

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА ОСВІТНЬОЇ ІНФОРМАЦІЙНОЇ
СИСТЕМИ З ВИКОРИСТАННЯМ ШТУЧНОГО
ІНТЕЛЕКТУ»

Виконав: студент 4 курсу, групи 6.1220з
спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

освітньої програми Комп'ютерні науки

(назва освітньої програми)

Ю.В. Пасічник

(ініціали та прізвище)

Керівник доцент кафедри комп'ютерних наук,
доцент, д.е.н. Полуктова Н.Р.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент професор кафедри програмної інженерії, доцент,
к.ф.-м.н., Кудін О.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

Освітня програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук,
д.т.н., доцент

_____ Шило Г.М.

(підпис)

“ 22 ” грудня _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Пасічнику Юрію Вікторовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка освітньої інформаційної системи з використанням штучного інтелекту

керівник роботи Полуектова Наталія Робертівна, д.е.н., доцент

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

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2181-с

2. Строк подання студентом роботи 15.05.2024 р

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Розробка освітньої інформаційної системи

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

презентація

6. Консультанти розділів роботи

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

7. Дата видачі завдання 22.12.2023

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	08.01.2024	
2.	Збір вихідних даних.	17.01.2024	
3.	Обробка методичних та теоретичних джерел.	22.01.2024	
4.	Розробка першого та другого розділу.	04.02.2024	
5.	Розробка третього розділу.	15.03.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	15.05.2024	
7.	Захист кваліфікаційної роботи.	30.05.2024	

Студент _____
(підпис)

Ю.В. Пасічник _____
(ініціали та прізвище)

Керівник роботи _____
(підпис)

Н.Р. Полуктова _____
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О.Г. Спиця _____
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка освітньої інформаційної системи з використанням штучного інтелекту»: 43 с., 20 рис., 7 табл., 5 джерел, 4 додатки.

API, CHATGPT, JQUERY, MVC, MYSQL, PHP, TELEGRAM, ІНФОРМОВАНІСТЬ, ОСВІТА, ОСВІТНЯ СИСТЕМА, ШТУЧНИЙ ІНТЕЛЕКТ, ЯКІСТЬ.

Об'єкт дослідження – освітня інформаційна система.

Мета роботи: покращення та автоматизація освіти через інтеграцію можливостей штучного інтелекту в освітній процес.

Методи дослідження – аналіз, синтез, аналогія, прототипування.

В результаті був створений додаток, який поєднує класичну освіту з можливостями онлайн-платформ, пропонуючи можливість навчатися як в аудиторії, так і віддалено через застосунок, особливістю якого є використання штучного інтелекту для створення завдань і тестів, що адаптуються до потреб і рівня знань кожного класу, забезпечуючи більш ефективний і персоналізований процес навчання.

SUMMARY

Bachelor's Qualifying Theses «Development of an educational information system using artificial intelligence»: 43 pages, 20 figures, 7 tables, 5 references, 4 supplements.

API, CHATGPT, INFORMEDNESS, JQUERY, MVC, MYSQL, PHP, QUALITY, TELEGRAM, EDUCATION, EDUCATIONAL SYSTEM, ARTIFICIAL INTELLIGENCE.

Object of the study – educational information system.

Aim of the study: enhancement and automation of education through the integration of artificial intelligence capabilities into the education process.

Methods of research – analysis, synthesis, analogy, prototyping.

As a result, an application has been created that combines traditional education with the capabilities of online platforms, offering the opportunity to learn both in the classroom and remotely through the application. Its feature lies in the use of artificial intelligence to create tasks and tests that adapt to the needs and level of knowledge of each class, ensuring a more efficient and personalized learning process.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	7
1 Новітні технології в освітній системі.....	8
2 Розробка та архітектура веб-сайту	9
2.1 Архітектура.....	9
2.1.1 Вибір та налаштування веб-серверу	9
2.1.2 Моделювання бази даних.....	10
2.1.3 Архітектура сайту	15
2.1.4 Логічна структура сайту.....	16
2.2 Інтеграція API месенджера	18
2.2.1 Реєстрація та інтеграція боту.....	18
2.2.2 Реєстрація та інтеграція боту.....	18
2.2.3 Налаштування серверу для обміну повідомленнями	19
2.2.4 Створення логіки API для оповіщень	21
2.3 Штучний інтелект	23
2.3.1 Інтеграція в програмну середу	23
2.3.2 Створення та обробка запитів	24
3 програмна реалізація.....	27
Висновки	34
Перелік посилань.....	35
Додаток А Створення бази даних та структури таблиць	36
Додаток Б API запитів до штучного інтелекту OpenAI	38
Додаток В API запитів до Telegram Bot.....	40
Додаток Г Набір функцій інтерфейсу користувачів.....	41

ВСТУП

Протягом віків освіта пройшла довгий шлях від простих способів передачі знань до інтенсивного використання передових технологій. Спочатку навчання здійснювалося усно, а для фіксації матеріалів та виконання завдань використовувались паперові носії інформації, такі як щоденники, записні книжки або зошити.

Проте з появою цифрових технологій та активного використання мережі інтернет освітні програми зазнали радикальних змін, відкриваючи нові можливості для навчання та розвитку. У наш час неможливо уявити навчання без онлайн-платформ з безліччю курсів та тестів. Вони стали доступні кожному, пропонуючи широкий вибір курсів, навчального матеріалу, інструментів самостійного навчання та контролю.

Хоча зростання онлайн-освіти відкрило нові перспективи для масштабування освітніх програм, не слід забувати, що онлайн-платформи не можуть прийти на зміну традиційним освітнім установам. Саме поєднання традиційних систем освіти з онлайн-платформами створюють оптимальне рішення для сучасного навчання.

Отримані результати вивчення напрямку представляють собою нові методи та підходи до розвитку освітніх платформ, поєднуючи традиційні способи освіти з цифровими можливостями.

Дана кваліфікаційна робота спрямована на вивчення та розробку інтеграції штучного інтелекту в освітній процесі з метою оптимізації навчання та підвищення якості освіти.

Розроблений додаток дозволить виконувати персоналізацію навчальних програм з застосуванням актуальних знань в різних галузях освіти.

Надалі передбачається дослідження можливостей застосування штучного інтелекту в підготовці навчальних матеріалів.

1 НОВІТНІ ТЕХНОЛОГІЇ В ОСВІТНІЙ СИСТЕМІ

Останні роки набувають величезної популярності онлайн-платформи з навчання, надаючи доступ до освітніх ресурсів кожному, хто має доступ до мережі інтернет. До найбільш відомих платформ належать edX, Coursera, Khan Academy, Udeemy. Кожна з них пропонує широкий вибір лекцій, курсів, матеріалів, які задовольняють потреби майже всіх спеціалізацій та напрямків навчання.

Незважаючи на їхню інноваційність, все вони мають свої недоліки у порівнянні з традиційними формами навчання. Розглянемо основні мінуси онлайн-навчання та запропонуємо шляхи покращення освітнього процесу через інтеграцію традиційних методів і сучасних технологій, включаючи залучення штучного інтелекту.

Основною перевагою онлайн-платформ, яка робить їх привабливими для користувача є доступність і гнучкість. Онлайн-курси доступні з будь-якої точки світу в будь-який час. Проте, вони також мають безліч недоліків. Це -відсутність взаємодії, вимоги до мотивації та самодисципліни, обмежені можливості практичного навчання.

Метою роботи є інтеграція цифрових можливостей в освітній процес для створення комбінованих курсів, поєднуючих традиційну освіту з онлайн-завданнями та усуваючих головний недолік онлайн-курсів у вигляді заделегіть заготовлених завдань і тестів, що не можуть гарантувати якість оцінювання знань. В якості експерименту по розширенню якості процесу навчання поєднаємо штучний інтелект. Саме він допоможе створювати персоналізовані навчальні плани на основі актуальних матеріалів. Створювання унікальних завдань та тестів для кожного класу знизить ймовірність шахрайства і забезпечити об'єктивність оцінювання знань.

2 РОЗРОБКА ТА АРХІТЕКТУРА ВЕБ-САЙТУ

2.1 Архітектура

При створенні веб-застосунку важливо врахувати його архітектуру. Саме вона визначає структуру і взаємозв'язки між модулями системи. Необхідно ретельно спроектувати модель бази даних, розробити код програмного забезпечення. Дотримання ефективності при побудові забезпечує швидку та надійну роботу, спрощує процес розширення функціональності в майбутньому.

2.1.1 Вибір та налаштування веб-серверу

Для розробки був обраний PHP - популярна серверна мова програмування, що ідеально підходить для створення динамічного контенту та додатків. Розповсюдженість і добра документованість, безліч готових рішень, доступ до безлічі бібліотек та перехід на фреймворки за технічними потребами забезпечують її популярність.

В якості системи управління базами даних (СУБД) в поєднанні з PHP зазвичай застосовується MySQL. Вона базується на мові SQL і забезпечує швидкий доступ до даних. Її надійне та ефективне зберігання даних відмінно підходить для роботи з динамічними даними веб-ресурсів, надаючи інструменти управління базами даних, включаючи індексацію та транзакції.

Ключовою перевагою вибору PHP і MySQL є їх сумісність з різними API. Інтеграція з API мобільного застосунку обміну повідомлень Телеграм та API штучного інтелекту від OpenAI може бути виконана без необхідності встановлення додаткових рішень, наприклад, через Composer. Це значно спрощує процес розробки та прискорює впровадження нових функцій.

