

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Системотехніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Інтеграція елементів штучного інтелекту у систему онлайн навчання
(тема)

Виконав:

студент 2 курсу, групи ІТПм-22-1

Бродягіна Д.А.

(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма «Інформаційні технології проектування»

(повна назва освітньої програми)

Керівник проф. Калита Н.І.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

_____ (підпис)

Гребеннік І.В.

(прізвище, ініціали)

2024 р.

Я як студент ХНУРЕ розумію і підтримую політику закладу із академічної доброчесності. Я не надавав і не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

13.01.2023

 Бродягіна Д.А.

Кваліфікаційна робота не містить відомостей заборонених до відкритого опублікування.

Керівник кваліфікаційної роботи



Кваліфікаційна робота виконана у відповідності до стандартів, що діють в Україні.

Керівник кваліфікаційної роботи



Попередній захист проведено 13.01.24

1. Керівник кваліфікаційної роботи



Факультет _____ Комп'ютерних наук _____
 Кафедра _____ Системотехніки _____
 Рівень вищої освіти _____ другий (магістерський) _____
 Спеціальність _____ 122 Комп'ютерні науки _____
 (код і повна назва)
 Тип програми _____ освітньо-професійна _____
 Освітня програма _____ «Інформаційні технології проектування» _____
 (повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
 (підпис)

« _____ » _____ 20 _____ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Бродягіній Дарині Анатоліївні _____
 (прізвище, ім'я, по батькові)

- Тема роботи: Інтеграція елементів штучного інтелекту у систему онлайн навчання
 затверджена наказом по університету від 20.11. 2023 р. № 1373Ст
 - Термін подання студентом роботи до екзаменаційної комісії: 16.02.2024 р
 - Вихідні дані до роботи: Розробити інтеграцію Штучного інтелекту в систему онлайн-навчання з метою генерації контенту за обраною мовою запита. Розробити відправлення запиту та отримання його відповіді їх відображення
-
- Перелік питань, що потрібно опрацювати в роботі: 4.1 Вступ 4.2 Аналіз предметної області 4.2.1 Опис сучасного стану розвитку систем онлайн навчання 4.2.2 Аналіз застосування досліджуваних технологій в існуючих системах 4.3 Постановка задачі на розробку 4.3.1 Постановка задачі на дослідження 4.3.2 Персоналізоване навчання 4.3.3 Автоматичне оцінювання 4.3.4 Чат-боти для підтримки навчання 4.3.5 Інтерактивні навчальні допоміжні інструменти 4.3.6 Індивідуальні рекомендації 4.3.7 Розробка проєктних рішень 4.3.7.1 Обґрунтування вибору ШІ 4.3.7.2 Багатомовність 4.3.7.3 Етичні міркування та безпека 4.3.7.4 Гнучкість у застосуванні, інтеграції та налаштування 4.3.8 Вимоги до програмного забезпечення 4.3.9 Опис програмного забезпечення 4.3.9.1 PHP – опенсорс проєкт 4.3.9.2 Швидка розробка платформи 4.3.9.3 Гарна сумісність та гнучкість 4.3.9.4 Сервіси з відмінною документацією та масштабованістю 4.3.9.5 Оновлення та підтримка 4.3.9.6 Можливості вибору хостингу чи сервера 4.3.9.7 Стабільна робота онлайн-сервісів 4.3.9.8 Валідація введення 4.3.9.9 Екранування виведення 4.3.9.10 Запобігання sql-ін'єкцій 4.3.9.11 Запобігання XSS 4.3.9.12 Запобігання CSRF 4.4 Реалізація поставленої задачі на дослідження 4.5 Висновки 4.6 Перелік джерел посилання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій: _____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на виконання роботи	16.10.2023	
	Аналіз сучасного стану систем онлайн навчання	17-23.12.2023	
	Аналіз існуючих систем онлайн навчання	24-31.12.2023	
	Вибір середовища розробки	01-17.12.2023	
	Розробка програми	18-30.12.2023	
	Оформлення пояснювальної записки та програмної документації	01-13.01.2024	
	Оформлення графічної частини та презентаційних матеріалів комп'ютерного захисту	13.01.2024	
	Представлення на рецензування	14.01.2024	
	Представлення кваліфікаційної роботи в ДЕК	16.01.2024	

Дата видачі завдання 16.10.2023 р.

Студент _____
(підпис)

Бродягіна Д.А.

Керівник роботи _____
(підпис)

проф Калига Н.І.
(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи: 57 с., 16 рис., 2 додатки 24 джерел інформації.

СИСТЕМА ОНЛАЙН НАВЧАННЯ, ШІ, LARAVEL, MYSQL, OPENAI, PAYPAL, PHP

Об'єкт дослідження – генерація контенту системи онлайн-навчання штучним інтелектом.

Предметом дослідження - інтеграція штучного інтелекту у web-сервіс реалізований мовою PHP, використовуючи фреймворк Laravel.

Мета дослідження – інтеграція елементів штучного інтелекту у систему онлайн навчання для пришвидшення процесу створення навчального контенту для системи онлайн-навчання та зменшення навантаження на викладачів і контент-менеджерів системи.

Методи дослідження – системний аналіз, технології онлайн навчання, штучний інтелект, методи та засоби розробки програмного забезпечення.

У роботі проведено аналіз існуючих систем онлайн навчання з інтегрованим штучним інтелектом, їх роль у системі та функціональність. Визначено можливі варіанти інтеграції штучного інтелекту в систему онлайн навчання. Був реалізований функціонал звернення та відображення відповіді від штучного інтелекту при створенні та редагуванні публікацій платформи.

Система була розроблена на основі завдання на кваліфікаційну роботу і може бути використана в реальних умовах.

Платформа призначена для корпоративних клієнтів і може використовуватися компаніями різних розмірів.

ABSTRACT

Master's Thesis: 57 pages, 16 figures, 2 appendices, 24 title.

LEARNING MANAGEMENT SYSTEM, AI, LARAVEL, MYSQL, OPENAI, PAYPAL, PHP

Research Object – The generation of content for an online learning system by artificial intelligence.

Subject of Research – The integration of artificial intelligence into a web service implemented in PHP language, using the Laravel framework.

Research Objective – To integrate artificial intelligence elements into the online learning system to accelerate the process of creating educational content for the online learning system and to reduce the workload on teachers and content managers of the system.

Research Methods – System analysis, online learning technologies, artificial intelligence, methods, and tools for software development.

In the work, an analysis of existing online learning systems with integrated artificial intelligence was conducted, focusing on their role in the system and functionality. Possible options for integrating artificial intelligence into the online learning system were identified. Functionality for making requests and displaying responses from artificial intelligence in the creation and editing of platform publications was implemented.

The system was developed based on a qualification work task and can be used in real conditions.

The platform is designed for corporate clients and can be used by companies of various sizes.

ЗМІСТ

Перелік умовних позначень	9
Вступ.....	10
1 Аналіз предметної області.....	12
1.1 Опис сучасного стану розвитку систем онлайн навчання.....	12
1.2 Аналіз застосування досліджуваних технологій в існуючих системах	17
2 Потановка задачі на розробку	21
2.1 Постановка задачі на дослідження.....	21
2.2 Персоналізоване навчання	22
2.3 Автоматичне оцінювання.....	23
2.4 Чат-боти для підтримки навчання.....	24
2.5 Інтерактивні навчальні допоміжні інструменти	25
2.6 Індивідуальні рекомендації.....	26
2.7 Розробка проєктних рішень	27
2.7.1 Обґрунтування вибору ШІ	27
2.7.2 Багатомовність	27
2.7.3 Етичні міркування та безпека	28
2.7.4 Гнучкість у застосуванні, інтеграції та налаштування	29
2.8 Вимоги до програмного забезпечення.....	29
2.9 Опис програмного забезпечення	32
2.9.1 РНР – опенсорс проєкт	34
2.9.2 Швидка розробка платформи	34
2.9.3 Гарна сумісність та гнучкість	35

	8
2.9.4 Сервіси з відмінною документацією та масштабованістю.....	36
2.9.5 Оновлення та підтримка.....	36
2.9.6 Можливості вибору хостингу чи сервера.....	37
2.9.7 Стабільна робота онлайн-сервісів.....	37
2.9.8 Валідація введення.....	38
2.9.9 Екранування виведення.....	38
2.9.10 Запобігання sql-ін'єкцій.....	39
2.9.11 Запобігання XSS.....	39
2.9.12 Запобігання CSRF.....	41
3 Реалізація поставленої задачі на дослідження.....	43
Висновки.....	56
Перелік джерел посилання.....	57
Додаток А Графічні матеріали.....	59
Додаток Б Текст програми.....	64

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних;

ІС – інформаційна система;

СУБД – система керування базами даних;

ШІ – штучний інтелект;

API (Application Programming Interface, API) — набір визначень підпрограм;

ER – Entity-Relation;

ID – identifier;

IDEF0 – нотація опису бізнес-процесів;

IDEF1X – методологія розробки моделей даних;

PDO - PHP Data Objects;

UX – досвід користувача;

ВСТУП

Актуальність теми. У епоху технологій, онлайн платформи для продажу курсів стали звичним явищем, зменшуючи попит на офлайн майстер-класи та школи мистецтв. Така ситуація спонукає митців переходити до онлайн формату, щоб привабити клієнтів. Онлайн платформи мають кілька переваг: вони звільняють від необхідності знаходити та орендувати простір, від оплати комунальних послуг, вони усувають потребу в оренді обладнання та його обслуговуванні. Крім того, вони зменшують витрати на персонал, таких як викладачі та менеджери, та не обмежують кількість користувачів кімнатним простором чи кількістю викладачів. Нарешті, ресурси доступні в будь-який час та з будь-якої точки світу з інтернет-з'єднанням. В сучасному освітньому контексті інтеграція штучного інтелекту в систему онлайн навчання обумовлена стрімким розвитком технологій штучний інтелект (ШІ) та зростаючою популярністю онлайн освіти, особливо в контексті глобальних викликів, починаючи з пандемії COVID-19, продовжуючи повномасштабним вторгненням росії на території України. Штучний інтелект відкриває нові можливості для персоналізації навчання, адаптування освітнього процесу до індивідуальних потреб та стилів навчання студентів. Водночас, він сприяє автоматизації адміністративних та навчальних процесів, знижуючи навантаження на викладачів та підвищуючи ефективність управління курсами. ШІ також може робити освіту більш доступною для різних груп населення, включаючи людей з особливими потребами або тих, хто проживає в віддалених регіонах. Крім того, з огляду на швидкі зміни на ринку праці, інтеграція ШІ в освіту важлива для підготовки студентів до викликів та професій майбутнього. Водночас, ця тема має значну важливість через необхідність розгляду етичних питань та потенційних викликів, пов'язаних з використанням ШІ в освітніх цілях, що допомагає формувати відповідальний та збалансований підхід до інтеграції цих технологій.

У якості предмета дослідження визначено інтеграцію ШІ в систему онлайн навчання для генерації контенту системи.

Об'єкт дослідження – генерація контенту системи онлайн-навчання штучним інтелектом. Предметом дослідження визначено інтеграцію штучного інтелекту у web-сервіс реалізований мовою PHP, використовуючи фреймворк Laravel.

