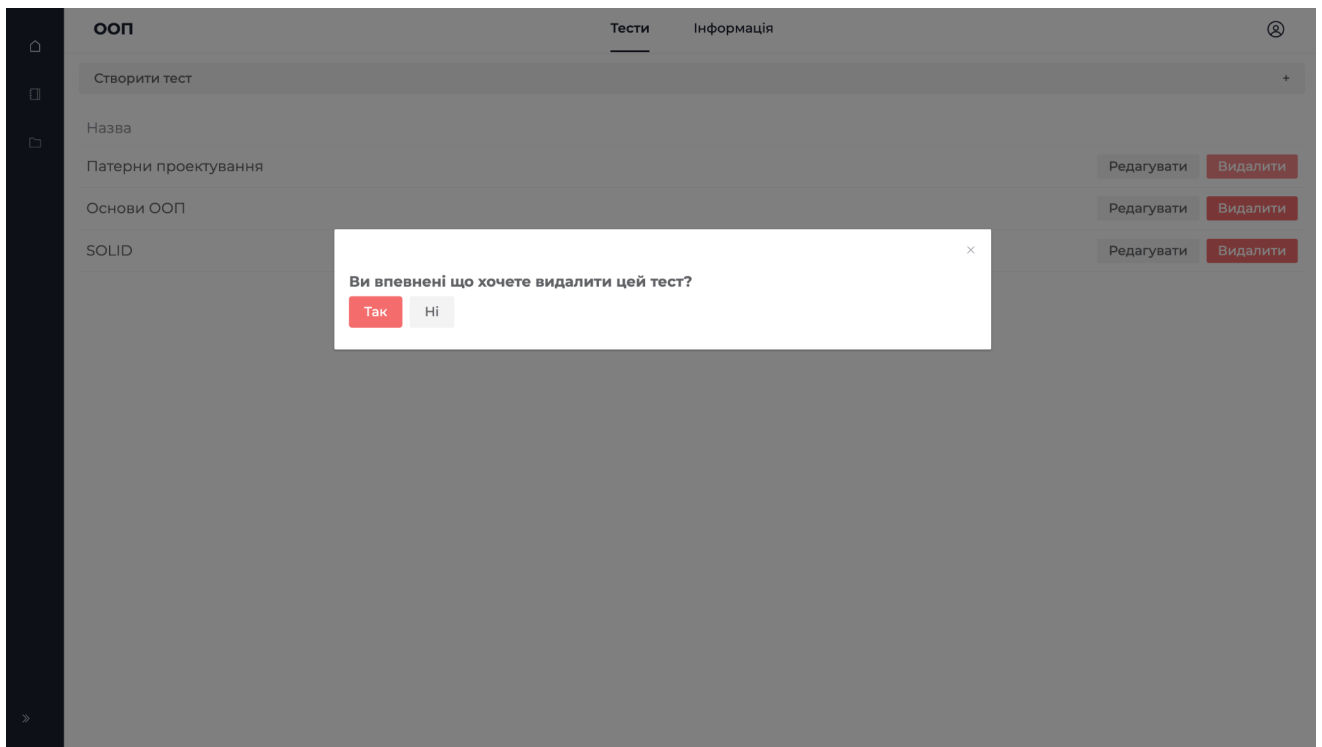


Рисунок 17 - Модальне вікно з видаленням тесту

На сторінці інформації доступні всі деталі про предмет - назва, перелік вчителів та запрошувальне посилання (рис. 18).

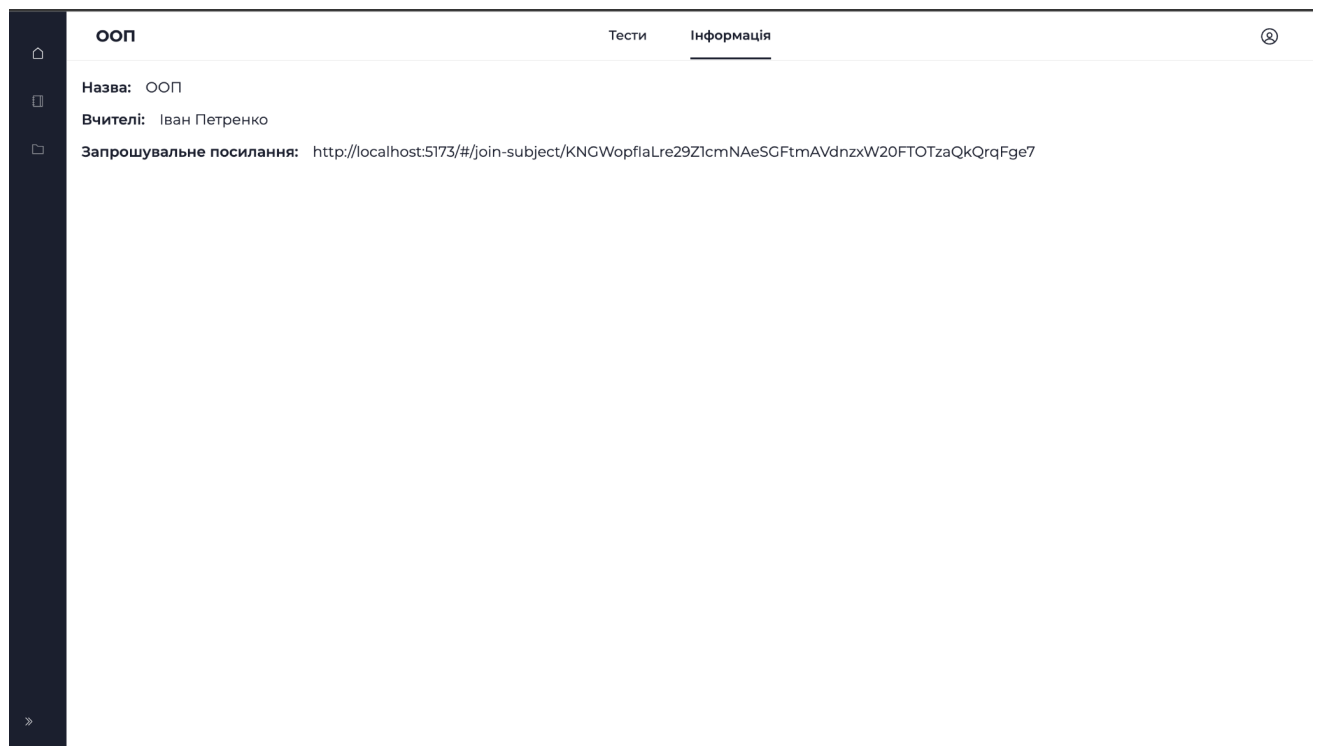


Рисунок 18 - Сторінка інформації про предмет

- Сторінка тесту

Перехід на сторінку тесту можливий при натисканні на елемент таблиці в списку всіх тестів предмету.

Користувачу доступні всі питання та можливість додати нове. Для цього потрібно ввести назву питання, очікувану відповідь та максимально можливий бал, що може даватись за нього (рис. 19).

