

Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки

«Затверджую»
в.о. завідуючого кафедри
комп'ютерних систем та робототехніки
_____ к. ф.-м. н., доцент Максим Хруслов
«___» червня 2025 р.

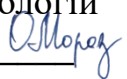
Пояснювальна записка

до кваліфікаційної роботи
бакалавра

на тему: **«МОДЕЛЬ ВЕБ-ДОДАТКУ ІЗ ВИКОРИСТАННЯМ ШТУЧНОГО
ІНТЕЛЕКТУ ДЛЯ ГЕНЕРАЦІЇ ПЕРСОНАЛІЗОВАНИХ
АСТРОЛОГІЧНИХ ПРОГНОЗІВ»**

Галузь знань: 12 – Інформаційні технології.
Спеціальність 123 – Комп'ютерна інженерія.
Освітня програма «Комп'ютерна інженерія».

Захищено на засіданні
Екзаменаційної комісії № 44
протокол № __ від __.06.2025 р.
Оцінка ____ / ____
Голова Екзаменаційної комісії
_____ **ЧУГАЙ А.М.**

Виконав:
студент групи КІ– 41
Фьодоров Данило Романович 
Керівник: доцент з во кафедри
комп'ютерних систем та робототехніки,
PhD з інформаційних технологій
МОРОЗ Ольга Юріївна 

Рецензент: доцент з во кафедри
інтелектуальних програмних систем і
технологій, к.т.н., доцент
ГАМЗАЕВ Рустам Олександрович _____

Харків – 2025

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел та трьох додатків. Загальний обсяг роботи складає 58 сторінки, з яких 46 сторінок – основна частина, що включає 12 рисунків та 6 лістингів.

Метою кваліфікаційної роботи є реалізація моделі інтелектуального веб-застосунку для автоматизованого формування натальної карти користувача на основі його персональних даних та інтерпретації отриманих результатів із використанням мультимодальної великої мовної моделі GPT-4o. від OpenAI.

Об'єктом дослідження є процес обробки персоналізованих даних у веб-застосунку із використанням штучного інтелекту та інтерпретації мультимодальної інформації.

Предмет дослідження – методи та архітектури інтеграції мультимодальних мовних моделей у веб-застосунку для обробки графічного та текстового контенту.

У роботі здійснено аналіз підходів до побудови інтелектуальних консультаційних систем, розроблено програмну реалізацію веб-застосунку, протестовано його функціональність, стабільність і здатність до масштабування.

Практична значущість полягає в можливості застосування розробленого інструменту як цифрового сервісу для персоналізованого прогнозування без участі експерта, з перспективою використання в системах самодіагностики та онлайн-консультацій. Результати роботи можуть бути впроваджені як у вигляді самостійного застосунку, так і як компонент цифрових платформ у сфері персоналізованих сервісів.

Ключові слова: ШТУЧНИЙ ІНТЕЛЕКТ, НАТАЛЬНА КАРТА, ВЕЛИКА МОВНА МОДЕЛЬ, GPT-4O, WEB-ЗАСТОСУНОК, АСТРОЛОГІЯ, NEXT.JS, TAILWIND CSS, АВТОМАТИЗАЦІЯ, ІНТЕРПРЕТАЦІЯ, МУЛЬТИМОДАЛЬНІСТЬ.

ABSTRACT

The explanatory note to the bachelor's qualification work consists of an introduction, three chapters, conclusions, a list of sources used, and three appendices. The total volume of the work is 58 pages, of which 46 pages are the main part, including 12 figures and 6 listings.

The purpose of the thesis is to implement a model of an intelligent web application for the automated formation of a user's natal chart based on their personal data and the interpretation of the results obtained using the multimodal large language model GPT-4o from OpenAI.

The object of the study is the process of processing personalized data in web applications using artificial intelligence and the interpretation of multimodal information.

The subject of the study is the methods and architectures for integrating multimodal language models into web applications for processing graphic and text content.

The work analyzes approaches to building intelligent consulting systems, develops a software implementation of a web application, and tests its functionality, stability, and scalability.

The practical significance lies in the possibility of using the developed tool as a digital service for personalized forecasting without the involvement of an expert, with the prospect of use in self-diagnostic and online consultation systems. The results of the work can be implemented both as a standalone application and as a component of digital platforms in the field of personalized services.

Keywords: ARTIFICIAL INTELLIGENCE, NATAL CHART, LARGE LANGUAGE MODEL, GPT-4O, WEB APPLICATION, ASTROLOGY, NEXT. JS, TAILWIND CSS, AUTOMATION, INTERPRETATION, MULTIMODALITY.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. Теоретичні основи побудови інтелектуальних веб-застосунків на базі мультимодальних мовних моделей.	11
1.1 Історія Штучного інтелекту	11
1.2. Інтелектуальні комп'ютерні системи: визначення, класифікація, архітектурні підходи	12
1.3 Сучасні веб-технології для реалізації інтерактивних користувацьких сервісів	14
1.4. Принципи побудови мультимодальних моделей штучного інтелекту (на прикладі GPT-4o).....	15
1.5. Особливості обробки графічної та текстової інформації у веб-застосунках	16
1.6 Огляд існуючих рішень в області астрологічного прогнозування: аналіз функціоналу, недоліки.	18
Висновки за розділом 1	18
РОЗДІЛ 2. Проектування моделі інтелектуального веб-застосунку для генерації персоналізованих астрологічних прогнозів.....	20
2.1 Аналіз задачі та постановка технічних вимог.....	20
2.2 Вибір архітектури веб-застосунку.....	21
2.3 Авторизація користувача та збереження даних	23
2.4 Побудова натальної карти користувача.....	25
2.5 Використання GPT-4o для інтерпретації натальної карти	30
2.6 Реалізація користувацького інтерфейсу.....	34
2.7 Проблеми, що виникли під час розробки, та шляхи їх вирішення	38
Висновок за розділом 2.....	39
РОЗДІЛ 3. Реалізація, тестування та аналіз результатів роботи моделі веб-застосунку	41
3.1 Реалізація користувацького інтерфейсу та серверної частини системи.....	41
3.2 Розгортання веб-застосунку на хмарному середовищі	41
3.3 Тестування функціональності: коректність побудови карти, стабільність генерації інтерпретацій	42

3.4 Аналіз якості та релевантності згенерованих прогнозів.....	42
3.5 Оцінка продуктивності та масштабованості системи	43
3.6 Аналіз переваг і обмежень запропонованої моделі	44
3.7 Перспективи розвитку	45
Висновок за розділом 3.....	45
ВИСНОВКИ.....	47
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	49
ДОДАТКИ	51

ПЕРЕЛІК ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

JavaScript (JS) – Динамічна, об'єктно-орієнтована прототипна мова програмування.

React – Відкрита JavaScript бібліотека для створення інтерфейсів користувача.

Next JS – Фреймворк React, який розширює його можливості додатковими функціями, включаючи рендеринг на стороні сервера та генерацію статичних веб-сайтів.

CSS – Це спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду.

API – Інтерфейс програмування застосунків (application programming interface), спосіб взаємодії комп'ютерних програм між собою.

ІІІ – Штучний інтелект

AI – Artificial Intelligence (Штучний інтелект).

LLM (Large Language Model) – велика мовна модель, штучний інтелект який вміє розуміти та створювати тексти, схожі на ті, що пише людина.

Prompt – запит або інструкція, яку користувач надсилає до мовної моделі, щоб отримати відповідь; формулювання запиту значною мірою впливає на якість результату.

Base64 – формат кодування бінарних даних у текстовий вигляд, часто використовується для передавання файлів через текстові протоколи.

SSR (Server Side Rendering) – Рендеринг сторінки на сервері перед її доставкою клієнту; покращує SEO та швидкість першого завантаження.

SSG (Static Site Generation) – Попередня генерація HTML-сторінок на етапі білду, що забезпечує високу швидкодію та кешування.

ISR (Incremental Static Regeneration) – Гібридний метод генерації, який дозволяє оновлювати сторінки після білду без повного повторного рендерингу.

SEO (Search Engine Optimization) – Оптимізація вебсторінок і контенту для підвищення видимості сайту в результатах пошукових систем.

ІКС – Інтелектуальна комп'ютерна система, здатна самостійно аналізувати дані, навчатися та приймати рішення без прямого програмування.

ML (Machine Learning) – Машинне навчання, підхід у штучному інтелекті, за допомогою якого система навчається на основі даних без явного програмування.

NLP (Natural Language Processing) – Обробка природної мови, галузь ІІІ, яка займається аналізом, розумінням і генерацією людської мови.

ВСТУП

У сучасному цифровому середовищі спостерігається стрімке зростання інтересу до персоналізованих сервісів, здатних адаптувати свій функціонал до індивідуальних запитів користувача. Особливої актуальності набувають інтелектуальні веб-застосунки, що поєднують методи штучного інтелекту з високопродуктивними архітектурами обробки даних у реальному часі. У цьому контексті перспективним напрямом є створення систем, здатних не лише збирати та аналізувати персоналізовану інформацію, а й надавати автоматизовані інтерпретації результатів, що раніше потребували участі фахівця.

Астрологічні сервіси, попри свою прикладну специфіку, активно інтегруються в цифрові екосистеми як елементи особистісної аналітики, психологічної підтримки та планування. Водночас більшість доступних онлайн-рішень потребують участі експерта або працюють за жорстко запрограмованими шаблонами, що знижує гнучкість та індивідуалізацію.

Враховуючи сучасні досягнення в галузі комп'ютерної інженерії, зокрема застосування мультимодальних мовних моделей (наприклад, GPT-4o), відкриваються нові можливості для побудови веб-застосунків, які здатні інтегрувати графічні та текстові дані для формування змістовних, персоналізованих висновків. Це дозволяє реалізовувати автоматизовані консультаційні сервіси нового покоління. У поєднанні з технологіями штучного інтелекту (ШІ), астрологія отримує потенціал до масової автоматизації, масштабування, підвищення точності та швидкості аналітики. Сучасні великі мовні моделі (LLM), демонструють здатність не лише опрацьовувати природну мову, але й аналізувати візуальні дані. Це відкриває нові можливості в автоматизованій інтерпретації натальних карт.

Актуальність роботи зумовлена необхідністю створення доступних та ефективних інструментів для автоматизованої інтерпретації складних персоналізованих даних, а також зростанням попиту на цифрові консультаційні сервіси, здатні функціонувати без залучення вузькопрофільних спеціалістів.

Метою кваліфікаційної роботи є реалізація моделі інтелектуального веб-застосунку для автоматизованого формування натальної карти користувача на основі його персональних даних та інтерпретації отриманих результатів із використанням мультимодальної великої мовної моделі GPT-4o від OpenAI.

Об'єктом дослідження є процес обробки персоналізованих даних у веб-застосунку із використанням штучного інтелекту та інтерпретації мультимодальної інформації.

Предмет дослідження – методи та архітектури інтеграції мультимодальних мовних моделей у веб-застосунки для обробки графічного та текстового контенту.

У роботі здійснено аналіз підходів до побудови інтелектуальних консультаційних систем, розроблено програмну реалізацію веб-застосунку, протестовано його функціональність, стабільність і здатність до масштабування.

Новизна роботи полягає в застосуванні великої мовної моделі (LLM) для автоматизованої інтерпретації натальної карти, що формується на основі введених користувачем даних. Раніше такі карти або оброблялися вручну, або піддавалися обмеженій інтерпретації. У роботі запропоновано архітектурну модель інтелектуального веб-застосунку, що поєднує інструменти веб-розробки з мультимодальними великими мовними моделями (наприкладі GPT-4o) для інтерпретації складних структурованих (астрологічні дані) та неструктурованих (текстова інформація) даних. Зокрема, реалізовано інтеграцію графічного модулю генерації натальної карти та її текстову інтерпретацію без участі експерта, що є інноваційним підходом у створенні інтелектуальних консультаційних систем.

Практична значущість. Результати роботи можуть бути використані для розробки цифрових консультаційних платформ, які надають персоналізовану аналітику на основі введених користувачем даних без необхідності залучення фахівця. Запропоноване рішення є масштабованим, придатним для інтеграції у більші екосистеми (наприклад, сервіси персонального розвитку, самодіагностики, психологічної або астрологічної підтримки). Крім того, реалізована система демонструє можливості використання сучасних мовних

моделей у веб-інтерфейсах, що є актуальним для фахівців з комп'ютерної інженерії.

Для досягнення мети визначено комплекс задач, що будуть реалізовані у ході дослідження.

