


**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА ІМЕНІ О. М. БЕКЕТОВА**


**Пояснювальна записка
до кваліфікаційної роботи бакалавра**

на тему: «Розробка вебдодатку для генерації текстового та візуального контенту на основі ШІ»

Виконав: студентка 4 курсу, групи КН 2021-1
спеціальності
122 Комп'ютерні науки
(шифр і назва спеціальності)

Тетяна САНЖАРЕВСЬКА 
(ім'я та прізвище)

Керівник: проф. Наталія СІЗОВА 
(ім'я та прізвище)

Рецензент: к.т.н. Микола КАРПЕНКО 
(ім'я та прізвище)

м. Харків – 2025р.

Харківський національний університет міського господарства імені О. М. Бекетова
(повне найменування закладу вищої освіти)

Навчально-науковий Інститут енергетичної, інформаційної
та транспортної інфраструктури

Кафедра комп'ютерних наук та інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 Комп'ютерні науки

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КНтаІТ



Марина

НОВОЖИЛОВА

« 24 » 06 2025 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Санжаревської Тетяни Сергіївни

(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка вебдодатку для генерації текстового та візуального контенту на основі ШІ»

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

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

затверджені наказом закладу вищої освіти від « 09 » травня 2025 р. № 341-03

2. Термін подання студентом роботи 20.06.2025

3. Вихідні дані до роботи ШТУЧНИЙ ІНТЕЛЕКТ, ГЕНЕРАЦІЯ КОНТЕНТУ, ВЕБДОДАТОК, GPT-4, DALL E? МОБІЛЬНИЙ ІНТЕРФЕЙС, АВТОМАТИЗАЦІЯ, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати предметну область та наявні аналоги; надати аргументоване обґрунтування вибору комплексу інструментального середовища та технічної платформи; визначити ключовий функціонал системи генерації текстового та візуального контенту; провести його системний аналіз; розробити програмний код основних модулів вебдодатку з інтеграцією API GPT-4 і DALL·E; реалізувати адаптивний інтерфейс користувача з підтримкою мобільних пристроїв за допомогою технології Progressive Web App (PWA); здійснити комплексне тестування функціональності, стабільності та зручності використання програмного забезпечення.

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

6. Консультанти розділів роботи

Розділ	Ім'я та Прізвище, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ I	 Наталія СІЗОВА	11.05.2025	10.05.2025
Розділ II	 Наталія СІЗОВА	17.05.2025	15.05.2025
Розділ III	 Наталія СІЗОВА	21.05.2025	30.05.2025
Розділ IV	 Вікторія Малишева	27.05.2025	15.06.2025

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вибір теми дипломної роботи	03.05.2025	Викон.
2	Затвердження тем, наукових керівників, завдань та календарного плану підготовки дипломної роботи	05.05.2025	Викон.
3	Написання I розділу	10.05.2025	Викон.
4	Написання II розділу	15.05.2025	Викон.
5	Написання III розділу	20.05.2025	Викон.
6	Написання IV розділу	30.05.2025	Викон.
7	Подання дипломної роботи керівнику	05.06.2025	Викон.
8	Робота по усуненню зауважень керівника, уточнення і доповнення практичного матеріалу, оформлення додатків до роботи	10.06.2025	Викон.
9	Подання доопрацьованого варіанту роботи керівнику	15.06.2025	Викон.
10	Захист матеріалів дипломної роботи на засіданні кафедри	18.06.2025	Викон.
11	Офіційний захист матеріалів дипломної роботи на засіданні екзаменаційної комісії	22.06.2025	Викон.

Студент

(підпис)

Керівник роботи

(підпис)

Тетяна САНЖАРЕВСЬКА

(прізвище та ініціали)

Наталія СІЗОВА

(прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка бакалаврської роботи студента групи КН 2021-1 спеціальності 122 «Комп'ютерні науки» Санжаревської Тетяни Сергіївни на тему "Розробка вебдодатку для генерації контенту за допомогою штучного інтелекту" складається з 4 розділів, містить 56 сторінку, 15 таблиць, 8 рисунків, 4 додатки та 24 джерела.

Кваліфікаційну роботу бакалавра присвячено комплексній розробці вебдодатку, основною метою якого є автоматизована генерація як текстового, так і візуального контенту на основі сучасних методів штучного інтелекту. У межах проєкту було спроектовано архітектуру системи, побудовано структуру бази даних, а також реалізовано алгоритми генерації, що взаємодіють із відповідними API (Application Programming Interface) для створення контенту високої якості. Значну увагу в роботі приділено створенню мобільної версії застосунку, що забезпечує адаптацію інтерфейсу під пристрої з різними розмірами екранів, а також дає змогу користувачам працювати з сервісом у будь-який час і в будь-якому місці, незалежно від типу пристрою.

Структура роботи передбачає логічний поділ на чотири ключові розділи, кожен із яких виконує окрему функцію в загальній системі дослідження. У розділі «Загальні положення» здійснено всебічний аналіз предметної області, розглянуто сучасні технологічні рішення, наявні аналоги та сформульовано обґрунтування актуальності обраної тематики, що підкреслює практичну цінність і наукову новизну розробки. Розділ «Інформаційне та математичне забезпечення» містить опис логічної та фізичної моделі даних, структури запитів і контенту, а також представлення алгоритмічної бази, яка забезпечує функціонування системи. У розділі «Програмне та технічне забезпечення» наведено обґрунтування вибору технологічного стеку, інструментів розробки, особливостей реалізації програмних модулів, а також подано детальне керівництво для кінцевого користувача, з урахуванням як десктопних, так і мобільної версій інтерфейсу. Останній розділ – «Охорона праці» – присвячено

питанням безпечної організації роботи з комп'ютерною технікою, ергономіки, дотримання санітарно-гігієнічних норм, а також аналізу чинної нормативно-правової бази, що регламентує діяльність розробника програмного забезпечення.

Ключові слова:

ШТУЧНИЙ ІНТЕЛЕКТ, ГЕНЕРАЦІЯ КОНТЕНТУ, ВЕБДОДАТОК, GPT-4, DALL E, МОБІЛЬНИЙ ІНТЕРФЕЙС, АВТОМАТИЗАЦІЯ, КОРИСТУВАЦЬКИЙ ІНТЕРФЕЙС.

ANNOTATION

Explanatory note of the bachelor`s thesis of the student of the group KN 2021-1, specialty 122 “Computer Science” Tatiana Sanzharevska on the topic “Development of a web application with content generation using artificial intelligence (with the development of a mobile version)” consist of 4 chapters, contains 56 pages, 15 tables, 8 figures, 4 appendices and 24 sources.

The bachelor`s thesis is devoted to the complex development of a web application, the main purpose of which is the automated generation of both textual and visual content based on modern artificial intelligence methods. The project included designing the system architecture, building the database structure, and implementing generation algorithms that interact with the appropriate APIs to create high-quality content. Considerable attention was paid to the creation of a mobile version of the application, which ensures the adaptation of the interface to devices with different screen sizes and allows users to work with the service anytime and anywhere, regardless of the type of device.

The structure of the work provides for a logical division into four key sections, each of which performs a separate function in the overall research system. The General Provisions section provides a comprehensive analysis of the subject area, considers modern technological solutions, existing analogues, and formulates a justification for the relevance of the chosen topic, which emphasizes the practical value and scientific novelty of the development. The section “Information and Mathematical Support” contains a description of the logical and physical data model, query structure and content, as well as a presentation of the algorithmic base that ensures the functioning of the system. The Software and Hardware section provides a rationale for the choice of the technology stack, development tools, and features of the implementation of software modules, as well as a detailed end-user guide, taking into account both desktop and mobile versions of the interface. The last section, Occupational Health and Safety, is devoted to the safe organization of work

with computer equipment, ergonomics, compliance with sanitary and hygienic standards, and analysis of the current regulatory framework governing the activities of software developers.

Keywords:

ARTIFICIAL INTELLIGENCE, CONTENT GENERATION, WEB APPLICATION, GPT-4, DALL E, MOBILE INTERFACE, AUTOMATION, USER INTERFACE

ЗМІСТ

ВСТУП.....	10
РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	16
1.1 Актуальність і предмет дослідження	16
1.2 Аналіз сучасних рішень і технологічних засад	17
1.3 Проблематика впровадження інтелектуальних систем.....	19
1.4 Архітектура, процес автоматизації та постановка задачі.....	20
Висновки до розділу	22
РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	24
2.1 Аналіз предметної області	24
2.1.1 Виділення об'єктів дослідження	24
2.1.2 Опис обмежень на дані	26
2.2 Проектування системи	27
2.2.1 Концептуальне проектування.....	28
2.2.2 Архітектурна модель.....	29
2.2.3 Побудова концептуальної моделі.....	30
2.2.4 ER-діаграма.....	31
2.3 Математичне та алгоритмічне забезпечення	32
2.3.1 Опис моделей реалізації методів	32
Висновки до розділу	33
РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	34
3.1 Вибір інструментів та технологій розробки	34
3.2 Структура та функціональні модулі системи	34

3.3 Демонстрація інтерфейсу користувача	35
3.4 Тестування програмного забезпечення	38
3.4.1 Функціональне тестування	39
3.4.2 Тестування адаптивності	39
3.4.3 UX/UI тестування та зручність використання	40
Висновки до розділу	40
РОЗДІЛ 4 ОХОРОНА ПРАЦІ.....	42
4.1 Регулювання питань охорони праці на законодавчому рівні	42
4.2 Виявлення потенційних небезпек стосовно об'єкту проектування	43
4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження	44
4.3.1 Теоретичні аспекти оцінки ризиків	44
4.3.2 Матриця оцінки ризиків.....	44
4.3.3 Побудова дерева подій.....	45
4.3.4 Рекомендації щодо безпечної організації праці розробники	45
Висновки до розділу	46
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ	53
ДОДАТОК А	53
ДОДАТОК Б.....	54
ДОДАТОК В.....	55
ДОДАТОК Г	56

ВСТУП

Актуальність теми даного дослідження обумовлена загальносвітовими тенденціями, пов'язаними зі стрімким зростанням обсягів цифрової інформації, а також зростаючими вимогами до оперативного створення якісного, релевантного та привабливого контенту для найрізноманітніших цілей. Сучасні комунікаційні платформи, зокрема соціальні мережі, блоги, онлайн-ЗМІ, електронна комерція та тематичні вебсайти, потребують постійного оновлення інформації, що створює серйозне навантаження як на окремих користувачів, так і на команди контент-менеджерів та редакторів.

Традиційні підходи до створення контенту, засновані на ручній роботі, часто виявляються надто повільними, затратними за ресурсами, а також залежними від людського фактора, що не відповідає сучасним вимогам до масштабованості й ефективності цифрового середовища. У зв'язку з цим, виникає об'єктивна необхідність у розробці технологічних рішень, здатних автоматизувати або суттєво спростити процес генерації контенту.

Особливої актуальності набуває використання методів штучного інтелекту (ШІ), зокрема моделей машинного навчання, для створення інтелектуальних систем генерації контенту, які здатні продукувати тексти, зображення, відео та інші інформаційні об'єкти з високим рівнем достовірності, логічності та креативності. У цьому контексті розробка вебдодатку, що інтегрує сучасні ШІ-рішення для генерації текстового та візуального контенту, є не лише доцільною, а й вкрай необхідною у світлі поточного попиту.

Крім того, враховуючи постійну мобільність сучасного користувача та домінування мобільного трафіку в загальній структурі доступу до Інтернету, особливу значущість має реалізація мобільної версії програмного продукту. Вона забезпечує доступ до функціоналу системи з будь-якого пристрою та будь-який час, що значною мірою підвищує зручність використання,

розширює цільову аудиторію та відповідає актуальним стандартам цифрових сервісів.

