

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна графіка та Web-дизайн»

Група: 4КГ-08

Дипломний проект

здобувачки освіти денної форми навчання

КГ.08.14.000.ДП

***ЛЕВШАНОВОЇ
ВІКТОРІЇ АНДРІЇВНИ***

м. Одеса

2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 123 «Комп'ютерна інженерія»

Освітньо-професійна програма: «Комп'ютерна графіка та Web-дизайн»

Група: 4КГ-08

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту

Проектний матеріал складається з пояснювальної записки на 75 сторінках та графічного (перзентаційного) матеріалу на 15 аркушах (слайдах)

Дипломник _____ (Левшанова В. А.)

Керівник _____ (Шувалова І. В.)

Консультанти:

з економічного розділу _____ (Канський М. Ю.)

з розділу охорони праці та техніки безпеки _____ (Чорновол Н. І.)

з нормоконтролю _____ (Петрашова В. І.)

старший консультант _____ (Кривченко Ю. В.)

До захисту допущений

Голова циклової комісії _____ (Кривченко Ю. В.)

Завідувач відділення _____ (Краснокутська К. Г.)

Захист «30» серпня 2025 р.

Протокол ЕК № 8

Оцінка ЕК 4/20071 / 755

Секретар ЕК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення Ком'ютерних систем Комісія КТ та ІІ
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна графіка та Web-дизайн»

ЗАТВЕРДЖУЮ

Заст. дир. з НВР Беркань І. В.

« 19 » 08 2025 року

ЗАВДАННЯ

на дипломний проєкт (роботу)

Левшанової Вікторії Андріївни

1. Тема проєкту (роботи) Створення web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту

Затверджена наказом по коледжу від « 14 » 11 2024 р., наказ № 246

2 Термін здачі закінченого проєкту (роботи) _____

3. Вихідні дані до проєкту (роботи)

Вивчення можливостей API моделей, Vue 3; великі мовні моделі (LLM); OpenAi, Метод "create"; Мова програмування JavaScript; використанням платформи з відкритим кодом Nodejs, фреймворк для створення серверних додатків NestJS, фреймворк LangChainтехнологія Swagger; MongoDB.

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

Аналіз існуючих програмних рішень; Формування вимог до сервера;

Порядок розробки серверу; Проектування структури сервера; Розробка бази даних;

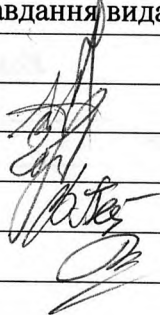
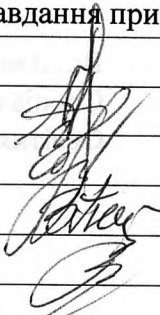
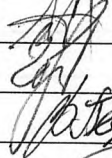
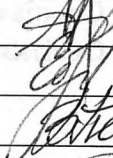
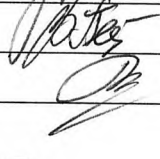
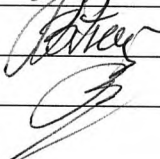
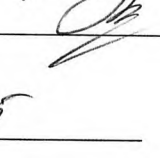
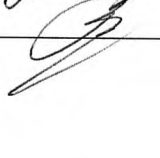
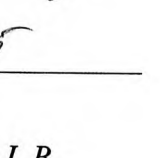

Тестування створеного програмного продукту; Економічний розділ;

Охорона праці та техніка безпеки; Висновки до ДП.

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

Презентація Power Point – 15 слайдів: Титульний слайд; Взаємозв'язк між frontend і backend-розробкою; Скріншот структури проєкту; Скріншот коду файлів server.js; Скріншот сруктури сервера; Скріншот функції взаємодії з API Anthropic; Скріншот вікна програми; Скріншоти роботи програми; Результати тестування.


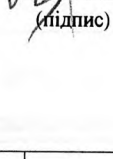
6. Консультанти по проекту, із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Шувалова І. В.		
Економічний розділ	Канський М.Ю.		
Розділ охорони праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кривченко Ю. В.		

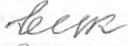
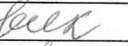
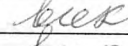
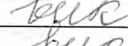
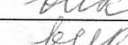
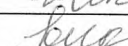
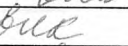
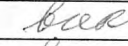
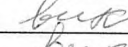
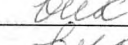
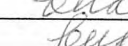
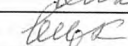
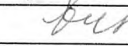


7. Дата видачі завдання 29.04.25

Керівник Шувалова І. В.

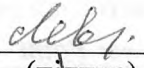
Завдання прийняв до виконання Левшанова В.А.


(підпис)

(підпис)


КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Вступ. Постановка мети та задач проектування	29.04.25	
2	Розробка структури програми	01.05.25	
3	Обґрунтування вибору розробки програми	03.05.25	
4	Адаптація та налаштування моделей	05.05.25	
5	Налаштування структури проекту	07.05.25	
6	Розробка бекенд та фронтенд частини	09.05.25	
7	Тестування створеного програмного продукту	15.05.25	
8	Економічний розрахунок	17.05.25	
9	Опис охорони праці та техніки безпеки	19.05.25	
10	Аналіз результатів проектування.	21.05.25	
11	Оформлення пояснювальної записки	23.05.25	
12	Оформлення графічної (презентаційної)	30.05.25	
13	частини	05.06.25	
14	Підготовка доповіді для захисту	12.06.25	
15	Малий захист дипломного проекту	16.06.25	

Дипломник


(підпис)

Керівник


(підпис)

ЗМІСТ

Вступ.....	7
1 Основний розділ.....	8
1.1 Процес розробки програми для генерації повідомлень за допомогою штучного інтелекту.....	8
1.2 Аналіз ринку.....	11
1.3 Формулювання технічних вимог.....	14
1.4 Проектування технічної частини програми «Інтелектуальний помічник».....	16
1.4.1 Порівняння провідних мовних моделей (LLM).....	16
1.4.2 Вибір моделі.....	18
1.4.3 API для створення повідомлень у Claude 3 Opus від Anthropic.....	19
1.4.4 Опис потокового API для створення повідомлень у Claude 3 Opus від Anthropic.....	21
1.5 Опис функціональності програми «Інтелектуальний помічник».....	23
1.6 Технології, які використовуються в розробці програми.....	24
1.6.1 Технології, використані для бекенд розробки.....	25
1.6.2 Опис технології розробки фронтенд частини.....	27
1.7 Програмна реалізація програми «Інтелектуальний помічник».....	30
1.7.1 Розробка Фронтенду.....	30
1.7.2 Налаштування проекту.....	32
1.8 Опис роботи програми.....	48
2 Економічний розділ.....	55
2.1 Резюме.....	55
2.2. Визначення трудомісткості розробки програмного забезпечення.....	56
2.3 Розрахунок ціни програмного продукту.....	57
3 Розділ охорона праці та техніки безпеки.....	60
3.1 Аналіз та безпека умов праці працівника на робочому місці.....	61
3.2 Розробка заходів з охорони праці.....	62

3.3 Організація робочого місця користувача ПК.....	63
3.4 Пожежна безпека.....	64
Висновки.....	65
Перелік використаних інформаційних джерел.....	66
Додаток А. Програмний для Vue + FontAwesome + Social Sharing.....	67
Додаток В. Слайди мультимедійної презентації.....	68

ВСТУП

Штучний інтелект (ШІ) - це наука і технологія, що займаються створенням інтелектуальних машин і комп'ютерних програм. Ці системи прагнуть розуміти та моделювати людський інтелект, проте не обмежуються біологічно відтворюваними методами.

Штучний інтелект (ШІ) - це галузь інформатики, яка займається розробкою інтелектуальних машин, здатних виконувати завдання, які зазвичай потребують людського інтелекту. Системи штучного інтелекту створені для навчання на досвіді, розпізнавання закономірностей і ухвалення рішень на основі вхідних даних.

Класифікація типів штучного інтелекту (станом на 2024 рік).

1. Артифіційний суперінтелект (Artificial Super Intelligence, ASI)

Гіпотетичний штучний інтелект, який не тільки здатен відтворити найвищі здібності людини, але й перевершити їх. Прихильники ASI вважають, що він зможе розуміти думки та почуття людини з метою підкорення її волі.

2. Штучний загальний інтелект (Artificial General Intelligence, AGI)

Гіпотетичний інтелект, який знаходиться на рівні розуму нижчому за ASI. AGI може виконувати ті самі дії, що і людина.

3. Слабий, або обмежений інтелект (Artificial Narrow Intelligence, ANI)

Інтелект, який демонструє слабкі натяки на розум і призначений лише для виконання вузького кола завдань.

ANI не може діяти незалежно від людини чи розвиватися самостійно.

Інтелект - це психологічна якість, що включає здатність адаптуватися до нових ситуацій, вчитися на основі досвіду, розуміти і застосовувати абстрактні концепції, а також використовувати знання для управління навколишнім середовищем. Інтелект об'єднує всі пізнавальні здібності людини, такі як сприйняття, пам'ять, мислення та уяву.

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		7

1 ОСНОВНИЙ РОЗДІЛ

Створення програми для генерації повідомлень за допомогою штучного інтелекту включає кілька ключових етапів і компонентів. Основна мета полягає у розробці алгоритмів та моделей, здатних генерувати текстові повідомлення та зображення, які максимально імітують людський творчий процес.

У 1980-х роках вчені Барр та Файгенбаум, що працювали в галузі теорії обчислень, запропонували таке визначення штучного інтелекту (ШІ):

"Штучний інтелект - це галузь інформатики, яка займається створенням інтелектуальних комп'ютерних систем, здатних виконувати завдання, що традиційно асоціюються з людським розумом, такі як розуміння мови, навчання, міркування, та вирішення проблем."

З часом до штучного інтелекту стали відносити різноманітні алгоритми та програмні системи, що здатні вирішувати завдання так, як це робить людина, використовуючи свої розумові процеси. Основні властивості штучного інтелекту включають розуміння мови, здатність навчатися, а також мислити та діяти.

Штучний інтелект - це комплекс взаємопов'язаних технологій і процесів, які розвиваються стрімко та якісно. До таких технологій належать:

Обробка тексту природною мовою: включає розпізнавання та розуміння текстової інформації, аналіз синтаксису і семантики мови, а також генерацію тексту з урахуванням контексту.

Машинне навчання: галузь штучного інтелекту, яка вивчає алгоритми і моделі, що дозволяють комп'ютерам автоматично навчатися на основі даних і покращувати свою продуктивність з часом.

Експертні системи: програмні системи, що моделюють експертний досвід і знання в певній області, дозволяючи робити висновки та рекомендації у відповідних ситуаціях.

Віртуальні агенти (чат-боти та віртуальні помічники): програмні агенти, що виконують завдання комунікації з користувачами, використовуючи природну

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		8

мову, відповідаючи на запитання, надаючи інформацію та виконуючи певні дії.

Системи рекомендацій: системи, які аналізують інформацію про користувача та його поведінку, і надають рекомендації щодо продуктів, послуг, контенту або дій, які можуть бути цікавими або корисними для користувача.

Ці технології швидко розвиваються і дозволяють створювати інтелектуальні системи, здатні наблизитися до можливостей людського розуму.

1.1 Процес розробки програми для генерації повідомлень за допомогою штучного інтелекту

Перед початком проектування необхідно визначення вимоги та проаналізувати ринок подібних програм. По-перше визначаємо цільову аудиторію програми для генерації повідомлень за допомогою штучного інтелекту

1. Професійні користувачі:

Маркетологи: Використовуватимуть програму для створення рекламних матеріалів, повідомлень для соціальних медіа, та контенту для веб-сайтів. Інструмент допоможе автоматизувати створення креативних текстів, зменшуючи витрати часу та зусиль.

Дизайнери: Програма стане в нагоді для швидкого генерування ідей та макетів. Дизайнери можуть використовувати створені зображення як натхнення для своїх проектів або безпосередньо в роботі.

Контент-менеджери: Для регулярного наповнення сайтів, блогів, новинних порталів якісним контентом.

Програма допоможе автоматизувати частину робочого процесу, забезпечуючи свіжий та релевантний контент.

Журналісти та копірайтери: Можуть використовувати інструмент для генерації ідей, надання основного тексту для статей або створення варіантів заголовків та підзаголовків.

2. Освітній сектор:

Студенти та викладачі: Для підготовки навчальних матеріалів, презентацій,

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		9

наукових статей та проектів. Програма допоможе генерувати ідеї, формулювати тези та створювати візуальний супровід до навчальних матеріалів.

Онлайн-курси та платформи: Можуть використовувати програму для створення навчального контенту, завдань та візуалізацій, що сприяє покращенню навчального процесу та залученню студентів.

3. Малі та середні підприємства:

Підприємці та стартапи: Можуть використовувати програму для створення маркетингових матеріалів, презентацій для інвесторів, соціальних медіа контенту та іншої рекламної продукції. Інструмент допоможе зекономити ресурси та швидше реалізувати ідеї.

Інтернет-магазини: Для автоматизації створення описів продуктів, рекламних банерів та контенту для соціальних мереж, що сприяє підвищенню ефективності маркетингових кампаній.

4. Розважальна індустрія:

Блогери та інфлюенсери: Використовуватимуть програму для створення унікального контенту, включаючи тексти для постів, зображення для соціальних медіа та відео, що допоможе залучати та утримувати аудиторію.

5. Інші потенційні користувачі: Автори та письменники: Для генерації ідей, написання чернеток, створення сюжетів та діалогів. Програма може стати корисним інструментом для натхнення та допомоги у творчому процесі.

Фахівці з HR: Для автоматизації створення текстів вакансій, внутрішніх комунікацій та мотиваційних листів, що сприяє покращенню ефективності роботи з персоналом.

Визначення цільової аудиторії дозволяє зрозуміти, які функції та можливості програми є найбільш важливими для користувачів, що сприяє розробці продукту, який максимально відповідає потребам різних груп споживачів.

Виявлення основних функціональних вимог: створення, редагування та відправлення повідомлень. Збір нефункціональних вимог: швидкість роботи, зручність використання, безпека даних.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		10

1.2 Аналіз ринку

Перед тим, як я почав проектувати програму, необхідно було дослідити існуючих аналогів програм для генерації повідомлень, визначити у конкурентних програм переваги і недоліки, а також розглянути та вивчити користувацькі відгуки для покращення майбутнього продукту.

1. Розмір ринку та зростання:

Загальний ринок: Ринок програм на базі штучного інтелекту (ШІ) постійно зростає, зокрема сегмент програм для генерації тексту та зображень. За останніми даними, очікується значне зростання цього ринку протягом найближчих років завдяки збільшенню попиту на автоматизацію процесів створення контенту.

Темпи зростання: За оцінками аналітиків, середньорічний темп зростання (CAGR) для програм, що використовують ШІ для генерації контенту, складає близько 25-30%.

2. Ключові гравці на ринку:

На ринку штучного інтелекту, особливо в сегменті великих мовних моделей (LLM), виділяються кілька провідних компаній, кожна з яких має свої унікальні розробки:

1. OpenAI

GPT-4: Останнє покоління їхньої мовної моделі, яке здатне генерувати текст, розв'язувати складні завдання, створювати зміст, перекладати текст, а також здійснювати обговорення різних тем. Модель показує вражаючі результати у багатьох сферах використання.

2. Google AI (DeepMind)

PaLM: асштабована мовна модель, що використовує методи зниження затрат на навчання, що дозволяє їй виконувати завдання з розуміння мови і логічного висновкування на вищому рівні.

3. Anthropic

Claude: Мовна модель, яка розроблена з акцентом на безпеку та відповідність людським етичним стандартам. Вона створена для виконання завдань з генерації

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		11

тексту з урахуванням нюансів людського спілкування.

4. Facebook AI (Meta AI)

LLaMA: Нова мовна модель, яка має на меті оптимізувати виконання завдань на різних мовах і забезпечувати високу ефективність обробки мовних даних на менш потужних комп'ютерних системах.

5. Microsoft

Azure AI: Інтеграція їх LLM у хмарні сервіси, що забезпечує потужні можливості для розробників та підприємств у створенні і впровадженні мовних рішень.

Ці компанії продовжують розвивати свої технології та пропонувати рішення, які формують майбутнє штучного інтелекту. Останні розробки великих мовних моделей відіграють ключову роль у цьому процесі, надаючи інструменти для рішення складних завдань від автоматизації до нових способів взаємодії між людиною і машиною.

Генерація зображень: Програми, що створюють або редагують зображення, включаючи фотографії, ілюстрації, графіку.

Комбіновані рішення: Програми, які можуть генерувати як текст, так і зображення.

Цільові аудиторії:

Підприємства (B2B): Маркетингові агенції, медіакомпанії, інтернет-магазини.

Індивідуальні користувачі (B2C): Блогери, письменники, студенти, дизайнери.

4. Попит та тенденції:

Зростаючий попит на автоматизацію: Більше компаній шукають способи автоматизувати створення контенту для підвищення продуктивності та скорочення витрат.