Задачі дослідження:

1. Проаналізувати сучасні підходи до побудови інтелектуальних комп'ютерних систем, зокрема веб-застосунків, що використовують штучний інтелект для персоналізованого аналізу даних.
2. Дослідити архітектуру та функціональні можливості мультимодальних мовних моделей (зокрема GPT-4o від OpenAI) у контексті їх інтеграції у веб-застосунки.
3. Оцінити технічні вимоги до системи автоматизованого формування натальної карти на основі персональних даних користувача (дата, час, місце народження).
4. Розробити архітектурну модель веб-застосунку, що забезпечує побудову натальної карти та генерацію її інтерпретації за допомогою LLM.
5. Реалізувати прототип інтелектуального веб-застосунку із використанням сучасних засобів веб-розробки та API мультимодальної моделі.
6. Здійснити тестування функціональності, стабільності та якості результатів, зокрема оцінити якість персоналізованих прогнозів та швидкодію системи.
7. Визначити можливості подальшого розвитку моделі.

Для досягнення мети та виконання поставлених задач у роботі застосовуються такі **методи дослідження**:

1. **Аналіз літературних та наукових джерел** – вивчення сучасних теоретичних підходів до розробки інтелектуальних веб-систем, мультимодальних мовних моделей та астрологічних інформаційних технологій.
2. **Метод системного проектування** – моделювання архітектури веб-застосунку, розробка структурних схем, що відображають взаємодію компонентів системи.
3. **Програмна інженерія** – вибір технологій, розробка прототипу застосунку з використанням сучасних фреймворків і API (зокрема OpenAI GPT-4o).

4. Інтеграційне тестування – перевірка коректності роботи компонентів веб-застосунку, оцінка взаємодії між підсистемами, забезпечення стабільності роботи системи.

5. Експертна оцінка та аналіз якості результатів – перевірка адекватності та релевантності згенерованих астрологічних прогнозів за допомогою методів порівняння з існуючими сервісами або консультаціями фахівців.

6. Експериментальний метод – проведення тестування функціональності, аналіз продуктивності, збір відгуків користувачів (за наявності) для удосконалення моделі.

7. Метод адаптації та оптимізації – налаштування параметрів системи для підвищення ефективності генерації прогнозів і покращення користувацького досвіду.

Запропонована модель сприятиме автоматизації створення персоналізованих астрологічних прогнозів, що відкриває перспективи для її використання у різних інформаційних сервісах. Отже, розробка інтелектуального веб-застосунку, який поєднує передові веб-технології та можливості штучного інтелекту, є актуальним завданням з огляду на сучасні тенденції цифрової трансформації.

РОЗДІЛ 1.

ТЕОРЕТИЧНІ ОСНОВИ ПОБУДОВИ ІНТЕЛЕКТУАЛЬНИХ ВЕБ-ЗАСТОСУНКІВ НА БАЗІ МУЛЬТИМОДАЛЬНИХ МОВНИХ МОДЕЛЕЙ.

1.1 Історія Штучного інтелекту

Ідея створення штучного інтелекту має давнє походження й бере початок ще з античних уявлень про штучні форми життя. Наукове осмислення штучного інтелекту (ШІ) почалося у XX столітті: від створення автоматизованих пристроїв, таких як шахова машина Леонардо Торреса-і-Кеведо, до розробки першого електронного комп'ютера (Атанасов-Беррі) та теоретичних основ нейронних мереж (Мак-Каллок і Піттс, 1943).

У 1950-х роках закладено фундамент сучасного розуміння ШІ: запропоновано Тест Тюрінга, вперше використано термін «штучний інтелект» (1956), з'являються перші навчальні системи. У 1970–1990-х роках розвиваються експертні системи, зворотне поширення помилки і застосування ШІ в медицині та іграх.

У XXI столітті спостерігається стрімкий розвиток глибокого навчання, комп'ютерного зору, обробки природної мови. Серед ключових подій – поява IBM Watson (2011), ImageNet, голосових асистентів, GPT-3 (2020), DALL·E та мультимодальних моделей OpenAI і Google (2021–2024). Сучасні мовні моделі типу GPT-4 та Gemini 1.5 здатні виконувати складні аналітичні, генеративні та інтерактивні завдання, що відкриває нові перспективи для застосування ШІ в освітньому, культурному та сервісному середовищах, зокрема в автоматизованому формуванні астрологічних прогнозів. У 2020 році компанія OpenAI представила мовну модель GPT-3, яка мала 175 мільярдів параметрів і демонструвала здатність генерувати текст, схожий на людський, вести діалоги, писати код, перекладати мови та створювати креативні тексти на основі запитів природною мовою. Вона стала прикладом потужності великих мовних моделей і продемонструвала, що ШІ може виконувати різноманітні мовні завдання майже без додаткового навчання.

У 2021-2023 роках OpenAI представила DALL·E, а згодом DALL·E 2 і DALL·E 3 – генеративні моделі, здатні створювати деталізовані зображення за текстовими описами. Вони використовують глибоке навчання та трансформери для перетворення ідей на реалістичні або художні зображення без потреби в навичках графічного дизайну.

У лютому 2024 року компанія Google випустила в обмеженому бета-доступі модель Gemini 1.5, яка здатна працювати з контекстами до 1 мільйона токенів. Це дозволяє моделі ефективніше обробляти великі обсяги інформації, краще розуміти й утримувати контекст у довгих діалогах і складних завданнях.

1.2. Інтелектуальні комп'ютерні системи: визначення, класифікація, архітектурні підходи

Інтелектуальні комп'ютерні системи (ІКС) є результатом багаторічного розвитку галузі штучного інтелекту та інформаційних технологій. Вони становлять собою комплекс апаратних і програмних засобів, які здатні імітувати інтелектуальну діяльність людини: аналізувати інформацію, виявляти закономірності, робити прогнози, приймати рішення в умовах невизначеності, навчатися на основі отриманого досвіду.

ІКС активно застосовуються у сфері фінансів, медицини, освіти, промисловості, телекомунікацій, персональних сервісів тощо. Особливої уваги заслуговує їх роль у створенні інтелектуальних веб-застосунків, де вони забезпечують адаптацію до користувача, персоналізовану взаємодію та автоматичну обробку складної інформації.

Інтелектуальна комп'ютерна система – це система, яка здатна виконувати функції, що традиційно вважаються інтелектуальними для людини: сприйняття, навчання, міркування, прийняття рішень, комунікація природною мовою. У таких системах використовуються методи штучного інтелекту, зокрема машинне навчання, нейронні мережі, логічні виведення, експертні правила тощо.

Класифікація інтелектуальних комп'ютерних систем.

ІКС можна класифікувати за різними критеріями:

1. За способом реалізації інтелекту:

- **Експертні системи** – базуються на формалізованих правилах і базі знань. (наприклад, медичні діагностичні системи).
- **Нейронні мережі** – імітують роботу мозку, здатні до навчання; застосовуються для розпізнавання образів, прогнозування.
- **Системи на основі машинного навчання (ML)** – навчаються на основі вхідних даних.
- **Гібридні системи** – поєднують кілька підходів для досягнення більшої точності та гнучкості.

2. За функціональним призначенням:

- **Рекомендаційні системи** (наприклад, Netflix, Amazon).
- **Прогностичні моделі** (для оцінки ризиків, прогнозування попиту).
- **Консультаційні агенти** (чат-боти, віртуальні помічники).
- **Аналітичні платформи** (аналітика великих даних).

3. За типом оброблюваних даних:

- **Текстові** – NLP-моделі (обробка природної мови).
- **Графічні** – комп'ютерне бачення.
- **Мультимодальні** – поєднання тексту, зображень, звуку (наприклад, GPT-4o).

Архітектурні підходи до побудови ІКС

Архітектура інтелектуальної системи визначає її логічну структуру, компоненти, способи взаємодії між ними. Існують різні підходи до проектування ІКС, залежно від цілей системи:

- **Модульна архітектура.** Система поділена на окремі модулі: блок введення даних, блок аналізу, блок прийняття рішень, блок виводу результатів. Це спрощує масштабування й підтримку системи.
- **Клієнт-серверна архітектура.** Частина функцій (інтерфейс, збір даних) виконується на клієнті, а обробка інтелектуальних задач – на сервері або в хмарі (через API доступ до мовної моделі, як у GPT-4o).
- **Мікросервісна архітектура.** Кожен функціональний компонент реалізується окремим мікросервісом (наприклад, сервіс авторизації,

генерації карти, аналізу зображення), що взаємодіє API. Це забезпечує гнучкість і надійність при масштабуванні.

- **Архітектура на основі агентів.** Включає інтелектуальні агенти, кожен з яких виконує конкретне завдання, обмінюється знаннями та координує дії з іншими агентами. Підходить для систем зі складною поведінкою.
- **Архітектура з використанням зовнішніх LLM через API.** В сучасних веб-застосунках замість локального навчання часто використовується доступ до хмарних мовних моделей (GPT-3.5, GPT-4o), які приймають текстові або мультимодальні запити і повертають результат у вигляді тексту або структурованих даних (JSON).

1.3 Сучасні веб-технології для реалізації інтерактивних користувацьких сервісів

Створення сучасних веб-застосунків передбачає використання інструментів, які забезпечують швидкодію, адаптивність і зручну інтеграцію з зовнішніми сервісами. Основу таких систем складають як фронтенд, так і бекенд-технології.

Фронтенд (інтерфейс користувача):

- **HTML, CSS, JavaScript** – базові технології для структури та стилю сторінок.
- **React, Vue.js, Angular** – бібліотеки й фреймворки для створення динамічних SPA.
- **Next.js, Nuxt.js** – фреймворки для серверного рендерингу, що покращують SEO та продуктивність.
- **Tailwind CSS, Bootstrap** – CSS-фреймворки для швидкої та адаптивної верстки.

Бекенд (серверна логіка):

- **Node.js, FastAPI, Django, Laravel** – популярні фреймворки для побудови серверної частини, API та роботи з базами даних.
- **JWT, OAuth2** – технології авторизації та безпечної автентифікації.
- **REST API, GraphQL** – протоколи обміну даними між клієнтом і сервером.

Бази даних:

- **PostgreSQL, MySQL** – реляційні бази для структурованих даних.
- **MongoDB, Redis** – NoSQL-рішення для зберігання документів або кешу.

Інші важливі технології:

- **Docker, CI/CD (GitHub Actions, GitLab CI)** – автоматизація розгортання та тестування.
- **OpenAI API, сторонні сервіси** – інтеграція ШІ та зовнішніх обчислювальних ресурсів.

Поєднання цих технологій дозволяє створювати ефективні та масштабовані сервіси, що забезпечують високу якість взаємодії з користувачем.

1.4. Принципи побудови мультимодальних моделей штучного інтелекту (на прикладі GPT-4o)

Мультимодальні моделі штучного інтелекту (ШІ) – це моделі, здатні працювати з різними типами даних одночасно: текстом, зображеннями, аудіо, відео та іншими. Вони відкривають нові можливості для створення інтелектуальних систем, що розуміють контекст комплексно, подібно до людини. Одним із сучасних прикладів таких моделей є GPT-4o, розроблена компанією OpenAI.

Архітектурні основи мультимодальних моделей

Мультимодальні моделі базуються на архітектурі трансформера, яка дозволяє обробляти дані у вигляді послідовностей токенів. Щоб модель могла працювати з різними типами вхідної інформації (наприклад, текст та зображення), ці дані попередньо **перетворюються в уніфіковане представлення** – вектори (ембедінги), які потім обробляються єдиною нейронною мережею.

У випадку GPT-4o, обробка зображень відбувається через спеціалізовані вхідні компоненти, які "перекладають" візуальний контент у форму, придатну для аналізу мовною моделлю. Завдяки цьому, GPT-4o може давати осмислені відповіді не лише на текстові запити, але й на зображення, графіки, скріншоти тощо.

Особливості GPT-4o

- **Універсальність:** модель підтримує мультимодальність у режимі реального часу, включаючи обробку тексту та зображень.

- **Інтегрований контекст:** GPT-4o здатна розуміти одночасний контекст із кількох джерел – наприклад, питання до зображення.
- **Єдина модель без окремих модулів:** на відміну від попередників, GPT-4o не використовує окремі компоненти для тексту й зображень – усі типи даних обробляються спільним трансформером, що покращує швидкість і узгодженість відповідей.
- **Висока швидкодія:** модель здатна швидко обробляти великі обсяги даних, що робить її придатною для інтеграції у веб-застосунки.

Застосування мультимодальних моделей

Мультимодальні моделі типу GPT-4o широко застосовуються в таких завданнях:

- Автоматична інтерпретація зображень (у тому числі складних, як-от графіки чи карти);
- Генерація описів, пояснень, підписів до зображень;
- Відповіді на запитання щодо вмісту зображень;

1.5. Особливості обробки графічної та текстової інформації у веб-застосунках

У веб-застосунках, що орієнтовані на зручну та інформативну взаємодію з користувачем, особливу роль відіграє інтеграція графічного та текстового контенту. Сучасні сервіси все частіше поєднують ці два типи інформації, що вимагає застосування відповідних технічних рішень, форматів і підходів до обробки.