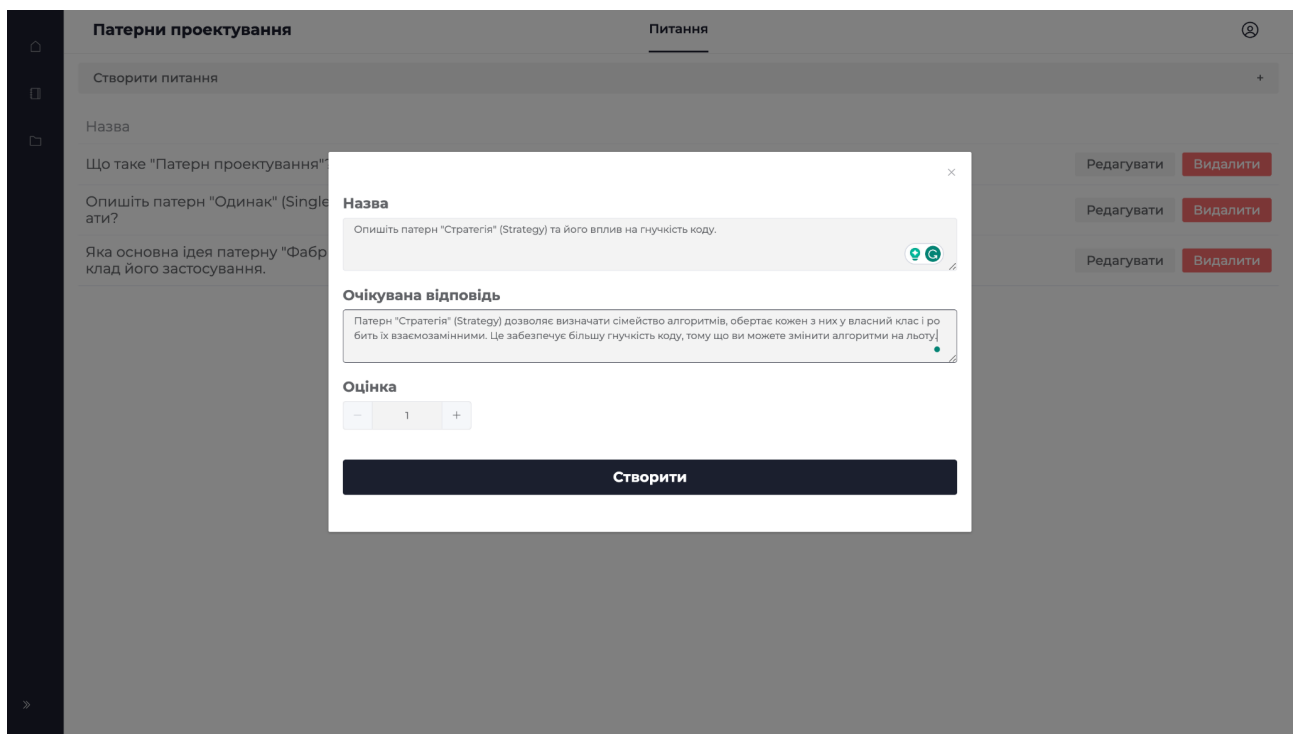


Рисунок 19 - Модальне вікно для створення питання

Питання можна редагувати або видалити, використовуючи відповідні кнопки. При редагуванні з'являється аналогічно модальне вікно, що й при створенні. При видаленні з'являється вікно для підтвердження операції.

- Сторінка курсів

Використовуючи бокову навігацію можна потрапити до сторінки курсів.

Вона представляє собою перелік всіх наявних курсів, а також можливість створити новий. Для цього потрібно задати його назву та обрати прив'язаний предмет (рис. 20).

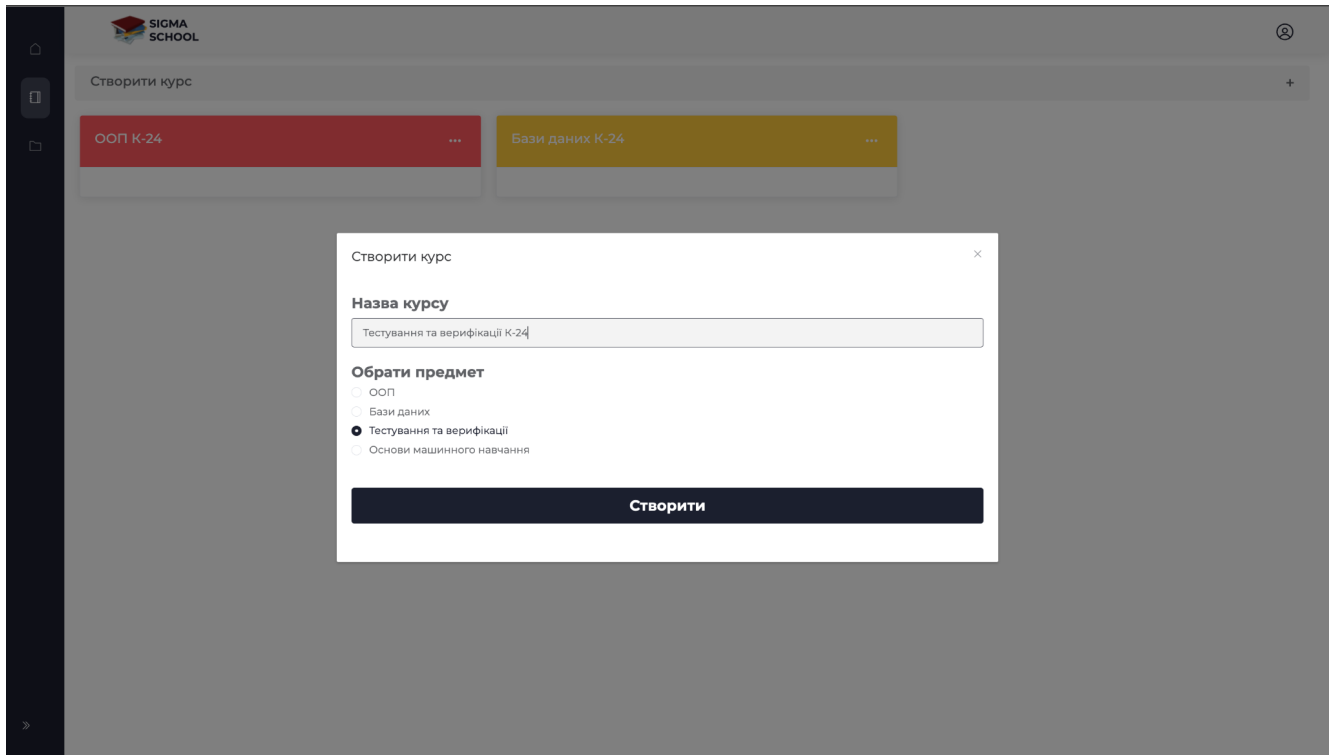


Рисунок 20 - Модальне вікно для створення курсу

Одразу після створення відкривається модальне вікно з запрошувальним посиланням для приєднання на курс, яке можна одразу скопіювати (рис. 21).

Посилання на курс може бути розісланим як студентам, так і викладачам. При приєднанні користувач автоматично отримує роль студента, однак це може бути змінено на детальній сторінці курсу.