Мета дослідження – інтеграція елементів штучного інтелекту у систему онлайн навчання для пришвидшення процесу створення навчального контенту для системи онлайн-навчання та зменшення навантаження на викладачів і контент-менеджерів системи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Опис сучасного стану розвитку систем онлайн навчання

Штучний інтелект (ШІ) став необхідністю в сучасному світі, що проявляється майже у всіх галузях нашого життя, від роботи та освіти до медицини та розваг. Він робить життя зручнішим та безпечнішим, допомагає вирішувати складні завдання та відкриває безліч нових можливостей, таких як створення роботів, здатних виконувати складні завдання, розробка інтелектуальних систем управління, розвиток автономних транспортних засобів та багато інших інновацій. Наприклад, в медицині ШІ вже допомагає швидше та точніше діагностувати захворювання [1], а в економіці - аналізувати ринкові тенденції. ШІ здатний аналізувати великі обсяги даних та приймати рішення на основі цієї аналітики. Управлінці можуть використовувати його для оптимізації бізнес-процесів, а вчені - для пошуку нових рішень у науці та дослідженнях, розв'язуючи складні проблеми та вдосконалюючи свої рішення. ШІ є ключовою технологією майбутнього, адже він постійно розвивається і приводить до створення нових продуктів та послуг, які полегшують життя кожної людини. Інтеграція штучного інтелекту в системи онлайн навчання відкриває можливість покращення якості освіти та забезпечує доступ до неї.

Системи онлайн навчання стали ключовим елементом цього трансформаційного процесу [2], надаючи студентам та учням унікальні можливості доступу до знань та навчання в будь-який час і в будь-якому місці. Вони дозволяють революціонізувати процес освіти, роблячи його більш доступним, гнучким та персоналізованим для кожного учня, допомагають забезпечити освіту в будь-який час для людей з різних країн та географічних регіонів, що робить їх невід'ємною частиною сучасної освіти; знижують бар'єри щодо відстані та місця проживання, дозволяючи учням отримувати освіту без потреби бути фізично присутнім на заняттях чи в навчальних закладах; дають учням можливість вибирати гнучкий графік навчання, навіть у вечірні години та вихідні, що є зручним для працюючих людей або тих, хто має затісний графік.

Системи онлайн навчання пропонують різноманітні формати, такі як відеоуроки, інтерактивні вправи, віртуальні лекції, вебінари та інше, забезпечуючи можливість вибору таких навчальних ресурсів, які найкраще відповідають потребам учня та його стилю навчання. Свою популярність системи онлайн-навчання почали активно набирати на початку COVID-19 та продовжує набирати популярність з початку повномасштабного вторгнення. На рисунку 1.1 зображено відношення кількості користувачів систем до року починаючи з 2010.



Рисунок 1.1 – Графік використання систем онлайн-навчання

Онлайн навчання відкриває безмежні можливості для доступу до освіти, але водночас несе з собою визначені недоліки, які важливо враховувати при розробці та впровадженні цього формату навчання.

Одним з основних недоліків онлайн навчання є брак безпосереднього, очного спілкування між студентами та викладачами. Це не просто віддаляє студентів від традиційного середовища навчання, але й ускладнює процес обговорення навчального матеріалу, обміну думками та ідей. Електронні листи чи чати не

можуть повністю замінити живе спілкування та емоційний обмін, що відбувається під час особистої бесіди. Багато дисциплін вимагають практичного втілення отриманих знань, наприклад, лабораторних робіт, експериментів, або фізичних тренувань. Онлайн формат часто не може забезпечити належних умов для такого роду діяльності, що обмежує різноманітність та якість навчального процесу.

Такого типу навчання вимагає високого рівня самодисципліни та самомотивації від студентів. Відсутність фізичної присутності в класі та менш строгий контроль з боку викладачів може призводити до відкладання завдань, зниження концентрації та втрати інтересу до навчання. Для деяких студентів подолання цих викликів може стати непереборним.

З розвитком мобільних додатків для навчання, таких як Duolingo [3] для вивчення мов або Photomath [4] для вирішення математичних задач, мобільне навчання стало невід'ємною частиною освітнього процесу для багатьох людей. Інший важливий аспект мобільного навчання - це адаптивне навчання, де контент та методики викладання адаптуються під індивідуальні потреби кожного студента. Мобільні технології значно спростили реалізацію цього підходу завдяки їх персоналізованості та інтерактивності. Системи адаптивного навчання використовують алгоритми штучного інтелекту для аналізу відповідей студентів та налаштування навчального процесу відповідно до їхніх потреб та прогресу.

Онлайн навчання, несумнівно, внесло великі зміни не тільки в життя студентів, а і в професійне життя викладачів. Вони стикаються з необхідністю інтегрувати новітні технології у навчальний процес. Це передбачає вивчення та застосування спеціалізованого програмного забезпечення, цифрових платформ для відеоконференцій, а також інших інструментів цифрового навчання.

Змінилися й методи викладання. Те, що раніше можна було донести за допомогою живого діалогу та безпосередньої взаємодії, тепер потребує нових підходів. Викладачі адаптують свої лекції, перетворюючи їх у цифровий формат, намагаючись зробити матеріал не лише інформативним, але й залучаючим. Вони постійно шукають способи зберегти інтерактивність та відчуття спільності, незважаючи на фізичну віддаленість.

Ще одним випробуванням є забезпечення ефективного спілкування та обратного зв'язку. Без можливості побачити вираз обличчя студента або його реакцію на пояснення, важко оцінити рівень розуміння матеріалу та вчасно коригувати підхід до викладання.

І, звісно, не можна не згадати про робоче навантаження. Підготовка до онлайн занять, ведення переписки зі студентами, оцінювання робіт – все це вимагає значного часу та зусиль. Викладачі часто опиняються в ситуації, коли робочий день не має чітких меж, а завдання вимагають безперервної уваги.

Проте, онлайн навчання також відкрило двері до нових можливостей для професійного зростання. Участь у вебінарах, онлайн курсах, обмін досвідом з колегами з різних куточків світу – все це стало більш доступним.

Незважаючи на значний потенціал, мобільне навчання також стикається з викликами, такими як питання цифрового розриву, де не всі студенти мають однаковий доступ до мобільних пристроїв або інтернету. Інша проблема - це збереження уваги студентів в умовах постійних відволікань, таких як соціальні мережі. Однак, перспективи мобільного навчання залишаються вражаючими, особливо в контексті неперервного технологічного прогресу та зростання інтересу до освіти, що базується на гнучкості та доступності. Мобільні технології не лише розширюють можливості для навчання, але й відкривають нові горизонти для індивідуалізації та ефективності освітнього процесу.

Розвиток систем онлайн-навчання, інтегрованих зі штучним інтелектом, є однією з найбільш перспективних тенденцій в освітній сфері сьогодення. Ця інтеграція відкриває нові горизонти для персоналізації навчання та підвищення його ефективності.

Інтеграція ШІ у системи онлайн навчання розпочалася як відповідь на зростаючі потреби сучасної освіти у підвищенні ефективності, персоналізації та доступності навчального процесу. У світі, де цифрові технології швидко розвиваються, а вимоги до гнучкості та індивідуалізації навчання зростають, ШІ надає нові можливості для оптимізації та інновацій в освітньому процесі.

Переваги інтеграції ШІ включають здатність до персоналізації навчання, підвищення його ефективності та гнучкості, автоматизацію рутинних завдань, а також забезпечення додаткової підтримки студентам та викладачам. Це включає розробку персоналізованих навчальних планів, автоматичне оцінювання, підтримку через чат-боти та надання індивідуальних рекомендацій, які допомагають підвищити залученість та мотивацію студентів.

Однак інтеграція ШІ несе в собі й певні виклики та недоліки [5]. Основними з них є високі вимоги до якості даних та необхідність їх постійного оновлення, потенційні етичні питання, пов'язані з конфіденційністю та безпекою даних, а також ризик зменшення людської взаємодії в освітньому процесі. Також існує виклик у забезпеченні об'єктивності та неперервного вдосконалення ШІ-систем, щоб вони відповідали різним освітнім потребам та вимогам.

Фінансова сторона інтеграції ШІ також є важливою. Хоча впровадження ШІ може вимагати значних початкових інвестицій у розробку та інтеграцію систем, воно може принести довгострокову економію завдяки автоматизації процесів та оптимізації ресурсів. Також існує потенціал для зменшення витрат на навчальний персонал та адміністративні послуги за рахунок автоматизації рутинних завдань.

Системи з ШІ використовують дані про взаємодію студентів із навчальним матеріалом для створення індивідуалізованого навчального досвіду. ШІ аналізує відповіді студентів, їхні успіхи та труднощі, адаптуючи навчальний контент таким чином, щоб він відповідав їхнім потребам. Наприклад, якщо студент виявляє труднощі з певною темою, система може запропонувати додаткові ресурси або завдання для поглибленого вивчення цієї теми.

Крім адаптації навчального процесу, ШІ також використовується для створення більш динамічних і взаємодіючих форм навчання. Це може включати в себе інтерактивні вправи, ігри, симуляції, які роблять навчання більш захоплюючим і ефективним.

Штучний інтелект також може сприяти автоматизації оцінювання та відгуків [6]. Використовуючи ШІ для аналізу відповідей студентів, системи можуть

надавати миттєві та персоналізовані відгуки, що значно підвищує швидкість та ефективність навчального процесу.

Однак, з розвитком систем онлайн-навчання з інтегрованим ШІ також пов'язані певні виклики. Це стосується питань конфіденційності та безпеки даних, а також потреби в розробці алгоритмів, які будуть справедливими та неупередженими. Крім того, існує необхідність у забезпеченні доступності цих технологій для всіх студентів, незалежно від їхнього географічного розташування та фінансових можливостей.

1.2 Аналіз застосування досліджуваних технологій в існуючих системах

У сучасному світі онлайн-навчання, багато платформ інтегрують штучний інтелект (ШІ) для забезпечення більш ефективного та індивідуалізованого навчального досвіду. Прикладом такої платформи є відома Coursera [7], яка є однією з лідерів у сфері онлайн освіти, пропонуючи сертифіковані курси від провідних університетів та інституцій світу. Coursera використовує машинне навчання для рекомендації курсів, заснованих на інтересах та попередньому прогресі студентів, і має можливість автоматично перевіряти завдання у деяких курсах. На рисунку 1.2 показано як виглядає користувацький інтерфейс Coursera:

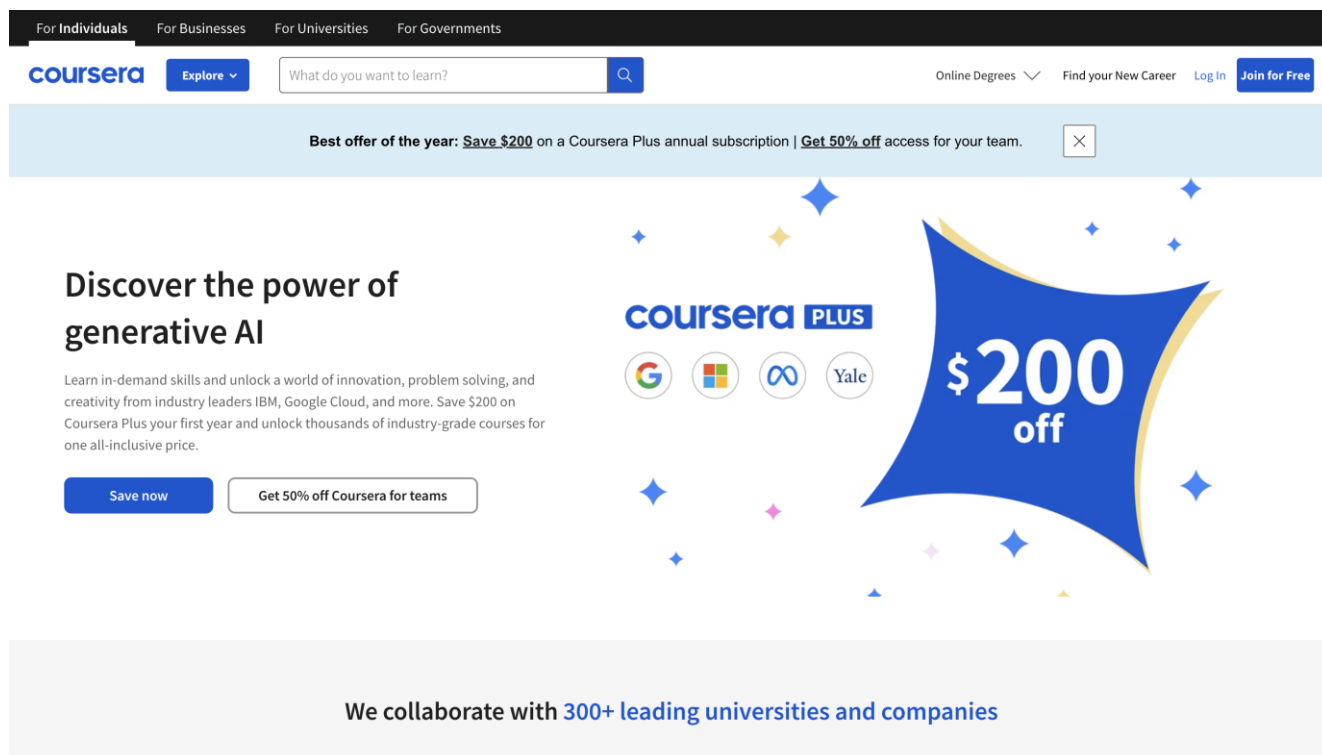


Рисунок 1.2 - Користувацький інтерфейс Coursera

Khan Academy [8], інша впливова освітня платформа, пропонує безкоштовний доступ до широкого спектру навчальних матеріалів у різноманітних дисциплінах. Ця платформа використовує ШІ для адаптації навчального контенту та завдань відповідно до індивідуальних потреб та можливостей кожного студента, а також надає детальні звіти про їхні досягнення та прогрес. На рисунку 1.3 показано як виглядає користувацький інтерфейс Khan Academy:

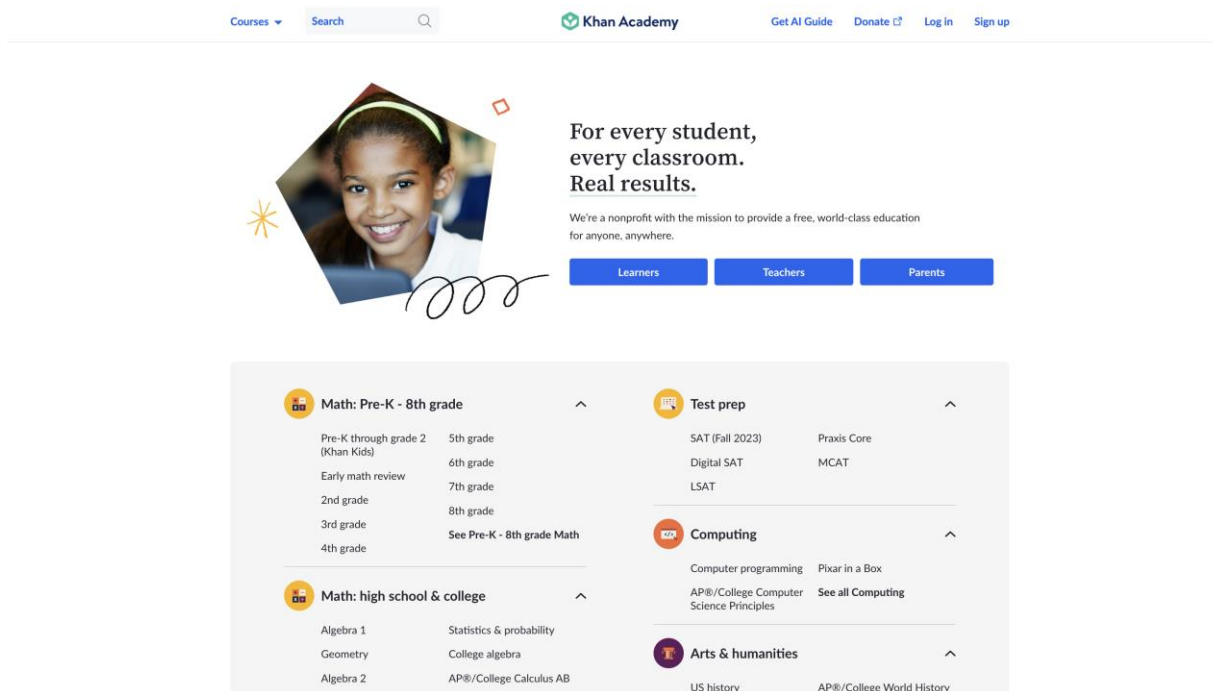


Рисунок 1.3 - Користувацький інтерфейс Khan Academy

Більшість інших платформ, таких як edX [9] та Udacity [10], також впроваджують ШІ для рекомендацій навчальних матеріалів або як частину своєї онлайн-підтримки через чат. Це дозволяє студентам отримувати більш персоналізоване навчання та підтримку, підвищуючи загальну якість та ефективність освітнього процесу.

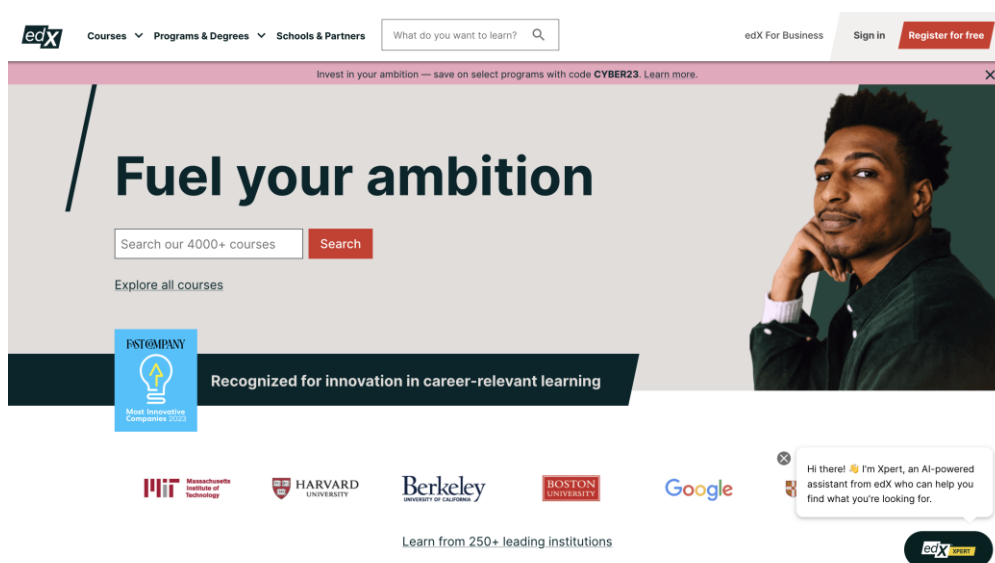


Рисунок 1.4 - Користувацький інтерфейс edX

Отже, системи онлайн-навчання з інтегрованим ШІ мають величезний потенціал для трансформації освітнього середовища. Вони пропонують новаторський підхід до навчання, який може бути адаптований під індивідуальні потреби кожного студента, одночасно відкриваючи двері для більш ефективного та доступного освітнього досвіду. Як і у випадку з будь-якими новими технологіями, важливо збалансувати інновації з відповідальним управлінням та етичними розглядами.

2 ПОСТАНОВКА ЗАДАЧІ НА РОЗРОБКУ

2.1 Постановка задачі на дослідження

Необхідно реалізувати інтеграцію ІІІ у систему онлайн навчання задля підвищення ефективності роботи викладачів системи та заохочення потенційних студентів до навчання саме на цій платформі.

Об'єкт дослідження – генерація текстового контенту для платформи онлайн-навчання.

Метою дослідження є створення навчального матеріалу з допомогою ІІІ, що дозволяє прискорити процес розробки навчального контенту, зменшити навантаження на викладачів та забезпечити доступ до актуальних матеріалів. Зі сторони організації це дозволить зменшити штат співробітників, прискорити процес наповнення сайту контентом та постійно його оновлювати новим. Викладач, сформувавши запит за потрібною тематикою, отримує запропонований ІІІ матеріал, який можна відредагувати та опублікувати.

Інтеграція ІІІ до системи онлайн-навчання включає в себе локалізацію обраною мовою для надання можливості вивчати матеріал іншими мовами широкій аудиторії користувачів. Реалізація надання підтримки у вигляді мовної адаптації студентам, для яких мова системи не є рідною, дозволяє розширити доступ до освіти та полегшує навчання для студентів з різних країн.

Необхідно проаналізувати і обрати технології, що будуть використовуватися для розробки і власне створити програмний продукт, в якому повинен бути реалізований ряд функцій:

- введення запиту на генерацію;
- отримання відповіді на запит;
- локалізація відповіді за обраною мовою.

Для оптимізації роботи викладачі системи онлайн навчання необхідно додати поле введення запиту до інтерфейсу редагування та створення постів. Доступ до цього функціоналу матиме контент-менеджер, викладач та адміністратор сайту.

Результатом роботи є згенерована відповідь на запит обраною користувачем мовою.

2.2 Персоналізоване навчання

Персоналізоване навчання з використанням штучного інтелекту в онлайн освіті відкриває нові горизонти в індивідуальному підході до кожного студента. Використовуючи складні алгоритми аналізу даних, системи здатні глибоко аналізувати поведінкові та навчальні патерни студентів, включаючи їхні відповіді, час, проведений на вивчення певних тем, та інтерактивність з курсовим матеріалом. На основі цього аналізу система адаптує навчальний матеріал, пропонуючи теми, які найкраще відповідають рівню розуміння та інтересам кожного студента. Це означає, що два студенти, які проходять один і той же курс онлайн, можуть мати зовсім різний навчальний досвід, відповідний до їхнього індивідуального рівня знань та швидкості навчання. Такий підхід не тільки сприяє більш ефективному засвоєнню матеріалу, але й робить навчання більш захоплюючим та мотивуючим, оскільки кожен студент відчуває, що навчання відбувається відповідно до його особистих потреб і інтересів. Крім того, персоналізоване навчання з використанням ШІ дозволяє вчителям та освітнім закладам краще розуміти потреби своїх студентів, аналізувати ефективність навчальних матеріалів та швидко вносити корективи в освітній процес. Це також допомагає ідентифікувати студентів, які потребують додаткової підтримки, та пропонувати їм відповідні ресурси та допомогу.

Завдяки широкому спектру можливостей, що їх надає штучний інтелект, персоналізоване навчання стає не просто методом навчання, але й потужним інструментом, який може радикально змінити та покращити сучасну систему онлайн освіти, роблячи її більш гнучкою, доступною та ефективною для кожного студента.

2.3 Автоматичне оцінювання

Автоматичне оцінювання [11] за допомогою штучного інтелекту в системах онлайн навчання революціонує процес оцінювання студентських робіт. Цей підхід використовує алгоритми машинного навчання та обробки природної мови для автоматизації оцінки відповідей студентів, особливо в тестах з відкритими відповідями або есе. Це не тільки економить час викладачів, але й забезпечує об'єктивність та послідовність в оцінюванні, усуваючи людський фактор, який може призвести до суб'єктивних оцінок.

З використанням ШІ викладачі можуть зосередитися на більш складних та важливих аспектах навчання, таких як розробка курсів, менторство та забезпечення індивідуального підходу до студентів. Системи автоматичного оцінювання здатні швидко обробляти великі обсяги текстових відповідей, надаючи зворотний зв'язок студентам у значно коротші терміни, ніж це могло б бути зроблено вручну. Такий зворотний зв'язок може включати не тільки оцінки, але й коментарі щодо покращення робіт, ключових аспектів, на які слід звернути увагу, та рекомендацій для подальшого розвитку навичок.

Крім того, автоматичне оцінювання відкриває можливості для більш гнучких та інноваційних форм оцінювання. Наприклад, системи можуть враховувати не тільки відповіді, але й процес їх розробки, аналізуючи способи мислення та підходи студентів до розв'язання задач. Це дозволяє глибше зрозуміти навчальний процес та підходи студентів, а також ідентифікувати потенційні області для подальшого розвитку та підтримки.