Обробка графічної інформації

Графічні дані у веб-середовищі можуть включати фотографії, ілюстрації, схеми, графіки, карти, а також векторні зображення. Їхня обробка у веб-застосунках передбачає:

- **Завантаження або динамічну генерацію зображень** – наприклад, створення інфографіки, генерація діаграм на основі даних користувача, інтеграція мап чи аватарів.

- **Оптимізацію форматів і розмірів** – зображення мають відповідати вимогам швидкодії масштабованості та адаптивності.
- **Відображення з адаптацією до пристрою** – підтримка responsive-дизайну з урахуванням типу екрана та щільності пікселів.
- **Інтерактивність** – підтримка подій, анімацій, масштабування або зміни вмісту без перезавантаження сторінки.

У деяких випадках графіка також використовується як засіб для візуалізації результатів аналізу або побудови персоналізованого контенту (наприклад, графіки активності користувача, карти народження тощо).

Обробка текстової інформації

Текстова інформація є основним способом передачі змісту та взаємодії з користувачем у більшості веб-додатків. Обробка тексту включає:

- **Формування динамічного контенту** – наприклад, адаптивні повідомлення, особисті звернення або результати обчислень.
- **Валідацію введених даних** – перевірка текстових полів у формах (ім'я, адреса, повідомлення), з урахуванням мовних та регіональних особливостей.
- **Стилізацію та форматування** – використання CSS для створення списків, заголовків, таблиць тощо.
- **Пошук, фільтрація та сортування** – особливо актуально у застосунках зі списками або великою кількістю даних.

Ключовим фактором при роботі з текстом є підтримка багатомовності, шрифтів з розширеним набором символів, а також зрозумілого UX для користувачів різного рівня підготовки.

Комбіноване використання графіки і тексту

Поєднання текстової і графічної інформації дає змогу:

- Створювати інтерфейси, що краще сприймаються користувачем;
- Забезпечувати більш точну візуалізацію результатів (наприклад, текстовий опис даних + графік);
- Реалізовувати інтуїтивну навігацію та зворотний зв'язок.

Приклади комбінованих підходів можна побачити в особистих кабінетах, аналітичних панелях, онлайн-магазинах, освітніх платформах, інформаційних сервісах. Таким чином, ефективна обробка графіки та тексту є основою сучасних веб-застосунків, які прагнуть до гнучкості, адаптивності та персоналізації.

1.6 Огляд існуючих рішень в області астрологічного прогнозування: аналіз функціоналу, недоліки.

Сучасні онлайн-сервіси, такі як Astro.com, AstroSeek та Astromix, дозволяють будувати натальні карти та отримувати астрологічні інтерпретації. Кожен із них має свої переваги, проте водночас обмежений у індивідуалізації та зручності використання. Вони здебільшого використовують класичний підхід: після введення даних народження користувач отримує графічну карту та текстовий опис, згенерований за заздалегідь підготовленими шаблонами.

Попри функціональність, такі сервіси мають низку обмежень:

- відсутність глибокої персоналізації прогнозів;
- використання шаблонних текстів без адаптації до контексту;
- застарілий або перевантажений інтерфейс, незручний для новачків.

Крім того, більшість платформ не підтримують мультимодальний аналіз, який поєднував би текстову та візуальну інформацію для глибшого розуміння карти.

Таким чином, актуальною є розробка нового інтелектуального веб-застосунку, що поєднає:

- сучасний UX/UI;
- можливості мультимодальних моделей;
- гнучку інтерпретацію на основі ШІ;
- інтерактивність і персоналізований підхід.

Даний проєкт покликаний усунути зазначені недоліки та покращити користувацький досвід в астрологічних консультаційних сервісах.

Висновки за розділом 1

У першому розділі було розглянуто ключові поняття та технології, необхідні для створення інтелектуального веб-застосунку з використанням мультимодальних мовних моделей. Проаналізовано етапи розвитку штучного інтелекту, класифікацію інтелектуальних систем та сучасні підходи до їх архітектури. Особливу увагу приділено можливостям моделі GPT-4o, здатної обробляти як текстову, так і графічну інформацію. Також проаналізовано недоліки існуючих астрологічних сервісів – шаблонність, складний інтерфейс, відсутність персоналізації – що підтверджує доцільність створення автоматизованого рішення нового покоління.

РОЗДІЛ 2.

ПРОЄКТУВАННЯ МОДЕЛІ ІНТЕЛЕКТУАЛЬНОГО ВЕБ-ЗАСТОСУНКУ ДЛЯ ГЕНЕРАЦІЇ ПЕРСОНАЛІЗОВАНИХ АСТРОЛОГІЧНИХ ПРОГНОЗІВ

2.1 Аналіз задачі та постановка технічних вимог

У межах даної роботи було поставлено завдання створити веб-застосунок, що забезпечує автоматичне формування натальної карти користувача та її інтерпретацію за допомогою мультимодальної моделі GPT-4o. Зважаючи на це, до системи висуваються технічні вимоги, що охоплюють усі етапи – від збору вхідних даних до формування текстового прогнозу.

Вхідні дані

Для побудови натальної карти необхідно зібрати від користувача такі персональні дані:

- дата народження;
- точний час народження (у форматі 24-годинного часу);
- геолокація (місце народження: місто, країна).

Ці параметри є критично важливими для точного розрахунку положення планет та формування достовірної карти.

Вихідні дані

Система повинна забезпечити наступне:

- побудова натальної карти у вигляді графічного зображення;
- передача зображення до GPT-4o для обробки;
- формування персоналізованого текстового прогнозу.

Обмеження системи

- підтримка формату зображення лише PNG з кодуванням у base64;
- ліміти OpenAI API на об'єм вхідних даних;
- обробка помилок API та некоректних запитів.

Функціональні вимоги

- веб-інтерфейс із адаптивною версткою;
- форма реєстрації/введення особистих даних;

- побудова карти за допомогою стороннього API;
- інтеграція з GPT-4o через OpenAI API;
- збереження результату;

Нефункціональні вимоги

- швидкий час відповіді (до 20 секунд);
- зручність і доступність інтерфейсу (UI/UX);
- масштабованість рішення для подальшого розширення функціоналу;

2.2 Вибір архітектури веб-застосунку

Розробка сучасного веб-застосунку вимагає чітко продуманої архітектури, що забезпечить масштабованість, швидкодію, безпеку й зручність у використанні. У межах цієї роботи було обрано стек технологій, до якого входять: Next.js, React, Tailwind CSS, а також зовнішні API для побудови натальної карти та GPT-4o від OpenAI для генерації текстових прогнозів.

Обґрунтування вибору технологій

- Next.js – сучасний React-фреймворк, що дозволяє реалізовувати серверний рендеринг (SSR), статичну генерацію сторінок (SSG), а також динамічну генерацію (ISR) [15]. Ці механізми сприяють покращенню SEO-оптимізації, зменшенню часу першого завантаження та створюють умови для зручного масштабування проєкту. Крім того, Next.js надає вбудовану підтримку API-ендпоінтів, що дозволяє створювати повноцінний бекенд без потреби у зовнішньому сервері [2].

- React – потужна бібліотека для побудови динамічних інтерфейсів користувача з використанням компонентного підходу, яка дозволяє ефективно управляти станом застосунку та покращити UX [1].

- Tailwind CSS – утилітарний CSS-фреймворк, що дозволяє створювати адаптивний інтерфейс без необхідності написання кастомного CSS. Він забезпечує швидку та якісну верстку, а також активно використовується в багатьох сучасних UI-бібліотеках [3].

- OpenAI GPT-4o – модель штучного інтелекту з підтримкою мультимодальності, що дозволяє інтерпретувати зображення натальної карти й генерувати відповідні текстові описи. Вона інтегрується через OpenAI API [6].

Загальна архітектура застосунку

Архітектура веб-застосунку реалізована за принципом поділу на клієнтську та серверну частину:

- *Клієнтська частина* – забезпечує введення даних користувача, відображення стану процесу, інтерфейс для перегляду результату, а також взаємодію з формами й елементами UI. Усі компоненти реалізовані на базі React з використанням Tailwind CSS для стилізації.
- *Серверна частина (API Next.js)* – відповідає за обробку запитів, передачу даних до зовнішніх сервісів (для побудови карти та інтеграції з GPT), обробку відповідей та логіку взаємодії з клієнтською частиною.
- *Зовнішні сервіси* – окремі API: для побудови натальної карти за вхідними даними (геолокація, дата, час народження), для конвертації зображення у потрібний формат, та OpenAI API, у який передається зображення у форматі base64 разом із промптом для аналізу моделю.

Переваги обраної архітектури

- *Масштабованість*: Next.js дозволяє легко додавати нові сторінки, API-маршрути або модулі без зміни всієї структури.
- *Гнучкість*: Компонентний підхід React дає змогу повторно використовувати UI-блоки та швидко вносити зміни.
- *Продуктивність*: завдяки серверному рендерингу і попередній генерації сторінок підвищується швидкість завантаження застосунку.
- *Динамічність*: взаємодія із застосунком відбувається без перезавантаження сторінок, що забезпечує високу швидкість роботи та зручність для користувача.
- *Інтеграція зі III*: використання GPT-4o дозволяє реалізувати глибоку персоналізацію прогнозів і працювати із зображенням напряму.

Усі компоненти об'єднані у логічну структуру, що дозволяє гнучко масштабувати проєкт, додавати нові функції, а також оптимізувати інтеграцію зі сторонніми сервісами.

Архітектура веб застосунку зображена на Рис. 2.1.

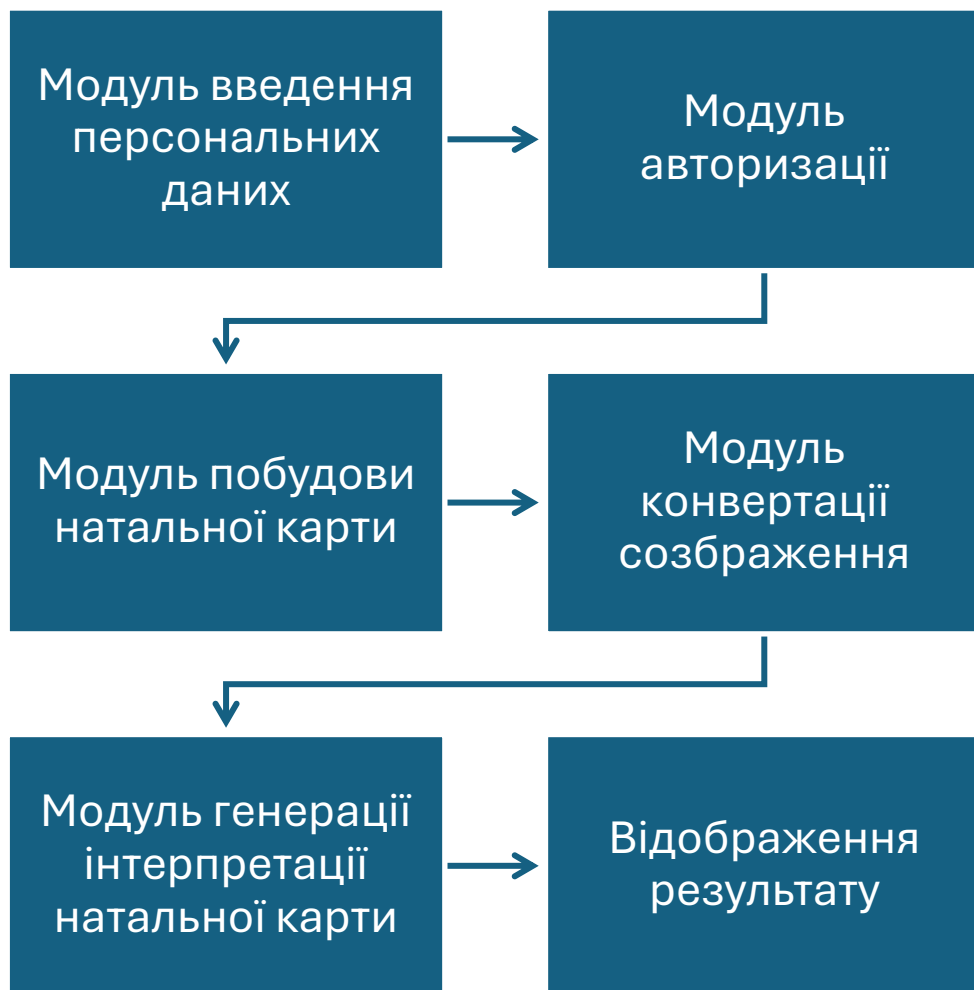


Рисунок 2.1 – Архітектура веб-застосунку.