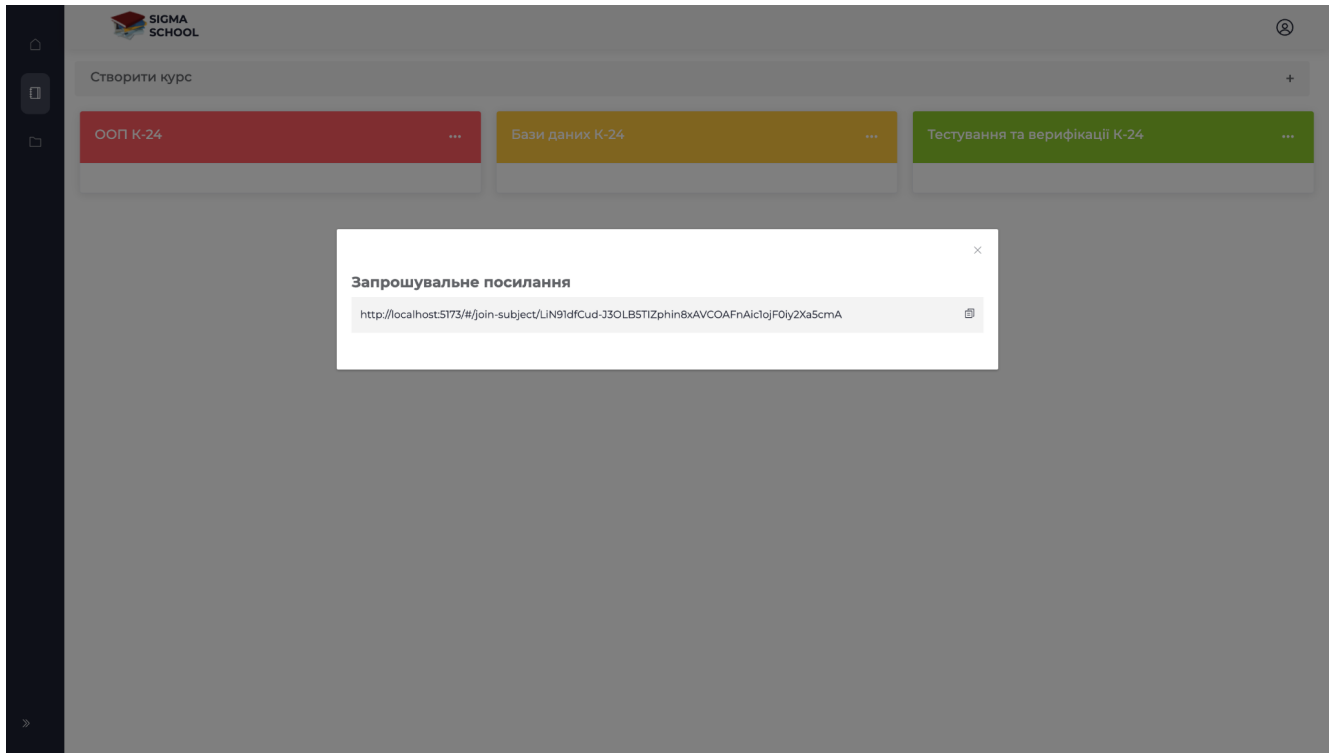


Рисунок 21 - Модальне вікно із запрошувальним посиланням

По кожному курсу доступний перегляд детальної інформації при натисканні на три крапки в правому верхньому куті картки. Можна побачити назву, предмет, до якого прив'язаний курс, а також запрошувальне посилання. Прямо в цьому модальному вікні можна відредагувати назву курсу, або видалити його (рис. 22).

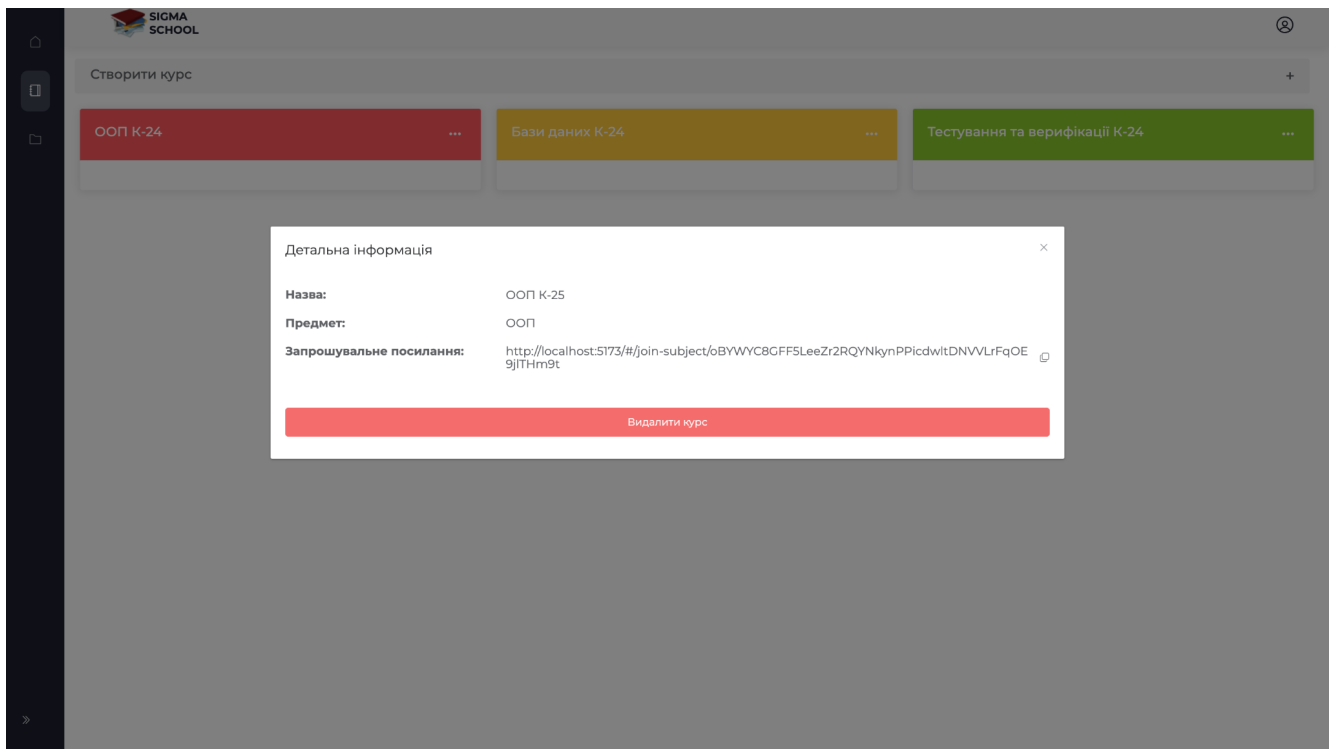


Рисунок 22 - Модальне вікно з інформацією про курс

При натисканні на картку курсу відбувається перехід на детальну сторінку курсу.

- Сторінка окремого курсу

За замовчуванням відображається список завдань для даного курсу. Також передбачена можливість створити нове завдання. Для цього потрібно ввести його назву, кінцеву дату проходження та обрати один тест зі списку шаблонних тестів, що відносяться до пов'язаного предмету (рис. 23).