Метою даної кваліфікаційної роботи є розробка функціонального та інтуїтивно зрозумілого вебдодатку з можливістю автоматизованої генерації текстового та візуального контенту на основі сучасних алгоритмів штучного інтелекту. Особлива увага при цьому приділяється реалізації мобільної версії додатку, що дозволяє забезпечити зручний доступ до його функціоналу з будь-якого мобільного пристрою, незалежно від платформи, типу чи роздільної здатності екрана. Такий підхід відповідає сучасним тенденціям цифровізації та зростаючому попиту на мобільні сервіси з високим рівнем інтерактивності та автоматизації.

Для досягнення поставленої мети в межах дослідження необхідно вирішити низку взаємопов'язаних і практичних завдань:

1. Провести глибокий аналіз предметної області, вивчити особливості ринку контент-генерації, сформулювати базові вимоги до функціональності, архітектури та інтерфейсу майбутнього вебдодатку.

2. Ознайомитися з наявними підходами, технологіями та інструментами, які застосовуються для генерації текстового та графічного контенту із використанням методів штучного інтелекту, зокрема з API (Application Programming Interface) GPT-4 та DALL E.

3. Обґрунтувати вибір конкретних моделей, алгоритмів та підходів, що будуть використані у системі, з урахуванням критеріїв якості, швидкодії, масштабованості та обмежень зовнішніх сервісів.

4. Спроекувати архітектуру вебдодатку, включаючи серверну та клієнтську частини, а також розробити логічну та фізичну модель бази даних, що забезпечує збереження та управління контентом, користувачами та запитами.

5. Реалізувати ключову функціональність, зокрема генерацію текстового та візуального контенту на основі введених користувачем запитів,

інтеграцію з зовнішніми ШІ-сервісами, обробку помилок та виняткових ситуацій.

6. Розробити мобільну версію вебдодатку, що буде адаптована до екранів мобільних пристроїв шляхом реалізації адаптивного інтерфейсу або застосування підходів Progressive Web App (PWA).

7. Провести тестування працездатності системи, включаючи функціональне, мобільне та безпекове тестування, а також оцінити ефективність, зручність використання та відповідність поставленим вимогам.

Об'єктом дослідження у межах даної кваліфікаційної роботи є процес автоматизованої генерації інформаційного контенту за допомогою засобів штучного інтелекту. Під контентом у цьому випадку розуміється як текстова інформація (статті, пости, описові тексти тощо), так і візуальні матеріали (зображення, ілюстрації), які генеруються у відповідь на запити користувача без прямого втручання людини в процес створення.

Предметом дослідження є методи, моделі та алгоритми штучного інтелекту, що лежать в основі процесу генерації контенту, а також програмні засоби, фреймворки та технології, які використовуються для реалізації вебдодатку відповідного призначення. Особлива увага приділяється аналізу ефективності використання моделей машинного навчання, зокрема мовних і генеральних моделей, таких як GPT-4 та DALL·E, а також практичній реалізації інтерфейсів доступу до їх API (Application Programming Interface). Крім того, предмет дослідження охоплює питання адаптації програмного продукту до мобільних пристроїв, включаючи реалізацію мобільної версії за допомогою технологій Progressive Web App (PWA).

У процесі виконання кваліфікаційної роботи було застосовано комплекс взаємопов'язаних методів наукового та прикладного характеру, що забезпечили досягнення поставленої мети та реалізацію практичних завдань дослідження. Зокрема, було використано такі підходи:

- Аналіз літературних джерел, наукових публікацій та існуючих розробок у сфері штучного інтелекту, генерації контенту, веброботи та

мобільних інтерфейсів. Це дозволило сформувати теоретичну базу дослідження, виявити актуальні тенденції розвитку технологій і визначити найбільш ефективні інструменти для реалізації проєкту.

- Системний аналіз та моделювання, що передбачили побудову структурної архітектури вебдодатку, логічного поділу функціональних компонентів, формалізацію взаємозв'язків між елементами системи, а також моделювання процесів генерації контенту з урахуванням потреб користувача.

- Методи машинного навчання та обробки природної мови (NLP), що стали основою для інтеграції ШІ-функціоналу. Було використано сучасні великі мовні моделі (LLM), здатні створювати логічно узгоджений текстовий контент, а також генеративні моделі зображень. Застосування API OpenAI (GPT-4 та DALL·E) дало змогу реалізувати якісну генерацію на основі вхідних параметрів користувача.

- Об'єктно-орієнтоване програмування (ООП), яке використовувалося як базовий підхід у побудові внутрішньої логіки серверної частини вебдодатку. Такий підхід забезпечив модульність, повторне використання коду та зручність у масштабуванні проєкту.

- Методи тестування програмного забезпечення, включаючи функціональне тестування, перевірка адаптивності інтерфейсу, тестування на реальних мобільних пристроях та симуляторах. Також було проведено перевірку працездатності сервісів генерації, коректності обробки помилок, безпеки запитів та стабільності системи в цілому.

Теоритична значущість виконаного дослідження полягає в поглибленні уявлень щодо можливостей застосування методів штучного інтелекту, зокрема мовних і генеративних моделей, для автоматизації процесів створення інформаційного контенту. У межах роботи розглянуто принципи побудови інтелектуальних систем генерації тексту й зображень, а також проаналізовано архітектурні рішення та підходи до інтеграції API сучасних ШІ-моделей у програмні засоби. Отримані результати можуть слугувати базою для

подальших досліджень у суміжних напрямках, зокрема в галузі природної мовної обробки, генеративного дизайну, автоматизованої журналістики тощо.

Практична значущість роботи полягає в створенні цілісного, працездатного вебдодатку, який забезпечує генерацію як текстового, так і візуального контенту на основі введених користувачем запитів. Реалізоване рішення дозволяє значною мірою автоматизувати процес створення інформаційних матеріалів, що, у свою чергу, сприяє економії часу, зниженню витрат і підвищенню продуктивності як окремих користувачів, так і команд, які займаються наповненням вебресурсів, соціальних мереж, маркетингових платформ тощо. Додатково реалізована мобільна версія додатку забезпечує повноцінний доступ до функціональності сервісу з мобільних пристроїв, що підвищує зручність експлуатації, розширює цільову аудиторію та відповідає сучасним вимогам до мобільності програмних засобів.

У межах виконаної кваліфікаційної роботи були отримані результати, що мають як наукове, так і практичне значення. Виносимо такі основні положення:

1. Архітектурна модель та функціональні можливості розробленого вебдодатку, який реалізує процес генерації контенту за допомогою методів штучного інтелекту. Запропоновано багаторівневу архітектуру, що включає клієнтську, серверну частини та взаємодію з зовнішніми API. Здійснено обґрунтований вибір засобів реалізації, включаючи використання Flask, HTML/CSS, JavaScript, Bootstrap та SQLite.

2. Реалізовані алгоритми генерації текстового та візуального контенту, які базуються на інтеграції з API OpenAI (моделі GPT-4 і DALL·E). Забезпечено формування осмислених текстів і релевантних зображень на основі користувацьких запитів, а також створено логіку обробки відповідей, що гарантує стабільність та якість результатів.

3. Результати функціонального тестування вебдодатку, які демонструють його працездатність, відповідність поставленим вимогам та ефективність генерації контенту. Проведено оцінку швидкості обробки

запитів, стабільності роботи інтерфейсу, а також загального користувацького досвіду на різних пристроях і в різних браузерах.

4. Функціональні можливості та особливості мобільної версії вебдодатку, реалізованої з використанням технології Progressive Web App (PWA). Забезпечено адаптацію інтерфейсу для мобільних пристроїв, підтримку офлайн-режиму, встановлення додатку на головний екран смартфона, що відповідає сучасним стандартам мобільних рішень.

РОЗДІЛ 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Актуальність і предмет дослідження

У сучасному цифровому середовищі вебдодатки, що використовують методи штучного інтелекту (ШІ) для генерації контенту, демонструють стрімке зростання популярності та впливу. Це зумовлено як технічним прогресом у галузі машинного навчання, так і зростаючим попитом на автоматизовані засоби створення інформаційного наповнення. Сучасні комунікаційні платформи – соціальні мережі, блоги, електронна комерція – потребують постійного оновлення якісного контенту, що створює навантаження на редакторів і маркетологів.

Згідно з аналітичними даними компанії MarketsandMarkets, глобальний ринок штучного інтелекту у 2024 році оцінювався в 214,6 мільярда доларів США, з річним темпом зростання (CAGR) у 35,7%. Підсегмент автоматизованої генерації контенту, зокрема текстового та візуального, має темпи зростання понад 47% на рік[1]. Це підтверджує актуальність дослідження, спрямованого на створення універсального інструменту для генерації контенту.

Окремої уваги потребує питання мобільного доступу до ШІ-функціональності. Згідно з даними DataReportal (2024), понад 92% користувачів Інтернету у світі здійснюють доступ саме з мобільних пристроїв[8]. Це формує потребу в адаптивних і прогресивних рішеннях, зокрема у форматі Progressive Web App (PWA), здатних забезпечити користувацький комфорт незалежно від типу пристрою.

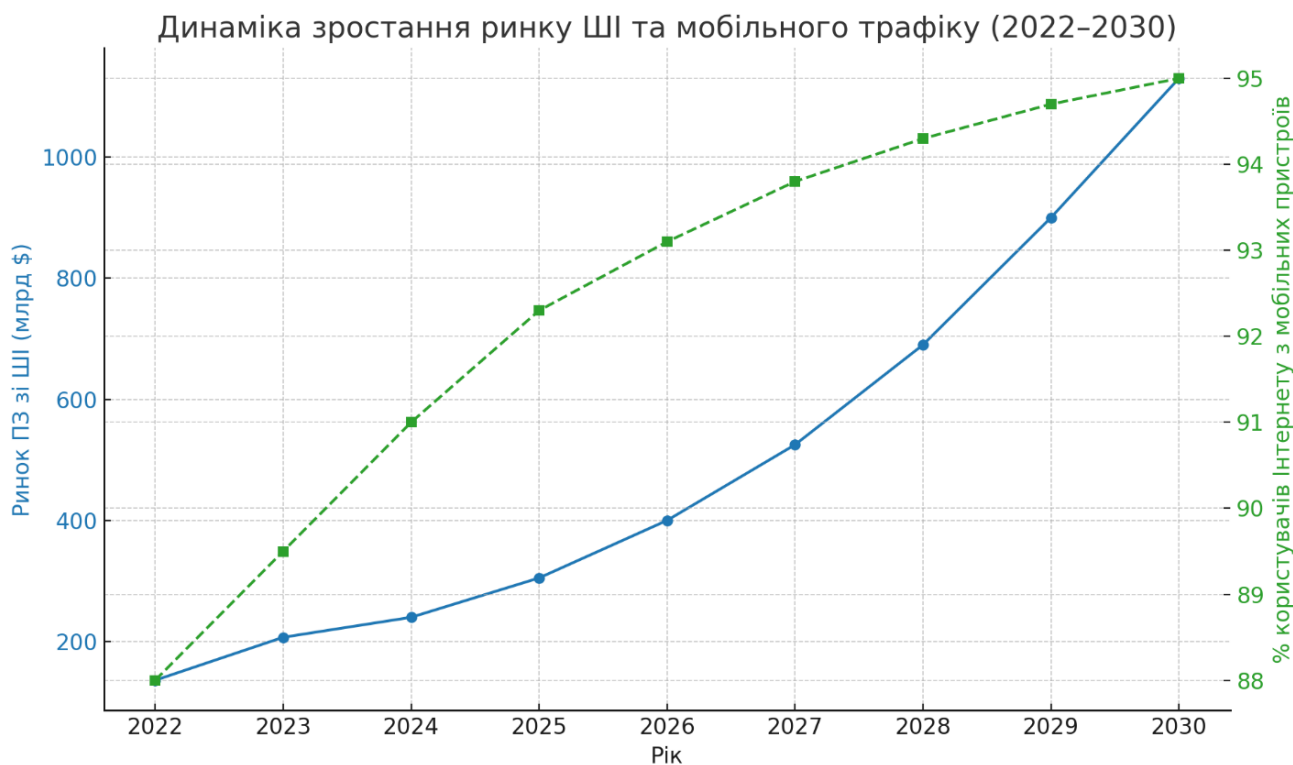


Рисунок 1.1 – Динаміка зростання ринку ШІ та мобільного трафіку

У зв'язку з вищенаведеним, предметом даного дослідження є процес автоматизованої генерації текстового та візуального контенту з використанням моделей GPT-4 і DALL·E через вебдодаток з адаптивним інтерфейсом. Предметна область охоплює програмну інженерію, штучний інтелект, UX/UI-дизайн та нормативне забезпечення цифрових сервісів. Об'єктом дослідження є програмне забезпечення, що реалізує функціонал генерації інформаційного контенту, а мета – розробка доступного та адаптивного вебдодатку для українськомовного користувача.

