

«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ БІЗНЕСУ ТА СУЧАСНИХ
ТЕХНОЛОГІЙ**

**ФОРМА НАВЧАННЯ ДЕННА
КАФЕДРА МАТЕМАТИЧНОГО МОДЕЛЮВАННЯ ТА СОЦІАЛЬНОЇ
ІНФОРМАТИКИ**

Допускається до захисту

Завідувач кафедри _____ О.В. Ольховська
(підпис)

«_____» _____ 2025 р.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО БАКАЛАВРСЬКОЇ РОБОТИ**

**на тему
РОЗРОБКА ІНФОРМАЦІЙНОГО ВЕБСАЙТУ ПО ГРІ WARFRAME**

зі спеціальності 122 «Комп'ютерні науки»

Виконавець роботи Юдкін Дмитро Олександрович _____ «___» _____ 2025р.
(підпис)

Науковий керівник к.ф.-м.н., доц., Чілікіна Тетяна Василівна
_____ «___» _____ 2025р.
(підпис)

ПОЛТАВА 2025 р.

«ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ»

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ О.В. Ольховська
(підпис)

«_____» _____ 2025р.

ЗАВДАННЯ ТА КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ БАКАЛАВРСЬКОЇ РОБОТИ

Студент(ка) спеціальності 122 «Комп'ютерні науки»

Прізвище, ім'я, по батькові Юдкін Дмитро Олександрович

1. Тема «Розробка інформаційного вебсайту по грі Warframe»

затверджена наказом ректора № _____ від «__» _____ 2025 р.

Термін подання студентом бакалаврської роботи _____ «__» _____ 2025 р.

2. Вихідні дані до дипломної роботи: офіційна документація та матеріали, розміщені на ресурсах, присвячених грі Warframe, зокрема <https://www.warframe.com/uk> та <https://wiki.warframe.com>; аналіз структури сучасних інформаційних сайтів і геймерських порталів, що використовуються для ознайомлення користувачів з контентом і функціональністю відеоігор; загальні принципи створення UI/UX для інформаційних вебзастосунків; вимоги до розробки й організації SPA-застосунків із використанням React, TypeScript, Tailwind CSS, Vite тощо; системні технічні обмеження, пов'язані з запуском проєкту в браузері.

Практична частина дипломної роботи представлена односторінковим багатосекційним вебзастосунком з переходами на окремі сторінки. Ці сторінки вміщують інформацію про персонажів, фракції, внутрішньоігрову економіку, корабель гравця та загальні механіки гри. У структурі сайту також передбачено мультимедійний блок з трейлером, розділ відгуків, кнопки для переходу на зовнішні ресурси, а також інші інтерактивні елементи.

3. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Вступ

1. ПОСТАНОВКА ЗАДАЧІ

1.1. Формулювання задачі

1.2. Обмеження та вимоги до системи

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Аналіз предметної області

2.2. Аналіз аналогів (огляд існуючих сайтів подібної тематики)

2.3. Вибір програмних засобів та технологій розробки

2.4. Характеристика середовища

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Основні поняття та моделі

3.2. Структура вебзастосунків типу SPA

3.3. Особливості організації інтерфейсів

4. ПРАКТИЧНА ЧАСТИНА

4.1. Логічна та фізична структура сайту

4.2. Реалізація основних сторінок і компонентів

4.3. Реалізація навігації, функціональних та мультимедіа блоків

4.4. Стилізація та кастомізація інтерфейсу

4.5. Тестування сайту

Висновки

4. Перелік графічного матеріалу (з точним визначенням кількості блок-схем, іншого графічного матеріалу) _____

5. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1. Постановка задачі	Чілікіна Т.В.		
2. Інформаційний огляд	Чілікіна Т.В.		
3. Теоретична частина	Чілікіна Т.В.		
4. Практична частина	Чілікіна Т.В.		

6. Календарний графік виконання бакалаврської роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		

Зміст роботи	Термін виконання	Фактичне виконання
8. Доповідь студента на кафедрі		
9. Доробка (за необхідністю), рецензування		

Дата видачі завдання «___» _____ 2025 р.

Студент _____ Юдкін Дмитро Олександрович
(підпис)

Науковий керівник _____ к.ф.-м.н., доц., Чілікіна Тетяна Василівна
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту бакалаврської роботи

Дипломна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «___» _____ 2025 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

РЕФЕРАТ

Записка:

ІНФОРМАЦІЙНИЙ САЙТ, ВЕБРОЗРОБКА, WARFRAME, ІНТЕРАКТИВНІСТЬ, АДАПТИВНІСТЬ

Об'єкт дослідження - інформаційний вебсайт, орієнтований на популярну комп'ютерну гру Warframe.

Мета роботи - створення сучасного інформаційного вебсайту про гру Warframe із застосуванням веб-технологій, для забезпечення зручного й ефективного доступу до інформації для користувачів.

Методи дослідження - аналіз і порівняння функціональних можливостей існуючих аналогів вебсайтів присвячених комп'ютерним іграм; проектування архітектури застосунку із застосуванням компонентного підходу; програмування за допомогою JavaScript-фреймворку React та статичної типізації через TypeScript; використання інструментів зборки Vite і стилізації на основі Tailwind CSS. Також використано методи тестування та аналізу функціональності вебсайту.

Результат дослідження - функціональний інформаційний вебсайт, який містить такі ключові сторінки: Головна, Персонажі, Корабель, Ігрова система, Фракції, Ресурси. Реалізовано повну маршрутизацію між сторінками, розроблено інтерактивні елементи інтерфейсу та впроваджено адаптивну верстку, яка забезпечує коректне відображення на мобільних і настільних пристроях. Вебсайт демонструє високу швидкість завантаження, зручну навігацію та інтерактивність.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	9
1. ПОСТАНОВКА ЗАДАЧІ	
1.1. Формулювання задачі.....	11
1.2. Обмеження та вимоги до системи.....	12
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	
2.1. Аналіз предметної області.....	14
2.2. Аналіз аналогів (огляд існуючих сайтів подібної тематики).....	15
2.3. Вибір програмних засобів та технологій розробки.....	15
2.4. Характеристика середовища.....	16
3. ТЕОРЕТИЧНА ЧАСТИНА	
3.1. Основні поняття та моделі.....	20
3.2. Структура вебзастосунків типу SPA.....	21
3.3. Особливості організації інтерфейсів.....	22
4. ПРАКТИЧНА ЧАСТИНА	
4.1. Логічна та фізична структура сайту.....	24
4.2. Реалізація основних сторінок і компонентів.....	26
4.3. Реалізація навігації, функціональних та мультимедіа блоків.....	29
4.4. Стилiзація та кастомізація інтерфейсу.....	30
4.5. Тестування сайту.....	32
ВИСНОВКИ.....	35
СПИСОК ЛІТЕРАТУРИ.....	36
ДОДАТОК А.....	37

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

API (Application Programming Interface) - інтерфейс прикладного програмування, який дозволяє застосункам взаємодіяти між собою або з сервером.

SPA (Single Page Application) - односторінковий вебзастосунок, у якому всі сторінки підвантажуються без перезавантаження HTML-документа.

DOM (Document Object Model) - модель структури HTML-документа, яка дозволяє програмно змінювати вміст і структуру вебсторінки.

UI (User Interface) - інтерфейс користувача, графічне або текстове середовище, через яке користувач взаємодіє із застосунком.

UX (User Experience) - досвід користувача, враження та ефективність взаємодії з інтерфейсом.

URL (Uniform Resource Locator) - уніфікований вказівник ресурсу; адреса, за якою можна отримати доступ до вебсторінки або файлу.

JSON (JavaScript Object Notation) - текстовий формат зберігання та обміну даними, що використовується для серіалізації об'єктів.

JSX (JavaScript XML) - розширення синтаксису JavaScript, яке дозволяє описувати структуру інтерфейсу у вигляді HTML-подібного коду (використовується у React).

HTML (HyperText Markup Language) - мова розмітки гіпертексту, що визначає структуру вебсторінок.

CSS (Cascading Style Sheets) - таблиці стилів, що використовуються для опису зовнішнього вигляду HTML-документів.

Tailwind CSS - утилітарний CSS-фреймворк, що дозволяє будувати інтерфейс шляхом комбінування класів прямо в розмітці.

Vite - сучасний інструмент збірки вебпроектів, що забезпечує швидкий запуск і оптимізацію проекту.

React - JavaScript-бібліотека для побудови інтерфейсів, яка базується на компонентному підході та віртуальному DOM.

PvE (Player versus Environment) - ігровий режим, у якому гравці взаємодіють із комп'ютерними супротивниками або сценаріями, а не між собою.

TypeScript - мова програмування на базі JavaScript із підтримкою статичної типізації.