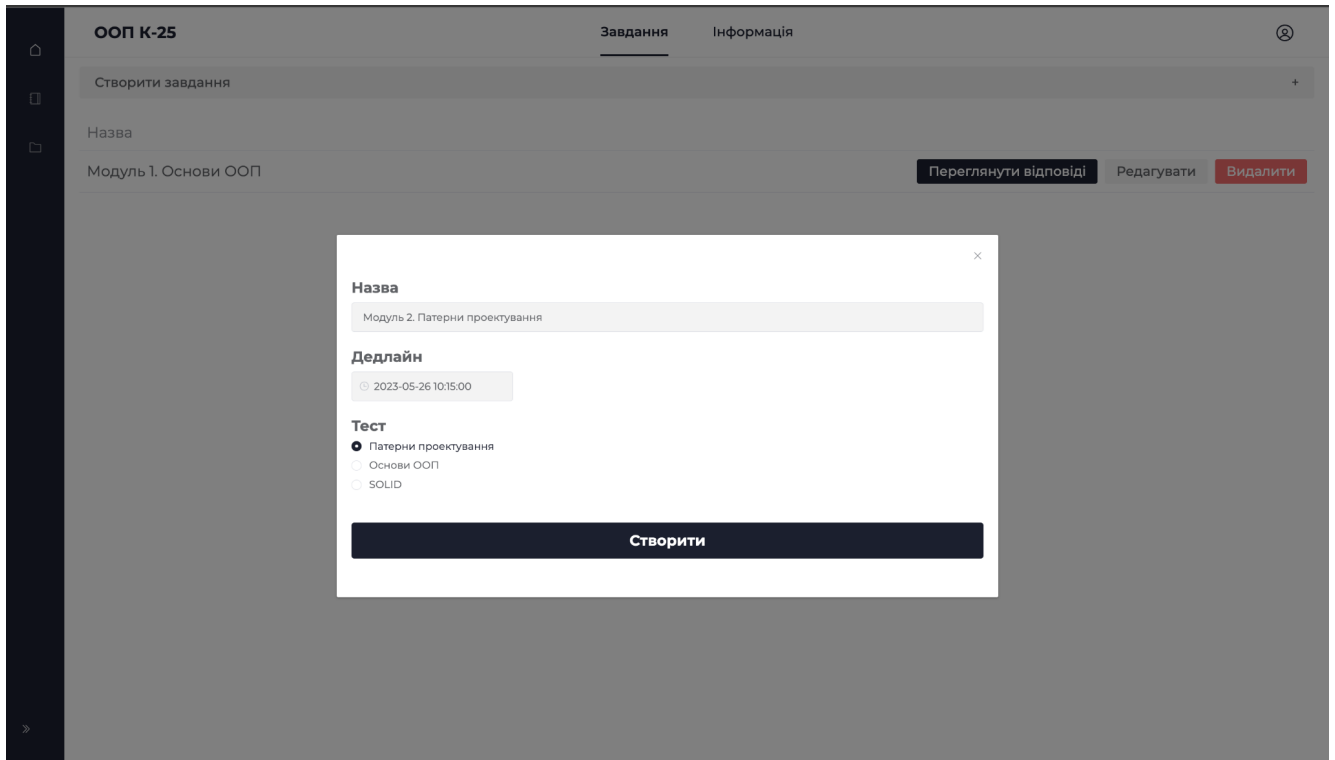


Рисунок 23 - Модальне вікно для створення завдання

Також вчителю доступні можливості перегляду відповідей по даному завданню, редагуванню або видаленню. Редагування передбачає зміну назви або кінцевої дати виконання. При видаленні з'являється вікно з підтвердженням операції.

У верхньому меню навігації міститься ще одна вкладка - перегляд інформації по курсу. Там містяться назва, прив'язаний предмет, запрошувальне посилання та список користувачів. Власник курсу може також змінювати права користувачів (рис. 24).

The screenshot shows a web interface for a course. At the top, there are two tabs: 'Завдання' (Tasks) and 'Інформація' (Information), with 'Інформація' being the active tab. The page title is 'ООП К-25'. Below the title, there are three fields: 'Назва: ООП К-25', 'Предмет: ООП', and 'Запрошувальне посилання: http://localhost:5173/#/join-course/oBYWYC8GFF5LeeZr2RQYNkynPPicdwitDNVVLrFqOE9JlTHm9t'. Under the heading 'Користувачі:', there is a table with three columns: 'Ім'я', 'Електронна пошта', and 'Роль'. The table contains two rows of user data. A red button labeled 'Зробити вчителем' is located to the right of the second row.

Ім'я	Електронна пошта	Роль
Іван Петренко	teacher1@test.com	Власник
Сергій Довженко	student1@test.com	Студент

Рисунок 24 - Сторінка з інформацією про курс

- Сторінка завдання

Натиснувши на завдання зі списку завдань, можна перейти на його сторінку. Якщо завдання активне, тобто його термін виконання ще не пройшов, то з'явиться кнопка розпочати тест (рис. 25).

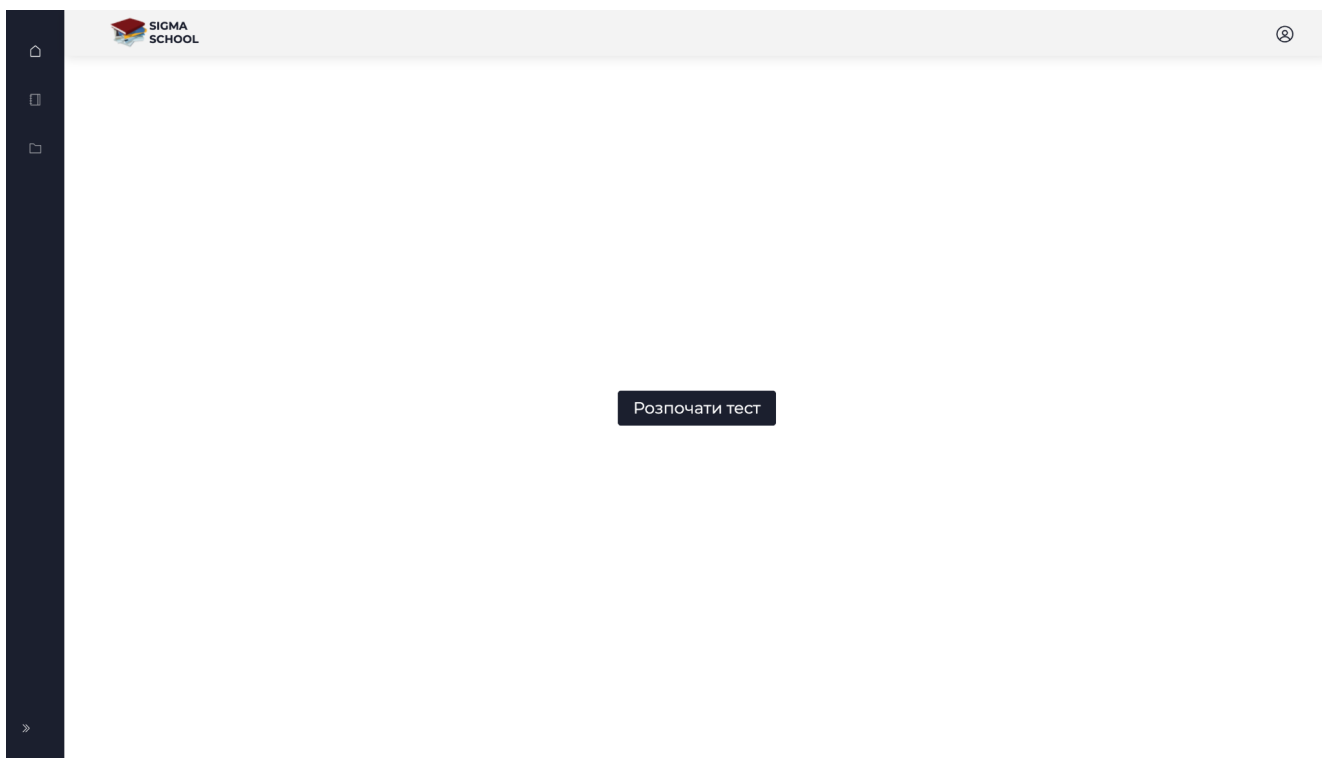


Рисунок 25 - Сторінка завдання в початковому стані

Після того, як кнопка натиснута, з'являється список питань цього тесту. Вгорі міститься таймер, який відраховує кількість часу, що залишилась на виконання. Бокова панель містить список питань для навігації між ними, також навігація може відбуватись за допомогою стрілок (рис. 26).