Покращення якості контенту: Програми стають дедалі досконалішими, що дозволяє створювати більш якісний і реалістичний контент.

Інтеграція з іншими інструментами: Багато програм інтегруються з іншими

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		12

системами, такими як платформи управління контентом (CMS), CRM-системи, маркетингові платформи.

5. Виклики та можливості:

Конкуренція: Висока конкуренція серед постачальників ШІ-рішень.

Етичні питання: Питання авторських прав, маніпуляція контентом, етичні наслідки використання ШІ.

Технічні обмеження: Якість і точність генерації контенту можуть варіюватися. Можливості: Розширення функціональності: Додавання нових функцій, таких як голосове управління, інтерактивні можливості.

Ринки, що розвиваються: Зростаючі ринки в Азії, Африці та Латинській Америці пропонують значні можливості для зростання.

Співпраця з великими гравцями: Партнерства з відомими компаніями можуть сприяти швидшому впровадженню та прийняттю технологій.

Із вивченого матеріалу можемо зробити висновок, що ринок програм для генерації повідомлень за допомогою ШІ є перспективним та швидкозростаючим сегментом, з великим потенціалом для інновацій та розширення. Висока конкуренція стимулює постійне вдосконалення продуктів та послуг, що надаються, а зростаючий попит на автоматизацію створення контенту вказує на значні можливості для розвитку в цьому напрямку.

1.3. Формулювання технічних вимог

Сформулюємо технічні вимоги до програми для генерації повідомлень і зображень за допомогою штучного інтелекту:

1. Загальні вимоги:

Платформа: Підтримка Windows, macOS, Linux.

Інтерфейс користувача: Інтуїтивно зрозумілий, зручний інтерфейс (GUI).

Мови програмування: JavaScript (для front-end і back-end).

Архітектура: Клієнт-серверна архітектура з можливістю масштабування.

2. Функціональні вимоги:

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		13

Генерація текстових повідомлень:

Використання моделей нейронних мереж для генерації тексту (наприклад, GPT-3).

Підтримка різних мов (англійська, українська, японська та інші популярні мови).

Можливість вибору стилю письма (формальний, неформальний, технічний, креативний тощо).

Врахування контексту та тематики для релевантної генерації тексту.

Інтеграція з системами перевірки граматики та стилю (наприклад, Grammarly).

3. Нефункціональні вимоги:

Продуктивність:

Швидкий час відгуку при генерації тексту (не більше 5 секунд для тексту)

Оптимізація використання ресурсів системи (CPU, GPU, RAM).

Безпека:

Захист даних користувачів та конфіденційної інформації.

Використання шифрування для зберігання та передачі даних.

Захист від SQL-ін'єкцій, XSS-атак та інших загроз.

Масштабованість:

Підтримка горизонтального та вертикального масштабування для обробки великих обсягів запитів.

Використання хмарних рішень для зберігання та обробки даних (наприклад, AWS, Google Cloud).

Сумісність:

Інтеграція з популярними системами управління контентом (CMS), CRM-системами та іншими бізнес-інструментами.

Підтримка API для доступу до функціональності програми іншими системами.

Тестування програми під високими навантаженнями для перевірки її стабільності.

Користувацьке тестування: Тестування інтерфейсу користувача з метою забезпечення його зручності та інтуїтивності.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		14

5. Документація:

Технічна документація: Докладні описи архітектури програми, алгоритмів та моделей, що використовуються.

Користувацька документація: Інструкції для користувачів щодо встановлення, налаштування та використання програми.

Вибір правильного програмного забезпечення для проектування штучного інтелекту може бути складним завданням, враховуючи безліч доступних варіантів.

1.4 Проектування технічної частини програми «Інтелектуальний помічник»

1.4.1 Порівняння провідних мовних моделей (LLM)

У сфері штучного інтелекту спостерігається значне зростання, і багато компаній, включаючи відомі імена, такі як OpenAI, Google, Anthropic та Meta, пропонують свої мовні моделі (LLM). Кожна з цих компаній розробила унікальні рішення, які мають свої сильні та слабкі сторони. У цьому порівнянні розглядаються ключові характеристики та особливості мовних моделей цих провідних компаній, що дозволить краще зрозуміти, які з них найбільше підходять для різних завдань та потреб.

Порівняння провідних компаній у сфері штучного інтелекту та їхніх мовних моделей (LLM)

1. OpenAI. Модель GPT-4o.

Сильні сторони:

Найсучасніші результати в розумінні та генерації природної мови.

Відмінно генерує текст, схожий на людський, та підтримує контекст у довгих розмовах. Широко використовується у різних додатках, від чат-ботів до створення контенту.

Слабкі сторони:

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		15

Вимагає значних обчислювальних ресурсів.

Висока вартість використання передових версій.

2. Google. Модель Gemini-1.5-Pro.

Сильні сторони:

Сильна в багатомовному розумінні та генерації коду.

Передові можливості в логічному мисленні та перекладі.

Використовує інфраструктуру та дані Google.

Слабкі сторони:

Проблеми з конфіденційністю через практики обробки даних.

Менша доступність для публіки в порівнянні з моделями OpenAI.

3. Anthropic. Модель Claude 3 Opus.

Сильні сторони:

Орієнтована на безпеку та зрозумілість.

Створена з багатьма запобіжними заходами для забезпечення етичного використання та мінімізації шкідливих результатів.

Ефективна в додатках, що потребують надійних та безпечних відповідей.

Слабкі сторони:

Може відставати у продуктивності від моделей OpenAI та Google.

Новий гравець на ринку з менш широким використанням.

4. Meta. Модель Llama-3.

Сильні сторони:

Відкрите джерело, що сприяє прозорості та внескам спільноти.

Добрий баланс між продуктивністю та ефективністю.

Гнучкість для дослідників та розробників для тонкої настройки та розгортання.

Слабкі сторони:

Занепокоєння щодо неправильного використання моделей з відкритим кодом.

Може не відповідати продуктивності власницьких моделей у певних завданнях.

Розглянувши існуючі можливості, можна зробити такий підсумок:

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		16

GPT-4o від OpenAI відомий своїми провідними результатами у завданнях природної мови, але потребує високих витрат та ресурсів.

Gemini-1.5-Pro від Google використовує інфраструктуру Google для потужних можливостей багатомовного та логічного мислення, хоча є деякі проблеми з конфіденційністю.

Claude 3 Opus від Anthropic акцентує увагу на безпеці та етичних аспектах, що робить його придатним для додатків, де потрібна надійність.

Llama-3 від Meta надає альтернативу з відкритим джерелом, сприяючи прозорості та залученню спільноти, хоча існують ризики неправильного використання.

Ці компанії та моделі демонструють різні сильні сторони та сфери застосування залежно від конкретних потреб та пріоритетів користувачів.

1.4.2 Вибір моделі

Вибір моделі Claude 3 Opus від Anthropic для цього проекту обґрунтовується трьома відмінними характеристиками, які роблять її надзвичайно привабливою для використання:

1. Безпека та етичні аспекти

Claude 3 Opus розроблена з акцентом на безпеку та етичність, що робить її ідеальною для додатків, де надійність і відповідність етичним стандартам є критичними. Anthropic зосереджується на мінімізації шкідливих результатів та запобіганні використанню моделі в неправильних цілях, що важливо для проектів з підвищеними вимогами до безпеки.

2. Висока продуктивність

Згідно з рейтингом LMSys Chatbot Arena Leaderboard, Claude 3 Opus займає високу позицію з оцінкою 1248. Це свідчить про її здатність ефективно виконувати завдання природної мови на високому рівні, що забезпечує відмінні результати в реальних застосуваннях.

3. Широке застосування та надійність:

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		17

Claude 3 Opus демонструє відмінні результати в додатках, що потребують надійних та безпечних відповідей. Вона підходить для різноманітних завдань, від обробки природної мови до генерації тексту, що робить її універсальним інструментом для розробників. Вибір Claude 3 Opus дозволить скористатися перевагами безпечної, продуктивної та надійної моделі, яка відповідає сучасним вимогам до етичного використання штучного інтелекту.

1.4.3 API для створення повідомлень у Claude 3 Opus від Anthropic

Створення повідомлення: API для створення повідомлень дозволяє надсилати структурований список вхідних повідомлень з текстовим та/або зображувальним вмістом, на основі яких модель генерує наступне повідомлення в розмові.

Використання API

URL: <https://api.anthropic.com/v1/messages>

Метод: POST

Параметри запиту:

model (string): Вибір моделі, наприклад, "claude-3-opus-20240229".

messages (array): Список вхідних повідомлень, кожне з яких має роль (user або assistant) та вміст (content).

max_tokens (integer): Максимальна кількість токенів для генерації.

metadata (object): Метадані запиту (опціонально).

stop_sequences (array): Текстові послідовності, що зупиняють генерацію (опціонально).

stream (boolean): використовувати потокову передачу відповідей (опціонально).

system (string): Системний промт для встановлення контексту (опціонально).

temperature (number): Рівень випадковості у відповідях (опціонально).

tools (array): Визначення інструментів, які може використовувати модель (опціонально).

Код запиту на JavaScript:

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		18

```
import Anthropic from '@anthropic-ai/sdk';

// Ініціалізація клієнта
const anthropic = new Anthropic();

// Функція для відправлення повідомлення та обробки відповіді
async function sendMessage() {
  try {
    const response = await anthropic.messages.create({
      model: "claude-3-opus-20240229",
      max_tokens: 1024,
      messages: [
        {
          role: "user",
          content: "Hello, world"
        }
      ]
    });

    // Вивід отриманої відповіді
    console.log("Відповідь асистента:", response.content[0].text);
    console.log("Використано токенів:", response.usage);
  } catch (error) {
    console.error("Помилка при виклику API:", error);
  }
}

// Викликаємо функцію
```

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		19

sendMessage();

Основні можливості:

Підтримка тексту та зображень: API підтримує як текстовий, так і зображувальний вміст.

Гнучкість у параметрах: Можливість встановлення різних параметрів, таких як максимальна кількість токенів, температури генерації та використання інструментів.

Підтримка потокової передачі: Можливість отримання відповідей у режимі реального часу за допомогою серверних подій

1.4.4 Опис потокового API для створення повідомлень у Claude 3 Opus від Anthropic

Потокове API дозволяє надсилати запити та отримувати відповіді поступово, використовуючи події, що надсилаються сервером (SSE). Це особливо корисно, коли потрібна миттєва взаємодія з користувачем, наприклад, у чат-ботах або інтерактивних додатках. Ось деякі важливі деталі про цю функціональність:

Переваги потокового API

1. Швидка взаємодія: Потокове API забезпечує миттєвий відгук, що покращує користувацький досвід.
2. Гнучкість: Ви отримуєте відповіді частинами, що дозволяє обробляти та відображати дані в режимі реального часу.

3.Збереження з'єднання: Використання SSE забезпечує стабільне з'єднання між клієнтом і сервером, що знижує затримки.

Розглянемо як це працює:

Запит: Ви надсилаєте запит на створення повідомлення з параметром "stream": true.

1. Події: Сервер надсилає події типу message_start, content_block_start, content_block_delta, content_block_stop, message_delta, message_stop та ping.

2. Обробка: Клієнт обробляє кожну подію в режимі реального часу, що дозволяє відразу відображати частини відповіді.

Використання API

URL: <https://api.anthropic.com/v1/messages>

Метод: POST

Параметри запиту:

"model": вибір моделі, наприклад, "claude-3-opus-20240229".

"messages": список вхідних повідомлень з ролями (user або assistant) та вмістом (content).

"max_tokens": максимальна кількість токенів.

"stream": встановити true для потокової передачі відповідей.

Код реалізації на JavaScript:

```
import Anthropic from '@anthropic-ai/sdk';  
  
// Ініціалізація клієнта  
  
const client = new Anthropic();  
  
// Асинхронна функція для потокового надсилання повідомлення  
  
async function streamMessage() {  
  
  try {  
  
    const stream = await client.messages.stream({  
  
      messages: [  
  
        {  
  
          role: 'user',  
  
          content: 'Hello'  
  
        },  
  
      ],  
  
      model: 'claude-3-opus-20240229',  
  
      max_tokens: 1024  
  
    });
```

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		21

```
// Обробка події 'text' у потоці
stream.on('text', (text) => {
  console.log('Отриманий текст:', text);
});

// Обробка завершення потоку
stream.on('end', () => {
  console.log('Потік завершено');
});

// Обробка помилок потоку
stream.on('error', (err) => {
  console.error('Помилка під час отримання потоку:', err);
});

} catch (error) {
  console.error('Помилка виклику API:', error);
}
}

// Викликаємо функцію
streamMessage();
```

Типи подій:

- message_start: початок повідомлення.
- content_block_start, content_block_delta, content_block_stop: блоки вмісту.
- message_delta: зміни у повідомленні.
- message_stop: кінець повідомлення.
- ping: підтримка з'єднання.
- error: повідомлення про помилки.

Іноді сервер може надсилати помилки у потоковому режимі. Наприклад, під час високого навантаження ви можете отримати помилку `overloaded_error`, що відповідає HTTP 529 у звичайному контексті.

Приклад помилки:

```
event: error
```

```
data: {"type": "error", "error": {"type": "overloaded_error", "message": "Overloaded"}}
```

Ця інформація вказує, що сервер перевантажений і не може обробити запит. Для детальнішої інформації зверніться до документації Anthropic.

Для розробників, які прагнуть створювати інтерактивні додатки, де важлива швидкість та ефективність, потокове API від Anthropic стане ідеальним рішенням. Це схоже на динамічну адаптацію під час участі у проектних зустрічах або інтерактивних сесіях з командою, де миттєвий зворотний зв'язок є ключовим фактором успіху.

1.5 Опис функціональності програми «Інтелектуальний помічник»

Інтелектуальний помічник - це інструмент для створення повідомлень на всі випадки життя. Інтерфейс програми дозволяє користувачеві налаштувати наступні параметри:

Мова: Вибір мови повідомлення, українська, англійська та інші).

Тип повідомлення: Вибір категорії повідомлення, такого як, оголошення, сповіщення тощо.

Стиль повідомлення: Вибір стилю повідомлення (формальний або неформальний).

Тип виведення: Формат виведення повідомлення (текст або інше).

Контекст події: Поле для опису ситуації або контексту, на основі якого буде створено повідомлення.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		23

Після заповнення всіх параметрів, користувач натискає кнопку "Показати результат", щоб отримати згенероване повідомлення, яке можна переглянути та за потреби редагувати.

1.6 Технології, які використовуються в розробці програми

Я вирішила розділити бекенд і фронтенд (frontend and backend) частини на два окремі репозиторії, щоб зручніше було розділити відповідальність між ними. Весь код писатиму на JavaScript, обравши Node.js для бекенду та Vue 3 для фронтону. Це відмінне поєднання для взаємодії через API. Для опису ендпоінтів зручно використовувати Swagger, що спрощує роботу з документацією інтерфейсів.

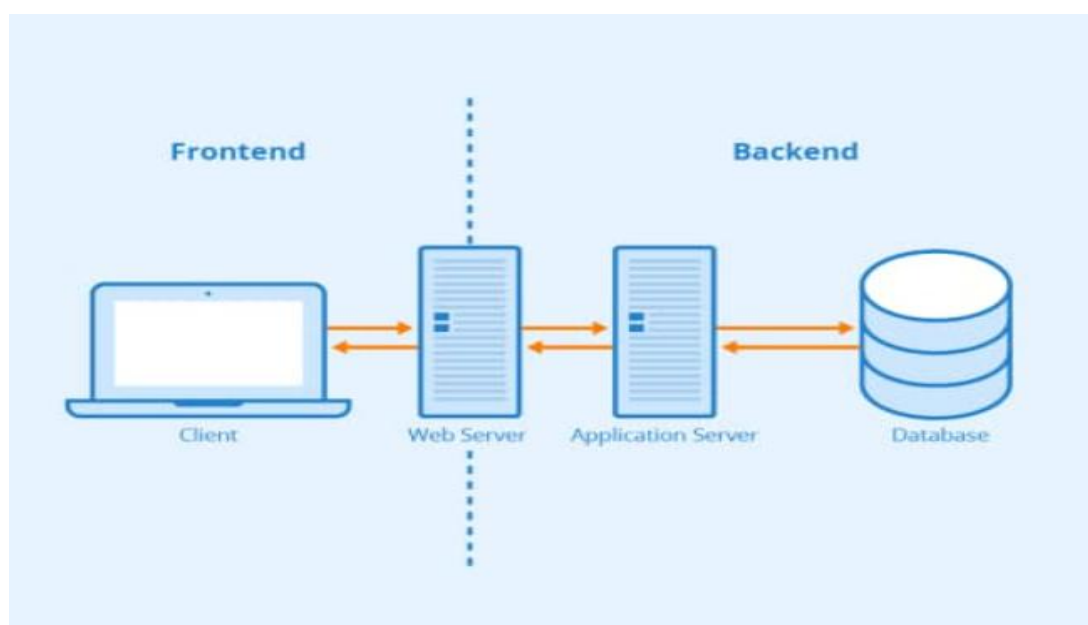


Рисунок 1.1. Взаємозв'язк між frontend і backend-розробкою

Бекенд- і фронтенд-розробники взаємодіють, щоб створити повноцінний веб-додаток. Бекенд-розробник надає API, який визначає доступні функції та методи взаємодії з сервером. Фронтенд-розробник використовує ці API для доступу до сервера та отримання необхідних даних і ресурсів.