Git - система контролю версій, яка дозволяє відстежувати зміни у файлах під час розробки проєкту.

GitHub - вебсервіс для розміщення репозиторіїв Git та співпраці над програмними проєктами.

Компонент - структурна одиниця React-застосунку, яка містить логіку, розмітку та стилі для певної частини інтерфейсу.

Маршрутизація - система переходу між сторінками в межах SPA, що працює без перезавантаження браузера.

Користувач - фізична особа, яка переглядає або взаємодіє з вебсайтом.

ВСТУП

Сьогоднішній розвиток ігрової індустрії вимагає постійного вдосконалення як самих ігор, так і супровідних інформаційних ресурсів, що допомагають гравцям краще орієнтуватися у складних геймплейних механіках. Однією з таких ігор є Warframe [12] - безкоштовний онлайн-шутер з елементами RPG, який має глибоку внутрішню структуру, насичений лор та унікальні механіки розвитку персонажа. Саме зацікавленість у цій грі стала поштовхом до створення власного інформаційного вебсайту, який мав би на меті впорядкувати ключові знання про Warframe, адаптовані для україномовної аудиторії.

Мета дипломної роботи - розробка інформаційного сайту по грі Warframe з урахуванням сучасних технологій веброзробки та принципів зручності користувача (UX). Для досягнення цієї мети необхідно було:

- проаналізувати існуючі сайти, що висвітлюють тематику гри Warframe;
- обрати оптимальні інструменти для реалізації односторінкового застосунку (SPA);
- реалізувати дизайн, структуру, мультимедійні блоки, а також функціональність переходу між сторінками;
- протестувати роботу сайту та оцінити його зручність.

Об'єкт розробки - процес створення інформаційних вебресурсів у тематиці відеоігор.

Предмет розробки - структура, функціональність та інтерфейс сайту, що реалізований за допомогою сучасного інструментарію фронтенд-розробки.

Методи дослідження, що були використані у процесі розробки, включають:

- порівняльний аналіз аналогічних сайтів для визначення кращих підходів до подачі інформації;
- огляд та тестування фреймворків (React, Vue, Angular), серед яких було обрано React, як найбільш зручний та адаптивний для створення SPA;

- практичне застосування фреймворку React у поєднанні з TypeScript, Tailwind CSS та системою збірки Vite.

Практична цінність дипломної роботи полягає у створенні повноцінного адаптивного сайту з логічною структурою та можливістю його подальшого розміщення у вільному доступі.

1. ПОСТАНОВКА ЗАДАЧІ

1.1. Формулювання задачі.

Warframe - через велику кількість механік і термінів, новим гравцям часто складно зорієнтуватися. Це стало підставою для розробки інформаційного сайту, метою якого є надання стислої, структурованої та зрозумілої інформації про основні аспекти гри. Під час проєктування особливу увагу було приділено саме зручності для новачків, водночас передбачено і можливість для досвідчених гравців швидко переходити на зовнішні ресурси для отримання більш детальної інформації. Такий підхід дозволяє поєднати базовий контент із можливістю глибшого ознайомлення з темою.

Завдання, полягало у створенні односторінкового застосунку (SPA) з інтуїтивним інтерфейсом, адаптованим до різних екранів, реалізованого за допомогою сучасних інструментів веброботки. Сайт повинен мати чітку навігацію, логічну структуру, мультимедійні можливості, базову адаптивність і зручність у використанні.

Основні задачі, які вирішуються в межах дипломної роботи:

- виокремлення ключових розділів гри Warframe, що підлягають опису на сайті;
- розробка логічної структури сайту й визначення навігаційних зв'язків між сторінками;
- вибір актуальних фреймворків і технологій;
- створення інтерфейсу із можливістю вбудовування мультимедійного контенту;
- реалізація механізму завантаження сторінок без оновлення (SPA-навігація);
- додавання функціональних блоків (відео, переходи, завантаження, тощо);
- тестування функціоналу та перевірка відображення на різних пристроях.

1.2. Обмеження та вимоги до системи.

Під час розробки інформаційного вебсайту для гри Warframe було визначено низку функціональних та нефункціональних вимог, що забезпечують працездатність, зручність і актуальність ресурсу.

Функціональні вимоги:

- сайт повинен містити інформацію за ключовими напрямками гри: ігрова система, персонажі (ворфрейми), ресурси й торгівля, фракції, кораблі;
- повинна бути реалізована зручна навігація між сторінками без перезавантаження (SPA-навігація);
- кожна сторінка повинна мати чітко структурований зміст із заголовками, текстовими блоками та, за потреби, мультимедіа (наприклад, відео);
- передбачена кнопка переходу на офіційний сайт гри для завантаження;
- у нижній частині окремих сторінок реалізовано перехід на зовнішні ресурси з розширеною інформацією.

Нефункціональні вимоги:

- сайт має бути орієнтований переважно на перегляд з персональних комп'ютерів, але базова адаптивність під інші типи екранів також повинна зберігатися;
- час завантаження сторінки повинен залишатися мінімальним завдяки використанню Vite як збирача;
- сторінки мають бути стилістично єдиними, з дотриманням кольорової гами, притаманної грі Warframe;
- рішення має бути реалізоване з використанням сучасних технологій: React (бібліотека інтерфейсів), TypeScript (сувора типізація), Tailwind CSS (стилізація), Vite (швидкий дев-сервер);
- код сайту має бути структурований, розбитий на компоненти, що полегшує його підтримку та розширення.

Обмеження:

- у поточній реалізації не передбачено повноцінної серверної частини (бекенду), оскільки основна увага зосереджена на поданні інформації та зручному інтерфейсі;
- усі дані на сайті представлені у статичному вигляді, без запиту до зовнішніх API;
- інформація оновлюється вручну, тож актуальність матеріалу залежить від своєчасності її редагування.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Аналіз предметної області.

Одною із яскравих безкоштовних багатокористувацьких онлайн-ігор є Warframe - гра, розроблена канадською студією Digital Extremes. Її перший публічний реліз відбувся у 2013 році, а з того часу вона постійно оновлюється та розширюється, залучаючи нову аудиторію та підтримуючи інтерес спільноти ветеранів гри (див. рисунок 2.1).

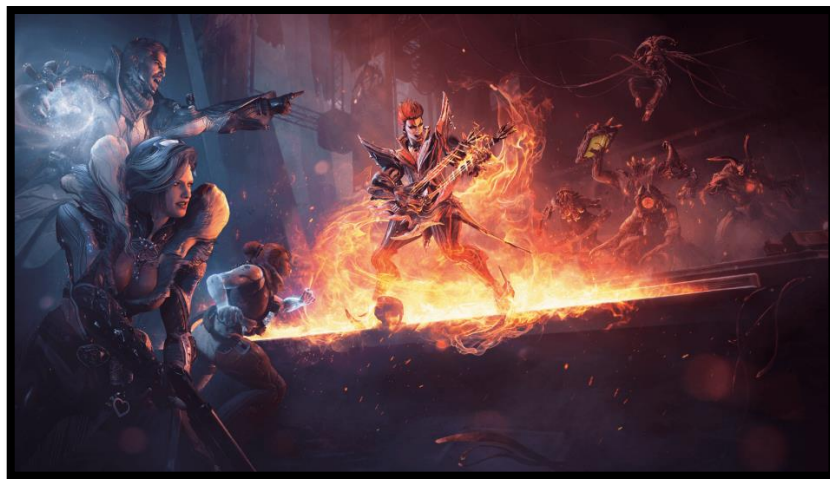


Рисунок 2.1 - Актуальний логотип гри Warframe

Warframe поєднує елементи шутера від третьої особи, рольової гри та кооперативного PvE-досвіду у науково-фантастичному сетингу. Користувачі виступають у ролі Тенно - стародавньої раси воїнів, які керують бойовими біомеханічними костюмами, відомими як ворфрейми. Кожен ворфрейм має унікальні здібності та стиль гри, що дозволяє гравцям адаптувати свій підхід до бойових ситуацій.

Гра має велику кількість механік: зоряну карту з десятками місій, систему крафту, гілку розвитку персонажів, модифікатори, синдикати, бойові арени, систему обміну ресурсами між гравцями тощо. Такий обсяг контенту, а також регулярні оновлення, можуть створити труднощі в орієнтуванні для нових користувачів, особливо без належної документації.

Незважаючи на наявність офіційного довідника та фанатських вікі-ресурсів, багато гравців шукають стислий, візуально привабливий і

структурований сайт, де подано лише найнеобхіднішу інформацію. Саме таку задачу виконує розроблений інформаційний вебсайт. Він концентрується на основних аспектах гри, таких як персонажі, ігрова система, корабель Оператора, фракції, ресурси та механіка торгівлі, забезпечуючи швидкий доступ до ключових знань, необхідних для комфортної гри.

