**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**Факультет інформаційних технологій**

**Кафедра інтелектуальних технологій**

**ЗВІТ**

**з проектно-технологічної практики**

студента групи *КН - 31*

*Кошулька Ярослава Олександровича \_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис студента)*

База практики: *ТОВ “Навчальний центр “ФРЕШКОД”*

Керівник практики: *директор Лобкова Ксенія Вячеславівна*

Оцінка керівника практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**М.П.**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (*підпис керівника від підприємства*)

**підприємства**

Керівник практики: *асистент Паткін Євген Дмитрович*

Оцінка керівника практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (*підпис керівника від Університету*)

Оцінка, отримана при захисті: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (*підписи членів комісії*)

**КИЇВ 2025**

**Зміст**

[Вступ 2](#_Toc189828511)

[РОЗДІЛ 1. ХАРАКТЕРИСТИКА ПІДПРИЄМСТВА-БАЗИ ПРАКТИКИ 4](#_Toc189828512)

[1.1 Характеристика підприємства «Freshcode» 4](#_Toc189828513)

[РОЗДІЛ 2. ЗМІСТ РОБОТИ ЗА ПЛАНОМ-ГРАФІКОМ ПРАКТИКИ 5](#_Toc189828514)

[2.1 Пояснення роботи за планом-графіком практики 5](#_Toc189828515)

[РОЗДІЛ 3. ЗМІСТ РОБОТИ ЗА ІНДИВІДУАЛЬНИМ ЗАВДАННЯМ 7](#_Toc189828516)

[3.1 Опис результатів виконання практичного завдання з курсу «Знайомство із ІТ. Верстка вебсайтів» 7](#_Toc189828517)

[3.2 Опис результатів виконання практичного завдання з курсу «Мінікурс по застосуванню штучного інтелекту в програмуванні» 13](#_Toc189828518)

[3.2.1 Виділення функціональних та нефункціональних вимог 13](#_Toc189828519)

[3.2.2 Створення макету 15](#_Toc189828520)

[3.2.3 Розробка Backend частини 16](#_Toc189828521)

[3.2.4 Розробка Frontend частини 18](#_Toc189828522)

[3.2.5 Тестування роботи веб-додатку 21](#_Toc189828523)

[3.3 Опис результатів виконання завдань з курсу «Знайомство з ІТ менеджментом» 24](#_Toc189828524)

[3.3.1 Виконання завдань впродовж курсу 24](#_Toc189828525)

[3.3.2 Порівняння результатів 30](#_Toc189828526)

[Висновок 35](#_Toc189828527)

[Перелік використаних джерел 37](#_Toc189828528)

# Вступ

**Мета роботи:** ознайомлення з процесами проектування, розробки, тестування та використання компонентів комп’ютерних систем і інформаційних технологій у середовищі профільних підприємств, а також безпосередня участь студентів у цих процесах. Поглиблення теоретичних знань у галузі комп’ютерних наук, розвиток навичок роботи з інформаційними джерелами та технічною документацією. Проходження навчальних курсів базі підприємства “Навчальний центр “ФРЕШКОД”. Виконання практичних завдань у вигляді створення веб-застосунків та проходження тестових блоків.

**Для вирішення поставленої мети необхідно вирішити такі завдання:**

* ознайомитись з платформою та завданнями практики;
* успішно пройти теоретичні частини кожного курсу;
* ознайомитись з практичними завданнями на кожному курсі;
* освоїти роботу з інструментом Figma для створення та редагування дизайн-макетів;
* виконати практичні завдання курсів, використовуючи потрібні інструменти та знання, здобуті впродовж навчання та проходження практики

**Опис використаних засобів розробки:**

Для виконання практичних завдань курсів було використано низку інструментів. Дизайн-макети створювались та редагувались за допомогою онлайн-сервісу «Figma».

Верстка макету сайту з курсу «Знайомство із ІТ. Верстка вебсайтів» відбувалась за допомогою мови розмітки HTML та каскадних таблиць стилей CSS, для написання скриптів використано Javascript.