1.6.1 Технології, використані для бекенд розробки

Node.js - це відкритий серверний середовище для виконання JavaScript, яке побудоване на двигуні V8 від Google Chrome. Це дозволяє використовувати JavaScript не тільки на клієнтській стороні (в браузері), але й на сервері.

Особливості Node.js:

Асинхронне та неблокуюче введення/виведення (I/O): Це означає, що Node.js може обробляти багато запитів одночасно без блокування основного потоку, що покращує продуктивність та швидкодію.

Подієво-орієнтована модель: Використовується для управління синхронними операціями. Це дозволяє створювати високопродуктивні та масштабовані мережеві програми.

npm (Node Package Manager): Це потужний менеджер пакетів, що дозволяє встановлювати та управляти бібліотеками JavaScript. Він містить величезний каталог пакунків, які можна інтегрувати у свій проект для різних функціональних можливостей.

Express.

Express - це мінімалістичний та гнучкий веб-фреймворк для Node.js, який робить розробку веб-додатків швидкою та організованою.

Особливості Express:

Маршрутизація: Проста та інтуїтивно зрозуміла система маршрутів дозволяє визначати, як ваш додаток відповідає на різні HTTP запити.

Middleware: Це функції, які мають доступ до об'єкту запиту (req), об'єкту відповіді (res) та наступної функції обробки. Middleware можуть виконувати завдання, такі як перевірка користувача, логування та обробка запитів.

Шаблонізація: Підтримує різні двигуни шаблонів, що дозволяють генерувати HTML на сервері перед відправкою на клієнт.

Visual Studio Code (VSCode)

Visual Studio Code - це безкоштовний редактор коду від Microsoft, який підтримує багато мов програмування та має велику кількість розширень.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		25

Особливості VSCode: Підтримка багатьох мов: Включає JavaScript, TypeScript.

Розширення: Надає можливість налаштування редактора під конкретні потреби за допомогою розширень для підсвічування синтаксису, форматування коду, інтеграції з Git та іншими інструментами.

Інтегрований термінал: Дозволяє виконувати команди безпосередньо з редактора.

Інтелектуальне завершення коду: Використовує IntelliSense для автоматичного завершення коду.

Git - це система контролю версій, яка допомагає ефективно керувати змінами в коді. Вона була створена Лінусом Торвальдсом.

Особливості Git:

Розподілена архітектура: Кожен розробник має свою локальну копію репозиторію, що дозволяє працювати незалежно.

Гнучкість у використанні гілок: Легко створювати, зливати та переключатися між гілками коду.

Журнал змін: Детальний журнал змін дозволяє відслідковувати, хто зробив зміни, коли вони були зроблені, і що саме було змінено.

Захист від втрати даних: Завдяки розподіленій архітектурі дозволяє відновити втрачені дані з інших копій репозиторію.

Інтеграція з робочими процесами: Може бути інтегрована з системами стеження за помилками, неперервною інтеграцією/доставкою (CI/CD) та іншими інструментами.

NPM (Node Package Manager)

NPM - це пакетний менеджер для Node.js, що використовується для керування залежностями проекту.

Команди NPM:

`npm install`: Встановлює всі залежності, вказані у файлі `package.json`.

`npm install <package-name>`: Встановлює конкретний пакет.

`npm uninstall <package-name>`: Видаляє пакет з проекту.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		26

`npm update`: Оновлює всі пакети до останніх версій.

`npm run <script-name>`: Виконує скрипт, вказаний у `package.json`.

Swagger

Swagger - це інструмент для створення, документування та тестування RESTful API.

Особливості Swagger:

Автоматична генерація документації: Swagger дозволяє автоматично генерувати документацію для вашого API на основі ваших кодових коментарів та анотацій.

Інтерактивна документація: Згенерована документація включає інтерактивні елементи, що дозволяють тестувати API прямо з документації.

Підтримка різних мов: Swagger підтримує різні мови програмування та фреймворки.

Інтеграція з CI/CD: Swagger може бути інтегрований з системами неперервної інтеграції та доставки (CI/CD) для автоматичного оновлення документації.

Node.js та Express забезпечують потужний стек технологій для розробки серверної частини веб-додатків, надаючи можливість виконувати JavaScript на сервері та використовуючи корисні функції для спрощення розробки веб-додатків. VSCode та Git створюють ефективне середовище для розробки програмного забезпечення, яке дозволяє легко управляти кодом та розробляти проекти. NPM спрощує керування залежностями проекту, надаючи доступ до численних бібліотек та інструментів. Swagger дозволяє створювати, документувати та тестувати API, забезпечуючи зручність та ефективність у роботі з RESTful API.

1.6.2 Опис технології розробки фронтенд частини

Фронтенд (Frontend) - це публічна частина веб-додатку, з якою користувач безпосередньо взаємодіє. До фронтенду належить відображення функціональних елементів користувацького інтерфейсу, обробка дій користувача та виконання

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		27

відповідних операцій на стороні клієнта. Простими словами, фронтенд - це все, що бачить і з чим взаємодіє користувач при відкритті веб-сторінки.

Веб-додаток являє собою клієнт-серверну систему, де клієнтом зазвичай виступає браузер користувача, а сервером - веб-сервер. Логіка роботи веб-додатку розподіляється між клієнтською та серверною сторонами: обробка даних і зберігання інформації здійснюється переважно на сервері, тоді як клієнтська сторона відповідає за відображення даних та обробку взаємодії користувача.

Інформаційний обмін між клієнтом і сервером відбувається через мережу Інтернет за допомогою відповідних протоколів (переважно HTTP/HTTPS). Іншими словами, фронтенд - це те, що бачить користувач і з чим взаємодіє кожного разу, коли підключається до Інтернету та відкриває будь-який браузер.

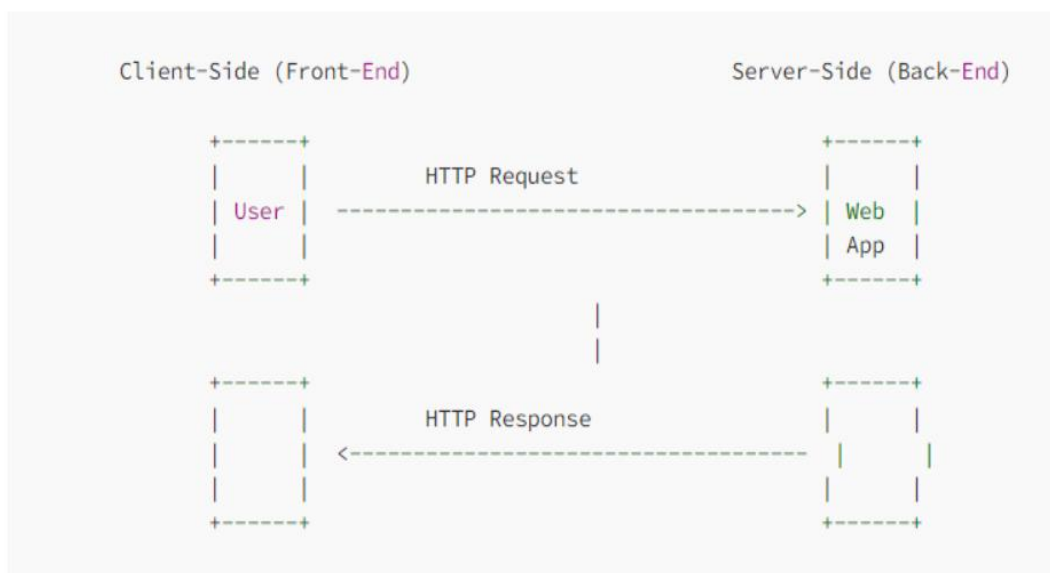


Рисунок 1.2. Спрощена схема взаємодії між frontend і backend

Для розробки фронтенд частини будуть використовуватися Vue 3 та Vite, сучасні технології створення веб-додатків. Vue 3 є останньою версією популярного фреймворку Vue.js, який включає покращення продуктивності та новий API Composition для кращої організації коду. Vite, розроблений засновником Vue, забезпечує швидкісну розробку з інструментом "гарячої"

заміни модулів, що дозволяє відразу бачити зміни без перезавантаження сторінки.

Vue 3 - це сучасний прогресивний фреймворк, написаний на мові TypeScript і призначений для створення веб-прикладень на рівні клієнта. Основна область застосування даного фреймворка - це створення та організація користувацького інтерфейсу. На рисунку 1.3.показана плагін Vite для тестування Vue 3.

Vue 3, оновлена версія відомого фреймворку Vue.js для розробки веб-додатків, включає:

API Composition: Дозволяє розподілити логіку програми на зрозумілі, дрібні компоненти.

Збільшена ефективність: Оптимізація завдяки статичному аналізу дерева рендерингу. Вбудована підтримка TypeScript.

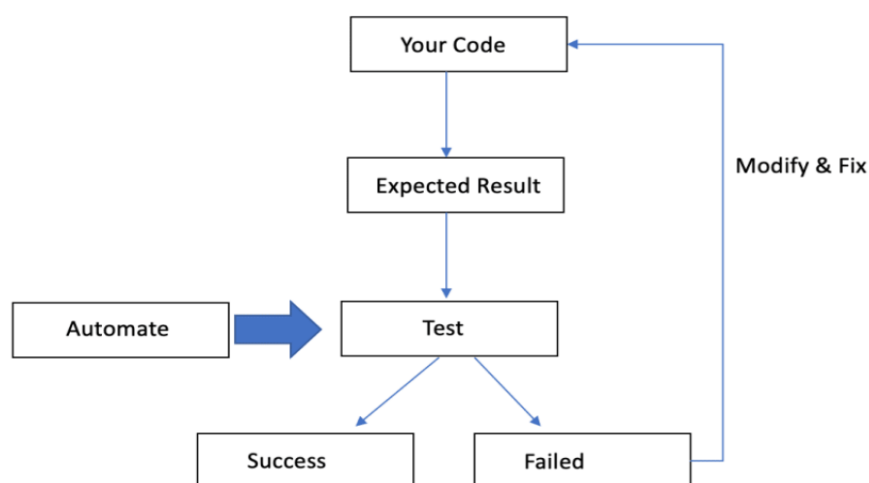


Рисунок 1.3. Плагін Vite для тестування Vue 3

Vite, розроблений Еваном Ю, пропонує: "Гаряча" заміна модулів (HMR): Відображення змін у реальному часі.

Підтримка TypeScript, JSX, CSS, JSON та інших модулів заздалегідь.

Сумісність з сучасними фреймворками: Vue 3, React, Preact.

Простота налаштування: Ефективне управління проектами у режимах розробки та продакшен. Vue 3 та Vite разом формують сильний, зручний для розробників стек для створення веб-додатків.

1.7 Програмна реалізація програми «Інтелектуальний помічник»

1.7.1 Розробка Фронтенду

Щоб почати новий проект на Vue 3 з використанням Vite, спершу потрібно встановити Node.js і npm. Після цього ви можете використовувати командний рядок для створення нового проекту.

Ось основні кроки:

1. Встановлюємо Vite глобально на нашому комп'ютері за допомогою npm:

```
npm install -g create-vite
```

2. Створюємо новий проект Vue 3 за допомогою Vite:

```
create-vite my-vue-project --template vue
```

Ця команда створить нову директорію з ім'ям нашого проекту, яка міститиме всі необхідні файли для проекту Vue 3.

3. Переходимо до директорії нашого проекту:

```
cd my-vue-project
```

4. Встановлюємо всі залежності проекту:

```
npm install
```

5. Запускаємо сервер розробки:

```
npm run dev
```

Після цього ваш проект Vue 3 з Vite повинен бути доступний в браузері за адресою : <http://localhost:5173>.

1. Встановлення Node.js

Поетапно встановлюємо Node.js двома різними способами: безпосередньо з офіційного сайту Node.js та за допомогою Node Version Manager (NVM).

Метод 1: Встановлення Node.js безпосередньо з офіційного сайту

1. Захожу на сайт Node.js:

Відкрийте веб-браузер і перейдіть на <https://nodejs.org>

2. Вибераю версію для завантаження:

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		30

Я бачу опції для завантаження версії "LTS" або "Current". Версію LTS (Long Term Support) рекомендується для більшості користувачів, оскільки вона стабільніша.

Завантажую інсталятор: Натисною на посилання інсталятора для Windows або macOS в залежності від операційної системи. Це завантажить виконуваний файл (.msi для Windows і .pkg для macOS).

Запускаю інсталятор: Після завершення завантаження запустіть інсталятор. Це розпочне майстер налаштування.

Для Windows: слідую інструкціям у майстру налаштування. Приймаю угоду про ліцензію, вибираю папку для встановлення і натискайте кнопку "Далі", поки не досягне кнопки "Встановити".

Для macOS: відкрийте файл .pkg і слідуйте інструкціям для встановлення Node.js.

Перевіряю встановлення: Відкриваю командний рядок (Windows) або термінал (macOS). Введіть `node -v` і натисніть Enter. Це покаже версію Node.js, яка була встановлена, підтверджуючи встановлення. Також перевіряю npm (node package manager) за допомогою введення `npm -v` і натискання Enter.

Метод 2: Встановлення Node.js за допомогою NVM (Node Version Manager)

1.Встановлюю NVM: Для Windows завантажте та встановіть `nvm-windows` зі сторінки <https://github.com/coreybutler/nvm-windows/releases>

Для macOS та Linux відкрийте термінал і виконайте наступну команду `curl` для завантаження та встановлення NVM:

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
```

Після встановлення закрити і знову відкрити термінал, щоб почати використовувати `nvm`.

1.Перевіряю встановлення NVM:

У мому терміналі ввожу `nvm --version`. Це покаже версію NVM, яка була встановлена, підтверджуючи, що вона готова до використання.

Встановлення Node.js за допомогою NVM:

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		31

Встановлюю останню версію Node.js LTS, виконавши: `css`

Copy code: `nvm install --lts`

Або ви можна встановити конкретну версію Node.js, вказавши версію: Copy code

```
nvm install 20.14.0
```

Перемикаюсь між встановленими версіями Node:

Є можливість перемикатися між різними встановленими версіями, використовуючи: `nvm use 18.13.0`

Підтверджуємо встановлення Node.js:

Щоб підтвердити версію Node.js, яка в даний час використовується, вводимо: `node -v`

Ці два методи забезпечують гнучкість, залежно від того, чи хочете ви мати фіксовану версію (пряме встановлення) чи можливість перемикання між кількома версіями (використовуючи NVM).

1.7.2 Налаштування проекту

Створюю нову директорію для проекту, наприклад, "backend ". Ініціалізуйте проект, виконавши в директорії `npm init -y`, що створить файл `package.json` для управління залежностями проекту.

1. Встановлення залежностей.

Залежності, які використовуються в проекті, описуються у форматі JSON у файлі `package.json`. "dependencies":

```
{
  "@anthropic-ai/sdk": "^0.22.0",
  "axios": "^ 1.7.2",
  "body-parser": "^1.20.2",
  "cors": "^2.8.5",
  "dotenv": "^16.4.5",
  "eslint": "^9.4.0",
```

```
"express": "^4.19.2",
"helmet": "^7.1.0",
"pino-http": "^10.1.0",
"swagger-ui-express": "^5.0.1",
}
```

Кожен пакет виконує певну роль у створенні та підтримці веб-сервера на Node.js з використанням Express.js. Ось детальний опис кожної залежності:

1. Express.js (“express”):

Це швидкий, невеликий та гнучкий фреймворк для веб-додатків Node.js, який надає набір потужних функцій для створення веб-додатків та API.

2. @anthropic-ai/sdk:

Це SDK (набір інструментів для розробників), який дозволяє інтегрувати додаток із сервісами Anthropic AI, забезпечуючи легкий доступ до їхніх API для роботи зі штучним інтелектом.

3. Axios (“axios”):

HTTP клієнт для виконання асинхронних запитів до інших HTTP API. Він дозволяє виконувати запити на сервер чи API і легко обробляти відповіді.

4. Body-parser (“body-parser”):

Middleware для Express.js, який аналізує вхідні JSON, сирі, текстові та URL-кодовані дані, що подаються через HTTP запити, і забезпечує легкий доступ до них через “req.body”.

5. CORS (“cors”):

Middleware, що дозволяє налаштувати політику спільного використання ресурсів на різних джерелах (Cross-Origin Resource Sharing) для вашого сервера. Це необхідно для того, щоб браузери дозволяли вашому веб-додатку інтерактивно спілкуватися з ресурсами, що знаходяться на іншому домені.

6. Dotenv (“dotenv”):

Завантажує змінні оточення з файлу “.env” у “process.env”, що дозволяє легко керувати приватними налаштуваннями без змін у коді.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		33

7.ESLint (“eslint”):

Інструмент для статичного аналізу коду, який допомагає виявляти проблемні шаблони або код, що не відповідає визначеним стилям кодування, що забезпечує більшу чистоту і послідовність коду.

8. Helmet (“helmet”):

Допомагає захистити ваш додаток, встановлюючи різні заголовки HTTP, що можуть допомогти убезпечити додаток від деяких відомих вразливостей веб-додатків.

9. Pino-http (“pino-http”):

Логгер для HTTP запитів і відповідей, що базується на бібліотеці “pino”, відомій своєю високою швидкістю та низьким впливом на продукція команда завантажить та встановить Express.js та всі його залежності в папку "node_modules" вашого проекту.

1.7.3 Реалізація сервера

У директорії "backend " створений файл server.js. У цьому файлі налаштує сервер Express.js і створюю кінцеву точку /api/completions для обробки POST-запитів. Скриншот коду server.js представлено на рисунку 1.4.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		34