2.2. Аналіз аналогів (огляд існуючих сайтів подібної тематики).

Перед початком розробки вебсайту було проведено аналіз кількох існуючих інформаційних ресурсів, пов'язаних з грою Warframe. Основною метою цього огляду було визначення сильних та слабких сторін подібних сайтів, а також виявлення можливостей для покращення структури та подання інформації у власному проєкті.

Серед найпопулярніших джерел - офіційна вікі Warframe (<https://warframe.fandom.com>), яка містить детальні описи всіх елементів гри. Незважаючи на глибину контенту, структура сторінок є перевантаженою, що може ускладнити сприйняття матеріалу новачками. Інтерфейс платформи також не оптимізований для швидкого доступу до ключової інформації.

Інші ресурси, такі як офіційний сайт гри (<https://www.warframe.com>), фокусуються переважно на маркетинговому контенті, оновленнях та трейлерах. Там бракує структурованого довідника, що об'єднував би базові знання про гру в компактній формі.

Також є фанатські блоги, YouTube-канали та форуми, однак більшість із них не мають зручної навігації або підтримки мультимедійних блоків з адаптивною версткою.

Таким чином, постала потреба у створенні сайту, який би поєднував візуальну простоту, адаптивність, структурованість та зрозумілість подачі матеріалу. Саме ці якості закладено в основу розробленого інформаційного ресурсу.

2.3. Вибір програмних засобів та технологій розробки.

При створенні вебзастосунку було прийнято рішення використовувати сучасний стек технологій, орієнтований на зручність розробки, високу продуктивність та підтримку адаптивного дизайну.

Основним фреймворком було обрано React [8,9], що забезпечує компонентну архітектуру, високу продуктивність та широке ком'юніті підтримки. Завдяки використанню TypeScript, вдалося досягти кращої передбачуваності коду, типізації та зниження кількості помилок під час розробки.

Для швидкої збірки та оптимізації проєкту використовується Vite [2] - інструмент, який значно прискорює процес запуску та гарячого оновлення (hot reload), що важливо для розробки SPA-додатків.

Візуальна частина сайту стилізується за допомогою Tailwind CSS - утилітарного CSS-фреймворку, що дозволяє швидко формувати адаптивні та сучасні інтерфейси без надлишкового перевантаження стилів. Це також значно полегшило структурування шаблонів сторінок і зменшило кількість кастомного CSS-коду.

2.4. Характеристика середовища.

Для розробки вебсайту було обрано сучасне середовище розробки Visual Studio Code (VS Code) [13], що забезпечує зручність при написанні та структуризації коду, завдяки великій кількості доступних розширень і підтримці різних мов програмування. Проєкт створено з використанням збірки Vite - сучасного інструменту для швидкої ініціалізації та розгортання вебпроєктів, який відзначається високою швидкістю компіляції та підтримкою модульної структури.

До складу встановлених і задіяних розширень входили:

- ESLint - для аналізу та автоматичного виправлення помилок у коді (див. рисунок 2.2);

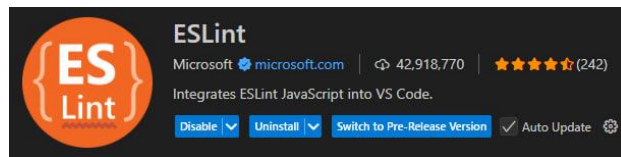


Рисунок 2.2 - Вигляд розширення ESLint

- Prettier - для форматування вихідного коду згідно з заданими правилами стилю (див. рисунок 2.3);

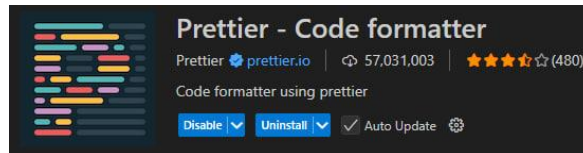


Рисунок 2.3 - Вигляд розширення Prettier

- Tailwind CSS IntelliSense - для автозаповнення класів Tailwind CSS (див. рисунок 2.4);



Рисунок 2.4 - Вигляд розширення Tailwind CSS IntelliSense

- IntelliCode - для надання контекстно-залежних підказок (див. рисунок 2.5);

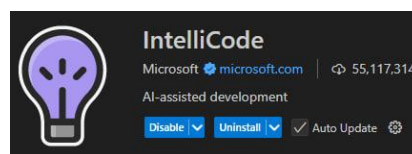


Рисунок 2.5 - Вигляд розширення IntelliCode

- Quokka.js - для тестування JS/TS фрагментів у реальному часі (див. рисунок 2.6);



Рисунок 2.6 - Вигляд розширення Quokka.js

- HTML to CSS autocompletion, JavaScript snippets, JSX HTML, Babel JavaScript, тощо - для зручності написання коду та розмітки (див. рисунок 2.7).

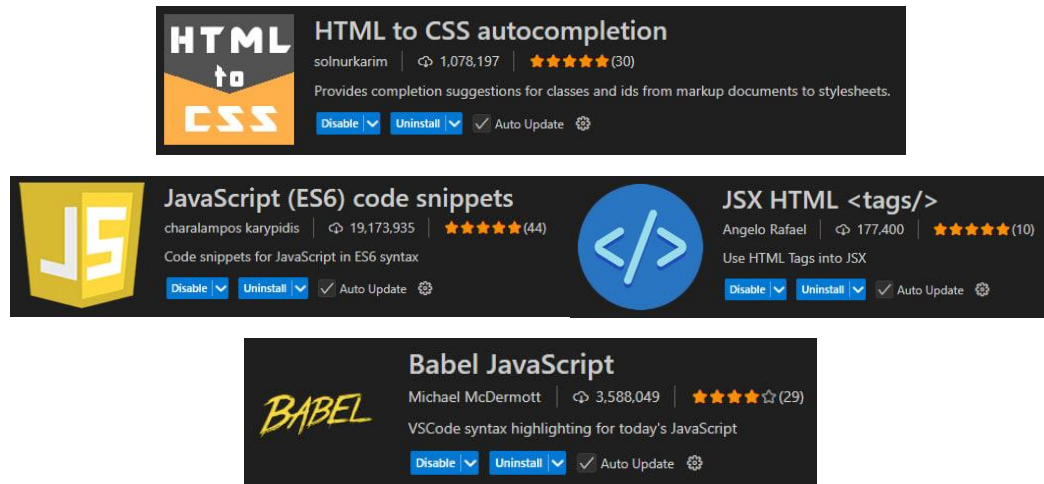


Рисунок 2.7 - Набір додаткових розширень

Під час роботи було встановлено і налаштовано Node.js, необхідний для використання менеджера пакетів npm, через який встановлювались бібліотеки та залежності. Для контролю версій використовувався Git, а також було створено віддалений репозиторій на GitHub для зберігання й резервного копіювання коду проєкту.

Проєкт локально запускався через npm-скрипти, які автоматично компілювали код і відкривали локальний сервер у браузері (див. рисунок 2.8). Розгортання та тестування вебсайту здійснюється в середовищі розробника, із використанням Vite як локального сервера. Запуск здійснюється командою `npm run dev`, що забезпечує швидке збирання та гаряче оновлення сторінок у браузері під час розробки. Такий підхід дозволив проводити оперативне тестування компонентів сайту та швидко вносити правки в інтерфейс.

```
PS C:\Users\dimon\OneDrive\Робочий стіл\project> npm run dev  
  
> vite-react-typescript-starter@0.0.0 dev  
> vite  
  
VITE v5.4.11 ready in 156 ms  
  
→ Local:   http://localhost:5173/  
→ Network: use --host to expose  
→ press h + enter to show help
```

Рисунок 2.8 - Запуск локального серверу.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Основні поняття та моделі.

Веброзробка - це процес створення вебзастосунків, які можуть взаємодіяти з користувачем у реальному часі та працювати у браузері без потреби в традиційних десктопних програмах. В основі більшості сучасних вебпроектів лежить модель клієнт-серверної взаємодії, де браузер користувача виступає клієнтом, а спеціалізований сервер обробляє запити і надає відповіді, часто у форматі HTML, CSS, JavaScript або даних JSON.

Сучасні вебзастосунки часто реалізуються у форматі SPA (Single Page Application - односторінковий застосунок). Їхня особливість полягає у тому, що завантаження сторінки відбувається лише один раз, після чого вся взаємодія здійснюється без повного оновлення сторінки. Це досягається завдяки динамічному оновленню вмісту DOM (Document Object Model) [11] - внутрішньої структури HTML-документа, яку браузер інтерпретує як дерево елементів (див. рисунок 3.1). DOM є однією з ключових концепцій у веброзробці, оскільки саме через нього відбувається зміна вигляду та поведінки інтерфейсу без перезавантаження сторінки. За допомогою DOM-маніпуляцій розробники можуть змінювати будь-які елементи на сторінці: додавати, видаляти чи змінювати контент у відповідь на дії користувача.

