

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

*Спеціальність: 121 «Інженерія програмного забезпечення»*

*Освітньо-професійна програма: «Розробка програмного забезпечення»*

*Група: 4РП-07*

# **Дипломний проект**

**здобувача освіти денної форми навчання  
РП.07.01.000.ДП**

***БОЙКА  
ВЛАДИСЛАВА  
ДМИТРОВИЧА***

**м. Одеса  
2024 р.**

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

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-07

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

### Розробка програми для генерації повідомлень за допомогою штучного інтелекту

Проектний матеріал складається з пояснювальної записки на 68 сторінках та графічного (перзентаційного) матеріалу на 15 аркушах (слайдах).

Дипломник Бойко (Бойко В.Д.)

Керівник Кунуп (Кунуп Т. В.)

#### Консультанти:

з економічного розділу Іванченков (Іванченков В. С.)

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

з нормоконтролю Петрашова (Петрашова В. І.)

старший консультант Кривченко (Кривченко Ю. В.)

#### До захисту допущений

Голова циклової комісії Кривченко (Кривченко Ю. В.)

Завідувач відділення Скорнякова (Скорнякова О. В.)

Захист «17» 06 2024 р.

Протокол ДКК № 1

Оцінка ДКК 4(добре)/755

Секретар ДКК Скорнякова

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

Відділення Ком'ютерних систем Комісія КТ та ПІ  
Спеціальність 121 – «Інженерія програмного забезпечення»  
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ

Заст. дир. з НВР Беркань І. В.  
« 15 » 01 2024 року

ЗАВДАННЯ

на дипломний проєкт (роботу)

Бойко Владиславу Дмитровичу

1. Тема проєкту (роботи) Розробка програми для генерації повідомлень за допомогою штучного інтелекту

ної  
вра

Затверджена наказом по коледжу від « 02 » 11 2023 р., наказ № 244-А2-ОД

2 Термін здачі закінченого проєкту (роботи) 10.06, 2024

ого

3. Вихідні дані до проєкту (роботи)  
Вивчення можливостей API моделей,

мова програмування JavaScript. з використанням платформи з відкритим кодом Nodejs, фреймворк для створення серверних додатків NestJS, фреймворк LangChain, OpenAi, технологія Swagger; MongoDB

ого  
і в  
ез

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.

го  
су

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)  
Презентація Power Point – 15 слайдів: Взаємозв'язк між frontend і backend-розробкою; Структура проєкта; Скриншот коїву файлів; Структура сервера; Скриншот функції взаємодії з API Anthropic. Скриншоти 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	Кунуп Т. В.		
Економічний розділ	Іванченков В. С.		
Розділ охорони праці	Чорновол Н. І.		
Нормоконтроль	Петрашова В. І.		
Старший консультант	Кривченко Ю. В.		

7. Дата видачі завдання 15.01.24

Керівник

Кунуп Т. В.

(підпис)

Завдання прийняв до виконання

Бойко В.Д.

(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	29.04.24	виконано
2	Розробка структури програми	01.05.24	виконано
3	Обґрунтування вибору розробки програми	03.05.24	виконано
4	Адаптація та налаштування моделей	05.05.24	виконано
5	Налаштування структури проекту	07.05.24	виконано
6	Розробка бекенд та фронтенд частини	09.05.24	виконано
7	Тестування створеного програмного продукту	15.05.24	виконано
8	Економічний розрахунок	17.05.24	виконано
9	Опис охорони праці та техніки безпеки	19.05.24	виконано
10	Аналіз результатів проектування.	21.05.24	виконано
11	Оформлення пояснювальної записки	23.05.24	виконано
12	Оформлення графічної (презентаційної)	30.05.24	виконано
13	частини	05.06.24	виконано
14	Підготовка доповіді для захисту	07.06.24	виконано
15	Малий захист дипломного проекту	10.06.24	виконано

Дипломник

(підпис)

Керівник

(підпис)



# ЗМІСТ

<b>ВСТУП</b>	7
<b>1. ОСНОВНА ЧАСТИНА</b>	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 Проєктування 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	24
1.6.2 Опис технології розробки фронтенд частини	27
1.7 Програмна реалізація програми «Інтелектуальний помічник»	28
1.7.1 Розробка Фронтенду	28
1.8 Опис роботи програми	47
<b>2. ЕКОНОМІЧНИЙ РОЗДІЛ</b>	55
2.1 Резюме	55
2.2. Визначення трудомісткості розробки програмного забезпечення	56
2.3 Розрахунок ціни програмного продукту	57
<b>3. ОХОРОНА ПРАЦІ</b>	61
3.1 Аналіз та безпека умов праці працівника на робочому місці	61

Зм.	Арк.	№ докум.	Підп.	Дата

3.2 Розробка заходів з охорони праці	62
3.3 Організація робочого місця користувача ПК	63
3.4 Пожежна безпека	64
<b>ВИСНОВКИ</b>	66
<b>ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ</b>	68
<b>Додаток А</b> Слайди мультимедійної презентації	69

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		6

# 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 2021 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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	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>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		9

## 2. Освітній сектор:

Студенти та викладачі: Для підготовки навчальних матеріалів, презентацій, 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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		10

Виявлення основних функціональних вимог: створення, редагування та відправлення повідомлень.

Збір нефункціональних вимог: швидкість роботи, зручність використання, безпека даних.

## 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.

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		11

## 2. Google AI (DeepMind)

PaLM : Масштабована 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 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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		12

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 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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		13

### 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:

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.

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		14

### 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.

### 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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		15

## 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:

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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		16

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 можу зробити такий підсумок:

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.

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		17

Claude-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:

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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		18

### 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 запити (опціонально).

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

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

`system (string)`: Системний промпт для встановлення контексту (опціонально).

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

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

Код запити на JavaScript:

```
import Anthropic from '@anthropic-ai/sdk';  
const anthropic = new Anthropic();  
await anthropic.messages.create({  
  model: "claude-3-opus-20240229",
```

					РП 07. 01 001. 00 ДП ПЗ	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		19

```
max_tokens: 1024,  
messages: [  
  {"role": "user", "content": "Hello, world"}  
]  
]);
```

Код відповіді:

```
{  
  "content": [  
    {  
      "text": "Hi! My name is Claude.",  
      "type": "text"  
    }  
  ],  
  "id": "msg_013Zva2CMHLNnXjNJKqJ2EF",  
  "model": "claude-3-opus-20240229",  
  "role": "assistant",  
  "stop_reason": "end_turn",  
  "stop_sequence": null,  
  "type": "message",  
  "usage": {  
    "input_tokens": 10,  
    "output_tokens": 25  
  }  
}
```

Основні можливості:

1. Підтримка тексту та зображень: API підтримує як текстовий, так і зображувальний вміст.
2. Гнучкість у параметрах: Можливість встановлення різних параметрів, таких як максимальна кількість токенів, температури генерації та використання інструментів.
3. Підтримка потокової передачі: Можливість отримання відповідей у режимі реального часу за допомогою серверних подій

					РП 07. 01 001. 00 ДП ПЗ	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		20

## 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:

1. Запит: Ви надсилаєте запит на створення повідомлення з параметром "stream": true.
2. Події: Сервер надсилає події типу message\_start, content\_block\_start, content\_block\_delta, content\_block\_stop, message\_delta, message\_stop та ping.
3. 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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		21

"stream": встановити true для потокової передачі відповідей.

Код реалізації на JavaScript:

```
import Anthropic from '@anthropic-ai/sdk';  
const client = new Anthropic();  
await client.messages.stream({  
  messages: [{role: 'user', content: "Hello"}],  
  model: 'claude-3-opus-20240229',  
  max_tokens: 1024,  
}).on('text', (text) => {  
  console.log(text);  
});
```

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.

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		22

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.

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

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		23

## 1.6 Технології, які використовуються в розробці програми

Я вирішив 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 ендпойнтів зручно використовувати Swagger, що спрощує роботу з документацією інтерфейсів.