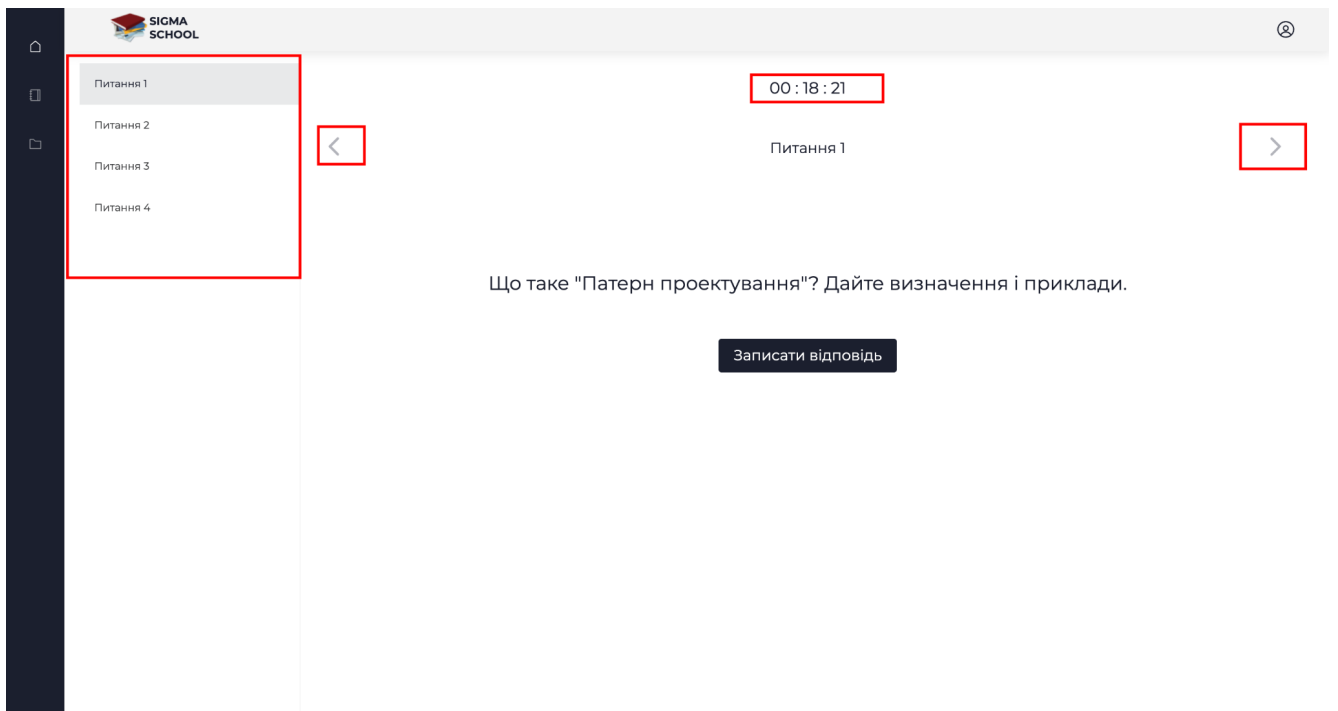


Рисунок 26 - Сторінка тесту під час проходження

Кожне питання містить можливість записати відповідь. При натисканні на кнопки в браузері з'являється сповіщення з запитом на дозвіл використання мікрофону (рис. 27). Щоб записати відповідь потрібно надати доступ.

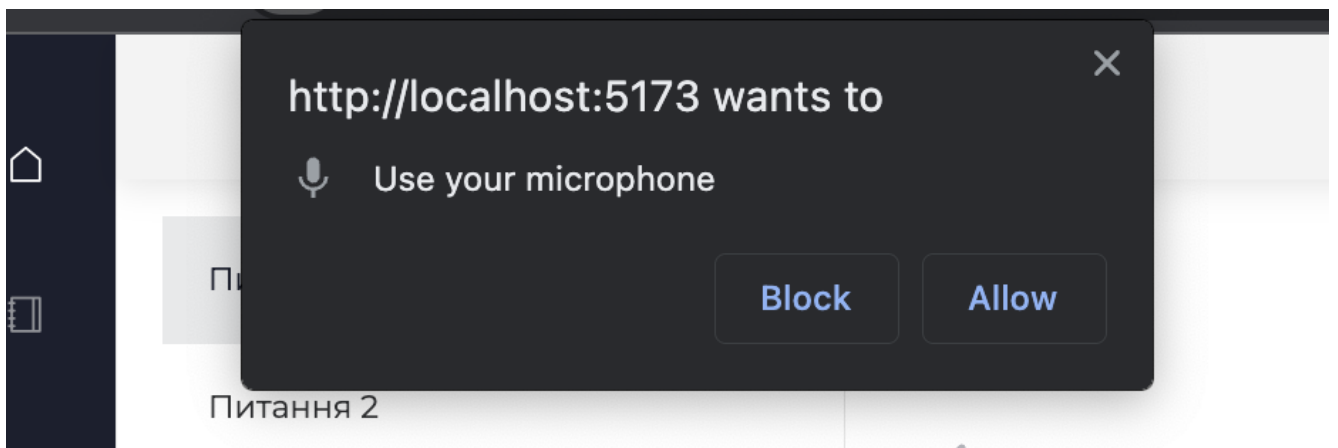


Рисунок 27 - Запит на дозвіл мікрофону

Після того, як відповідь була записано усно на екрані з'являється її текстове представлення (рис. 28).