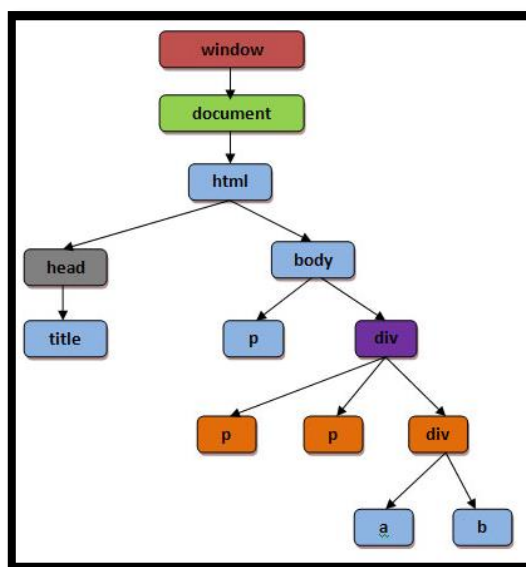


Рисунок 3.1 – Схематичне представлення моделі DOM у вигляді дерева елементів HTML-документа

Іншою важливою моделлю, що використовується у розробці, є компонентна модель. Вона дозволяє розділити інтерфейс застосунку на окремі, ізольовані блоки (компоненти), кожен із яких відповідає за власну логіку, стан і візуальне представлення. Компонентна архітектура значно спрощує підтримку коду, повторне використання частин інтерфейсу та масштабування застосунку.

На практиці вся взаємодія між компонентами, зміна стану, обробка подій і оновлення DOM здійснюється з урахуванням реактивного підходу, який забезпечує миттєву реакцію застосунку на зміну даних. Описані вище моделі стали основою для реалізації інформаційного вебзастосунку, орієнтованого на зручне і швидке отримання даних без надмірного навантаження на браузер і без повторного завантаження сторінок.

3.2. Структура вебзастосунків типу SPA.

Односторінкові застосунки (Single Page Applications, або SPA) є сучасною архітектурною моделлю веброзробки, яка дозволяє створювати динамічні й інтерактивні сайти без необхідності перезавантаження сторінки при переході між розділами. Основна ідея полягає в тому, що при першому завантаженні клієнт отримує лише один HTML-документ, а подальша взаємодія з сайтом відбувається через асинхронне оновлення його вмісту.

Основні елементи структури SPA:

- кореневий HTML-файл (index.html) - містить контейнер (наприклад, `<div id="root">`), у який динамічно підвантажується весь контент застосунку;
- головний JavaScript-компонент (App.tsx) - керує відображенням сторінок та компонентів через внутрішню логіку маршрутизації;
- сторінки (pages/*.tsx) - є окремими візуальними блоками, які умовно представляють різні «вебсторінки» всередині одного HTML-файлу;

- компоненти (components/*.tsx) - повторно використовувані елементи інтерфейсу (наприклад, кнопки, секції, картки), що будуються відповідно до принципів компонентного підходу React.

SPA застосовує механізм клієнтської маршрутизації: адреса в URL змінюється (наприклад, /characters або /resources), але сам HTML-документ не перезавантажується. Це забезпечується бібліотеками маршрутизації, як-от react-router-dom, які дозволяють організовувати переходи між сторінками з використанням <Routes> та <Route>.

Переваги SPA:

- висока швидкість відгуку - після початкового завантаження сторінки зміни відбуваються миттєво;
- менше навантаження на сервер - передаються лише необхідні дані у вигляді JSON або іншим способом;
- кращий користувацький досвід - плавні анімації, динамічні елементи, інтерактивність.

3.3. Особливості організації інтерфейсів.

Організація інтерфейсу користувача є важливою складовою при створенні будь-якого вебзастосунку, оскільки саме інтерфейс визначає зручність взаємодії користувача з системою. В рамках дипломного проєкту основним завданням було створити логічно структурований, візуально зрозумілий і привабливий інтерфейс, який дозволяє швидко орієнтуватися у вмісті сайту.

Для досягнення цієї мети було використано такі підходи:

- 1) Структурованість - інтерфейс розділено на сталі блоки, які повторюються на всіх сторінках сайту: навігаційна панель у верхній частині, контентна секція по центру та нижній блок (футер) з посиланнями. Це забезпечує єдину логіку розташування елементів та зменшує когнітивне навантаження на користувача.
- 2) Візуальна ієрархія - інформація подається у вигляді логічних секцій із заголовками, підзаголовками, відступами, кнопками, зображеннями та

інтерактивними елементами. Завдяки цьому користувач може швидко визначити головну думку кожного розділу, навіть переглядаючи сайт поверхнево.

3) Адаптивність - хоча сайт орієнтований переважно на перегляд з ПК, деякі стилі й компоненти були розміщені з урахуванням адаптивної верстки - елементи змінюють своє розташування або розмір відповідно до ширини екрана, забезпечуючи базову зручність на планшетах або мобільних пристроях.

4) Колірна схема та типографіка - вся стилізація витримана у темній гамі з контрастними елементами, що відповідає атмосфері гри Warframe. Використано один основний шрифт у поєднанні з різними розмірами та насиченістю для розмежування заголовків і звичайного тексту. Активні елементи мають візуальний зворотній зв'язок при наведенні (ефекти наведення, анімації тощо).

5) Інтерактивність і навігація - використано фіксовану навігаційну панель, яка дозволяє переходити між сторінками сайту без перезавантаження. На сторінках також розміщено кнопки швидкого доступу до відео, зовнішніх ресурсів, розширеної інформації, що спрощує взаємодію.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Логічна та фізична структура сайту.

Перш ніж перейти до реалізації функціональності, було створено логічну та фізичну структуру сайту, що дозволяє розділити проєкт на зрозумілі блоки, полегшити його підтримку та масштабування.

Логічна структура визначає змістовне наповнення сайту та поділ інформації на окремі тематичні сторінки. Сайт складається з таких основних сторінок (див. рисунок 4.1):

- головна сторінка - короткий опис проєкту, вступна інформація, відео-блок, відгуки, кнопка завантаження гри;
- персонажі гри - опис початкових варфреймів, здібностей, принципів отримання та прокачування;
- корабель гравця - інформація про Орбітер, функціональні модулі (Арсенал, Кузня тощо), інтерфейс корабля;
- ігрова система - опис типів місій, особливостей зоряної карти, геймплейних механік;
- фракції та синдикати - опис основних ворогуючих фракцій, вплив системи репутації та альянсів;
- ресурси і торгівля - поділ ресурсів за типами, принципи здобуття, особливості обміну між гравцями.

Окрім основних сторінок, на сайті реалізовано: навігаційну панель (з можливістю переходу між сторінками), футер (нижню частину з посиланнями на зовнішні ресурси та соціальні мережі) та додаткові блоки (мультимедіа, анімовані елементи, кнопки переходу тощо).