Таким чином, інтеграція автоматичного оцінювання за допомогою штучного інтелекту в онлайн освіту значно підвищує ефективність та якість навчального процесу, забезпечуючи об'єктивність оцінок та підтримуючи викладачів у їхній роботі.

2.4 Чат-боти для підтримки навчання

Інтеграція чат-ботів на базі штучного інтелекту в системи онлайн навчання відкриває нові можливості для підтримки та залучення студентів. Ці інтелектуальні асистенти, які працюють цілодобово, забезпечують швидке реагування на запитання студентів, надаючи пояснення, керівництва та допомогу в навчальному процесі. Використання ШІ дозволяє чат-ботам не лише відповідати на стандартні запитання, але й адаптувати відповіді відповідно до контексту та індивідуальних потреб студентів, забезпечуючи більш персоналізований досвід. Чат-боти можуть бути використані для різноманітних цілей: від базової навігації по курсу та допомоги в організаційних питаннях до підтримки в засвоєнні складних тем та підготовці до екзаменів. Вони також можуть виконувати роль мотиваторів, нагадуючи про терміни здачі робіт та допомагаючи студентам в плануванні їх навчального процесу.

Важливою перевагою чат-ботів є їх здатність надавати миттєву взаємодію, що особливо цінно в онлайн навчанні, де студенти можуть відчувати себе ізольованими без безпосереднього контакту з викладачами та одногрупниками. Чат-боти забезпечують почуття підтримки та взаємодії, що є ключовим для збереження мотивації та залученості студентів. Крім того, використання чат-ботів на базі ШІ в освіті сприяє ефективності управління ресурсами, оскільки вони можуть обробляти велику кількість одночасних запитів, знижуючи навантаження на викладацький склад та адміністративний персонал. Це дозволяє освітнім закладам оптимізувати свої робочі процеси та зосередитися на більш важливих задачах, таких як розробка курсів та безпосередня робота зі студентами.

Таким чином, впровадження чат-ботів на базі штучного інтелекту в онлайн освіту вносить значний вклад у підвищення якості навчального досвіду, забезпечуючи персоналізовану підтримку та сприяючи ефективності освітнього процесу.

2.5 Інтерактивні навчальні допоміжні інструменти

Інтеграція інтерактивних навчальних допоміжних інструментів на базі штучного інтелекту в онлайн освітні системи відкриває нові можливості для підвищення ефективності та залучення студентів. Ці інструменти, такі як віртуальні лабораторії, симуляції та ігрові платформи, використовують ШІ для створення реалістичних та глибоких навчальних досвідів, які підходять для широкого спектра дисциплін, від природничих наук до мистецтва. Ці інструменти дозволяють студентам експериментувати в контрольованому, але гнучкому віртуальному середовищі, де вони можуть вивчати складні концепції та навички без ризику або високих витрат, які часто пов'язані з реальними лабораторними роботами або практичними заняттями. Наприклад, студенти медицини можуть використовувати віртуальні симуляції для відпрацювання хірургічних технік або студенти інженерії можуть проектувати та тестувати конструкції в симульованих умовах.

Використання ШІ в цих інструментах дозволяє адаптувати навчальний досвід до індивідуальних потреб та стилів навчання студентів, пропонуючи різні сценарії та рівні складності. Це не тільки покращує засвоєння матеріалу, але й робить навчання більш захоплюючим та мотивуючим, оскільки студенти бачать безпосередні результати своїх дій та експериментів. Крім того, інтерактивні інструменти забезпечують викладачам цінні дані про процес навчання студентів, дозволяючи ідентифікувати області, у яких студенти можуть мати труднощі, та вчасно надавати додаткову допомогу або ресурси. Це сприяє більш індивідуалізованому та ефективному підходу до навчання.

Таким чином, інтеграція інтерактивних інструментів на базі штучного інтелекту значно збагачує навчальний процес в онлайн освіті, роблячи його більш динамічним, залучаючим та ефективним для студентів різних спеціальностей.

2.6 Індивідуальні рекомендації

Використання штучного інтелекту для надання індивідуальних рекомендацій в системах онлайн навчання відіграє ключову роль у підвищенні якості та ефективності освітнього процесу. ШІ-орієнтовані алгоритми аналізують дані про навчальний прогрес та вподобання студентів, щоб надати персоналізовані рекомендації щодо курсів, навчальних матеріалів та навіть конкретних тем для подальшого вивчення. Ця технологія дозволяє студентам отримати більш цілеспрямоване навчальне досвід, забезпечуючи їм матеріали, які відповідають їх індивідуальному рівню знань, інтересам та навчальним цілям. Такий підхід не лише сприяє глибшому засвоєнню знань, але й підтримує мотивацію студентів, оскільки вони відчують, що навчальний процес враховує їхні унікальні потреби та інтереси.

Для викладачів та освітніх закладів це також означає можливість краще розуміти потреби своїх студентів. На основі даних, зібраних штучним інтелектом, вони можуть адаптувати свої навчальні плани, ресурси та методики, щоб забезпечити більш ефективне навчання. Крім того, систематичний аналіз навчальних трендів та результатів допомагає освітнім установам постійно вдосконалювати свої програми та курси. ШІ-підсилені рекомендаційні системи також можуть виявляти зони, в яких студенти можуть мати труднощі, та автоматично пропонувати додаткові ресурси або вправи для покращення розуміння матеріалу. Це не тільки забезпечує більш індивідуальний підхід до навчання, але й допомагає запобігати відставанню студентів у своєму навчальному прогресі.

Загалом, інтеграція штучного інтелекту для надання індивідуальних рекомендацій в онлайн освіті створює більш гнучку, адаптивну та студентоцентричну навчальну середу, що сприяє підвищенню якості освіти та забезпечує більш глибоке та ефективне засвоєння знань.

2.7 Розробка проєктних рішень

2.7.1 Обґрунтування вибору ШІ

OpenAI використовує передові алгоритми та велику базу даних для своїх текстових генераторів, що робить їх особливо ефективними у порівнянні з іншими системами. Моделі, як GPT (Generative Pre-trained Transformer), базуються на архітектурі трансформерів, яка вважається однією з найефективніших для розуміння та генерації мови. Ця архітектура заснована на механізмі уваги, що дозволяє моделі зосереджуватись на важливих аспектах тексту, формуючи точні та зв'язні відповіді.

Тренування моделей відбувається на величезних обсягах даних, включаючи книги, веб-сторінки, статті та інші форми письмової мови. Це забезпечує глибоке розуміння мови та її контекстів. Крім того, попереднє навчання на цих даних дає моделям здатність загального розуміння мови, яке потім може бути адаптовано до конкретних завдань за допомогою тонкого налаштування.

Використання технік машинного навчання дозволяє моделям OpenAI постійно вчитися та оптимізувати свої відповіді на основі зворотного зв'язку та нових даних. Це забезпечує їх постійне вдосконалення і покращення можливостей генерації тексту. Такий підхід гарантує, що текстові генератори OpenAI можуть ефективно вирішувати складні завдання обробки мови, створюючи якісний, релевантний і зв'язний текст.

2.7.2 Багатомовність

Використання OpenAI включає багатомовні можливості, які значно розширюють сферу його застосування. Текстові генератори OpenAI, такі як моделі GPT, підтримують ряд мов, що робить їх доступними для міжнародної аудиторії. Ця багатомовність не обмежується лише основними світовими мовами, але також включає різноманіття менш поширених мов, що дозволяє користувачам з різних

країн та культурних середовищ взаємодіяти з технологією OpenAI у їхній рідній мові.

Ця багатомовність значно підвищує доступність та корисність генераторів тексту для широкого спектра застосувань, включаючи міжнародне спілкування, навчання мовам, культурний обмін та глобальний бізнес. Можливість генерувати та розуміти текст на різних мовах також відкриває двері для більш широкого діапазону освітніх, культурних та комерційних можливостей, сприяючи глобалізації знань та інформації. Завдяки цьому, OpenAI є не тільки потужним інструментом у генерації тексту, але й важливим засобом у зміцненні міжкультурного діалогу та обміну.

2.7.3 Етичні міркування та безпека

OpenAI приділяє велику увагу етичним міркуванням та безпеці у своїх текстових генераторах, що є важливим аспектом у світі штучного інтелекту. Компанія розробляє свої системи з урахуванням потенційних ризиків, пов'язаних з автоматичною генерацією тексту, як-от поширення неправдивої інформації, упередженість та неналежне використання. Щоб забезпечити етичне використання своїх технологій, OpenAI впроваджує різні механізми безпеки та фільтрації, які допомагають запобігти зловживанням та мінімізувати можливість упередженості.

Ці зусилля включають розробку алгоритмів, які виявляють і фільтрують небажаний контент, наприклад, мову ворожнечі або неправдиву інформацію. Також важливою є робота над усуненням упередженості в алгоритмах, щоб забезпечити справедливе та об'єктивне представлення інформації. Крім того, OpenAI активно співпрацює з дослідницькою спільнотою та користувачами для збору зворотного зв'язку та постійного вдосконалення своїх систем.

Ці заходи гарантують, що технології OpenAI не тільки ефективні та інноваційні, але й безпечні та відповідальні у своєму використанні. Зосередження на етичних міркуваннях та безпеці допомагає OpenAI встановлювати високі

стандарти в галузі штучного інтелекту та забезпечувати довіру користувачів до своїх продуктів.

2.7.4 Гнучкість у застосуванні, інтеграції та налаштування

OpenAI вирізняється своєю гнучкістю у застосуванні та можливостями інтеграції та налаштування, що робить його ідеальним для широкого спектра застосувань. Моделі, розроблені OpenAI, такі як GPT, можуть бути адаптовані до різних потреб і сценаріїв використання, від простих чат-ботів до складних систем обробки природної мови. Ця адаптивність дозволяє використовувати технології OpenAI в різних галузях, включаючи освіту, бізнес, медіа та розваги.

Щодо інтеграції та налаштування, OpenAI пропонує гнучкі інтерфейси та API, які дозволяють розробникам легко інтегрувати його технології у власні додатки та сервіси. Це дає можливість користувачам налаштувати функціональність моделей згідно з їх конкретними потребами, наприклад, налаштувати стиль відповідей, враховувати специфіку домену чи використовувати спеціальні набори даних для підвищення точності відповідей.

Ця гнучкість та можливості налаштування роблять OpenAI не лише потужним інструментом для генерації тексту, але й важливим активом для інновацій та креативності в різних областях. Відкритість та адаптивність технологій OpenAI сприяють створенню більш інтелектуальних, інтерактивних та корисних цифрових досвідів для користувачів по всьому світу.

2.8 Вимоги до програмного забезпечення

Для створення онлайн платформи з інтегрованим штучним інтелектом обрано мову програмування PHP [12] та фреймворк Laravel [13].

PHP є однією з найпопулярніших мов програмування для створення веб-ресурсів. Вона широко підтримується більшістю веб-хостингових провайдерів, і часто включена у стандартні пакети хостингу, що робить її легко доступною для

запуску веб-проектів. Ця мова легко інтегрується з більшістю популярних систем управління базами даних, таких як MySQL, PostgreSQL та інші, і підтримує об'єктно-орієнтований, процедурний та функціональний стилі програмування, що забезпечує високу гнучкість при створенні веб-ресурсів. Як безкоштовна та відкрита мова програмування, PHP дозволяє її вільно використовувати, модифікувати та розповсюджувати. Крім того, існують численні потужні фреймворки на PHP, такі як Laravel.