Створення веб-застосунку на курсі «Мінікурс по застосуванню штучного інтелекту в програмуванні» відбувалось за допомогою Javascript-бібліотеки для розробки користувацьких інтерфейсів React. Стилі задавались за допомогою CSS. Як СУБД використовувалась MongoDB, для реалізації сервісної частини було обрано середовище виконання Node.js та фрейморк Express.js.

Виконання практичної частини курсу «Знайомство з ІТ менеджментом» відбувалось за допомогою платформ «Miro», «GanttPro» та «Trello». Практичне завдання з порівняння виконувалось за допомогою LLM ChatGPT.

# РОЗДІЛ 1. ХАРАКТЕРИСТИКА ПІДПРИЄМСТВА-БАЗИ ПРАКТИКИ

## 1.1 Характеристика підприємства «Freshcode»

Проектно-технологічна практика проходила на базі навчального центру української компанії «Freshcode», яка надає високоякісні програмні рішення для бізнесу на різних платфомах, включаючи веб-розробку, мобільну розробку, UI/UX дизайн, тестування програмного забезпечення, послуги DevOps та багато іншого. За більш ніж 14 років свого існування компанія має більше 200 завершених проєктів, більше 150 спеціалістів в штаті та 20 країн для співпраці.

Метою навчального центру є надання студенту максимум знань та навичок, пояснення бізнес-процесу комерційних замовлень, алгоритму взаємодії із замовниками, менеджерами, тестувальниками та іншими розробниками.

Для проходження практики було запропоновано три курси різних направлень:

* «Знайомство із ІТ. Верстка вебсайтів»;
* «Мінікурс по застосуванню штучного інтелекту в програмуванні»;
* «Знайомство з ІТ менеджментом».

Кожен з них містить відео-лекції та контрольні тестові блоки. Для проходження перших двох курсів по закінченню потрібно виконати відповідні практичні завдання. Для курсу з верстки – це верстка односторінового сайту за макетом з Figma, а для курсу із застосування ШІ в програмуванні – створення базового «ToDo» веб-додатку за допомогою ресурсів та можливостей LL-моделі «ChatGPT». Курс з IT менеджменту по закінченню вимагає успішного проходження блоку тестових запитань та завдання з порівняння результатів отриманих впородовж практики з результатами, згенерованими LLM ChatGPT.

# РОЗДІЛ 2. ЗМІСТ РОБОТИ ЗА ПЛАНОМ-ГРАФІКОМ ПРАКТИКИ

## 2.1 Пояснення роботи за планом-графіком практики

Протягом проходження практики в навчальному центрі «Freshcode» виконання завдань відбувалось відповідно до затвердженого плану-графіку.

1. Спершу було проведено ознайомлення з навчальною платформою, доступними ресурсами та структурою курсів.
2. В рамках курсу «Знайомство із ІТ. Верстка вебсайтів» було закріплено базові знання щодо веб-технологій, зокрема HTML та CSS. Розглянуто основи адаптивної верстки. Практичне завдання курсу передбачало верстку макету веб-сторінки з використанням HTML та CSS. Детальніше про виконання цього завдання буде викладено в розділі 3.
3. На курсі «Мінікурс по застосуванню ШІ в програмуванні» проведено ознайомлення з методами застосування штучного інтелекту в програмуванні, зокрема LLM ChatGPT. Фінальним завданням на курсі було створення ToDo веб-додатку, в процесі роботи рекомендувалось використовувати LLM ChatGPT. Детальне пояснення ходу розробки додатку наведено в розділі 3.
4. Курс «Знайомство з ІТ-менеджментом» надав основи управління IT-проєктами, методами планування та контролю процесів розробки. Проведено ознайомлення з поняттями mind mapping, risk catalog та WBS, які були закріплені за допомогою практичних завдань з організації процесу роботи над реальним проєктом «Нескорені». Було отримано досвід роботи в інструментах GanttPRO, Miro та Trello, кожен з них виконував окрему роль в організації роботи. Календарний графік створювався за допомогою Google Spreadsheets та GanttPRO, платформа Trello допомагала симулювати реальний потік робіт команди на проєкті. Miro використовувалась для збереження всієї потрібної інформації у вигляді таблиць, зображень та посилань. Скріншоти проведеної роботи з поясненнями наведені в розділі 3.