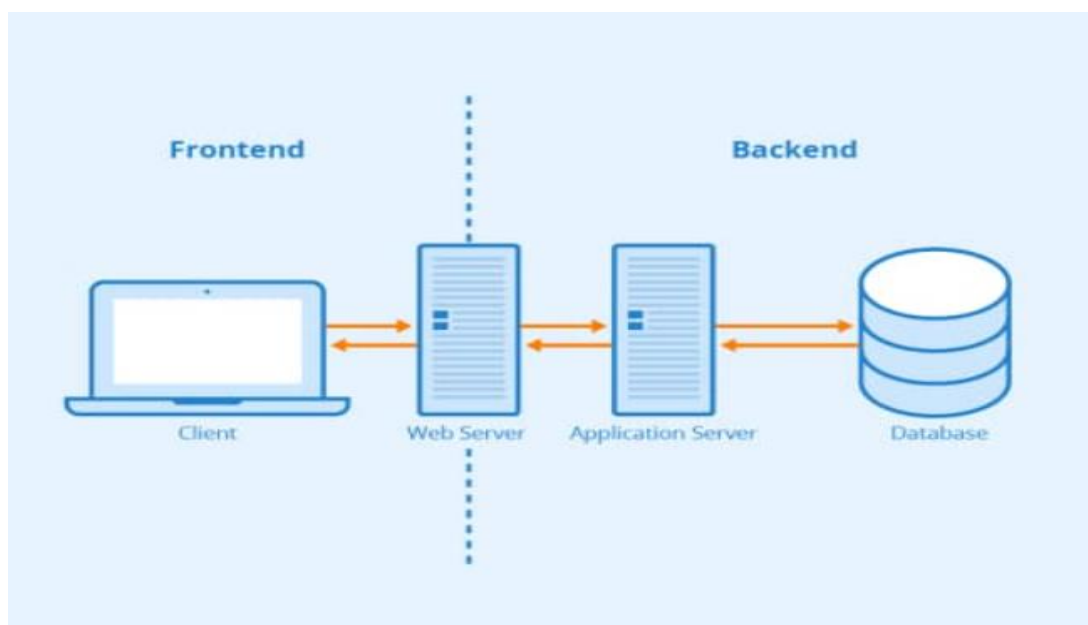


Рисунок 1.1. Взаємозв'язк між frontend і backend-розробкою

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.

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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		25

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.

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

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

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		26

## 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 web-додатку, з якою користувач може взаємодіяти і контактувати напряму. У Frontend входить відображення функціональних завдань призначеного для користувача інтерфейсу, що виконуються на стороні клієнта, а також обробка запитів користувачів. Фронтенд – це все те, що бачить користувач при відкритті web-сторінки. У свою чергу, web-

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		27

додаток – клієнт-серверний додаток, в якому клієнтом виступає в основному браузер, а сервером – web-сервер.

Логіка web-додатку розподілена між сервером і клієнтом, зберігання даних здійснюється переважно на сервері, обмін інформацією відбувається у мережі. Простіше кажучи, це те, що бачить користувач і які дії виконує кожен раз, коли підключається до мережі інтернет і відкриває будь-який браузер.

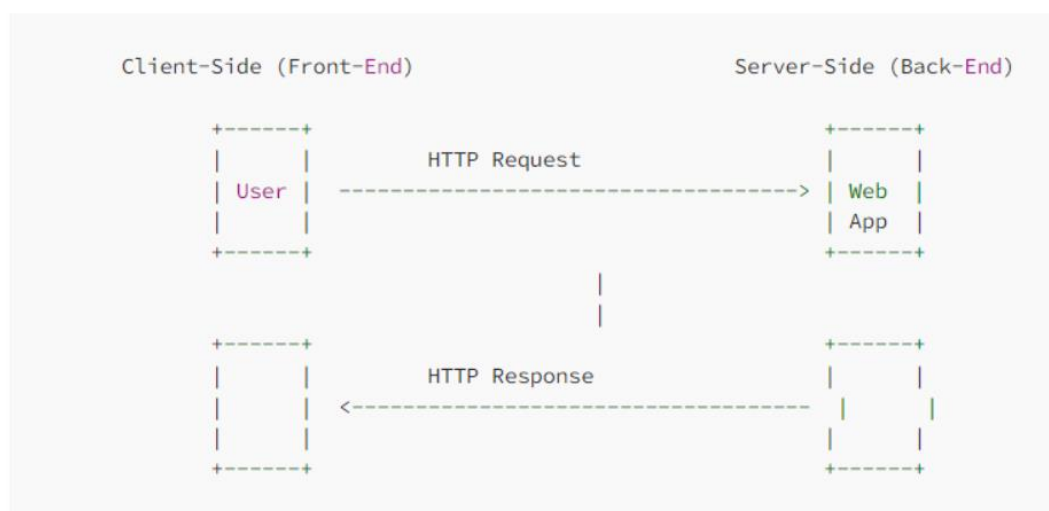


Рисунок 1.2. Спрощена схема взаємодії між frontend і backend

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.

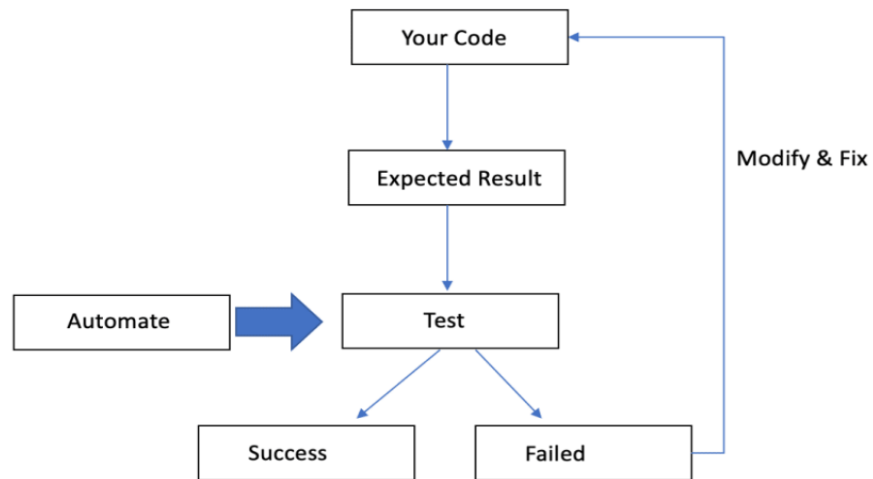


Рисунок 1.3. Плагін Vite для тестування Vue 3

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. Відвідайте сайт Node.js:

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

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		30

## 2. Виберіть версію для завантаження:

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

## 3. Завантажте інсталятор:

Натисніть на посилання інсталятора для Windows або macOS в залежності від вашої операційної системи. Це завантажить виконуваний файл (.msi для Windows і .pkg для macOS).

## 4. Запустіть інсталятор:

Після завершення завантаження запустіть інсталятор. Це розпочне майстер налаштування.

Для Windows: слідуйте інструкціям у майстру налаштування. Прийміть угоду про ліцензію, виберіть папку для встановлення і натискайте кнопку "Далі", поки не досягнете кнопки "Встановити".

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

## 5. Перевірте встановлення:

Відкрийте командний рядок (Windows) або термінал (macOS).

Введіть `node -v` і натисніть Enter. Це покаже версію Node.js, яка була встановлена, підтверджуючи встановлення.

Також перевірте `npm` (node package manager) за допомогою введення `npm -v` і натискання Enter.

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

### 1. Встановлюємо 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:

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		31

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

Після встановлення закрийте і знову відкрийте термінал, щоб почати використовувати nvm.