Відсутність спеціальних вимог до програмних пакетів забезпечує встановлення навіть на звичайний хостінг, отже робить ідеальним для

соціальних проектів з обмеженим фінансуванням. Немає потреби в купівлі дорогого виділеного серверу та адміністрування, оскільки більшість провайдерів хостингу пропонують підтримку PHP та MySQL за замовчуванням.

Це рішення також забезпечує високу гнучкість та масштабованість. З зростанням трафіку на сайті або збільшенням обсягу, можливість масштабувати.

У підсумку, вибір технологій для розробки забезпечує надійність, гнучкість, безпеку та економічність, роблячи цей стек технологій відмінним вибором для реалізації сучасного проекту.

2.1.2 Моделювання бази даних

Моделювання бази даних є ключовим етапом у розробці інформаційної системи, допомагає визначити оптимальну структуру для зберігання даних та встановлення відносин між сутностями. Однією з головних задач роботи є планування таблиць, полей та відносин, що дозволять ефективно працювати з інформацією та забезпечуватимуть зручний доступ до неї.

База даних навчального закладу являє собою складну систему, що складається з множини взаємопов'язаних таблиць, кожна з яких відповідає за зберігання та управління певним типом даних. Взаємозв'язки таблиць за допомогою зовнішніх ключей відіграють ключову роль у забезпеченні цілісності, узгодженості та ефективності роботи бази даних. Коректні зовнішні ключі допомагають уникнути дублювання даних, спрощують виконання запитів, підтримують складні правила і забезпечують правильні зв'язки між різними типами даних.

Вивчивши потреби та вимоги до планування нашої бази даних для освітньої системи, щоб забезпечити ефективне зберігання та доступ до інформації студентів, викладачів, освітніх курсів та інші аспекти навчального процесу визначаємо таблиці даних (рис. 2.1):

- користувачі;
- класи;

- класи_учні;
- предмети;
- заняття;
- розклад;
- оцінки;

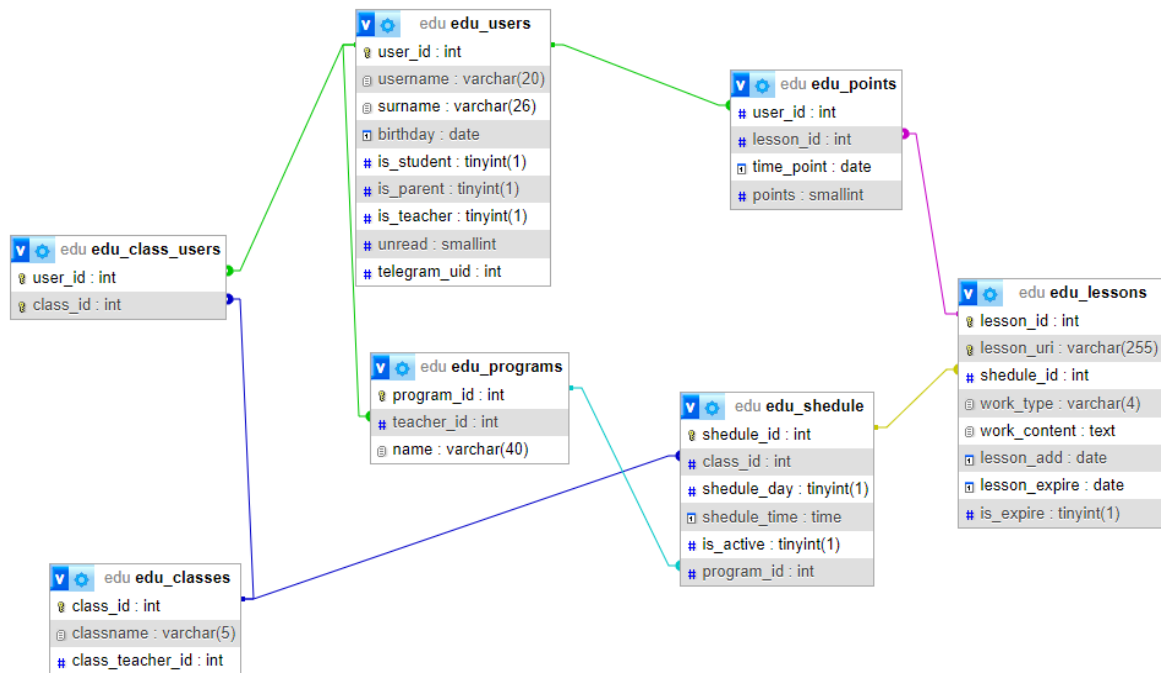


Рисунок 2.1 – Графічне відображення структури бази даних

Опишемо окремо структуру, відповідність полей і навіщо були створені взаємозв'язки таблиць.

Таблиця користувачів «edu_users» зберігає інформацію про користувачів системи, включаючи студентів, батьків та вчителів (табл. 2.1).

Це основна таблиця в базі даних, яка зберігає інформацію про користувачів системи, включаючи студентів, батьків та вчителів. Кожен запис у таблиці відповідає одному користувачеві.

Таблиця класів «edu_classes» зберігає інформацію про класи (табл. 2.2). Кожен запис відповідає одному класу, включаючи його назву та ідентифікатор вчителя, відповідального за клас.

Таблиця 2.1 – Структура таблиці користувачів

Назва поля	Опис
user_id	Унікальний ідентифікатор користувача.
username	Ім'я користувача.
surname	Прізвище користувача.
birthday	Дата народження.
is_student	Вказує, чи є користувач студентом.
is_parent	Вказує, чи є користувач батьком студента.
is_teacher	Вказує, чи є користувач вчителем.
unread	Кількість непрочитаних сповіщень користувача.
telegram_uid	Ідентифікатор користувача зовнішнього застосунку обміну повідомлень. Використовується для авторизації та інтеграції з функціями застосунку.

Таблиця 2.2 – Структура таблиці класів

Назва поля	Опис
class_id	Унікальний ідентифікатор класу, який автоматично збільшується при додаванні нового запису. Використовується для однозначної ідентифікації кожного класу в системі.
classname	Назва класу.
class_teacher_id	Ідентифікатор вчителя, відповідального за клас.

Таблиця зв'язку класів з учнями «edu_class_users» пов'язує користувачів із класами (табл. 2.3). Взаємопов'язаність таблиці з «edu_users» надає ефективне виконання запитів, наприклад, частий запит на знаходження всіх студентів певного класу.

Таблиця розкладу класів «edu_shedule» зберігає інформацію про розклади класів (табл. 2.4). Інформація про розклад класів зберігається в даній таблиці і

допомагає уникати дублювання даних у таблиці «edu_classes». Це значно зменшує надмірність і підвищує ефективність зберігання даних.

Таблиця 2.3 – Структура таблиці зв'язку класів з учнями

Назва поля	Опис
user_id	Посилання на user_id таблиці edu_users.
class_id	Посилання на class_id таблиці edu_classes.

Таблиця 2.4 – Структура таблиці розкладу класів

Назва поля	Опис
shedule_id	Унікальний ідентифікатор розкладу.
class_id	Посилання на class_id таблиці edu_classes.
program_id	Посилання на program_id таблиці edu_programs.
shedule_day	Числове значення, еквівалентне дню тижня.
shedule_time	Час у форматі ЧЧ:ММ:СС.
is_active	Маркер активності

Таблиця «edu_programs» зберігає інформацію про навчальні програми, пов'язані з вчителями (табл. 2.5). Містить посилання на таблицю «edu_users», пов'язуючи вчителів із навчальними програмами, які веде кожен учитель, та відстежувати відповідальність за проведення програм.

Таблиця 2.5 – Структура таблиці навчальних програм

Назва поля	Опис
program_id	Посилання на user_id таблиці edu_users.
teacher_id	Посилання на user_id таблиці edu_users.
name	Назва навчальної програми.

Ключі забезпечують коректність посилань між таблицями, отримуючи узгодженість даних. Наприклад, видалення запису в таблиці класів «edu_classes» не призведе до порушення узгодженості даних. Всі пов'язані записи в таблиці «edu_schedule» також будуть автоматично оновлені або видалені.

Таблиця «edu_lessons» зберігає інформацію про уроки (табл. 2.6). Пов'язує уроки з розкладом за допомогою schedule_id, допомагаючи легко знаходити уроки, що належать до конкретного розкладу і керувати ними. Також дана таблиця зберігає інформацію в поле work_content, отриману від запиту до штучного інтелекту, залучаючи її в створенні завдань і тестів в навчальному процесі.

Таблиця 2.6 – Структура таблиці уроків

Назва поля	Опис
lesson_id	Унікальний ідентифікатор уроку.
lesson_uri	Унікальна адреса уроку у вигляді md5 хешу.
shedule_id	Посилання на shedule_id таблиці edu_schedule.
work_type	Тип завдання.
work_content	Дані відповіді штучного інтелекту.
lesson_add	Дата створення уроку.
lesson_expire	Дата закінчення терміну уроку.
is_expire	Значення закінчення терміну уроку.

Таблиця «edu_points» зберігає інформацію про бали, отримані користувачами за уроки (табл. 2.7). Таблиця пов'язує користувачів з уроками та балами, які вони отримали. Гарантує, що кожен користувач, зазначений у таблиці оцінок, дійсно існує у таблиці користувачів.

Таблиця 2.7 – Структура таблиці оцінок

Назва поля	Опис
user_id	Посилання на user_id таблиці edu_users.
lesson_id	Посилання на lesson_id таблиці edu_lessons.
time_point	Дата отримання балів.
points	Кількість отриманих балів.

Впроваджено нормалізацію бази даних, індексацію та зовнішні ключі, що забезпечує ефективну організацію і узгодженість даних. Завдяки оптимізації продуктивності бази даних максимально знижено час виконання запитів.