```
const app = express();
app.use(helmet());
const uncaughtException = async () => {
  process.on('uncaughtException', (err) => {
    console.error('uncaughtException');
    console.error(err);
  });
};

export const create = async () => {
  app.use(cors());
  app.use(pinoHttp({ level: 'error' }));
  app.use(
    bodyParser.urlencoded({
      extended: true,
    }),
  );
  app.use(bodyParser.json());
  app.get('/', (req, res) => {
    res.send('Ok');
  });
  app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
  app.post('/api/completions', async (req, res) => {
    try {
      const prompt = req.body;
      const result = await getAnthropicMessages(prompt);
      res.json(result);
    } catch (error) {
      res.status(500).send(String(error));
    }
  });
};
```

Рисунок 1.4. Скриншот коду файлу server.js

Кінцева точка `/api/completions` приймає POST-запити, що містять вхідні дані від фронтенд частини у форматі JSON, обробляє їх і взаємодіє з API Anthropic для отримання згенерованих відповідей.

Інтеграція з API Anthropic.

Функція `getAnthropicMessages` відповідає за відправку вхідних даних користувача в API Anthropic, використовуючи такі параметри, як тип моделі та ліміт токенів. Для автентифікації використовується API-ключ, який зберігається у

```

src > controllers > JS anthropic.js > ...
import Anthropic from '@anthropic-ai/sdk';
import dotenv from 'dotenv';
dotenv.config();
const apiKey = process.env.ANTHROPIC_API_KEY;

export const getAnthropicMessages = async ({ systemContent, userContent }) => {
  const anthropic = new Anthropic({ apiKey });
  const content = `
    System:
    |   ${systemContent}
    User:
    |   ${userContent}
  `;
  const msg = await anthropic.messages.create({
    model: 'claude-3-opus-20240229',
    max_tokens: 1024,
    messages: [
      { role: 'user', content }
    ],
  });
  return msg.content.map(
    (cnt, i) => ({ text: cnt.text, index: i });
  );
};

```

Рисунок 1.5. Скриншот функції взаємодії з API Anthropic

Цей код JavaScript є функцією, яка взаємодіє з SDK Anthropic AI для створення та отримання повідомлень.

Перші кілька рядків імпортують необхідні модулі. Модуль “Anthropic” - модуль без залежностей, який завантажує змінні оточення з файлу “.env” у “process.env”. Виклик “dotenv.config” завантажує ці змінні. Змінна “ANTHROPIC_API_KEY” отримується з змінних оточення та зберігається в “apiKey”.

Функція “getAnthropicMessages” є асинхронною функцією, яка приймає об’єкт з властивостями “systemContent” та “userContent” як аргумент. Ця функція експортується, що означає, що її можна імпортувати та використовувати в інших модулях.

У функції створюється новий екземпляр “Anthropic” з переданим “apiKey” у параметрах опцій. Цей екземпляр використовується для взаємодії з Anthropic AI.

Змінна “content” є шаблонним рядком, який форматує “systemContent” і “userContent” у конкретний формат. Цей вміст потім використовується для створення нового повідомлення за допомогою методу “anthropic.messages.create”.

Метод “create” викликається з об'єктом, який вказує модель для використання (“claude-3-opus-20240229”), максимальну кількість токенів для генерації (“1024”) та повідомлення для використання на вході. Повідомлення - це масив об'єктів, кожен з яких має роль 'user' та властивість “content”. У цьому випадку є одне повідомлення з попередньо відформатованим вмістом.

Метод “create” повертає обіцянку, яка переходить у стан вирішеності з повідомленням. Ключове слово “await” використовується для призупинки виконання функції до вирішення обіцянки.

Нарешті, вміст повернутого повідомлення відображається у масив об'єктів, кожен з яких має властивості “text” і “index”. Властивість “text” - це текст повідомлення, а “index” - індекс повідомлення в масиві. Цей масив потім повертається функцією.

Запуск сервера.

Запустіть сервер за допомогою команди “npm run start”.

Для запуску сервера в режимі розробки потрібно виконати команду “npm run dev”. Ця команда буде перезавантажувати сервер і приймати нові зміни під час розробки.

```
"scripts": {
  "start": "node src/index.js",
  "dev": "nodemon src/index.js",
  "eslint": "eslint src",
  "audit": "npm audit",
  "outdated": "npm outdated",
  "test": "NODE_OPTIONS=--experimental-vm-modules npx jest",
  "test-one": "NODE_OPTIONS=--experimental-vm-modules jest -t 'test axios' --watch",
  "prepare": "husky install"
},
```

Рисунок 1.6. Скриншот команди запуску і тестування сервера

Виконавши команду: “npm run dev” у терміналі можна побачити повідомлення

```
> Inteligent assistant@1.0.0 dev
> nodemon src/index.js

[nodemon] 2.0.22
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node src/index.js`
Server started on port 3000
```

Рисунок 1.7. Скриншот старт сервера у терміналі

Щоб протестувати сервер, можна перейти за URL-адресою <http://localhost:3000/api-docs/> і побачити.

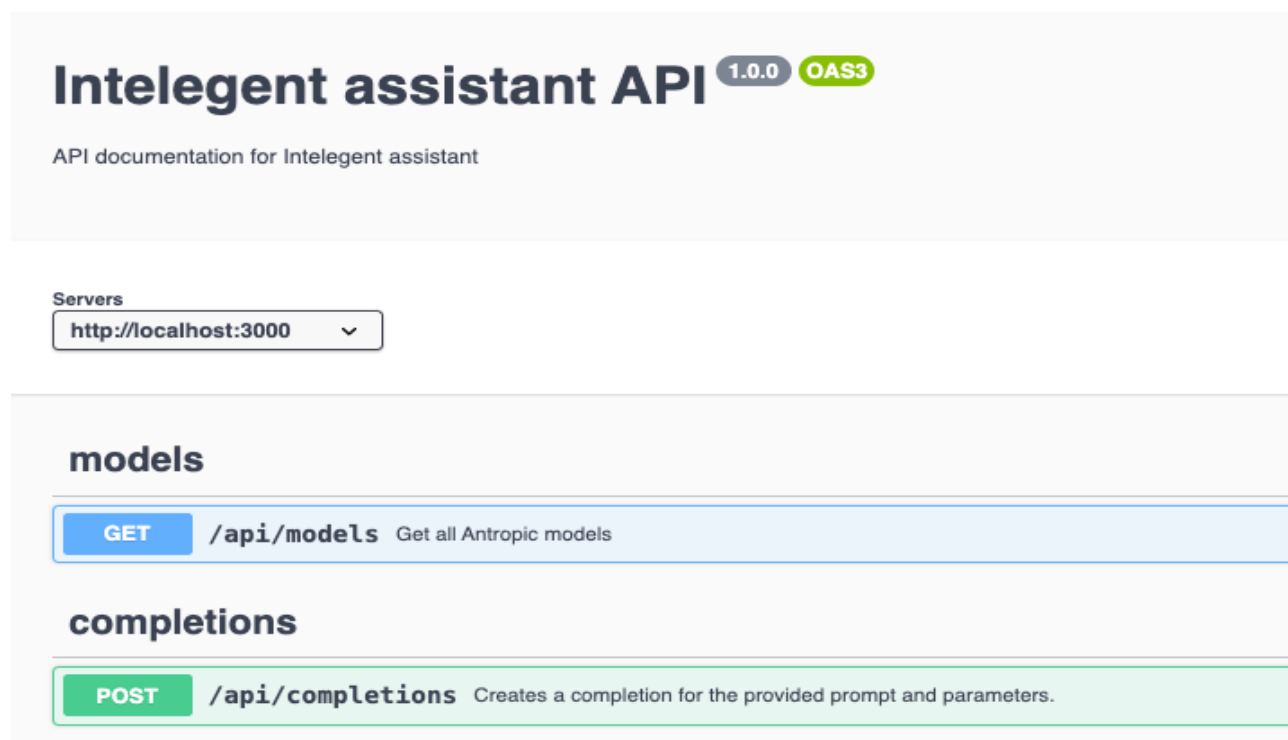


Рисунок 1.8. Скріншот Intelligent Assistant API в Swagger

Безпека:

Щоб ANTHROPIC_API_KEY не зберігати в репозиторії, створимо файл .env

```
backend > .env
1 # Do not share your ANTHROPIC API key with anyone! It should remain a secret.
2 ANTHROPIC_API_KEY=sk-ant-api03-ufZlgxLo8rre0eodwIteWxEQ-e64EMFP00RfLceglLqCkTLII_cU3w8VSR2eTdUu
3 8Y9htPb8rg-zohvigAA
4 PORT=3000
```

Рисунок 1.9. Скріншот файла.env

За допомогою бібліотеки dotenv зможемо отримувати його значення в програмі.

```
“import dotenv from 'dotenv';
```

```
dotenv.config();
```

```
const apiKey = process.env.ANTHROPIC_API_KEY;
```

Щоб розпочати роботу над новим проектом на Vue 3 із використанням збірника Vite, перш за все, вам потрібно встановити Node.js і npm, якщо вони ще не встановлені на вашому комп'ютері. Це забезпечить необхідне середовище для роботи з сучасними веб-технологіями. Ось детальний опис кроків:

Встановлення Vite: Для глобального встановлення Vite на вашому комп'ютері виконайте наступну команду в командному рядку:

```
“npm install -g create-vite”
```

Це дозволить вам використовувати Vite для створення нових проектів будь-де на вашому комп'ютері.

Створення нового проекту Vue 3: За допомогою Vite ви можете легко створити новий проект Vue 3: “create-vite my-vue-project --template vue”.

Виконання цієї команди створить нову папку з назвою проекту my-vue-project, яка включатиме всі стандартні файли та конфігурації для проекту на Vue3

Перехід до директорії проекту: Введіть команду нижче, щоб перейти до папки вашого проекту: “cd my-vue-project”

1. Встановлення залежностей: Ініціалізуйте та встановіть усі необхідні залежності вашого проекту за допомогою npm: “npm install”

2. Запуск сервера для розробки: Запустіть локальний сервер розробки, щоб перевірити ваш проект у дії: “npm run dev”

Мій проект тепер буде доступний у браузері за адресою: <http://localhost:5173>.

Зазначимо, що наведена конфігурація є лише вихідною точкою. В залежності від ваших потреб, вам може знадобитися додати додаткові плагіни, налаштування або бібліотеки, щоб розширити функціональність вашого проекту.

Структура проекту: детально представлена на відповідному рисунку в документації, що дозволяє легко зорієнтуватися в організації файлів і компонентів проекту.

Розглянемо структуру проекту:

Цей проект на Vue 3 має структуру з додатковими інструментами та конфігураціями, які спрямовані на поліпшення процесу розробки та розгортання.

Ось детальний опис його структури:

Папки та Файли:

frontend:

node_modules: Містить всі залежності проекту, встановлені через npm або yarn.

public:

favicon.ico: Іконка сайту, що з'являється у вкладці браузера.

loader.gif: Зображення, яке може бути використане як індикатор завантаження.

src:

assets:

main.scss: Файл SCSS для глобальних стилів.

components:

ChatComponent.vue: Vue компонент для чат-функціоналу.

ResultModal.vue: Vue компонент для модальних вікон, які відображають результати або інформацію.

api.js: Модуль для взаємодії з зовнішніми API.

App.vue: Основний кореневий компонент Vue.

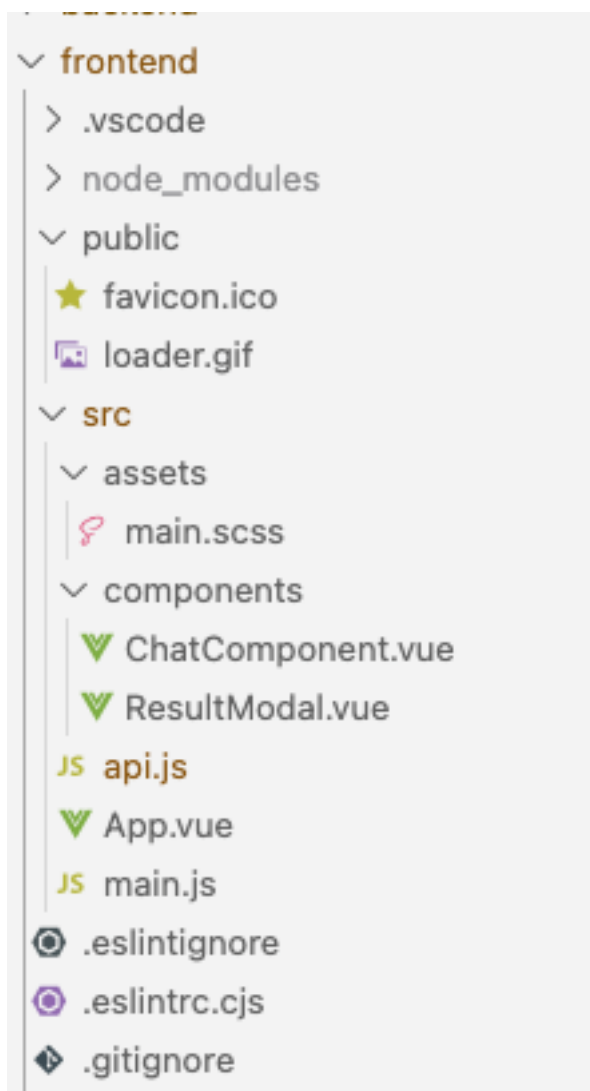


Рисунок 1. 10. Скриншот структури проекту

main.js: Точка входу для ініціації Vue додатка.

.eslintrc.js: Конфігурація для ESLint.

eslintignore: Визначає файли, які ESLint ігноруватиме.

.gitignore: Вказує файли, які Git ігноруватиме.

gitlab-ci.yml: Конфігурація для GitLab CI/CD, що автоматизує процеси тестування, збірки та розгортання.

.prettierrc.json: Конфігурація для Prettier, інструменту для форматування коду.

Dockerfile: Інструкції для створення Docker контейнера для проекту.

index.html: Головний HTML файл, що використовується як шаблон.

package-lock.json та package.json: Вміщують інформацію про залежності проекту.

postcss.config.js: Конфігурація для PostCSS, яка дозволяє використовувати різні плагіни для обробки CSS.

README.md: Файл з описом проекту, його структурою, інструкціями по встановленню тощо.

tailwind.config.js: Конфігурація для Tailwind CSS, фреймворку для стилізації.

vite.config.js: Конфігурація для Vite, інструменту для збірки, який забезпечує швидке оновлення модулів.

package.json - це ключовий файл у проектах на Node.js, який відіграє важливу роль у управлінні проектом та його залежностями. Цей файл містить різні метадані, які є важливими для проекту, включаючи його залежності, скрипти, версії та інше.

Поле "name" вказує назву проекту, яка в цьому випадку є "vue-latest". Поле "version" вказує поточну версію проекту, яка є "0.0.0" тут. Поле "private": true означає, що цей пакет не має бути опублікований на npm.

Розділ "scripts" містить команди, які можна виконувати в терміналі за допомогою npm run <назва-скрипта>.

“npm run start” - ініціює запуск вашого проекту;

“npm run dev” - активує режим розробки для проекту;

“npm run lint” – проводить аналіз коду за допомогою лінтера.

Лінтер - це утиліта, яка перевіряє код на наявність стилістичних недоліків, помилок форматування, потенційних помилок та інших загроз якості коду;

“npm run format” - застосовує інструмент Prettier для надання єдиної стилістичної форми коду у вашому проекті. Prettier допомагає уніфікувати стиль написання коду по всьому проекту.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		42

frontend > {} package.json > {} dependencies

```
1  {
2    "name": "vue-latest",
3    "version": "0.0.0",
4    "private": true,
5    "scripts": {
6      "start": "vite",
7      "dev": "vite --open",
8      "build": "vite build",
9      "preview": "vite preview",
10     "lint": "eslint . --ext .vue,.js,.jsx,.cjs,.mjs --fi",
11     "format": "prettier --write src/"
12   },
13   "dependencies": {
14     "@fortawesome/fontawesome-svg-core": "^6.4.0",
15     "@fortawesome/free-brands-svg-icons": "^6.4.0",
16     "@fortawesome/free-regular-svg-icons": "^6.4.0",
17     "@fortawesome/free-solid-svg-icons": "^6.4.0",
18     "@fortawesome/vue-fontawesome": "^3.0.3",
19     "@headlessui/vue": "^1.7.13",
20     "@heroicons/vue": "^2.0.18",
21     "@tailwindcss/forms": "^0.5.3",
22     "axios": "^1.4.0",
23     "vue": "^3.2.47",
24     "vue3-toast": "^0.0.1",
25     "vue-social-sharing": "^4.0.0-alpha4"
26   },
```

Рисунок 1.11. Скріншот файлу package.json

Розділ "dependencies" перераховує бібліотеки, які ваш проект використовує безпосередньо. Кожна залежність вказана з версією. Знак ^ перед номером версії означає, що npm може встановлювати оновлення, які не змінюють найлівішу цифру версії, яка не є нулем. Наприклад, 1.2.3 дозволяє встановлювати до 2.0.0.

Розділ "devDependencies" перераховує бібліотеки, необхідні для розробки проекту. Ці бібліотеки не потрібні у виробництві і зазвичай включають фреймворки для тестування, збірники або транспілятори. Правила версіонування такі ж, як і для "dependencies".

У цьому конкретному файлі package.json, проект є додатком Vue.js, який використовує Vite як інструмент для збірки, ESLint для лінтингу, Prettier для форматування коду, та включає різні інші бібліотеки, як-от FontAwesome, Axios, та Tailwind CSS.

Код головного файлу index.html, що використовується як шаблон.