# РОЗДІЛ 3. ЗМІСТ РОБОТИ ЗА ІНДИВІДУАЛЬНИМ ЗАВДАННЯМ

## 3.1 Опис результатів виконання практичного завдання з курсу «Знайомство із ІТ. Верстка вебсайтів»

Фінальним практичним завданням на курсі з верстки було створення макету веб-сторінки за допомогою HTML та CSS. Для верстки пропонувався наступний макет (рисунок 3.1)

A group of people in hats

Description automatically generated

Рис. 3.1

Для виконання було запропоновано два рівні складності:

* рівень складності простий: зверстати перший екран сайту.
* рівень складності середній: зверстати всю сторінку.

В рамках практики було прийнято рішення відтворити всю сторінку. Для візуалізації структури сайту створено UML діаграму компонентів:

A diagram of a website

Description automatically generatedРис. 3.2 – Структура сайту у вигляді UML діаграми компонентів

Сайт є адаптивним, використовує для цього відносні одиниці виміру, медіа-запити та Javascript скрипти. Скрипти реалізованого бургер-меню та адаптивного хедеру зберігаються у файлі header.js, який має алгоритм, що поданий на рисунку 3.3.

A diagram of a project

AI-generated content may be incorrect.

Рис. 3.3 – Алгоритм роботи скрипту header.js у вигляді UML діаграми активності

Для демонстрації адаптивності створеного сайту на рисунках 3.4.1 – 3.6 наведені скріншоти вигляду сайту на ПК, планшеті та телефоні.

A group of people in hats and blankets

Description automatically generated

Рис. 3.4.1 – Зовнішній вигляд сайту при використанні ПК

A person walking a bicycle

Description automatically generated

Рис. 3.4.2 – Зовнішній вигляд сайту після скролу при використанні ПК

A screenshot of a phone

Description automatically generated

Рис. 3.5.1 – Зовнішній вигляд сайту при використанні планшету

A screenshot of a cell phone

Description automatically generated

Рис. 3.5.2 – Зовнішній вигляд сайту з відкритим меню при використанні планшету

A screen shot of a phone

Description automatically generated

Рис. 3.6.1 – Зовнішній вигляд сайту при використанні телефону

A screenshot of a phone

Description automatically generated

Рис. 3.6.2 – Зовнішній вигляд сайту з відкритим меню при використанні телефону

## 3.2 Опис результатів виконання практичного завдання з курсу «Мінікурс по застосуванню штучного інтелекту в програмуванні»

### 3.2.1 Виділення функціональних та нефункціональних вимог

Фінальне завдання до курсу із застосування ШІ в програмуванні полягало у створенні веб-додатку “ToDo List”, що дозволяє:

* додавати завдання;
* відмічати завдання як виконані;
* видаляти завдання

Також було запропоновано два рівня (базовий та просунутий). У другому варіанті до frontend потрібно було додати backend, а дані зберігати у базі даних. Було прийнято рішення створити повноцінний веб-додаток, який би дозволяв створювати замітки та керувати ними будь-якому користувачу. Для досягнення цілі було виділено функціональні та нефункціональні вимоги (таблиці 3.1 – 3.2):