2.1.3 Архітектура сайту

При плануванні системи використовуємо гібридну модель, яка поєднує елементи MVC (Model-View-Controller) у підході до організації архітектури. Хоча це не є чистою реалізацією MVC, вона включає деякі принципи цієї архітектурної моделі. Розподіл на роботу з даними (модель), обробку запитів (контролер) та відображення результатів (шаблон) дозволяє використовувати принципи MVC при розробці та підтримці освітньої системи.

Компоненти, що відповідають за роботу з базою даних та бізнес логіку зосереджені в папках engine/classes та engine/modules;

Контролери або модулі, що відповідають за обробку запитів користувачів, взаємодію з моделями для отримання необхідних даних та вибір шаблону відображення:

- engine/modules;

Шаблони, що відповідають за відображення інформації користувачам та містять HTML-код, розташовані у папці:

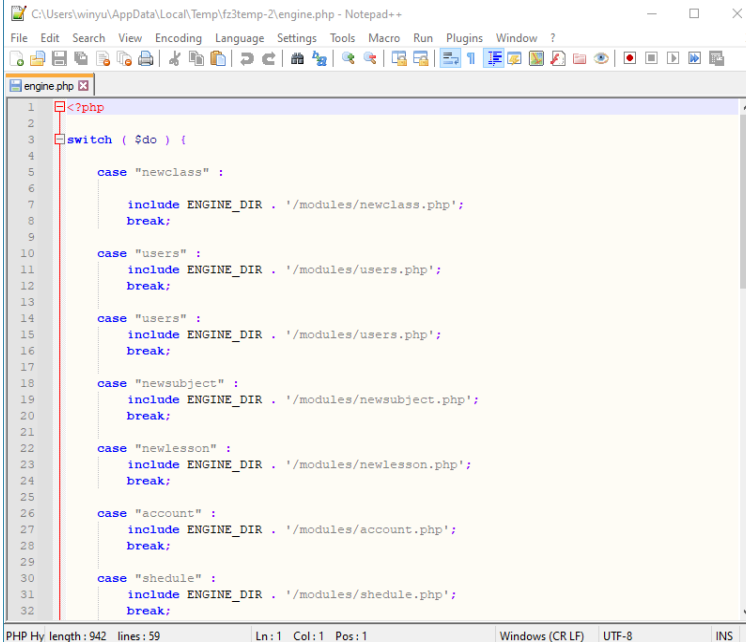
- templates/{theme};

Окремо виділені конфігураційні файли у вигляді масивів даних для з'єднання з базою даних, параметри системи, опції взаємодії з сторонніми API:

- engine/config;

2.1.4 Логічна структура сайту

В основі архітектури сайту лежить чітке розділення на функціональні модулі. Центральним елементом, що відповідає за маршрутизацію та перенаправлення запитів на модулі виступає engine.php (рис. 2.2). Кожен запит перевіряється на правило, перенаправляючи на відповідний модуль каталогу, якщо він описаний правилами.



```

1 <?php
2
3 switch ( $do ) {
4
5     case "newclass" :
6
7         include ENGINE_DIR . '/modules/newclass.php';
8         break;
9
10    case "users" :
11        include ENGINE_DIR . '/modules/users.php';
12        break;
13
14    case "users" :
15        include ENGINE_DIR . '/modules/users.php';
16        break;
17
18    case "newsobject" :
19        include ENGINE_DIR . '/modules/newsobject.php';
20        break;
21
22    case "newlesson" :
23        include ENGINE_DIR . '/modules/newlesson.php';
24        break;
25
26    case "account" :
27        include ENGINE_DIR . '/modules/account.php';
28        break;
29
30    case "shchedule" :
31        include ENGINE_DIR . '/modules/shchedule.php';
32        break;

```

Рисунок 2.2 – Маршрутизація модулів

Всі основні модулі системи знаходяться в каталозі modules та мають окремий функціональний зміст:

- модуль users відповідає за відображення та редагування користувачів системи;
- модуль newsobject забезпечує реєстрацію нових навчальних предметів та закріплення до вчителів;

- модуль newlesson додає новий урок з урахуванням створеного розкладу занять;
- модуль account забезпечує редагування персональних даних та настройку сповіщень від системи на мобільні пристрої;
- модуль schedule організовує, сортує додані і закріплені за класами предмети згідно календарного тижня;
- модуль homework генерує завдання з використанням штучного інтелекту, а також перевіряє на проходження завдань;
- модуль login підтримує авторизацію користувача в системі;
- модуль config зберігає та ініціалізує настройки для роботи системи;

Чітке розподілення модулів дозволяє не лише забезпечити масштабованість та гнучкість системи, але й спрощує розробку та підтримку інформаційної системи. Структура модульності надає можливість оновлювати чи вносити зміни без порушень загальної логіки роботи системи.

Використання шаблонів дозволяє застосовувати та підтримувати єдиний зовнішній вигляд сторінок, покращуючи користувацький досвід. Завдяки розділенню шаблонів та логіки (рис. 2.3) немає потреби переписувати програмний код.

```
function __construct(){
    $this->dir = ROOT_DIR . '/templates/';
}

function set($name, $var) {
    if( is_array( $var ) && count( $var ) ) {
        foreach ( $var as $key => $key_var ) {
            $this->set( $key, $key_var );
        }
    } else
        $this->data[$name] = str_ireplace( "{include", "{#123;include", $var );
}

function set_block($name, $var) {
    if( is_array( $var ) && count( $var ) ) {
        foreach ( $var as $key => $key_var ) {
            $this->set_block( $key, $key_var );
        }
    } else
        $this->block_data[$name] = str_ireplace( "{include", "{#123;include", $var );
}

function load_template($tpl_name) {
```

Рисунок 2.3 – Шаблонування модулів

2.2 Інтеграція API месенджера

2.2.1 Реєстрація та інтеграція боту

В епоху технологічного прогресу і зростанням можливостей інтелектуальних систем, необхідність ефективного оповіщення стає критично важливою для забезпечення оперативного реагування на події. Поєднання з месенджерами має значний потенціал для освітньої системи, оскільки саме освітні заклади стикаються з необхідністю ефективної комунікації зі студентами та батьками.

Можна відокремити основні позитивні зміни в освітніх процесах:

- швидка комунікація;
- швидка комунікація;
- моніторинг успішності академічного процесу;
- оповіщення про події;

2.2.2 Реєстрація та інтеграція боту

Реєстрація бота – перший крок у створенні бота (рис. 2.4) , який буде взаємодіяти з користувачами через цю платформу.

Процес реєстрації займає декілька кроків та потребує зареєстрований і активований акаунт в застосунку обміну повідомлень:

- зв'язатися з BotFather, офіційним ботом для створення і керування іншими ботами;
- розпочати чат з BotFather, натиснувши Start, щоб почати чат;
- створити нового бота надіславши команду /newbot, яка запустить процес створення нового бота;
- вказати унікальну назву для відображення у списку контактів та сповіщень;
- вказати ім'я бота;

- отримати унікальний ідентифікатор для з'єднання сервісів з платформою Telegram;

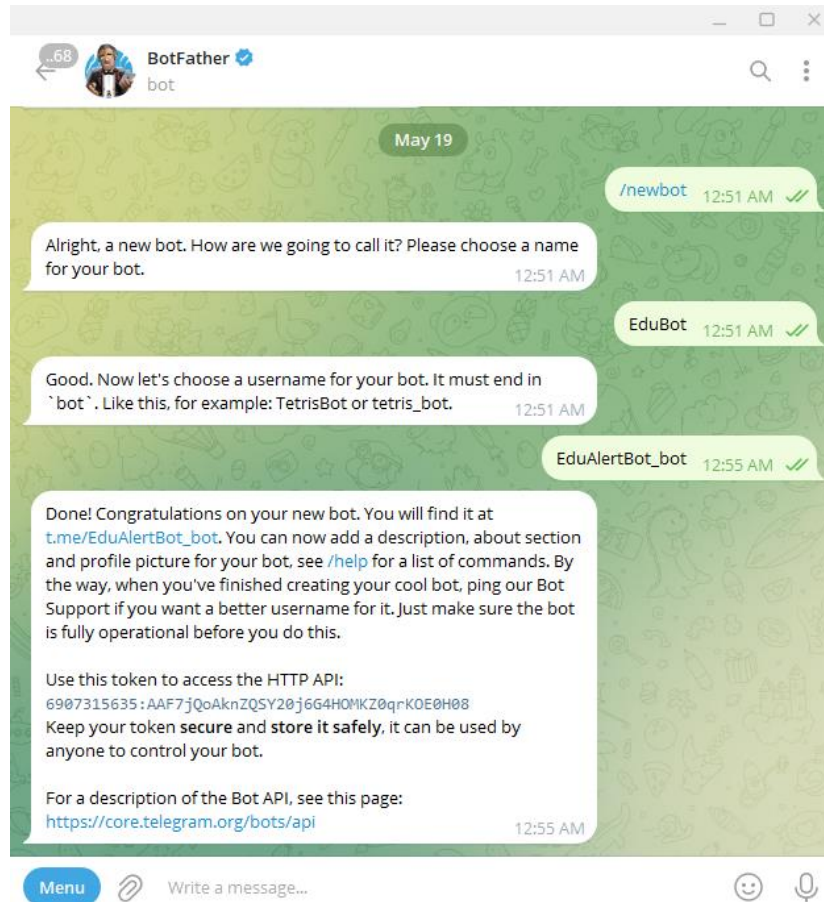


Рисунок 2.4 – Процес реєстрації бота

2.2.3 Налаштування серверу для обміну повідомленнями

Для початку прийому та відправки повідомлень у Telegram за допомогою вебхуку, необхідно розробити програмний код для реєстрації вебхуку.