## 2. Перевіряємо встановлення NVM:

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

## 3. Встановіть Node.js за допомогою NVM:

Встановіть останню версію Node.js LTS, виконавши:

```
css
```

```
Copy code
```

```
nvm install --lts
```

Або ви можете встановити конкретну версію Node.js, вказавши версію:

```
Copy code
```

```
nvm install 20.14.0
```

## 4. Перемикайтеся між встановленими версіями Node:

Ви можете перемикатися між різними встановленими версіями, використовуючи:

```
nvm use 18.13.0
```

## 5. 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. Налаштування проекту:

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

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		32

## 2. Встановлення залежностей.

Залежності, які використовуються в проєкті, 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”):

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		33

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 браузері 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.

#### 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.

### 3. Реалізація сервера

У директорії "backend" створений файл server.js. У цьому файлі налаштує сервер Express.js і створить створім кінцеву точку /api/completions для обробки POST-запитів.

```
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 шаблоном 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

```
> Inteleagent 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.

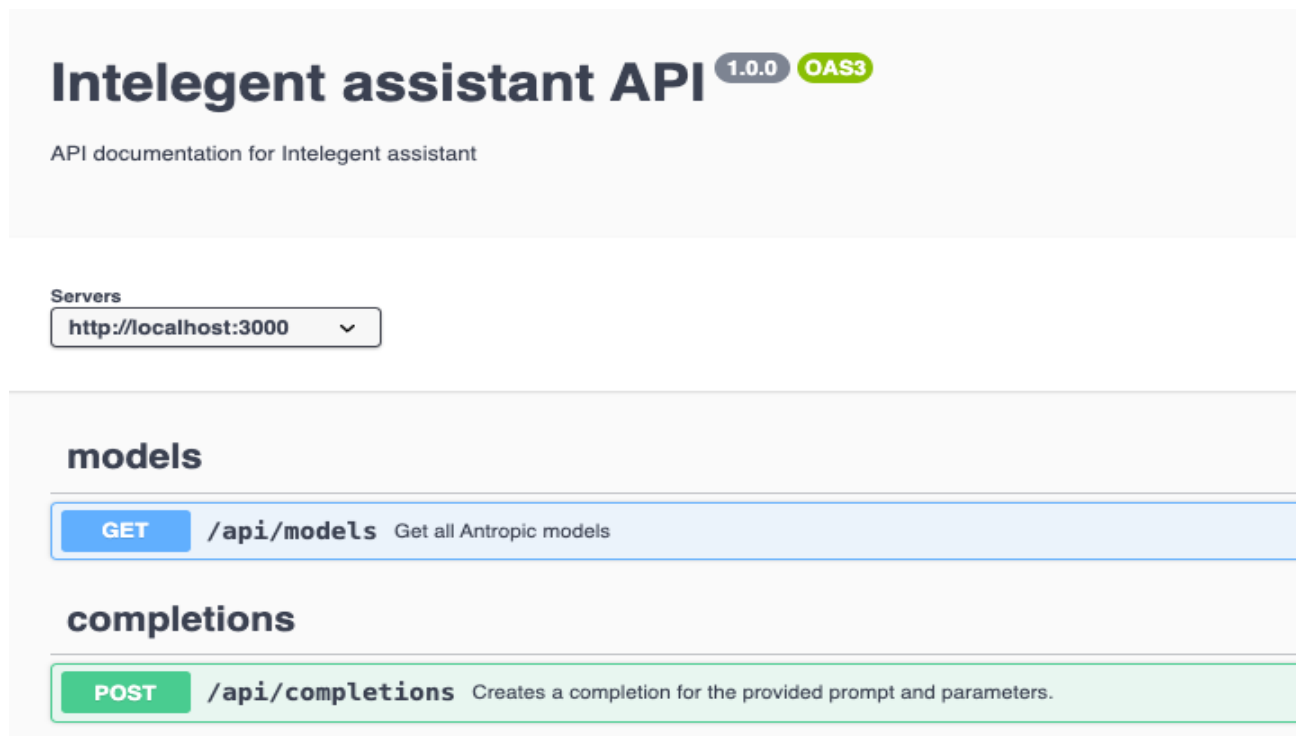


Рисунок 1.8. Скриншот Intelligent Assistant API в Swagger

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-ufZlgxLo8rreOeodwIteWxEQ-e64EMFP00RfLcegl1qCkTLIi_cU3w8VSR2eTduu
  8Y9htPb8rg-zohvigAA
3 PORT=3000
4 |
```

Рисунок 1.9. Скриншот 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:

1. 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”

					РП 07.01 001.00 ДП ПЗ	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		39

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

3. 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:

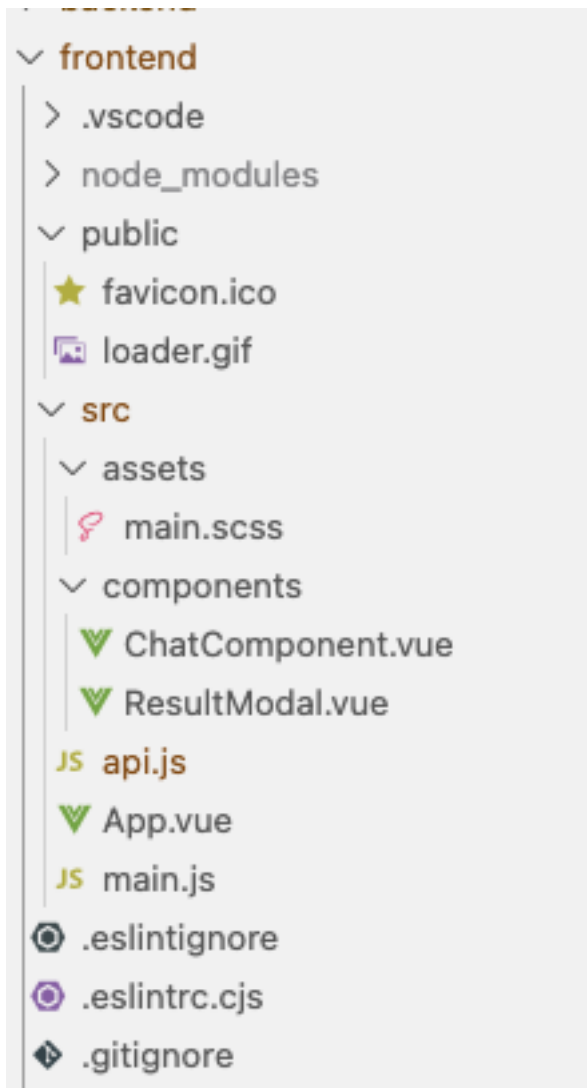


Рисунок 10. Скриншот структури проекту

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.

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;

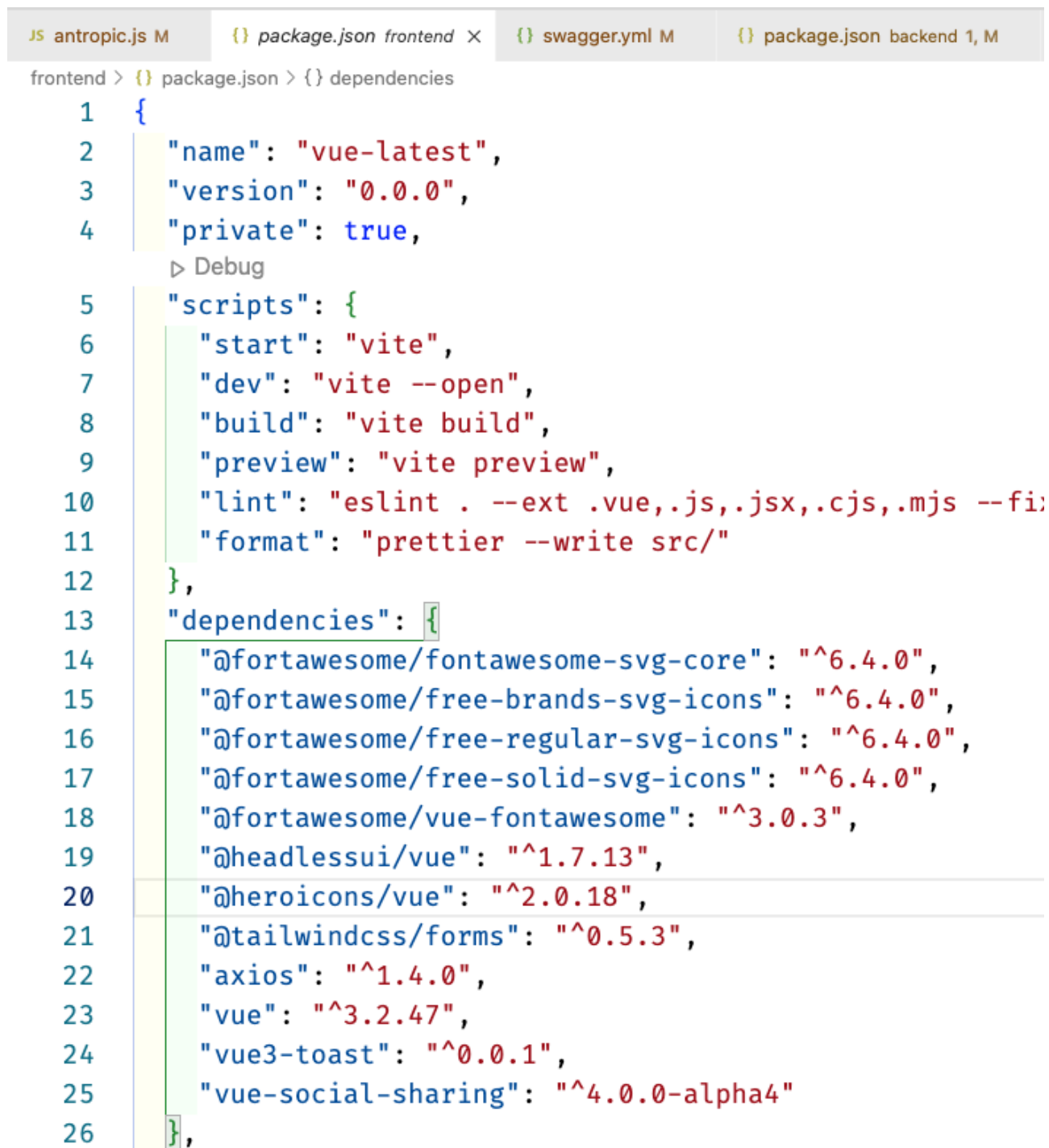
"`npm run dev`" - активує режим розробки для проекту;

"`npm run lint`" – проводить аналіз коду за допомогою лінера.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		42

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.