```
index.html x App.vue JS api.js M JS server.js M JS antropic.js M package.json frontend swagger.yml
frontend > index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <link rel="icon" href="/favicon.ico" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <link rel="stylesheet" href="https://rsms.me/inter/inter.css" />
8     <title>Інтелектуальний помічник</title>
9   </head>
10  <body>
11    <div id="app"></div>
12    <script type="module" src="/src/main.js"></script>
13  </body>
14 </html>
15
```

Рисунок 1.12. Скріншот файлу index.html:

Ось переклад на українську мову опису структури базового HTML документа, який є точкою входу для веб-додатка:

Це базова структура HTML документа, яка використовується як вхідна точка для веб-додатка.

Декларація “<!DOCTYPE html>” на початку вказує веб-браузеру про версію HTML, у якій написана сторінка. У цьому випадку, це HTML5.

Тег “<html lang="en">” є коренем HTML документа. Атрибут “lang” вказує мову документа, яка в даному випадку англійська.

У середині тегу “<head>“, містяться кілька тегів meta і посилань:

Тег “<meta charset="UTF-8">“ встановлює кодування символів для документа, яке в цьому випадку UTF-8. UTF-8 підтримує всі символи та знаки в Unicode, що робить його дуже універсальним.

Тег “<link rel="icon" href="/favicon.ico">“ використовується для вказівки фавікону веб-сторінки, який зазвичай відображається в адресному рядку браузера або поруч із назвою сайту в списку закладок.

Тег “<meta name="viewport" content="width=device-width, initial-scale=1.0">“ використовується для адаптивного веб-дизайну. Він встановлює ширину сторінки так, щоб вона відповідала ширині екрану пристрою (яка змінюватиметься залежно від пристрою) і початковий масштаб, коли сторінка вперше завантажується браузером.

Тег “<link rel="stylesheet" href="https://rsms.me/inter/inter.css">“ використовується для підключення зовнішнього CSS файлу, щоб стилізувати веб-сторінку.

Тег “<title>“ використовується для вказівки заголовка веб-сторінки, який відображається в заголовку вкладки браузера.

У тегу “<body>“ розміщений “<div>“ з ідентифікатором "app". Це типово для веб-додатків, що використовують JavaScript фреймворки, як Vue або React, які "монтують" додаток.

Нарешті, тег “<script type="module" src="/src/main.js"></script>“ включений для включення JavaScript файлу, який містить логіку веб-додатка. Атрибут “type="module”“ використовується, коли ви хочете імпортувати ES6 модулі у ваш скрипт.

Файл main.js точка входу для ініціалізації Vue додатка

```

> src > JS main.js > ...
import './assets/main.scss'
import { FontAwesomeIcon } from '@fortawesome/vue-fontawesome'
import VueSocialSharing from 'vue-social-sharing'
import { createApp } from 'vue'
import { library } from '@fortawesome/fontawesome-svg-core'
import App from './App.vue'
import { faUserSecret } from '@fortawesome/free-solid-svg-icons'
import {
  faTwitter,
  faFacebook,
  faLinkedin,
  faTelegram
} from "@fortawesome/free-brands-svg-icons";
import { fas } from "@fortawesome/free-solid-svg-icons";
const app = createApp(App);
library.add(faUserSecret, faTwitter, faFacebook, faLinkedin, faTelegram, fas);
app.component("font-awesome-icon", FontAwesomeIcon);
app.use(VueSocialSharing);
app.mount('#app');

```

Рисунок 1.13. Скріншот файлу main.js стартовий файл для проекту:

Цей файл JavaScript є точкою входу для додатка Vue.js. Він відповідає за налаштування додатка та його приєднання до певного елемента в HTML.

На першому рядку “import './assets/main.scss’” імпортується основний SCSS (Sassy CSS) файл. SCSS - це CSS-препроцесор, який додає функції як змінні, вкладеність та міксини, що робить роботу з CSS більш ефективною та легкою.

Далі імпортуються кілька модулів з різних бібліотек:

“FontAwesomeIcon” з “@fortawesome/vue-fontawesome” - це компонент, який надається бібліотекою FontAwesome, що дозволяє використовувати іконки FontAwesome у вашому компоненті Vue.

“VueSocialSharing” з “vue-social-sharing” - плагін для Vue.js, який дозволяє реалізувати функціонал соціального обміну.

“createApp” з “vue” - це функція, яка створює новий екземпляр застосунку Vue. “library” з “@fortawesome/fontawesome-svg-core”- це частина API FontAwesome, яка дозволяє додавати іконки до бібліотеки, які можна буде використовувати в усьому додатку.

“App” з “./App.vue” - основний компонент вашого додатку Vue.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		46

Також імпортуються кілька іконок з “@fortawesome/free-solid-svg-icons” і “@fortawesome/free-brands-svg-icons”.

Після імпорту необхідних модулів створюється новий додаток Vue за допомогою “const app = createApp(App);”.

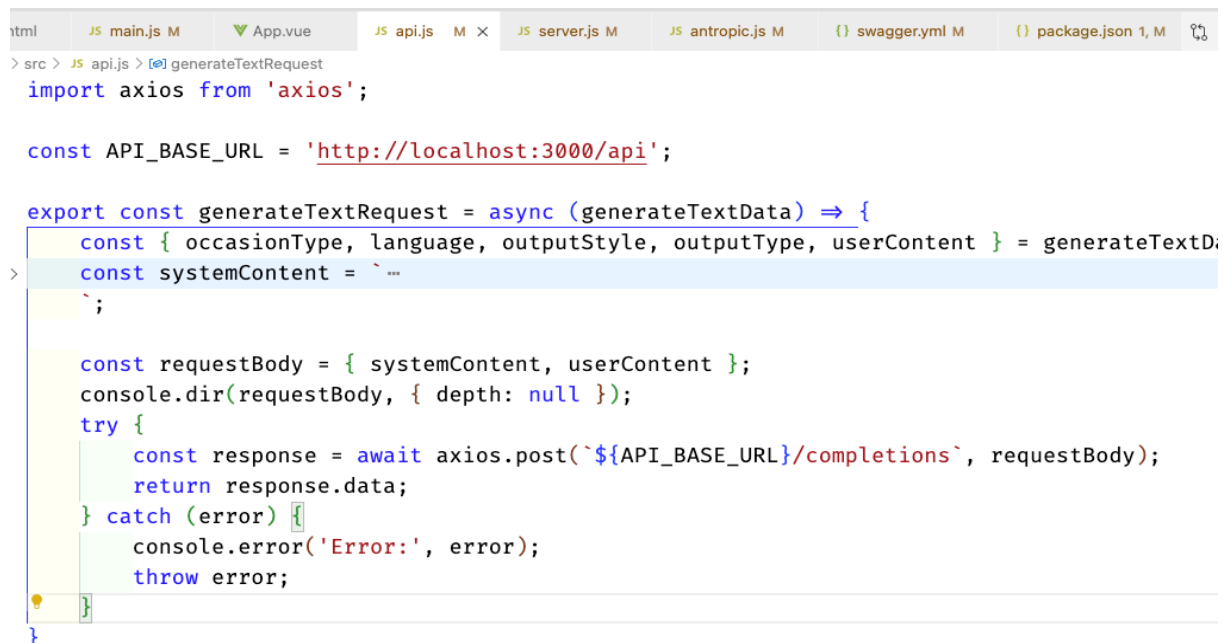
Далі іконки FontAwesome додаються до бібліотеки FontAwesome за допомогою “library.add(faUserSecret, faTwitter, faFacebook, faLinkedin, faTelegram, fas);”.

Компонент “FontAwesomeIcon” реєструється глобально з “app.component('font-awesome-icon', FontAwesomeIcon);”. Це означає, що його можна використовувати в будь-якому компоненті застосунку без необхідності його імпортування щоразу.

Застосунок Vue встановлюється з “app.use(VueSocialSharing);”.

Додаток Vue прикріплюється до HTML елемента з id “app” за допомогою “app.mount('#app');”. Це місце, де Vue додаток буде відображений в HTML.

У файлі api.js міститься код для взаємодії з серверною частиною



```
html JS main.js M App.vue JS api.js M x JS server.js M JS antropic.js M {} swagger.yml M {} package.json 1, M
> src > JS api.js > generateTextRequest
import axios from 'axios';

const API_BASE_URL = 'http://localhost:3000/api';

export const generateTextRequest = async (generateTextData) => {
  const { occasionType, language, outputStyle, outputType, userContent } = generateTextData;
  const systemContent = `...
  `;