Вебхук – механізм, який дозволяє серверу Telegram надсилати оновлення стороннім серверам в режимі реального часу. Замість постійного опитування серверу на предмет нових подій, вебхук дозволяє зареєструвати унікальне посилання в мережі інтернет, на яке безпосередньо будуть надсилатися ці оновлення.

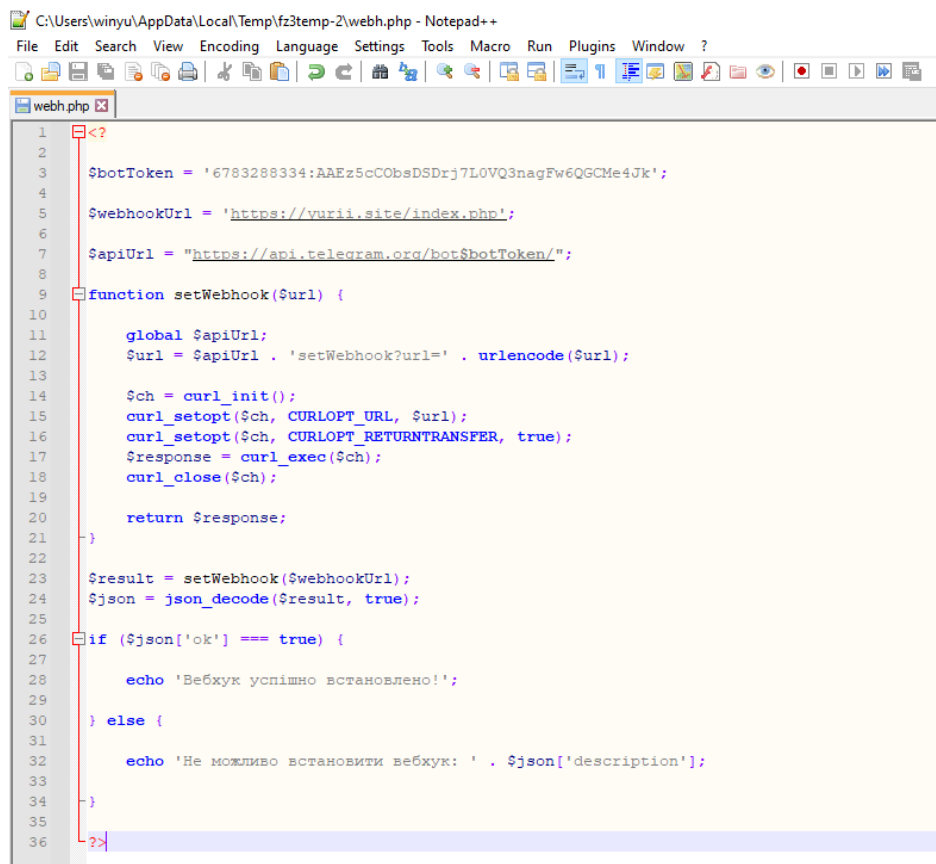
Коли користувач взаємодіє з нашим ботом або відправляє повідомлення, сервер обміну повідомлень надсилає запит на зареєстровану адресу нашого вебхуку, повідомляючи про виниклу подію. Завдяки цьому користувацький бот

отримує оновлення миттєво і забезпечує швидку та ефективну взаємодію з користувачами. Це робить вебхуки корисним інструментом для створення шкідливої взаємодії з користувачами.

Створення файлу реєстрації вебхуку – перший крок у процесі налаштування взаємодії з сервером додатку обміну повідомлень (рис. 2.2). У цьому файлі створюється логіка, а також виконуються всі необхідні налаштування для безперервного отримання оновлень від Telegram.

Для налаштування необхідно зареєструвати унікальне посилання на наш сервер. Обов'язковою умовою є доступ виключно через захищене з'єднання HTTPS. Забезпечення безпеки передачі даних є умовою обміну повідомленнями. Після успішної реєстрації всі події надходять на це посилання у вигляді JSON-об'єктів, які можна обробляти в реальному часі.

Після успішної реєстрації і отримання унікального ідентифікатора створюємо файл реєстрації вебхука (рис. 2.5).



```

1  <?
2
3
4  $botToken = '6783288334:AAEz5cCObsDSDrj7L0VQ3nagFw6QGCM4Jk';
5  $webhookUrl = 'https://vurii.site/index.php';
6
7  $apiUrl = "https://api.telegram.org/bot$botToken/";
8
9  function setWebhook($url) {
10
11     global $apiUrl;
12     $url = $apiUrl . 'setWebhook?url=' . urlencode($url);
13
14     $ch = curl_init();
15     curl_setopt($ch, CURLOPT_URL, $url);
16     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
17     $response = curl_exec($ch);
18     curl_close($ch);
19
20     return $response;
21 }
22
23 $result = setWebhook($webhookUrl);
24 $json = json_decode($result, true);
25
26 if ($json['ok'] === true) {
27     echo 'Вебхук успішно встановлено!';
28 } else {
29
30     echo 'Не можливо встановити вебхук: ' . $json['description'];
31 }
32
33
34
35
36

```

Рисунок 2.5 – Реєстрація вебхука

Файл нараховує три змінні, які будуть мати необхідні дані для реєстрації вебхуку. Змінна `botToken` містить унікальний ідентифікатор, змінна `webhookUrl` містить адресу на нашому сервері для обробки запитів, змінна `apiUrl` має посилання на сервер Telegram.

2.2.4 Створення логіки API для оповіщень

Для відправки повідомлень до стороннього API створюємо функцію отримання і компонування запиту, а також відправки з подальшим одержанням відповіді (рис. 2.6):

```
define('BOT_TOKEN', 'YOUR_BOT_TOKEN');
function sendMessage($chat_id, $text) {
    $url = 'https://api.telegram.org/bot' . BOT_TOKEN . '/sendMessage';
    $data = array(
        'chat_id' => $chat_id,
        'text' => $text,
    );
    $ch = curl_init($url);
    curl_setopt($ch, CURLOPT_POST, 1);
    curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
    curl_setopt($ch, CURLOPT_HTTPHEADER, array('Content-Type:
application/json'));
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    $response = curl_exec($ch);
    curl_close($ch);
    return $response;
}
```

Рисунок 2.6 – Функція взаємодії з стороннім API

Для подальшої роботи достатньо ініціювати файл в нашому engine.php на моменті маршрутизації запитів і можливо додавати правила відправки в кожному модулі системи.

Наприклад, якщо в розкладі з'явився новий предмет, достатньо викликати функцію файлу взаємодії з API Telegram sendLessonNotification (рис. 2.7):

```
function sendNewLessonNotification($lesson_name) {
    $message = "Новий урок: $lesson_name";
    sendMessage('CHAT_ID', $message);
}
```

Рисунок 2.7 – Функція відправки повідомлення

При потребі залучити дані з бази даних, наприклад, інформування про нові оцінки користувача системи, достатньо зробити запит та передати дані до функції взаємодії (рис. 2.8):

```
$query = $db->query("SELECT `points`, `edu_points` FROM " . USERPREFIX .
"_users WHERE `user_id`='$id'");

if ($db->num_rows($query) > 0) {
    $row = $db->get_row($query);
    $points = $row['points'];
    sendNewGradeNotification($points);
}

function sendNewGradeNotification($grade) {
    $message = "Нова оцінка: $grade";
    sendMessage('CHAT_ID', $message);
}
```

Рисунок 2.8 – Функція відправки даних з даними користувача

2.3 Штучний інтелект

Штучний інтелект від OpenAI надає широкі можливості по інтеграції в сторонні системи для покращення та автоматизації процесів. Модель проходить всебічне навчання із різних джерел, включаючи книги, статті та інші письмові матеріали, що сприяє всебічному розвитку, розуміє та генерує дані на різні теми, враховуючи контекст. Даний підхід забезпечує високу ступінь універсальності при створенні відповідей на запити користувача.

Інтеграція з освітньою платформою для створення тестів та перевірки завдань ідеальна з багатьох причин. По-перше, це його можливість генерувати різноманітний та якісний контент, що відповідає навчальній програмі та різним рівням учнів. Миттєва перевірка завдань та обгрунтовані пояснення помилок прискорюють процес навчання, підвищуючи ефективність навчання.