2.3 Авторизація користувача та збереження даних

Для забезпечення персоналізованого досвіду користувача, а також безпечного зберігання результатів аналізу натальної карти, у системі реалізовано повноцінну авторизацію на основі JWT-токенів.

Інтерфейс взаємодії реалізовано за допомогою FastAPI, де авторизація, редагування профілю, зміна паролю та збереження натальних даних реалізовані у вигляді REST-ендпоінтів. Основною перевагою такого підходу є гнучкість, безпека та масштабованість сервісу. Приклад REST-ендпоінтів наведено у лістингу 2.1.

```

user_router = APIRouter()
natal_router = APIRouter()
@user_router.get('/session/', summary='Get session', response_model=AccessTokenSchema)
def session(user: AppUser = Depends(JWTService.get_active_user)):
    data = UserService.session(user)
    return data
@user_router.patch('/profile/edit/', summary="Edit profile", response_model=UserSchema)
def profile_edit(data: ProfileEditSchema, user: AppUser =
Depends(JWTService.get_active_user)):

```

```

...
birthdate - 01.01.2000
birthtime - 12:00:00 or null
...
if data.birthtime:
    if len(data.birthtime.split(':')) != 3:
        raise HTTPException(status_code=400, detail='invalid_birthtime')
    user = UserService.profile_edit(user, data)
    return UserSchema.from_orm(user)
@user_router.patch('/profile/password/edit/', summary="Edit password", status_code=200)
def password_edit(data: PasswordEditSchema, user: AppUser =
Depends(JWTService.get_active_user)):
    UserService.password_edit(user, data)

@user_router.delete('/profile/delete/', summary="Delete profile", status_code=200)
def profile_delete(user: AppUser = Depends(JWTService.get_active_user)):
    UserService.profile_delete(user)

@natal_router.get('/', summary="Get natal", response_model=str)
def natal_get(user: AppUser = Depends(JWTService.get_active_user)):
    return user.natal_data

@natal_router.patch('/save/', summary='Save natal', status_code=200)
async def natal_save(request: Request, user: AppUser =
Depends(JWTService.get_active_user)):
    user.natal_data = (await request.body()).decode("utf-8")
    await sync_to_async(user.save)()

```

Лістинг 2.1. – Реалізація API-маршрутів для керування профілем користувача та збереження натальних даних.

Основні функції авторизації та персонального кабінету:

- **Отримання активної сесії користувача:** Ендпоінт GET /session/ дозволяє отримати токен доступу та основні дані поточного авторизованого користувача. Захист реалізовано за допомогою Depends(JWTService.get_active_user), що перевіряє автентичність запиту.
- **Редагування персональних даних:** Через запит PATCH /profile/edit/ користувач може змінити своє ім'я, дату та час народження. Передбачена валідація, зокрема формат часу (HH:MM:SS) перевіряється вручну. Це дозволяє оновити натальну інформацію без створення нового облікового запису.
- **Зміна пароля:** Окремий ендпоінт PATCH /profile/password/edit/ дозволяє користувачеві змінити пароль. Це підвищує рівень безпеки й дозволяє самостійно управляти обліковими даними.

- **Видалення профілю:** Якщо користувач хоче повністю видалити свій обліковий запис, він може скористатися ендпоінтом DELETE /profile/delete/. Це забезпечує дотримання принципів конфіденційності та права на забуття.
- **Збереження результатів інтерпретації натальної карти:** Після того, як штучний інтелект (ШІ) згенерував текстову інтерпретацію натальної карти, вона може бути збережена в акаунті користувача через PATCH /natal/save/. Ці дані можна отримати згодом через GET /natal/, що дозволяє користувачеві повертатися до своїх результатів у будь-який момент.

2.4 Побудова натальної карти користувача

Формування натальної карти є одним із ключових етапів у реалізації веб-застосунку. Натальна карта – це астрологічна схема, яка відображає положення небесних тіл на момент народження людини. Для її побудови необхідно мати точні астрономічні розрахунки, які виконуються за допомогою зовнішніх астрологічних сервісів.

Збір персональних даних

Користувач проходить реєстрацію де вводить три основні параметри: дату, час та місце народження. Реєстрація відбувається покроково, після неї ці дані передаються до серверної частини застосунку через захищений API-ендпоінт, де ініціюється запит до стороннього сервісу для побудови карти.

Рисунок 2.2 – Перший етап реєстрації.

Перший етап реєстрації містить в собі форму с заповненням дати народження, всі поля є обов'язковими, якщо користувач не заповнить хоч 1 поле, то буде показана помилка, та яке саме поле потрібно заповнити. Відповідний інтерфейс подано на рисунку 2.2.

Другий етап реєстрації містить в собі форму с заповненням місця народження, також на всіх кроках крім першого можна повернутися назад, щоб виправити якісь дані. Відповідний інтерфейс подано на рисунку 2.3.

Рисунок 2.3 – Другий етап реєстрації.

На третьому етапі, користувач має ввести свій точний час народження, але якщо він не знає точного часу, тоді може вибрати пункт “Я не знаю свій час народження”. Відповідний інтерфейс подано на рисунку 2.4.

Рисунок 2.4 – Третій етап реєстрації.

На четвертому етапі реєстрації користувачу потрібно вказати свою стать та ввести своє ім'я, далі користувач переходить до останнього етапу реєстрації. Відповідний інтерфейс подано на рисунку 2.5.

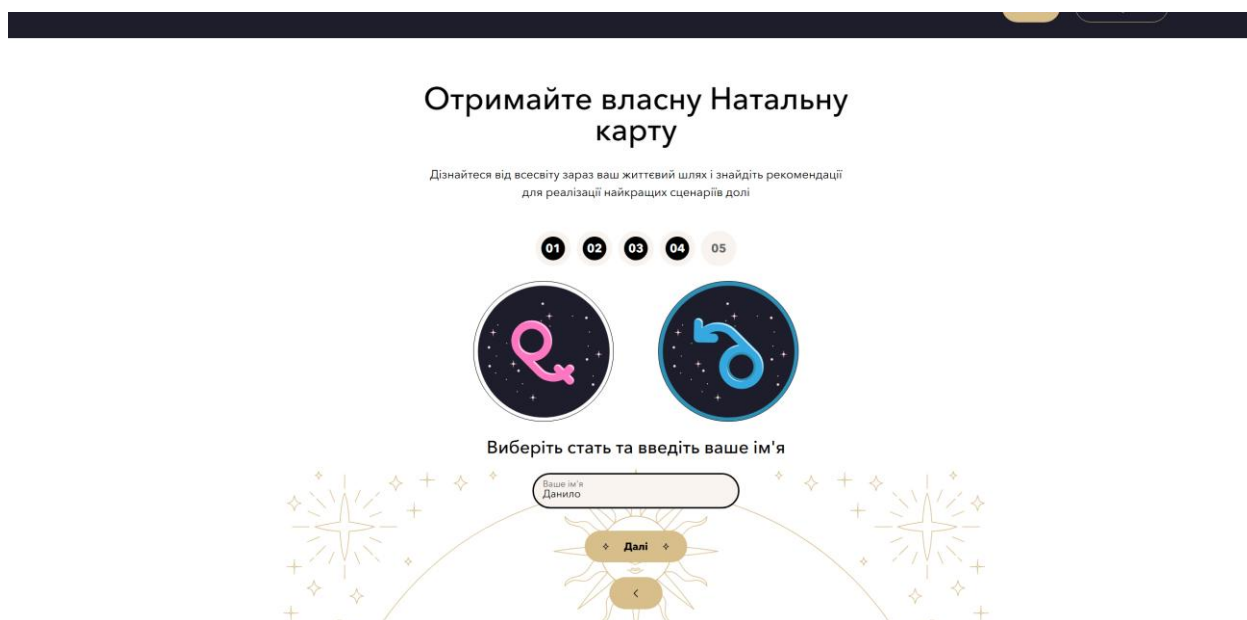


Рисунок 2.5 – Четвертий етап реєстрації.

Останній етап реєстрації включає в себе поле для введення пошти та погодження з угодою користувача та політикою конфіденційності.

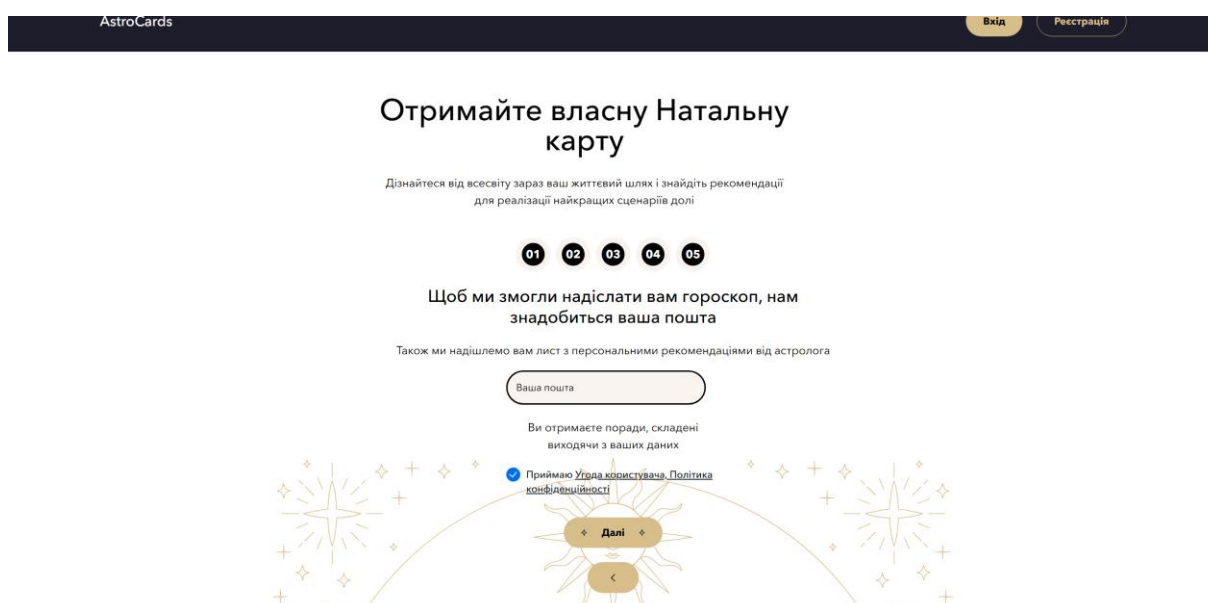


Рисунок 2.6 – Останній етап реєстрації.

Після цього етапу відбувається запит для реєстрації нового користувача, після вдалого запиту, відбувається формування натальної карти стороннім API та передача цього зображення для інтерпретації до GPT-4o mini. Відповідний інтерфейс подано на рисунку 2.6.

Генерація карти через сторонній API

Для створення натальної карти було обрано сторонній сервіс, що надає можливість розрахунку астрологічних даних. Запит включає координати (широта, довгота), дату і точний час, а відповіддю є зображення карти. Реалізацію API-ендпоінта для надсилання запиту до цього сервісу подано у лістингу 2.2.

Зображення генерується у форматі SVG, але зараз OpenAi Api не підтримує цей формат, тому необхідно конвертувати отримане зображення у формат який підтримується (наприклад PNG), потім це зображення переводиться у формат base64 для подальшої передачі в GPT-4o. Це дозволяє моделі аналізувати структуру карти.

```
import { NextResponse } from "next/server";

const apiKey = process.env.NATAL_API_KEY;

export async function POST(request: Request) {
  if (!apiKey) {
    return NextResponse.json({ error: "Missing API key" }, { status: 500 });
  }

  try {
    const { year, month, day, hour, min, lat, lon, tzzone } =
      await request.json();

    const body = {
      year: Number(year),
      month: Number(month),
      date: Number(day),
      hours: Number(hour),
      minutes: Number(min),
      seconds: 0,
      latitude: Number(lat),
      longitude: Number(lon),
      timezone: Number(tzzone),
    };

    const res = await fetch(
      "https://json.freeastrologyapi.com/western/natal-wheel-chart",
      {
        method: "POST",
```

```

        headers: {
            "Content-Type": "application/json",
            "x-api-key": apiKey,
        },
        body: JSON.stringify(body),
    }
});

if (!res.ok) {
    const errorText = await res.text();
    console.error("API error:", errorText);
    return NextResponse.json({ error: errorText }, { status: res.status });
}

const data = await res.json();
return NextResponse.json(data);
} catch (error) {
    console.error("Server error:", error);
    return NextResponse.json(
        { error: "Internal Server Error" },
        { status: 500 }
    );
}
}
}

```

Лістинг 2.2 – API route для отримання натальної карти.