1.2 Аналіз сучасних рішень і технологічних засад

На сучасному ринку представлено низку рішень, які частково або повністю реалізують функції генерації контенту з використанням ШІ. Найвідоміші з них – ChatGPT, DALL·E, Copy.ai, Jasper, Canva AI [2–6]. Вони пропонують генерацію текстів, зображень, дизайнів та маркетингових

матеріалів. Їх використання дозволяє пришвидшити створення контенту, оптимізувати витрати та персоналізувати повідомлення.

Однак аналіз функціональності цих сервісів виявляє низку обмежень:

- більшість із них мають платну модель доступу (freemium/підписка);
- деякі сервіси не підтримують українську мову або працюють з нею некоректно;
- обмежена адаптація до мобільних пристроїв, складність навігації та відсутність офлайн-функціоналу;
- недостатній контроль користувача над параметрами генерації;
- відсутність універсального середовища з підтримкою і текстоого, і графічного контенту одночасно.

Таблиця 1.1 – Порівняння функціональних можливостей популярних сервісів

Назва	Основна функціональність	Технологія	Мобільна підтримка	Комерційна модель
ChatGPT	Генерація тексту	GPT-4 (OpenAI)	Через браузер, API	Freemium
DALL·E 3	Створення зображень	OpenAI	Через інтеграції	Частково безкоштовно
Jasper	Копірайтинг, маркетинг	GPT-4	Так (додаток)	Платна
Canva AI	Генерація дизайнів і текстів	Власна модель AI + інтеграції	Так	Freemium
Copu.ai	Рекламні тексти, SEO	GPT-3	Так	Freemium
Midjourney	Художня генерація зображень	Власна модель, Discord API	Немає	Платна

Разом з тим, сучасні технології, такі як API GPT-4 і DALL·E, дозволяють створювати високоякісний контент із точним налаштування параметрів генерації. Адаптація інтерфейсу за допомогою технологій PWA робить можливим створення кросплатформеного рішення, доступного з будь-якого

пристрою. Таким чином, поєднання відкритих інтерфейсів API, адаптивного дизайну, локалізованої мовної підтримки та врахування практичних потреб користувачів дає змогу створити більш зручне й ефективне програмне рішення.

1.3 Проблематика впровадження інтелектуальних систем

Попри очевидні переваги, впровадження ІС на основі ШІ пов'язане з низкою викликів, які охоплюють етичні, правові, технічні та соціальні аспекти.

1. Етичні загрози: використання ШІ для створення контенту без маркування джерела може призводити до поширення дезінформації[9].

2. Юридичні ризики: питання авторського права на згенерований контент залишається відкритим. Згідно з українським законодавством[9] та AI Act ЄС [10], відповідальність за зміст автоматично створених матеріалів покладається на користувача, але цей підхід не врегульовано на глобальному рівні.

3. Технічні обмеження: генеративні моделі потребують значних обчислювальних ресурсів. API OpenAI мають обмеження за кількістю запитів, розміром запиту й часом відповіді.

4. Захист персональних даних: застосування ШІ повинно відповідати вимогам GDPR [11], а також принципам етичного використання ШІ, закріпленим у Рекомендаціях ЮНЕСКО (2021) [12].

Таблиця 1.2 – Приклади етичних і правових норм у сфері ШІ

№	Документ/джерело	Юрисдикція/орган	Основні положення
1	Закон України "Про авторське право..."	Україна	Не визначає окремо статус ШІ але охоплює автоматизоване створення творів
2	Загальний регламент захисту даних (GDPR)	ЄС	Регламентує обробку персональних даних користувачів, включаючи ШІ-сервіси
3	AI Act (2023)	ЄС	Встановлює ризик-орієнтований підхід до регулювання систем ШІ: високий, середній, низький
4	Рекомендації ЮНЕСКО (2021)	ЮНЕСКО	Принципи: повага до прав людини, прозорість, інклюзія, відповідальність розробників
5	Digital Services Act	ЄС	Посилює відповідальність за алгоритмічну обробку контенту та боротьбу з маніпуляціями в ШІ-сервісах

Ці чинники формують набір вимог до розробки ШІ-систем: необхідно забезпечити прозорість обробки, безпеку збереження даних, пояснюваність рішень, налаштування рівня відповідальності.

1.4 Архітектура, процес автоматизації та постановка задачі

Розробка системи передбачає побудову архітектури вебдодатку, що складається з клієнтської частини (інтерфейс), серверної частини (логіка обробки запитів) та бази даних (збереження інформації). Важливим є забезпечення взаємодії із зовнішніми сервісам – GPT-4 і DALL·E – через API-запити.

Автоматизований процес можна описати так:

1. Користувач вводить запит через інтерфейс;
2. Вибирає тип генерації (текст/зображення);
3. Система надсилає запит до OpenAI API;

4. Отриманий результат обробляється й відображається у вікні перегляду;
5. Результат можна зберегти або надіслати повторний запит.

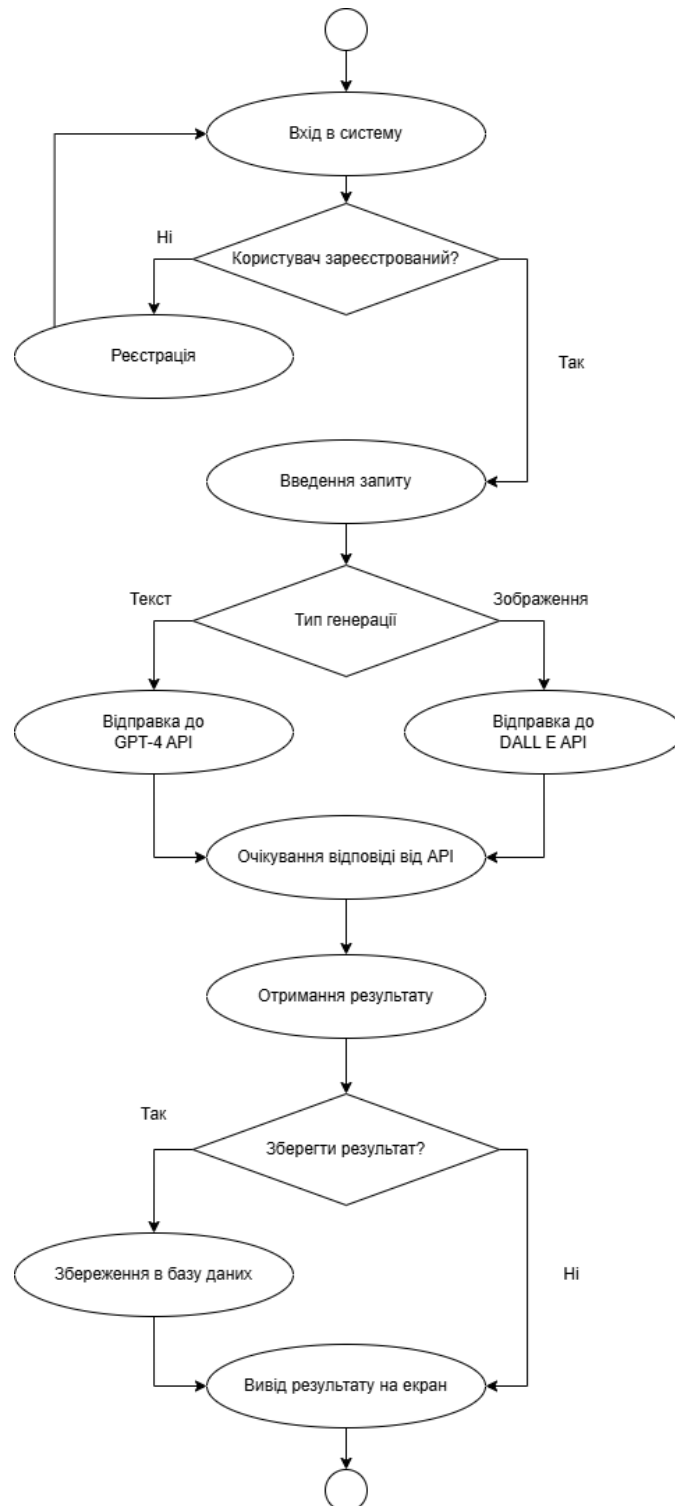


Рисунок 1.2 – UML-діаграма діяльності

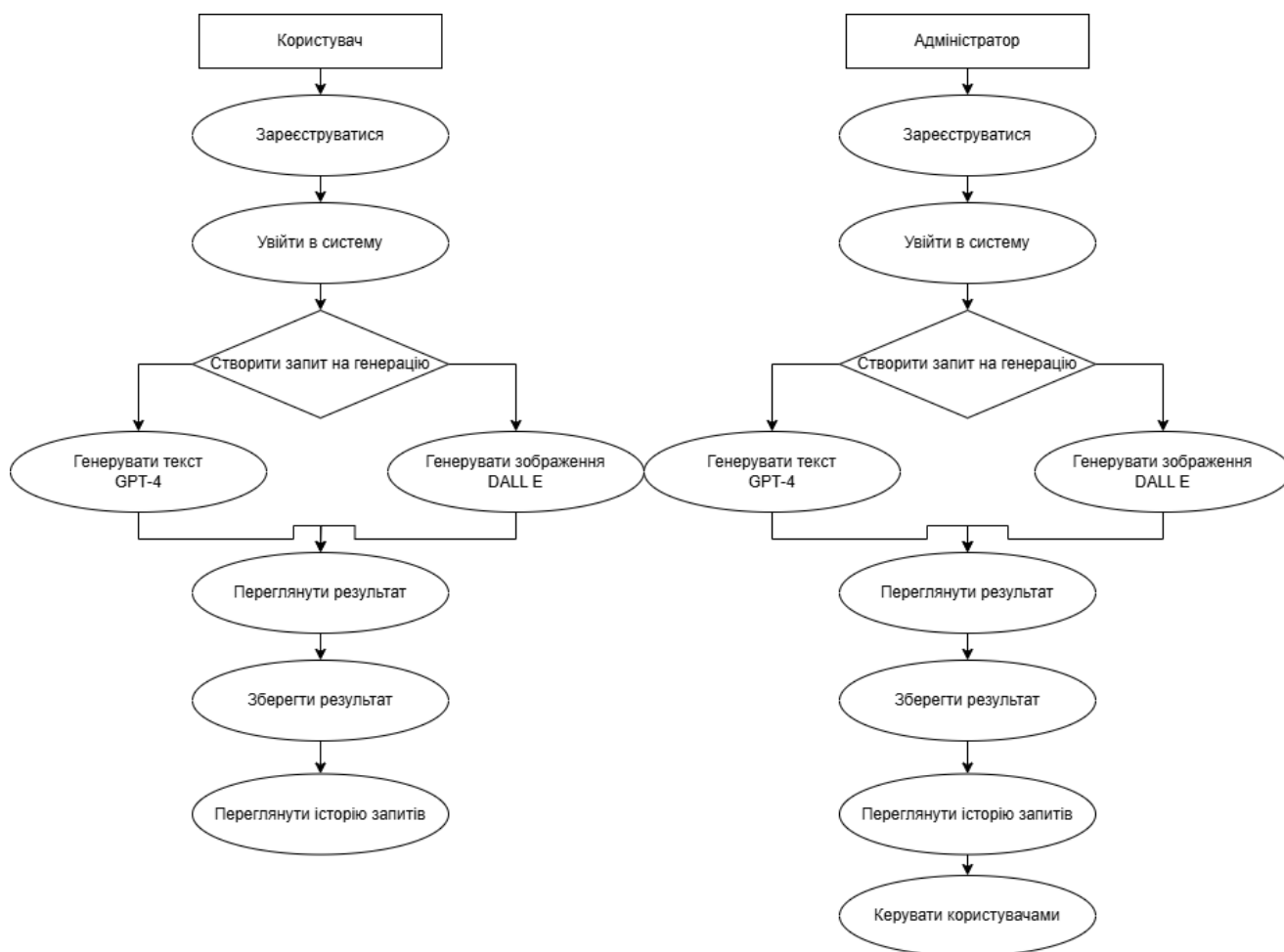


Рисунок 1.3 – UML-діаграма використання

Постановка задачі: створити вебдодаток, що забезпечує генерацію текстового та візуального контенту на основі моделей GPT-4 та DALL·E з адаптивним інтерфейсом і відповідністю етичним, правовим та технічним вимогам. Система повинна мати реєстрацію, підтримку історії запитів, захист даних та багатомовну підтримку.