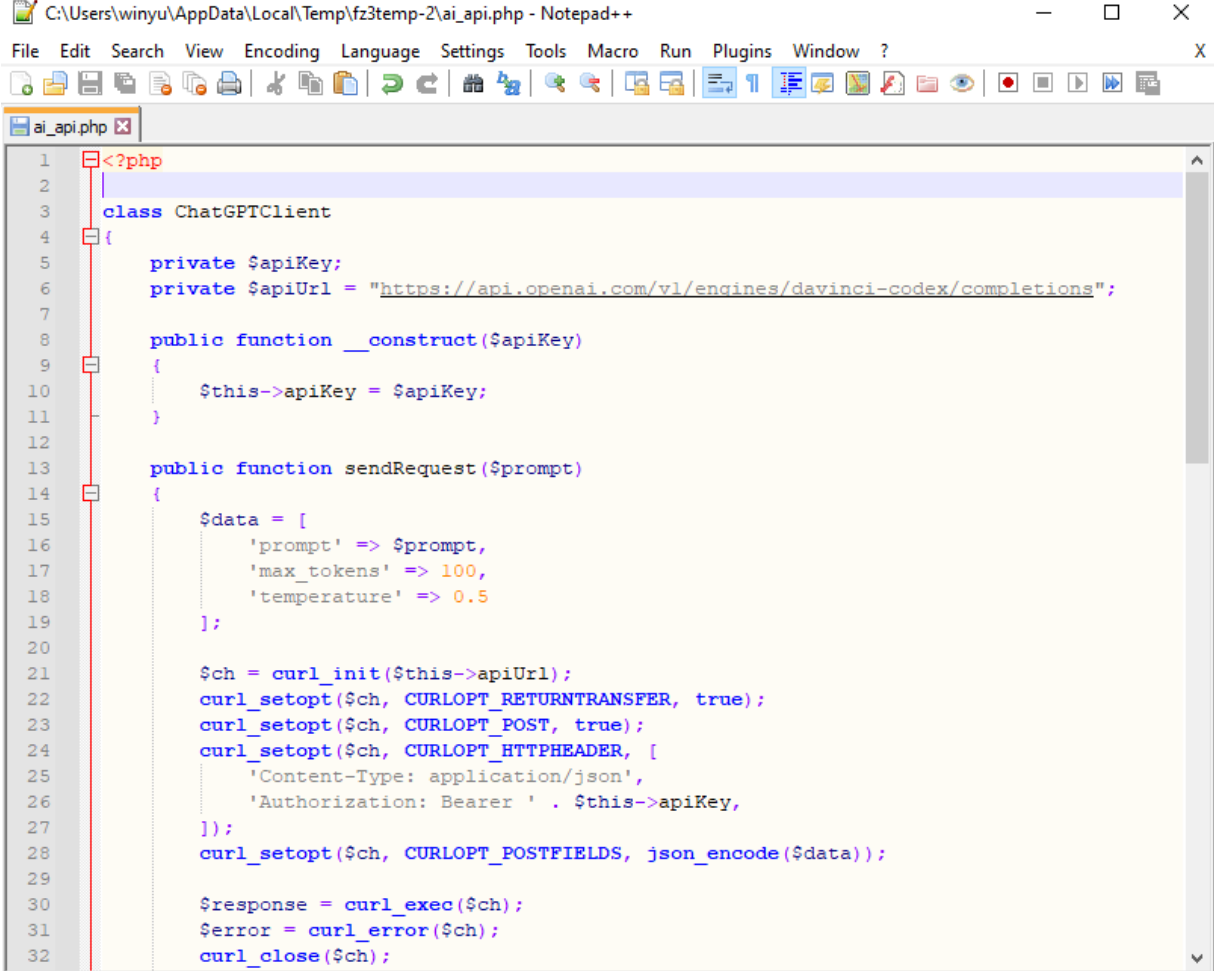
Значною перевагою використання готової моделі штучного інтелекту є відсутність необхідності вдосконалення самого штучного інтелекту без витрат ресурсів на його навчання та оновлення. Достатньо підтримувати актуальність засобу інтеграції. Скорочення витрат на розробку та обслуговування надає ресурси для зосередження на покращенні автоматизації та оптимізації процесів основної системи.

2.3.1 Інтеграція в програмну середу

Інтеграція коду для взаємодії з API ChatGPT значно розширює функціональність системи, забезпечуючи можливість використання штучного інтелекту для генерації матеріалів тестування учнів та аналізу в перевірці робіт. Спочатку переконаємося, що наш сервер має встановлене розширення cURL, яке необхідне для виконання HTTP-запитів.

Наступним кроком є реєстрація на сайті OpenAI та отримання API ключа, для автентифікації запитів до ChatGPT API.

Додаємо наш файл `ai_api.php` з класом обробки запитів до каталогу класів та формуємо логіку (рис. 2.9). Розміщуємо до каталогу `engine` з класами нашого проекту.



```

1  <?php
2
3  class ChatGPTClient
4  {
5      private $apiKey;
6      private $apiUrl = "https://api.openai.com/v1/engines/davinci-codex/completions";
7
8      public function __construct($apiKey)
9      {
10         $this->apiKey = $apiKey;
11     }
12
13     public function sendRequest($prompt)
14     {
15         $data = [
16             'prompt' => $prompt,
17             'max_tokens' => 100,
18             'temperature' => 0.5
19         ];
20
21         $ch = curl_init($this->apiUrl);
22         curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
23         curl_setopt($ch, CURLOPT_POST, true);
24         curl_setopt($ch, CURLOPT_HTTPHEADER, [
25             'Content-Type: application/json',
26             'Authorization: Bearer ' . $this->apiKey,
27         ]);
28         curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));
29
30         $response = curl_exec($ch);
31         $error = curl_error($ch);
32         curl_close($ch);

```

PHP Hy length: 1,366 lines: 54 Ln: 2 Col: 1 Pos: 8 Windows (CR LF) UTF-8 INS

Рисунок 2.9 – Клас обробки запитів до штучного інтелекту

2.3.2 Створення та обробка запитів

Запит до API починаємо з створення екземпляру класу, який відповідає за взаємодію з стороннім API штучного інтелекту. Створюємо новий об'єкт `ChatGPTClient` та в якості параметру конструктора передаємо ключ. Після створення об'єкта викликаємо метод `sendRequest`, який відправляє запит до API ChatGPT (рис. 2.10). У відповідь повертається масив даних у форматі JSON. Важливо те, що формат відповіді ми задаємо у запиті до штучного інтелекту з метою отримати дані у форматі, який зручний для подальшої обробки.

Згенерований за допомогою моделі штучного інтелекту текстовий вміст зберігаємо в базу даних. Після цього ми можемо продовжити обробку отриманої інформації або відобразити користувачеві відповідно до потреб.

```
$chatGPTClient = new ChatGPTClient($apiKey);
$response = $chatGPTClient->sendRequest($request);
$result = $db->safesql($response['choices'][0]['text']);
$db->query("UPDATE edu_lessons SET work_content = '$result' WHERE id = '$lesson_id'");
```

Рисунок 2.10 – Запит та відповідь від штучного інтелекту

Нижче представлена відповідь даних у форматі JSON для створення тесту (рис. 2.11). Містить ідентифікатор питання, текст питання, масив відповідей на питання, де кожна відповідь має свій ідентифікатор. Кожен елемент масиву має прапорець маркування, що позначає, чи є ця відповідь правильною.

Цей формат дозволяє чітко структурувати дані, щоб легко ідентифікувати контент. Він ідеально підходить для автоматизованого оброблення даних і реалізації функцій тестування та оцінювання.

Дані також зберігаємо у незмінному форматі для забезпечення уніфікації даних та зручності оновлення за потребою.

```
{
  "id": 1,
  "question": "Question",
  "answers": [
    {"qid": 1, "answer": "Answer", "is_correct": 1},
    {"qid": 2, "answer": "Answer", "is_correct": 0},
    {"qid": 3, "answer": "Answer", "is_correct": 0},
  ]
}
```

Рисунок 2.11 – Відповідь штучного інтелекту

Отримавши дані у необхідному форматі ми можемо систематизувати обробку відповідей за шаблоном. Додаємо функцію по обробці вхідних даних (рис. 2.12). оцінювання.

```
$data = json_decode($json_data, true);  
foreach ($data as $item) {  
    $question_id = $item['id'];  
    $question_text = $item['question'];  
    foreach ($item['answers'] as $answer) {  
        $answer_id = $answer['qid'];  
        $answer_text = $answer['answer'];  
        $is_correct = $answer['is_correct'] ? 1 : 0;  
        echo "Answer ID: $answer_id, Text: $answer_text, Correct: $is_correct\n";  
    }  
}
```

Рисунок 2.12 – Функція обробки JSON відповіді

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

Виконавши роботу отримуємо робочу версію нашої системи. Для підтримки актуальності контактів і можливість інформаційних оповіщень нашої освітньої системи була введена реєстрація та авторизація виключно за допомогою найпопулярнішого на момент розробки застосунку обміну повідомлень Telegram (рис. 3.1).

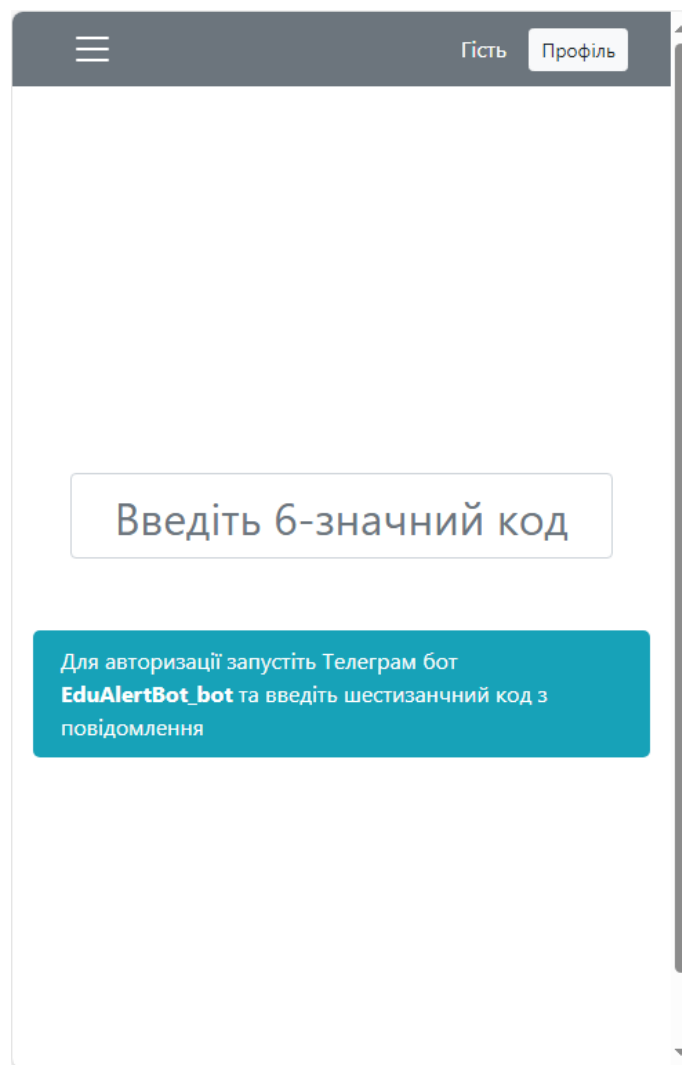


Рисунок 3.1 – Процес реєстрації за допомогою месенджера

Рішення з інтеграції надає одразу декілька переваг. Функції перевірки користувача за номером телефону і актуалізація даних переходять в обов'язки месенджера. Зареєструватися може лише верифікований за номером телефона

користувач. В результаті ми отримуємо не лише підтвердженого користувача, а й функції сповіщень.