Після отримання натальної карти у форматі SVG необхідно здійснити її конвертацію у формат PNG, оскільки OpenAI API не підтримує SVG. Для цього реалізовано окремий запит до сервісу конвертації зображень. Реалізацію API-маршруту для цього етапу подано у лістингу 2.3.

```

import { NextResponse } from "next/server";
import FormData from "form-data";
import fetch from "node-fetch";

const secret = process.env.CONVERT_API_KEY;

export async function POST(request: Request) {
    const form = new FormData();
    const { svgUrl } = await request.json();

    form.append("file", svgUrl);
    form.append("StoreFile", "true");
    form.append("FileName", "astro");
    form.append("ImageResolution", "200");
    form.append("ScaleImage", "false");
    form.append("ScaleProperties", "true");
    form.append("ScaleLarger", "false");
    form.append("ImageHeight", "20");
    form.append("ImageWidth", "20");
}

```

```

form.append("ImageInterpolation", "false");
form.append("TransparentColor", "<string>");

try {
  const res = await fetch(
    `https://v2.convertapi.com/convert/svg/to/png?secret=${secret}`,
    {
      method: "POST",
      headers: form.getHeaders(),
      body: form,
    }
  );

  if (!res.ok) {
    throw new Error("Network response was not ok");
  }

  const data = await res.json();
  return NextResponse.json(data);
} catch (error) {
  console.error("Error fetching data:", error);
  return NextResponse.error();
}
}

```

Лістинг 2.3 – API route для конвертації зображення.

Таким чином, реалізовано повний цикл побудови натальної карти, що слугує вхідною точкою для подальшої обробки штучним інтелектом.

2.5 Використання GPT-4o для інтерпретації натальної карти

Одразу після створення натальної карти у форматі зображення необхідно здійснити її інтерпретацію за допомогою моделі штучного інтелекту GPT-4o, яка здатна аналізувати графічну інформацію та формувати логічні текстові висновки на її основі. У цьому підпункті розглянуто процес побудови запиту, формування промпту та обробки відповіді моделі.

Формування запиту до GPT-4o

Запит до OpenAI API будується як масив `input`, що містить два об'єкти: перший – текстовий промпт із вказівкою задачі, другий – зображення карти. Зображення передається у вигляді `data:image/png;base64,...`, а також вказується рівень деталізації через параметр `detail: "high"` для кращої якості аналізу.

У тексті промпту модель інструктується повертати лише JSON-об'єкт – без markdown, підписів чи додаткових пояснень. Це дозволяє точно обробити відповідь програмно без додаткового очищення. Приклад промпту наведено у лістингу 2.4.

```
const prompt = `
    Ти астролог. Проаналізуй надану карту та зроби висновки згідно з текстом
    користувача.
    Відповідь повинна бути тільки у форматі JSON, без markdown, без
    пояснень.
    Без пояснень, без підписів, без коду – тільки чистий JSON-об'єкт.
    Текст користувача:
        Опиши загальну інформацію взаємодії планет і будинків натальної
        карти людини.
    Поверни відповідь суворо у вигляді JSON-масиву:
    [
      {
        "title": "Короткий заголовок (наприклад, 'Вплив Сонця')",
        "text": "Змістовний опис на 3-5 речень"
      },
      {
        "title": "...",
        "text": "..."
      }
    ]
`;
```

Лістинг 2.4 – Приклад промпту, який буде відправлятися до LLM .

Обробка відповіді

Отримана відповідь від GPT-4o проходить попередню обробку: видаляються потенційні обгортки, і текст парситься у валідний JSON. Якщо відповідь не є масивом або не відповідає очікуваній структурі – генерується відповідна помилка.

Цей підхід дозволив забезпечити повністю автоматизовану інтеграцію ШІ в інтерфейс застосунку, де користувач бачить лише кінцевий структурований результат – без потреби взаємодії з технічними деталями. Реалізація API-запиту до моделі OpenAI наведена у лістингу 2.5.

```
import { NextRequest, NextResponse } from "next/server";
import OpenAI from "openai";
```

```

const openai = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY,
});

export async function POST(req: NextRequest) {
  try {
    const { image, text } = await req.json();

    if (!image || !text) {
      return NextResponse.json(
        { error: "Need to upload a file and enter text" },
        { status: 400 }
      );
    }

    const prompt = `
      Ти астролог. Проаналізуй надану карту та зроби висновки згідно з текстом
      користувача.
      Відповідь повинна бути **тільки у форматі JSON**, без markdown, без пояснень.

      Без пояснень, без підписів, без коду – тільки **чистий JSON-об'єкт**.

      Текст користувача:

      ${text}
    `;

    const response = await openai.responses.create({
      model: "gpt-4o-mini",
      input: [
        {
          role: "user",
          content: [
            {
              type: "input_text",
              text: prompt,
            },
            {
              type: "input_image",
              image_url: `${image}`,
              detail: "high",
            },
          ],
        },
      ],
    });

    let res = response.output_text ?? "";

    res = res
      .trim()
      .replace(/^```json\s*/, "")
      .replace(/\s*```$/, "");

    let data;
    try {
      data = JSON.parse(res);
    } catch (err) {
      console.error("OpenAI JSON parsing error:", err);
      return NextResponse.json(

```



```

        { error: "Failed to parse JSON from OpenAI" },
        { status: 500 }
    );
}

if (!Array.isArray(data)) {
    return NextResponse.json(
        { error: "OpenAI response is not an array" },
        { status: 500 }
    );
}

return NextResponse.json({ data });
} catch (error: unknown) {
    if (error instanceof Error) {
        console.error("Error:", error.message);
        return NextResponse.json({ error: error.message }, { status: 500 });
    } else {
        return NextResponse.json({ error: "Unknown error" }, { status: 500 });
    }
}
}
}

```

Лістинг 2.5 – API route, запит до OpenAi.

Запит який виконується на стороні клієнта до API route, таких запитів може бути кілька, тобто для кожного пункту опису натальної карти ми робимо свій запит, приклад у лістингу 2.6.

```

async function fetchFuture(pngUrl: string, age: number) {
    const response = await fetch("/api/openai", {
        method: "POST",
        body: JSON.stringify({
            image: pngUrl,
            text: `Опиши майбутнє людини, яка ввела свої дані, проаналізувавши натальну
            карту на найближчі 5 років. Її вік – ${age}`.
        })
    });
}

```

Поверни відповідь суворо у вигляді JSON-масиву:

```

[
  {
    "title": "Пік",
    "text": "Змістовний опис на 3-5 речень"
  },
  {
    "title": "...",
    "text": "..."
  }
]
),
});

```

```

    if (!response.ok) {
      throw new Error("Failed to fetch");
    }

    return response.json();
  }
}

```

Лістинг 2.6 – запит до API route з OpenAi.

Труднощі, які виникли

Під час розробки було виявлено кілька суттєвих проблем:

- деякі відповіді GPT-4о приходили в markdown-форматі, навіть попри явне обмеження – це потребувало додаткового очищення;
- модель генерує помилки при пошкодженому або обрізаному base64-зображенні;
- JSON не завжди повністю коректний, потрібна перевірка перед парсингом.

Завдяки додатковій валідації та обробці помилок вдалося досягти стабільної взаємодії з OpenAI API й забезпечити надійне створення персоналізованих прогнозів на основі вхідного зображення.

2.6 Реалізація користувацького інтерфейсу

Інтерфейс користувача є критично важливою складовою будь-якого сучасного веб-застосунку. У межах даного проєкту інтерфейс реалізований за допомогою Next.js і стилізований із використанням Tailwind CSS, що дозволило створити адаптивне, зручне й швидкодієне середовище для взаємодії користувача із системою.

Основні елементи інтерфейсу:

- Форма введення персональних даних (реєстрація): користувач вводить ім'я, дату, час та місце народження. Валідація відбувається у реальному часі.
- Редагування даних: користувач у будь-який момент може змінити введені дані (наприклад, виправити час або місце народження). Після збереження змін система автоматично заново формує натальну карту та її інтерпретацію.
- Візуалізація та інтерпретація карти: побудована натальна карта відображається прямо на сторінці. Поруч виводиться детальна астрологічна інтерпретація.

- Відображення станів процесу: користувач бачить повідомлення про помилки, та інші вказівки.
- Головна сторінка: на головній сторінці користувач знайомиться з проєктом, а також з причинами, чому варто знати свою натальну карту.

Адаптивність інтерфейсу

Інтерфейс повністю адаптований до різних типів пристроїв: мобільних телефонів, планшетів і комп'ютерів. Tailwind CSS забезпечує коректне масштабування елементів, зміну розміщення блоків у залежності від розміру екрана, та зручну взаємодію навіть на невеликих дисплеях без необхідності писати стилі для кожного компонента CSS.

UX-дизайн застосунку:

Під час розробки застосунку було приділено особливу увагу користувацькому досвіду (UX). Основною метою було забезпечення інтуїтивно зрозумілої навігації, логічної структури елементів та мінімізації кількості дій, які повинен виконати користувач для досягнення бажаного результату.

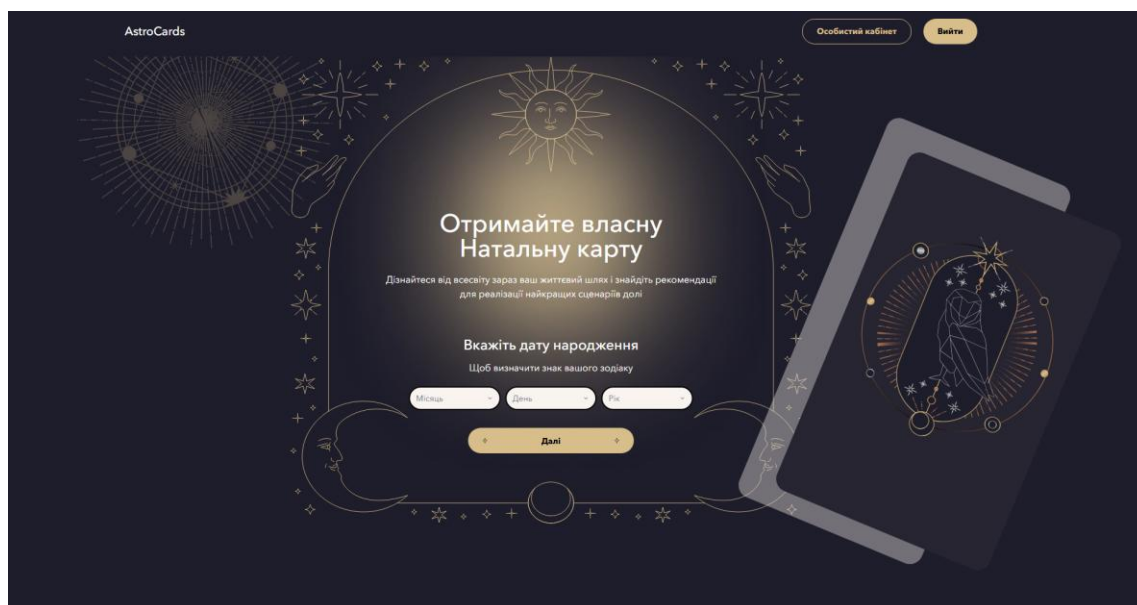


Рисунок 2.7 – форма для початку реєстрації на головній сторінці

Для того щоб розпочати процес реєстрації та отримання натальної карти, на головній сторінці є форма для початку реєстрації, приклад на рис. 2.7.

Візуальні елементи згруповано за змістом і логікою використання. Інформаційні підказки, динамічні повідомлення про стан дій які відбуваються – усе це підвищує ефективність взаємодії користувача з сайтом.

Особливістю UX-дизайну є також наявність головної сторінки, яка допомагає новому користувачеві швидко зорієнтуватися в призначенні сайту та покроково пройти процес створення натальної карти. Приклад на рис 2.8.

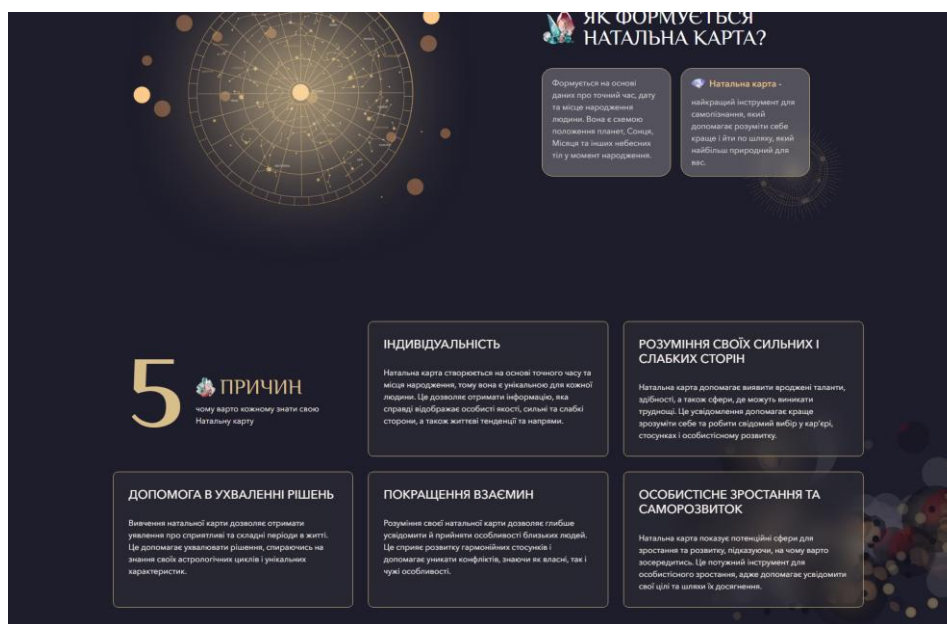


Рисунок 2.8 – інформація для користувача на головній сторінці.