Laravel, як сучасний PHP фреймворк, є відмінним вибором для створення системи управління навчанням (LMS) через свою масштабованість та гнучкість, що дозволяє легко масштабувати проект і підтримувати великі та складні веб-додатки. Його вбудовані можливості безпеки, такі як захист від SQL ін'єкцій, хрестового сайтового скриптування (XSS) та хрестового сайтового фальсифікації запитів (CSRF), є важливими для LMS, де обробляються конфіденційні дані студентів та викладачів. Laravel підтримує високий рівень модульності завдяки пакетам і бандлам, дозволяючи легко розширювати функціональність системи. Eloquent ORM спрощує роботу з базами даних, а система міграцій баз даних допомагає зберігати базу даних у синхронізації з кодовою базою. Laravel має велику та активну спільноту, численні пакети та ресурси, що можуть допомогти в розробці LMS. Його використання моделі-виду-контролера (MVC) архітектури допомагає у організації коду та робить його більш читабельним і легшим для управління, що є ключовим для великих проектів. Підтримка вбудованого тестування з PHPUnit та зручні засоби для проведення функціональних та юніт-тестів є важливими для забезпечення якості та надійності системи LMS. Консольний інтерфейс Artisan спрощує багато повсякденних завдань розробки та підтримки. Всі ці особливості роблять Laravel відмінним вибором для розробки масштабованих, безпечних та функціонально багатих систем управління навчанням.

База даних реалізована з використанням СУБД MySQL [14] і адмініструється з допомогою веб-інтерфейсу phpMyAdmin [15], який надає графічний доступ до адміністрування сервера MySQL, виконання SQL-запитів, а також перегляду та редагування вмісту таблиць баз даних.

MySQL є надзвичайно популярною системою управління базами даних у світі веб-розробки, особливо в поєднанні з PHP. Це означає, що існує велика кількість документації, підтримки спільноти та ресурсів для навчання, що робить її легкою для вивчення та використання. Також Laravel має вбудовану підтримку MySQL, що забезпечує плавну інтеграцію. Eloquent ORM, який є частиною Laravel, оптимізований для роботи з MySQL, дозволяючи розробникам ефективно виконувати запити до бази даних та керувати даними з використанням об'єктно-орієнтованого підходу. Це робить розробку додатків швидкою та інтуїтивно зрозумілою. MySQL відома своєю високою продуктивністю, надійністю та легкістю управління. Це робить її ідеальним вибором для розробки веб-додатків, які потребують швидкого доступу до даних та стабільності, особливо в середовищах з високим рівнем трафіку. Крім того, MySQL є безкоштовною та відкритою системою, що робить її доступною для великої аудиторії розробників. Її широка популярність та підтримка різноманітними хостинговими сервісами також є важливими факторами, що роблять MySQL логічним вибором для багатьох проектів на основі PHP та Laravel.

Для розробки клієнтської частини використовувались такі технології: HTML як мова розмітки документів для відображення веб-сторінок у браузері, CSS для опису стилів сторінок та JavaScript (JS) для створення сценаріїв веб-сторінок. JS дозволяє взаємодіяти з користувачем, керувати браузером, асинхронно взаємодіяти з сервером, а також змінювати структуру та вигляд веб-сторінки на боці клієнта (користувача).

Для реалізації штучного інтелекту обрано OpenAI [16]. OpenAI надає доступ до своїх інструментів та API, які можна використовувати для створення інтелектуальних застосунків та продуктів. OpenAI відмінно підходить як генератор контенту завдяки своїм передовим можливостям машинного навчання та обробки природної мови. Це забезпечує здатність генерувати тексти, які не просто граматично коректні, але й змістовні, релевантні та візуально привабливі. ШІ OpenAI може адаптуватися до різних стилів та форматів, що робить його універсальним інструментом для створення широкого спектру контенту - від блогів

та маркетингових матеріалів до технічних статей та творчих текстів. Крім того, OpenAI може швидко генерувати великі обсяги контенту, що економить час та ресурси. Його здатність аналізувати та опрацьовувати великі обсяги даних дозволяє створювати персоналізований та орієнтований контент, що є ключовим для залучення аудиторії та підвищення ефективності маркетингових зусиль. OpenAI також надає можливість генерувати ідеї та концепції, які можуть бути використані для інноваційних проєктів та креативних ініціатив. Все це робить OpenAI потужним інструментом для генерації контенту, який може задовольнити різноманітні потреби та вимоги.

2.9 Опис програмного забезпечення

PHP на сьогодні є однією з найбільш популярних та всебічних мов програмування, зосереджених на створенні веб-ресурсів. Це пояснюється рядом переваг, які вона пропонує розробникам. Код PHP не є громіздким і основні скрипти можна написати всього в кілька рядків, завдяки простому та зрозумілому синтаксису, який спирається на переваги високорівневої мови Perl та процедурної мови C.

PHP відомий своїми широкими можливостями, особливо у написанні серверних скриптів. Він здатний виконувати всі функції, які доступні для інших CGI-програм, включаючи обробку форм, створення динамічних сторінок, відправку та прийом cookies. PHP також здатен виконувати багато інших завдань.

Основна відмінна риса PHP - його практичність. Мова розроблена таким чином, щоб забезпечити програмістам інструменти для швидкого та ефективного вирішення різноманітних завдань. Laravel, створений у 2011 році, є одним з найпопулярніших фреймворків завдяки своїй гнучкості та можливостям. Цей фреймворк дозволяє розробляти навіть складні сайти та структури, ставши зручним інструментом для створення додатків.

Основні переваги Laravel включають:

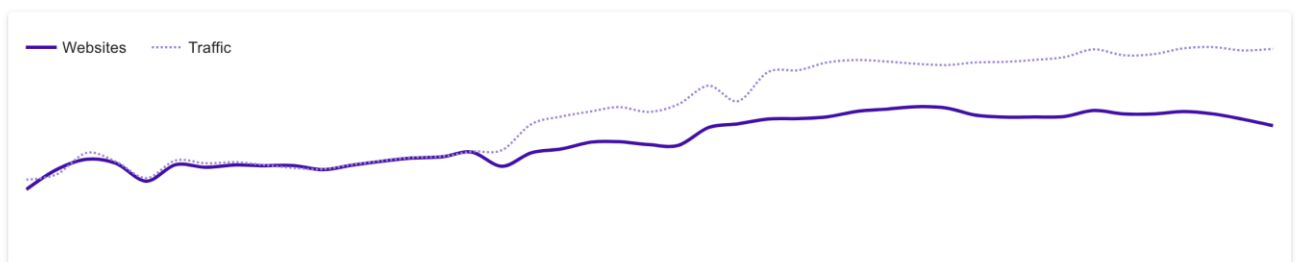
– високий рівень безпеки. Завдяки розвитку та покращенню фреймворку з часу його створення, Laravel забезпечує високий рівень безпеки. ORM (Object-Relational Mapping) в Laravel запобігає SQL-ін'єкціям, нормалізуючи параметри запитів. Міжсайтові скрипти також ефективно контролюються шляхом екранування html-тегів, перетворюючи потенційно небезпечний вміст у безпечний текст;

– використання шаблонів Blade. Blade - це система шаблонів Laravel, яка забезпечує гнучкість та ефективність при створенні унікальних макетів сайтів. З її допомогою розробники можуть легко стандартизувати процеси розробки, використовуючи один шаблон у різних частинах проекту.

Наразі існує більше 110 000 веб-сайтів, що використовують даний фреймворк [17].

Laravel usage trend

This graph shows the growth of Laravel since липень 2020 p..



Laravel demographics

A breakdown of countries and languages used by Laravel websites.

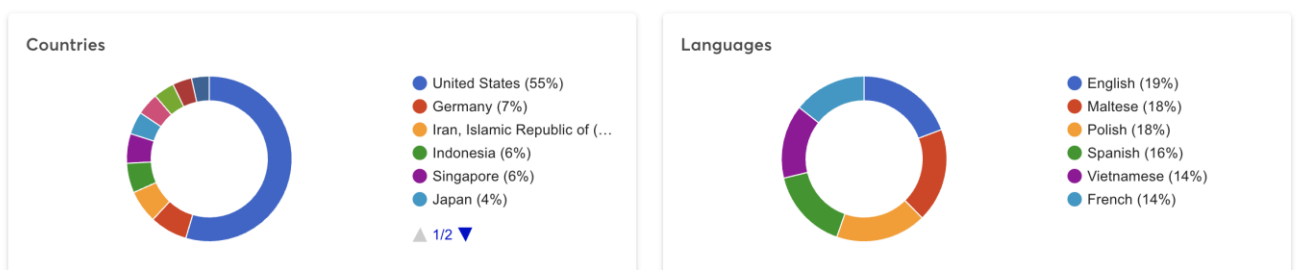


Рисунок 2.1 – Використання фреймворку з 2020 року

Laravel доступний у вільному доступі, що дозволяє легко знайти інформацію, рекомендації, посібники та коментарі від розробників. Ця прозорість і доступність робить Laravel зручним для використання як розробниками, так і клієнтами.

Що стосується безпеки, Laravel ефективно протидіє двом основним видам загроз: SQL-ін'єкціям та крос-сайт скриптингу. Захист від SQL-ін'єкцій забезпечується за допомогою ORM, який запобігає використанню "сирих" SQL-запитів. Проблема крос-сайт скриптингу вирішується за допомогою екранування небезпечних HTML-тегів. Для підвищення рівня безпеки Laravel також використовує додаткову валідацію даних, що вводяться в форми.

2.9.1 PHP – опенсорс проект

Відкрита доступність PHP робить його особливо привабливим для IT-менеджерів та розробників, що є однією з ключових переваг цієї мови. Вільна ліцензія PHP сприяє створенню великої та активної спільноти, яка постійно працює над удосконаленням і розширенням функціональності мови. Така спільнота забезпечує тисячі готових рішень, дозволяючи розробникам ефективно використовувати наявні інструменти без необхідності "винаходити велосипед".

Водночас, легкість і доступність PHP мають і зворотний бік. Низький поріг входження може призвести до створення неякісних продуктів, що шкодить репутації технології в цілому. Це вимагає відповідального підходу до вибору професійної команди для розробки проектів. Для IT-компаній виклик полягає у великій кількості неякісних спеціалістів на ринку, яких іноді важко відрізнити від кваліфікованих професіоналів.

2.9.2 Швидка розробка платформи

PHP відрізняється тим, що він частково об'єктно-орієнтований, що є однією з його ключових переваг. Об'єктно-орієнтоване програмування (ООП) [18] в PHP дозволяє повторно використовувати код, що значно економить час та ресурси під

час розробки. Ця можливість повторного використання коду підвищує ефективність та зменшує витрати на розробку.

У світі PHP існує велика кількість фреймворків, таких як Laravel, Yii2, Symfony [19], Zend, а також систем управління контентом (CMS), як Wordpress[20], Joomla, Drupal[21], OpenCart та інші. Кожен фреймворк чи CMS має свої унікальні особливості та призначений для різних цілей, що робить PHP гнучким інструментом для розробників.

PHP був спеціально оптимізований для швидкої розробки веб-додатків. Він містить вбудовані функції для роботи з POST і GET запитами, обробки URL та HTML. Це означає, що для клієнтів розробка проєктів на PHP може бути більш бюджетною і швидкою, забезпечуючи високу рентабельність інвестицій.

2.9.3 Гарна сумісність та гнучкість

Універсальність і гнучкість - одні з ключових переваг PHP. Ця мова програмування сумісна з основними операційними системами, такими як UNIX, Linux, Windows і MacOS. Також PHP підтримується більшістю веб-серверів, включаючи Apache, Netscape, Microsoft IIS, NGINX, iPlanet, Xitami, Caudium і Tornado, а також різними системами управління базами даних, такими як MySQL, PostgreSQL, MongoDB та інші. Ця взаємосумісність робить PHP ідеальним вибором для розробки крос-платформових застосунків, дозволяючи максимально ефективно використовувати існуючу інфраструктуру з оптимальними бюджетами.