Користувачу пропонується авторизуватися за допомогою боту месенджера. Відкривши бота та ініціював діалог користувач отримує унікальний 6-значний код, згенерований нашою системою, який дійсний короткий термін. При успішній авторизації відбувається вхід у систему з персоналізацією даних, знайдених в базі даних за цим ідентифікатором месенджера (рис. 3.2). Якщо вхід вперше, відбувається реєстрація в нашій системі і автоматична авторизація.

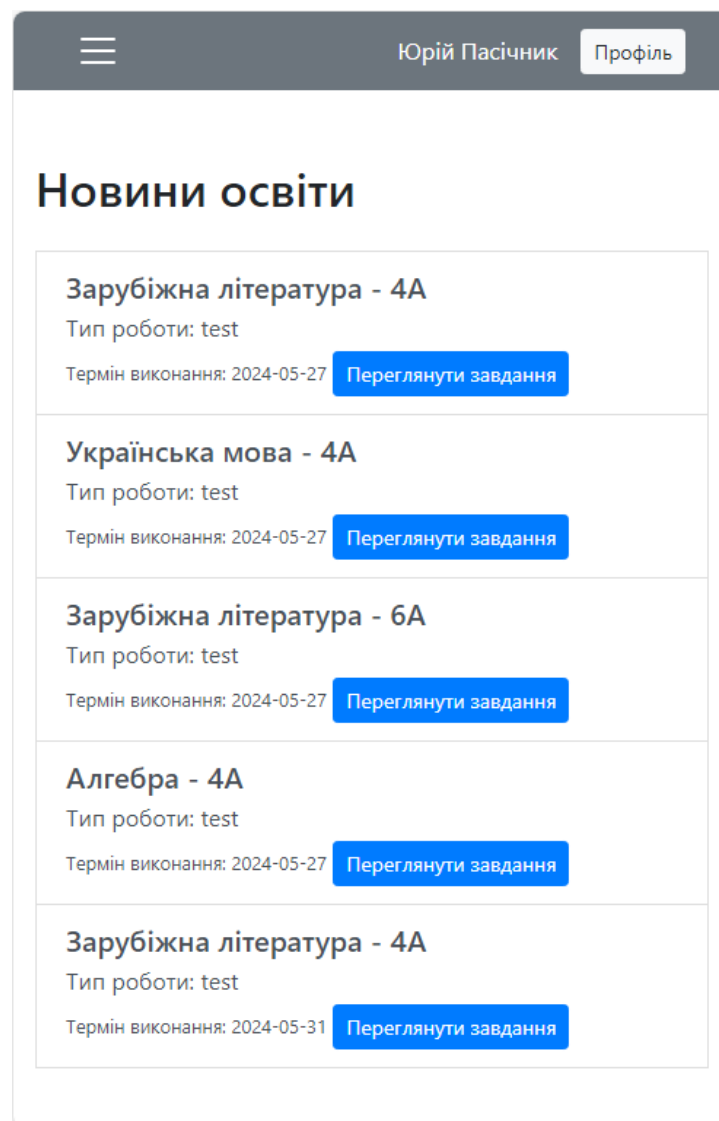
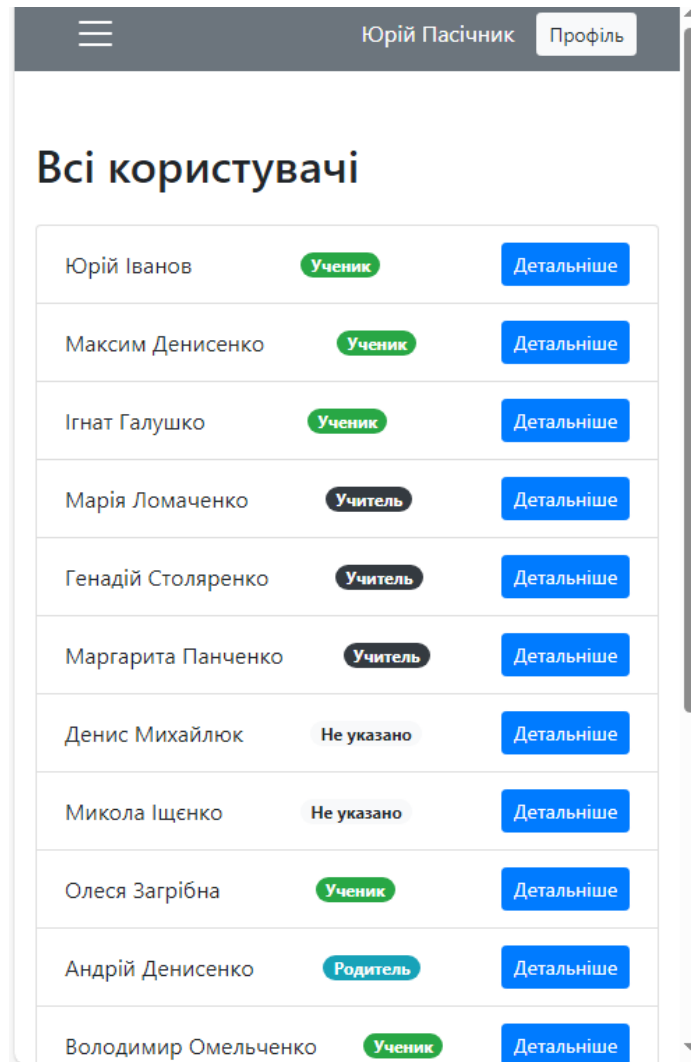


Рисунок 3.2 – Персоналізація після авторизації

Після успішної реєстрації всі користувачі опиняються в базі даних з інформацією про унікальний ідентифікатор в застосунку Telegram, а також

базовою інформацією, такою як ім'я та прізвище. Ці дані користувач або вчитель може виправити в своєму кабінеті.

Кожен вчитель має доступ до списку всіх користувачів з можливістю підтвердити належність користувача до учня, або батьків учня (рис. 3.3).



Юрій Пасічник Профіль		
Всі користувачі		
Юрій Іванов	Ученик	Детальніше
Максим Денисенко	Ученик	Детальніше
Ігнат Галушко	Ученик	Детальніше
Марія Ломаченко	Учитель	Детальніше
Генадій Столяренко	Учитель	Детальніше
Маргарита Панченко	Учитель	Детальніше
Денис Михайлюк	Не указано	Детальніше
Микола Іщенко	Не указано	Детальніше
Олеся Загібна	Ученик	Детальніше
Андрій Денисенко	Родитель	Детальніше
Володимир Омельченко	Ученик	Детальніше

Рисунок 3.3 – Список користувачів

Успішно підтвержені акаунти мають 3 типи користувача:

- вчитель;
- учень;
- батько;

Вчителі мають окремі функції, більшість з яких це контроль освітнього процесу, доступ до формування списку класів та предметів, створення та

перевірка завдань за допомогою штучного інтелекту, формування розкладу для класів (рис.3.4).

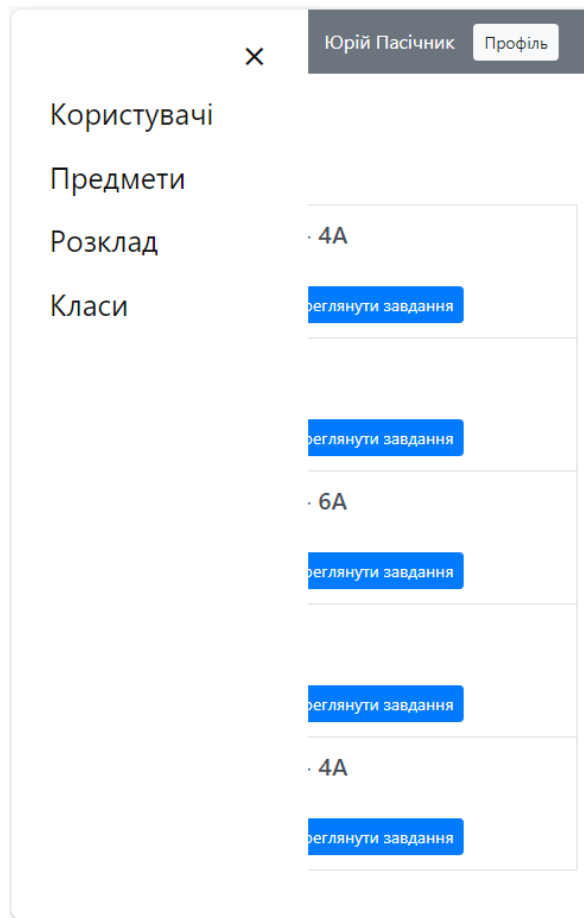


Рисунок 3.4 – Меню швидкого доступу вчителів

Учні мають доступ до своїх предметів, розкладу занять, та можливість проходження домашніх завдань (рис. 3.5). Для швидкого доступу кожна група має своє окреме меню.