Таблиця 3.1 – Функціональні вимоги

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Назва функції | Роль | Опис | Функціональні вимоги |
| UC1 | Додавання завдань | Користувач | Додавання завдань, які потрібно виконати. | FR-1: Користувач повинен мати змогу додавати завдання |
| UC2 | Видалення завдань | Користувач | Видалення раніше створених завдань | FR-2: Користувач повинен мати змогу видаляти раніше створені завдання |
| UC3 | Редагування завдань | Користувач | Редагування інформації про завдання (назва, дата та час закінчення, пріоритет, статус виконання) | FR-3: Користувач повинен мати змогу змінювати назву завдання, час та дату закінчення, його пріоритет та позначати завдання як виконане/не виконане |
| UC4 | Використання Google-акаунту для входу | Користувач | Вхід до додатку за допомогою Google-акаунту | FR-4: Користувач повинен мати змогу використовувати Google-акаунт для входу в додаток |
| UC5 | Створення секцій/блоків завдань | Користувач | Створення секцій/блоків завдань | FR-5: Користувач повинен мати змогу створювати секції /блоки завдань для їх класифікації |
| UC6 | Редагування назви секції/блоку | Користувач | Редагування назви створеної секції/блоку | FR-6: Користувач повинен мати змогу змінювати назву у створеної секції/блоку |
| UC7 | Видалення секції/блоку | Користувач | Видалення створеної секції/блоку | FR-7: Користувач повинен мати змогу видаляти раніше створені секції/блоки |
| UC8 | Видалення завдань разом із секцією/блоком | Система | Каскадне видалення створених завдань, що належать до секції/блоку, що підлягає видаленню | FR-8: Система повинна автоматично видаляти всі наявні завдання в секції/блоці, за умови видалення батькіської секції/блоку |

Таблиця 3.2 – Нефункціональні вимоги

|  |  |
| --- | --- |
| Ідентифікатор | Опис |
| BF-1 | Додаток повинен забезпечувати високу швидкість роботи при наявності великої кількості завдань |
| Q-1 | Додаток повинен мати оптимізований код для швидкої відповіді на дії користувача та мінімізації використання ресурсів системи |
| Q-2 | Додаток повинен бути стійким до помилок користувача та правильно обробляти відповідні виключні ситуації |
| BF-2 | Додаток повинен використовувати JWT для аутентифікації та авторизації користувачів, забезпечуючи безпечний доступ до захищених ресурсів. Токени повинні генеруватися сервером після успішної аутентифікації і передаватися на клієнтську сторону для подальшого використання при кожному запиті. |
| Q-3 | Термін дії JWT повинен бути обмежений. |

### 3.2.2 Створення макету

Після виділення всіх вимог було створено макет додатку у Figma (рисунок 3.7)

A screenshot of a computer

Description automatically generated

Рис. 3.7 – Макет веб-додатку створений у Figma

### Розробка Backend частини

За завданням створення додатку має відбуватись за допомогою можливостей ChatGPT. Сформую запит та дізнаюсь, які інструменти краще обрати для створення серверної частини. Результат запиту наведено на рисунку 3.8

A screenshot of a computer program

AI-generated content may be incorrect.

Рис. 3.8 – Результат запиту до ChatGPT

Отже, для створення серверної частини додатку було обрано Node.js, Express.js та СУБД MongoDB.

* Node.js був обраний завдяки своїй високій продуктивності, що дозволяє ефективно обробляти велику кількість одночасних запитів. Це ідеально підходить для створення масштабованих веб-додатків. Express.js використовується як фреймворк для Node.js, оскільки він дозволяє швидко створювати RESTful API, зручний для інтеграції з клієнтською частиною додатку. Його гнучкість і простота в налаштуванні дозволяють ефективно організувати маршрутизацію та обробку запитів.
* MongoDB була вибрана як СУБД завдяки своїй схемній гнучкості та можливості зберігання великих обсягів неструктурованих даних. Це також дозволяє легко масштабувати базу даних, що важливо для додатків, які потребують високої доступності та масштабованості.

