

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «РОЗРОБКА ІНФОРМАЦІЙНОГО  
ВЕБРЕСУРСУ З КІБЕРСПОРТУ»

Виконав: студент 3 курсу, групи 6.1211-пі-с  
спеціальності 121 інженерія програмного забезпечення  
(шифр і назва спеціальності)

програмна інженерія (зі скороченим  
освітньої програми терміном навчання)  
(назва освітньої програми)

А.В. Козоріз

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,  
доцент, к.т.н. Лимаренко Ю.О.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри електроніки, інформаційних систем  
та програмного забезпечення ІННІ ЗНУ,  
доцент, к.т.н. Заяц В.І.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія (зі скороченим терміном навчання)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри програмної  
інженерії, к.ф.-м.н., доцент

\_\_\_\_\_ Лісняк А.О.

(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**З А В Д А Н Н Я**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Козорізу Артему Віталійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка інформаційного вебресурсу з кіберспорту

керівник роботи Лимаренко Юлія Олексіївна, к.т.н., доцент

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

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

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

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Розробка вебсайту на тематику кіберспортивних змагань.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

презентація до захисту

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 25.12.2023 р.

**КАЛЕНДАРНИЙ ПЛАН**

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	09.01.2024	
2.	Збір вихідних даних.	14.02.2024	
3.	Обробка методичних та теоретичних джерел.	06.03.2024	
4.	Розробка першого та другого розділу.	12.04.2024	
5.	Розробка третього розділу.	20.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	27.05.2024	
7.	Захист кваліфікаційної роботи.	21.06.2024	

Студент \_\_\_\_\_  
(підпис)

А.В. Козоріз  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

Ю.О. Лимаренко  
(ініціали та прізвище)

**Нормоконтроль пройдено**

Нормоконтролер \_\_\_\_\_  
(підпис)

А.В. Столярова  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка інформаційного вебресурсу з кіберспорту»: 46 с., 24 рис., 1 табл., 10 джерел.

АДАПТИВНИЙ ДИЗАЙН, БАЗИ ДАНИХ, БЕКЕНД, ВЕБРОЗРОБКА, ІНФОРМАЦІЙНИЙ РЕСУРС, КІБЕРСПОРТ, ТЕСТУВАННЯ, ФРОНТЕНД, ХОСТИНГ, АРІ.

Об'єкт дослідження – розробка вебсайту на тематику змагань з кіберспорту з використанням АРІ запитів.

Мета роботи: розробити сучасний та інтерактивний інформаційний вебресурс, присвячений кіберспорту. Цей ресурс має стати платформою для інформування користувачів про останні новини, події, турніри, команди та гравців у світі кіберспорту. Він також включатиме аналітичні статті, рейтинги та результати матчів, що дозволить шанувальникам та професіоналам зберігати зв'язок з останніми тенденціями в індустрії.

Метод дослідження – аналіз і порівняння технологій розробки інформаційних систем; вивчення класифікацій та принципів проектування інформаційних систем; проведення порівняльного аналізу для вибору програмного забезпечення.

## SUMMARY

Bachelor's qualifying paper "Development of an Information Web Resource on eSports": 46 pages, 24 figures, 1 table, 10 references.

API, ADAPTIVE DESIGN, BACKEND, DATABASES, ESPORTS, FRONTEND, HOSTING, INFORMATION RESOURCE, TESTING, WEB DEVELOPMENT

Object of research – development of a website on the topic of esports competitions using API requests.

Purpose: to develop a modern and interactive information web resource dedicated to esports. This resource should become a platform for informing users about the latest news, events, tournaments, teams, and players in the world of esports. It will also include analytical articles, ratings, and match results, allowing fans and professionals to keep in touch with the latest trends in the industry.

Research methods: analysis and comparison of information system development technologies; study of classifications and principles of information system design; comparative analysis for software selection.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат .....	4
Summary .....	5
Вступ.....	8
1 Аналіз предметної області.....	10
1.1 Опис об'єкту автоматизації.....	10
1.2 Обґрунтування необхідності автоматизації .....	11
1.3 Функціональні вимоги.....	11
1.4 Дослідження аналогів сайту.....	12
1.5 Формування вимог до майбутнього вебсайту .....	17
2 Проєктування вебсайту.....	19
2.1 Обґрунтування вибору засобів розробки.....	19
2.1.1 javascript.....	19
2.1.2 React.....	21
2.1.3 Material UI.....	22
2.1.4 Axios .....	23
2.1.5 Sass.....	24
2.1.6 API .....	26
2.1.7 opendota .....	27
2.1.8 Openproj.....	28
2.2 План реалізації.....	29
2.3 Діаграма прецедентів.....	30
2.4 Макет вебсайту .....	32
3 Реалізація проєкту вебсайту.....	33
3.1 Налаштування програмного середовища .....	33
3.2 Router.js .....	34
3.3 Система API запитів .....	36

3.4 Меню та інші сторінки .....	43
Висновки .....	45
Перелік посилань.....	46

## ВСТУП

Кіберспорт, як один із найдинамічніших напрямів у світі розваг, набуває все більшої популярності. З кожним роком зростає кількість гравців, уболівальників та турнірів, що робить необхідність доступу до оперативної та надійної інформації надзвичайно актуальною. Створення спеціалізованих інформаційних вебресурсів дозволяє забезпечити користувачів останніми новинами, статистикою матчів, аналітичними матеріалами та іншими важливими даними про кіберспорт.

Розвиток технологій та широке розповсюдження інтернету сприяють тому, що інформаційні вебресурси стають основним джерелом новин для багатьох людей. Користувачі можуть швидко та зручно отримувати актуальну інформацію про турніри, команди, гравців та результати матчів. Це дозволяє фанатам кіберспорту бути в курсі всіх подій, а професійним гравцям краще розуміти ситуацію світовому рейтингу та можливість проводити аналіз майбутнього суперника.

Для бізнесів, які працюють у сфері кіберспорту, наявність таких ресурсів також є важливою. Вони допомагають відстежувати ринкові тенденції, аналізувати конкурентів та адаптувати свої стратегії для залучення більшої кількості клієнтів. Крім того, інформаційні вебресурси можуть служити платформою для реклами та просування різних продуктів та послуг, пов'язаних з кіберспортом.

Мета дослідження – розробка інтелектуальної інформаційної системи на тему «кіберспорт».

Завдання дослідження – розглянути особливості та існуючі рішення вебсайтів на тему кіберспорту. Проаналізувати сучасні підходи та технології до розробки складних вебсайтів і вибрати відповідну для створення інтерактивного та інформаційно насиченого вебсайту на тему кіберспорту.

Об'єктом дослідження є процес розробки інформаційного вебресурсу з кіберспорту.



Предметом дослідження є технології створення власного інформаційного ресурсу для прихильників кіберспорту для перегляду головних новин та статистики з цієї тематики.

Методи дослідження – аналіз і порівняння технологій розробки інформаційних систем; вивчення класифікацій та принципів проектування інформаційних систем; проведення порівняльного аналізу для вибору програмного забезпечення.

Практичне значення одержаних результатів дослідження полягає у створенні вебсайту на тему кіберспорту, який надаватиме корисну інформацію про різні аспекти кіберспорту. Результати дослідження можуть бути використані для розробки інтерактивної платформи, де користувачі зможуть отримувати останні новини, аналітику, огляди турнірів, та іншу релевантну інформацію, що сприятиме популяризації кіберспорту і покращенню користувацького досвіду.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Опис об'єкту автоматизації

Основна мета проєктування полягає у створенні інформаційного сайту для прихильників кіберспорту. Цей сайт має бути лаконічним та зрозумілим, надаючи актуальну інформацію про кіберспорт. У рамках цього проєкту ми прагнемо досягти наступних цілей.

*Актуальність інформації:* вебсайт повинен оперативно відображати найсвіжіші новини та події з світу кіберспорту. Це включає висвітлення турнірів, результатів матчів, аналітичних статей, інтерв'ю з гравцями та інші релевантні матеріали.

*Лаконічність і зрозумілість:* інформація на сайті повинна бути подана в такому форматі, щоб користувачі могли швидко знаходити і сприймати необхідні дані. Це передбачає використання інтуїтивно зрозумілих меню, зручної навігації та чіткої структури сторінок.

*Відкритість і масштабованість:* сайт має бути побудований таким чином, щоб його можна було легко розширювати та оновлювати без значних зусиль. Це означає використання сучасних технологій та фреймворків, що дозволяють легко додавати нові функції і розділи.

*Доступність для редагування:* адміністрування сайту має бути простим і доступним, щоб відповідальні за контент особи могли безперешкодно вносити зміни і доповнення. Це передбачає наявність зручної системи управління контентом (CMS).

*Зручний інтерфейс:* інтерфейс сайту повинен бути зрозумілим для широкої аудиторії, включаючи тих, хто звик користуватися іншими вебсайтами. Це означає, що дизайн має бути сучасним і водночас простим, без зайвих елементів, але з усією необхідною інформацією.

*Підтримка різних платформ:* сайт повинен коректно відображатися на різних пристроях – як на комп'ютерах, так і на мобільних пристроях.

Адаптивний дизайн є ключовим фактором для досягнення цієї мети.

Таким чином, кінцевий результат повинен представляти собою відкриту, зрозумілу, масштабовану та легку у редагуванні систему, яка відповідає потребам і очікуванням прихильників кіберспорту. Важливо забезпечити високий рівень юзабіліті, щоб користувачі могли з легкістю знаходити потрібну інформацію та отримувати задоволення від взаємодії з сайтом.

## **1.2 Обґрунтування необхідності автоматизації**

Розміри нової інформації, яка щодня з'являється, великі, тому потрібно записувати їх у цифровому форматі. Це не лише зручно для подальшого аналізу, але й необхідно для користувачів Інтернету. На сьогоднішній день майже немає завдання, якого б не можна було вирішити через Інтернет. Від покупки товарів до отримання інформації – все доступно в мережі. Вебресурси потрібні для перегляду новин кіберспорту, огляду статистики та іншої тематичної інформації.

## **1.3 Функціональні вимоги**

Функціональні вимоги:

- публікація розкладу турнірів та матчів Dota2;
- перегляд трансляцій матчів прямо на вебсайті;
- подання детальної інформації про гравців, команди, результати матчів та статистику;
- можливість коментувати матчі та обговорювати їх у спільноті глядачів;
- сповіщення про найближчі турніри та матчі через електронну пошту або повідомлення.