Особистий кабінет містить у собі такі пункти, як управління акаунтом та натальна карта, приклад на рис. 2.9.

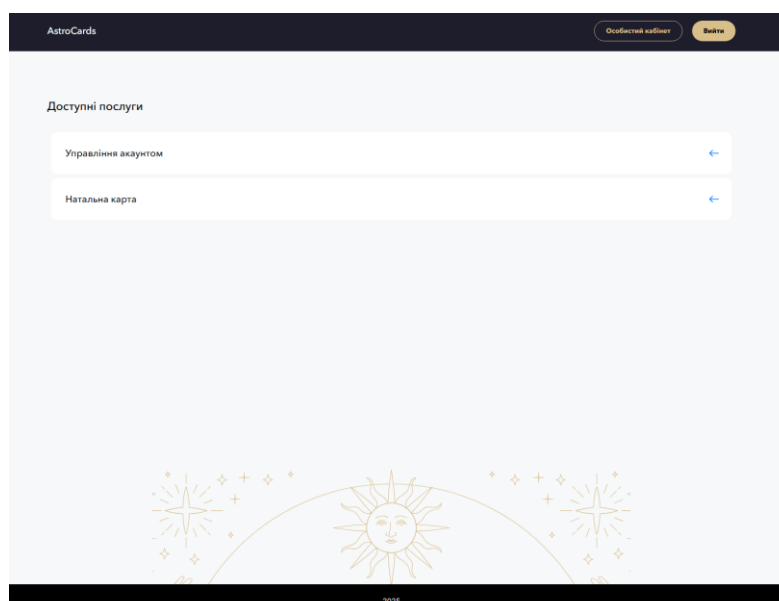


Рисунок 2.9 – Особистий кабінет користувача

При відкритті пункту «Натальна карта», користувач бачить зображення власної карти та її інтерпретацію, приклад наведено у рис. 2.10.

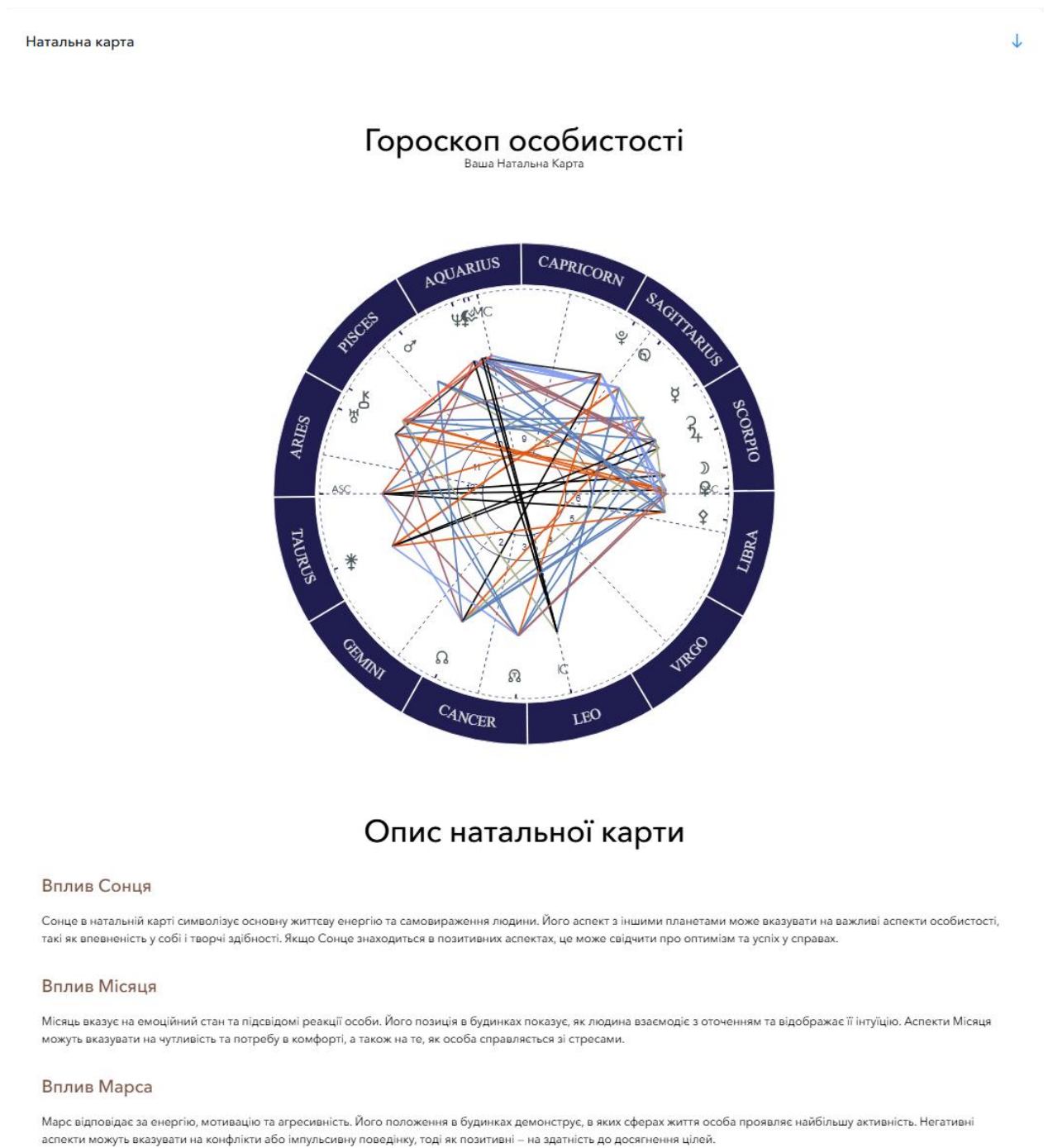


Рисунок 2.10 – Натальна карта та її опис.

У розділі «Управління акаунтом» користувач може змінити особисту інформацію, таку як ім'я, дату та час народження. Приклад наведено у рис. 2.11.

доступні послуги

Управління акаунтом

Дата народження:

Січень 02 2019

Місце народження:

Україна Харків

Час народження:

12:00:00

☐ Я не знаю час народження

Ваше ім'я:

Ваше ім'я
Данило

Ваша стаття:

Змінити пароль:

Старий пароль Новий пароль

При зміні даних від облікового запису, натальна карта буде завантажена по новою

Зберегти

[Видалити обліковий запис](#)

Рисунок 2.11 – Управління аккаунтом.

2.7 Проблеми, що виникли під час розробки, та шляхи їх вирішення

У процесі розробки веб-додатку було виявлено низку проблем, які вимагали технічного, організаційного та логічного вирішення. Нижче наведено основні труднощі, а також шляхи їх вирішення.

Передача зображення до OpenAI API

Однією з найбільших проблем стала правильна передача зображення до OpenAI API. Виявилося, що API не підтримує формат SVG, який часто генерується астрологічними сервісами. Зображення має бути у форматі PNG або JPG з точним дотриманням структури `data:image/png;base64,...` або `data:image/jpeg;base64,...`. У перших реалізаціях саме через використання SVG запити не проходили валідацію, що спричиняло помилки.

Рішення: для усунення цієї проблеми було впроваджено проміжний API, який автоматично конвертує SVG у допустимий формат (наприклад, PNG). Після цього зображення кодується у base64 і передається у запиті до OpenAI. Такий підхід забезпечив сумісність та стабільну передачу зображень.

Некоректна відповідь моделі

Навіть при правильно складеному запиті модель могла повертати відповіді у вигляді форматованого markdown або тексту, який не є JSON, попри пряму інструкцію. Це унеможливлювало подальший парсинг відповіді без додаткової обробки.

Рішення: було реалізовано фільтрацію відповіді, очищення від зайвих символів та додано додаткову обробку відповіді.

Некоректні відповіді GPT-4o

Незважаючи на інструкцію, модель іноді повертала markdown або текст із коментарями.

Рішення: очищення відповіді перед парсингом, уточнення промптів.

Нестабільна генерація

GPT-4o міг відповідати дуже загально або скорочено.

Рішення: деталізовані інструкції в промптах, встановлення шаблонів структури відповіді.

Висновок за розділом 2

У другому розділі було здійснено повне проектування моделі інтелектуального веб-застосунку для генерації персоналізованих астрологічних прогнозів. Визначено технічні вимоги до системи, обґрунтовано вибір архітектури та стеку технологій, що забезпечують масштабованість, продуктивність і інтеграцію з зовнішніми сервісами.

Розроблено механізм авторизації та збереження персональних даних, реалізовано побудову натальної карти користувача на основі введених даних із подальшою автоматизованою інтерпретацією за допомогою GPT-4o.

Також було розглянуто технічні проблеми, що виникли під час реалізації, і запропоновано ефективні шляхи їх вирішення. Таким чином, розділ закладає фундамент для практичної реалізації, тестування та оцінювання запропонованої моделі у наступному етапі роботи.

РОЗДІЛ 3.

РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ РОБОТИ МОДЕЛІ ВЕБ-ЗАСТОСУНКУ

3.1 Реалізація користувацького інтерфейсу та серверної частини системи

Реалізація інтелектуального веб-застосунку була здійснена з урахуванням вимог до адаптивності, швидкодії та зручності використання. Для побудови інтерфейсу користувача використано фреймворк Next.js у поєднанні з React та стилізацією за допомогою Tailwind CSS, що забезпечує сучасний та адаптивний дизайн.

Інтерфейс включає покрокову реєстрацію користувача, введення персональних даних (дата, час, місце народження), перегляд натальної карти та текстової інтерпретації. Усі дії супроводжуються валідацією та динамічними підказками, що підвищує зручність взаємодії навіть для новачків.

Серверна частина реалізована за допомогою API-ендпоінтів Next.js та FastAPI, що дозволяє обробляти запити, передавати дані до зовнішніх сервісів (генерація натальної карти, інтеграція з GPT-4o), а також зберігати результати у профілі користувача.

Основні функції серверної частини:

- авторизація та аутентифікація користувача з використанням JWT-токенів;
- побудова натальної карти через сторонній API;
- конвертація зображень у формат, сумісний із GPT-4o;
- формування текстових прогнозів за допомогою мультимодальної моделі;
- збереження результатів та історії прогнозів у базі даних.

Поєднання сучасного фронтенду та ефективної серверної логіки забезпечило стабільну, масштабовану та зручну систему, придатну до подальшого розширення функціоналу.

3.2 Розгортання веб-застосунку на хмарному середовищі

Для розгортання веб-застосунку було обрано платформу Vercel – сучасне хмарне середовище, оптимізоване для Next.js-проектів. Завдяки глибокій інтеграції з GitHub, Vercel забезпечує автоматичне безперервне розгортання

(CI/CD): кожен коміт у репозиторій ініціює оновлення продакшн-версії без додаткових дій з боку розробника.

Vercel надає зручний інтерфейс для керування середовищем виконання, налаштування змінних середовища (API-ключі, URL-адреси бекенду тощо), а також моніторингу роботи застосунку в реальному часі. Сторінки генеруються на сервері або статично, залежно від конфігурації, що забезпечує високу продуктивність і короткий час завантаження.

Завдяки використанню Vercel, розгортання стало швидким, надійним і не вимагає складних налаштувань інфраструктури.

3.3 Тестування функціональності: коректність побудови карти, стабільність генерації інтерпретацій

У процесі тестування було перевірено ключові функціональні можливості веб-застосунку: побудову натальної карти на основі введених даних та генерацію текстової інтерпретації за допомогою моделі GPT-4o. Система стабільно приймає вхідні дані (дата, час, місце народження), надсилає запит до зовнішнього API для побудови карти, а потім формує запит до мовної моделі на основі отриманого зображення. Прогноз генерується у межах від 10 до 40 секунд і відповідає заданому контексту. Усі функції протестовані як у хмарному середовищі (Vercel), так і в режимі локальної розробки.