Остання група батьків не має персональних функцій. Група лише отримує інформаційні повідомлення про нові завдання, проходження домашніх завдань та отримані оцінки учня, за яким закріплений акаунт батька чи батьків.

Користувачі з статусом вчителя мають можливість формувати класи з користувачів, верифікованих як учні. Для цього кожен вчитель може зайти в окрему категорію класів і додати назву класу з переліку, який формується по шаблону 1-12 клас та позначку від А до Г. Опинившись в даному каталозі можна

побачити учнів для додавання. Відображаються лише учні, що не додані до жодного класу.

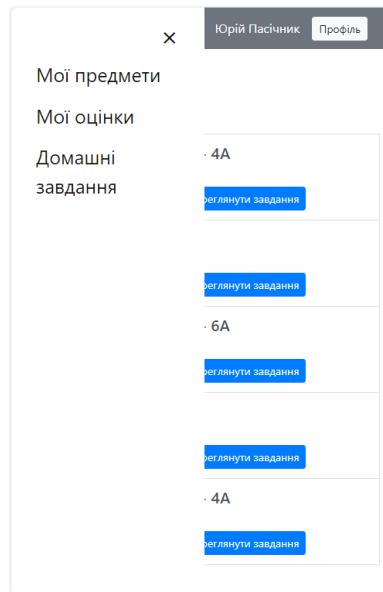


Рисунок 3.5 – Меню швидкого доступу учнів

Щоб обрати учня достатньо лише натиснути на тумблер поряд з ім'ям та прізвищем (рис. 3.6). Після формування списку та натискання кнопки підтвердження клас буде додано, а користувача направить в меню додавання класу в шкільний розклад.

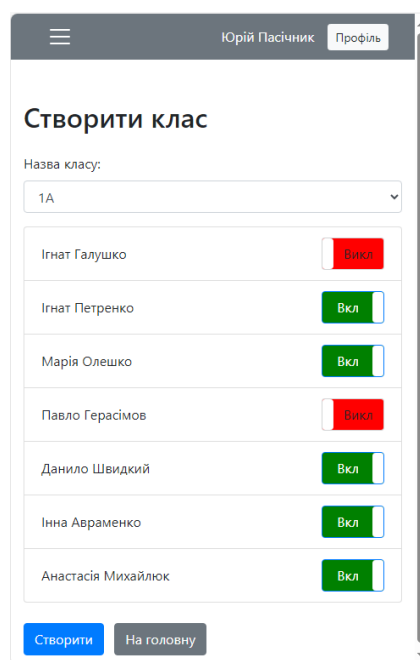
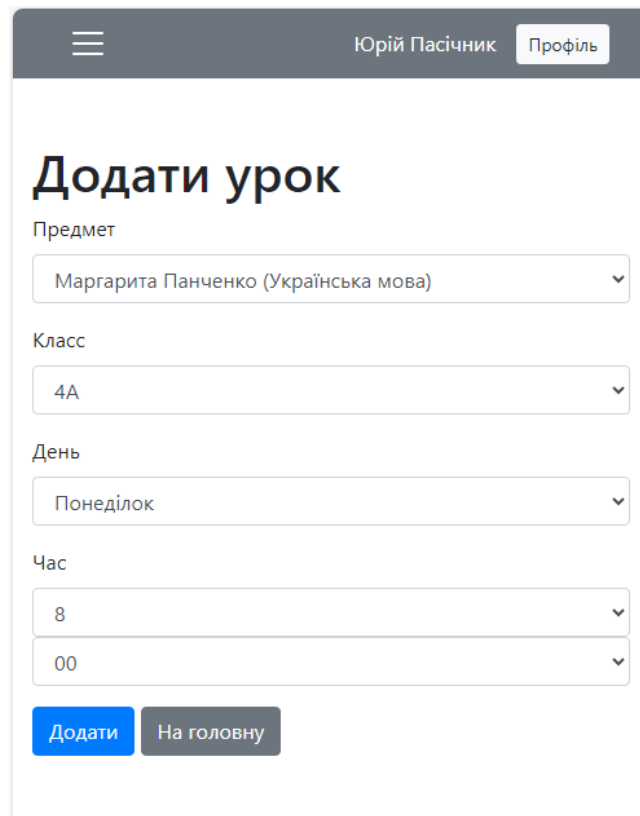


Рисунок 3.6 – Меню створення класу

В меню шкільного доступу відображається актуальний розклад всіх шкільних занять відсортирований за днями тижня. Для вчителів розклад містить усі доступні класи з маркуванням тих класів, предмет яких проводить той вчитель, який переглядає меню. Для учнів меню містить відсортовані за днями тижня предмети, до яких відноситься учень. В даному меню вчитель має можливість додати новий предмет до розкладу.

При створенні нового уроку за розкладом можливо додавати лише уроки за предметами, які були створені та закріплені за вчителем раніше. Також закріпити проведення уроку відбувається лише за сформованими класами (рис. 3.7). День і час проведення не обмежується попередніми діями.



Юрій Пасічник Профіль

Додати урок

Предмет

Маргарита Панченко (Українська мова)

Клас

4А

День

Понеділок

Час

8

00

Додати На головну

Рисунок 3.7 – Меню додавання уроку до розкладу

Кожен урок, доданий до розкладу має інформацію про терміни проходження завдань. Вчителі отримують можливість створити завдання або тести використовуючи функції штучного інтелекту. Для цього треба вибрати формат домашнього завдання і створити опис вимог. У випадку тестів до запиту додається системне уточнення до формату формування відповіді штучного

інтелекту. Відповіді штучного інтелекту зберігаються і в подальшому використовуються для відображення і перевірки інформації після виконання учнями (рис. 3.8).

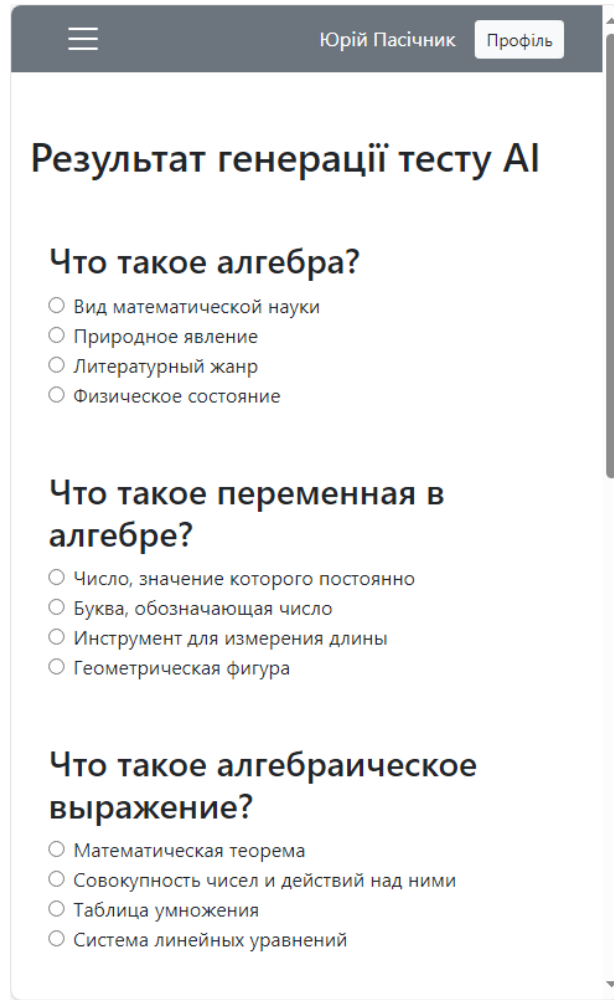


Рисунок 3.8 – Відображення обробленого результату відповіді штучного інтелекту

Після проходження тесту або завдання система підраховує бали та заносить до бази даних. Учні і батьки автоматично отримують інформацію про нараховані бали за урок за допомогою відправки повідомлень на застосунок обміну повідомлень Telegram, використаний при реєстрації в нашій освітній системі.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи був створений додаток, який демонструє потенціал симбіозу онлайн-навчання та традиційного підходу до освіти, з використанням штучного інтелекту при створенні завдань. Залучення можливостей штучного інтелекту у розробленій системі сприяє автоматизації процесу створення завдань, що дуже корисне у гібридних форматах навчання, де потрібно поєднувати віддалене та традиційне навчання.

Ключовою перевагою такого підходу є збереження якості навчального матеріалу, а також можливість індивідуалізації навчання, навіть якщо вчителя немає на місці. Більше того, генерація завдань в один клік на основі найсвіжіших даних дозволяє створювати індивідуальні тести для кожної групи чи учня, що перешкоджає можливості списування, поліпшуючи аналіз успішності.

Такий підхід забезпечує не лише максимальну ефективність навчального процесу, а й підвищує його чесність та рівень персоналізації. Відтак, наукові дослідження у напрямку розвитку взаємодії освіти з штучним інтелектом мають великий потенціал для подальшого вдосконалення навчального середовища та підвищення якості освіти загалом.