Для проєктування бази даних було виділено сутності предметної області (таблиця 3.3)

Таблиця 3.3 – Сутності БД

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва атрибуту | Ідентифікатор | Тип значення |
| Сутність «Завдання» | | | |
| 1 | Ідентифікатор | \_id | ObjectId |
| 2 | Ідентифікатор типу | sectionId | ObjectId |
| 3 | Назва | title | String |
| 4 | Кінцева дата | dueDate | Date |
| 5 | Пріоритетність | priority | Number |
| 6 | Статус виконання | isCompleted | Boolean |
| Сутність «Тип Завдання» | | | |
| 1 | Ідентифікатор | \_id | ObjectId |
| 2 | Назва | title | Name |
| 3 | Ідентифікатор власника | ownerId | ObjectId |
| Сутність «Користувач» | | | |
| 1 | Ідентифікатор | \_id | ObjectId |
| 2 | Посилання на аватар | avatarUrl | String |

### 3.2.4 Розробка Frontend частини

В якості фрейморку для клієнтської частини було обрано React, для цього є декілька причин:

* react має чудову сумісність з Node.js та MongoDB (MERN stack). Цей стек технологій забезпечує безшовну інтеграцію між серверною та клієнтською частинами додатку, що дозволяє значно спростити розробку та підтримку системи;
* висока популярність вирішує проблеми відсутності навчальних матеріалів;
* react забезпечує можливий розвиток додатку з використанням React Native.

Для запобігання майбутніх проблем зі створенням архітектури веб-додатку було створено діаграму компонентів, на якій показано взаємодію та залежності всіх компонентів (рисунок 3.9)

A screenshot of a diagram

Description automatically generated

Рис. 3.9 – Структура додатку у вигляді UML діаграми компонентів

Для зображення базового випадку роботи веб-додатку в динаміці створено діаграму послідовності (рисунок 3.10)

A screenshot of a diagram

AI-generated content may be incorrect.

Рис. 3.10 – Алгоритм роботи додатку у вигляді діаграми послідовності

В ході розробки ChatGPT використовувався для знаходження багів коду, виправлення помилок та консультацій. Деякі з запитів та відповідей на них наведені на рисунках 3.11 – 3.12

A screenshot of a computer program

AI-generated content may be incorrect.

Рис. 3.11 – Приклад запиту з виправлення помилок

A screenshot of a computer

AI-generated content may be incorrect.

Рис. 3.12 – Приклад запиту з консультацією

### 3.2.5 Тестування роботи веб-додатку

Для ефективного тестування додатку було створено набір відповідних тест-кейсів для охоплення різноманітних аспектів програми. Їх опис наведено в таблиці 3.4

Таблиця 3.4 – Опис створених тест кейсів

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Опис** | **Передумови** | **Кроки** | **Очікуваний результат** |
| ТС0001 | Авторизація в системі | Сайт успішно завантажився | Натиснути на кнопку аватару  Увійти в акаунт | Успішна авторизація, відображення даних користувача |
| ТС0002 | Вихід із системи | Авторизація успішна | Натиснути на кнопку аватару | Успішний вихід із системи, дані користувача не відображуються |
| ТС0003 | Створення секції завдань | Авторизація успішна | Натиснути на кнопку «Add» у списку секцій завдань | Успішне створення секції із завданнями |
| ТС0004 | Створення завдання | Авторизація успішна | Натиснути на кнопку «Add» у списку завдань | Успішне створення завдання. Успішне створення завдання та секції (якщо секція не була створена раніше) |
| ТС0005 | Редагування секції | Успішне створення секції | Натиснути на назву секції Ввести нову назву | Успішна зміна назви |
| ТС0006 | Редагування завдання | Успішне створення завдання | Натиснути на потрібний параметр завдання, змінити параметр | Успішна зміна потрібного параметру |
| ТС0007 | Адаптивність на ПК | Сайт успішно завантажився | Використання девайсу, ширина якого більше за 1024px | Коректне відображення та функціонування всіх елементів інтерфейсу |
| ТС0008 | Адаптивність на планшетах | Сайт успішно завантажився | Використання девайсу, ширина якого менша за 1024px | Коректне відображення та функціонування всіх елементів інтерфейсу |
| ТС0009 | Адаптивність на телефонах | Сайт успішно завантажився | Використання девайсу, ширина якого менша за 375px | Коректне відображення та функціонування всіх елементів інтерфейсу |

Для зручного збору, аналізу та представлення основних результатів тестування, що було проведено за створеними тест кейсами наведено звіт по тестуванню (таблиця 3.5), в якому по стовпчикам представлено ID тест кейса, його короткий опис та позначення результату тестування.