3.4 Аналіз якості та релевантності згенерованих прогнозів

Згенеровані за допомогою моделі GPT-4o астрологічні прогнози продемонстрували високий рівень якості, логічності та тематичної відповідності введеним даним. Система формує змістовні текстові інтерпретації, які охоплюють основні елементи натальної карти – положення планет, аспекти, знаки зодіаку та будинки.

У деяких випадках може спостерігатися незначна повторюваність або надмірна обережність формулювань, що пов'язано з обмеженим обсягом вхідного контексту. Проте загальний рівень змістовності та стилістики відповідає очікуванням від сучасної інтелектуальної системи прогнозування.

Таким чином, мультимодальна генерація прогнозів забезпечує гнучке та осмислене формування текстів, що значно перевищує якість традиційних шаблонних інтерпретацій.

3.5 Оцінка продуктивності та масштабованості системи

Оцінювання продуктивності веб-застосунку було проведено за допомогою інструменту Google Lighthouse. Застосунок показав високі результати за всіма основними критеріями:

- Performance – 99/100
- Accessibility – 94/100
- Best Practices – 93/100
- SEO – 100/100

Ключові метрики продуктивності:

- First Contentful Paint – 0.6 с
- Largest Contentful Paint – 1.0 с
- Total Blocking Time – 0 мс
- Cumulative Layout Shift – 0
- Speed Index – 0.7 s

Ці показники свідчать про швидке завантаження, відсутність затримок при взаємодії, стабільну візуалізацію інтерфейсу та високу оптимізацію коду.

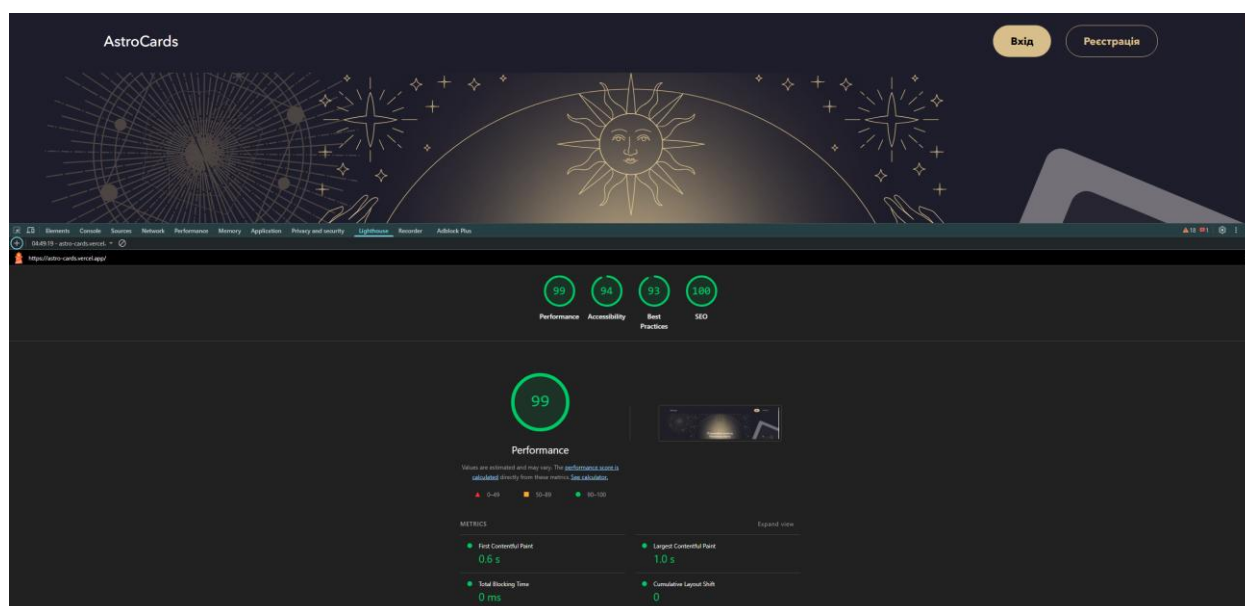


Рисунок 3.1 – Оцінка продуктивності Lighthouse

Завдяки розгортанню на платформі Vercel, система підтримує автоматичне масштабування та безперебійне обслуговування запитів користувачів. Архітектура дозволяє масштабуватися без потреби у ручному втручанні, що робить застосунок придатним для широкого впровадження.

Таким чином, веб-застосунок повністю відповідає сучасним вимогам до швидкодії, масштабованості та стабільності. Скріншот оцінки продуктивності наведено у рис. 3.1.

3.6 Аналіз переваг і обмежень запропонованої моделі

Запропонована модель веб-застосунку поєднує сучасні веб-технології, інтеграцію з мультимодальною мовною моделлю GPT-4o та інтуїтивно зрозумілий користувацький інтерфейс. Вона орієнтована на автоматизоване формування астрологічних прогнозів із високим рівнем персоналізації.

Основні переваги:

- **Мультимодальність:** система здатна аналізувати як текстові, так і графічні дані (зображення натальної карти), що дозволяє створювати глибші й точніші інтерпретації.
- **Автоматизація:** повна автоматична побудова карти та генерація прогнозу без участі астролога.
- **Сучасний інтерфейс:** реалізований за допомогою Next.js і Tailwind CSS, інтерфейс є швидким, адаптивним і зручним для користувача.
- **Висока продуктивність:** підтверджена Lighthouse-метриками (Performance – 99/100, SEO – 100/100 тощо).

Основні обмеження:

Залежність від зовнішніх сервісів: побудова карти та генерація прогнозів потребує доступу до сторонніх API, що створює ризики у разі зміни політики чи недоступності сервісів.

- **Обмеження обсягу контексту:** мовна модель має ліміти на розмір запиту, що впливає на глибину інтерпретацій.
- **Мовна локалізація:** на поточному етапі реалізовано лише одну мову інтерфейсу та виводу.

Таким чином, модель є ефективним рішенням для автоматизації астрологічних прогнозів і водночас має потенціал для подальшого вдосконалення.

3.7 Перспективи розвитку

Запропонована модель інтелектуального веб-застосунку має значний потенціал для подальшого розвитку та вдосконалення. Існує кілька напрямів, які можуть суттєво розширити функціональність системи та покращити користувацький досвід:

- **Локалізація:** додавання багатомовної підтримки дозволить адаптувати сервіс для більш широкої аудиторії та вийти на міжнародний ринок. Локалізація має охоплювати як інтерфейс, так і генерацію інтерпретацій.
- **Інтеграція чату з ШІ:** реалізація повноцінного інтерактивного чату з мультимодальною мовною моделлю дозволить користувачам отримувати прогнози в режимі діалогу, ставити додаткові запитання щодо власної натальної карти, уточнювати аспекти та отримувати розгорнуті відповіді в реальному часі.
- **Розширення функціоналу:** додавання нових типів прогнозів дозволить поглибити астрологічний аналіз і зробити сервіс більш повноцінним.

Усі ці напрямки відкривають можливості для масштабування проєкту, підвищення рівня персоналізації та залучення нових сегментів користувачів. Запропонована архітектура легко адаптується до розширення функціоналу без значних змін у базовій логіці системи.

Висновок за розділом 3

Розроблений застосунок успішно реалізує повний цикл: від введення персональних даних користувача до отримання натальної карти у вигляді зображення та її подальшої інтерпретації за допомогою штучного інтелекту. Таким чином, поставлену мету роботи досягнуто повністю.

У процесі розробки було виконано всі основні завдання:

- Створено адаптивний веб-інтерфейс з можливістю реєстрації та редагування персональних даних;

- Реалізовано генерацію натальної карти;
- Впроваджено механізм передачі зображення до GPT-4o;
- Забезпечено генерацію інтерпретації карти для користувача;

Отримані результати повністю відповідають поставленим технічним вимогам і демонструють стабільну роботу системи. Автоматичне створення прогнозу відбувається без залучення користувача до складних технічних процесів, що значно підвищує зручність використання.

На основі проведених досліджень і реалізації можна зробити висновок, що сучасні мовні моделі здатні ефективно застосовуватись у сфері астрології для автоматизованої інтерпретації графічних даних. Застосунок може бути використаний не лише в індивідуальному користуванні, а й у рамках комерційних сервісів для автоматичного створення гороскопів та консультацій.

З точки зору наукової та технічної значущості, ця розробка демонструє реальний приклад застосування мультимодального ШІ у вузькій предметній галузі. Вона також може бути базою для майбутнього розширення мультимодальних сервісів.

У соціальному контексті дана система може сприяти популяризації астрології серед широкої аудиторії, спростити доступ до інтерпретації натальних карт, а також забезпечити альтернативу дорогим або складним консультаціям.

ВИСНОВКИ

Отже, історія розвитку штучного інтелекту свідчить про поступову еволюцію від ідей і концепцій до практичних реалізацій, здатних вирішувати складні інтелектуальні завдання. Від перших автоматизованих пристроїв і логічних моделей до сучасних генеративних мовних і візуальних систем – ШІ перетворився на потужний інструмент аналізу, моделювання й комунікації.

Сьогодні штучний інтелект активно впроваджується в різноманітні сфери: освіту, медицину, культуру, побут. Його здатність працювати з природною мовою та великими обсягами даних відкриває нові можливості для автоматизованих сервісів, зокрема для генерації текстів, персоналізованих рекомендацій і прогнозів. Ці тенденції зумовлюють актуальність дослідження, спрямованого на розробку інтелектуальної системи, що поєднує ШІ та астрологічні алгоритми.

У результаті виконання кваліфікаційної роботи було реалізовано повноцінний інтелектуальний веб-застосунок, який забезпечує побудову натальної карти та її інтерпретацію на основі введених користувачем персональних даних. Для досягнення цього було інтегровано сучасні фреймворки та зовнішні сервіси, включаючи Next.js, Tailwind CSS, а також мультимодальну модель GPT-4o.

Найціннішим досягненням є реалізація повністю автоматизованого процесу: від збору вхідних даних до формування структурованої відповіді з використанням мовної моделі, яка здатна працювати з візуальною інформацією.

Було подолано низку технічних труднощів, зокрема пов'язаних з правильною передачею зображення до вибраної LLM-моделі та стабільністю відповідей штучного інтелекту (ШІ). Це покращило загальну надійність системи та її взаємодію з користувачем.

У майбутньому функціонал системи можна розширити. Наприклад:

- реалізувати нові описи та поради для користувача.
- додати багатомовну підтримку.
- зробити також мобільний застосунок.

Загалом, дана система є прикладом ефективного застосування сучасних технологій у прикладній предметній області, що має як наукову, так і практичну та соціальну значущість.

Запропонована модель сприятиме автоматизації створення персоналізованих астрологічних прогнозів, що відкриває перспективи для її використання у різних інформаційних сервісах.

Таким чином, результати кваліфікаційної роботи є актуальними, практично значущими та перспективними для подальшого розвитку.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. React Documentation [Електронний ресурс]. – Режим доступу: URL: <https://react.dev/learn> (дата звернення – 04.06.2025).
2. Next JS Documentation [Електронний ресурс]. – Режим доступу: URL: <https://nextjs.org/docs> (дата звернення – 04.06.2025).
3. Tailwind CSS Documentation [Електронний ресурс]. – Режим доступу: URL: <https://tailwindcss.com/docs/styling-with-utility-classes> (дата звернення – 04.06.2025).
4. JavaScript – MDN Web Docs [Електронний ресурс]. – Режим доступу: URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення – 04.06.2025).
5. TypeScript Documentation [Електронний ресурс]. – Режим доступу: URL: <https://www.typescriptlang.org/docs/> (дата звернення – 04.06.2025).
6. OpenAI Platform Documentation [Електронний ресурс]. – Режим доступу: URL: <https://platform.openai.com/docs/overview> (дата звернення – 04.06.2025).
7. IBM. History of Artificial Intelligence [Електронний ресурс]. – Режим доступу: URL: <https://www.ibm.com/think/topics/history-of-artificial-intelligence> (дата звернення – 04.06.2025).
8. NextAuth.js Documentation [Електронний ресурс]. – Режим доступу: URL: <https://next-auth.js.org/getting-started/introduction> (дата звернення – 04.06.2025).
9. AstrologyAPI Documentation [Електронний ресурс]. – Режим доступу: URL: <https://astrologyapi.com/docs> (дата звернення – 04.06.2025).
10. ConvertAPI Documentation [Електронний ресурс]. – Режим доступу: URL: <https://www.convertapi.com/api> (дата звернення – 04.06.2025).
11. FastAPI Documentation [Електронний ресурс]. – Режим доступу: URL: <https://fastapi.tiangolo.com/> (дата звернення – 04.06.2025).
12. Zod Documentation [Електронний ресурс]. – Режим доступу: URL: <https://zod.dev/> (дата звернення – 04.06.2025).