## 1.4 Дослідження аналогів сайту

Провівши попередній аналіз, зведемо переваги та недоліки аналогів сайту до таблиці 1.1.

Таблиця 1.1 – Результати дослідження аналогів

№	Назва	Функціонал	Переваги	Недоліки
1	cyber-score	<ul style="list-style-type: none"> <li>– авторизація</li> <li>– перегляд матчів</li> <li>– перегляд новин/статей</li> <li>– перегляд інформації про команди та турніри</li> </ul>	<ul style="list-style-type: none"> <li>– функціональність</li> <li>– адаптивний інтерфейс</li> </ul>	– перевантаженість рекламою
2	dltv	<ul style="list-style-type: none"> <li>– авторизація</li> <li>– перегляд матчів</li> <li>– перегляд новин/статей</li> <li>– перегляд інформації про команди та турніри</li> <li>– перегляд загального рейтингу команд та гравців</li> </ul>	<ul style="list-style-type: none"> <li>– простий та зручний інтерфейс</li> <li>– адаптивний інтерфейс</li> </ul>	– відсутність української мови
3	dota2.net	<ul style="list-style-type: none"> <li>– авторизація</li> <li>– перегляд матчів</li> </ul>	<ul style="list-style-type: none"> <li>– простий та зручний інтерфейс</li> </ul>	– застарілий дизайн

Продовження таблиці 1.1

№	Назва	Функціонал	Переваги	Недоліки
3	dota2.net	– перегляд новин/статей – перегляд інформації про команди та турніри	– адаптивний інтерфейс – має власні сервіси з торгівлі/обміну внутрішньо ігровими предметами – в порівнянні з іншими аналогами, має ширший функціонал	
4	navi.gg	– авторизація – перегляд матчів – перегляд новин/статей – перегляд інформації про команди та турніри	– простий та зручний інтерфейс – адаптивний інтерфейс	– інколи повільне завантаження – фокус на одну команду

Розглянемо детальніше кожен з проєктів (рис. 1.1 – 1.4).

До найбільш вагомих плюсів даного вебсайту (див. рис. 1.1) можна віднести інтуїтивно зрозумілий інтерфейс, зручне меню, реалізований лайв-чат для зареєстрованих користувачів та наявність української мови. Значних мінусів мною не було знайдено.

Плюси вебсайту (див. рис. 1.2):

- добре реалізований користувацький інтерфейс;

- багато корисної інформації на головній сторінці;
- адаптивний інтерфейс;
- можливість реєстрації.

Мінуси вебсайту (рис. 1.2): відсутність української мови інтерфейсу.

Інших значних проблем не було виявлено.

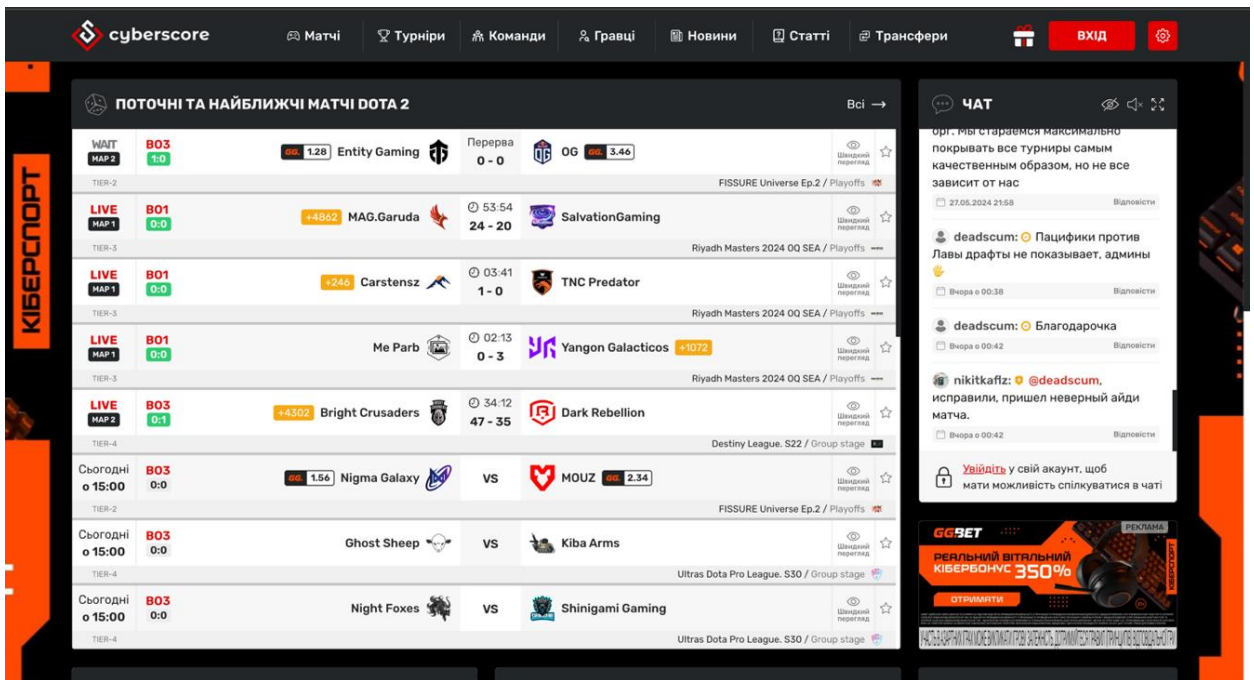


Рисунок 1.1 – Вигляд головної сторінки аналогічного вебсайту «Cyberscore»

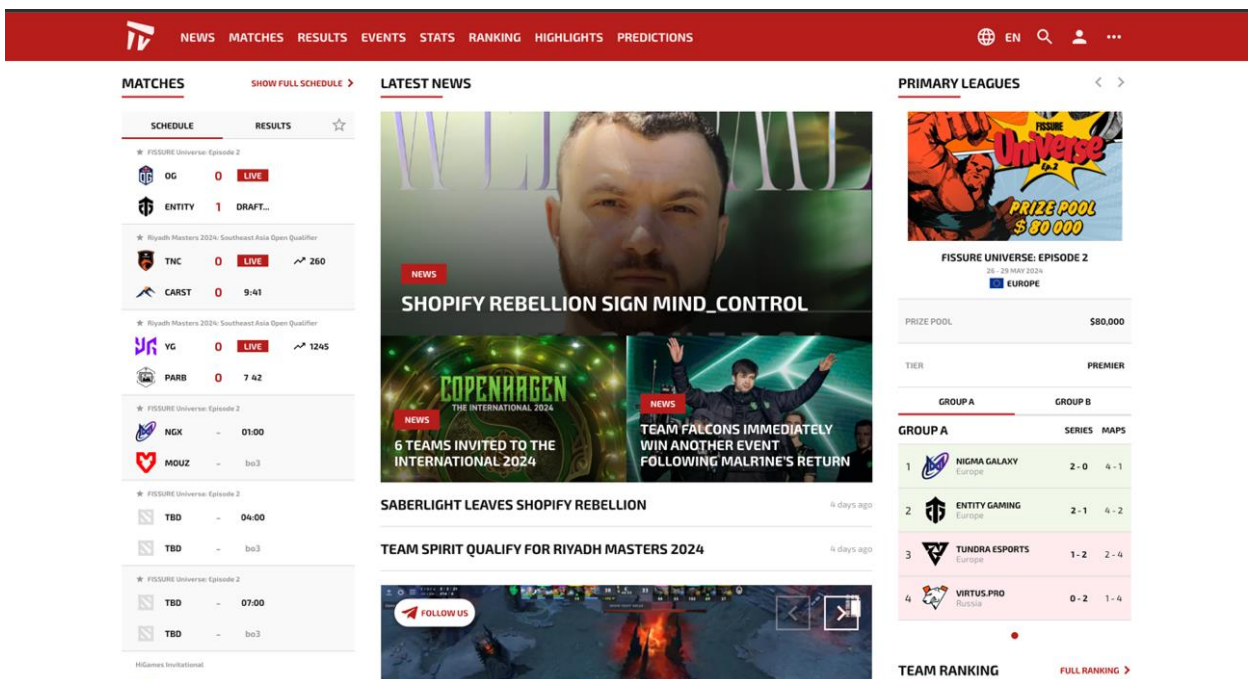


Рисунок 1.2 – Вигляд головної сторінки аналогічного вебсайту «dltv.org»

### Плюси вебсайту (рис. 1.3):

- актуальність інформації – сайт надає свіжі новини та оновлення про гру dota 2, що дозволяє користувачам бути в курсі останніх подій, патчів та турнірів;
- зручний інтерфейс – навігація на сайті проста та інтуїтивно зрозуміла, що робить його доступним для користувачів різного рівня досвіду;
- ком'юніті – наявність форумів та інших засобів комунікації дозволяє гравцям обмінюватися досвідом, задавати питання та ділитися стратегіями;
- різноманітний контент – сайт містить різноманітні матеріали, включаючи статті, гайди, відео та інші ресурси, що охоплюють різні аспекти гри;
- мобільна версія – сайт добре адаптований для мобільних пристроїв, що дозволяє користувачам зручно переглядати контент на ходу.

### Мінуси вебсайту (рис. 1.3):

- реклама – надмірна кількість реклами може відволікати користувачів та ускладнювати навігацію по сайту;
- відсутність унікального контенту – частина контенту може дублюватися з іншими сайтами, що робить його менш привабливим для користувачів, які шукають ексклюзивну інформацію;
- застарілий дизайн – дизайн сайту може виглядати застарілим у порівнянні з сучасними стандартами вебдизайну, що може вплинути на привабливість для нових користувачів;
- повільне завантаження – якщо сайт має велику кількість медіа-контенту, це може призвести до повільного завантаження сторінок, що негативно впливає на користувацький досвід;
- обмежена підтримка мов – сайт може мати обмежену кількість мовних версій, що ускладнює доступ для міжнародних користувачів, які не володіють основною мовою сайту.