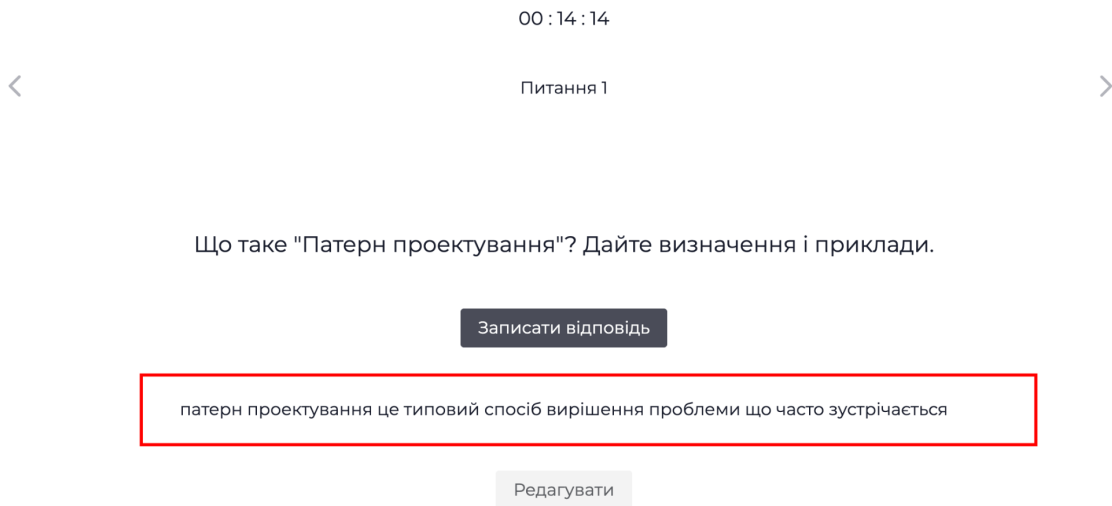


Рисунок 28 - Представлення записаної відповіді у вигляді тексту

На сторінці з останнім питанням тесту з'являється кнопка "Завершити тест" (рис. 29).

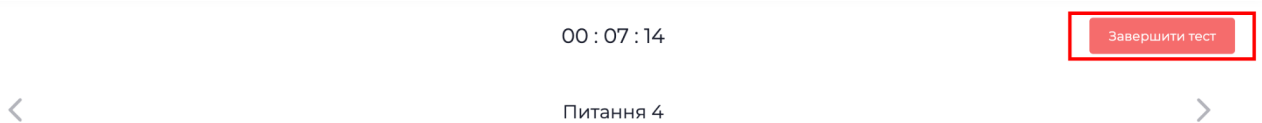


Рисунок 29 - Кнопка завершення тесту

Після завершення результати відправляються на сервер, де потрібен певний час для їх перевірки. Приклад оціненої роботи зображений на рисунку 30. Для кожної відповіді надані пояснення щодо коректності та зароблений бал.

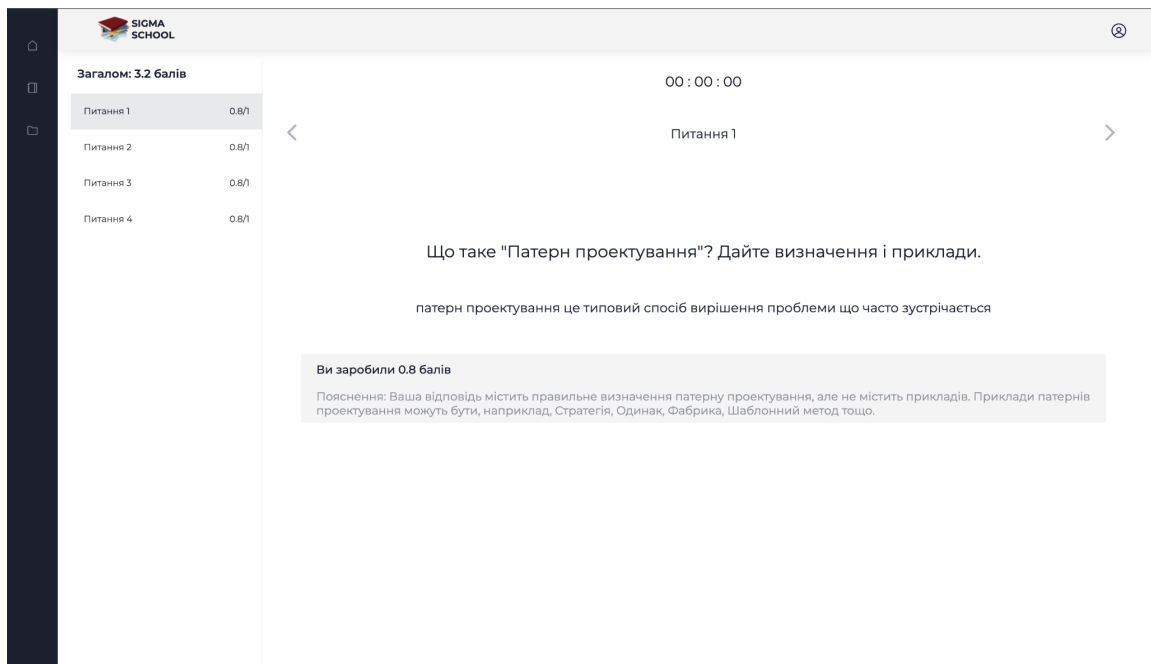


Рисунок 30 - Приклад оціненої роботи

- Особистий профіль

Перехід на сторінку особистого профілю можливий через верхнє меню. Там міститься фото користувача, або іконка за замовчуванням. При натисканні на неї з'являється випадаюче меню (рис. 31). Також є кнопка для виходу.

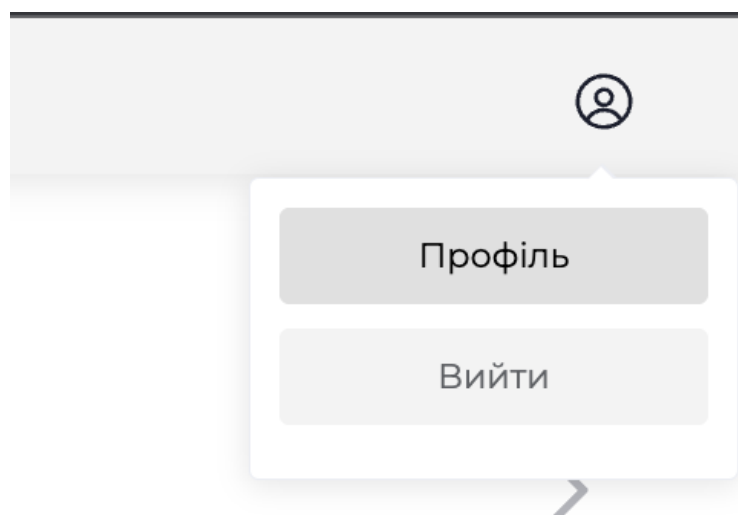
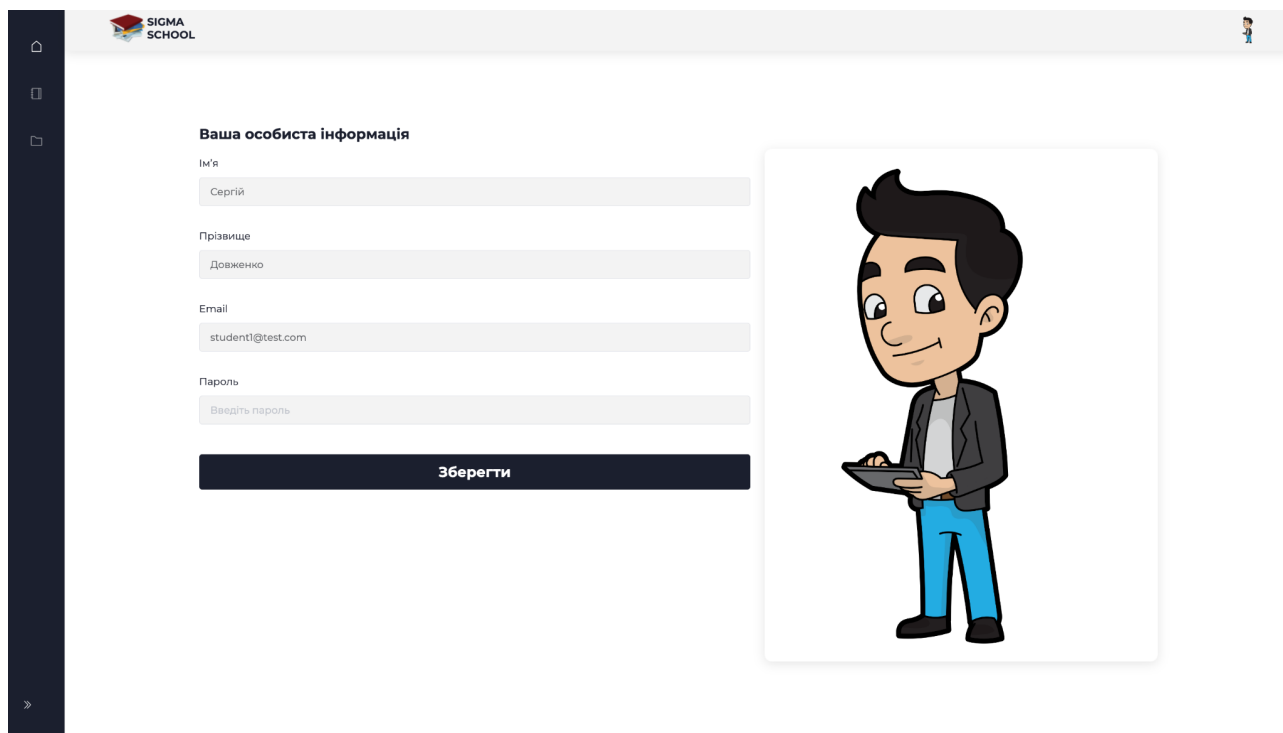