```
JS antropic.js M {} package.json frontend X {} swagger.yml M {} package.json backend 1, M
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. Скриншот 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. Скриншот файлу index.html:

Ось переклад на українську мову опису структури базового HTML документу, 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.

Усередині 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 типowo 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

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		45

```

> 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. Скриншот 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 компонентах 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.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		46



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**  
Створіть 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 запити користувача.

Вміст вікна:

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

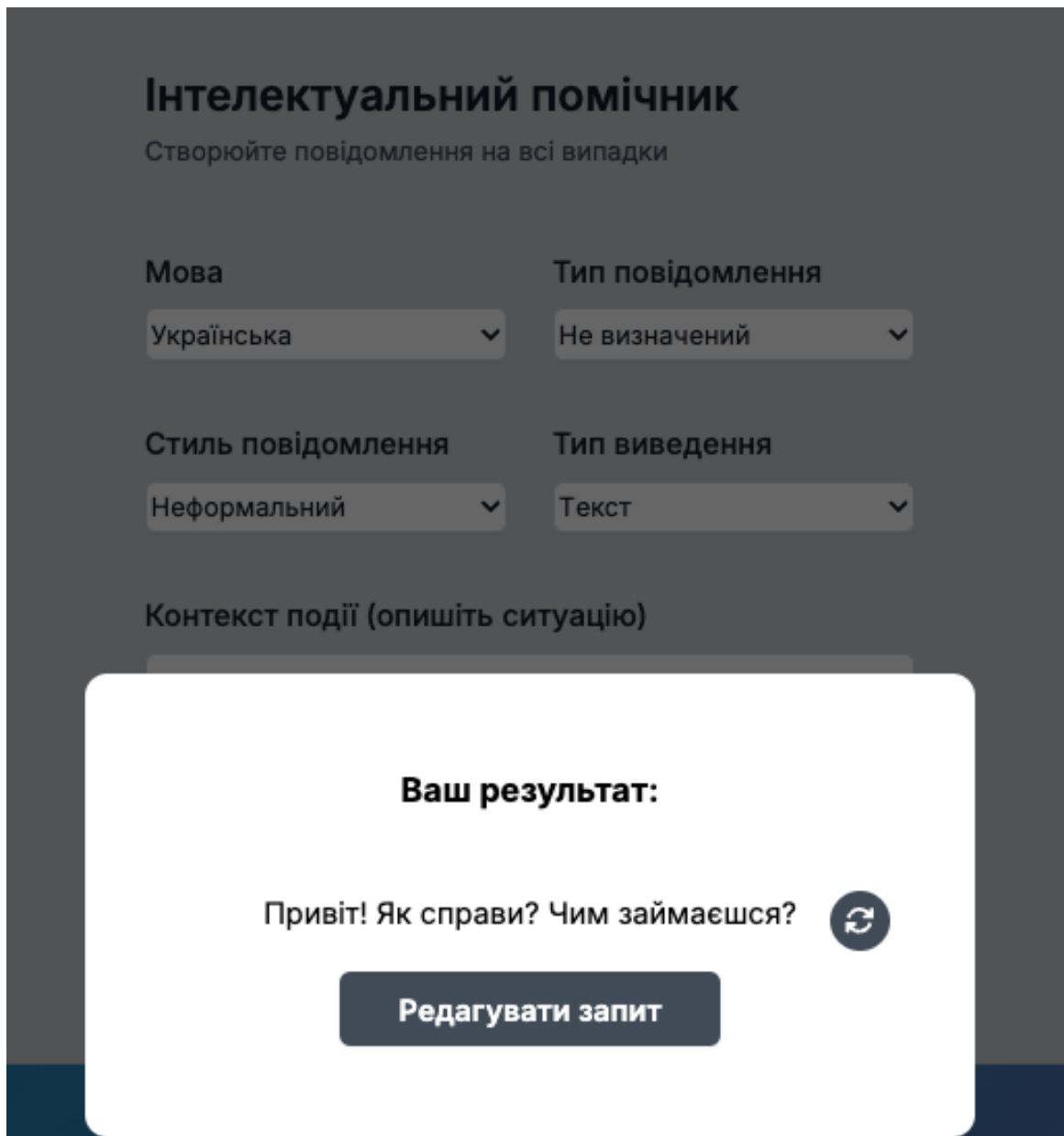


Рисунок 1.16. Скриншот результат роботи

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. Скриншот результат роботи завдання



Рисунок 1.18. скриншот виконання роботи завдання

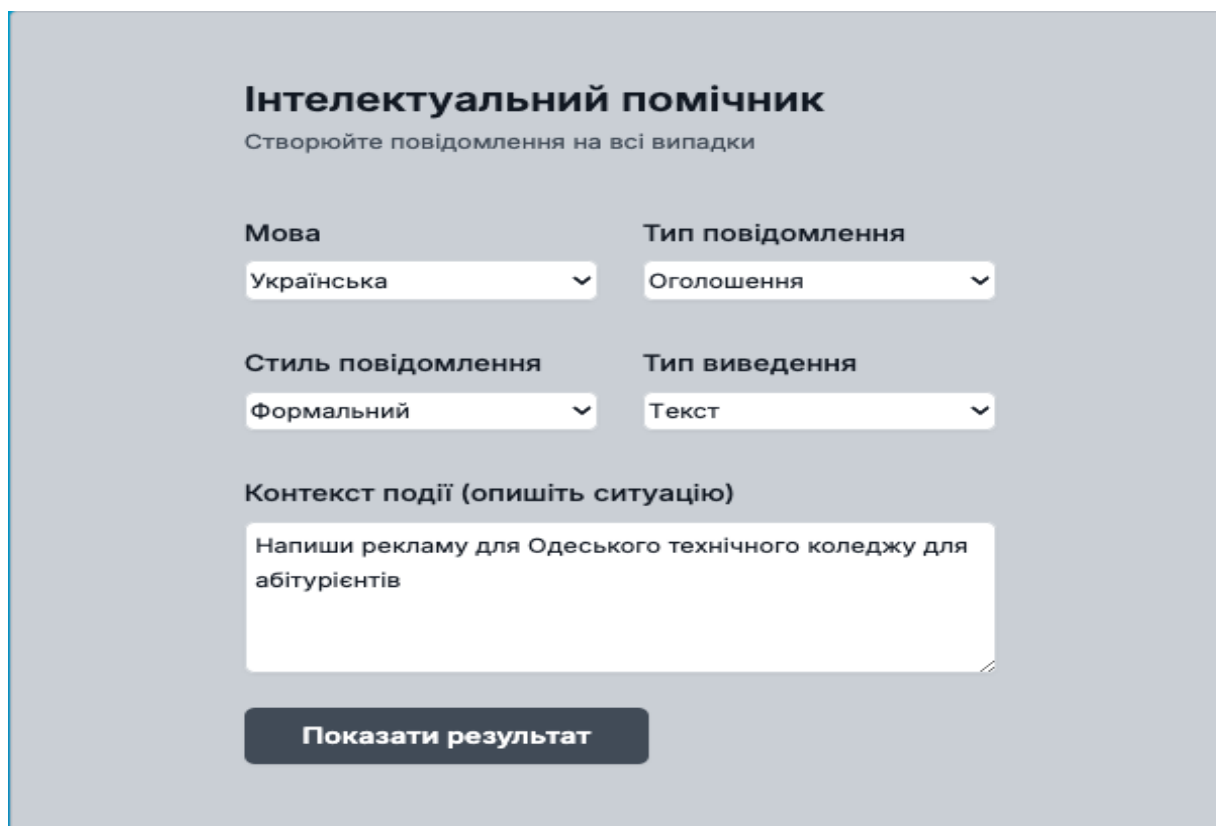


Рисунок 1.19. Скриншот роботи завдання

## 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-вимірвальна 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](http://www.otk.od.ua)



Редагувати запит

Рисунок 1.20. Скриншот виконання роботи завдання

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		53

# 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

## 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 1.21. Скриншот виконання роботи завдання

Зм.	Арк.	№ докум.	Підп.	Дата

РП 07. 01 001. 00 ДП ПЗ

Арк

54

**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 1.22. Скриншот виконання роботи завдання

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		55

## 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

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		56

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, 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, що враховує ступінь новизни

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
1.ТЗ	$T_{PT3}=25,15$	$T_{KK}=0,7*N_{T3}=0,7*2=1,4$	$T_{HK}=0,15*N_{T3}=0,15*2=0,30$
2.Розробка ПП	$T_{PTP}=31,44$	$T_{KK}=0,7*N_{TP}=0,7*18=12,6$	$T_{HK}=0,15*N_{TP}=0,15*18=2,7$
3.Розробка РП	$T_{RPP}=109,41$	$T_{KK}=0,7*N_{RP}=0,7*25=17,5$	$T_{HK}=0,15*N_{RP}=0,15*25=3,8$
4.Розробка ПЗ	$T_{PZ}=1,5*$ $*N_{PZ}=1,5*20=30$	$T_{KK}=0,7*N_{T3}=0,7*20=14$	$T_{HK}=0,15*N_{PZ}=0,15*20=3,0$
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			$\Sigma T_{HK}=9,8$

### 2.3 Розрахунок ціни програмного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машино – години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 2.6. Відповідно до статті 8 «Закону про Державний бюджет України на 2024» встановлено мінімальну заробітну плату у місячному розмірі з 1 квітня 2024 року - 8000 гривень; мінімальну погодинну тарифну ставку – 46,00 грн.

Таблиця 2.6 Розрахунок основної заробітної плати виконавців.

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

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		59

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	-	-	-	$V_{Mi}=240$
Транспортно– заготівельні витрати (10%)				$V_{тр\_з} = 0,1 \times V_{M1} = 24$
Усього				$V_M = V_{Mi} + V_{тр\_з} = 264,00$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведеною в таблиці 2.8.

Таблиця 2.8. Розрахунок статей витрат планової собівартості

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

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

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

Де  $p$  – плановий рівень рентабельності (10-15%).

Оптова ціна (кошторисна вартість) визначається по формулі:

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

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:

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

					<b>РП 07.01 001.00 ДП ПЗ</b>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		60

### 3. ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

Охорона праці на виробництві завжди була дуже важлива, отже саме завдяки рекомендаціям з охорони праці, персонал, який працює на підприємстві створює алгоритм виконання робочих завдань з чітким дотриманням рекомендацій. Основне завдання охорони праці – це створення та проведення заходів, спрямованих на захист життя, працездатності та здоров'я людини у процесі трудової діяльності.

При роботі з комп'ютером, як і в багатьох інших галузях, повинні враховуватись нормативи освітлення, температура, відносна вологість і сили вібрації. Але при роботі у приміщенні з комп'ютером найважливішим є дотримання правил пожежної безпеки, це вогнестійкість приміщення, також рівень звукового шуму, характеристики електромагнітних, ультрафіолетових та інфрачервоних полів.