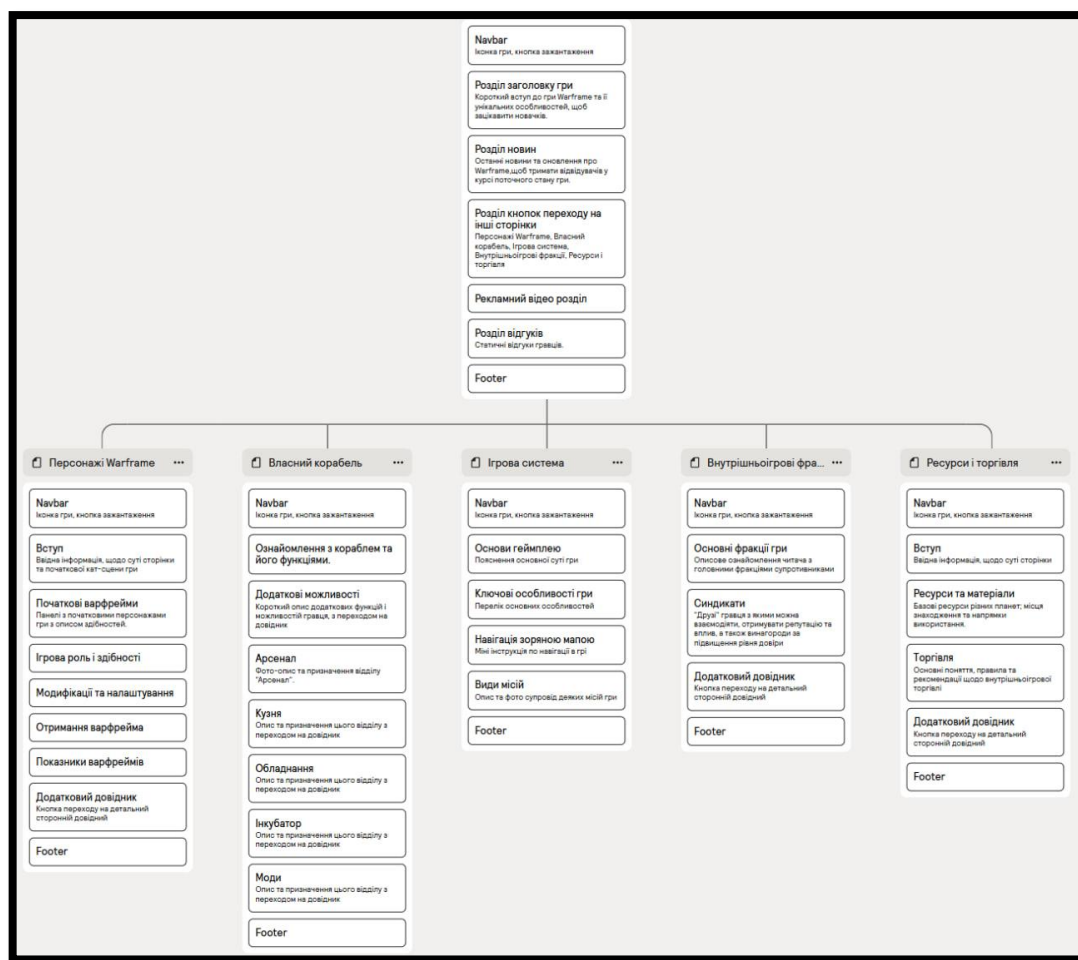


Рисунок 4.1 – Логічна структура сайту

Фізична структура описує файлову організацію та структуру директорій у проєкті (див. рисунок 4.2).

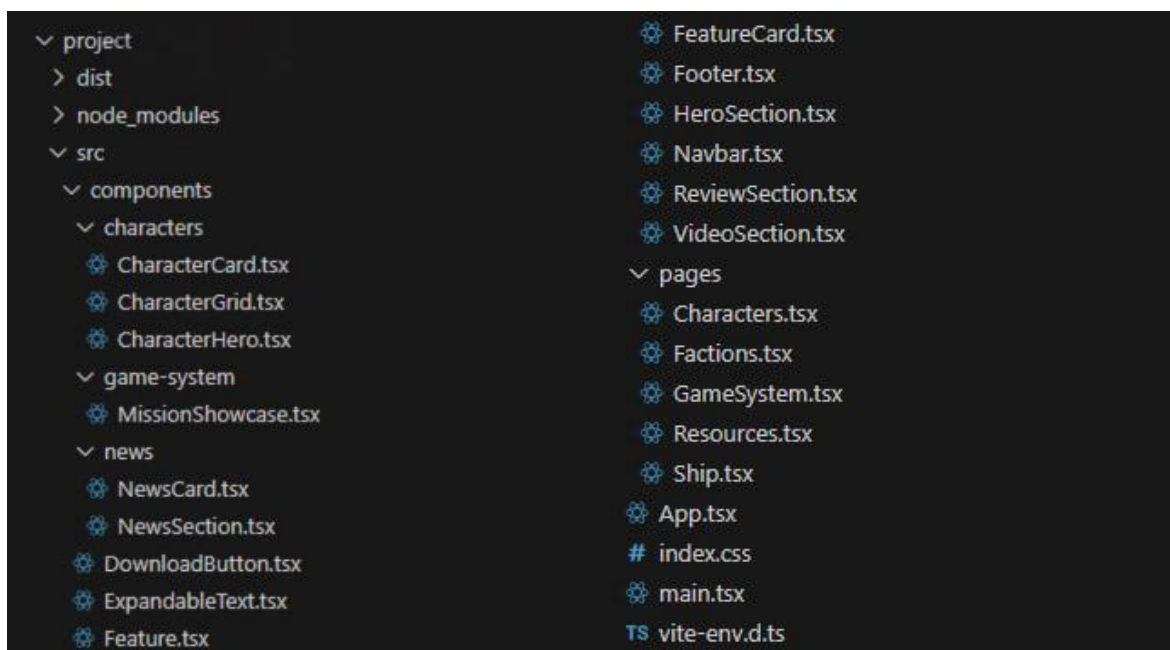


Рисунок 4.2 – Фізична структура сайту

4.2. Реалізація основних сторінок і компонентів.

У процесі створення інформаційного сайту про гру Warframe було реалізовано набір повноцінних сторінок, кожна з яких відповідає окремому тематичному блоку гри. Кожна сторінка містить власну структуру, логіку відображення контенту та використовує компонентний підхід, що забезпечує гнучкість і повторне використання елементів.

Основні сторінки сайту:

1) Головна сторінка (відображається через App.tsx) (див. рисунок 4.3):

- містить вступний опис гри, кнопки «See more» та «Watch Trailer».
- включає мультимедійний відеоблок, секцію новин, блоки з картками розділів сайту, а також розділ з відгуками гравців.
- реалізовано через компоненти: HeroSection, VideoSection, NewsSection, ReviewSection, Feature, DownloadButton, Footer.

2) Сторінка “Персонажі” (Characters.tsx) (див. рисунок 4.4):

- інформація про перших доступних гравцю варфреймів (Excalibur, Mag, Volt).
- додано розділ про здібності, прокачку, типи варфреймів та умови їх отримання.
- реалізовано через компоненти CharacterHero, CharacterGrid, CharacterCard.

3) Сторінка “Корабель гравця” (Ship.tsx) (див. рисунок A.1 у Додатку A):

- опис функцій Орбітера: Арсенал, Кузня, Командний центр та ін.
- побудовано із візуально розділених секцій із заголовками й текстом.

4) Сторінка “Ігрова система” (GameSystem.tsx) (див. рисунок A.2 у Додатку A):

- включає типи місій, сюжетні квести, побічні завдання.
- структуровано як окремі блоки з описами, розміщеними у картках.

5) Сторінка “Фракції” (Factions.tsx) (див. рисунок A.3 у Додатку A):

- подано короткий опис основних фракцій гри та їхній вплив.
- окремий підрозділ присвячений синдикатам і системі репутації.

6) Сторінка “Ресурси і торгівля” (Resources.tsx) (див. рисунок А.4 у Додатку А):

- структурований перелік основних типів ресурсів, способи їх отримання.
- роз’яснено принципи обміну, важливість для крафту й прогресу гравця.

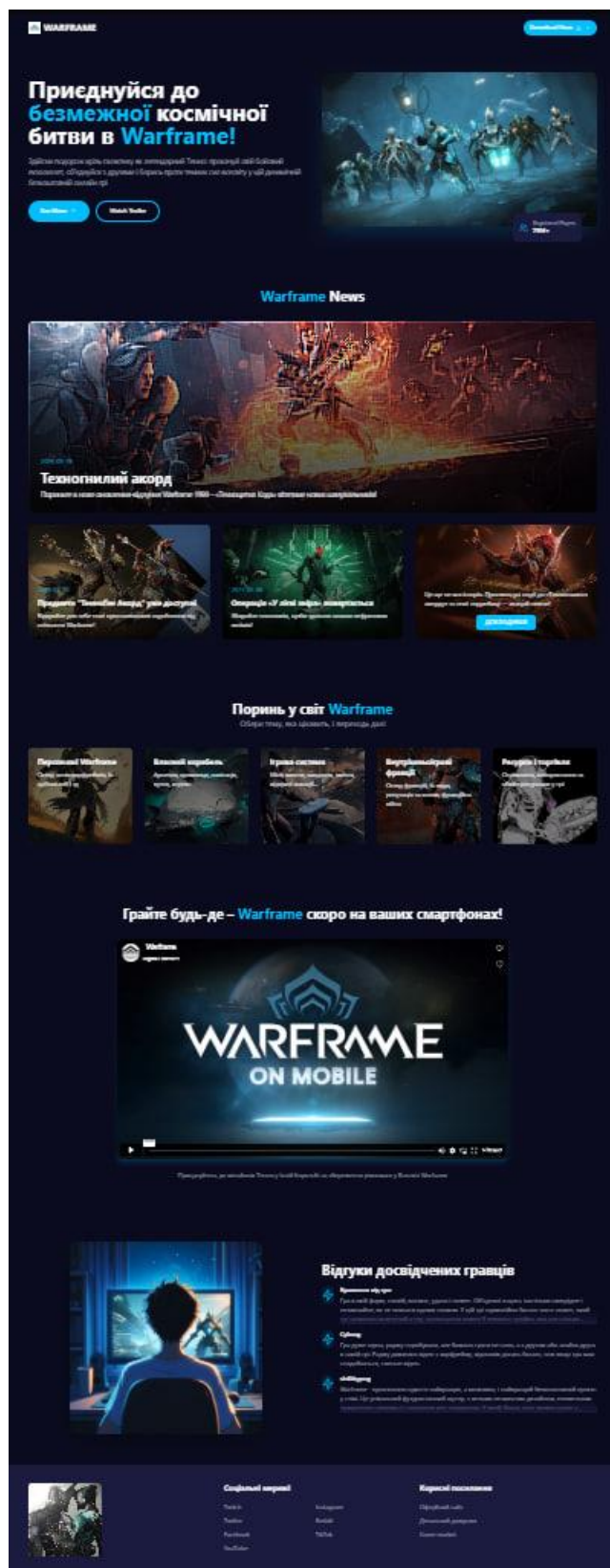


Рисунок 4.3 - Головна сторінка сайту



Рисунок 4.4 - Сторінка “Персонажі”