13. MDN Web Docs. Fetch API [Электронный ресурс]. – Режим доступа: URL: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API (дата обращения – 04.06.2025).
14. MDN Web Docs. JSON [Электронный ресурс]. – Режим доступа: URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON (дата обращения – 04.06.2025).
15. Felipe Cezar. Understanding All Types of Page Rendering in Next.js [Электронный ресурс]. – Режим доступа: URL: <https://dev.to/felipecezar01/understanding-all-types-of-page-rendering-in-nextjs-1fbi> (дата обращения – 04.06.2025).

ДОДАТКИ

Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту
Кафедра комп'ютерних систем та робототехніки
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Бакалавр**
Галузь знань: 12 – Інформаційні технології
Спеціальність: 123 «Комп'ютерна інженерія»
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри комп'ютерних
систем та робототехніки

к. ф.-м. н., доц. ХРУСЛОВ М. М.
«02» жовтня 2024 року

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

ФЬОДОРОВА Данили Романовича

(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Модель веб-додатку із використанням штучного інтелекту для генерації персоналізованих астрологічних прогнозів»**

керівник роботи **Мороз Ольга Юріївна**, PhD з інформ. технологій, доцент з во кафедри

затверджені наказом по університету від № 4101-5/962 від 16.04.2025 року

2. Строк подання студентом роботи **30 травня 2025 року**

3. Перелік питань, які потрібно розробити

1. Дослідження методів штучного інтелекту для створення астрологічних прогнозів.
2. Аналіз способів збору та обробки даних для персоналізованих прогнозів.
3. Огляд можливих ризиків і захисту персональних даних користувачів.
4. Розробка механізму авторизації та збереження профілів користувачів.
5. Вибір технологій для інтеграції штучного інтелекту у web-застосунок.
6. Створення системи генерації персоналізованих прогнозів.
7. Тестування точності та якості прогнозів.
8. Оптимізація швидкості та зручності використання сайту.

4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Затвердження теми роботи	05.09.2024 - 02.10.2024
2	Аналіз та пошук методичної літератури щодо використання штучного інтелекту для генерації персоналізованих астрологічних прогнозів	01.01.2025 – 20.01.2025
3	Огляд і аналіз способів збору та обробки даних для персоналізованих прогнозів	21.01.2025 – 10.02.2025
4	Вибір технологій для інтеграції штучного інтелекту у web-застосунок.	11.02.2025 – 28.02.2025
5	Розробка механізму авторизації та збереження профілів користувачів	01.03.2025 – 15.03.2025
6	Створення системи генерації персоналізованих астрологічних прогнозів	16.03.2025 – 05.04.2025
7	Тестування точності та якості прогнозів	06.04.2025 – 20.04.2025
8	Підготовка і оформлення звітних матеріалів. Написання статті за матеріалами кваліфікаційної роботи.	21.04.2025 – 30.04.2025
9	Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи. Оформлення списку літератури	01.05.2025 – 10.05.2025
10	Оформлення пояснювальної записки кваліфікаційної роботи відповідно вимогам до звітів про НДР.	01.05.2025- 30.05.2025-
11	Оформлення звіту про переддипломну практику	01.05.2025- 30.05.2025-
12	Представлення кваліфікаційної роботи керівнику та рецензенту	30.05.2025-

5. Дата видачі завдання *02 жовтня 2024 року.*

Студент

Фьодоров Д. Р.

ініціали, прізвище

підпис



Керівник роботи

Мороз О. Ю.

ініціали, прізвище

підпис



Додаток Б

**Технічне завдання
на розробку програмного виробу
«Модель веб-додатку із використанням штучного інтелекту для
генерації персоналізованих астрологічних прогнозів»**

1.	Введення	1.1. Назва: «Модель веб-додатку із використанням штучного інтелекту для генерації персоналізованих астрологічних прогнозів» 1.2. Галузь застосування: 12- інформаційні технології, а саме архітектури інтелектуального веб-застосунку, що поєднує методи веб-розробки з мультимодальними мовними моделями для автоматизованої побудови та інтерпретації натальної карти. Особисті онлайн-сервіси, цифрові помічники, психологічні консультації, маркетинг персоналізованих продуктів.
2.	Підстава для розробки	2.1. Навчальний план за спеціальністю: 123 «Комп'ютерна інженерія» 2.2. Завдання на кваліфікаційну роботу: Розробка веб-додатку для генерації персоналізованих астрологічних прогнозів із використанням штучного інтелекту
3.	Призначення розробки	3.1. Мета розробки програмного виробу: Створити веб-додаток, який генерує персоналізовані астрологічні прогнози на основі дати, часу й місця народження користувача із використанням ШІ. 3.2. Призначення програмного виробу: Автоматизована генерація якісних астрологічних прогнозів у зручному інтерфейсі, доступному для широкого кола користувачів. 3.3. Вихідні дані для розробки : Ім'я, дата народження, місце народження, час народження користувача; розрахована натальна карта;
4.	Технічні вимоги до програмного виробу	4.1. Вимоги до функціональних характеристик: Введення даних користувача, потім розрахунок натальної карти та формування запиту до штучного інтелекту, після цього йде генерація інтерпретації натальної карти та виведення результату 4.2. Вимоги до надійності: Стійкість до помилкових введень, повідомлення про помилки API. 4.3. Вимоги до умов експлуатації:

		Веб-браузер, підключення до Інтернету 4.4. Вимоги до складу і параметрів технічних засобів: Будь-який сучасний ПК або смартфон, браузер 4.5. Вимоги до інформаційної та програмної сумісності: Працює на ОС Windows, Linux, macOS, Android, iOS; підтримка основних браузерів 4.6. Вимоги до маркування та упаковки: немає 4.7. Вимоги до транспортування і зберігання: немає 4.8. Спеціальні вимоги: немає.	
5.	Вимоги до програмної документації	Програмною документацією до виробу «Модель веб-додатку із використанням штучного інтелекту для генерації персоналізованих астрологічних прогнозів» вважати: 1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).	
6.	Вимоги до техніко-економічних показників	Програмною документацією до виробу «Модель веб-додатку із використанням штучного інтелекту для генерації персоналізованих астрологічних прогнозів» вважати: 1) Справжнє Технічне завдання на розробку виробу (представити у вигляді Додатку Б до пояснювальної записки до кваліфікаційної роботи).	
7.	Стадії і етапи розробки	Дата	Назва етапу
		Від 05.09.2024 До 02.10.2024	Затвердження теми роботи.
		Від 01.01.2025 До 20.01.2025	Аналіз та пошук літератури щодо використання OpenAI API та методів ШІ для астрологічного прогнозування.
		Від 21.01.2025 До 10.02.2025	Аналіз способів побудови натальної карти та обробки астрологічних даних
		Від 11.02.2025 До 28.02.2025	Вибір технологій для веб-додатку: фреймворки, API.
		Від 01.03.2025 До 15.03.2025	Розробка механізму реєстрації та збереження профілів користувачів
		Від 16.03.2025 До 05.04.2025	Інтеграція генерації персоналізованих

		Від 06.04.2025 До 20.04.2025 Від 21.04.2025 До 30.04.2025 Від 01.05.2025 До 10.05.2025 Від 01.05.2025 До 30.05.2025 Від 01.05.2025 До 30.05.2025 30.05.2025	астрологічних прогнозів через OpenAI API Тестування якості згенерованих прогнозів, перевірка релевантності відповідей Підготовка та оформлення звітних матеріалів. Написання статті за результатами розробки Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи. Оформлення списку літератури Оформлення пояснювальної записки кваліфікаційної роботи відповідно вимогам до звітів про НДР. Оформлення звіту про переддипломну практику Подання кваліфікаційної роботи керівнику та рецензенту
8.	Порядок контролю і приймання програмного продукту (моделі)	1. Перевірку ходу розробки програми виконувати раз в 3 тижні. 2. Захист розробленої моделі провести на засіданні Екзменаційної комісії. 3. Пояснювальну записку подати в електронному вигляді в 1 примірнику.	

Виконавець
студент групи КІ- 41 _____  ФЬОДОРОВ Д.Р.

Замовник _____  МОРОЗ.О.Ю

Додаток В

**Програма та методика випробувань програмного виробу
«Модель веб-додатку із використанням штучного інтелекту для
генерації персоналізованих астрологічних прогнозів»**

1. Об'єкт випробувань**1.1 Найменування програмного виробу**

Веб-додаток для генерації персоналізованих астрологічних прогнозів із використанням штучного інтелекту

1.2 Область застосування

Особисті онлайн-сервіси, цифрові помічники, мобільні додатки, психологічні консультації, маркетинг персоналізованих продуктів

1.3 Умовне призначення розробки

Автоматична генерація натальної карти та індивідуального прогнозу на основі даних користувача з використанням штучного інтелекту.

2. Мета випробувань

Перевірити відповідність функціональних характеристик веб-додатку технічному завданню, зокрема:

- коректність реєстрації та входу користувача;
- автоматичну генерацію натальної карти при реєстрації;
- генерацію прогнозу при зміні профілю;
- коректність інтеграції з API (стабільність запитів, релевантність результатів);
- швидкість обробки запитів та відповідей;
- зручність інтерфейсу.

3. Загальні положення**3.1 Підстави щодо випробувань**

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

3.2 Місце та тривалість випробувань

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

3.3. «Обсяг випробувань»

Приймальні випробування програмного виробу проводяться в обсязі відповідно до цієї Програми та методики випробувань.

3.4 Організації, які беруть участь у випробуваннях

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

4. Вимоги до програмного виробу

- Після реєстрації користувача одразу генерується натальна карта та прогноз.
- При зміні даних у профілі – автоматично оновлюється інтерпретація.
- Відображення натальної карти.
- Прогноз формується за допомогою API та має бути текстовим, пов'язаним з характеристиками карти.

- Інтерфейс повинен бути адаптивним, з підтримкою мобільних пристроїв.
- Час очікування відповіді від API – не більше 30 секунд.

5. Вимоги до програмної документації

Документація повинна включати:

- технічне завдання;
- API-інструкції (структура запиту, параметри);
- методику перевірки якості результатів.

6. Засоби та порядок випробувань

6.1 Засоби випробувань

Для проведення випробувань використовується персональний комп'ютер або ноутбук з ОС Windows або macOS/Linux, браузером (Google Chrome або Firefox), підключенням до Інтернету.

Засоби тестування: Postman, DevTools, інструменти налагодження браузера.

6.2 Порядок проведення випробувань

Як правило, випробування проводяться в два етапи:

- ознайомчий (1-й етап);
- випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1. Перевірку комплектності програмної документації.
2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
3. Перевірку комплектності складу технічних і програмних засобів.
4. Методику проведення перевірок на 1 етапі випробувань.
5. Якість програмної документації перевіряється на відповідність вимогам стандартів ЕСПД.

2-й етап – функціональне тестування: перевірка фактичної роботи всіх модулів застосунку згідно з технічним завданням та вимогами користувача.

Перевірка працездатності веб-додатку

Методика Тесту 1: Реєстрація користувача

- Запуск програми у браузері за URL-адресою;
- Вибір пункту "Реєстрація";
- Введення дати, часу та місця народження;
- Підтвердження реєстрації;
- Очікуваний результат: автоматичне створення профілю, побудова натальної карти та генерація текстового прогнозу з OpenAI API.

Висновок: У результаті виконання Тесту 1 веб-додаток вважається працездатним, якщо система створює профіль і одразу виводить прогноз.

Методика Тесту 2: Оновлення профілю

- Вхід у систему під створеним обліковим записом;
- Зміна дати народження;
- Натискання «Зберегти»;
- Очікуваний результат: повторна генерація натальної карти та оновлення тексту прогнозу у профілі.

Висновок: У результаті виконання Тесту 2 система коректно реагує на зміну даних та генерує новий прогноз.

Методика Тесту 3: Перевірка роботи OpenAI API

- Надсилення сформованого prompt з інформацією натальної карти до OpenAI API;
- Отримання відповіді протягом 30 секунд;
- Оцінка змістовності й релевантності тексту.

Висновок: У результаті виконання Тесту 3 підтверджено функціональність інтеграції з API.

Методика Тесту 4: Адаптивність інтерфейсу

- Відкриття програми на мобільному пристрої;
- Навігація між основними екранами;
- Перевірка збереження повної функціональності.

Висновок: Якщо всі компоненти відображаються коректно, додаток вважається адаптивним.

Методика Тесту 5: Обробка помилкових даних

- Введення некоректної дати або логіну;
- Очікуваний результат: система виводить повідомлення про помилку, не генерує прогноз.

Виконавець

студент групи КІ- 41 _____ ФЬОДОРОВ Д.Р.