  const requestBody = { systemContent, userContent };
  console.dir(requestBody, { depth: null });
  try {
    const response = await axios.post(`${API_BASE_URL}/completions`, requestBody);
    return response.data;
  } catch (error) {
    console.error('Error:', error);
    throw error;
  }
}
```

Рисунок 1.14. Скріншот взаємодії з бекенд частиною

Цей JavaScript код визначає асинхронну функцію “generateTextRequest”, яка робить POST запит до API-кінцевої точки. Функція експортується, тому її можна використовувати в інших частинах додатку.

1.8 Опис роботи програми

Мій проект тепер буде доступний у браузері за адресою: <http://localhost:5173>.

Дизайн вікна програми.

Інтерфейс, зображений на скріншоті, належить до інтелектуального помічника, призначеного для створення повідомлень на основі користувацьких вказівок.

Основні елементи інтерфейсу:

1. Мова: Випадаючий список, який дозволяє користувачеві вибрати мову повідомлення. Опції включають українську, англійську, французьку, німецьку та японську мови.

Інтелектуальний помічник
Створюйте повідомлення на всі випадки

Мова: Українська
Тип повідомлення: Не визначений
Стиль повідомлення: Неформальний
Тип виведення: Текст

Контекст події (опишіть ситуацію)

Показати результат

Рисунок 1.15. Скріншот вікна програми

2. Тип повідомлення: Випадаючий список для вибору типу повідомлення, який може варіюватися від не визначеного, привітань, інформаційних повідомлень до офіційних повідомлень.

3. Стиль повідомлення: Випадаючий список, що дозволяє вибрати стиль тексту повідомлення, такий як неформальний, формальний, офіційно-діловий, технічний, короткий, або нейтральний.

4. Тип виведення: Випадаючий список, що дозволяє користувачеві вибрати формат виведення тексту, включаючи простий текст, email, SMS, HTML та Markdown.

5. Контекст події (опишіть ситуацію): Текстове поле, де користувач може ввести деталі чи описати ситуацію, на основі якої потрібно створити повідомлення.

Відправлення даних:

Під формою знаходиться кнопка "Показати результат", яка активує процес створення повідомлення на основі введених даних.

Інтерфейс забезпечує інтуїтивно зрозуміле та ефективне засіб для користувачів створювати повідомлення, налаштовуючи їх мову, стиль, тип та контекст, що забезпечує високий рівень персоналізації.

Вікно результату на рисунку 1.16. Вікно результату на скріншоті є частиною інтерфейсу інтелектуального помічника, що відображає кінцевий результат генерації повідомлення.

Заголовок вікна:

Ваш результат: Цей текст вказує на те, що нижче відображається результат запиту користувача.

Вміст вікна:

Текст вікна результату містить згенероване повідомлення, яке у даному прикладі є привітанням: "Привіт! Як справи? Чим займаєшся?"

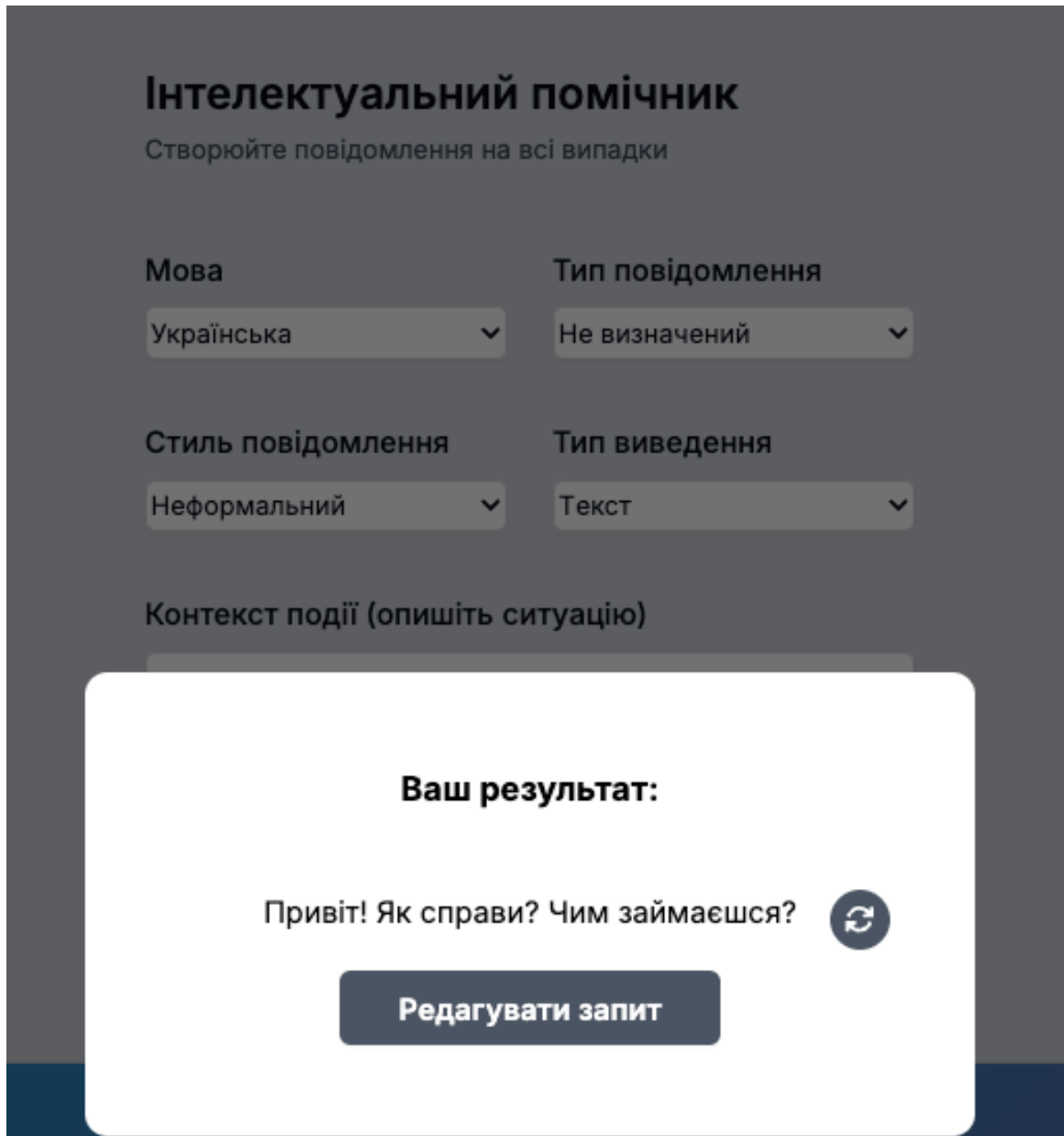


Рисунок 1.16. Скриншот результат роботи

Кнопки:

Редагувати запит: Кнопка дозволяє користувачеві змінити вхідні параметри запиту для генерації нового повідомлення або скорегувати поточне. Це забезпечує можливість ітеративної взаємодії з системою для досягнення бажаного результату.

Іконка у вигляді кругових стрілок збоку від тексту - це кнопка для регенерації відповіді. Це вікно результату ефективно демонструє відповідь системи і забезпечує користувачеві зручний засіб для подальших дій, базуючись на отриманому результаті.

Приклади роботи:

Інтелектуальний помічник
Створюйте повідомлення на всі випадки

Мова: Українська
Тип повідомлення: Попередження
Стиль повідомлення: Формальний
Тип виведення: Текст

Контекст події (опишіть ситуацію)
Вітаю Я сьогодні прихворіла і не зможу зустрітись з Вами.

Показати результат

Рисунок 1.17. Скріншот результат роботи завдання

Ваш результат:

Шановний пане/пані, Цим повідомленням змушена попередити Вас про те, що через погане самопочуття я, на жаль, не зможу бути присутньою на нашій запланованій зустрічі сьогодні. Прощу вибачення за будь-які незручності, які це може спричинити. Сподіваюся на Ваше розуміння та прошу перенести нашу зустріч на інший, більш слушний час. Про дату та час наступної зустрічі я обов'язково повідомлю додатково, щойно мій стан здоров'я покращиться. Ще раз перепрошую за зміни в наших планах та дякую за терпіння. З повагою, [Ваше ім'я]

Редагувати запит

Рисунок 1.18. Скріншот виконання роботи завдання

Інтелектуальний помічник

Створюйте повідомлення на всі випадки

Мова

Українська ▾

Тип повідомлення

Оголошення ▾

Стиль повідомлення

Формальний ▾

Тип виведення

Текст ▾


Контекст події (опишіть ситуацію)

Напиши рекламу для Одеського технічного коледжу для абітурієнтів

Показати результат

Рисунок 1.19. Скріншот роботи завдання

Ваш результат:



ОДЕСЬКИЙ ТЕХНІЧНИЙ КОЛЕДЖ ОГОЛОШУЄ НАБІР АБИТУРІЄНТІВ

Шановні абітурієнти! Одеський технічний коледж запрошує вас стати частиною нашої престижної освітньої установи. Ми пропонуємо високоякісну освіту в галузі технічних наук, що відповідає сучасним вимогам ринку праці. Наш коледж має потужну матеріально-технічну базу, що включає сучасні лабораторії, комп'ютерні класи та майстерні. Викладацький склад коледжу налічує висококваліфікованих фахівців, які забезпечать вам ґрунтовну теоретичну та практичну підготовку.

Навчання в Одеському технічному коледжі відкриває широкі перспективи для подальшого професійного розвитку та кар'єрного зростання. Наші випускники є затребуваними спеціалістами на ринку праці та успішно працюють у провідних компаніях галузі. Ми пропонуємо такі спеціальності: - Комп'ютерна інженерія - Електроніка та телекомунікації - Автоматизація та комп'ютерно-інтегровані технології - Метрологія та інформаційно-вимірювальна техніка - Галузеве машинобудування

Подайте документи до Одеського технічного коледжу та станьте частиною нашої дружньої студентської спільноти! Ми чекаємо на вас та готові допомогти реалізувати ваші мрії та амбіції. За детальнішою інформацією звертайтеся до приймальної комісії Одеського технічного коледжу. Наша адреса: м. Одеса, вул. Балківська, 54. Телефон: (048) 722-00-00 Веб-сайт: www.otk.od.ua

Редагувати запит

Рисунок 1.20. Скріншот виконання роботи завдання

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		52

Інтелектуальний помічник

Створюйте повідомлення на всі випадки

Мова

Українська

Тип повідомлення

Не визначений

Стиль повідомлення

Офіційно-діловий

Тип виведення

Текст

Контекст події (опишіть ситуацію)

Завтра нарада в 10 ранку

Показати результат

Рисунок 1.21. Скріншот завдання

Ваш результат:

Шановні колеги, Хочу нагадати, що завтра о 10:00 відбудеться важлива нарада. Прошу всіх бути присутніми вчасно та підготувати необхідні матеріали для обговорення. Якщо у вас виникли непередбачені обставини і ви не зможете бути присутніми, будь ласка, повідомте про це завчасно. Дякую за вашу увагу та співпрацю. Очікую на зустріч з вами завтра. З повагою, [Ваше ім'я]



Редагувати запит

Рисунок 1.22. Скріншот виконання роботи завдання

Ізм.	Лист	№ докум.	Підпис	Дата

КГ 08. 14 001. 00 ДП ПЗ

Арк.

53

Ваш результат:

Dear [Recipient], Please be advised that a meeting has been scheduled for tomorrow morning at 10:00 AM. Your attendance is required. Kindly ensure that you are well-prepared for the meeting and arrive on time. If you have any conflicts with this scheduled time, please inform me as soon as possible so that alternative arrangements can be made if necessary. Thank you for your attention to this matter. I look forward to your presence at the meeting tomorrow. Best regards, [Your Name]



Редагувати запит

Рисунок 1.23. Скріншот виконання роботи завдання

2 ЕКОНОМІЧНА ЧАСТИНА

2.1 Резюме

Темою даного дипломного проекту є «Розробка програми для генерації повідомлень за допомогою штучного інтелекту».

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

2.2. Визначення трудомісткості розробки програмного забезпечення.

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначаємо обсяг програмних засобів, у тисячах умовних машинних команд програми аналога

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт.

Таблиця 2.1 Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_0 , усл машинних командах.
1. ПП СУБД	2500 – 9800
2. Комплексні системи ведення БД	950 – 7430
3. ПП організації обчислювального процесу	13000 – 10200

Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить V_0 в умовних машинних командах, трудомісткості визначати на основі табл.2.2

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7\div 0,8$): $T^a = 262 \times 0,8 = 209,6$ (люд/годин).

Таблиця.2.2 Обсяг ПП

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{T3} = T^a p \times L_1 \times K_H \quad (2.1)$$

$$T_{TP} = T^a p \times L_2 \times K_H \quad (2.2)$$

$$T_{PP} = T^a p \times L_3 \times K_H \times K_T \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага і-го етапу розробки (див. табл. 2.2.);

K_H – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.3.);

K_T – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.4.).

Таблиця 2.2.Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
ТЗ (L_1)	0,15	0,12	0,12
ТП (L_2)	0,16	0,15	0,11
РП (L_3)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.3. Значення поправочного коефіцієнта

Код ступеня новизни	Ступінь новизни	Значення K_H
А	Принципово нові ПО	1,75 – 1,2
Б	ПО – розвиток визначеного параметричного ряду	1,0 – 0,8
В	ПО маючий аналог	0,7

Таблиця 2.4. Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблювального ПО типовими програмами, %	Значення K_T
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{ТЗ} = T^a * L_1 * K_H = 209,6 * 0,12 * 1,0 = 25,15 \text{ (люд/годин)} \quad (2.1)$$

Трудомісткість розробки технічного проекту

$$T_{ТП} = T^a * L_2 * K_H = 209,6 * 0,15 * 1,0 = 31,44 \text{ (люд/годин)} \quad (2.2)$$

Трудомісткість розробки робочого проекту

$$T_{РП} = T^a * L_3 * K_H * K_T = 209,6 * 0,58 * 1,0 * 0,9 = 109,41 \text{ (люд/годин)} \quad (2.3)$$

Для подальших розрахунків визначили кількість папера, витраченого на кожен етап: технічне завдання $N_{ТЗ} = 2$ (стор), розробка ТП $N_{ТП} = 18$ (стор), розробка робочого проекту $N_{РП} = 25$ (стор), пояснювальна записка відповідно $N_{ПЗ} = 20$ (стор)

Розрахунок зведений у таблицю 2.5

Таблиця 2.5. Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, годин.			
	1	2	3	4
1.ТЗ	$T_{ТЗ} = 25,15$	$T_{КК} = 0,7 * N_{ТЗ} = 0,7 * 2 = 1,4$	$T_{НК} = 0,15 * N_{ТЗ} = 0,15 * 2 = 0,30$	
2.Розробка ТП	$T_{РТП} = 31,44$	$T_{КК} = 0,7 * N_{ТП} = 0,7 * 18 = 12,6$	$T_{НК} = 0,15 * N_{ТП} = 0,15 * 18 = 2,7$	
3.Розробка РП	$T_{РРП} = 109,4$	$T_{КК} = 0,7 * N_{РП} = 0,7 * 25 = 17,5$	$T_{НК} = 0,15 * N_{РП} = 0,15 * 25 = 3,8$	
4.Розробка ПЗ	$T_{ПЗ} = 1,5 * N_{ПЗ} = 1,5 * 20 = 30$	$T_{КК} = 0,7 * N_{ПЗ} = 0,7 * 20 = 14$	$T_{НК} = 0,15 * N_{ПЗ} = 0,15 * 20 = 3,0$	

Ізм.	Лист	№ докум.	Підпис	Дата
------	------	----------	--------	------

КГ 08. 14 001. 00 ДП ПЗ

Арк.

57

1	2	3	4
Усього, в т.ч.:	$\Sigma T = 227,80$		
- на розробку	$\Sigma T_p = 196,00$		
- контроль керівника		$\Sigma T_{kk} = 45,5$	
нормоконтроль			$\Sigma T_{нк} = 9,8$

2.3 Розрахунок ціни програмного продукту

У цьому розділі для формування ціни здійснюється розрахунок основної заробітної плати виконавців, матеріальних витрат, вартості машино-години та витрат, пов'язаних із розробкою програмного забезпечення. Розрахунок основної заробітної плати виконавців наведено у таблиці 2.6. Відповідно до статті 8 «Закону про Державний бюджет України на 2024 рік», з 1 квітня 2024 року встановлено мінімальну заробітну плату у розмірі 8000 гривень на місяць, а мінімальну погодинну тарифну ставку - 46,00 гривень за годину.

Таблиця 2.6 Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
Розробка ПП	196,00	60,00	11760,00
Контроль керівника	45,5	100,00	4550,00
Нормоконтроль	9,8	100,00	980,00
Усього	-	-	$\Sigma Z_o = 17290,00$

Таблиця 2.7 Розрахунок матеріальних витрат на розробку ПП

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна Одиниці	Вартість, грн.
Папір	Лист А4	60	4,0	240
Разом	-	-	-	$B_{mi} = 240$
Транспортно-заготівельні витрати (10%)				$B_{tr_z} = 0,1 \times B_{m1} = 24$
Усього				$B_m = B_{mi} + B_{tr_z} = 264,00$

На основі отриманих даних за окремими статтями витрат було складено калькуляцію планової собівартості проєктного продукту в цілому, відповідно до встановленої форми, наведеної у таблиці 2.8.

Таблиця 2.8. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
Матеріали	264,00	V_M (див. табл. 2.7)
Основна заробітна плата	17290,00	Z_o (див. табл. 2.6)
Додаткова заробітна плата	2593,50	$Z_d = 0,15 \times Z_o = 0,15 * 17290,00$
4. Відрахування до єдиного фонду соціального внеску	4374,37	$V_{\text{с.с.в.}} = 0,22 \times (Z_o + Z_d) = 0,22 * (17290,00 + 2593,50)$
5. Накладні витрати	5187,00	$V_{\text{нак.}} = 0,3 \times Z_o = 0,3 * 17290,00$
Повна собівартість	29708,87	$C_{\text{пов.}} = V_M + Z_o + Z_d + V_{\text{с.с.в.}} + V_{\text{нак.}} = 264,00 + 17290,00 + 2593,50 + 4374,37 + 5187,00$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{\text{п}} * P) / 100 = (29708,87 * 10) / 100 = 2970,88 \text{ грн} \quad (2.4)$$

де, p – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

$$Ц_o = C_{\text{пов.}} + П = 29708,87 + 2970,88 \text{ грн} = 32679,75 \text{ грн} \quad (2.5)$$

Податок на додану вартість визначаємо по наступній формулі:

$$ПДВ = 0,2 * Ц_o = 0,2 * 32679,75 = 6535,95 \text{ грн} \quad (2.6)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_p = Ц_o + ПДВ = 32679,75 + 6535,95 = 39215,70 \text{ грн} \quad (2.7)$$

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

Охорона праці у виробництві завжди відіграє важливу роль. Саме завдяки рекомендаціям з охорони праці персонал підприємства формує алгоритм виконання робочих завдань з обов'язковим дотриманням вимог безпеки. Основна мета охорони праці - це організація та проведення заходів, спрямованих на захист життя, здоров'я та працездатності працівників у процесі трудової діяльності.

Під час роботи за комп'ютером, як і в інших сферах діяльності, необхідно враховувати нормативи щодо освітлення, температури, відносної вологості та рівня вібрації. Проте для офісних приміщень, де використовуються комп'ютери, особливе значення має дотримання правил пожежної безпеки - забезпечення вогнестійкості приміщення. Крім того, важливо контролювати рівень звукового шуму, а також характеристики електромагнітних, ультрафіолетових та інфрачервоних полів.

У межах цього дипломного проєкту охорона праці розглядається на прикладі забезпечення безпеки праці веб-розробника, який працює в офісному середовищі.

3.1 Аналіз та безпека умов праці працівника на робочому місці

Під час будь-якої роботи за комп'ютером на працівника можуть впливати небезпечні та шкідливі виробничі фактори. Вони поділяються на фізичні та психофізіологічні.

Серед фізичних факторів найбільш поширеними є підвищена температура повітря у робочій зоні, підвищений рівень шуму та знижена вологість повітря. Ці фактори зазвичай виникають у приміщеннях із працюючими комп'ютерами, оскільки в процесі їх експлуатації виділяється тепло, що підвищує температуру та знижує рівень вологості. Крім того, комп'ютерне обладнання генерує електростатичні та електромагнітні поля у діапазонах від 5 Гц до 2 кГц та від 2 до 400 кГц, що призводить до підвищеного рівня електромагнітного випромінювання та накопичення статичної електрики на робочому місці.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		60

Ще одним важливим фактором є якість освітлення. В офісних приміщеннях часто бракує природного світла, а використання штучного освітлення не завжди відповідає нормативам. Це може призводити як до недостатньої, так і до надмірної яскравості освітлення, що, у свою чергу, негативно впливає на зір і загальне самопочуття працівника.

До психофізіологічних факторів належать фізичні та нервово-психічні перевантаження. При роботі з комп'ютером домінують нервово-психічні перевантаження. Працівники, зокрема програмісти та розробники, стикаються з напруженням сенсорних аналізаторів через тривале зосередження на екрані, монотонністю роботи та необхідністю тривалого підтримання концентрації. Додатковим чинником нервово-психічного напруження може бути необхідність спілкування з клієнтами під час підготовки технічних завдань, що створює додаткове емоційне навантаження.

3.2 Розробка заходів з охорони праці

Виробниче освітлення

Штучне освітлення в приміщеннях, де розміщено робочі місця з використанням відеодисплейних терміналів (ВДТ), повинно здійснюватися за допомогою системи загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, де переважає робота з документами, допускається застосування комбінованої системи освітлення: поряд із загальним освітленням встановлюються додаткові світильники місцевого освітлення для створення оптимальних умов праці.

Згідно з вимогами ДСТУ EN 12464-1:2016 «Світло та освітлення. Освітлення робочих місць у приміщеннях», рівень освітленості для робочих місць із використанням ВДТ має складати не менше ніж 500 лк (люкс) на робочій поверхні. При роботі, що пов'язана з обробкою документів, рекомендується підвищений рівень освітленості - до 750 лк - для зменшення зорового навантаження.

					КГ 08. 14 001. 00 ДП ПЗ	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		61

Крім того, для запобігання втомі очей важливо забезпечити правильний розподіл яскравості в полі зору працівника, уникати прямого та відбитого блиску, а також обирати лампи з відповідним індексом передачі кольорів ($Ra \geq 80$) та температурою світла в межах 4000–5000 К, що забезпечує комфортне сприйняття інформації з екрана.

Мікроклімат

Під час роботи у приміщеннях з великою кількістю комп'ютерів, які класифікуються як приміщення з підвищеною небезпекою електротравм, спостерігається значне теплове навантаження. У літній період температура повітря в таких приміщеннях може перевищувати 35 °С, що негативно впливає на стан здоров'я працівників. Тому в подібних умовах необхідно забезпечувати ефективне охолодження повітря, а також регулювати рівень вологості за допомогою спеціалізованого кліматичного обладнання.

Відповідно до вимог ДСН 3.3.6.042-99 "Гігієнічні вимоги до відеодисплейних терміналів, персональних електронно-обчислювальних машин та організації роботи", температура повітря в офісних приміщеннях повинна становити від 22 до 25 °С, відносна вологість - у межах 40–60 %, швидкість руху повітря - не більше 0,1 м/с. У разі перевищення встановлених нормативів передбачено скорочення тривалості робочого дня працівників на 10 % для зменшення впливу несприятливих метеорологічних факторів на організм.

3.2 Організація робочого місця користувача ПК

Конструкція робочого місця користувача ПК та взаємне розташування всіх його елементів (сидіння, органів керування, засобів відображення інформації) відповідають антропометричним, фізіологічним та психологічним вимогам, а також специфіці виконуваної роботи. Робочі меблі повинні забезпечувати можливість індивідуального регулювання параметрів відповідно до зросту та

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		62

особливостей статури працівника з метою підтримання комфортної та безпечної робочої пози.

Поверхня робочого столу має бути пофарбована матовою фарбою для запобігання небажаним відбиттям світла. Дисплей встановлюється таким чином, щоб його верхній край знаходився на рівні очей працівника, на відстані приблизно 70 см від очей, що відповідає рекомендованому діапазону 60–90 см. Частота оновлення екрана становить 100 Гц, що задовольняє вимогу щодо частоти мерехтіння ($f > 70$ Гц), тим самим зменшуючи візуальне навантаження.

Робоче місце розташоване перпендикулярно до віконних прорізів з метою запобігання прямому та відбитому блиску на поверхні екрана, який може виникати як від природного, так і від штучного освітлення. Відповідно до теми дипломного проєкту, робоче місце програміста оснащено пристроями, що генерують електромагнітне випромінювання, тому організація робочого простору передбачає дотримання вимог щодо мінімізації впливу електромагнітних полів на працівника.

3.3 Пожежна безпека

Забезпечення пожежної безпеки на об'єкті праці є важливою складовою створення безпечних та здорових умов праці. Відповідна організація пожежної безпеки дозволяє знизити ризик виникнення пожежі, забезпечити своєчасне реагування та мінімізувати можливі наслідки. Проходи до аварійних виходів мають бути постійно вільними та мати ширину не менше 1 метра. У разі накопичення великої кількості горючих відходів слід використовувати спеціально відведені сміттєзбірники.

Електроприлади повинні використовуватися виключно за прямим призначенням. У разі їх пошкодження необхідно негайно відключити живлення та привести прилад у пожежобезпечний стан.

Первинні засоби пожежогасіння застосовуються для боротьби з пожежами на початковій стадії. До них належать: пожежні кран-комплекти; вогнегасники; пожежний інвентар (резервуари з водою, ящики з піском, пожежні відра, лопати);

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		63

переносний пожежний інструмент (кирки, сокири, багри, ломи тощо).

Для гасіння пожеж промисловість випускає різноманітні типи вогнегасників. Найбільшого поширення набули водопінні, водяні, газові (вуглекислотні) та порошкові вогнегасники. З точки зору ефективності пожежогасіння, універсальності та економічності найбільш перспективними вважаються порошкові вогнегасники. Первинні засоби пожежогасіння розміщують на пожежних щитах, які встановлюють на виробничій території з розрахунку один щит на кожні 5000 м². Пожежні щити фарбують у червоний колір.

Згідно з вимогами Правил пожежної безпеки, на кожному поверсі будівлі адміністративного призначення повинно бути встановлено не менше двох вогнегасників з масою заряду вогнегасної речовини 5 кг і більше. Використання вогнегасників без призначення відповідального за їх обслуговування не допускається. На підприємстві забороняється палити у невідведених для цього місцях. Зберігання легкозаймистих матеріалів (наприклад, паперу) ближче ніж: 1 метр від електрощитів; 0,15 м від приладів центрального водяного опалення; 0,6 м від сповіщувачів автоматичної пожежної сигналізації - заборонено.

Документація повинна зберігатися у спеціально відведених для цього шафах. Для запобігання розповсюдженню пожежі на об'єкті встановлюються автоматичні протипожежні системи, які складаються з: датчиків пожежної сигналізації; звукових сповіщувачів; аварійних кнопок; приймально-контрольної панелі, що обробляє сигнали від датчиків і передає відповідну інформацію на пульт пожежної охорони. Протипожежна сигналізація призначена для раннього виявлення пожежі та забезпечення своєчасного реагування. Крім установки пожежної сигналізації, підприємство повинно укласти договір з ліцензованою організацією на технічне обслуговування цієї системи. До обслуговування входять регламентні роботи, передбачені нормами, а також усунення несправностей. Періодичність перевірок узгоджується із замовником, але має проводитися не рідше одного разу на місяць. Організувати зустріч підрозділів пожежної охорони та надати їм консультаційну та іншу допомогу під час ліквідації пожежі.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		64

ВИСНОВКИ

У ході виконання даного дипломного проекту було розроблено та впроваджено програму для генерації текстових повідомлень, яка використовує алгоритми штучного інтелекту. Основні аспекти та результати проекту включають такі пункти. Було створено програмний продукт на базі штучного інтелекту, який аналізує вхідні дані та генерує текстові повідомлення. Алгоритм використовує моделі машинного навчання та обробки природної мови для створення змістовних та контекстуально релевантних текстів. Розроблено зручний і інтуїтивно зрозумілий інтерфейс користувача, який дозволяє легко вводити вхідні дані та отримувати відповіді від системи. Інтерфейс підтримує різноманітні функціональні можливості, такі як налаштування параметрів генерації та перегляд історії повідомлень.

Точність та релевантність: Проведено оцінку якості генерованих текстів, яка показала високий рівень точності та релевантності повідомлень до заданих тем та контекстів. Використання передових технік машинного навчання дозволило оптимізувати процес створення текстів.

Впроваджені сучасні методи захисту даних забезпечують безпеку введеної інформації та генерованих повідомлень. Програма гарантує дотримання політик конфіденційності та етичних норм.

Система була протестована на різних типах вхідних даних, що дозволило оцінити її універсальність і здатність адаптуватися до різних умов використання. Тестування підтвердило стабільність та ефективність програми при великому навантаженні. Можливості масштабування: Програма розроблена з можливістю легкого масштабування, що дозволяє збільшувати кількість одночасних користувачів та обсяги обробки даних без втрати продуктивності.

Реалізація цього проекту демонструє значний потенціал застосування штучного інтелекту у створенні програм для генерації текстових повідомлень, що може бути корисним у багатьох областях, від бізнес-комунікацій до освітніх технологій.

					<i>КГ 08. 14 001. 00 ДП ПЗ</i>	Арк.
Ізм.	Лист	№ докум.	Підпис	Дата		65

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Барановська Т. П. Штучний інтелект: сучасні технології та перспективи розвитку // Наукові праці ОНАХТ. – 2021. – № 3. – С. 45–52.
2. Бровко П. А. Машинне навчання і його застосування в інтелектуальних системах // Інформаційні технології і комп'ютерна інженерія. – 2020. – № 1(55). – С. 78–84.
3. Глущенко О. В. Етичні аспекти застосування штучного інтелекту в цифрових сервісах // Етика і культура в сучасному цифровому суспільстві. – 2021. – № 1. – С. 59–64.
4. Дьяконов А. В. Моделі обробки природної мови у прикладних системах III // Системи обробки інформації. – 2021. – Вип. 3 (160). – С. 97–102.
5. Ковальчук С. І. Нейромережі та глибоке навчання: основи та практичні застосування // Вісник Київського національного університету імені Тараса Шевченка. Серія: Кібернетика. – 2022. – № 1. – С. 12–19.
6. Пономаренко А. М. Архітектура систем зі штучним інтелектом: проектування, масштабування, оптимізація // Інформаційні технології та комп'ютерна інженерія. – 2022. – № 4. – С. 50–57.
7. Семенченко В. М. Безпека інформації в інтелектуальних системах // Інформаційна безпека людини, суспільства, держави. – 2021. – № 2. – С. 34–40.
8. Стороженко Ю. Б. Інтерфейси користувача для інтелектуальних систем: usability та адаптивність // Комп'ютерні науки та інформаційні технології. – 2020. – № 2(12). – С. 21–26.
9. 9 Brown T. et al. Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – P. 1877–1901.
10. Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press, 2016. – 775 р.

Додаток А. Програмний для Vue + FontAwesome + Social

Sharing

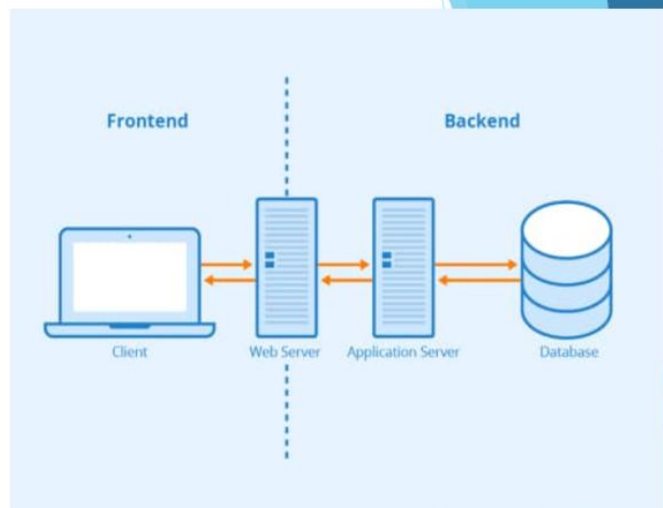
```
import './assets/main.scss'
import { FontAwesomeIcon } from '@fortawesome/vue-fontawesome'
import VueSocialSharing from 'vue-social-sharing'
import { createApp } from 'vue'
import { library } from '@fortawesome/fontawesome-svg-core'
import App from './App.vue'
import { faUserSecret } from '@fortawesome/free-solid-svg-icons'
import {
  faTwitter,
  faFacebook,
  faLinkedin,
  faTelegram
} from '@fortawesome/free-brands-svg-icons'
import { fas } from '@fortawesome/free-solid-svg-icons'
const app = createApp(App);
// Додаємо потрібні іконки у бібліотеку
library.add(faUserSecret, faTwitter, faFacebook, faLinkedin, faTelegram, fas);
// Регіструємо компонент іконок
app.component('font-awesome-icon', FontAwesomeIcon);
// Підключаємо плагін соціального поширення
app.use(VueSocialSharing);
// Монтуємо застосунок у DOM
app.mount('#app');
```

ДИПЛОМНИЙ ПРОЕКТ НА ТЕМУ:

Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту.

- ▶ Виконала студентка гр.4КГ-08 Левшанова В.А.

Взаємозв'язок між frontend і backend-розробкою



Скріншот коду файлу `server.js`