Рисунок 31 - Випадаюче меню

На сторінці профілю можливо змінити особисту інформацію - ім'я, прізвище, електронну пошту, пароль або фото (рис. 32).



The screenshot displays the 'Ваша особиста інформація' (Your personal information) section of a user profile. It features a dark sidebar on the left with navigation icons. The main content area has a header with the 'SIGMA SCHOOL' logo and a user icon. Below the header, there are four input fields for personal information: 'Ім'я' (Name) with the value 'Сергій', 'Прізвище' (Surname) with 'Довженко', 'Email' with 'student1@test.com', and 'Пароль' (Password) with the placeholder 'Введіть пароль'. A dark 'Зберегти' (Save) button is positioned below these fields. To the right of the form is a large white rectangular area containing a cartoon illustration of a young man with dark hair, wearing a grey jacket and blue pants, holding a tablet.

Рисунок 32 - Сторінка профілю

РОЗДІЛ 7 ПОРІВНЯННЯ РЕАЛІЗОВАНОГО ФУНКЦІОНАЛУ З КОНКУРЕНТАМИ

По результатам проведеної роботи можна порівняти функціонал систем Google Classroom з Google Forms, Microsoft Teams з Microsoft Forms та Sigma School. Для зручності оформимо результати у вигляді таблиці 1.

Таблиця 1 - Порівняння функцій розробленої системи та існуючих

	Sigma School	Google Classroom та Google Forms	Microsoft Teams та Microsoft Forms
Сутності	предмет, курс	курс	курс
Призначення завдань	на окрему групу	на окрему групу, окремих студентів	на окрему групу, окремих студентів
Оцінювання відкритих відповідей	автоматично	тільки вручну	тільки вручну
Система доступів	власник, вчитель, студент.	вчитель, студент	власник, учасник, гість, ІТ адміністратор
Наявність голосового вводу	+	-	-
Види питань, що можуть бути в тесті	тільки відкриті з голосовим вводом	тести з одним/декількома варіантами відповідей,	тести з одним/декількома варіантами відповідей,

		відкриті питання, поля з вибором дати, завантаженням файлу + вибір часу, з випадуючого меню, сітка з кількома варіантами вибору, таблиця з декількома варіантами відповідей по кожному рядку	відкриті питання, поля з вибором дати, завантаженням файлу + ранжування, шкала вподобання
Можливість залишати пояснення до окремої відповіді	+	-	-

З порівняння можна зробити висновок, що система Sigma School містить суттєву відмінність у порівнянні з Google Classroom та Google Forms, Microsoft Teams та Microsoft Forms в тому, що вона передбачає автоматичне оцінювання відкритих відповідей та наявність голосового вводу, чого немає в конкурентів. Однак потенційними покращеннями були б такі нововведення, як можливість створювати питання різних типів, тести з однією або кількома варіантами відповідей, а також призначення завдань на окремих студентів.

ВИСНОВКИ

Розроблено систему автоматичного оцінювання відповідей Sigma School, яка дозволяє проводити тестування, а також вести облік робіт студентів та завдань.

В тому числі, описано проблему оцінювання відкритих питань та можливості використання засобів штучного інтелекту для її розв'язання, проведено аналіз існуючих методів для аналізу тексту, впроваджено систему, яка інтегрується з GPT для отримання автоматичного оцінювання й порівняно її з існуючими конкурентами згідно функціоналу, який вони вміщують.

На момент розробки система Sigma School готова до використання в реальному навчальному процесі і містить в собі функціонал для ведення обліку студентів, їх робіт й завдань.

Даний сервіс вміщує достатню кількість характеристик, які відрізняють його від існуючих конкурентів і привнесуть неоціненну користь в освітній процес. Автоматичне оцінювання пришвидшить процес перевірки й оптимізує навчальний час, що дозволить виділити його на більш критичні й важливі аспекти.

З'ясовано, що точність оцінювання напряму залежить від моделі, що використовується. Модель GPT-4, що була використана в реалізованій системі, суттєво відрізняється від свого попередника GPT-3 якістю роботи та кількістю доступних параметрів. З розвитком технологій ШІ вона може бути легко замінена іншою версією GPT, що покращить якість аналізу відповідей.

Зроблено висновки, що деякі можливості можуть бути додані в систему для більш зручного та універсального використання. Потенційні доповнення були описані в ході порівняння з конкуруючими системами і можуть бути розглянуті при подальшому розвитку системи. Одне з основних нововведень полягає у проведенні тестувань з математичних дисциплін, які включають формули і символи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