Висновки до розділу

У межах першого розділу було проведено всебічний аналіз предметної області, яка стосується автоматизованої генерації текстового та візуального контенту з використанням інструментів штучного інтелекту. Це дало змогу визначити ключові технологічні, функціональні та правові аспекти, пов'язані

з реалізацією таких систем. На основі вивчених матеріалів було встановлено, що в умовах цифрової трансформації та глобального зростання інформаційного обсягу потреба в подібних рішеннях постійно зростає.

Окрема увага приділялася порівнянню сучасних сервісів генерації контенту. Аналіз продемонстрував, що хоча на ринку вже існують популярні рішення, такі як ChatGPT, DALL·E, Jasper, Canva AI тощо, більшість із них мають певні обмеження. Зокрема, це стосується недостатньої підтримки української мови, обмеженої адаптації до мобільних платформ, закритості функціоналу або платної моделі доступу. Також були виявлені проблеми з персоналізацією запитів і недостатньою зручністю користування на різних пристроях.

Крім технічних аспектів, у розділі розглянуто проблематику впровадження ШІ-систем у реальні продукти з точки зору етики, права та відповідальності. Особливо важливим є дотримання вимог щодо захисту персональних даних користувачів, прозорості роботи алгоритмів та юридичного статусу згенерованого контенту. Було проаналізовано міжнародні та національні нормативні документи, зокрема GDPR, AI Act, законодавство України в сфері авторських прав тощо.

На основі проведеного аналізу було чітко сформульовано предмет, об'єкт, мету та завдання кваліфікаційної роботи. Зокрема, встановлено необхідність розробки інтелектуального вебдодатку, який би поєднував у собі можливості генерації як текстового, так і графічного контенту, мав дружній та адаптивний інтерфейс, підтримував українську мову і відповідав етичним та правовим вимогам сучасного цифрового середовища.

Таким чином, результати розділу заклали необхідне підґрунтя для подальшого логічного проектування системи, визначення її функціональних компонентів, побудови архітектури, моделювання структури даних, а також реалізації практичних механізмів генерації контенту. Це дозволяє зробити висновок про актуальність і доцільність обраного напрямку дослідження в межах комп'ютерних наук.

РОЗДІЛ 2 ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Аналіз предметної області

2.1.1 Виділення об'єктів дослідження

У межах даної кваліфікаційної роботи досліджується процес генерації текстового та візуального контенту із застосуванням методів штучного інтелекту в рамках вебдодатку, орієнтованого на кінцевого користувача. З метою системного підходу до аналізу предметної області було визначено ключові об'єкти, які безпосередньо або опосередковано беруть участь у функціонуванні системи, а також побудовано первинну інформаційну модель взаємодії між ними.

До основних об'єктів системи належать:

- Користувач – головний суб'єкт взаємодії з додатком. Може мати різні рівні доступу, що визначають його повноваження у межах системи:
 - Незареєстрований користувач – має обмежений доступ до функціоналу (наприклад, обмежену кількість генерацій);
 - Зареєстрований користувач – має повний доступ до основних функцій системи, можливість переглядати історію запитів, зберігати результати тощо;
 - Адміністратор – користувач із розширеними правами управління (модерація, перегляд статистики, зміна параметрів).
- Інтерфейс додатку – фронтендна частина системи, яка забезпечує користувачу зручний та зрозумілий доступ до всіх функцій. Інтерфейс реалізовано з урахуванням принципів адаптивного дизайну для коректної роботи як у десктопних браузерах, так і на мобільних пристроях. Основні

елементи: поля вводу, кнопки вибору типу генерації, область виводу результатів, історія запитів.

- Серверна частина (бекенд) – відповідає за обробку запитів від клієнта, маршрутизацію, зв'язок з базою даних та зовнішніми ШІ-сервісами, збереження результатів генерації, ведення сесій користувачів та забезпечення бізнес-логіки додатку.

- Зовнішні API ШІ-сервісів – зокрема API компанії OpenAI, до якого надсилаються запити на генерацію:

- GPT-4[2] – для створення текстового контенту на основі введених користувачем фраз;
- DALL·E[3] – для генерації зображень за текстовим описом. Важливою особливістю є параметризація запитів, обмеження по кількості генерацій, та обробка часу відповіді.

- База даних – централізоване сховище, що містить всю важливу інформацію для забезпечення роботи додатку:

- Облікові записи користувачів (логіни, email, паролі у зашифрованому вигляді);
 - Історію генерації (запити, результати, типи контенту, дата/час);
 - Конфігураційні параметри генерації;
 - Службову інформацію (сесії, логування, тощо).

Усі зазначені об'єкти перебувають у тісному взаємозв'язку між собою та утворюють єдину інформаційну систему. Взаємодія між компонентами забезпечує цілісне функціонування вебдодатку, підтримує циклічність обробки запитів, збереження результатів та зворотній зв'язок із користувачем.

У таблиці 2.1 подано основні сутності, що використовуються у моделі, та їх атрибути.

Таблиця 2.1 – Атрибути основних сутностей

Сутність	Атрибути
Користувач	ID, Ім'я, Email, Пароль (хешований), Роль (звичайний/адмін), Дата реєстрації
Запит	ID, Тип запиту (текст / зображення), Вміст запиту, Дата та час запиту, ID користувача
Результат (Контент)	ID, Тип контенту, Файл (текст/зображення), Формат, Статус, Пов'язаний ID запиту
Сесія	ID сесії, ID користувача, Час входу, Час виходу, IP-адреса
Повідомлення про помилку	ID, Тип помилки, Код відповіді API, Пов'язаний ID запиту, Час виникнення

Загальна логічна модель системи розроблена з урахуванням принципів масштабованості, надійності та функціональної гнучкості. Її структура зосереджена на ефективному обслуговуванні генеративних запитів користувачів, зокрема на забезпеченні швидкої взаємодії з зовнішніми API сервісів штучного інтелекту, збереженні результатів генерації у відповідних форматах, а також на контролі дій користувачів – включаючи реєстрацію, автентифікацію, зберігання історії, налаштування та права доступу. Особливу увагу приділено можливості подальшого масштабування системи: передбачено розділення функціональних компонентів, модульність реалізації та зручність додавання нових функцій без необхідності перебудови всієї архітектури. Такий підхід забезпечує стабільну роботу додатку в умовах зростання навантаження та дозволяє адаптувати систему до розширення цільової аудиторії в майбутньому.

2.1.2 Опис обмежень на дані

Проектована система повинна функціонувати в умовах, які забезпечують стабільність, безпеку та відповідність політиці сервісів, що надають API. У зв'язку з цим, визначаються наступні обмеження:

- Обмеження на довжину запиту: Для запитів GPT-4 – максимум 1000 символів; Для запитів DALL·E - 1 коротке речення або фраза (до 256 символів) через специфіку генерації зображень.
- Кількість запитів: Один користувач має обмеження на 20 запитів/годину для збереження стабільності системи. Це обмеження може бути змінено адміністратором.
- Формат і розмір файлів: Зображення генеруються у форматі .png розмір – 1024x1024 пікселів (встановлено API DALL·E); Тексти зберігаються у вигляді .txt або в базі даних як текстові рядки.
- Політика контенту: Система блокує запити, які можуть містити образливу лексику, насильство, дискримінацію, або суперечать політиці OpenAI. Це забезпечується за допомогою систем фільтрації, які перевіряють запити до їх обробки.
- Безпека облікових записів: Паролі зберігаються в базі даних у вигляді хешів із застосуванням алгоритмів bcrypt або SHA-256. Сесії мають обмеження за часом (30 хвилин бездіяльності – автоматичне завершення).

2.2 Проектування системи

Проектування інформаційної системи є одним із ключових та найвідповідальніших етапів у процесі розробки програмного забезпечення. Саме на цій стадії закладається основа для подальшої реалізації функціональних можливостей, масштабованості, надійності та ефективності майбутнього додатку. Проектування передбачає побудову логічної структури системи, визначення її основних функціональних компонентів, опис взаємодії між ними, а також розробку моделі обробки й зберігання даних, що циркулюють у процесі роботи.

У межах даного етапу формалізуються вимоги до системи, визначаються архітектурні шаблони, програмні інтерфейси, модулі, обмінні формати та інші

технічні характеристики, які забезпечують цілісність функціонування вебдодатку. Особливу увагу приділено інформаційній моделі, яка описує сутності, атрибути та зв'язки між даними, що дозволяє ефективно організувати логіку роботи з користувачами, запитами та результатами генерації.

Урахування всіх взаємозв'язаних елементів на етапі проектування дозволяє мінімізувати ризики помилок під час реалізації, полегшити тестування, забезпечити зручність підтримки та можливість подальшого розширення системи. Таким чином, проектування є не лише формальним етапом розробки, а й фундаментом, від якого залежить успішність програмного продукту загалом.

2.2.1 Концептуальне проектування

Концептуальне проектування є одним із перших важливих етапів створення програмного забезпечення, оскільки саме на цьому рівні формується загальне бачення майбутньої інформаційної системи. Основна мета цього етапу – побудувати абстрактну модель додатку, яка відображає ключові функціональні компоненти, визначає їхню структуру, взаємозв'язки та логіку взаємодії між елементами системи. Таким чином, закладається основа для технічного проектування реалізації, тестування та подальшого супроводу розробки.

У процесі концептуального проектування здійснюється визначення архітектурного стилю системи (наприклад, клієнт-серверна модель), формалізується модульна структура, що включає фронтенд, бекенд, базу даних і зовнішні сервіси, а також визначаються програмні засоби, які будуть використані для реалізації функціоналу. Одночасно з цим розробляються сценарії використання системи (use cases), що описують типову поведінку користувачів у різних ситуаціях.

Особливу увагу на цьому етапі приділено визначенню ролей користувачів (гостьовий доступ, зареєстрований користувач, адміністратор) і

їхніх повноважень, що безпосередньо впливають на доступ до функціоналу системи та управління контентом. Також моделюються потоки даних, що циркулюють між модулями (наприклад, запит → генерація → відповідь → збереження), з урахуванням ефективності, безпеки та розширюваності.

Концептуальна модель є орієнтиром для всієї подальшої роботи над системою, оскільки дозволяє узгодити вимоги, зменшити ризики непорозумінь на етапі реалізації та забезпечити цілісність архітектурного підходу.

2.2.2 Архітектурна модель

Архітектура вебдодатку є критично важливою для забезпечення ефективної взаємодії між користувачем, внутрішньою логікою системи та зовнішніми сервісами штучного інтелекту. У процесі проектування була обрана класична трирівнева клієнт-серверна модель, що дозволяє гнучко розмежувати обов'язки між компонентами, забезпечити масштабованість та полегшити подальший супровід.

Загальна структура системи відображена у таблиці 2.2.