```
l > src > # server.js > create
const app = express();
app.use(helmet());
const uncaughtException = async () => {
  process.on('uncaughtException', (err) => {
    console.error('uncaughtException');
    console.error(err);
  });
};

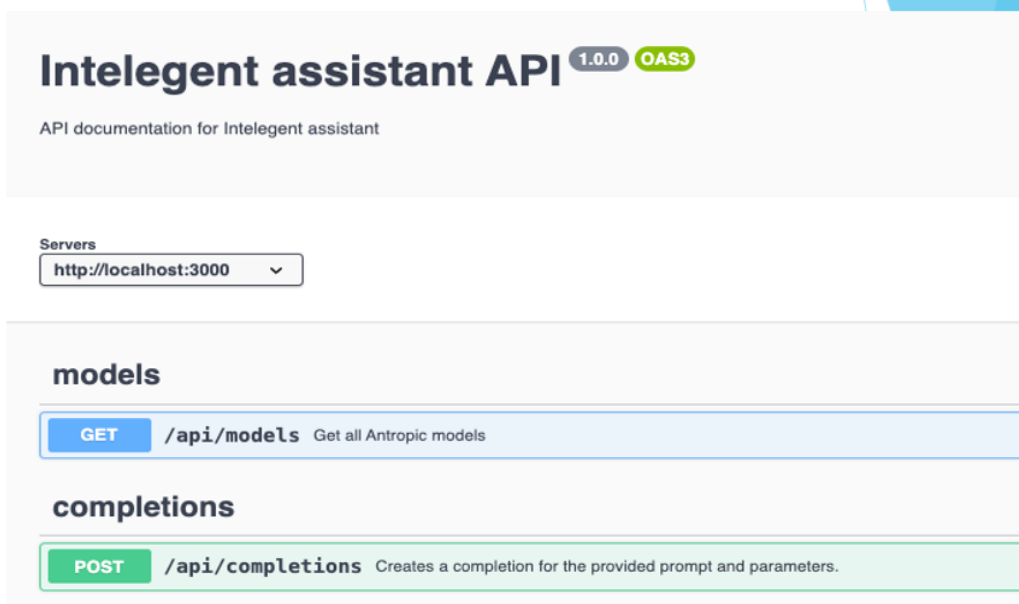
export const create = async () => {
  app.use(cors());
  app.use(pinoHttp({ level: 'error' }));
  app.use(
    bodyParser.urlencoded({
      extended: true,
    }),
  );
  app.use(bodyParser.json());
  app.get('/', (req, res) => {
    res.send('Ok');
  });
  app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerDocument));
  app.post('/api/completions', async (req, res) => {
    try {
      const prompt = req.body;
      const result = await getAnthropicMessages(prompt);
      res.json(result);
    } catch (error) {
      res.status(500).send(String(error));
    }
  });
};
```

Скріншот функції взаємодій з API Anthropic

```
src > controllers > # anthropic.js >
import Anthropic from '@anthropic-ai/sdk';
import dotenv from 'dotenv';
dotenv.config();
const apiKey = process.env.ANTHROPIC_API_KEY;