Для аналізу охорони праці у дипломному проєкті досліджується безпека праці розробника веб-сторінок у офісному приміщенні.

#### 3.1 Аналіз та безпека умов праці працівника на робочому місці

Під час будь-якого виду роботи за комп'ютером, на працівника можуть мати дію небезпечні фактори виробничого середовища, а саме: фізичні та психофізіологічні небезпечні й шкідливі виробничі фактори.

Серед фізичних небезпечних факторів, найпоширеніші це підвищена температура повітря робочої зони, підвищений рівень шуму, знижена вологість повітря – це звичайні фактори, які виникають при роботі у приміщеннях з комп'ютерами, через їх роботу на робочому місці підіймається температура та знижується вологість повітря. Окрім цього, комп'ютер випромінює електростатичні та електромагнітні поля у діапазоні від 5 Гц до 2 кГц та від 2 до 400 кГц, тож робота за комп'ютером включає ще підвищений рівень

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		61

електромагнітний випромінювання та підвищений рівень статичної електрики. У офісних приміщеннях не завжди є достатня кількість природного освітлення у такому разі присутня велика кількість штучного освітлення, яке у свою чергу не завжди правильно налаштоване, з цього виникає, що світло може бути недостатньо яскравим або дуже яскравим.

Психофізіологічні виробничі небезпечні фактори поділяються на фізичні перевантаження та нервово-психічні перевантаження, при роботі з комп'ютером найчастіше друге. У нервово-психічних перевантаженнях програміст зазнає перенапругу аналізаторів та монотонність праці, інколи, ще й розмовну перенапругу, коли розробнику потрібно складати технічне завдання разом з клієнтом.

### **3.2 Розробка заходів з охорони праці**

#### **Виробниче освітлення**

Штучне освітлення в приміщеннях з робочими місцями, обладнаними ВДТ має здійснюватися системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, у разі переважної роботи з документами, допускається застосування системи комбінованого освітлення (крім системи загального освітлення, додатково встановлюються світильники місцевого освітлення).

#### **Мікроклімат**

При роботі у приміщеннях з великою кількістю комп'ютерів, приміщення з якими класифікуються як приміщення з підвищеною небезпекою електротравм, температура повітря влітку може становити більше 35 С, що погано впливає на здоров'я людини, тож у таких приміщеннях повітря повинне охолоджуватись та понижена вологість повітря повинна регулюватись спеціальним обладнанням.

Відповідно до норм ДСН 3.3.6.042-99 температура повітря в офісі повинна становити 22-25 С, вологість повітря 40-60%, швидкість руху повітря не більше

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		62

0,1 м/с. Якщо ці норми перевищені, робочий день працівника повинен бути скорочений на 10%.

### **3.3 Організація робочого місця користувача ПК**

Конструкція робочого місця користувача ПК й взаємне розташування всіх його елементів (сидіння, органи керування, засобу відображення інформації) відповідають антропометричним, фізіологічним і психологічним вимогам, а також характеру роботи. Конструкція робочих меблів повинна забезпечувати можливість індивідуального регулювання відповідно росту працюючих для підтримки зручної пози. Робочий стіл повинен бути пофарбований матовою фарбою. Дисплей розташований так, що його верхній край перебуває на рівні очей на відстані близько 70 см, що укладається в у припустимі рамки від 60 до 90 см. Частота мерехтіння екрана  $f_{\text{мер}}=100$  Гц, що відповідає умові  $f_{\text{мер}}>70$  Гц.

Робоче місце розташоване перпендикулярно віконним прорізам, це зроблено з тією метою, щоб виключити пряму й відбиту мерехтливність екрана від вікон і приладів штучного освітлення.

Згідно темі дипломного проекту робоче місце програміста укомплектовано пристроями з електромагнітним випромінюванням.

### **3.4 Пожежна безпека**

Забезпечення пожежної безпеки на об'єкті праці є важливою частиною роботи по створенню безпечних та здорових умов праці.

Прохід до аварійних виходів повинен бути вільний, шириною не менше 1 метру, у разі великої кількості горючих відходів потрібно використовувати відведені сміттєзбірники. Електроприлади повинні використовуватися тільки для їхнього прямого призначення, а у разі пошкодження приладів, слід вимкнути їх живлення та привести до пожежобезпечного стану.

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		63

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), а також різний переносний пожежний інструмент (кирки, сокири, багри, ломи і т. ін.).

Для гасіння пожеж промисловість випускає різні вогнегасники. Найбільшого поширення набули водопінні, водяні, газові (вуглекислотні) і порошкові. За ефективністю пожежогасіння гасіння, економічністю та іншими показниками більш перспективними вважаються порошкові вогнегасники.