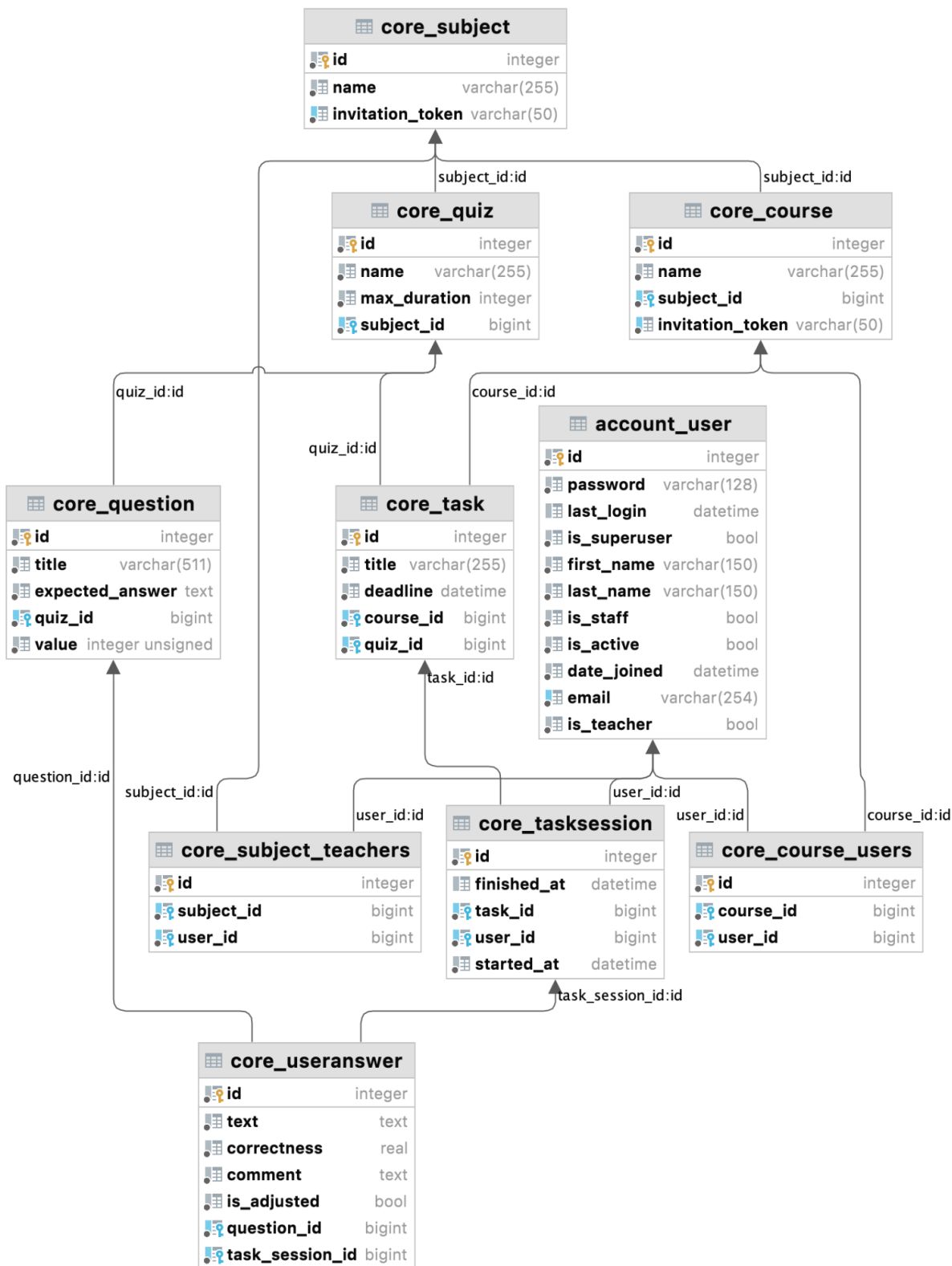
1. GPT-4 [Електронний ресурс] // OpenAI. – Режим доступу: <https://openai.com/product/gpt-4> (дата звернення: 06.05.2023). – Назва з екрана.
2. PyCharm: the Python IDE for Professional Developers by JetBrains [Електронний ресурс] // JetBrains. – Режим доступу: <https://www.jetbrains.com/pycharm/> (дата звернення: 06.05.2023). – Назва з екрана.
3. Welcome to Python.org [Електронний ресурс] // Python.org. – Режим доступу: <https://www.python.org/> (дата звернення: 06.05.2023). – Назва з екрана.
4. Microsoft. Visual Studio Code - Code Editing. Redefined [Електронний ресурс] / Microsoft // Visual Studio Code - Code Editing. Redefined. – Режим доступу: <https://code.visualstudio.com/> (дата звернення: 06.05.2023). – Назва з екрана.
5. JavaScript | MDN [Електронний ресурс] // MDN Web Docs. – Режим доступу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 06.05.2023). – Назва з екрана.
6. Getting started with Django [Електронний ресурс] // Django Project. – Режим доступу: <https://www.djangoproject.com/start/> (дата звернення: 06.05.2023). – Назва з екрана.
7. Home - Django REST framework [Електронний ресурс] // Home - Django REST framework. – Режим доступу: <https://www.django-rest-framework.org/> (дата звернення: 06.05.2023). – Назва з екрана.
8. Vue.js - the progressive javascript framework | vue.js [Електронний ресурс] // Vue.js - The Progressive JavaScript Framework | Vue.js. – Режим доступу: <https://vuejs.org/> (дата звернення: 06.05.2023). – Назва з екрана.
9. Element - The world's most popular Vue UI framework [Електронний ресурс] // Element - The world's most popular Vue UI framework. – Режим доступу:

- <https://element.eleme.io/#/en-US> (дата звернення: 06.05.2023). – Назва з екрана.
10. PostgreSQL [Електронний ресурс] // PostgreSQL. – Режим доступу: <https://www.postgresql.org/> (дата звернення: 06.05.2023). – Назва з екрана.
 11. Attention is all you need [Електронний ресурс] / А. Vaswani [та ін.] // Advances in neural information processing systems. – 2017. – С. 30. – Режим доступу: <https://arxiv.org/abs/1706.03762> (дата звернення: 06.05.2023). – Назва з екрана.
 12. Web Speech API - Web APIs | MDN [Електронний ресурс] // MDN Web Docs. – Режим доступу: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API (дата звернення: 06.05.2023). – Назва з екрана.
 13. Google for Education [Електронний ресурс] // Google for Education. – Режим доступу: <https://edu.google.com/workspace-for-education/classroom/> (дата звернення: 06.05.2023). – Назва з екрана.
 14. Google Forms: Online Form Creator | Google Workspace [Електронний ресурс] // Google. – Режим доступу: <https://www.google.com/forms/about/> (дата звернення: 06.05.2023). – Назва з екрана.
 15. Video Conferencing, Meetings, Calling | Microsoft Teams [Електронний ресурс] // Microsoft Teams. – Режим доступу: <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software> (дата звернення: 06.05.2023). – Назва з екрана.
 16. Microsoft Forms - Easily create surveys, quizzes, and polls. [Електронний ресурс] // Microsoft Forms - Easily create surveys, quizzes, and polls. – Режим доступу: <https://forms.office.com/> (дата звернення: 06.05.2023). – Назва з екрана.
 17. About [Електронний ресурс] // OpenAI. – Режим доступу: <https://openai.com/about> (дата звернення: 07.05.2023). – Назва з екрана.

18. OpenAI API [Електронний ресурс] // OpenAI API. – Режим доступу: <https://platform.openai.com/docs/libraries> (дата звернення: 07.05.2023). – Назва з екрана.
19. String Similarity Metrics – Edit Distance | Baeldung on Computer Science [Електронний ресурс] // Baeldung on Computer Science. – Режим доступу: <https://www.baeldung.com/cs/string-similarity-edit-distance#levenshtein-distance> (дата звернення: 08.05.2023). – Назва з екрана.

ДОДАТОК А

Діаграма класів



ДОДАТОК Б

Діаграма прецедентів для ролі вчителя (власника курсу)