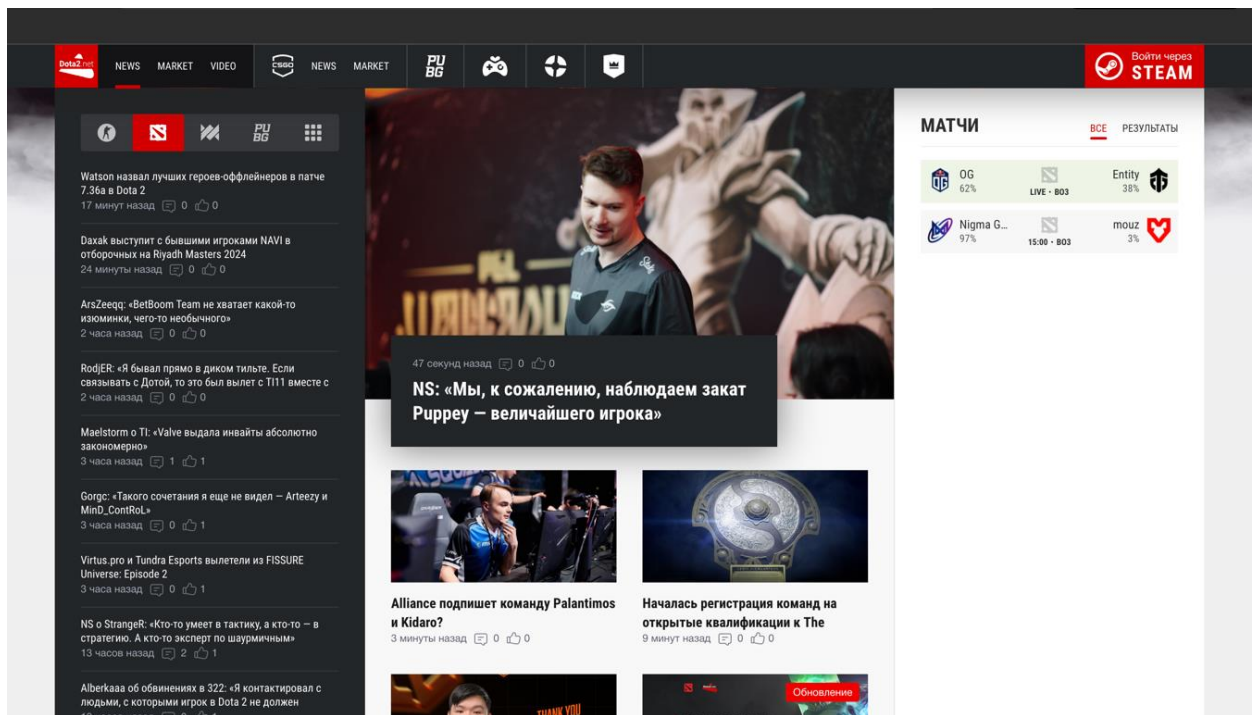


Рисунок 1.3 – Вигляд головної сторінки аналогічного вебсайту «dota.net»

Плюси сайту navi.gg (рис. 1.4):

- актуальність інформації – сайт регулярно оновлюється свіжими новинами та подіями про команду natus vincere та кіберспорт в цілому;
- зручний інтерфейс – сайт має просту та інтуїтивно зрозумілу навігацію, що дозволяє легко знаходити необхідну інформацію;
- різноманітний контент – крім новин, сайт пропонує відео, інтерв'ю, статті та інший контент, що цікавить прихильників кіберспорту;
- професійний дизайн – сайт виглядає сучасно і привабливо, що робить його зручним для користування;
- соціальна інтеграція – сайт добре інтегрований з соціальними мережами, що дозволяє користувачам легко ділитися контентом та стежити за новинами.

Мінуси сайту navi.gg (рис. 1.4):

- реклама – наявність реклами може відволікати користувачів і заважати комфортному перегляду контенту;
- обмежена мова – сайт може мати обмежену кількість мовних версій,



- що робить його менш доступним для міжнародної аудиторії;
- інколи повільне завантаження – через велику кількість медіа-контенту сторінки можуть завантажуватися повільніше;
  - відсутність деяких функцій – можливо, деякі користувачі очікують більше інтерактивних функцій, таких як форуми або чати;
  - фокус на одну команду – сайт зосереджений на одній команді, що може бути нецікавим для тих, хто хоче отримувати інформацію про інші команди та події в кіберспорті.

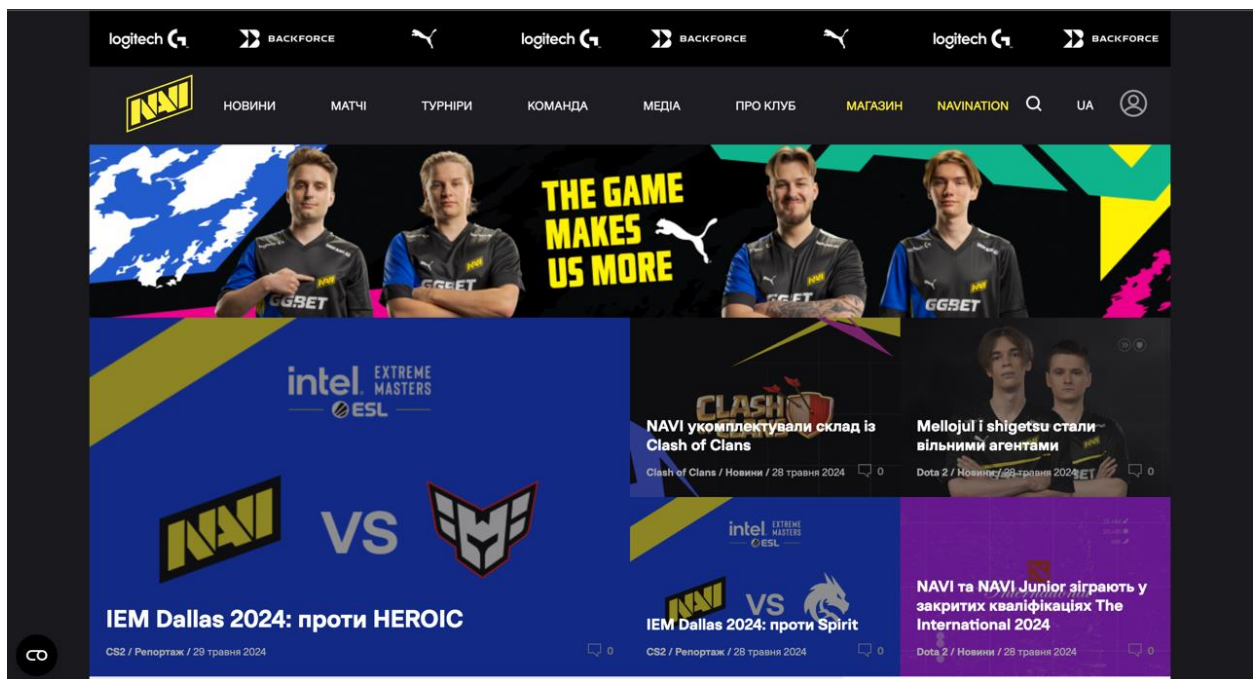


Рисунок 1.4 – Вигляд головної сторінки аналогічного вебсайту «navi.gg»

## 1.5 Формування вимог до майбутнього вебсайту

Порівняння схожих за темою вебсайтів, таких як dota.net та navi.gg, показує, що успішні інформаційні платформи для прихильників кіберспорту мають певні спільні риси та вимоги. Вони надають актуальну та своєчасну інформацію, пропонують зручний та інтуїтивно зрозумілий інтерфейс, містять різноманітний контент та інтегровані з соціальними мережами. Водночас,

надмірна кількість реклами та обмежена мовна підтримка можуть негативно впливати на користувацький досвід.

На основі цього аналізу вимоги до проєкту, що розроблюється, мають включати:

- актуальність і регулярне оновлення контенту;
- зручний інтерфейс;
- різноманітний контент;
- мінімізація реклами;
- підтримка кількох мов;
- швидкість завантаження;
- соціальна інтеграція.

Врахування цих вимог допоможе створити конкурентоспроможний та популярний вебсайт для прихильників кіберспорту.

Після аналізу усіх існуючих схожих проєктів, можна прийти до висновку, що усі сайти створені та підтримуються з метою надавати максимально нову, актуальну на сьогодні інформацію. Спільними ж проблемами при цьому є:

- відсутність на великій кількості вебсайтів повної підтримки української мови;
- необхідність втручання людини для оновлення даних;
- велика кількість реклами на сторінках.

Отже, головною метою мого майбутнього проєкту буде створити вебсайт, на якому усі дані будуть оновлюватись автоматично і який буде повністю пристосований для україномовного користувача.

## 2 ПРОЄКТУВАННЯ ВЕБСАЙТУ

### 2.1 Обґрунтування вибору засобів розробки

#### 2.1.1 JavaScript

JavaScript – це динамічна, об’єктно-орієнтована мова програмування, яка використовується для створення інтерактивних ефектів на вебсторінках. Вона була розроблена Бренданом Айхом у 1995 році для Netscape Navigator.

JavaScript дозволяє додавати такі функції до вебсайтів:

- інтерактивні елементи користувацького інтерфейсу (ui);
- динамічні зміни html та css;
- анімації та ефекти;
- обробка форм та валідація даних;
- створення ігор та мультимедійних додатків;
- робота з куки та локальним сховищем браузера.

Мова JavaScript є інтерпретованою, що означає, що код виконується безпосередньо в браузері без необхідності компіляції. Це робить процес розробки швидшим та інтерактивнішим.

*Основні концепції JavaScript.*

Щоб ефективно використовувати JavaScript, важливо зрозуміти його основні концепції:

- змінні: дозволяють зберігати та маніпулювати даними. оголошуються за допомогою let, const або var;
- функції: блоки коду, які виконують певні завдання та можуть приймати аргументи. оголошуються за допомогою function;
- об’єкти: колекції пов’язаних даних та функціональності. об’єкти можуть бути вбудованими (наприклад, math, date, array) або створеними користувачем;

- масиви: впорядковані колекції даних, доступ до яких здійснюється за індексом;
- умовні оператори: дозволяють виконувати різні дії залежно від умов, наприклад, `if...else`, `switch`;
- цикли: повторюють блок коду певну кількість разів, наприклад, `for`, `while`, `do...while`;
- події: дії, які відбуваються в браузері, наприклад, натискання кнопки миші, натискання клавіші, завантаження сторінки. обробляються за допомогою обробників подій.

Розуміння цих концепцій є ключовим для написання ефективного та функціонального JavaScript-коду.