Повторно використовувані компоненти:

- 1) `Navbar.tsx` - фіксована верхня навігаційна панель для переходу між сторінками.
- 2) `Footer.tsx` - нижній блок із соціальними посиланнями та корисними зовнішніми ресурсами.
- 3) `ExpandableText.tsx` - компонент для скороченого тексту з можливістю «читати більше».
- 4) `FeatureCard.tsx`, `NewsCard.tsx`, `CharacterCard.tsx` - адаптовані блоки відображення контенту з динамічним оформленням.
- 5) `DownloadButton.tsx` - кнопка завантаження гри з посиланням на офіційний сайт.

4.3. Реалізація навігації, функціональних та мультимедіа блоків.

Для забезпечення зручної взаємодії користувача з сайтом було реалізовано набір ключових елементів інтерфейсу, які не тільки забезпечують навігацію між сторінками, а й підвищують динамічність, інформативність та візуальну привабливість ресурсу.

Навігація реалізована у вигляді фіксованої верхньої панелі (`Navbar.tsx`), яка містить:

- логотип сайту;
- назви основних розділів (посилання на головну, персонажів, фракції, корабель тощо).

Переходи між сторінками реалізовано з використанням `React-маршрутизації`, що дозволяє переміщуватися по сайту без перезавантаження сторінки. Такий підхід зберігає швидкість роботи та відповідає структурі `SPA`.

У межах сайту реалізовано декілька функціональних компонентів:

- 1) кнопка “Download now” (`DownloadButton.tsx`) - фіксоване посилання на офіційну сторінку завантаження гри;

- 2) кнопка “Watch Trailer” - миттєвий перехід до інтернет ресурсу з трейлером гри;
- 3) кнопки “See more” / “Детальніше” - використовуються для переходу на тематичні сторінки з головної;
- 4) блок з розширенням тексту (ExpandableText.tsx), який дозволяє приховувати частину тексту та розгортати її за натисканням, що особливо зручно при великому обсязі інформації.

Однією з ключових особливостей реалізації є додавання відео-контенту без використання бекенду. На головній сторінці розміщено відеоблок (VideoSection.tsx), у який вмонтовано трейлер гри за допомогою сервісу Vimeo. Відео реалізовано у вигляді `<iframe>` з налаштованими розмірами, заголовком і можливістю повноекранного режиму (див. рисунок 4.5). Враховуючи технічні обмеження в деяких середовищах, попередньо було протестовано різні сервіси відеохостингу, серед яких Vimeo виявився найстабільнішим та найзручнішим.

```
<iframe
  className="absolute inset-0 w-full h-full"
  src="https://player.vimeo.com/video/1059874886?h=a64cef053a"
  width="640"
  height="360"
  frameBorder="0"
  allowFullScreen
  onLoad={handleIframeLoad}
  onError={handleIframeError}
/>
```

Рисунок 4.5 - Фрагмент коду вставки відео з сервісу Vimeo

4.4. Стилізація та кастомізація інтерфейсу.

Для візуального оформлення сайту було використано Tailwind CSS - сучасний утилітарний CSS-фреймворк, який дозволяє швидко та ефективно стилізувати компоненти без створення окремих CSS-файлів. Tailwind дає змогу формувати дизайн безпосередньо у класах HTML-розмітки, що значно

прискорює розробку та зменшує обсяг дублюючого коду. Це дозволило зберегти унікальність дизайну, не втрачаючи при цьому гнучкість у верстці та масштабованість інтерфейсу.

Основні принципи стилізації (див. рисунок 4.6):

- 1) темна тема оформлення була обрана як основна, що відповідає візуальній стилістиці гри Warframe (темне тло, контрастний білий текст, синьо-фіолетові акценти);
- 2) шрифти та розміри підібрано з урахуванням зручності читання. Заголовки оформлені з використанням класів text-2xl, text-3xl, а текст - через text-base, text-lg;
- 3) всі елементи мають чітко виражену візуальну ієрархію: розміри, відступи (p-*, m-*), кольори (text-*, bg-*) та обводки (rounded-*) налаштовані вручну для кожного блоку.

```

<header className="container mx-auto px-6 py-20 grid grid-cols-1 lg:grid-cols-2 gap-12 items-center">
  <div>
    <h1 className="text-5xl lg:text-6xl font-bold mb-6">
      Приєднуйся до <span className="text-[#00BFFF]">Безмежної</span> космічної битви в <span className="text-[#00BFFF]">Warframe!</span>
    </h1>
    <p className="text-gray-400 text-lg mb-8">
      Здійсни подорож крізь галактику як легендарний Тенно: прокачай свій бойовий екзоскелет, об'єднайся з друзями і борись проти темних сил в
    </p>
    <div className="flex gap-4">
      <button
        onClick={scrollToFeatures}
        className="bg-[#00BFFF] hover:bg-[#0099FF] px-8 py-3 rounded-full font-semibold flex items-center gap-2 group"
      >
        See More
        <ChevronDown className="w-5 h-5 transition-transform group-hover:translate-y-1" />
      </button>
      <a
        href="https://youtu.be/Msbl8lFHzrZI?si=DmA9JKrfJ05VaqxV"
        target="_blank"
        rel="noopener noreferrer"
        className="border border-[#00BFFF] hover:bg-[#00BFFF]/10 px-8 py-3 rounded-full font-semibold text-center"
      >
        Watch Trailer
      </a>
    </div>
  </div>
  <div>
    <div className="relative">
      
      <div className="absolute -bottom-10 right-10 bg-[#1A1B3E] p-4 rounded-xl shadow-lg">
        <div className="flex items-center gap-3">
          <Users className="w-6 h-6 text-[#00BFFF]" />
          <div>
            <p className="text-sm text-gray-400">Registered Players</p>
            <p className="font-bold">75M+</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</header>

```

Рисунок 4.6 - Фрагмент стилізації головної сторінки

Робота з графічним контентом реалізована за допомогою зовнішнього хостингу зображень. Усі ілюстрації, що використовуються на сторінках сайту (персонажі, фон, елементи інтерфейсу), попередньо були оптимізовані та завантажені на спеціалізовану платформу онлайн-хостингу, після чого підключені до коду за допомогою прямого посилання (URL-адреси). Це дозволило:

- мінімізувати об'єм локального проєкту;
- забезпечити швидке завантаження сторінок при запуску;
- уникнути потреби у створенні окремої серверної частини для зберігання мультимедіа.

Усі посилання використовуються у відкритому доступі та підтримуються хостингом без обмежень у використанні, що також спрощує перевірку та редагування контенту під час розробки.

Адаптивність - Tailwind CSS надає вбудовану підтримку адаптивної верстки через медіакласи (sm:, md:, lg: тощо). У проєкті застосовано:

- гнучке вирівнювання блоків для різних розмірів екрана;
- адаптацію тексту та зображень, щоб уникнути обрізання контенту;
- зміни структури секцій на менших екранах (наприклад, вертикальне розташування замість горизонтального).

Хоча сайт первинно орієнтований на перегляд із ПК, базова адаптивність дозволяє комфортно працювати з ним і на планшетах або ноутбуках з різною роздільністю.

Ефекти взаємодії - щоб покращити користувацький досвід, було реалізовано:

- ефекти наведення на кнопки (hover:bg-*, hover:text-*);
- тіньові ефекти (shadow-*) для блоків новин, карток персонажів та кнопок;
- анімоване розгортання тексту в ExpandableText.tsx;
- плавний скрол до секцій при натисканні кнопок ("See more").

4.5. Тестування сайту.

Після завершення основної розробки було проведено локальне тестування сайту з метою перевірки його працездатності, коректного відображення вмісту та загальної стабільності функціонування.