Первинні засоби пожежогасіння розміщують на пожежних щитах, які встановлюють на виробничій території з розрахунку один щит на 5000 м<sup>2</sup>. Вони фарбуються у червоний колір.

Згідно Правил, на кожному поверсі будинку адміністративного призначення повинно знаходитися не менше двох вогнегасників з масою заряду вогнегасної речовини 5 кг і більше. Експлуатація вогнегасників без призначення відповідального за організацію цієї роботи не допускається.

Забороняється палити на підприємстві, крім спеціально виведених для цього місцях, забороняється зберігати легкозаймисті матеріали, такі як папір ближче ніж 1 метр від електрощитів, 0,15 м від приладів центрального водяного опалення та 0,6 м від сповіщувачів автоматичної пожежної сигналізації, також документація повинна зберігатися у спеціально відведених для цього шафах.

Для запобігання розповсюдження пожежі встановлюють протипожежні системи, які складаються з датчиків, звукових сповіщувачів, аварійних кнопок, приймально-контрольної панелі, яка виступає як аналізатор інформації, яку отримали датчики і відправляє ці данні на пульт пожежної охорони. Протипожежна сигналізація призначення для виявлення пожежі на початковому етапі.

Підприємство крім установки пожежної сигналізації на своєму об'єкті, має укласти договір на обслуговування даної системи з фірмою, що має на це ліцензію.

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		64

В обслуговування входить проведення встановлених нормами регламентних робіт, а так само усунення несправностей в роботі системи. Періодичність перевірки узгоджується з замовником, але повинна бути не рідше ніж один раз на місяць.

У разі, якщо пожежі не вдалось уникнути, необхідно:

1. терміново повідомити пожежну охорону по телефону 101, вказати при цьому адресу, кількість поверхів, місце виникнення пожежі, наявність людей, своє прізвище;
2. організувати евакуацію людей та матеріальних цінностей;
3. повідомити про виникнення пожежі адміністрацію та чергового (за його наявності);
4. вимкнути, у разі необхідності, струмоприймачі та вентиляцію;
5. розпочати гасіння пожежі наявними первинними засобами пожежогасіння;
6. організувати зустріч підрозділів пожежної охорони й надати їм консультаційну та іншу допомогу в процесі гасіння пожежі.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		65

## 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.

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

Тестування та валідація: 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

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		66

може бути корисним у багатьох областях, від бізнес-комунікацій до освітніх технологій. Програма відкриває нові можливості для автоматизації та поліпшення інтерактивного спілкування.

В дипломному проекті було проведено економічний розрахунок проекту та розглянуті питання охорони праці.

					<i>РП 07.01 001.00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		67

# ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ.

1. Уенді Боггс, Майкл Боггс. UML і Rational Rose – М.: Лори, 2018.
2. (SWEBOOK) TECHNICAL REPORT ISO/IEC TR 19759 IEEE First edition 2024-09-15
3. CMMI® for Development, Version 1.2, CMU/SEI-2016-TR-008 ESC-TR-2016-008
4. PHP Fast & Easy Web Development, 2nd Edition. ISBN:193184187x; Premier Press 2014
5. Буділов В. Основи програмування для Інтернета.: «БХВ-П», 2023.- 733с.
6. Глушков С. В. Програмування Web-сторінок / С. В. Глушков, І. А. Жакін, Т. С. Хачиров. – Харків: Фоліо, 2005. – 390 с.
7. Проценко О.Б. Web-програмування та web-дизайн. Технологія XML: Навчальний посібник. – Суми: Видавництво СумДУ, 2009. – 166 с.
6. Пасічник О.Г. Основи веб-дизайну. - К.:ВНУ. - 2009. – 354 с.
7. Холл М., Браун Л. Програмування для Web.- М.: Видавничий дім «Вільямс», 2002.- 1259 с.
9. Томсон Л., Веллінг Л. Розробка Web-приложень на PHP и MySQL.-К.: DiaSoft, 2003.- 650 с.
10. Скляр Д. PHP.С\Збірник рецептів.-М.: Символ-Плюс, 2003.- 643 с.
11. Інформаційні системи і технології: Навчальний посібник/ В.С. Пономаренко Р.К., Бутова І.В. Журавльова, Г.Н. Назарова та ін.; за ред. В.С. Пономаренка. - Київ: Академія, 202. - 542с.
12. Книга про HTML <http://htmlbook/> 3.Курс лекцій по HTML для початківців. <http://distancelearning.nuft.edu.ua/freecourses/html/index.html>

					<i>РП 07. 01 001. 00 ДП ПЗ</i>	Арк
Зм.	Арк.	№ докум.	Підп.	Дата		68

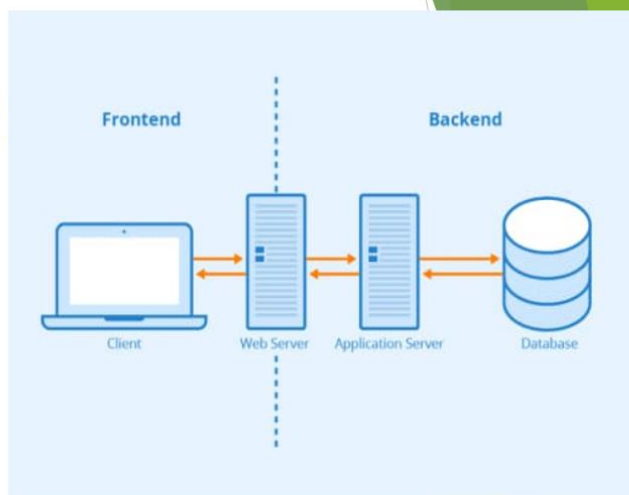
# ДИПЛОМНИЙ ПРОЕКТ НА ТЕМУ: Розробки програми для генерації повідомлень за допомогою штучного інтелекту.

Виконав студент гр.4РП-07: Бойко В.Д.  
Керівник ДП: К.Т.Н.Кунуп Т.В.

Підзаголовок слайда

## СЛАЙД 1

Взаємозв'язк  
між frontend і  
backend-  
розробкою



Активна

## СЛАЙД 2

## Скриншот коду файлу `server.js`