Таблиця 2.2 – Структура системи

Рівень	Опис
Клієнтський	Реалізований на HTML/CSS, JavaScript або React. Забезпечує взаємодію з користувачем: введення запитів, перегляд результатів, навігація, авторизація.
Серверний	Побудований на Python (Flask/Django). Обробляє запити, підключається до OpenAI API, зберігає дані в базі, керує сесіями та безпекою.
Зовнішні сервіси	GPT-4 (генерація тексту) та DALL·E (генерація зображень) – через OpenAI API.

Для реалізації кожного рівня було обрано оптимальні інструменти, з урахуванням стабільності, документованості та підтримки спільнотою.

Таблиця 2.3 – Інструменти до кожного компонента

Компонент	Інструмент/Технологія
Мови програмування	Python (бекенд), JavaScript (фронтенд)
Фреймворки	Flask/Django, Bootstrap/Tailwind
База даних	SQLite (або PostgreSQL)
AI API	OpenAI (GPT-4, DALL·E)
Візуалізація	Chart.js, UML-діаграми через Draw.io

Описана архітектура забезпечує надійність і адаптивність роботи системи при зростанні навантаження, дає змогу інтегрувати додаткові модулі та полегшує адаптацію під мобільні пристрої.

2.2.3 Побудова концептуальної моделі

Для опису логіки об'єктно-орієнтованої структури системи використано UML-діаграму класів, яка демонструє ключові сутності вебдодатку, їх атрибути, методи та взаємозв'язки. Це дозволяє чітко формалізувати логіку взаємодії між об'єктами, а також закласти основу для майбутньої програмної реалізації.

Таблиця 2.4 – Основні класи системи

Клас	Атрибути	Методи
User	id, username, email, password, role	register(), login(), logout()
Query	id, user_id, type (text/image), prompt, timestamp	submit(), validate()
GeneratedContent	id, query_id, content_type, file_path, format, status	save(), export()
Admin (наслідник User)	–	manage_users(), monitor_logs()

Такий підхід дозволяє реалізувати принцип інкапсуляції, розділити відповідальність між модулями та полегшити тестування й розширення функціональності. Крім того, успадкування класу Admin від User забезпечує збереження спільної логіки з можливістю розширення прав і методів.

Концептуальна модель є логічною надбудовою над архітектурною, деталізуючи поведінку об'єктів, що становлять основу функціоналу вебдодатку.

2.2.4 ER-діаграма

Для забезпечення ефективного зберігання, доступу та обробки даних у межах вебдодатку було спроектовано ER-діаграму (діаграму “сутність– зв’язок”), яка формалізує логічні відношення між ключовими сутностями інформаційної системи. Цей підхід дозволяє створити узгоджену, цілісну і масштабовану структуру бази даних.

Таблиця 2.5 – Основні сутності та їх зв'язки

Сутність	Зв'язок	Сутність, з якою пов'язано	Коментар
Користувач (User)	один-до-багатьох	Запит (Query)	Один користувач може створювати кілька запитів
Запит (Query)	один-до-багатьох	Контент (GeneratedContent)	Один запит може породити кілька результатів (текст/зображення)
Користувач (User)	один-до-багатьох	Сесія (Session)	Кожен користувач може мати багато входу в систему
Запит (Query)	один-до-багатьох	Помилка (ErrorLog)	Повідомлення про помилки зберігаються окремо з прив'язкою до запиту

Центральною сутністю виступає "Користувач", з яким безпосередньо пов'язано кілька підсистем: збереження запитів, управління контентом, моніторинг сесій та логування помилок. Це забезпечує повну трасованість взаємодій і дозволяє реалізувати зручний інтерфейс користувача з історією запитів та результатів.

Переваги запропонованої ER-моделі:

- "Мінімізація надлишковості (нормалізована структура)"
- "Логічна узгодженість даних між сутностями"

- "Готовність до масштабування (можливість додавання нових сутностей)"

Реалізація цієї структури в базі даних (наприклад, у SQLite чи PostgreSQL) дозволяє ефективно виконувати SQL-запити для генерації звітів, аналізу активності користувачів і забезпечення безпеки даних.

2.3 Математичне та алгоритмічне забезпечення

2.3.1 Опис моделей реалізації методів

Хоча генерація контенту здійснюється за допомогою зовнішніх ШІ-сервісів, у системі присутні й власні алгоритми, які відповідають за:

- Фільтрацію запитів: Використання регулярних виразів для виявлення некоректних запитів (фрази, які містять образливу лексику, заборонені теми тощо).
- Обробку помилок: Формальні правила перевірки відповіді API – при виявленні помилкового запиту користувачеві повертається повідомлення з поясненням.
- Логіку доступу: Проста рольова модель доступу реалізується через алгоритм:

```
if user.role == "admin":
    show_admin_panel()
elif user.role == "user":
    allow_generation()
else:
    redirect_to_login()
```

- Механізми кешування: Кешування результатів у БД для унеможливлення повторного звернення до API при ідентичному запиті.

Алгоритмічна база системи побудована на принципах простоти, ефективності та інтеграції з зовнішнім штучним інтелектом.

Висновки до розділу

У другому розділі кваліфікаційної роботи було здійснено проектування інформаційного та математичного забезпечення системи, що забезпечує автоматизовану генерацію текстового та візуального контенту з використанням методів штучного інтелекту. На основі аналізу предметної області виділено основні об'єкти системи, описано їх характеристики та взаємозв'язки. Сформульовано логічну модель, яка враховує структуру даних, користувацькі ролі, типи запитів і відповіді, а також сценарії взаємодії користувача з інтерфейсом системи.

Розроблено концепцію бази даних, що включає сутності "Користувач", "Запит", "Результат", "Сесія", "Повідомлення про помилку" з відповідними атрибутами, що забезпечує ефективне зберігання і обробку інформації в межах веб додатку. Особливу увагу приділено аспектам безпеки, включаючи хешування паролів, обмеження сесій за часом, контроль доступу та обмеження на параметри генерації.

Здійснено проектування архітектури інформаційної системи на основі клієнт-серверної моделі, з виділенням основних компонентів: фронтенду, бекенду, сховища даних і зовнішніх сервісів (API OpenAI – GPT-4 та DALL·E). Визначено ключові технології розробки, зокрема Flask, HTML/CSS, JavaScript, а також інтеграції з ШІ-моделями.

Також представлено UML-діаграми та таблиці, що формалізують інформаційні структури та поведінку системи. Ураховано обмеження щодо параметрів запитів, форматів контенту та відповідності вимогам зовнішніх API, що дозволяє забезпечити стабільну та безпечну роботу додатку.

Таким чином, розділ 2 створює концептуальне та структурне підґрунтя для реалізації функціональних модулів системи, слугує основою для програмної реалізації вебдодатку та гарантує його відповідність вимогам до масштабованості, безпеки та зручності використання.

РОЗДІЛ 3 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Вибір інструментів та технологій розробки

Ефективна розробка вебдодатку, який забезпечує генерацію текстового та візуального контенту за допомогою штучного інтелекту, вимагає ретельного добору технологічного стеку. Основними критеріями стали: простота інтеграції з OpenAI API, адаптивність інтерфейсу для мобільних пристроїв, зручність розгортання та підтримка сучасних вебстандартів.

Таблиця 3.1 – Вибір інструментів

Компонент	Технології/Інструменти	Обґрунтування вибору
Серверна частина	Python, Flask	Гнучкий фреймворк для створення REST-сервісів
Клієнтська частина	HTML5, CSS3, Bootstrap, JavaScript	Побудова адаптивного та кросбраузерного інтерфейсу
База даних	SQLite	Вбудована СУБД, проста в налаштуванні, підходить для MVP
Генерація контенту	OpenAI API (GPT-4, DALL·E 3)	Високий рівень якості текстової і візуальної генерації
Інструменти розробки	VS Code, Postman, Git	Зручність налагодження, тестування, командної роботи

Окрему увагу приділено реалізації мобільної версії інтерфейсу, яка виконана за принципами Progressive Web App (PWA). Це забезпечує швидке завантаження, офлайн-режим і можливість вставлення застосунку на головний екран пристрою без потреби в окремому мобільному додатку.

3.2 Структура та функціональні модулі системи

Система побудована за модульним принципом, що дозволяє ефективно масштабувати її функціональність, розділяти логіку та підтримувати код.

Архітектура додатку умовно поділена на клієнтську, серверну та зовнішню частину (інтеграція з OpenAI API). Кожен модуль виконує чітко визначену роль.

Таблиця 3.2 – Опис функціональності модулів

№	Назва модуля	Опис функціональності
1	Клієнтський інтерфейс	Інтерактивні форми введення запитів, вибір типу генерації, відображення результатів
2	Серверна логіка	Обробка HTTP-запитів, взаємодія з БД, передача даних до зовнішніх сервісів
3	Модуль генерації	Відправлення запитів до GPT-4 або DALL·E, обробка й форматування результатів
4	База даних	Збереження користувачів, історії генерацій, файлів, помилок та налаштувань
5	Модуль авторизації	Реєстрація/вхід користувача, хешування паролів, керування сесіями
6	Збереження/експорт	Збереження тексту у .txt, зображень у .png, кнопки експорту у фронтенді

Кожен компонент ізольований від інших, що полегшує тестування та модифікацію. Така модульність дозволяє розширити функціональність, не змінюючи основну логіку системи.

3.3 Демонстрація інтерфейсу користувача

Інтерфейс вебдодатку реалізовано з урахуванням принципів зручності (usability), доступності (accessibility) та естетичної простоти. Користувач взаємодіє з формами генерації тексту чи зображення, має доступ до історії запитів, повідомлень про помилки та кнопок експорту результатів.