Хоча PHP традиційно використовувався у тандемі з HTML, сучасні технології та архітектурні підходи, такі як REST і FULL REST [22], також розробляються на PHP. PHP легко інтегрується з іншими мовами програмування та технологіями, включаючи JavaScript, XML, WML тощо. Це дозволяє створювати складні і високофункціональні веб-додатки. Крім того, оскільки PHP-скрипти інтерпретуються на стороні сервера, проблеми з сумісністю браузерів мінімізуються. Це робить PHP відмінним вибором для розробки бекенду веб-додатків.

2.9.4 Сервіси з відмінною документацією та масштабованістю

Економічна ефективність для компаній часто досягається за рахунок розробки на основі ітерацій. Цей підхід дозволяє запускати та адаптувати різні програмні модулі послідовно, вносячи зміни та оновлення в процесі розвитку проекту. PHP особливо добре підходить для такого підходу, оскільки він підтримує розробку, розширення та підтримку масштабованих ІТ-проектів. Ця мова програмування дозволяє легко створювати і керувати великою кількістю сутностей з різними параметрами.

Однією з переваг PHP є його прозорість та відмінна документація. Якісно написаний PHP-код легко читати та розуміти, що знижує час, необхідний для вивчення та розуміння коду. Це важливо, оскільки відсутність якісного коду може бути проблемою, однак коли код написаний добре, він спрощує процес ітеративної розробки.

Завдяки простоті розуміння PHP-коду, можливе безпечне й ефективне впровадження ітеративного розвитку з мінімальними ризиками затримок, навіть у випадку зміни команди чи окремих розробників. Це робить PHP відмінним вибором для проектів, які вимагають гнучкості та ефективності у розробці.

2.9.5 Оновлення та підтримка

Підтримка та модифікація коду в PHP є ще однією важливою перевагою цієї мови програмування. Висока читабельність PHP-коду сприяє легкості його модифікації та зміни, що в свою чергу зводить до мінімуму час та зусилля, необхідні для оновлення та виправлення помилок. Ця особливість робить розроблені на PHP проекти гнучкими та швидко адаптованими до нових ринкових вимог, бізнес-завдань та викликів.

Ще один важливий аспект підтримки PHP полягає у внеску відкритої спільноти. Регулярні оновлення, нові функції та можливості, які надаються

спільнотою, забезпечують постійне вдосконалення PHP без додаткових витрат. Це створює активне середовище для інновацій та сприяє розвитку мови.

Крім того, простота коду PHP гарантує, що його підтримку може забезпечити будь-яка компетентна команда програмістів, навіть якщо вона не була залучена до первісної розробки проекту. Це значно спрощує процес управління та оновлення програмних продуктів, створених на PHP, забезпечуючи високий рівень гнучкості та масштабування для бізнесу.

2.9.6 Можливості вибору хостингу чи сервера

Універсальність PHP значною мірою впливає на його популярність серед хостингових та серверних провайдерів. Більшість хостингових компаній надають підтримку PHP як стандартну частину своїх послуг. Це означає, що клієнти можуть легко розміщувати свої PHP-базовані веб-сайти на цих серверах без необхідності встановлювати або налаштовувати додаткові модулі. Однією з переваг таких хостингових рішень є можливість вибору версії PHP. Це дозволяє користувачам налаштувати своє середовище розробки та вирішувати, яку версію PHP використовувати, в залежності від конкретних вимог їхнього проекту. Така гнучкість є важливою для розробників, які хочуть мати останні оновлення та виправлення безпеки або, навпаки, потребують стабільної, перевіреної версії PHP для підтримки існуючих додатків.

2.9.7 Стабільна робота онлайн-сервісів

Швидкість завантаження сайту в сучасному цифровому світі є критично важливою. З низькою швидкістю завантаження зростає ризик збоїв і зменшується задоволеність користувачів. Середній інтернет-користувач має терпіння чекати завантаження веб-сторінки лише 5-7 секунд. Якщо сайт завантажується повільно і змушує користувача чекати довше, висока ймовірність, що вони залишать сайт. Повернення відвідувача на повільно завантажуваний сайт можливе лише у

випадку, якщо він пропонує унікальний контент або унікальні послуги, що є рідкістю. Ефективна система керування контентом (CMS) або добре оптимізований веб-проект можуть значно покращити швидкість завантаження. Важливо створити веб-сайт, який не тільки привабливий і функціональний, але й оптимізований для швидкого завантаження. Це включає оптимізацію зображень, ефективно використання кешування, мінімізацію HTTP-запитів та використання сучасних веб-технологій для покращення продуктивності.

2.9.8 Валідація введення

Валідація введення є ключовим елементом у захисті вебзастосунок. Цей процес полягає у перевірці даних, введених користувачем у форми, на предмет їх правильності, безпечності та відповідності очікуваним форматам. Основна ідея полягає у тому, що всі дані, введені користувачем, спочатку вважаються потенційно хибними або небезпечними.

Процес валідації даних включає кілька кроків:

- перевірка формату даних: забезпечує, що введені дані відповідають визначеному формату (наприклад, електронна пошта має містити символ '@' та доменне ім'я);
- обмеження типу даних: визначає, що дані відповідають очікуваному типу (наприклад, числові поля не повинні приймати літери);
- перевірка на максимальну і мінімальну довжину: забезпечує, що введені дані мають припустиму довжину;
- фільтрація спеціальних символів та SQL-коду: запобігає внедренню шкідливого коду, такого як SQL-ін'єкції або скрипти.

2.9.9 Екранування виведення

Екранування виведення — це важливий процес в розробці вебзастосунків, який захищає від потенційних вразливостей, пов'язаних з внедренням шкідливого

коду. Він полягає у перетворенні певних спеціальних символів у безпечну форму, що запобігає їхньому некоректному інтерпретуванню веб-переглядачами або іншими системами.

У контексті HTML екранування включає перетворення символів, які можуть бути використані для впровадження HTML або JavaScript коду, наприклад, '<' перетворюється на '<', а '>' на '>'. Це запобігає можливості виконання шкідливого коду в браузері користувача (наприклад, у випадку міжсайтового скриптингу).

У контексті JavaScript або SQL екранування також важливе, але використовуються інші правила та методи для запобігання впровадженню скриптів або SQL-команд. Наприклад, в SQL може знадобитися екранувати апострофи, щоб запобігти SQL-ін'єкціям.

2.9.10 Запобігання sql-ін'єкцій

SQL-ін'єкція є однією з найпоширеніших і найнебезпечніших загроз безпеці веб-додатків. Вона виникає, коли зломисник вводить зловмисний SQL-код в інтерфейси вводу додатку, який потім виконується на сервері бази даних. Це може дозволити зломиснику читати, змінювати, видаляти дані або навіть виконувати адміністративні операції на базі даних.

Laravel використовує конструктор запитів, який застосовує прив'язки PDO (PHP Data Objects) для запобігання SQL-ін'єкціям. Прив'язки PDO забезпечують, що введені користувачем дані обробляються безпечним чином перед включенням їх у SQL-запит. При використанні конструктора запитів Laravel не потрібно екранувати рядки вручну, оскільки фреймворк автоматично здійснює безпечну обробку даних.

2.9.11 Запобігання XSS

Міжсайтовий скриптинг (XSS) є серйозною загрозою для безпеки веб-додатків. В атаках XSS, зломисники використовують вразливості веб-сайту для

вбудовування або виконання шкідливих скриптів на сторінках, що може призвести до крадіжки даних користувачів або інших шкідливих дій.

На рисунку нижче зображено приклад URL з GET-параметром, що яскраво ілюструє, як може бути здійснена проста атака XSS.

A dark gray rectangular box containing the URL: `b-a-i.online/public/?u=alert('Some%20strange%20words!')`

Рисунок 2.2 - приклад URL з GET-параметром

Зловмисник може вбудовувати шкідливий код у веб-сайт, модифікуючи параметр URL таким чином, щоб виконувати скрипти у браузері користувача.

Laravel, використовуючи Blade як свій шаблонізатор PHP, забезпечує деякий рівень захисту від XSS завдяки автоматичному екрануванню виведення. Синтаксис `{{ }}` в Blade автоматично екранує виведення, перетворюючи спеціальні символи на їхні HTML-еквіваленти, що запобігає виконанню шкідливого коду у браузері.

Проте, захист від XSS в Laravel не обмежується лише екрануванням. Ось кілька додаткових рекомендацій:

- очищення даних користувача. Використовуйте функції очищення даних для видалення небажаних символів або кодів, особливо в тих випадках, коли використовується виведення даних без екранування;
- перевірка введених користувачем даних. Створіть список очікуваних значень і переконайтеся, що дані, передані користувачами, відповідають цим очікуванням. Це особливо важливо для даних, що використовуються в URL-параметрах або в інших місцях, де можуть бути використані для внедрення скриптів;
- використання додаткових бібліотек безпеки. Хоча Laravel забезпечує базовий рівень захисту, розгляньте можливість використання спеціалізованих бібліотек і інструментів безпеки для подальшого зміцнення захисту від XSS.

Завжди важливо пам'ятати, що безпека веб-додатку є багат шаровою проблемою, яка вимагає уваги на різних рівнях стеку технологій та в різних аспектах розробки.

2.9.12 Запобігання CSRF

Підробка міжсайтського запиту (CSRF) дійсно є серйозною загрозою для веб-додатків. Вона дозволяє зловмисникам виконувати несанкціоновані дії, використовуючи права аутентифікованих користувачів. Laravel надає ефективний механізм для захисту від таких атак.

Якщо Laravel-додаток має маршрут, наприклад `/user/email`, для зміни адреси електронної пошти користувача, без належного захисту CSRF, зловмисний веб-сайт може створити форму, яка відправляє запит на цей маршрут без відома користувача.

Laravel ефективно захищає від CSRF-атак, використовуючи токени CSRF. Ось як це працює:

- генерація CSRF-токену. Коли користувачська сесія активується, Laravel автоматично генерує унікальний CSRF-токен для цієї сесії. Цей токен використовується для перевірки того, що саме аутентифікований користувач подає запит до додатку;
- включення токена у форми. Коли створюється HTML-форма для відправки запитів методами "POST", "PUT", "DELETE" тощо, розробник повинен включити поле `_token` у форму. Laravel надає зручну директиву Blade `@csrf`, яка автоматично додає приховане поле з CSRF-токеном у форму;
- перевірка CSRF-токена. Коли форма відправляється, Laravel перевіряє CSRF-токен, включений у запит. Якщо токен не співпадає з токеном у сесії користувача, запит відхиляється. Це запобігає можливості зловмисникам створювати форми, які взаємодіють з додатком без згоди користувача.

Отже, використання CSRF-токенів у Laravel є важливим компонентом захисту веб-додатку від CSRF-атак. Воно допомагає гарантувати, що лише легітимні запити, згенеровані самим додатком, обробляються сервером.

3 РЕАЛІЗАЦІЯ ПОСТАВЛЕНОЇ ЗАДАЧІ ДОСЛІДЖЕННЯ

Для реалізації механіки відправлення запиту на OpenAI API для генерації тексту та відображення відповіді у Laravel, необхідно створити контролер, який буде обробляти запити, та відповідний роут. Нижче наведено базовий приклад того, як це можна зробити.

Контролер у контексті веб-розробки, зокрема у фреймворках, що використовують архітектурний патерн Model-View-Controller (MVC), є центральною частиною, яка управляє взаємодією між користувачем, моделлю даних та інтерфейсом відображення (View). В MVC архітектурі, контролер відповідає за наступне:

- обробка вхідних запитів. Контролери приймають вхідні HTTP запити від користувачів, аналізують їх параметри, заголовки, тіло запиту тощо;
- прийняття рішень щодо обробки даних. На основі отриманого запиту, контролер визначає, які дії потрібно виконати. Наприклад, це може бути вилучення даних з бази даних, внесення змін у базу даних, виконання бізнес-логіки тощо;
- взаємодія з моделями. Контролери спілкуються з моделями для отримання, зміни або видалення даних. Модель у MVC відповідає за управління даними програми, бізнес-логіку та правила бізнесу;
- відправлення відповідей. Після обробки запиту та виконання необхідних операцій з даними, контролер відправляє відповідь назад до користувача. Це може бути HTML сторінка, JSON відповідь, редирект на іншу сторінку тощо;
- взаємодія з відображеннями (Views). У традиційному MVC, контролери також відповідають за вибір відповідних відображень (views) та передачу даних в них. Відображення - це те, що користувач бачить (наприклад, веб-сторінка).