Основні переваги використання JavaScript для веброзробки:

- клієнтська сторона: javascript дозволяє створювати інтерактивні та динамічні вебсторінки, покращуючи досвід користувача;
- крос-платформна сумісність: javascript підтримується практично всіма сучасними веббраузерами, забезпечуючи плавний досвід користувача;
- підтримка спільноти: javascript є однією з найпопулярніших мов програмування, тому існує величезна кількість ресурсів, інструментів, документації та підтримки спільноти;
- можливості клієнт-сервер: javascript може використовуватись на стороні сервера за допомогою `node.js`, що дозволяє створювати повноцінні вебдодатки;
- універсальність: javascript можна використовувати не тільки для веброзробки, а й для створення мобільних додатків, ігор, десктопних програм та навіть для керування роботами.

Щоб ефективно використовувати JavaScript для веброзробки, варто дотримуватись таких практик:

- підтримувати єдиний стиль коду при роботі в команді;
- використовувати належні стратегії тестування;

- стежити за останніми тенденціями та оновленнями мови;
- давати змістовні назви змінним, функціям та класам;
- розділяти логіку на модулі та компоненти;
- оптимізувати продуктивність, мінімізуючи кількість http-запитів та розмір файлів;
- використовувати сучасні фреймворки та бібліотеки для прискорення розробки.

Отже, JavaScript – це незамінний інструмент для створення сучасних, інтерактивних та привабливих вебсайтів. Його гнучкість, крос-платформність та підтримка спільноти роблять його одним з найпопулярніших засобів веброботи. Розуміння основ мови та дотримання найкращих практик дозволить створювати високоякісні вебдодатки [1].

### **2.1.2 React**

React.js – це популярний JavaScript-фреймворк, розроблений компанією Facebook для створення інтерактивних користувацьких інтерфейсів. Він використовує компонентний підхід до розробки, що дозволяє створювати багаторазові, модульні елементи інтерфейсу.

Основні переваги використання React.js:

- компонентна архітектура: react дозволяє розбивати інтерфейс на повторно використовувані компоненти, що спрощує розробку та підтримку;
- віртуальний dom: react використовує віртуальну модель документа, що дозволяє ефективно оновлювати лише змінені частини сторінки;
- декларативний підхід: react дозволяє описувати, як інтерфейс має виглядати в певному стані, а не вказувати, як його змінювати;
- велика екосистема: react має величезну спільноту розробників, що створюють бібліотеки, інструменти та плагіни;

- крос-платформність: react може використовуватись не лише для веброзробки, а й для створення мобільних додатків за допомогою react native.

Для ефективного використання React.js необхідно вивчити основи JavaScript, а також концепції, такі як компоненти, стан, ефекти, маршрутизація тощо. Існує багато навчальних ресурсів, які можуть допомогти в опануванні цього фреймворку [2].

### 2.1.3 Material UI

Material-UI – це популярна бібліотека компонентів для React, яка дозволяє швидко та ефективно створювати стильні та функціональні інтерфейси.

Переваги:

- готові компоненти: material-ui надає широкий набір готових компонентів, таких як кнопки, картки, діалогові вікна, іконки та інші, що спрощує розробку інтерфейсу та забезпечує консистентний дизайн;
- спрощена темізація: material-ui дозволяє легко налаштовувати тему та стилі компонентів, що дозволяє швидко адаптувати дизайн під бренд або вимоги проєкту;
- сумісність з react: material-ui розроблено спеціально для react, що забезпечує високу продуктивність та ефективність використання разом з цим фреймворком;
- широкий вибір функціональності: бібліотека містить різноманітні компоненти та можливості, що дозволяють створювати різноманітні та інтерактивні інтерфейси.

Недоліки:

- обмежена індивідуалізація: хоча material-ui надає гнучкість у темізації, іноді може бути складно досягти повної індивідуалізації

дизайну, особливо якщо потрібно відхилитися від стандартних material design принципів;

- розмір бібліотеки: завантаження всієї бібліотеки material-ui може призвести до збільшення розміру проєкту, що може вплинути на швидкодію та завантаження сторінки;
- необхідність додаткового навчання: для повного використання можливостей material-ui може знадобитися додатковий час на вивчення документації та особливостей бібліотеки.

*Доцільність використання.* Material-UI є відмінним вибором для розробки вебдодатків на React, особливо якщо вам потрібно швидко створити стильний та функціональний інтерфейс. Використання Material-UI дозволяє прискорити процес розробки, забезпечити консистентність дизайну та забезпечити готові рішення для багатьох типових компонентів. Однак, варто враховувати недоліки та обмеження бібліотеки при виборі її для проєкту [3].

#### **2.1.4 Axios**

Axios – це популярна бібліотека JavaScript для здійснення HTTP-запитів, яка була розроблена у 2016 році командою колишніх журналістів Politico – Джимом ВандеХаєм, Майком Алленом та Роєм Шварцем. Вони заснували компанію Axios, щоб створити новинний вебсайт, який би поєднував у собі стислість та ефективність інформаційних бюлетенів з глибиною та якістю матеріалів. Наведемо головні цілі використання.

*Спрощення клієнт-серверної комунікації:* Axios надає зручний інтерфейс для надсилання HTTP-запитів та обробки отриманих відповідей, що значно полегшує розробку вебдодатків [4].

*Підвищення продуктивності додатків:* Axios дозволяє оптимізувати роботу додатків за рахунок точного контролю над HTTP-запитами та відповідями, що призводить до швидшої та надійнішої роботи.

*Створення повторно використовуваного коду:* використовуючи Axios, розробники можуть створювати багаторазові блоки коду для взаємодії з API, що зменшує дублювання коду та покращує підтримку.

Переваги:

- кросплатформеність: axios працює як у браузерному, так і в node.js середовищі;
- розширена функціональність: axios має більший набір можливостей порівняно з деякими альтернативами, таких як перехоплювачі запитів та відповідей;
- зручний інтерфейс: axios надає зручний та інтуїтивно зрозумілий API для роботи з http-запитами;
- активна спільнота: axios має велику та активну спільноту розробників, що забезпечує широку підтримку та регулярні оновлення.

Недоліки:

- розмір: axios може бути дещо більшим за розміром порівняно з деякими мінімалістичними альтернативами;
- продуктивність: хоча axios є достатньо продуктивним, деякі альтернативи, такі як got, можуть бути ще швидшими;
- підсумовуючи, axios – це потужна та всебічна бібліотека для роботи з http-запитами, яка значно спрощує розробку вебдодатків та забезпечує високу продуктивність; її широка функціональність, зручний інтерфейс та активна спільнота роблять її популярним вибором серед розробників javascript.

### **2.1.5 Sass**

Sass – це мова стилів, яка була розроблена у 2006 році Хемптоном Кетлінгом. Вона була створена як розширення мови CSS, щоб надати



розробникам більше можливостей для створення та організації стилів.

Головні цілі використання Sass:

- підвищення ефективності написання css: sass надає розробникам додаткові можливості, такі як змінні, функції, вкладені правила та міксини, що дозволяє писати більш структурований та модульний CSS-код;
- покращення читабельності та підтримки: завдяки своїй синтаксичній структурі sass робить CSS-код більш зрозумілим та легшим для підтримки, особливо в великих проєктах;
- автоматизація рутинних завдань: sass дозволяє автоматизувати рутинні завдання, такі як генерація префіксів для різних браузерів, що заощаджує час розробників.

Переваги:

- розширена функціональність: sass надає розробникам потужні можливості, такі як змінні, функції, вкладені правила та міксини, що значно підвищує гнучкість та ефективність написання CSS;
- покращена читабельність: синтаксис sass робить CSS-код більш структурованим та зрозумілим, що полегшує його підтримку та розширення;
- кросбраузерна сумісність: sass може автоматично генерувати необхідні префікси для різних браузерів, що забезпечує кросбраузерну сумісність;
- широка підтримка: sass має велику та активну спільноту розробників, що забезпечує широку підтримку, регулярні оновлення та велику екосистему плагінів.

Недоліки:

- необхідність компіляції: оскільки sass є мовою, яка компілюється в CSS, це додає додатковий крок у процесі розробки;
- складність налаштування: для початку роботи з sass може знадобитися певна початкова конфігурація, що може бути складним

для початківців;

- додаткові залежності: використання sass вимагає додаткових залежностей, таких як компілятор sass, що може ускладнити налаштування проєкту.

Загалом, Sass є потужним інструментом, який значно підвищує ефективність та читабельність написання CSS-коду. Його розширена функціональність, кросбраузерна сумісність та широка підтримка роблять його популярним вибором серед розробників вебдодатків [5].

Також, у процесі розробки інформаційної системи для тих чи інших задач використовувались і інші бібліотеки, зображені на скріншоті нижче (рис. 2.1).

```
"dependencies": {
  "@emotion/react": "^11.11.4",
  "@emotion/styled": "^11.11.5",
  "@mui/material": "^5.15.15",
  "@testing-library/jest-dom": "^5.17.0",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "axios": "^1.6.8",
  "react": "^18.2.0",
  "react-dom": "^18.2.0",
  "react-router-dom": "^6.22.3",
  "react-scripts": "5.0.1",
  "sass": "^1.74.1",
  "web-vitals": "^2.1.4"
},
```

Рисунок 2.1 – Допоміжні бібліотеки, які використовувались у проєкті

### 2.1.6 API

Оскільки однією з ключових вимог до майбутнього додатку є актуальність даних, для оновлення великої частини інформації на вебсайті буде використана система API запитів. API-запити – це механізм, за

допомогою якого програми та додатки можуть обмінюватися даними та взаємодіяти один з одним. Ось основні цілі використання API-запитів:

- обмін даними між системами: api дозволяють одній системі запитувати та отримувати дані з іншої системи (наприклад, мобільний додаток може використовувати api для отримання даних з вебсервера);
- інтеграція функціональності: api дозволяють одній програмі використовувати функціональність іншої програми (наприклад, вебдодаток може використовувати api платіжної системи для обробки транзакцій);
- розширення можливостей додатків: api дозволяють розробникам додавати нові можливості до своїх додатків, використовуючи функціональність, надану іншими сервісами (наприклад, карти Google можуть бути вбудовані в вебсайт за допомогою API);
- автоматизація процесів: API дозволяють автоматизувати рутинні завдання, такі як оновлення даних або синхронізація між системами.

Загалом, API-запити є важливим механізмом для побудови інтегрованих, масштабованих та гнучких програмних систем, що взаємодіють між собою [6].