```
I > src > JS 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));
    }
  });
};
```

### СЛАЙД 3

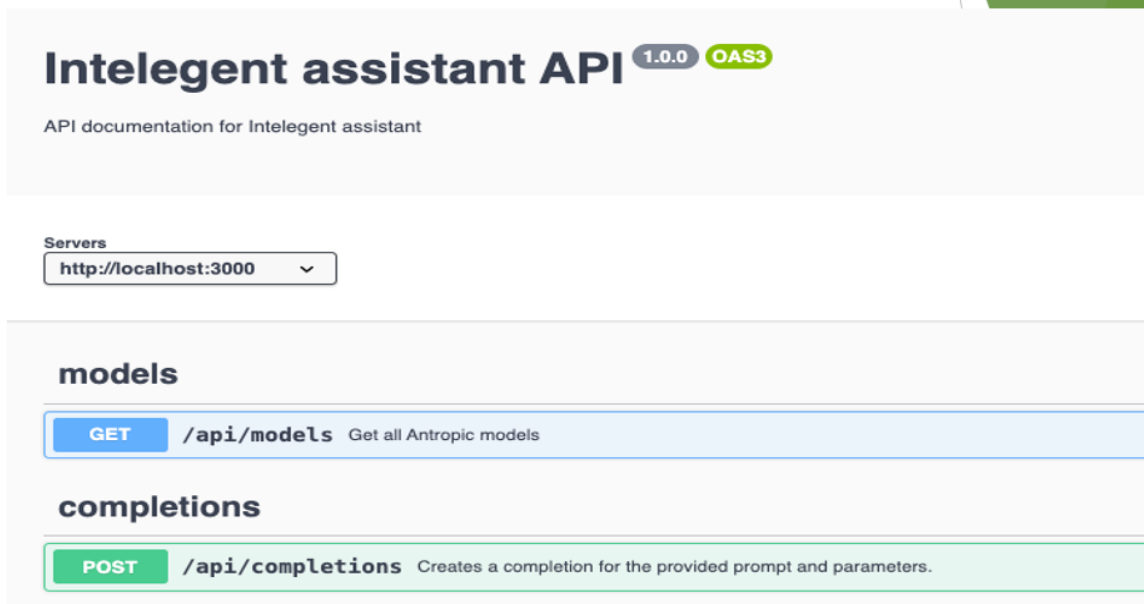
## Скриншот функції взаємодії з API Anthropic