Тестування виконувалось вручну у локальному середовищі за допомогою команди “`npm run dev`”, яка запускала сайт на вбудованому сервері Vite [2]. Під час перевірки послідовно тестувались такі аспекти:

- працездатність навігації між сторінками;
- відображення контенту кожної сторінки;
- правильне завантаження зображень та відео;
- коректна робота інтерактивних елементів (кнопок, анімацій, розгортання тексту);
- відповідність стилів очікуваному дизайну.

Перевірені функції:

- усі маршрути працюють відповідно до заданої логіки - при переході за посиланням змінюється лише вміст сторінки, без оновлення всієї HTML-структури.
- кнопка “Download now” відкриває офіційні сайти для завантаження гри.
- відео, вставлене через сервіс “Vimeo”, успішно відображається в `<iframe>` без помилок у браузері.
- реалізовано ефект розгортання тексту при наведенні, без візуальних збоїв.
- навігаційна панель коректно відображається на різних розмірах вікна.
- футер стабільно прикріплений у нижній частині кожної сторінки.

У процесі тестування були помічені дрібні неточності:

- при дуже малій ширині екрана окремі текстові блоки розтягуються;
- відео може не запускатися, якщо сайт переглядається в браузерах зі строгими CORS-налаштуваннями;
- у деяких компонентах відсутнє автоматичне перенесення довгих слів.

Загалом тестування показало, що сайт функціонує стабільно, усі ключові сторінки та компоненти відображаються відповідно до очікувань, а навігація

реалізована без помилок. Це дозволяє вважати розробку завершеною на етапі прототипу з високим рівнем готовності до розгортання.

Після локального тестування сайт було успішно опубліковано на хостинговій платформі Netlify [10], що дало змогу здійснити додаткову перевірку його працездатності в умовах реального доступу через браузер.

ВИСНОВКИ

У дипломній роботі було виконано повноцінну розробку інформаційного вебсайту, присвяченого грі Warframe. Проєкт охопив усі етапи створення SPA-застосунку: від аналізу предметної області та вибору інструментів до реалізації логіки, інтерфейсу, стилізації та тестування.

На основі поставленої мети та задач вдалося реалізувати багатосторінковий сайт, що складається з кількох тематичних розділів, зокрема: персонажі, корабель, ігрова система, ресурси, фракції та головна сторінка з мультимедійним контентом. Завдяки використанню сучасних технологій - React, TypeScript, Tailwind CSS та Vite - було досягнуто високої продуктивності, швидкості розробки та гнучкості структури.

У процесі реалізації особлива увага приділялася створенню зручного інтерфейсу, відповідного стилістиці гри, а також структурованому поданню контенту. Сайт має фіксовану навігаційну панель, адаптивні компоненти, мультимедійні вставки та можливість переходу на зовнішні ресурси. Також було реалізовано відеоблок із трейлером гри, анімації, кнопки завантаження та ефекти взаємодії.

Результати тестування підтвердили стабільну роботу сайту, відсутність критичних помилок і готовність до публічного розміщення. Сайт було успішно розгорнуто у відкритому доступі з використанням хостингової платформи Netlify, що дозволяє кожному охочому переглянути його роботу в режимі реального часу.

Результати дипломної роботи пройшли апробацію у вигляді тез та доповіді на XLVII міжнародній науковій студентській конференції. Це підтверджує актуальність тематики й практичну цінність створеного продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Черненко О. О. Методичні рекомендації до виконання і оформлення кваліфікаційної роботи для студентів за освітньою програмою «Комп'ютерні науки» / Черненко О. О. - Полтава: ПУЕТ, 2025. - 58 с.
2. Офіційна документація Vite: офіційний гід розробника [Електронний ресурс]. - Режим доступу: <https://vite.dev/guide>
3. Джон Дакетт. HTML и CSS. Розробка і дизайн веб-сайтів. - М.: Ексмо, 2017. - 480 с.
4. Річард Пулін. Школа дизайну: макет. Практичне посібник для студентів і дизайнерів. - Манн, Іванов і Фербер, 2019. - 232 с.
5. React: сучасні шаблони для розробки додатків. 2-ге видання / Е. Порселло, А. Бенкс. - К.: Видавництво "Новий формат", 2020. - 304 с.
6. Ефективний TypeScript: 62 способи поліпшити код / Ден Вандеркам. - К.: Видавництво "Фабула", 2022. - 288 с.
7. JavaScript повне керівництво. 7-е видання / Фленаган Д. - К.: Видавництво "Літера", 2021. - 928 с.
8. Офіційна документація React [Електронний ресурс]. - Режим доступу: <https://uk.legacy.reactjs.org/docs/getting-started.html>
9. Офіційний тьюторіал з React [Електронний ресурс]. - Режим доступу: <https://uk.legacy.reactjs.org/tutorial/tutorial.html>
10. Документація платформи Netlify [Електронний ресурс]. - Режим доступу: <https://docs.netlify.com/get-started>
11. Основи DOM на JavaScript [Електронний ресурс]. - Режим доступу: <https://яваскрипт.укр/DOM>
12. Вікіпедія: Warframe - [Електронний ресурс]. - Режим доступу: <https://uk.wikipedia.org/wiki/Warframe>
13. Відеоурок: Базове налаштування VS Code [Електронний ресурс]. - Режим доступу: https://youtu.be/zA6WNu_HGwY?si=STjdXmzwC2Oo6GyM

ДОДАТОК А

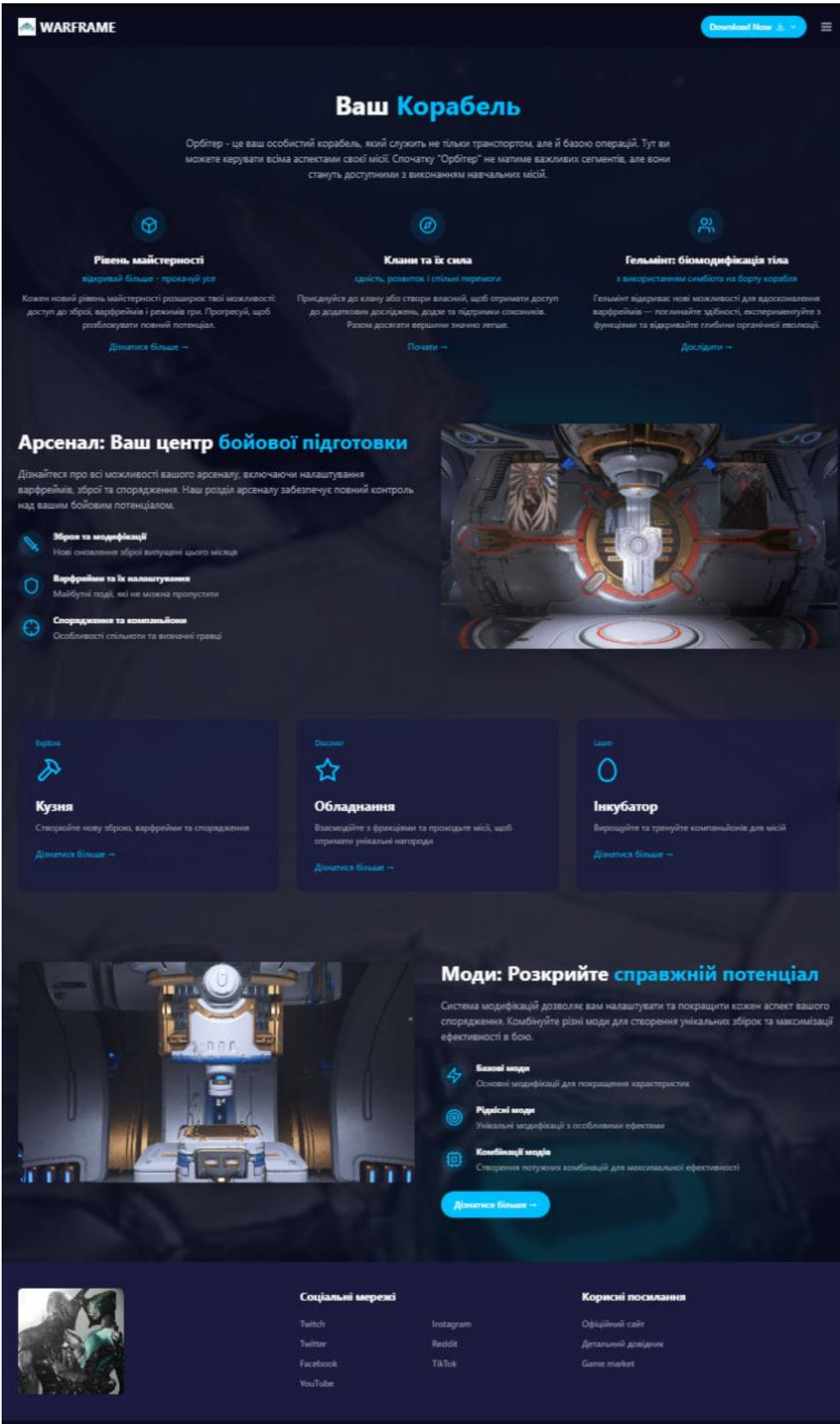


Рисунок А.1 – Сторінка “Корабель гравця”


[Download Now](#)

Основи геймплею

Warframe - це кооперативний шутер від третьої особи, де гравці можуть об'єднуватися в групи (загони) до чотирьох людей, або виконувати цілі завдання самостійно. Звичайні ГПО завдання можна знайти, відкривши зоряну мапу. Також існує ГПГ вміст — завдання Конклаву у Тешина і який доступний як з орбітра, так і на реле.