### **2.1.7 OpenDota**

Для розробки мого вебсайту я використовував сервіс API запитів “Opendota”.

Opendota – це безкоштовний сервіс аналітики даних для гри Dota 2, який надає детальну статистику та аналіз ігор для гравців та команд. Він використовує дані, отримані з API Dota 2, для надання всебічної аналітики, що дозволяє гравцям та командам покращувати свою гру та оптимізувати стратегії [7].

Основні можливості Opendota включають:

- детальний аналіз матчів, включаючи показники гравців, предмети, вбивства, смерті, асисті тощо;
- профілі гравців з їхньою статистикою, рейтингом, історією матчів та аналізом ефективності;
- аналіз команд, включаючи їхні найуспішніші герої, предмети, комбінації героїв тощо;
- детальну статистику по кожному герою, включаючи показники ефективності, популярність, рейтинг тощо;
- аналіз популярності та ефективності предметів, включаючи їхні комбінації та статистику по героях;
- аналіз ключових ігрових ситуацій, таких як першокров, руйнування башт, захоплення рошана тощо.

Opendota надає гравцям та командам потужний інструмент для аналізу своєї гри та пошуку шляхів для вдосконалення. Завдяки детальній статистиці та аналітиці, сервіс допомагає гравцям виявляти свої сильні та слабкі сторони, а командам – розробляти ефективні стратегії та склади.

### **2.1.8 Openproj**

OpenProj – безкоштовна програма з відкритим кодом для управління проектами, яка є альтернативою Microsoft Project. Її розробила компанія Projity у 2007 році. Цей застосунок забезпечує функції для планування, управління ресурсами та контролю за виконанням проєктів.

OpenProj дозволяє користувачам створювати та управляти завданнями, встановлювати їх залежності та визначати тривалість виконання. Програма підтримує створення графіків Gantt, які візуально відображають часові рамки проєкту. Також є можливість призначати ресурси на завдання, відслідковувати

їх використання та уникати перевантаження. Користувачі можуть налаштувати календарі для різних проєктів, включаючи робочі дні, вихідні та свята. Програма пропонує різноманітні звіти для аналізу прогресу проєкту, таких як звіти про завдання, ресурси та витрати.

Однією з головних переваг OpenProj є його безкоштовність, що робить його доступним для малого та середнього бізнесу. Програма сумісна з файлами Microsoft Project, що дозволяє легко імпортувати та експортувати дані. Вона працює на різних операційних системах, включаючи Windows, Mac OS та Linux.

Деякі користувачі зазначають, що інтерфейс OpenProj менш інтуїтивний порівняно з комерційними аналогами. Також, оскільки програма розробляється спільнотою, підтримка та регулярні оновлення можуть бути обмеженими.

OpenProj часто використовується малими компаніями для планування та управління внутрішніми проєктами. Викладачі та студенти застосовують програму для навчання управлінню проєктами. Неприбуткові організації, завдяки безкоштовній ліцензії, використовують OpenProj для координації своїх проєктів з обмеженими бюджетами.

OpenProj є потужним та доступним інструментом для управління проєктами, який пропонує необхідні функції для планування та контролю виконання завдань. Незважаючи на деякі обмеження, програма популярна серед користувачів, які шукають безкоштовну альтернативу комерційним продуктам.

## **2.2 План реалізації**

Для початку роботи над проєктом треба побудувати його план реалізації, що я і зробив за допомогою додатку OpenProj. Там я розбив процес розробки на етапи, приблизно розрахував час для їх завершення (рис. 2.2).

	📌	Название	Продолжи...	Начало	Окончание
1		☐ Інформаційний вебресурс	53 днів	15.02.24 8:00	29.04.24 17:00
2		☐ Аналіз вимог	5 днів	15.02.24 8:00	21.02.24 17:00
3		Визначення функціоналу	2 днів	15.02.24 8:00	16.02.24 17:00
4		Визначення основних функцій	3 днів	19.02.24 8:00	21.02.24 17:00
5		☐ Розробка серверної частини	31 днів	22.02.24 8:00	04.04.24 17:00
6		Налаштування сервісів	2 днів	22.02.24 8:00	23.02.24 17:00
7		Створення основної структури	5 днів	26.02.24 8:00	01.03.24 17:00
8		Реалізація API для взаємодії	10 днів	04.03.24 8:00	15.03.24 17:00
9		Реалізація бізнес-логіки	8 днів	18.03.24 8:00	27.03.24 17:00
10		Обробка помилок та валідація	6 днів	28.03.24 8:00	04.04.24 17:00
11		☐ Розробка клієнтської частини	30 днів	22.02.24 8:00	03.04.24 17:00
12		Налаштування сервісів	2 днів	22.02.24 8:00	23.02.24 17:00
13		Створення компонентів	12 днів	26.02.24 8:00	12.03.24 17:00
14		Реалізація логіки взаємодії	10 днів	13.03.24 8:00	26.03.24 17:00
15		Валідація та обробка помилок	6 днів	27.03.24 8:00	03.04.24 17:00
16		☐ Інтеграція клієнтської частини	18 днів	04.04.24 8:00	29.04.24 17:00
17		Перевірка взаємодії між частинами	3 днів	04.04.24 8:00	08.04.24 17:00
18		Виправлення помилок	5 днів	09.04.24 8:00	15.04.24 17:00
19		Оптимізація швидкості виконання	4 днів	16.04.24 8:00	19.04.24 17:00
20		Фінальне тестування та налаштування	6 днів	22.04.24 8:00	29.04.24 17:00
21		Реліз	2 днів	30.04.24 8:00	01.05.24 17:00

Рисунок 2.2 – План реалізації проекту у OpenProj

Основними етапами стали:

- аналіз вимог;
- розробка серверної частини;
- розробка клієнтської частини;
- інтеграція клієнтської частини.

### 2.3 Діаграма прецедентів

UML (Universal Modeling Language) – це універсальна мова моделювання, створена компанією Rational Software для оптимального і універсального опису предметної області та конкретних завдань у програмуванні. Будь-яке завдання моделюється за допомогою набору ієрархічних діаграм, кожна з яких представляє певну проекцію системи.

Діаграми прецедентів використовуються для моделювання вигляду системи з точки зору зовнішнього спостерігача. На діаграмі прецедентів графічно показана сукупність прецедентів та суб'єктів, а також відносини між ними.

Наведемо основні елементи діаграми прецедентів.

Суб'єкт (actor) – це будь-яка сутність, що взаємодіє із системою зовні або набір логічно пов'язаних ролей, виконуваних при взаємодії з прецедентами. Стандартним графічним позначенням суб'єкта на діаграмах є фігурка «чоловічка», під якою записується конкретне ім'я суб'єкта. Суб'єктом може бути не тільки людина, але й технічний пристрій, програма або будь-яка інша система, яка може впливати на модельовану систему так, як визначить розробник.

Прецеденти (use case) – це опис безлічі послідовностей дій (включаючи їх варіанти), які виконуються системою, щоб суб'єкт отримав результат, що має для нього певне значення. При цьому не уточнюється, яким чином буде реалізована взаємодія суб'єктів із системою. Це одна з найважливіших особливостей розробки прецедентів. Стандартним графічним позначенням прецеденту на діаграмах є еліпс, всередині якого знаходиться коротка назва прецеденту або ім'я у формі дієслова з пояснювальними словами.

Мною була побудована подібна діаграма для мого вебсайту. Акторами у ній виступають користувач вебсайту та адміністратор, а прецедентами усі можливі способи використання інформаційної системи. Побудована діаграма зображена на (рис. 2.3).

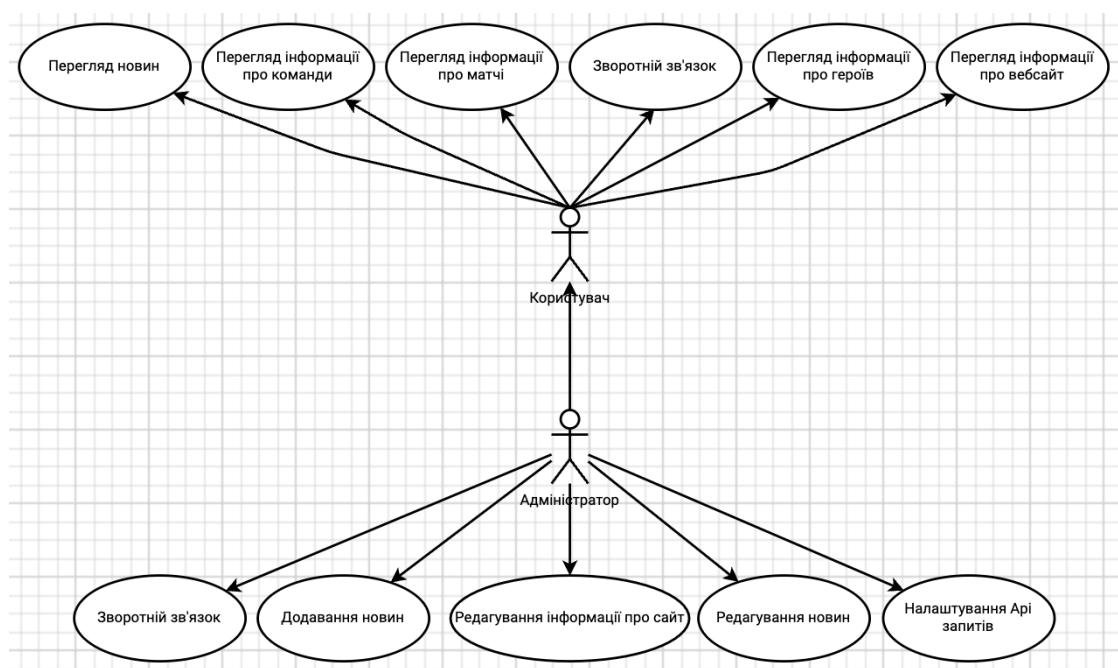


Рисунок 2.3 – Діаграма прецедентів

Діаграма прецедентів демонструє, що для сайта, що проектується, передбачено 2 види користувачів: користувач сайту і адміністратор сайту.

## 2.4 Макет вебсайту

Дизайн-макет сайту – це візуальне уявлення майбутнього вебсайту, створене з урахуванням теоретичних технічних можливостей, яке приблизно відтворює його зовнішній вигляд та сторінки. Макет представлений у вигляді зображення, яке буде відображатися у веббраузері (рис. 2.4).