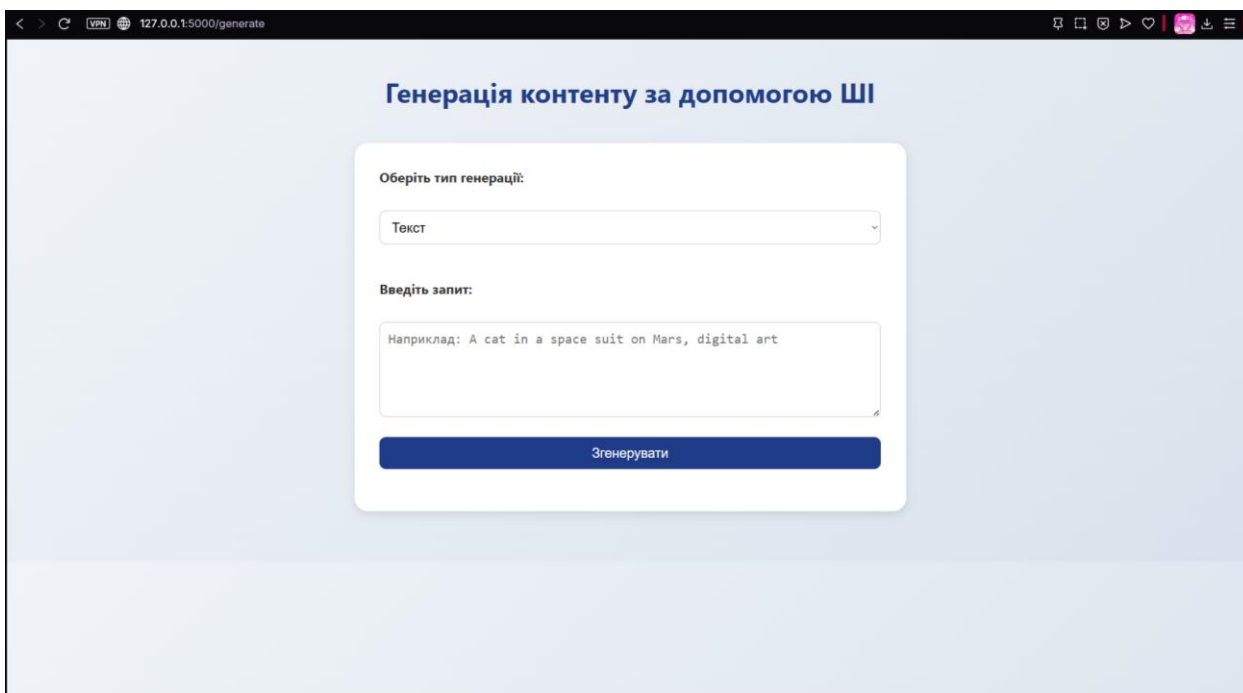


Рисунок 3.1 – Головна сторінка з формою генерації тексту

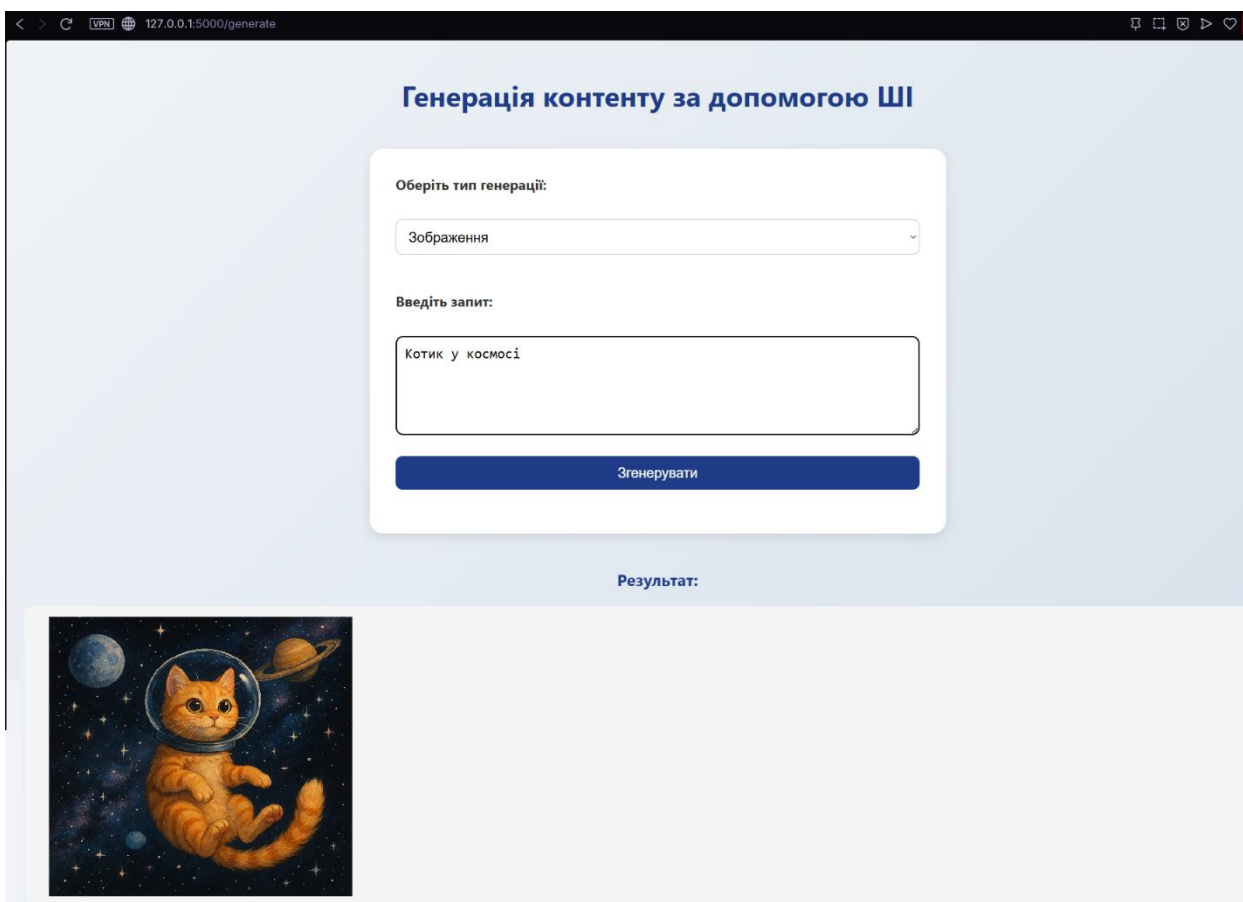


Рисунок 3.2 – Виведення згенерованого зображення

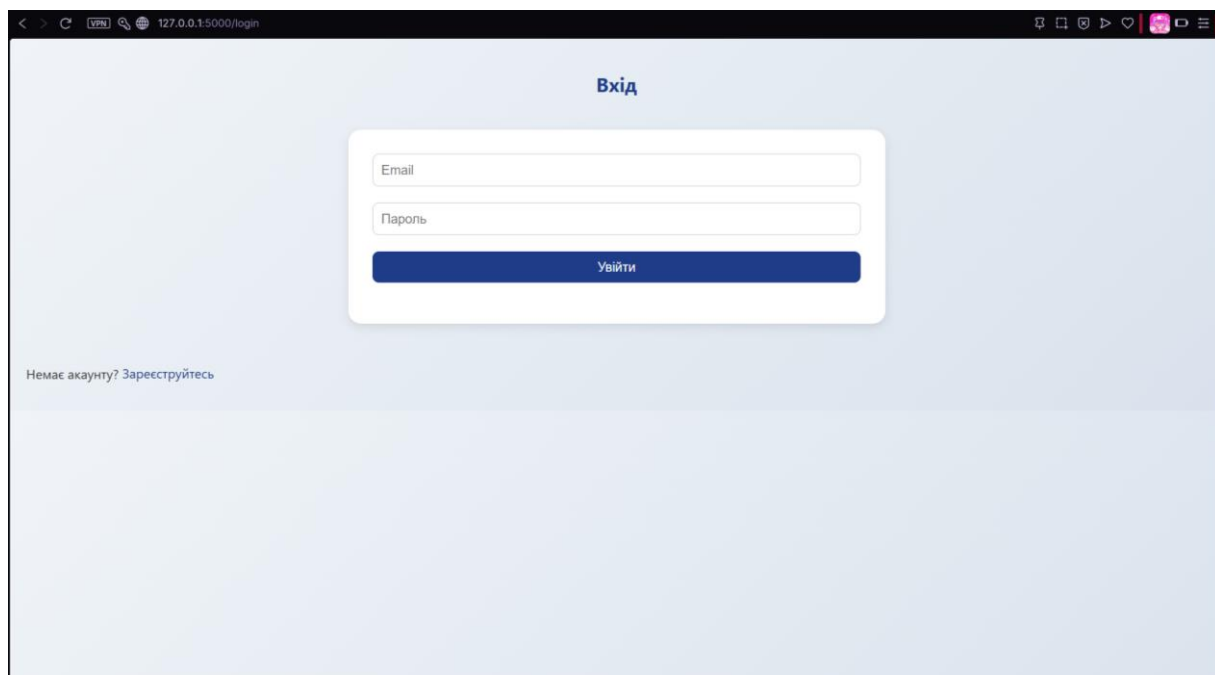


Рисунок 3.3 – Форма входу користувача

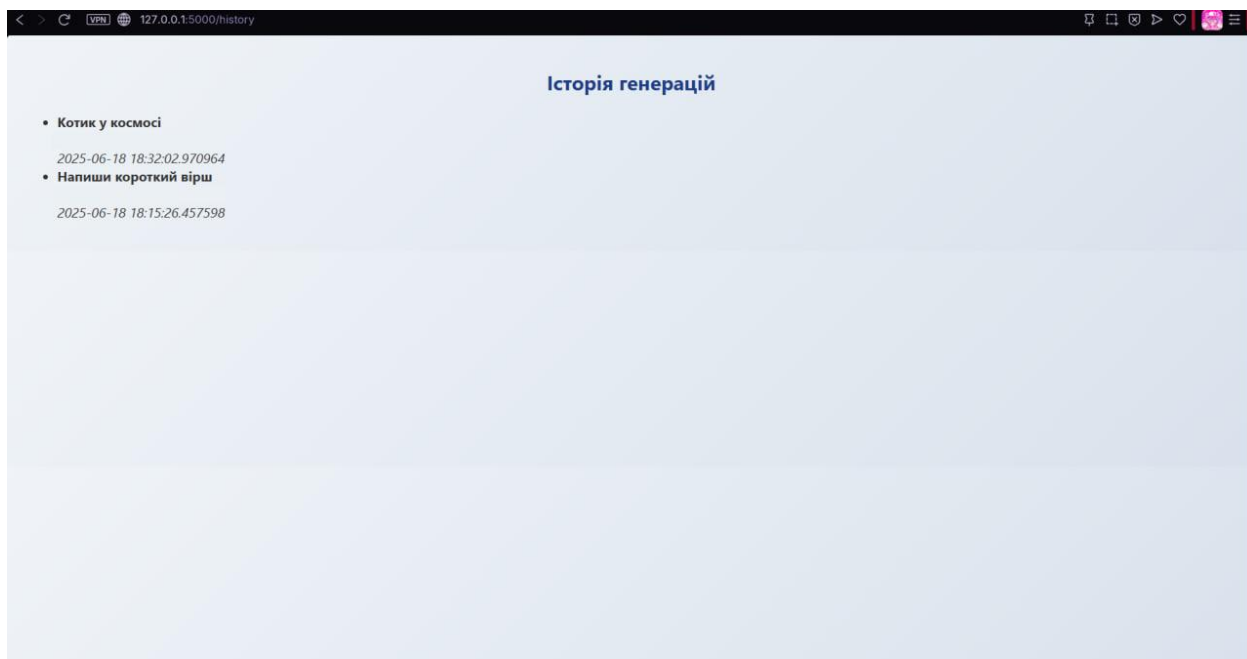


Рисунок 3.4 – Перегляд історії запитів

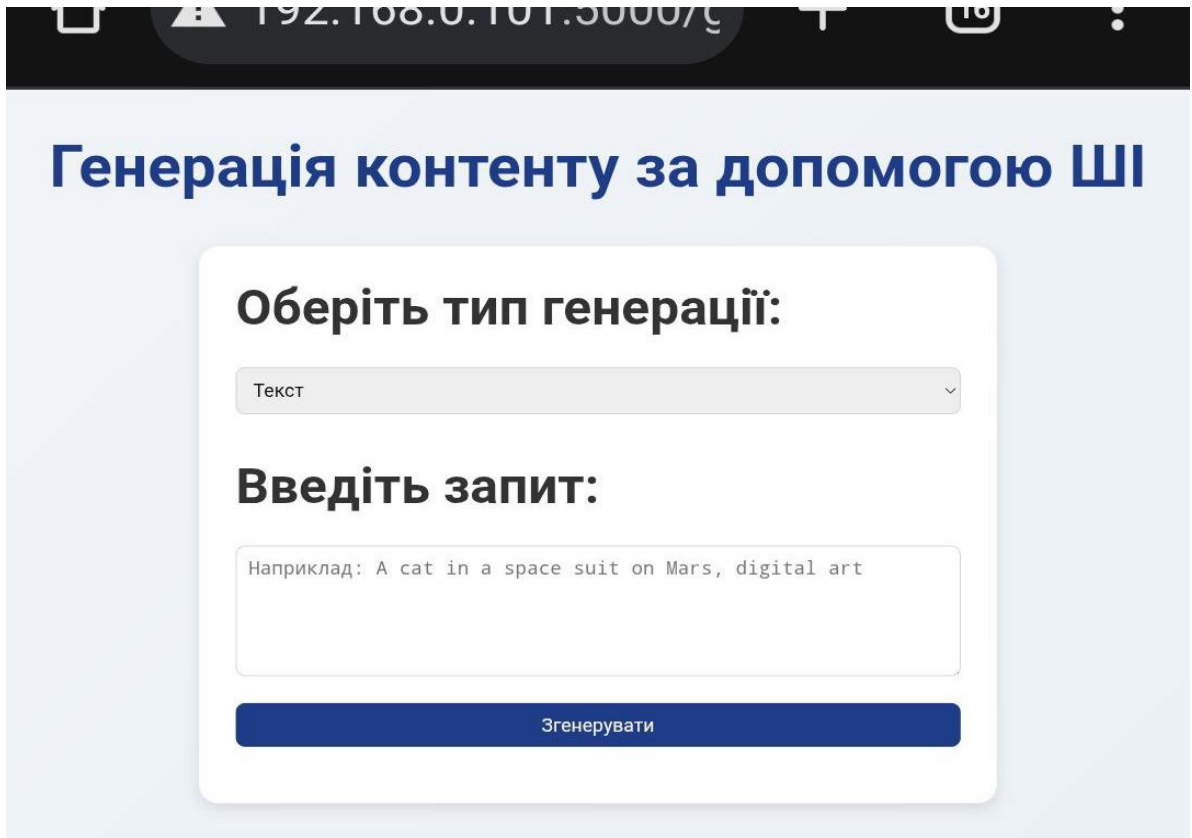


Рисунок 3.5 – Адаптивна мобільна версія додатку

Завдяки застосуванню Bootstrap, інтерфейс коректно адаптується під різні розміри екранів, зокрема смартфони й планшети, зберігаючи логіку взаємодії на будь-якому пристрої.