Таблиця 3.5 – Звіт по тестуванню

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Опис** | **Статус** | **Примітка** |
| ТС0001 | Авторизація в системі | Пройдено | - |
| ТС0002 | Вихід із системи | Пройдено | - |
| ТС0003 | Створення секції завдань | Пройдено | - |
| ТС0004 | Створення завдання | Пройдено | - |
| ТС0005 | Редагування секції | Пройдено | - |
| ТС0006 | Редагування завдання | Пройдено | - |
| ТС0007 | Адаптивність на ПК | Пройдено | - |
| ТС0008 | Адаптивність на планшетах | Пройдено | - |
| ТС0009 | Адаптивність на телефонах | Пройдено | - |

Для тестування адаптивності використовувались вбудовані можливості developer tools Google Chrome. Скріншоти з тестування наведено на рисунках 3.13 – 3.15

A screenshot of a computer

Description automatically generated

Рис. 3.13 – Вигляд веб-додатку на ПК

A screenshot of a computer

Description automatically generated

Рис. 3.14 – Вигляд веб-додатку на планшеті

A screenshot of a task

Description automatically generated

Рис. 3.15 – Вигляд веб-додатку на телефоні

## Опис результатів виконання завдань з курсу «Знайомство з ІТ менеджментом»

### 3.3.1 Виконання завдань впродовж курсу

Курс «Знайомство з ІТ менеджментом» мав практичне завдання, яке виконувалось разом з ментором в ході курсу та тестові тести за кожен змістовний блок.

Практична цінність полягала в реальному завданні з організації рекламної компанії проєкту «Нескорені». Перші дві лекції були націлені на ознайомлення з середовищем Miro, аналізу вхідних даних до проєкту, створення project outline та mind map (рисунки 3.16 – 3.17)

A screenshot of a phone

Description automatically generated

Рис. 3.16 – Вхідні дані у вигляді project outline

A diagram of a graph

Description automatically generated

Рис. 3.17 – Кроки маркетингової компанії проєкту у вигляді mind map

Для попередження ймовірних ризиків було створено risk catalog (рисунок 3.18)

A screenshot of a computer

Description automatically generated

Рис. 3.18 – Таблиця ризиків проєкту

Після аналізу проєкту та постановки задачі були визначені зацікавлені сторони проєкту, які були зведені до таблиці. Серед них є як проєктна команда, так і дистриб’ютори (рисунок 3.19)

A screenshot of a phone

Description automatically generated

Рис. 3.19 – Таблиця стейкхолдерів

Аналіз зацікавлених сторін та mind map допоміг створити WBS (work breakdown structure) – список робіт по проєкту, роботи були згруповані за тематикою та обмежені за часом. WBS висвітлено в таблиці 3.6