Рисунок 2.4 – Макет головної сторінки

Після того як я визначився з головними вимогами до майбутнього вебсайту, порівняв аналоги майбутньої інформаційної системи, розглянув їх переваги та недоліки та розробив план реалізації з діаграмою прецедентів та макетом основної сторінки сайту я зміг приступити безпосередньо до програмної реалізації проєкту.



## 3 РЕАЛІЗАЦІЯ ПРОЄКТУ ВЕБСАЙТУ

### 3.1 Налаштування програмного середовища

Для роботи над проєктом було встановлено WebStorm, а також VS Code для додаткового тесту розробленої системи.

WebStorm – це потужне інтегроване середовище розробки (IDE), створене компанією JetBrains для професійної розробки вебдодатків. Програма підтримує такі мови програмування як JavaScript, TypeScript, HTML, CSS та багато інших. Завдяки розширеним можливостям, таким як автоматичне завершення коду, рефакторинг, навігація по проєкту та інтеграція з системами контролю версій, WebStorm значно полегшує процес розробки.

Visual Studio Code (VS Code) – це легке, але потужне інтегроване середовище розробки (IDE), розроблене компанією Microsoft. Воно підтримує безліч мов програмування, включаючи JavaScript, Python, Java, C++ та багато інших. VS Code пропонує розширені функції редагування коду, такі як автозавершення, підсвічування синтаксису та вбудована підтримка відлагодження.

Після встановлення та створення проєкту у WebStorm було підключено допоміжні бібліотеки, такі як react, styled, mui/material, react, axios, sass. Приклад коду (рис. 3.1).

```
"dependencies": {
  "@emotion/react": "^11.11.4",
  "@emotion/styled": "^11.11.5",
  "@mui/material": "^5.15.15",
  "@testing-library/jest-dom": "^5.17.0",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "axios": "^1.6.8",
  "react": "^18.2.0",
  "react-dom": "^18.2.0",
  "react-router-dom": "^6.22.3",
  "react-scripts": "^3.0.1",
  "sass": "^1.74.1",
  "web-vitals": "^2.1.4"
},
```

Рисунок 3.1 – Код для виклику допоміжних бібліотек

Також, були прописані стилі для головної та інших сторінок за допомогою Sass коду, за своєю суттю схожому на CSS (рис. 3.2).

```
body {
  margin: 0;
  font-family: 'Roboto', sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

.container {
  background: url('../assets/bg.jpeg') center center no-repeat;
  background-size: cover;
  width: 100vw;
  height: 100vh;
  padding: 100px 100px 20px;
}
```

Рисунок 3.2 – Код для задання стилів сторінок

### 3.2 Router.js

Після цього підключимо Router.js – це JavaScript бібліотека, призначена для управління маршрутизацією в односторінкових вебдодатках (SPA). Вона допомагає розробникам керувати переходами між різними частинами додатку без необхідності перезавантаження сторінки. Основне завдання Router.js – відповідати за зміну URL та відображення відповідного контенту на основі змін цього URL.

Розглянемо призначення Router.js.

*Управління навігацією:* Router.js дозволяє розробникам визначати маршрути та асоціювати їх з певними компонентами або сторінками. Це забезпечує плавні переходи між різними частинами додатку.

*Динамічна зміна контенту:* бібліотека забезпечує динамічне завантаження та відображення контенту на основі зміни URL, що дозволяє уникнути повного перезавантаження сторінки.

*Підтримка історії браузера:* Router.js інтегрується з історією браузера, дозволяючи користувачам використовувати кнопки «Назад» та «Вперед» для навігації по додатку.

*Сео-оптимізація:* з правильними налаштуваннями, маршрутизатор може допомогти у створенні SEO-дружніх URL, що покращує індексацію додатку пошуковими системами.

Розглянемо, як працює Router.js.

*Визначення маршрутів:* спочатку розробник визначає маршрути у конфігураційному файлі або безпосередньо у коді додатку. Кожен маршрут зазвичай асоціюється з певним компонентом або функцією, яка виконується при активації маршруту.

*Перехоплення змін URL:* Router.js слухає зміни URL у браузері, які можуть відбуватися при кліках на посилання або через програмні зміни. Бібліотека обробляє ці зміни та визначає, який маршрут відповідає новому URL.

*Виконання відповідних дій:* коли відбувається зміна маршруту, Router.js виконує відповідну функцію або відображає компонент, пов'язаний з новим URL. Це дозволяє динамічно оновлювати контент без повного перезавантаження сторінки.

*Оновлення історії:* Router.js інтегрується з історією браузера, додаючи нові записи при зміні маршруту. Це дозволяє користувачам використовувати кнопки «Назад» і «Вперед» для навігації по додатку.

*Хендлери подій:* бібліотека може також використовувати хендлери подій для управління різними аспектами навігації, такими як передзавантаження даних або анімація переходів.

Отже, Router.js – це важливий інструмент для управління маршрутизацією в SPA-додатках. Він забезпечує плавні переходи між сторінками, покращуючи користувацький досвід та полегшуючи розробку динамічних вебдодатків [8].

Ось приклад коду з використанням Router.js у моєму проєкті (рис. 3.3).

```

import {Route, Routes} from "react-router-dom";
import Teams from "./pages/Teams";
import React from "react";
import Container from "./components/Container";
import Team from "./pages/Team";
import NavigateToDefault from "./components/NavigateToDefault";

const Router = () => {
  return (
    <Container>
      <Routes>
        <Route path="/teams" element={<Teams/>}/>
        <Route path="/teams/:teamId/:screen?" element={<Team/>}/>
        <Route path="/*" element={<NavigateToDefault/>}/>
      </Routes>
    </Container>;
  );
};

```

Рисунок 3.3 – Код з використанням router.js

У реалізованому проєкті, також прописана функція: “<Route path="/\*" element={<NavigateToDefault/>}/>”, яка при будь якому непередбачуваному шляху у посиланні буде пересилати користувача на головну сторінку.

Тепер, коли ми створили сторінки у яких будуть відображатись дані, та реалізували механізм навігації між ними, можна створити систему API запитів.

### 3.3 Система API запитів

Як вже було зазначено раніше, для побудови системи API запитів у майбутньому буде використано сервіс OpenDota.

Такий вибір пов’язаний з тим, що подібних ресурсів на обрану тематику, нажаль не так багато, а цей є відкритим, безкоштовним та найбільш стабільним.

На сайті сервісу ми можемо побачити інструкції з інтеграції запитів на ваш вебсайт (рис. 3.4).

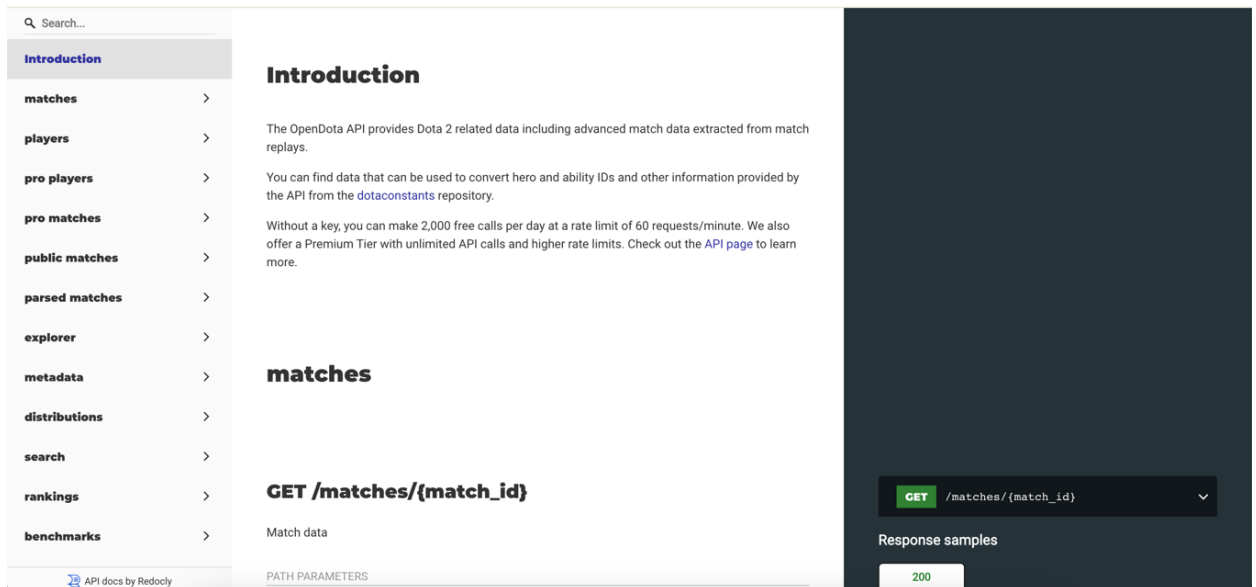


Рисунок 3.4 – Зовнішній вигляд сервісу OpenDota API

На прикладі розділу: «Teams», розглянемо вигляд подібних запитів та реалізуємо їх у проєкті (рис. 3.5).