3.4 Тестування програмного забезпечення

Якість та стабільність функціонування вебдодатку перевірялась через комплексне тестування, яке включало функціональні, адаптивні та UX-тести. Це дозволило виявити потенційні помилки, перевірити коректність роботи всіх компонентів, а також оцінити зручність використання.

3.4.1 Функціональне тестування

Таблиця 3.3

№	Тестовий сценарій	Вхідні дані	Очікуваний результат	Результат
1	Генерація тексту	"Напиши короткий вірш"	Повноцінний осмислений текст	Пройдено
2	Генерація зображення	"Котик у космосі"	PNG-файл 1024×1024	Пройдено
3	Реєстрація користувача	Email, пароль	Новий запис у таблиці users	Пройдено
4	Вхід користувача	Логін, пароль	Перенаправлення на головну сторінку	Пройдено
5	Збереження результатів	Результат генерації	Запис у таблиці results	Пройдено

3.4.2 Тестування адаптивності

Проведене тестування демонструє, що вебдодаток повністю адаптивний і коректно працює на більшості мобільних пристроїв. Візуальні компоненти масштабуються без втрати логіки чи зручності навігації.

Таблиця 3.4

№	Пристрій	Результат тесту	Статус
1	Смартфон Android (360x640)	Усі елементи на місці, кнопки адаптуються	Пройдено
2	iPhone 13 Pro (390x844)	Коректне відображення, шрифт адаптований	Пройдено
3	Планшет (768x1024)	Масштабування успішне, сітка адаптована	Пройдено

3.4.3 UX/UI тестування та зручність використання

Тестування інтерфейсу зосереджувалося на таких аспектах, як інтуїтивність навігації, зрозумілі повідомлення та час взаємодії з основними функціями.

Таблиця 3.5

Параметр оцінки	Пояснення	Оцінка
Час генерації результату	Менше 2 сек у стабільному середовищі API	Висока
Зрозумілість інтерфейсу	Інтуїтивна навігація, чіткі кнопки	Висока
Реакція на помилки	Зрозумілі повідомлення (наприклад, "Запит порожній")	Висока
Візуальна структура сторінки	Збалансована, приємна для сприйняття	Висока

Висновки до розділу

У третьому розділі кваліфікаційної роботи було безпосередньо реалізовано програмну частину системи, яка забезпечує генерацію текстового та візуального контенту із залученням сучасних технологій штучного інтелекту. Розробка проводилася відповідно до завдань, визначених у попередніх розділах, із урахуванням функціональних, технічних та користувацьких вимог.

На етапі вибору технологій було обґрунтовано доцільність використання Python із фреймворком Flask для побудови серверної логіки, а також HTML, CSS, Bootstrap і JavaScript – для розробки адаптивного клієнтського інтерфейсу. Такий вибір забезпечив високу гнучкість, масштабованість і зручність у підтримці коду. Застосування SQLite як простої та надійної бази даних дозволило легко інтегрувати збереження інформації про користувачів, запити та згенеровані результати.

Функціональна структура системи була побудована на принципах модульності, що забезпечило логічне розмежування обов'язків між компонентами. Створені модулі охоплюють як роботу з API OpenAI для

генерації тексту (GPT-4) та зображень (DALL·E), так і взаємодію з базою даних, обробку помилок, авторизацію користувачів, а також інтерфейс для перегляду результатів і історії генерацій.

Окремий акцент зроблено на реалізації адаптивного інтерфейсу, який був протестований на різних типах пристроїв – смартфонах, планшетах, десктопах – і продемонстрував стабільну та коректну роботу. Завдяки технології Progressive Web App (PWA), користувач отримує доступ до сервісу не тільки через браузер, а й у вигляді встановлюваного додатку, з підтримкою офлайн-режиму, що значно підвищує зручність і доступність.

Тестування охоплювало функціональні перевірки, UX/UI оцінювання, перевірку стабільності генерації, обробку виняткових ситуацій та реакцію системи на некоректні запити. Результати тестів підтвердили працездатність системи, відповідність заявленим вимогам і високий рівень зручності для кінцевого користувача.

Загалом, третій розділ продемонстрував, що вебдодаток повністю відповідає меті дослідження, реалізує заявлений функціонал, має інтуїтивний інтерфейс, стабільно взаємодіє з API сучасних ШІ-моделей і може бути використаний для широкого кола завдань, пов'язаних із автоматизованою генерацією контенту.

РОЗДІЛ 4 ОХОРОНА ПРАЦІ

4.1 Регулювання питань охорони праці на законодавчому рівні

Охорона праці є ключовим елементом системи безпеки праці в будь-якій галузі, зокрема й у сфері інформаційних технологій. Законодавче регулювання охорони праці в Україні здійснюється на основі Конституції України, Кодексу законів про працю України (КЗпП) [15], Закону України "Про охорону праці" №2694-ХІІ[14], а також численних підзаконних актів, стандартів і санітарних норм.

Закон України "Про охорону праці"[14] визначає загальні вимоги до організації безпечних і здорових умов праці на підприємствах, в установах і організаціях, незалежно від форми власності. Згідно зі ст. 13 цього Закону, роботодавець зобов'язаний створити умови праці відповідно до вимог нормативно-правових актів, а також забезпечити працівників необхідними засобами індивідуального захисту, навчанням і контролем.

У сфері ІТ безпечні умови праці регламентуються також іншими документами:

- ДСанПіН 3.3.2.007-98 – щодо гігієнічних вимог до ПЕОМ[17];
- ДСТУ ISO 45001:2019 – система управління охороною праці[16];
- ДБН В.2.5-28:2018 – вимоги до освітлення[18];
- ДСН 3.3.6.042-99 – мікроклімат приміщень[19];
- Закон України "Про авторське право і суміжні права", GDPR, AI Act (ЄС) – опосередковано впливають на умови цифрової безпеки та етики[24].

На рівні підприємства (ІТ-компанії, стартапу, фриланс-структури) питання охорони праці мають бути відображені у внутрішніх інструкціях, положеннях про охорону праці, правилах безпечного користування комп'ютерною технікою, а також в угодах з віддаленими працівниками.

У випадку індивідуальної розробки (як у цій дипломній роботі), відповідальність за дотримання вимог безпеки повністю покладається на розробника[20]. Це вимагає обізнаності з чинним законодавством, уміння ідентифікувати потенційні ризики та організувати робоче місце відповідно до встановлених вимог.

4.2 Виявлення потенційних небезпек стосовно об'єкту проектування

Розробка програмного забезпечення, зокрема вебдодатків, не є фізично важкою працею, проте супроводжується низкою професійних небезпек: фізичних, хімічних, психофізіологічних та загальносоціальних.

Таблиця 4.1 – Найтипівіші небезпеки для розробника ІТ-системи

№	Фактор	Група	Можливі наслідки
1	Недостатнє освітлення	Фізичний	Перевтома очей, головний біль
2	Тривала робота за ПК	Психофізіологічний	Порушення постави, статична перевтома
3	Випромінювання від моніторів, Wi-Fi, ПК	Фізичний	Електромагнітне навантаження
4	Перевантаження електромереж	Пожежонебезпека	Ризик займання техніки
5	Військова загроза, повітряна тривога	Соціальний	Переривання роботи, стрес
6	Надмірна тривалість робочого дня	Психологічний	Вигорання, зниження продуктивності
7	Порушення ергономіки	Ергономічний	Болі у спині, зап'ястях, шиї
8	Поганий мікроклімат (температура, вологість)	Фізичний	Дискомфорт, зниження працездатності
9	Використання несертифікованих пристроїв	Технічний	Ураження струмом, збій техніки
10	Відсутність психогігієнічної гігієни праці	Організаційний	Погіршення ментального здоров'я

В умовах війни до цього переліку додаються фактори ризику надзвичайних ситуацій – необхідність евакуації, перебої з енергопостачанням, нестабільність зв'язку, робота в укриттях[22].

4.3 Дослідження ризику реалізації потенційних небезпек на об'єкті проектування та розробка заходів щодо їх попередження

4.3.1 Теоретичні аспекти оцінки ризиків

Оцінка ризиків – це процес виявлення, аналізу та оцінювання небезпек на робочому місці з метою запобігання нещасним випадкам, захворюванням і втратам продуктивності. У сфері ІТ це стосується переважно довготривалих та кумулятивних ризиків[16][20][21].

Основні методи[23]:

- Метод матриці ризику – комбінує ймовірність та тяжкість наслідків;
- Дерево подій/відмов – логічна побудова сценаріїв реалізації небезпек;
- SWOT/HAZOP – у великих системах.

4.3.2 Матриця оцінки ризиків

Таблиця 4.2 – Робота за комп'ютером понад 8 год/день (ПРИКЛАД 1)

Критерій	Значення
Серйозність наслідків	С – серйозна (викликає хронічні захворювання)
Ймовірність	3 – висока
Рівень ризику	С3 – неприйнятний
Заходи	Робота 50/10 (50 хвилин + 10 хв відпочинку), вправи для ший/спини, зручне крісло, чергування задач

Таблиця 4.3 – Випромінювання монітора (ПРИКЛАД 2)

Критерій	Значення
Серйозність	В – помірна
Ймовірність	2 – середня
Рівень ризику	В2 – допустимий
Заходи	Сертифікована техніка, LCD-монітори, 50 см відстані, фільтри світла, режим "нічний екран"

4.3.3 Побудова дерева подій

Подія: Пожежа в домашньому офісі

Причини:

- Перевантаження розетки
- Старий блок живлення
- Відсутність захисного автомата

Наслідки:

- Втрата техніки
- Втрата даних
- Травмування

Профілактика:

- Джерело безперебійного живлення (UPS);
- Вогнегасник під рукою;
- Пожежна інструкція;
- Регулярна перевірка проводки.

4.3.4 Рекомендації щодо безпечної організації праці розробники

1. Ергономіка:

- Зручне крісло з регульованою висотою;
- Стіл на рівні ліктів;
- Монітор на відстані витягнутої руки.

2. Мікроклімат:
 - Температура: 21-24°C;
 - Вологість: 40-60%;
 - Провітрювання кожні 2-3 години.
3. Освітлення:
 - Природне та додаткове штучне освітлення з індексом кольору Ra > 80;
 - Уникати відбоисків і контрастів.
4. Організація часу:
 - Чіткий графік роботи;
 - Регулярні перерви;
 - Техніка Pomodoro.
5. Забезпечення безпеки:
 - Заземлення техніки;
 - Актуальне антивірусне ПЗ;
 - Використання ліцензованого програмного забезпечення.
6. Захист у разі війни:
 - Забезпечити можливість збереження роботи в офлайн-режимі;
 - Мати дублікати даних у хмарному сховищі;
 - Мати план дій при евакуації або аваріях.

Висновки до розділу

У цьому розділі було комплексно розглянуто законодавчі, технічні, гігієнічні та організаційні аспекти охорони праці, які мають ключове значення для фахівців, що працюють у сфері інформаційних технологій, зокрема – під час розробки вебдодатку з використанням технологій штучного інтелекту.