Таблиця 3.6 – Таблиця завдань по проєкту

|  |  |  |
| --- | --- | --- |
| Задача | Години | Коментарі |
| **Аналіз ЦА** |  |  |
| Зібрати інформацію про глядача | 32 |  |
| Скласти портрет глядача | 12 |  |
| **Обрати канали для просування** | **4** |  |
| **Скласти список медіа-каналів** |  |  |
| Instagram, Telegram,Tiktok | 10 |  |
| Youtube | 6 |  |
| Телеканали | 4 |  |
| Власники білбордів | 8 |  |
| Громадські заклади | 8 |  |
| Домовитись з власниками медіа-каналів |  |  |
| Описати формат поставок | 20 | Вхідні дані |
| Прописати та заключити договір | 24 |  |
| **Створити сторінки** |  |  |
| Instagram | 4 |  |
| Telegram | 4 |  |
| Facebook | 2 |  |
| Tiktok | 2 |  |
| Google ads | 2 |  |
| Youtube | 2 |  |
| **Розробити креативну концепцію** |  |  |
| Підготувати рекламний текст | 20 |  |
| Розробити рекламні банери | 40 |  |
| Підготувати креативи | 40 |  |
| **Розробити контент-план** | **16** |  |
| **Протестувати креативи** | **8** |  |
| **Затвердити маркетингові матеріали** | **8** |  |
| **Затвердити бюджет по проєкту** | **8** |  |
| **Підготовка до івентів** |  |  |
| Підготувати та надати рекламні матеріали | 16 |  |
| Написати та розіслати запрошення | 12 |  |
| Підготувати прес-реліз до фільму | 12 |  |
| Провести репетицію інтерв'ю | 16 |  |
| Підготувати анкету | 16 |  |
| **Запустити рекламу згідно з контент-планом** |  |  |
| **Проаналізувати ефективність рекламних каналів** |  |  |
| Аналіз ефективносіт користувачів в соц мережах | 32 |  |
| Аналіз ефективності користувачів на сторінках кінотеатрів | 8 |  |
| Скласти проміжний звіт | 8 |  |
| Адаптувати канали просування | 20 |  |
| Адаптувати рекламу | 40 |  |
| Моніторинг продажу квитків в кінотеатрах | 16 |  |
| **Проведення івентів** |  |  |
| Провести опитування | 4 |  |
| Створити звіт з відгуками | 8 |  |
| **Аналіз результатів кампанії** |  |  |
| Звіт по таргетованій рекламі | 4 |  |
| Звіт по рекламах | 4 |  |
| Звіт по контекстній рекламі | 4 |  |
| Звіт по кінотеатрам | 4 |  |
| Звіт по відгукам від глядачів | 6 |  |
| Підсумковий звіт | 6 |  |
| **Загальна кількість годин:** | **520** |  |

На курсі було проведено ознайомлення з інструментом GanttPro, за допомогою якого було створено календарний графік робіт зображений на рисунку 3.20

A screenshot of a computer

Description automatically generated

Рис. 3.20 – Календарний графік робіт в інструменті GanttPro

В завершенні курсу обов’язковим до проходження був контрольний тест, результати тестування зображені на рисунку 3.21



Рис. 3.21 – Результати фінального тестування

### 3.3.2 Порівняння результатів

Отримане завдання було виконано за допомогою використання LLM ChatGPT, отримані результати потрібно порівняти з результатами, що були отримані впродовж курсу. Для початку потрібно правильним чином задати промт, вказати роль виконавця, надати потрібні матеріали. Для роботи використовувалась модель 4o. Скріншот з частиною промту подано на рисунку 3.22.

A screenshot of a computer

Description automatically generated  
Рисунок 3.22 – Промт для ChatGPT

Наступним кроком було задано наступні задачі:

* створити список стейкхолдерів;
* створити risk catalog;
* створити WBS (з оцінкою задач в годинах);
* створити календарний графік робіт;
* запропонувати платформу для контролю командних робіт.

Результати виконаних завдань наведено на рисунках 3.23 – 3.27

A screenshot of a computer

Description automatically generated

Рис. 3.23 – Створений список стейкхолдерів

A screenshot of a computer

Description automatically generated

Рис. 3.24 – Створений Risk Catalog

A screenshot of a computer

Description automatically generated

Рис. 3.25 – Створений WBS

A white sheet with black text

Description automatically generated

Рис. 3.26 – Створений календарний графік робіт

A screenshot of a computer

Description automatically generated

Рис. 3.27 – Запропоновані платформи для організації виконання командних задач

ChatGPT версії 4o надав хороші відповіді на кожне з моїх питань.