```
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 });
  );
};
```

## СЛАЙД 4

### Скриншот Intelligent Assistant API в Swagger



## СЛАЙД 5

### Скриншот 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.

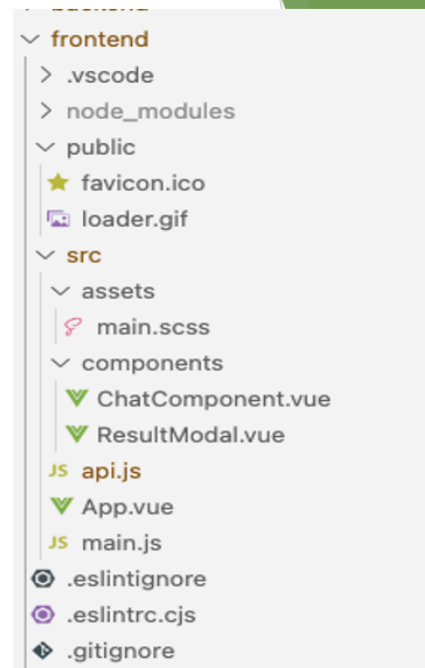
**assets:**

**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.



## СЛАЙД 6

Підзаголовок слайда

# Скриншот вікна програми

**Інтелектуальний помічник**  
Створіть повідомлення на всі випадки

Мова: Українська  
Тип повідомлення: Не визначений

Стиль повідомлення: Неформальний  
Тип виведення: Текст

Контекст події (опишіть ситуацію)

Показати результат

СЛАЙД 7

# Скриншот результату роботи

**Інтелектуальний помічник**  
Створіть повідомлення на всі випадки

Мова: Українська  
Тип повідомлення: Не визначений

Стиль повідомлення: Неформальний  
Тип виведення: Текст

Контекст події (опишіть ситуацію)

**Ваш результат:**

Привіт! Як справи? Чим займаєшся?

Редагувати запит

СЛАЙД 8

### 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**

## Скриншот результат роботи завдання

### СЛАЙД 9

**Скриншот 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**

### СЛАЙД 10

# Скриншот виконання завдання про рекламу коледжа.

## 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

## СЛАЙД 11

# Створення повідомлення на всі випадки

### Інтелектуальний помічник

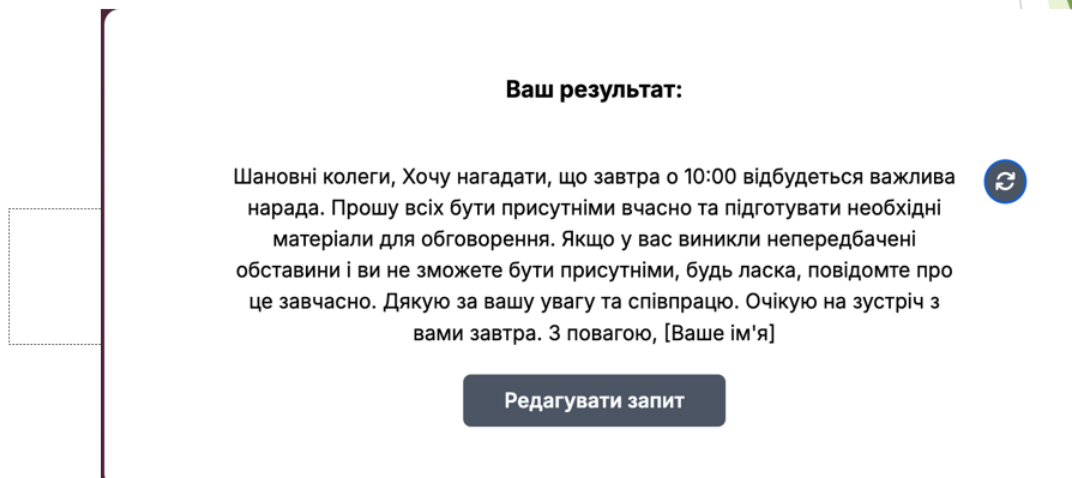
Створюйте повідомлення на всі випадки

Мова	Тип повідомлення
<input type="text" value="Українська"/>	<input type="text" value="Не визначений"/>
Стиль повідомлення	Тип виведення
<input type="text" value="Офіційно-діловий"/>	<input type="text" value="Текст"/>

Контекст події (опишіть ситуацію)

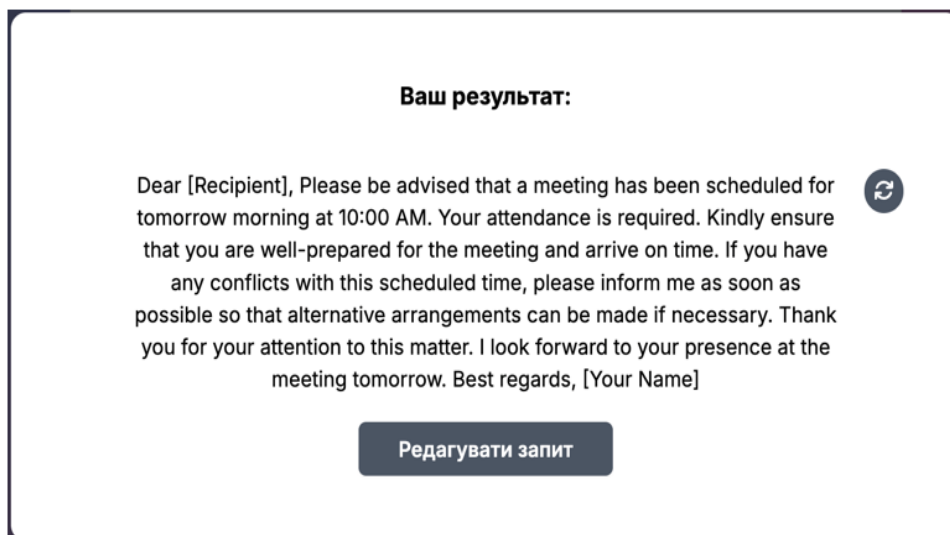
## СЛАЙД 12

## Скриншот виконання роботи завдання повідомлення на всі випадкі



### СЛАЙД 13

## Скриншот виконання іноземною мовою



### СЛАЙД 14

**ДЯКУЮ ЗА УВАГУ !!!**

**СЛАЙД 15**

**ВІДГУК**

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

Бойко Владиславу Дмитровичу

(прізвище, ім'я та по батькові)

Спеціальність: 121 “Інженерія програмного забезпечення”

Освітня програма: «Розробка програмного забезпечення»

Тема дипломного проекту: Розробка програми для генерації повідомлень  
за допомогою штучного інтелекта

**ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ**

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка містить 68 сторінок. У пояснювальній записці виконано опис штучного інтелекта та виконана розробка сервера для цифрових додатків на основі штучного інтелекта, а також його програмне забезпечення. Графічна частина складається з 15 слайдів мультимедійної презентації, які також містять креслення, передбачені технічним завданням. Якість виконання пояснювальної записки та графічної частини добра, розробку виконано в повному обсязі.

б) самостійність роботи над проектом: Протягом всього строку дипломного проектування та переддипломної практики здобувач освіти Бойко В.Д. поступово та послідовно виконував всі етапи розробки. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника

в) теоретична підготовка випускника (випускниці): Здобувач освіти Бойко В.Д. під час роботи над дипломним проектом вивчив достатню кількість літературних джерел та матеріалів за даною тематикою.

Вважаю, що теоретична підготовка дипломника добра і він готовий до захисту дипломного проекту

г) вміння розв'язувати виробничі та конструкторські питання \_\_\_\_\_  
Під час дипломного проектування здобувач освіти Бойко В.Д. мав змогу  
самостійно приймати окремі рішення з розробки програми для генерації  
повідомлень на основі штучного інтелекта та показав вміння організовано  
працювати над поставленим завданням, розробляти програмні коди за  
допомогою сучасних комп'ютерних програмних засобів та мов  
програмування.

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

Прізвище, ім'я, по батькові керівника дипломного проекту \_\_\_\_\_  
Кунуп Тетяна

Місце роботи і посада керівника дипломного проекту \_\_\_\_\_  
ВСП "Одеський технічний фаховий коледж ОНТУ", викладач  
спецдисциплін комісії комп'ютерних технологій та програмної інженерії,

Підпис \_\_\_\_\_ 

« 10 » червня 2024 р.

**РЕЦЕНЗІЯ**

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

**Бойка Владислава Дмитровича**

(прізвище, ім'я та по батькові)

Спеціальність **121 Інженерія програмного забезпечення**

Освітня програма **« Розробка програмного забезпечення»**

Керівник дипломного проекту (роботи) **к.т.н Кунуп Т.В.**

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) **Розробка програми створення повідомлень на базі штучного інтелекту**

Обсяг розрахунково-пояснювальної записки **68** сторінок

Обсяг графічної (презентаційної) частини **12** аркушів (слайдів)

**ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)**

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

Представлений на рецензію робота відповідає затверджений темі та виконаний відповідно технічному завданню. Дипломний проект є актуальним з погляду останніх рекомендацій розробки програмного забезпечення для створення повідомлень на базі штучного інтелекту .

б) характеристика виконання кожного розділу дипломного проекту (роботи) \_\_\_\_\_

Пояснювальна записка складається з технологічної частини, розробки структури програми та засобів програмування програми для створення повідомлень, опису і експлуатації засобів, економічної частини, розділу охорони праці та додатку. Перелічені розділи поетапно охоплюють розробку, виконані докладно та обґрунтовано. Розділ охорони праці містить загальну інформацію та вимоги до техніки безпеки оператора ЕОТ. Економічна частина проекту містить розрахунок \_\_\_\_\_ затрат \_\_\_\_\_ на \_\_\_\_\_ виконання програми \_\_\_\_\_

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

Графічна частина складається з 12 слайдів мультимедійної презентації, виконаної у програмному продукті MS PowerPoint, які містять структуру проекту, скріншоти роботи програми, передбачені технічним завданням. Пояснювальна записка виконана акуратно та у відповідності до норм. Якість виконання графічної частини проекту та пояснювальної записки висока, розробку виконано у повному обсязі

г) перелік позитивних якостей дипломного проекту (роботи) \_\_\_\_\_

У роботі обґрунтовано та розроблено програму створення повідомлень на базі штучного інтелекту

д) основні недоліки дипломного проекту (роботи) \_\_\_\_\_

1. У роботі відсутні необхідні схеми та алгоритми, що ілюструють етапи розробки.

2. З тексту роботи незрозуміло, які саме методи штучного інтелекту використовувались.

3. Робота оформлена не дуже охайно.

Оцінка розрахункової частини Добре

Оцінка графічної частини Добре

Загальна оцінка Добре

Прізвище, ім'я, по батькові рецензента Кіреєв Ігор Анатолійович

Місце роботи і посада рецензента Державний університет інтелектуальних технологій і зв'язку, доцент каф. інформаційної безпеки та передачі даних

Підпис: 

« 14 » червня 2024 р.



Ім'я користувача:  
Катерина Григоріївна Краснокутська

ID перевірки:  
1016329704

Дата перевірки:  
06.06.2024 22:15:22 EEST

Тип перевірки:  
Doc vs Internet + Library

Дата звіту:  
06.06.2024 22:18:33 EEST

ID користувача:  
100011688

Назва документа: 4РП-07 Бойко

Кількість сторінок: 51 Кількість слів: 7625 Кількість символів: 55969 Розмір файлу: 1.72 MB ID файлу: 1016129112

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

3.67%

## Схожість

Найбільша схожість: 0.81% з Інтернет-джерелом (<https://card-file.ontu.edu.ua/browse/title?scope=e5a3b909-ff7f-4df8-b9>).

3.67% Джерела з Інтернету 190

Сторінка 53

Не знайдено джерел з Бібліотеки

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%

## Вилучень

Немає вилучених джерел

## Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи 16

Підозріле форматування 17 сторінок

**ДОЗВІЛ  
НА РОЗМІЩЕННЯ  
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ  
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

***Бойка Владислава Дмитровича***

здобувач освіти гр. 4РП-07, та

***Кунуп Тетяна Василівна,***

керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до випускної кваліфікаційної роботи фахового молодшого бакалавра на тему:

***«Розробка програми для генерації повідомлень за допомогою штучного інтелекта». (автор роботи – Бойко В.Д., керівник роботи – Кунуп Т.В.)***

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2024 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

*Бойко*

/ Бойко В.Д./

Керівник

*Кунуп*

/ Кунуп Т.В./

« *10* » *06* 2024 р.