Було встановлено, що хоча розробка ПЗ не пов'язана з виробничими або механічними ризиками, вона супроводжується широким спектром прихованих небезпек – від ергономічних та фізіологічних до психологічних і техногенних. Особливу увагу було приділено дотриманню державних і міжнародних норм, таких як Закон України "Про охорону праці", ISO 45001, санітарні норми щодо мікроклімату, освітлення, електромагнітного випромінювання тощо. Аналіз нормативної бази дозволив визначити ключові вимоги до організації безпечного середовища праці в офісі або під час віддаленої роботи.

У процесі вивчення специфіки розробницької діяльності виявлено перелік найбільш поширених небезпек, які можуть впливати на фізичний та психоемоційний стан розробника: тривале перебування в статичному положенні, надмірне навантаження на зір, підвищений рівень стресу через дедлайни, неякісна організація робочого простору, а також ризики, пов'язані з порушенням пожежної безпеки чи нестабільністю електроживлення.

Проведена оцінка ризиків за допомогою матриці небезпек дозволила класифікувати небезпеки за рівнем критичності, що в свою чергу стало основою для формування практичних рекомендацій. Крім того, було продемонстровано побудову умовного "дерева подій", яке ілюструє ланцюг розвитку пожежонебезпечної ситуації в домашньому офісі – приклад, який набув особливої актуальності в умовах віддаленої роботи та нестабільного енергопостачання.

На підставі проведеного аналізу сформовано систему заходів щодо зниження професійних ризиків в ІТ-сфері, зокрема:

- Дотримання ергономічних норм при облаштуванні робочого місця (висота столу і крісла, положення монітора, підтримка підстави);
- Організація режиму праці та відпочинку (перерви, чергування діяльності, мікроперерви для очей);
- Забезпечення мікроклімату, вентиляції, освітлення відповідно до державних санітарних норм;

- Запровадження технічних заходів безпеки: використання сертифікованих пристроїв, джерел безперебійного живлення, антивірусного захисту, резервного копіювання;
- Обізнаність із планами евакуації, інструкціями з пожежної безпеки, наявність засобів гасіння загорянь;
- Збереження ментального здоров'я шляхом контролю навантаження, уникнення перевтоми, підтримки балансу між працею та відпочинком.

Наведені результати можуть бути використані як основа для створення інструкцій з охорони праці не лише у великих компаніях, але й у малих підприємствах ІТ-сектору, фриланс-командах, а також при індивідуальній проектній діяльності. В умовах сучасної диджиталізації та гібридного формату роботи важливо усвідомлювати, що охорона праці в ІТ – це не формальність, а необхідність, що прямо впливає на продуктивність, якість розробки, психічне здоров'я працівника та стабільність функціонування всієї системи.

Таким чином, реалізація навіть базових заходів безпеки, викладених у цьому розділі, дає змогу суттєво знизити вплив небезпечних та шкідливих факторів, забезпечуючи комфортні, стабільні та безпечні умови праці в ІТ-сфері.

ВИСНОВКИ

У межах виконання кваліфікаційної роботи було реалізовано повноцінний вебдодаток з функціональністю автоматизованої генерації текстового та візуального контенту на основі сучасних моделей штучного інтелекту (GPT-4 та DALL·E). Здійснено повний цикл проектування, розробки, реалізації та тестування системи, що дозволяє ефективно вирішувати завдання створення інформаційного наповнення для різноманітних користувацьких сценаріїв.

У ході дослідження:

- Проведено глибокий аналіз предметної області, виявлено актуальні технології та інструменти генерації контенту, визначено недоліки наявних аналогів і сформульовано обґрунтування необхідності розробки власного рішення.
- Розроблено архітектуру вебдодатку, що включає серверну й клієнтську частини, інтеграцію з зовнішніми API та базу даних для зберігання запитів і результатів. Запропонована структура забезпечує масштабованість, надійність та зручність розширення функціоналу.
- Реалізовано функціональність генерації текстів та зображень на основі запитів користувача. Забезпечено стабільну інтеграцію з OpenAI API, обробку винятків, а також збереження результатів з можливістю їх подальшого використання.
- Створено адаптивний інтерфейс користувача з урахуванням принципів UX/UI - дизайну, а також реалізовано мобільну версію додатку у вигляді прогресивного вебдодатку (PWA), що забезпечує доступність сервісу з будь-якого пристрою.
- Проведено комплексне тестування програмного забезпечення: функціональне, мобільне, безпекове та користувацьке, яке підтвердило

відповідність системи технічним вимогам і зручність взаємодії для кінцевих користувачів.

- У розділі з охорони праці проаналізовано нормативно-правову базу, виявлено потенційні небезпеки для розробників, проведено оцінку ризиків та надано обґрунтування рекомендації щодо покращення умов праці в ІТ-сфері.

У результаті виконання поставлених завдань створено функціональний, інтуїтивно зрозумілий вебдодаток із підтримкою мобільного доступу, що підтверджує як наукову, так і практичну цінність виконаної роботи. Розроблена система має високий потенціал для подальшого розширення функціональності, комерційного впровадження або адаптації під освітні й професійні потреби.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Artificial Intelligence Market – Global Forecast to 2027. MarketsandMarkets. URL:<https://www.marketsandmarkets.com/Market-Reports/artificial-intelligence-market-74851580.html>
2. ChatGPT – OpenAI. URL:<https://chat.openai.com/>
3. DALL·E – OpenAI. URL:<https://openai.com/dall-e>
4. Jasper – AI Content Platform. URL:<https://www.jasper.ai/>
5. Copy.ai – AI Copywriting Tool. URL:<https://www.copy.ai/>
6. Canva – Free Online Graphic Design Tool. URL:<https://www.canva.com/>
7. Artificial Intelligence Software Market Revenue Worldwide 2018–2025. Statista. URL:<https://www.statista.com/statistics/941835/worldwide-artificial-intelligence-software-market-revenue/>
8. Digital 2024: Global Overview Report – DataReportal – Global Digital Insights. URL:<https://datareportal.com/reports/digital-2024-global-overview-report>
9. Закон України "Про інформацію" від 02.10.1992 № 2657-XII. URL:<https://zakon.rada.gov.ua/laws/show/3792-12>
10. European Approach to Artificial Intelligence. European Commission. URL: <https://digital-strategy.ec.europa.eu/en/policies/european-approach-artificial-intelligence>
11. General Data Protection Regulation (GDPR). URL:<https://gdpr.eu/>
12. UNESCO Recommendation on the Ethics of Artificial Intelligence, 2021. URL:<https://unesdoc.unesco.org/ark:/48223/pf0000381137>
13. Санжаревська Т.С. Інформаційна система проактивного управління ресурсозабезпеченням міського господарства //Матеріали XVIII Всеукраїнської науково-технічної конференції «Сталий розвиток міст: поствоєнний період» (90-ї науково-технічної конференції ХНУМГ ім. О. М.

Бекетова) : в 5-и ч. / Ч. 2. – Харків : ХНУМГ ім. О. М. Бекетова, 2025. 297 с. – С.189 – 190. URL:

https://science.kname.edu.ua/images/dok/konferentsii/stalyirozvytok2019/2025/C.%202_Energeticna%20informacijna%20ta%20transportna%20infrastruktura_25.pdf

14. Закон України "Про охорону праці" від 14.10.1992 № 2694-XII. URL:<https://zakon.rada.gov.ua/laws/show/2694-12>

15. Кодекс законів про працю України (КЗпП). Офіційний текст. URL: <https://zakon.rada.gov.ua/laws/show/322-08>

16. ДСТУ ISO 45001:2019. Системи управління охороною праці. Вимоги та настанови щодо застосування.

17. ДСанПіН 3.3.2.007-98. Гігієнічні вимоги до персональних електронно-обчислювальних машин (ПЕОМ).

18. ДБН В.2.5-28:2018 – Природне і штучне освітлення

19. ДСН 3.3.6.042-99. Державні санітарні норми мікроклімату виробничих приміщень.

20. Бедрій Я.І. Основи охорони праці користувачів ПК. Навчальний посібник. – Львів: Видавництво ЛНУ, 2017. – 142 с.

21. Піскунова Л.Є. Безпека життєдіяльності. Навчальний посібник. – Харків: ФОП Панов А.М., 2020. – 160 с.

22. Кодекс цивільного захисту України від 02.10.2012 № 5403-VI. URL: <https://zakon.rada.gov.ua/laws/show/5403-17>

23. Малишева В.В. Методичні вказівки до виконання розділу з охорони праці у дипломних проєктах. – Харків: ХНУМГ ім. О.М. Бекетова, 2021. – 25 с.

24. AI Act. European Commission Digital Strategy. URL: <https://digital-strategy.ec.europa.eu/en/policies/artificial-intelligence-act>

ДОДАТКИ

ДОДАТОК А

Основна логіка app.py

/login

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = ''
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        user = User.query.filter_by(email=email, password=password).first()
        if user:
            session['user_id'] = user.id
            session['username'] = user.username
            return redirect(url_for('generate'))
        else:
            error = 'Невірні дані'
    return render_template('login.html', error=error)
```

/generate

```
@app.route('/generate', methods=['GET', 'POST'])
def generate():
    result = None
    if 'user_id' not in session:
        return redirect(url_for('login'))

    if request.method == 'POST':
        prompt = request.form['prompt']
        result = generate_text(prompt)
        new_entry = RequestHistory(
            user_id=session['user_id'],
            prompt=prompt,
            result=result
        )
        db.session.add(new_entry)
        db.session.commit()
    return render_template('generate.html', result=result)
```

/history

```
@app.route('/history')
def history():
    if 'user_id' not in session:
        return redirect(url_for('login'))
    records = RequestHistory.query.filter_by(user_id=session['user_id']).order_by(RequestHistory.created_at.desc()).all()
    return render_template('history.html', records=records)
```

ДОДАТОК Б

Генерація тексту api_orpenai.py

```
def generate_text(prompt):
    try:
        url = "https://api-inference.huggingface.co/models/google/flan-t5-large"
        headers = {
            "Authorization": f"Bearer {HF_API_KEY}"
        }
        payload = {
            "inputs": prompt
        }

        response = requests.post(url, headers=headers, json=payload)
        if response.status_code != 200:
            return f"Помилка: {response.status_code} - {response.text}"

        result = response.json()
        if isinstance(result, dict) and "error" in result:
            return f"Помилка: {result['error']}"

        return result[0]["generated_text"]
    except Exception as e:
        return f"Помилка генерації тексту: {e}"
```

ДОДАТОК В

HTML-шаблон

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <title>Генерація контенту</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
  <h1>Генерація контенту за допомогою ШІ</h1>

  <form method="POST">
    <label for="type">Оберіть тип генерації:</label><br>
    <select name="type" id="type">
      <option value="text">Текст</option>
      <option value="image">Зображення</option>
    </select><br><br>

    <label for="prompt">Введіть запит:</label><br>
    <textarea name="prompt" id="prompt" rows="5" cols="50" placeholder="Наприклад: A cat in a space suit on Mars, digital art">
    </textarea>

    <button type="submit">Згенерувати</button>
  </form>

  {% if result %}
    <h3>Результат:</h3>
    {% if result.endswith('.png') %}
      <br>
      <a href="{{ result }}" download>⬇ Завантажити зображення</a>
    {% else %}
      <pre>{{ result }}</pre>
    {% endif %}
  {% endif %}
</body>
</html>
```

ДОДАТОК Г

requirements.txt + .env.template

```
requirements.txt X
requirements.txt
1 flask
2 requests
3 python-dotenv
4 flask_sqlalchemy
5
```

```
$.env.template X
$.env.template
1 HF_API_KEY=hf_TwsuYf1VwhQSALrmWPRUekYIAGIanfpAA
2 REPLICATE_API_TOKEN=r8_UXp6qju6pu10sAfVQCgbgjH1DfhBuon1C995e
```