* список стейкхолдерів безумовно потребує корегування та доповнення людиною, було не згадано конкретні професії (копірайтер, аналітик і т.д.), проте представляє окремі підрозділи або компанії, що зацікавлені в проєкті, за потребою їх можна деталізувати;
* створений каталог ризиків комплексно описує головні ризики на проєкті з варіантами їх вирішення та варіантами попередження. Потребує лише доповнення та за потреби мінімального корегування з боку людини;
* WBS потребує редагування, адже деякі занесені туди роботи вже є виконаними і це позначено у вхідних даних. Проте, за графіком всі роботи з маркетингу будуть завершені вчасно, а на осінь буде винесено лише збір відгуків та формування звітності;
* календарний графік було побудовано за WBS, тому проблеми залишаються такими ж;
* запропоновані сервіси для контролю командних робіт є ширикозастосовними, інформація надана правдива та якісна.

Аналізуючи всі отримані результати можна сказати, що ChatGPT 4o може стати помічником проект-менеджера на реальному проєкті, відповіді не ідеальні, потребують корегування, проте – виконують базові вимоги, подають ідеї та можуть стати основою для роботи, яку продовжить проект-менеджер.

# Висновок

В процесі проходження проектно-технологічної практики було виконано комплекс завдань, спрямованих на набуття практичних навичок у сфері веб-розробки, застосування ШІ в програмування та IT-менеджменту

Протягом практики було детально ознайомлено з сучасними підходами до розробки програмного забезпечення, основами верстки веб-сторінок, принципами використання штучного інтелекту в програмуванні та основами управління IT-проєктами. Практичні завдання, виконані в рамках курсів, дозволили закріпити отримані знання шляхом створення реальних проєктів, таких як адаптивний веб-сайт та веб-додаток «ToDo List».

Під час проходження курсу з верстки веб-сайтів було розглянуто HTML, CSS, адаптивний дизайн, а також створено повноцінний веб-сайт на основі макета з Figma. Додатково було розроблено UML-діаграми для опису структури сайту та алгоритмів його роботи.

В межах курсу з використання ШІ в програмуванні було реалізовано веб-додаток «ToDo List», для чого використано сучасний технологічний стек: Node.js, Express.js, MongoDB та React. Було розроблено архітектуру додатку, визначено функціональні та нефункціональні вимоги, створено та протестовано API, а також реалізовано взаємодію фронтенду з бекендом.

Курс з IT-менеджменту допоміг розширити знання у сфері управління проєктами, включаючи використання таких інструментів, як Trello, GanttPRO та Miro. Всі знання були набуті в процесі практики на реальному проєкті «Нескорені». Це дозволило краще зрозуміти процес організації роботи команди та планування проєктів у реальних умовах. Отримані результати були порівняно з відповідями ChatGPT 4o, встановлено, що моделі такого типу можуть значно пришвидшити робочий процес та надати базові відомості або чорнові варіанти робіт.

Створені проєкти та набутий досвід можуть бути використані в подальшій професійній діяльності. А створені проєкти можна масштабувати в повноцінні незалежні pet-проєкти.

# Перелік використаних джерел

1. Node.js – Запускайте JavaScript будь-де. *Node.js – Run JavaScript Everywhere*. URL: <https://nodejs.org/uk>.
2. Express - Node.js web application framework. *Express - Node.js web application framework*. URL: <https://expressjs.com/>
3. Як використовувати JSON Web Tokens (JWT) для автентифікації. *DevZone*. URL: <https://devzone.org.ua/post/iak-vykorystovuvaty-json-web-tokens-jwt-dlia-avtentyfikatsiyi>.
4. What is OAuth 2.0 and what does it do for you? - Auth0. *Auth0*. URL: <https://auth0.com/intro-to-iam/what-is-oauth-2>
5. Quick Start – React. *React*. URL: <https://react.dev/learn>