Responses	
<div style="border: 1px solid green; padding: 5px; display: inline-block;"> <span style="color: green;">✓ 200 Success</span> </div>	
RESPONSE SCHEMA: <code>application/json; charset=utf-8</code>	
team_id	integer Team's identifier
rating	number The Elo rating of the team
wins	integer The number of games won by this team
losses	integer The number of losses by this team
last_match_time	integer The Unix timestamp of the last match played by this team
name	string or null Team name
tag	string The team tag/abbreviation

Рисунок 3.5 – Приклад відповіді на запит від сервісу

Отже, при подібному запиті, сервіс передасть нам наступні дані відносно команд: ID, рейтинг команди в очках, кількість перемог, кількість поразок, дата останнього зіграного матчу, назва та тег. Ось так це виглядає у коді: (рис. 3.6).

```
"team_id": 0,
"rating": 0,
"wins": 0,
"losses": 0,
"last_match_time": 0,
"name": "Newbee",
"tag": "string"
```

Рисунок 3.6 – Код запиту

Сам запит подається через посилання, яке виглядає наступним чином (рис. 3.7).

```
{
  "team_id": 9247354, "rating": 1674.51, "wins": 188, "losses": 60, "last_match_time": 1716740280, "name": "Team Falcons", "tag": "FLCN", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/231435057181870059/2B5CF9E9BA0A2DC303A13261444532AA08352843/",
  "team_id": 7119388, "rating": 1588.45, "wins": 538, "losses": 381, "last_match_time": 1716140682, "name": "Team Spirit", "tag": "TSPirit", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/1639179120711951766/CD7E8085C8527334205CC7885E9C18187BC17702/",
  "team_id": 8599101, "rating": 1559.33, "wins": 378, "losses": 243, "last_match_time": 1716740280, "name": "Gaimin Gladiators", "tag": "GG", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/1850419664501191993/5DAAB68FB5664029E1792A0F35E7483FE3FA026/",
  "team_id": 8261500, "rating": 1536.74, "wins": 293, "losses": 264, "last_match_time": 1716661500, "name": "Xtreme Gaming", "tag": "XG", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/2023835418754383891/3449F2F94858E332CE6FAD2C1160E3657EF33ACD/",
  "team_id": 8605863, "rating": 1532.57, "wins": 284, "losses": 220, "last_match_time": 1717448343, "name": "Entity", "tag": "Entity", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/20790295695924942/E3DF187AB08AC034E1027B99C843FF6E279BE8A/",
  "team_id": 7412785, "rating": 1520.12, "wins": 267, "losses": 1, "last_match_time": 1639928575, "name": "CyberBonch-1", "tag": "CB", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/1842537871043985153/774A18388EB3E73BFDEFCE0EDFD97B4F5C628B38/",
  "team_id": 8255888, "rating": 1501.49, "wins": 326, "losses": 281, "last_match_time": 1716725940, "name": "BetBoom Team", "tag": "BetBoom", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/2037351993974889309/0C315FDC88EA88A48EC3C879D946A4139CC9E/",
  "team_id": 9255706, "rating": 1482.93, "wins": 80, "losses": 64, "last_match_time": 1717247058, "name": "Aurora 1xBet", "tag": "Aurora", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/2362769341411270166/E13C592A0E744E1C386E90DE650BE36B85AE8137/",
  "team_id": 5, "rating": 1470.06, "wins": 1446, "losses": 1081, "last_match_time": 1717333829, "name": "G2 x iG", "tag": "iG", "logo_url": "https://steamcdn-a.akamaihd.net/apps/dota2/images/team_logos/5.png",
  "team_id": 2586976, "rating": 1456.75, "wins": 1007, "losses": 686, "last_match_time": 1717096655, "name": "OG", "tag": "OG", "logo_url": "https://steamcdn-a.akamaihd.net/apps/dota2/images/team_logos/2586976.png",
  "team_id": 5228054, "rating": 1450.04, "wins": 79, "losses": 36, "last_match_time": 1535061995, "name": "VGJ Storm", "tag": "VGJ.Storm", "logo_url": "https://steamcdn-a.akamaihd.net/apps/dota2/images/team_logos/5228054.png",
  "team_id": 5026801, "rating": 1429.8, "wins": 176, "losses": 116, "last_match_time": 1535053843, "name": "", "tag": "", "logo_url": "https://steamcdn-a.akamaihd.net/apps/dota2/images/team_logos/5026801.png",
  "team_id": 15, "rating": 1418.02, "wins": 1726, "losses": 1057, "last_match_time": 1717420451, "name": "LGD Gaming", "tag": "LGD", "logo_url": "https://steamcdn-a.akamaihd.net/apps/dota2/images/team_logos/15.png",
  "team_id": 2163, "rating": 1408.09, "wins": 1404, "losses": 950, "last_match_time": 1716401371, "name": "Team Liquid", "tag": "Liquid", "logo_url": "https://steamcdn-a.akamaihd.net/apps/dota2/images/team_logos/2163.png",
  "team_id": 8169775, "rating": 1407.57, "wins": 448, "losses": 368, "last_match_time": 1707305498, "name": "Geek Fam", "tag": "Geek", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/204750286753513272/C72214A86E6C450DABC1A18AE9539B5CAEE7D45/",
  "team_id": 8121295, "rating": 1403.61, "wins": 37, "losses": 9, "last_match_time": 1603906271, "name": "mudgolems", "tag": "", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/1635325934676609507/8E073AB63209CE73B08E05681C6CCE101483890/",
  "team_id": 8260983, "rating": 1402.96, "wins": 291, "losses": 206, "last_match_time": 1700362218, "name": "TSM", "tag": "TSM", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/1996813106806561034/BC39F00C11E1DC7D708A90CE493384A8B0966004/",
  "team_id": 8291895, "rating": 1402.68, "wins": 393, "losses": 256, "last_match_time": 1716901227, "name": "Tundra Esports", "tag": "Tundra", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/2031716132171967904/07B1688806309B22CDAD53AB421ECAF3D4B2E07E/",
  "team_id": 3214108, "rating": 1401.12, "wins": 136, "losses": 90, "last_match_time": 1498631388, "name": "Team NP", "tag": "NP", "logo_url": "https://steamcdn-a.akamaihd.net/apps/dota2/images/team_logos/3214108.png",
  "team_id": 8724984, "rating": 1394.56, "wins": 209, "losses": 161, "last_match_time": 1717342906, "name": "Virtus.pro", "tag": "VP", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/1796396122328355023/2E83383744EC893AD9FF3797C030984AD954AD2E/",
  "team_id": 9303484, "rating": 1389.91, "wins": 72, "losses": 64, "last_match_time": 1717467407, "name": "Heroic", "tag": "Heroic", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/2471984170520125054/B066431AF4D322D300DD5180CE8F6BAE85A7F5/",
  "team_id": 2519319, "rating": 1389, "wins": 45, "losses": 16, "last_match_time": 1445985009, "name": "(monkey) Business", "tag": "(monkey)", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/383162071968605685/B670804CDB13F184D122ACBB8F75E2D82C959CAF/",
  "team_id": 8894818, "rating": 1383.65, "wins": 206, "losses": 136, "last_match_time": 1717518512, "name": "PSG.Lost", "tag": "PSG.Lost", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/2071150481757595896/D4608C040893FB7F584A8F89A041EDCFD20380A9/",
  "team_id": 7819701, "rating": 1381.67, "wins": 76, "losses": 40, "last_match_time": 1601139882, "name": "VP.Prodigy", "tag": "VP.P", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/10093106397423917/9175453DE6C700E0CC6D547F43784819F1144A1A/",
  "team_id": 9142866, "rating": 1380.6, "wins": 133, "losses": 25, "last_match_time": 1693586046, "name": "Business Club", "tag": "bs", "logo_url": "https://steamusercontent-a.akamaihd.net/ugc/10093106397423917/9175453DE6C700E0CC6D547F43784819F1144A1A/"
}
```

Рисунок 3.7 – Інформація, яку ми отримуємо від сервісу

Якщо структурувати ці дані, то у фінальному вигляді ми отримаємо наступний код у вигляді відповіді (рис. 3.8).

```
[
  {
    "team_id": 9247354,
    "rating": 1674.51,
    "wins": 188,
    "losses": 60,
    "last_match_time": 1716740280,
    "name": "Team Falcons",
    "tag": "FLCN",
    "logo_url": "https://steamusercontent-
a.akamaihd.net/ugc/2314350571781870059/2B5C9FE9BA0A2DC303A13261444532AA08352843/"
  },
  {
    "team_id": 7119388,
    "rating": 1588.45,
    "wins": 538,
    "losses": 381,
    "last_match_time": 1716140682,
    "name": "Team Spirit",
    "tag": "TSpirit",
    "logo_url": "https://steamusercontent-
a.akamaihd.net/ugc/1839179120711951766/CD7E0885CB527334205CC7885E9C101B7BC17702/"
  },
  {
    "team_id": 8599101,
    "rating": 1550.33,
    "wins": 378,
    "losses": 243,
    "last_match_time": 1716740280,
    "name": "Gaimin Gladiators",
    "tag": "GG",
    "logo_url": "https://steamusercontent-
a.akamaihd.net/ugc/1850419664501191993/5DAAB68FB5604D29E1792A0F35E74B3FE3F3A026/"
  },
]
```

Рисунок 3.8 – Вигляд структурованої відповіді на запит

У наведеному коді є усі вище описані дані для команд: «Team Falcons», «TSpirit», «Gaimin Gladiators». Подібним чином, на наш вебсайт передаються дані для 100 команд у порядку спадання їх світвого рейтингу (рис. 3.9).

РАНГ	НАЗВА	РЕЙТИНГ	ПЕРЕМОГИ	ПОРАЗКИ
1	Team Falcons	1674.51	188	60
2	Team Spirit	1588.45	538	381
3	Gaimin Gladiators	1550.33	378	243
4	Xtreme Gaming	1536.74	293	264
5	Entity	1532.57	284	220
6	CyberBench-1	1520.12	267	1

Рисунок 3.9 – Приклад виведених на розроблюваний вебсайт даних, отриманих в результаті API запиту

У наступній таблиці (рис. 3.10) також ми можемо перейти на сторінку кожної окремої команди, де ми побачимо всю історію їх останніх матчів: у рамках якого чемпіонату або ліги проходила гра, скільки вона тривала, хто був суперником та з яким результатом завершилась зустріч для обраної команди. А також, загальну інформацію: назва та логотип команди, кількість виграних та програних ігр за всю історію та загальний рейтинг.

МАТЧІ		ГЕРОЇ		ГРАВЦІ	
ЛІГА	ТРИВАЛІСТЬ	РЕЗУЛЬТАТ	СУПЕРНИК		
Riyadh Masters 2024 at Esports World Cup Qualifiers	55m 31s	Lost Match		Virtus.pro	
Riyadh Masters 2024 at Esports World Cup Qualifiers	54m 55s	Won Match		Virtus.pro	
Riyadh Masters 2024 at Esports World Cup Qualifiers	50m 41s	Lost Match		Virtus.pro	

Рисунок 3.10 – Сторінка команди «Natus Vincere»



Також, ми можемо побачити кнопку «Обрати іншу команду», що за допомогою Router.js поверне нас до загального списку усіх команд.