Існує понад 30 типів завдань. Кожне завдання має певну заздалегідь визначену мету, хоча під час завдання можуть виникати другорядні цілі або змінюватися цілі (за винятком, убивства, оборони, прорив та перехоплення).

Кожен вузол завдання зайнятий принаймні однією протилежною фракцією, але можуть з'явитися дві або більше, які беруть участь у вторгненні або в сутичці.



Ключові особливості гри



Бойова система

Динамічні бої з використанням паркуру та унікальних здібностей



Тактичні елементи

Стратегічне планування та командна взаємодія



Різноманітність місій

Від скритих проникнень до масштабних битв



Розвиток персонажа

Глибока система прокачки та модифікацій



Навігація зоряною мапою

- Якщо навести курсор на вузол, ви отримаєте опис завдання, зокрема:
 - Мапа — очікувана тема середовища, від нього залежить вигляд завдання. Назва тайтлсету не вказується в описі завдання.
 - Назва завдання
 - Тип завдання — ціль або цілі, які необхідно виконати.
 - Фракція — фракція неігрових персонажів, яка зараз займає цей вузол. Кілька фракцій неігрових персонажів, які намагаються зайняти один вузол, розпочнуть перестрілку.
 - Рівень — початковий діапазон рівня ворогів у цьому вузлі. Наприклад, виживання, оборона, завада підйому, видобуток та супровід десертів — це нескінченні завдання, у яких вороги змінюватимуться залежно від часу та/або рейтингу обладнання гравця



Мобільна оборона

Знайти й захистити кілька консолей, поки Лотос зламуватиме ворожу систему





Соціальні мережі

- Twitch
- Twitter
- Facebook
- YouTube
- Instagram
- Reddit
- TikTok

Корисні посилання

- Офіційний сайт
- Детальний довідник
- Game market

Рисунок А.2 – Сторінка “Ігрова система”


[Download Now](#)

Внутрішньоігрові Фракції

У всесвіті Warframe немає постійної боротьби між могутніми фракціями, кожна з яких має власну історію, ідеологію та унікальні бойові особливості. Від технологічного Корпусу, до базових військових армій Гріндерів — усі вони формують складну політичну та військову структуру гри.

Розуміння цілей та тактик цих фракцій допоможе не лише ефективно діяти в Битві, а й глибше зануритися в ігровий світ. Більше місії, особливих поворотів і навіть швидше прогресу пов'язані з простим знанням цих цілей у грі.




Grineer

Мілітаристична імперія класичних воїнів.

Особливості:

- Висока броня та зброя
- Масова координація
- Промислові бази
- Технології масового виробництва

Corpus

Торгова імперія, що спеціалізується на технології.

Особливості:

- Передові технології
- Роботизовані війська
- Торгові майданчики
- Наукові дослідження

Infested

Біологічна зараза, що поглинає все на своєму шляху.

Особливості:

- Швидка мутація
- Колективний розвиток
- Біологічна зброя
- Зарешані території

Синдикати

У світі Warframe, як і в житті, підтримка однієї сторони рідко буває нейтральною. Приєднуючись до певного синдикату, ти автоматично здобуваєш друзів і ворогів серед інших фракцій. Кожен твій рішення впливає на баланс впливу у цьому політичному мікро-світі.

- Щоб зарекомендувати репутацію, достатньо відкрити вибраний синдикат та вирушити у звичайні місії — очікуй нагород за виконання завдань.
- Чим вищий твій рівень майстерності (Mastery Rank) — тим більше репутації можна зарекомендувати.
- Вірність обраній фракції виявляється — це тебе чекають унікальні місії, пригоди, зброя, аксесуари і навіть спеціальні Warframe-позначення.

Але твій вибір допомагає не лише, ти можеш виконати певні місії. У Warframe репутація — це не просто цифра, а реальний вплив на те, куди ти будеш у сліди, а в який — на мислі.



Що залишилось пити?

Ми аплодуємо, Cephalon Suda знає більше! Пірний у глибокій Wild.

[Детальніше про Wild](#)



Соціальний зв'язок

Twitch
Twitter
Facebook
YouTube

Корисні посилання

Офіційний сайт
Детальний довідник
Сторінка новин

Рисунок А.3 – Сторінка “Фракції”

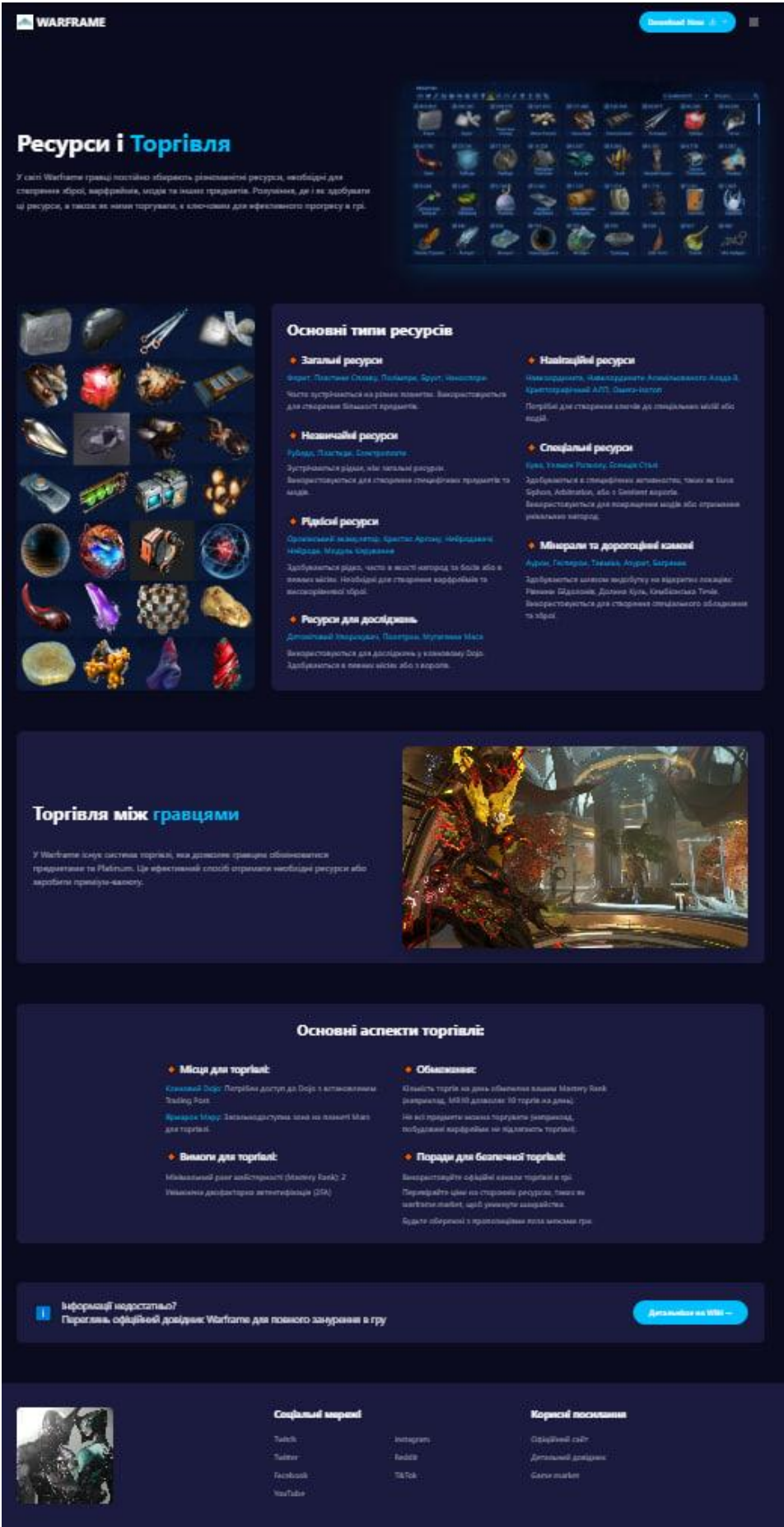


Рисунок А.4 - Сторінка “Ресурси і торгівля”