ПЕРЕЛІК ПОСИЛАНЬ

1. Jake Bryant, Christine Heitz, Saurabh Sanghvi, Dilip Wagle. Назва публікації з інтернет-сторінки. Загальна назва ресурсу. URL : <https://www.mckinsey.com/industries/education/our-insights/how-artificial-intelligence-will-impact-k-12-teachers/>. 2020. (дата звернення: 05.03.2024).
2. The fastest and most powerful platform for building AI products. URL : <https://core.telegram.org/bots>(дата звернення: 01.05.2024).
3. 10-те най-добри AI інструмента за образование. URL: <https://www.unite.ai/bg/10-best-ai-tools-for-education/>. (дата звернення: 10.03.2024).
4. Why Hidden Artificial Intelligence Features Make Such an Impact in Education. URL: <https://www.edsurge.com/news/2023-03-06-why-hidden-artificial-intelligence-features-make-such-an-impact-in-education>. (дата звернення: 14.03.2024).
5. Maher Joe Khan Omar Jian. Personalized learning through AI. URL: https://www.researchgate.net/publication/376814707_Personalized_learning_through_AI. 2023. (дата звернення: 24.04.2024).

ДОДАТОК А

Створення бази даних та структури таблиць

```

-- Створення бази даних
CREATE DATABASE edu_database;
USE edu_database;

-- Створення таблиці edu_users
CREATE TABLE edu_users (
  user_id INT(11) AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(20) NOT NULL,
  surname VARCHAR(26) NOT NULL,
  birthday DATE NOT NULL,
  is_student TINYINT(1) DEFAULT 0,
  is_parent TINYINT(1) DEFAULT 0,
  is_teacher TINYINT(1) DEFAULT 0,
  unread SMALLINT(5) DEFAULT 0,
  telegram_uid INT(11) DEFAULT NULL
);

-- Створення таблиці edu_classes
CREATE TABLE edu_classes (
  class_id INT(11) AUTO_INCREMENT PRIMARY KEY,
  classname VARCHAR(5) DEFAULT NULL,
  class_teacher_id INT(11) DEFAULT 0
);

-- Створення таблиці edu_class_users
CREATE TABLE edu_class_users (
  user_id INT(11),
  class_id INT(11),
  PRIMARY KEY (user_id, class_id),
  FOREIGN KEY (user_id) REFERENCES edu_users(user_id) ON DELETE CASCADE,
  FOREIGN KEY (class_id) REFERENCES edu_classes(class_id) ON DELETE CASCADE
);

-- Створення таблиці edu_shedule
CREATE TABLE edu_shedule (
  shedule_id INT(11) AUTO_INCREMENT PRIMARY KEY,
  class_id INT(11),
  shedule_day TINYINT(1) DEFAULT 0,
  shedule_time TIME,
  is_active TINYINT(1) DEFAULT 1,
  FOREIGN KEY (class_id) REFERENCES edu_classes(class_id) ON DELETE CASCADE
);

-- Створення таблиці edu_programs

```

```
CREATE TABLE edu_programs (  
  program_id INT(11) AUTO_INCREMENT PRIMARY KEY,  
  teacher_id INT(11),  
  name VARCHAR(40),  
  FOREIGN KEY (teacher_id) REFERENCES edu_users(user_id) ON DELETE SET NULL  
);  
  
-- Створення таблиці edu_lessons  
CREATE TABLE edu_lessons (  
  lesson_id INT(11) AUTO_INCREMENT PRIMARY KEY,  
  lesson_uri VARCHAR(255) UNIQUE,  
  shedule_id INT(11),  
  lesson_expire DATE,  
  is_expire TINYINT(1) DEFAULT 0,  
  FOREIGN KEY (shedule_id) REFERENCES edu_shedule(shedule_id) ON DELETE  
CASCADE  
);  
  
-- Створення таблиці edu_points  
CREATE TABLE edu_points (  
  user_id INT(11),  
  lesson_id INT(11),  
  time_point DATE,  
  points SMALLINT(5) DEFAULT NULL,  
  PRIMARY KEY (user_id, lesson_id),  
  FOREIGN KEY (user_id) REFERENCES edu_users(user_id) ON DELETE CASCADE,  
  FOREIGN KEY (lesson_id) REFERENCES edu_lessons(lesson_id) ON DELETE CASCADE  
);
```

ДОДАТОК Б

API запитів до штучного інтелекту OpenAI

```

<?php

class ChatGPTClient
{
    private $apiKey;
    private $apiUrl = "https://api.openai.com/v1/engines/davinci-codex/completions";

    public function __construct($apiKey)
    {
        $this->apiKey = $apiKey;
    }

    public function sendRequest($prompt)
    {
        $data = [
            'prompt' => $prompt,
            'max_tokens' => 100,
            'temperature' => 0.5
        ];

        $ch = curl_init($this->apiUrl);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_HTTPHEADER, [
            'Content-Type: application/json',
            'Authorization: Bearer ' . $this->apiKey,
        ]);
        curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($data));

        $response = curl_exec($ch);
        $error = curl_error($ch);
        curl_close($ch);

        if ($error) {
            throw new Exception("cURL Error: $error");
        }

        return json_decode($response, true);
    }
}

try {
    $apiKey = $config_ai['key'];
    $request = $_POST['request'];
}

```

```
$chatGPTClient = new ChatGPTClient($apiKey);  
$response = $chatGPTClient->sendRequest($request);  
  
echo json_encode($response, JSON_PRETTY_PRINT);  
} catch (Exception $e) {  
    echo 'Помилка: ' . $e->getMessage();  
}  
?>
```

ДОДАТОК В

API запитів до Telegram Bot

```
<?php
$botToken = $config_tg['secret_key'];

function send_message($uid, $text) {
    global $botToken;

    $telegramApiUrl = "https://api.telegram.org/bot$botToken/sendMessage";

    $data = [
        'chat_id' => $uid,
        'text' => $text
    ];

    $ch = curl_init($telegramApiUrl);

    curl_setopt($ch, CURLOPT_POST, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($data));

    $response = curl_exec($ch);

    curl_close($ch);

    return $response;
}

$update = json_decode(file_get_contents('php://input'), TRUE);

if (isset($update['message'])) {
    $message = $update['message'];
    $chatId = $message['from']['id'];
    $text = $message['text'];
}

?>
```


ДОДАТОК Г

Набір функцій інтерфейсу користувачів

```
<?php

function getUserRole($is_student, $is_teacher, $is_parent) {
    if ($is_student == 1) {
        return 'Учень';
    } elseif ($is_teacher == 1) {
        return 'Вчитель';
    } elseif ($is_parent == 1) {
        return 'Батько';
    } else {
        return 'Не вказано';
    }
}

function getUserColor($is_student, $is_teacher, $is_parent) {
    if ($is_student == 1) {
        return 'badge-success';
    } elseif ($is_teacher == 1) {
        return 'badge-dark';
    } elseif ($is_parent == 1) {
        return 'badge-info';
    } else {
        return 'badge-light';
    }
}

function generateOptionList()
{
    $letters = ['A', 'Б', 'В', 'Г'];
    $options = "";

    for ($i = 1; $i <= 12; $i++) {
        foreach ($letters as $letter) {
            $value = $i . $letter;
            $options .= '<option value="' . $value . '">' . $value . '</option>';
        }
    }

    return $options;
}

function dayList() {
```

```

$days = ['Понеділок', 'Вівторок', 'Середа', 'Четвер', 'П'ятниця', 'Субота'];
$list = [];
foreach ($days as $index => $day) {
    $list[] = '<option value="" . ($index + 1) . "">' . $day . '</option>';
}
return implode("", $list);
}

function hourList() {
    $list = [];
    for ($i = 8; $i <= 17; $i++) {
        $list[] = '<option value="" . $i . "">' . $i . '</option>';
    }
    return implode("", $list);
}

function minList() {
    $list = [];
    for ($i = 0; $i < 60; $i += 5) {
        $formatted = str_pad($i, 2, '0', STR_PAD_LEFT);
        $list[] = '<option value="" . $formatted . "">' . $formatted . '</option>';
    }
    return implode("", $list);
}

function programList($program_ids, $program_names) {
    $list = [];
    for ($i = 0; $i < count($program_ids); $i++) {
        $id = $program_ids[$i];
        $name = $program_names[$i];
        $list[] = '<option value="" . $id . "">' . $name . '</option>';
    }
    return implode("", $list);
}

function classList($class_ids, $class_names) {
    $options = "";
    for ($i = 0; $i < count($class_ids); $i++) {
        $options .= '<option value="" . $class_ids[$i] . "">' . $class_names[$i] . '</option>';
    }
    return $options;
}

function dayName($id) {
    $days = [
        "1" => "Понеділок",
        "2" => "Вівторок",
        "3" => "Середа",
        "4" => "Четвер",
        "5" => "П'ятниця",
        "6" => "Субота"
    ]
}

```

```

];

return isset($days[$id]) ? $days[$id] : "Невідомий день";
}

function generateOptions($min, $max) {
    $options = "";
    for ($i = $min; $i <= $max; $i++) {
        $options .= "<option value=\"\$i\">\$i</option>\n";
    }
    return $options;
}

function lessonHash($param) {
    $hash = md5($param + rand(1000));
    return $hash;
}

function generateUserStatus($is_user, $is_teacher, $is_parent) {
    $options = [];

    $allOptions = [
        "0" => "Не вказано",
        "1" => "Учень",
        "2" => "Вчитель",
        "3" => "Батько"
    ];

    $selectedValue = "0";

    if ($is_user == 1) {
        $selectedValue = "1";
    } elseif ($is_teacher == 1) {
        $selectedValue = "2";
    } elseif ($is_parent == 1) {
        $selectedValue = "3";
    }

    foreach ($allOptions as $value => $label) {
        $selected = ($value == $selectedValue) ? ' selected' : '';
        $options[] = "<option value=\"\$value\"$selected>\$label</option>";
    }

    return implode("", $options);
}

?>

```