export const getAnthropicMessages = async ({ systemContent, userContent }) => {
  const anthropic = new Anthropic({ apiKey });
  const content = `
System:
  ${systemContent}
User:
  ${userContent}
`;
  const msg = await anthropic.messages.create({
    model: 'claude-3-opus-20240229',
    max_tokens: 1024,
    messages: [
      { role: 'user', content }
    ],
  });
  return msg.content.map(
    (cnt, i) => ({ text: cnt.text, index: i });
  );
};
```

Скріншот Intelligent Assistant API в Swagger



Скріншот структури проекту:

Папки та Файли:

frontend:

node_modules: містить всі залежності проекту

api.js: Модуль для взаємодії з зовнішніми API.

App.vue: Основний кореневий компонент Vue.

main.js: Точка входу для ініціалізації Vue додатка.

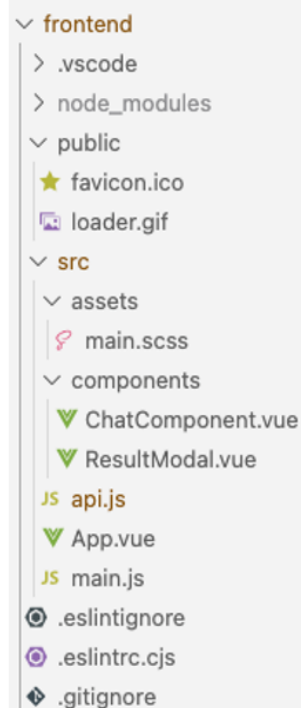
assets:

main.scss: Файл SCSS для глобальних стилів.

components:

ChatComponent.vue: Vue компонент для чат-функціоналу.

ResultModal.vue: Vue компонент для модальних вікон, які відображають результати або інформацію.



Скріншот вікна програми

Інтелектуальний помічник
Створіть повідомлення на всі випадки

Мова	Тип повідомлення
Українська	Не визначений
Стиль повідомлення	Тип виведення
Неформальний	Текст

Контекст події (опишіть ситуацію)

Показати результат


Скріншот результат роботи

Інтелектуальний помічник
Створіть повідомлення на всі випадки

Мова	Тип повідомлення
Українська	Не визначений
Стиль повідомлення	Тип виведення
Неформальний	Текст

Контекст події (опишіть ситуацію)

Ваш результат:

Привіт! Як справи? Чим займаєшся? 

Редагувати запит

Скріншот результат завдання

Інтелектуальний помічник
Створюйте повідомлення на всі випадки

Мова: Українська
Тип повідомлення: Попередження

Стиль повідомлення: Формальний
Тип виведення: Текст

Контекст події (опишіть ситуацію)
Вітаю Я сьогодні прихворіла і не зможу зустрітись з Вами.

Показати результат

Скріншот виконання роботи завдання: Написати рекламу для нашого коледжу.

Інтелектуальний помічник
Створюйте повідомлення на всі випадки

Мова: Українська
Тип повідомлення: Оголошення

Стиль повідомлення: Формальний
Тип виведення: Текст

Контекст події (опишіть ситуацію)
Напиши рекламу для Одеського технічного коледжу для абітурієнтів

Показати результат

Скріншот виконання завдання: Реклама коледжа.

Ваш результат:

ОДЕСЬКИЙ ТЕХНІЧНИЙ КОЛЕДЖ ОГОЛОШУЄ НАБІР АБІТУРІЄНТІВ

Шановні абітурієнти! Одеський технічний коледж запрошує вас стати частиною нашої престижної освітньої установи. Ми пропонуємо високоякісну освіту в галузі технічних наук, що відповідає сучасним вимогам ринку праці. Наш коледж має потужну матеріально-технічну базу, що включає сучасні лабораторії, комп'ютерні класи та майстерні. Викладацький склад коледжу налічує висококваліфікованих фахівців, які забезпечать вам ґрунтовну теоретичну та практичну підготовку.

Навчання в Одеському технічному коледжі відкриває широкі перспективи для подальшого професійного розвитку та кар'єрного зростання. Наші випускники є затребуваними спеціалістами на ринку праці та успішно працюють у провідних компаніях галузі. Ми пропонуємо такі спеціальності: - Комп'ютерна інженерія - Електроніка та телекомунікації - Автоматизація та комп'ютерно-інтегровані технології - Метрологія та інформаційно-вимірювальна техніка - Галузеве машинобудування

Подайте документи до Одеського технічного коледжу та станьте частиною нашої дружньої студентської спільноти! Ми чекаємо на вас та готові допомогти реалізувати ваші мрії та амбіції. За детальнішою інформацією звертайтеся до приймальної комісії Одеського технічного коледжу. Наша адреса: м. Одеса, вул. Балківська, 54. Телефон: (048) 722-00-00 Веб-сайт: www.otk.od.ua

Редагувати запит

Створення повідомлення на всі випадки

Інтелектуальний помічник

Створюйте повідомлення на всі випадки

Мова

Українська

Тип повідомлення

Не визначений

Стиль повідомлення

Офіційно-діловий

Тип виведення

Текст

Контекст події (опишіть ситуацію)

Завтра нарада в 10 ранку

Показати результат

Скріншот виконання роботи завдання: повідомлення на всі випадки

Ваш результат:

Шановні колеги, Хочу нагадати, що завтра о 10:00 відбудеться важлива нарада. Прошу всіх бути присутніми вчасно та підготувати необхідні матеріали для обговорення. Якщо у вас виникли непередбачені обставини і ви не зможете бути присутніми, будь ласка, повідомте про це завчасно. Дякую за вашу увагу та співпрацю. Очікую на зустріч з вами завтра. З повагою, [Ваше ім'я]

Редагувати запит

Скріншот виконання іноземною мовою

Ваш результат:

Dear [Recipient], Please be advised that a meeting has been scheduled for tomorrow morning at 10:00 AM. Your attendance is required. Kindly ensure that you are well-prepared for the meeting and arrive on time. If you have any conflicts with this scheduled time, please inform me as soon as possible so that alternative arrangements can be made if necessary. Thank you for your attention to this matter. I look forward to your presence at the meeting tomorrow. Best regards, [Your Name]

Редагувати запит

РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Левшанової Вікторії Андріївни

(прізвище, ім'я та по батькові)

Спеціальність 123 Комп'ютерна інженерія

Освітньо-професійна програма Комп'ютерна графіка і Web-дизайн

Керівник дипломного проекту (роботи) Шувалова Ірина Олегівна

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту

Обсяг розрахунково-пояснювальної записки 75 сторінок

Обсяг графічної (презентаційної) частини 15 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню

Робота відповідає технічному завданню до дипломного проекту. Виконана у відповідності з вимогами

б) характеристика виконання кожного розділу дипломного проекту (роботи)

При виконанні дипломного проекту здобувачка продемонструвала уміння використовувати останні досягнення науки та техніки, уміння працювати з літературою. Так, здобувачка виконала усі етапи проектування застосунку для генерації повідомлень із використанням технологій штучного інтелекту.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту (роботи)

Пояснювальна записка та графічна частина відповідає вимогам, виконана якісно та відображає основні елементи проектування застосунку для генерації повідомлень із використанням технологій штучного інтелекту.

г) перелік позитивних якостей дипломного проекту (роботи) _____

Тема дипломного проекту є актуальною, виконана у достатньому обсязі, якісно, відповідно до поставленого завдання. Реалізація цього проекту демонструє значний потенціал застосування штучного інтелекту у створенні програм для генерації текстових повідомлень, що може бути корисним у багатьох областях, від бізнес-комунікацій до освітніх технологій.

д) основні недоліки дипломного проекту (роботи) _____

Обмежена підтримка офлайн-режиму: якщо система працює тільки в онлайні, це може знизити її доступність у середовищах з нестабільним інтернетом. Залежність від зовнішніх API або хмарних сервісів, якщо система використовує зовнішні ШІ-сервіси (OpenAI), то це створює ризик у разі зміни тарифів або політик доступу. У завданні згадується NestJS та MongoDB, але в реалізації використано Express і жодної роботи з БД не показано.

Оцінка розрахункової частини _____ Добре

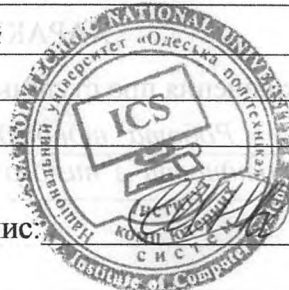
Оцінка графічної частини _____ Добре

Загальна оцінка _____ Добре

Прізвище, ім'я, по батькові рецензента _____ к.т.н. Шибасєва Наталя Олегівна

Місце роботи і посада рецензента _____ Національний університет «Одеська політехніка», доцент
кафедри інформаційних технологій

Підпис: _____



« 27 » 06 2025 р.

ВІДГУК

керівника на дипломний проєкт здобувача (здобувачки) освіти
відділення комп'ютерних систем

Левшанової Вікторії Андріївни

(прізвище, ім'я та по батькові)

Спеціальність: 123 "Комп'ютерна інженерія"

Освітньо-професійна програма: «Комп'ютерна графіка і Web-дизайн»

Тема дипломного проєкту: Розробка web-застосунку для генерації повідомлень
із використанням технологій штучного інтелекту

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЄКТУ

а) обсяг і якість виконання проєкту (графічного матеріалу і розрахунково-пояснювальної записки)

Дипломний проєкт виконано відповідно технічному завданню. Пояснювальна записка містить 75 сторінки. У пояснювальній записці виконано опис етапів, відповідно до індивідуального завдання, розділи пояснювальної записки відповідають етапам рішення завдання, поставленого у дипломному проєкті. Графічна частина складається з 15 слайдів мультимедійної презентації. Якість виконання пояснювальної записки та графічної частини добра, розробку виконано в повному обсязі.

б) самостійність роботи над проєктом: Протягом всього строку дипломного проєктування та переддипломної практики здобувачка освіти Левшанова Валерія Андріївна поступово та послідовно виконувала всі етапи розробки. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника

в) теоретична підготовка випускника (випускниці): Здобувачка освіти Левшанова Валерія Андріївна під час роботи над дипломним проєктом вивчила достатню кількість літературних джерел та матеріалів за даною тематикою.

Вважаю, що теоретична підготовка дипломниці добра і вона готова до захисту дипломного проєкту

г) вміння розв'язувати виробничі та конструкторські питання _____
Під час дипломного проектування здобувачка освіти Левшанова В.А. розробила web-застосунок для генерації повідомлень із використанням технологій штучного інтелекту. Дипломниця володіє навичками написання веб-застосунку за допомогою таких технологій як технологія Swagger, MongoDB, мова програмування JavaScript.

Оцінка розрахункової частини _____ Відмінно
Оцінка графічної частини _____ Відмінно
Загальна оцінка _____ Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту _____
Шувалова Ірина Олегівна

Місце роботи і посада керівника дипломного проекту Викладач комісії
КТ та ІІ ВСП «Одеський технічний фаховий коледж ОНТУ»

Підпис _____

«24» _____ 06 2025 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Левшанова Вікторія Андріївна
здобувачка освіти гр. 4КГ-08, та

Шувалова Ірина Олегівна,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту» (автор роботи – Левшанова В.А., керівник роботи – Шувалова І.О.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Левшанова В.А. /

Керівник



/ Шувалова І.О. /

«16» червня 2025 р.

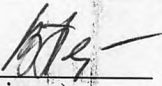
ДОВІДКА

циклової комісії КТ та ПІ
про допуск до захисту дипломного проекту
здобувача (здобувачки) освіти ІV курсу
відділення комп'ютерних систем групи 4КГ-08

Левшанової Вікторії Андріївни

на тему Розробка web-застосунку для генерації повідомлень
із використанням технологій штучного інтелекту

Висновок відповідальної особи за проведення нормоконтролю:
пояснювальна записка до дипломного проекту виконана з деякими
порушеннями ДСТУ та оформлена відповідно до вимог Положення про
дипломне проектування


(підпис)

18.06.2025
(дата)

Петрашова В.І.
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного
плагіату згідно звіту про перевірку від 14.06.2025 р. значення коефіцієнту
подібності в роботі становить 18,24%, коефіцієнт цитування – 6,28%.


(підпис)

18.06.2025
(дата)

Краснокутська К.Г.
(П.І.Б.)

Попередня експертиза (малий захист) дипломного проекту

здобувача (здобувачки) освіти

Левшанової В.А.
(П.І.Б.)

проведена « 18 » червня 2025 р.

Висновки Пояснювальна записка до дипломного проекту виконана у повному
обсязі. Випускна кваліфікаційна робота (дипломний проект) відповідає
вимогам Положення про дипломне проектування та рекомендована до
захисту.

Голова ЦК КТ та ПІ


(підпис)

Кривченко Ю.В.
(П.І.Б.)

Звіт подібності

Метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту

Автор

Науковий керівник / Експерт

Левшанова Вікторія Андріївна | Шувалова Ірина Олегівна

Підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фраз для коефіцієнта подібності 2



10261

Кількість слів

80585

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про **МОЖЛИВІ** маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		17
Інтервали		0
Мікропробіли		3
Білі знаки		0
Парафрази (SmartMarks)		77

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копія тексту означає, в якому джерелі він був знайдений. Ці джерела і значення коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Копія тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/browse/dateissued?scope=a11b717f-f1ce-43e0-a64e-be99f24c1255	179 1.74 %
2	https://card-file.ontu.edu.ua/bitstreams/e4afae26-0a7e-4a4d-afc2-94341838de2a/download	135 1.32 %
3	https://card-file.ontu.edu.ua/bitstreams/e4afae26-0a7e-4a4d-afc2-94341838de2a/download	66 0.66 %
4	https://card-file.ontu.edu.ua/bitstreams/e4afae26-0a7e-4a4d-afc2-94341838de2a/download	56 0.56 %
5	https://card-file.ontu.edu.ua/bitstreams/e4afae26-0a7e-4a4d-afc2-94341838de2a/download	50 0.49 %

4	https://card-file.ontu.edu.ua/bitstreams/7b1e10b9-0ac2-4b07-afc4-8cdf7db780/download	51 0.44 %
5	https://card-file.ontu.edu.ua/bitstreams/549ee9fe-7574-4ae5-b500-9fe2711f33e6/download	50 0.43 %
6	https://card-file.ontu.edu.ua/bitstreams/538ada8a-2c79-4b1e-b7d2-b0c97f68bc1c/download	49 0.42 %
7	https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-460f-9c49-cba3f9d50554/content	47 0.41 %
8	https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16a8-4070-bead-5562769b7c71/download	38 0.33 %
9	https://card-file.ontu.edu.ua/bitstreams/538ada8a-2c79-4b1e-b7d2-b0c97f68bc1c/download	37 0.32 %
10	https://card-file.ontu.edu.ua/bitstreams/538ada8a-2c79-4b1e-b7d2-b0c97f68bc1c/download	33 0.28 %

з домашньої бази даних (0.85 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Створення web-застосунку цифрового помічника з використанням Open AI <div style="background-color: #cccccc; padding: 2px;">[REDACTED]</div> 5/28/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	98 (6) 0.85 %

з програми обміну базами даних (0.71 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Інформаційна система «Адміністрування готельної мережі» <div style="background-color: #cccccc; padding: 2px;">[REDACTED]</div> 3/16/2025 National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute (National Technical University of Ukraine Igor Sikorskyi Kyiv Politech Institute)	18 (2) 0.16 %
2	Дорогань_А_Ю_Мірастр_Пояснювальна_записка_23112024 <div style="background-color: #cccccc; padding: 2px;">[REDACTED]</div> 11/25/2024 O.M.Beketov National University of Urban Economy in Kharkiv (ННІБЦІ)	16 (1) 0.14 %
3	Fesenko_mag_rob.docx <div style="background-color: #cccccc; padding: 2px;">[REDACTED]</div> 5/17/2021 Sumy State University (Кафедра інформаційних технологій)	15 (1) 0.13 %
4	Панченко.docx <div style="background-color: #cccccc; padding: 2px;">[REDACTED]</div> 6/12/2019 Sumy State University (Кафедра інформаційних технологій)	14 (1) 0.12 %
5	Ptashkin_diplom_bak_Osypchuk_fkn_2023.pdf <div style="background-color: #cccccc; padding: 2px;">[REDACTED]</div> 6/7/2023 V. N. Karazin Kharkiv National University (KKNU) (ННІ комп'ютерних наук та штучного інтелекту - кафедра комп'ютерних систем та робототехніки)	13 (1) 0.11 %
6	2016_6050101_Prodeus_Roman_Mykhailovych_9319 <div style="background-color: #cccccc; padding: 2px;">[REDACTED]</div> 10/25/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	6 (1) 0.05 %

з Інтернету (10.54 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/server/api/core/bitstreams/d5a3d14f-d5cb-460f-9c49-cba3f9d50554/content	211 (10) 1.82 %
2	https://card-file.ontu.edu.ua/bitstreams/538ada8a-2c79-4b1e-b7d2-b0c97f68bc1c/download	210 (10) 1.81 %

8	https://tnpu.edu.ua/about/public_inform/upload/Vykonannia_pratypozhezhenykh_zakhodiv.pdf	34 (2) 0.33 %
9	https://card-file.ontu.edu.ua/bitstreams/0e72a3b9-bdd7-4711-a3c6-dcdc1d4287cc/download	32 (2) 0.31 %
10	https://card-file.ontu.edu.ua/bitstreams/035f6436-20b4-4ee6-8e99-bede670e308b/download	31 (1) 0.30 %
11	https://elar.naiu.kiev.ua/bitstreams/7c96eef3-1b96-49cb-9889-5ed950cd0bb6/download	31 (1) 0.30 %
12	https://card-file.ontu.edu.ua/bitstreams/6cf43324-8f08-4031-ba42-f80b18efbbc8/download	24 (1) 0.23 %
13	https://dnaop.com/html/1948/doc-instrukcija-pro-zahodi-pozhezhoji-bezpeki-u-primishhenyah-ohoroni	17 (1) 0.17 %
14	https://ua-referat.com/?red=66399	12 (1) 0.12 %
15	https://zhatk.zl.ua/wp-content/uploads/2021/06/instrukciya-z-pozhezho%01%97-bezpeki-dlya-sluzhbovih-primishhen-zhatk..pdf	12 (1) 0.12 %
16	https://card-file.ontu.edu.ua/server/api/core/bitstreams/214d43de-5031-4ab6-849f-efa001b5416b/content	12 (1) 0.12 %
17	https://card-file.ontu.edu.ua/browse/author?scope=e5a3b909-ff7f-4df8-b9fa-5f8617c2f288&value=%D0%A1%D1%96%D0%BB%D0%B0%D1%94%D0%B2%20%D0%86%D0%B3%D0%BE%D1%80%D1%8C%20%D0%93%D0%B5%D0%BD%D0%B0%D0%B4%D1%96%D0%B9%D0%BE%D0%B2%D0%B8%D1%87	10 (1) 0.10 %
18	https://essuir.sumdu.edu.ua/bitstream/123456789/86882/1/Kravchenko_mag_rob.pdf	10 (1) 0.10 %
19	https://card-file.ontu.edu.ua/bitstreams/538ada8a-2c79-4b1e-b7d2-b0c97f68bc1c/download	9 (1) 0.09 %
20	https://www.morningdough.com/uk/ai-tools/best-ai-lead-generation-software/	8 (1) 0.08 %
21	https://card-file.ontu.edu.ua/bitstreams/63ee88cb-a3d0-4005-9cf2-0cff89f28c0d/download	8 (1) 0.08 %

Список прийнятих фрагментів (немає прийнятих фрагментів)

ПОРЯДКОВИЙ НОМЕР	ЗМІСТ	КІЛЬКІСТЬ ЄДИНАКОВИХ ЄДИНЦІВ (ФРАГМЕНТІВ)
------------------	-------	---

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП **ОДЕСЬКИЙ ТЕХНІЧНИЙ ЖАХОВИЙ КОЛЕДЖ ОНТУ**

Спеціальність: 123 **Комп'ютерна інженерія**

Освітня програма: **Комп'ютерна графіка та Web-дизайн**

Група: 4КГ-08

Дипломний проєкт здобувача освіти даної форми навчання **КГ_08_14_000_ДП**

Левшанової

Вікторії Андріївни

м. Одеса

2025_р. МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ ВСП **ОДЕСЬКИЙ ТЕХНІЧНИЙ ЖАХОВИЙ КОЛЕДЖ ОНТУ**

Спеціальність: 123 **Комп'ютерна інженерія**

Освітньо-професійна програма: **Комп'ютерна графіка та Web-дизайн**

Група: 4КГ-08

ПОЯСНЮВАЛЬНА ЗАПІСКА

до дипломного проєкту на тему:

Розробка web-застосунку для генерації повідомлень із використанням технологій штучного інтелекту

Проектний матеріал складається з пояснювальної записки на 75 сторінках та

графічного (перзентаційного) матеріалу на 15 аркушах (слайдах)

Дипломник _____ (Левшанова В. А.)

Керівник _____ (Шувалова І. В.)