Важливість контролерів:

- розділення відповідальності. Контролери допомагають розділити обробку даних і бізнес-логіку від користувацького інтерфейсу, що сприяє більш організованому та управлінному коду;
- легше тестування. Чітке розділення логіки в контролерах спрощує написання модульних тестів;
- гнучкість та масштабування. Зміни в бізнес-логіці чи інтерфейсі можуть бути впроваджені без великих змін у інших частинах програми.

У контексті Laravel, контролери є фундаментальним елементом, який дозволяє ефективно управляти запитам та відповідями в межах веб-додатку.

Перш за все потрібно відкрити файл контролера `OpenApiController`, який міститься у директорії `app/Http/Controllers/` проекту. У файлі контролера вказано необхідний простір імен та імпортовані залежності:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Http;
```

Рисунок 3.1 - Необхідний простір імен та імпортовані залежності

Розглянемо імпортовані до контролера залежності.

`Illuminate\Http\Request` — це клас у Laravel, який представляє HTTP запит, що надходить до системи. Він надає зручний інтерфейс для взаємодії з даними запиту. Ось деякі його важливі можливості:

- отримання Даних Запиту: Ви можете отримати доступ до даних запиту, таких як параметри GET або POST, cookies, заголовки тощо;
- перевірка Даних: Laravel надає засоби для валідації вхідних даних запиту;

– безпечний Доступ до Даних: Клас Request допомагає захистити систему від загроз, таких як SQL ін'єкції, шляхом санітації вхідних даних.

Цей клас є важливим для обробки вхідних даних у будь-якому запиті до Laravel додатку.

Illuminate\Support\Facades\Http — це фасад, який надає зручний інтерфейс для відправлення HTTP запитів з Laravel додатку. Він базується на популярній бібліотеці Guzzle і вбудований у Laravel, починаючи з версії 7.x. Деякі особливості Http фасаду включають:

- простота в відправленні запитів: Дозволяє легко відправляти GET, POST, PUT, DELETE та інші типи HTTP запитів;
- ланцюжковий синтаксис: Підтримує ланцюжковий синтаксис, що робить код чистішим та більш зрозумілим;
- автоматичне керування JSON: Автоматично перетворює відповіді JSON у масиви / об'єкти PHP;
- обробка Відповідей: Надає методи для перевірки статусу відповідей, отримання вмісту відповідей та іншого.

Цей фасад є надзвичайно корисним для взаємодії з зовнішніми HTTP сервісами та API.

У класі OpenAIController додано метод generateText. Цей метод буде приймати запит Request як аргумент:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\Http;

class OpenAIController extends Controller
{

    public function generateText(Request $request){...}
}
```

Рисунок 3.2 – Створений метод у класі OpenAIController

У середині методу `generateText` отримуємо API ключ OpenAI з `.env` файлу, у який заздалегідь було додано ключ.

Отримання API ключа від OpenAI є важливим кроком для тих, хто бажає інтегрувати свої програми з передовими можливостями штучного інтелекту, які пропонує ця платформа. Цей процес починається з відвідування офіційного вебсайту OpenAI та створення акаунту, що є стандартною процедурою для більшості онлайн-сервісів. Після реєстрації, необхідно пройти процес верифікації через електронну пошту, щоб забезпечити додатковий рівень безпеки.

Ключовим моментом є створення API ключа, який здійснюється в кабінеті користувача на сайті OpenAI. Цей ключ є унікальним ідентифікатором, який дозволяє запитам від програми бути автентифікованими і безпечно оброблятися серверами OpenAI. Під час створення ключа користувачам надається можливість вказати параметри доступу, які визначають обсяг можливостей, які будуть доступні через цей ключ.

Після отримання ключа, його необхідно ретельно зберігати. Зазвичай, для захисту конфіденційності та безпеки, ключ додається до файлу `.env` у проектах, що базуються на таких фреймворках, як Laravel. Це гарантує, що чутливі дані не потрапляють до систем контролю версій і не розповсюджуються публічно. Файл

.env (від англ. environment - середовище) використовується в багатьох програмних проєктах для зберігання конфігураційних параметрів, які можуть змінюватися в залежності від середовища, в якому запущено програму. Цей файл дозволяє визначати змінні середовища, які можна легко читати та змінювати, не вносячи змін у код додатку. Зазвичай .env файл містить змінні у вигляді KEY=VALUE, де KEY — це назва змінної, а VALUE — її значення.

```
$apiKey = env( key: 'OPENAI_API_KEY');
```

Рисунок 3.3. - присвоєння API ключа OpenAI з .env файлу

Перед тим як відправити HTTP запит до OpenAI потрібно визначити кілька ключових елементів:

- API ключ: Для автентифікації у OpenAI API потрібен API ключ. Цей ключ зазвичай зберігається в файлі .env проєкту Laravel для безпеки. Ви можете отримати його так: `env('OPENAI_API_KEY');`
- endpoint URL: URL адреса OpenAI API, до якої буде відправлено запит. Наприклад, для моделі GPT-3 це може бути `https://api.openai.com/v1/engines/davinci-codex/completions;`
- дані для запиту: Параметри запиту, які містять інформацію, необхідну для генерації тексту. Наприклад, це може бути текст запиту `prompt`, який ви хочете використовувати для генерації відповіді, та `max_tokens`, що визначає максимальну кількість токенів у відповіді.

Використовуючи фасад Http Laravel, створили та відправили HTTP запит до OpenAI.

```

$prompt = $request->input( key: 'prompt', default: 'Hi!');

$response = Http::withHeaders([
    "Content-Type" => "application/json",
    'Authorization' => 'Bearer '.$apiKey
])->post( url: 'https://api.openai.com/v1/chat/completions', [
    "model"           => "gpt-4-vision-preview",
    "temperature"    => 1,
    "top_p"           => 1,
    "n"               => 1,
    "stream"          => false,
    "max_tokens"     => 250,
    "presence_penalty" => 0,
    "frequency_penalty" => 0,
    "messages"       => [
        [
            "role"     => "user",
            "content"  => $prompt
        ]
    ],
]);

```

Рисунок 3.4 - відправлення HTTP запиту до OpenAI

На рисунку вище :

- `Http::withHeaders([])` - встановлює заголовки запиту. У цьому випадку додаємо заголовок `Authorization` з API ключем для аутентифікації;
- `->post($url, [])` - відправляє POST запит на вказаний URL з переданими даними. `$url` - це адреса API OpenAI, а другий аргумент - це масив даних, які включають `prompt` та `max_tokens`.

Після відправлення запиту потрібно обробити відповідь від OpenAI.

```

if ($response->successful()) {
    $generatedText = $response->json()['choices'][0]['message']['content'];
} else {
    $generatedText = response()->json(['error' => 'Answer error.'], status: 500);
}

return view( view: 'opanaï.text_generator', ['generatedText' => $generatedText]);

```

Рисунок 3.5 – Обробка відповіді від OpenAI

- `successful()` перевіряє, чи був запит успішним;
- `json()` перетворює відповідь JSON від OpenAI у масив;
- `['choices'][0]['message']['content']` отримує текст згенерований моделлю

OpenAI.

У сучасному веб-розробленні, створення роуту (маршрутизація) є фундаментальною концепцією, яка відіграє ключову роль у визначенні того, як веб-запити керуються всередині додатку. У контексті веб-фреймворків, таких як Laravel, цей процес набуває особливої важливості через його здатність елегантно та ефективно управляти взаємодією між користувацьким запитом і серверною логікою.

Коли мова йде про Laravel, маршрутизація здійснюється через визначення роутів у файлі маршрутів, який є частиною архітектури додатку. Цей файл діє як мапа або навігаційна система, вказуючи, які контролери та методи повинні бути викликані у відповідь на певні HTTP запити. Завдяки цьому, роутинг у Laravel не лише спрощує веб-розробку, але й додає додатковий рівень абстракції та організації.

Процес створення роуту в Laravel починається з визначення шляху (URI) та відповідного методу контролера, який має бути викликаний. Це робиться шляхом додавання відповідного запису у файл `web.php`, розташований у директорії `routes``. Такий запис вказує фреймворку, як обробляти певний шлях чи запит, забезпечуючи відповідне спрямування до визначеної логіки бізнес-процесів.

Важливою перевагою такого підходу є його гнучкість. Розробники можуть легко впроваджувати різноманітні стратегії обробки запитів, включаючи передачу параметрів, забезпечення захисту маршрутів через мідлвари (middleware), групування маршрутів для спільних атрибутів, а також створення RESTful маршрутів для роботи з API.

```
use App\Http\Controllers\OpenAIController;

Route::post(uri: '/generate-text', [OpenAIController::class, 'generateText']);
```

Рисунок 3.6 – Додавання роуту у файл web.php

Для більш зручної роботи з OpenAI API доцільно перенести логіку контролера на сторону JS. Таким чином користувач матиме змогу отримати потрібну інформацію без перезавантаження сторінки.

Так як користувач працюватиме зі сторінками створення та редагування постів у шаблонах сторінок додано input з кнопкою відправлення та область виведення відповіді.

```
<h2>Text Generator with OpenAI</h2>
<label for="prompt">Enter your prompt:</label><br>
<textarea id="prompt" name="prompt" rows="4" cols="50"></textarea><br>
<button type="submit" class="openai-send-btn">Generate Text</button>
<div class="response">
  <h2>Generated Text:</h2>
  <p class="openai-response"></p>
</div>
```

Рисунок 3.7 – HTML структура області взаємодії з OpenAI

– `<h2>Text Generator with OpenAI</h2>` це HTML-елемент заголовка другого рівня (`<h2>`), який використовується для відображення основного заголовка інтерфейсу. Текст "Text Generator with OpenAI" чітко інформує користувачів про призначення цього інтерфейсу;

– `<label for="prompt">Enter your prompt:</label>` - це елемент `<label>`, що використовується для опису поля введення (у цьому випадку, текстового поля). Атрибут `for` зі значенням "prompt" вказує, що цей ярлик пов'язаний з текстовим полем, ідентифікатор якого є "prompt";

– `<textarea id="prompt" name="prompt" rows="4" cols="50"></textarea>` - елемент `<textarea>` використовується для створення багаторядкового поля

введення, куди користувачі можуть вводити текст. Атрибути `id` та `name` зі значенням `"prompt"` встановлюють унікальний ідентифікатор та ім'я поля введення відповідно. Атрибути `rows` та `cols` визначають розмір поля введення в рядках та стовпцях;

– `<button type="submit" class="openai-send-btn">Generate Text</button>` - це елемент `<button>`, що використовується для створення кнопки, яка ініціює дію (у цьому випадку, генерацію тексту). Атрибут `type="submit"` вказує, що кнопка використовується для відправлення форми. Атрибут `class` зі значенням `"openai-send-btn"` може бути використаний для стилізації кнопки за допомогою CSS;

– `<div class="response">`: Елемент `<div>` з класом `"response"` використовується як контейнер для відображення згенерованого тексту. Це блоковий елемент, що може містити інші HTML-елементи;

– `<h2>Generated Text:</h2>`: Це другий елемент заголовка другого рівня, розташований всередині `<div class="response">`. Він вказує, що текст, який буде виведений нижче, є результатом генерації;

– `<p class="openai-response"></p>`: Елемент `<p>` з класом `"openai-response"` використовується для відображення тексту, згенерованого OpenAI. Початково він пустий, але очікується, що його зміст буде оновлюватися за результатами генерації тексту.