Ось так представлена таблиця результатів виглядає у кодї (рис. 3.11 – 3.12).

```
import * as React from 'react';
import Table from '@mui/material/Table';
import TableCell from '@mui/material/TableCell';
import TableContainer from '@mui/material/TableContainer';
import TableHead from '@mui/material/TableHead';
import TableRow from '@mui/material/TableRow';
import TeamItem from './TeamItem';
const columns = [
  {id: 'RANK', label: 'РАНГ', minWidth: 170},
  {id: 'NAME', label: 'НАЗВА', minWidth: 100},
  {
    id: 'RATING',
```

Рисунок 3.11 – Код для таблиці на сторінці команди

```
    label: 'РЕЙТИНГ',
    minWidth: 170,
  },
  {
    id: 'WINS',
    label: 'ПЕРЕМОГИ',
    minWidth: 170,
  },
  {
    id: 'LOSSES',
    label: 'ПОРАЗКИ',
    minWidth: 170,
  },
];
const TeamsTable = ({teams}) => {
```

Рисунок 3.12 – Код для таблиці на сторінці команди

Також, за допомогою аналогічних запитів отримаємо інформацію щодо гравців, що грали хоч раз за обрану команду та ігрових персонажів, яких найчастіше використовують команди (рис. 3.13).

The screenshot shows a REST client interface with the following details:

- Endpoint:** GET `/players/{account_id}`
- Path Parameters:** `account_id` (integer, required, Steam32 account ID)
- Response:** 200 Success
- Response Schema:** `application/json; charset=utf-8`
- Response Structure:**
  - `solo_competitive_rank`: integer or null (solo\_competitive\_rank)
  - `competitive_rank`: integer or null (competitive\_rank)
  - `rank_tier`: number or null (rank\_tier)
  - `leaderboard_rank`: number or null (leaderboard\_rank)
  - `profile`: object (profile)
- Response Content:**

```

{
  "solo_competitive_rank": 0,
  "competitive_rank": 0,
  "rank_tier": 0,
  "leaderboard_rank": 0,
  "profile": {
    "account_id": 0,
    "personaname": "420 booty",
    "name": "string",
    "plus": true,
    "cheese": 0,
    "steamid": "string",
    "avatar": "string",
    "avatarmedium": "string",
    "avatarfull": "string",
    "profileurl": "string",
    "last_login": "string",
    "loccountrycode": "string",
    "is_contributor": false,
    "is_subscriber": false
  }
}

```

Рисунок 3.13 – Формат даних для запиту “Players”

Для мого вебсайту, було реалізовано систему запитів у форматі представленому на (рис. 3.14).

```

import axios from "axios"

export const getPlayers = async (team_id) => {
  const { data } = await axios.get(`https://api.opendota.com/api/teams/${team_id}/players`);

  return data.slice(0, 50);
}

```

Рисунок 3.14 – Приклад запиту для розроблюваного вебресурсу

За допомогою наступного коду ми виконуємо API запит для категорії «Гравці», яку потім виводимо у таблицю.

Аналогічно, було прописано запити для усіх інших категорій, та сформовано відповідну папку з файлами у проєкті (рис. 3.15).

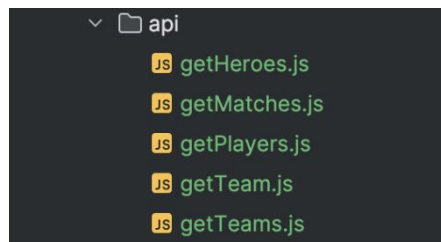


Рисунок 3.15 – Файли з усіма API запитами для вебсайту

### 3.4 Меню та інші сторінки

Також, для сайту було розроблено сторінки «Новини», «Про сайт» та меню для переходу між ними (рис. 3.16 – 3.18).

RANK	NAME	RATING	WINS	LOSSES
1	Team Falcons	1674.51	188	60
2	Team Spirit	1588.45	538	381
3	Gaimin Gladiators	1550.33	378	243
4	Xtreme Gaming	1536.74	293	264
5	Entity	1532.57	284	220
6	CyberBorch-1	1520.12	267	1

Рисунок 3.16 – Головна сторінка сайту

**News**

**Віцепрезидент BLAST: «Було б неймовірною честю бути долученими до The International одного дня»**  
9 травня, 19:42

---

**NAVI Junior – чемпіони European Pro League Season 17 – вони програли лише 2 карти за турнір**  
22 квітня, 09:32

Рисунок 3.17 – Пункт меню «Новини»

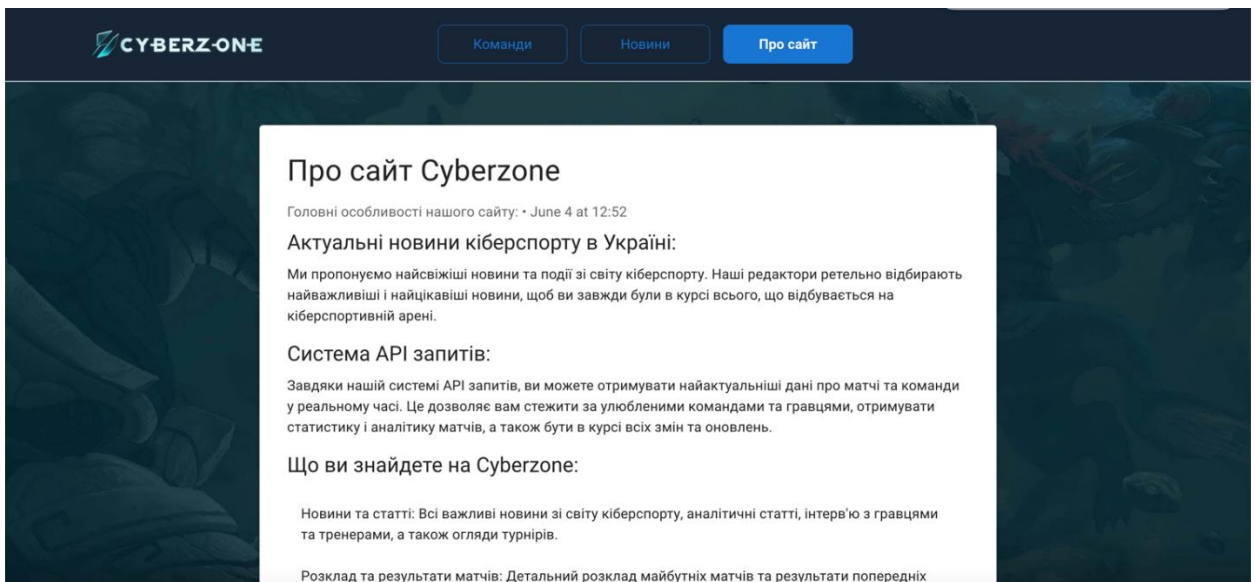


Рисунок 3.18 – Пункт меню «Про сайт»

Приклад коду сторінки «Про сайт» (рис. 3.19). Приклад коду сторінки «Новини» (рис. 3.20).

```
import React from 'react';
import { Container, List, ListItem, ListItemText, Typography } from "@mui/material";
import Paper from "@mui/material/Paper";
import Box from "@mui/material/Box";

const About = () => {
  return (
    <Container maxWidth="md">
      <Paper elevation={3} sx={{ p: 4 }}>
        <Typography variant="h4" component="h1" gutterBottom>
          Про сайт Cyberzone
        </Typography>
      </Paper>
    </Container>
  );
}
```

Рисунок 3.19 – Код для сторінки «Про сайт»

```
import React from 'react';
import { Link } from 'react-router-dom';
import { Stack } from "@mui/material";
const news = [
  {
    id: 1,
    title: "Віцепрезидент BLAST: «Було б неймовірною честю бути долученими до The International одного дня»",
  }
]
```

Рисунок 3.20 – Код для сторінки «Новини»

## ВИСНОВКИ

Під час написання дипломної роботи я розробив функціональний вебресурс з кіберспорту, використовуючи OpenDota API та мови програмування HTML, CSS, JavaScript. Аналіз аналогічних сайтів, розробка макетів сторінок та UML діаграм, а також реалізація проєкту підтвердили, що успішний вебресурс має бути не лише багатофункціональним та візуально привабливим, але й зручним для користувача.

Основний акцент був зроблений на інтеграцію OpenDota API, що дозволяє отримувати актуальні дані про команди та матчі. Це забезпечило користувачів найсвіжішою інформацією та дозволило відслідковувати останні події в світі кіберспорту в реальному часі. API надає великий обсяг даних, які були оброблені та представлені у зручному форматі на вебсайті.

Протягом роботи я зосередився на ключових аспектах веброзробки, таких як аналіз предметної області, проєктування сайту, реалізація функціональної частини та тестування. Результати показали, що використання сучасних технологій та дотримання стандартів розробки значно покращують якість вебдодатків, що є критично важливим для задоволення потреб сучасних користувачів.

Розроблений вебресурс з кіберспорту має унікальний дизайн, але основною його перевагою є інтеграція з OpenDota API, яка дозволяє користувачам завжди бути в курсі останніх подій у світі кіберспорту. Це забезпечує легкий доступ до детальної інформації про команди та їх матчі.

Підсумовуючи, можна сказати, що всі поставлені завдання в рамках дипломної роботи були успішно виконані. Інтеграція з OpenDota API стала ключовим елементом успіху проєкту, забезпечивши користувачів найсвіжішою та найрелевантнішою інформацією.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Образцова М. Довідник з фундаментальної теорії JavaScript. Вивчаємо програмування українською. URL: <https://dou.ua/forums/topic/43210/> (дата звернення: 13.03.2024).
2. Філософія React. URL: <https://uk.legacy.reactjs.org/docs/thinking-in-react.html> (дата звернення: 14.03.2024).
3. Get started with React and Material-UI in no time. URL: <https://v4.mui.com/getting-started/usage/#usage> (дата звернення: 14.03.2024).
4. HTTP-клієнт для браузера та node.js на основі Promise. *Axios*. URL: <https://axios-http.com/uk/docs/intro> (дата звернення: 17.03.2024).
5. Чеботарьов А. Sass (Syntactically Awesome StyleSheets). URL: <https://ac2epsilon.github.io/TRANS/CSS/SASS%20Full.html> (дата звернення: 19.03.2024).
6. API – Corefy. URL: <https://corefy.com/uk/glossary/api> (дата звернення: 22.03.2024).
7. OpenDota – Гра для Mac, Windows, Linux – WebCatalog. URL: <https://webcatalog.io/uk/games/opendota/> (дата звернення: 25.03.2024).
8. Flanagan D. JavaScript: The Definitive Guide, 6th Edition. O'Reilly Media, 2006. 121 p.
9. Duckett J. JavaScript and JQuery: Interactive Front-End Web Development. Wiley, 2014. 640 p.
10. Roy T. Fullstack D3 and Data Visualization: Build beautiful data visualizations with D3. Newline, 2019. 650 p.