Ця структура є інтуїтивно зрозумілою та функціональною, що робить її зручною для користувача.

Результат розробленої структури зображено на рисунку 4.9.

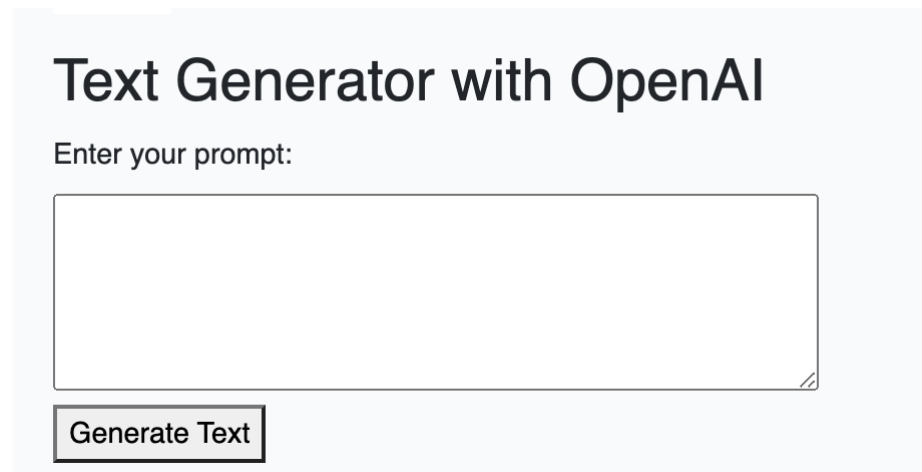


Рисунок 3.8 – Область для вводу запиту до ШІ

Розробка запитів на API з використанням JavaScript [23] стала невід'ємною частиною створення інтерактивних та динамічних вебзастосунків. Цей процес передбачає взаємодію веб-клієнта з сервером через веб-API, де JavaScript виконує роль посередника, ініціюючи запити та обробляючи відповіді.

Ключовим моментом в цій інтеракції є використання AJAX (Asynchronous JavaScript and XML), технології, яка дозволяє вебсторінкам асинхронно обмінюватися даними з сервером без необхідності перезавантаження сторінки. Це означає, що користувачі можуть продовжувати взаємодіяти зі сторінкою, поки дані відправляються або отримуються з сервера.

Використання об'єкта XMLHttpRequest або сучаснішого API fetch у JavaScript є основними способами реалізації AJAX запитів. XMLHttpRequest був довгий час стандартом для виконання HTTP запитів в JavaScript, але fetch пропонує більш сучасний, гнучкий та зручний у використанні інтерфейс.

Запит до API зазвичай має наступні етапи:

- формування запиту - ініціювання запиту до API з JavaScript, де вказуються URL API, метод запиту (наприклад, GET, POST), заголовки, параметри та тіло запиту в разі необхідності;
- відправлення запиту - використання fetch або XMLHttpRequest для асинхронної відправки запиту та очікування відповіді. Це включає в себе обробку обіцянок (promises) у випадку використання fetch;

- обробка відповіді - після отримання відповіді від API, вона обробляється для вилучення корисних даних, які можуть бути представлені у форматі JSON, XML або іншому;

- оновлення інтерфейсу користувача - використання отриманих даних для оновлення веб-сторінки. Це може включати додавання нового вмісту, оновлення елементів інтерфейсу або відображення повідомлень користувачу.

Передача ключів або чутливих даних з .env файлу в JavaScript на стороні клієнта вимагає обережності, оскільки це може створити серйозні ризики безпеки. Чутливі дані, такі як API ключі, не повинні бути відкрито доступними у клієнтському JavaScript, оскільки це може призвести до їх витоку та зловживань.

Найбезпечніший спосіб - не передавати чутливі дані на клієнтську сторону взагалі. Натомість, можна створити проксі-ендпоінт на сервері, який буде використовувати ключ для виконання запитів до зовнішнього API.

- клієнтський JavaScript відправляє запит на проксі-ендпоінт;
- сервер отримує цей запит, додає API ключ із .env файлу, та виконує запит до зовнішнього API;
- відповідь від зовнішнього API повертається через сервер назад до клієнта.

Цей підхід не тільки забезпечує безпеку ключа, але й дає більший контроль над запитом, які виконуються з системи.

```

if ($('#openai-send-btn')) {
  $('#openai-send-btn').click(function () : void {
    if ($('#prompt').val().length < 5) {
      return
    }

    let key : string = '';
    $.ajax({
      "url": "/generate-text",
      "method": "GET",
    }).done(function (response) : void {
      key = JSON.parse(response)
    });

    $.ajax({
      "url": "https://api.openai.com/v1/chat/completions",
      "method": "POST",
      "timeout": 0,
      "headers": {
        "Content-Type": "application/json",
        "Accept": "application/json",
        "Authorization": "Bearer ".key.api_key
      },
      "data": JSON.stringify( value: {
        "model": "gpt-4-vision-preview",
        "messages": [
          {
            "role": "user",
            "content": $('#prompt').val()
          }
        ],
        "temperature": 1,
        "top_p": 1,
        "n": 1,
        "stream": false,
        "max_tokens": 700,
        "presence_penalty": 0,
        "frequency_penalty": 0
      }),
    }).done(function (response) : void {
      $('#openai-response').text(response.choices[0].message.content)
    });
  })
}

```

Рисунок 3.9 – Реалізація логіки на JS

Після отримання відповіді з API повідомлення було виведене на екран.

Text Generator with OpenAI

Enter your prompt:

How to get master degree

Generate Text

Generated Text:

To get a master's degree, you usually need to follow these steps: 1. Complete a Bachelor's Degree: Before you can pursue a master's degree, you typically need to have completed an undergraduate degree (bachelor's degree) in a related field from an accredited institution. 2. Research Programs and Universities: Look for universities that offer the master's program you are interested in. Consider factors such as location, program length, cost, curriculum, and the reputation of the faculty. 3. Meet Admission Requirements: Check the specific admission requirements for the programs you are interested in. This may include a certain GPA, recommendation letters, work experience, a statement of purpose, and standardized test scores like the GRE or GMAT. 4. Submit Applications: Apply to the programs you have chosen before the application deadlines. Be sure to include all required documents and information. 5. Secure Funding: Master's degree programs can be expensive, so it's important to consider how you will pay for your education. Look into scholarships, grants, financial aid, assistantships, and student loans. 6. Enroll in a Program: Once accepted, enroll in the master's program and begin taking courses as outlined by the program curriculum. 7. Complete Coursework and Thesis/Project: Attend classes, complete coursework, and if required, conduct research for a thesis or capstone project. Some programs may also require internships or practicum experiences. 8. Meet Graduation Requirements: Fulfill all academic requirements, which may include maintaining a certain GPA, passing comprehensive exams, and successfully defending your thesis or project. 9. Obtain Your Degree: After meeting all requirements and completing all necessary coursework, you will be awarded your master's degree during a graduation ceremony. Remember to always check specific program requirements as they may vary from one institution to another.

Рисунок 3.10 – Результат виконання запиту

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було досліджено варіанти інтеграції штучного інтелекту у систему онлайн навчання та вже існуючі варіанти реалізації.

Під час виконання роботи були проведено аналіз предметної області, розглянуто сучасний стан технологій, проаналізовані вже існуючі системи з вбудованим штучним інтелектом, розглянуті їх методи інтеграції. Були встановлені основні функціональні вимоги до системи. Були розглянуті як переваги, так і недоліки інтегрування штучного інтелекту в систему онлайн навчання як для студента, так і для викладача.

Розглянуто можливі варіанти інтеграції ШІ до системи онлайн навчання, проведено аналіз вже існуючих систем. Встановлено, що розробники даних систем інтегрують штучний інтелект у свій продукт для поліпшення навчання студента, нехтуючи питанням покращення роботи викладача. В результаті дослідження можна зробити висновок, що ідеальним варіантом інтеграції штучного інтелекту є генератор контенту для системи онлайн навчання. Це дозволить зменшити навантаження на викладачів та прискорити процес створення статей.

Були виділені технології подальшої розробки системи. З допомогою даної інтеграції процес створення навчального контенту значно пришвидшиться, що приведе систему до регулярної активності та актуальності.

Результати дослідження доповідались на XVI міжнародній науково-практичній конференції «Інформаційні технології і автоматизація – 2023» [24].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Collecchia G., Gobbi R. AI in Clinical Practice: A Guide to Artificial Intelligence and Digital Medicine. Amsterdam, 2024. 200 p.
2. Офіційний сайт ukrinform.ua [Електронний ресурс] – Режим доступу: <https://www.ukrinform.ua/rubric-society/3781918-povnomasstabna-vijna-zbilsila-popit-na-onlajnkursi-ekspert.html>.
3. Офіційний сайт duolingo.com [Електронний ресурс] – Режим доступу: <https://www.duolingo.com/>.
4. Офіційний сайт photomath.com [Електронний ресурс] – Режим доступу: <https://photomath.com/en>.
5. Wilks Y. Artificial Intelligence: Modern Magic or Dangerous Future? London, 2019. 176 p.
6. Офіційний сайт uportal.com [Електронний ресурс] – Режим доступу: <https://fakty.com.ua/ua/ukraine/suspilstvo/20231220-shtuchnyj-intelekt-v-osviti-yak-tehnologiya-vplyvaye-na-navchannya-v-ukrayinskyh-shkolah/>.
7. Офіційний сайт Coursera.org [Електронний ресурс] – Режим доступу: <https://www.coursera.org/>.
8. Офіційний сайт KhanAcademy.org [Електронний ресурс] – Режим доступу: <https://www.khanacademy.org/>.
9. Офіційний сайт edX.org [Електронний ресурс] – Режим доступу: <https://www.edx.org/>.
10. Офіційний сайт Udacity.com [Електронний ресурс] – Режим доступу: <https://www.udacity.com/>.
11. Офіційний сайт lb.ua [Електронний ресурс] – Режим доступу: https://lb.ua/blog/olena_vyshniakova/547626_ai_osvita_yak_shtuchniy_intelekt.html.
12. Mann E. PHP Cookbook: Modern Code Solutions for Professional Developers 1st Edition. California, 2023. 432p.
13. Stauffer M. Laravel: Up and Running: A Framework for Building Modern PHP Apps 1st Edition. California, 2016. 456 p.

14. Reese G. MySQL Pocket Reference: SQL Functions and Utilities. California, 2007. 134 p.
15. Delisle M. phpMyAdmin Starter. Birmingham, 2012. 42 p.
16. Офіційний сайт документації OpenAI [Електронний ресурс] – Режим доступу: <https://platform.openai.com/docs/introduction>.
17. Офіційний сайт wappalyzer.com [Електронний ресурс] – Режим доступу: <https://www.wappalyzer.com/technologies/web-frameworks/laravel/>.
18. Lavin P. Object-Oriented PHP: Concepts, Techniques, and Code. San Francisco, 2006. 216 p.
19. Potencier F. Symfony 5: The Fast Track. Clichy, 2019. 324 p.
20. Messenlehner B., Coleman J. Building Web Apps with WordPress: WordPress as an Application Framework 2nd Edition. California, 2019. 532 p.
21. Byron A., Berry A., De Bondt B. Using Drupal: Choosing and Configuring Modules to Build Dynamic Websites. California, 2012. 496 p.
22. Richardson L. RESTful Web APIs: Services for a Changing World. California, 2013. 406 p.
23. Flanagan D. JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language. California, 2020. 706 p.
24. Бродягіна Д.А., Калита Н.І. Інтеграція штучного інтелекту у систему онлайн навчання// Інформаційні технології і автоматизація – 2023 / Матеріали XVI міжнародної науково-практичної конференції. Одеса, 19-20 жовтня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – с. 162-163.