Міністерство освіти і науки України

Луцький національний технічний університет

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»

РОЗРОБКА ІНТЕРНЕТ-ПОРТАЛУ НА ІГРОВУ ТЕМАТИКУ З ВИКОРИСТАННЯМ WEBPACK І SWIPER

DEVELOPMENT OF A GAMING-THEMED INTERNET PORTAL USING WEBPACK AND SWIPER

спеціальність 121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

освітня програма «Інженерія програмного забезпечення»
(назва освітньої програми)

Виконав: здобувач вищої освіти
Групи ІПЗ-41
Грицюк Владислав Андрійович
_____
(підпис)

Керівник:
к.т.н., доцент
Ящук Андрій Анатолійович
_____
(підпис)

Кваліфікаційну роботу
допущено до захисту
«__8__» ____06____ 2024 р.
к.т.н., доцент
Гарант освітньої програми:
Ліщина Наталія Миколаївна
_____
(підпис)

Луцьк – 2024 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет *комп'ютерних та інформаційних технологій*
Кафедра *інженерії програмного забезпечення*
Ступінь вищої освіти: *бакалавр*
Галузь знань: *12 Інформаційні технології*
Спеціальність: *121 Інженерія програмного забезпечення*
Освітня програма: *«Інженерія програмного забезпечення»*

ЗАТВЕРДЖУЮ
Завідувач кафедри
________ Н. Ліщина
«2» 02 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Грицюку Владиславу Андрійовичу
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: *розробка інтернет-порталу на ігрову тематику з використанням WebPack і Swiper*

Керівник роботи: *к.т.н., доцент, Ящук Андрій Анатолійович*
затверджені наказом закладу вищої освіти від «*30*» *грудня 2023 р.* № *477/01-02*
2. Строк подання здобувачем вищої освіти кваліфікаційної роботи *«5» червня 2024 р.*
3. Вихідні дані до роботи: *технічне та програмне забезпечення ЕОМ, вимоги до розробки програмного забезпечення, ергономічні вимоги до функціонування програмного засобу.*

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): *Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.*

5. Перелік графічного матеріалу: *20 рисунків.*

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	*Яцук А. А.*		
Специфікації вимог до розробленої системи	*Яцук А. А.*		
Розробка об'єкта проєктування	*Яцук А. А.*		
Нормоконтроль	*Повстяна Ю. С.*		
Гарант ОП	*Ліщина Н. М.*		
Показник запозичень тексту	14,6%		
Академічна доброчесність	*Яцук А. А.*		

7. Дата видачі завдання *«2» лютого 2024 р.*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	*Огляд літературних джерел по темі кваліфікаційної роботи бакалавра*	*07.02.2024 р.*	вик.
2	*Аналіз проблеми розробки та впровадження об'єкту проектування*	*27.02.2024 р.*	вик
3	*Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання*	*12.03.2024 р.*	вик.
4	*Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних*	*17.04.2024 р.*	вик.
5	*Практична реалізація об'єкта проектування та розробка бази даних*	*18.05.2024 р.*	вик.
6	*Тестування та налагодження об'єкта проектування*	*01.06.2024 р.*	вик.
7	*Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру*	*05.06.2024 р.*	вик-

Здобувач вищої освіти _____ (підпис) *Грицюк В. А.* (прізвище, ініціали)

Керівник кваліфікаційної роботи _____ (підпис) *Яцук А. А.* (прізвище, ініціали)

Міністерство освіти і науки України
Луцький національний технічний університет

Рецензія

на кваліфікаційну роботу бакалавра

Здобувач вищої освіти:
Грицюк Владислав Андрійович

Тема: розробка інтернет-порталу на ігрову тематику з використанням WebPack і Swiper.

Коротка характеристика кваліфікаційної роботи: було створено інтернет-портал з використанням сучасних технологій, таких як JS, SCSS, WebPack, Swiper. Визначено актуальність поставленої проблеми розробки інтернет-порталів, а саме проблеми високої конкуренції, контенту, локалізації та зміни в технологіях. Також було проаналізовано аналоги уже існуючих інтернет-порталів та визначено їх специфіку, методи і технології розробки.

Самостійні розробки і пропозиції автора: було створено дизайн сайту та розроблено модуль з розділом новин.

Практичне значення роботи: значимість інтернет-порталу це насамперед інформаційний ресурс, тобто він надає актуальну інформацію про гру, новини по ній, огляди, тощо.

Загальний висновок:
Роботу виконано у повному обсязі, у відповідності із завданням. Здобувач освіти Грицюк Владислав Андрійович заслуговує позитивну оцінку.

Рецензент: Саварин Павло Вікторович, к.п.н. доцент.
(прізвище, ім'я, по-батькові, посада)

Рецензент кваліфікаційної роботи _____
(підпис)

«7» червня 2023 р.

Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення

ВІДГУК
КЕРІВНИКА НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Здобувач вищої освіти Грицюк Владислав Андрійович
(прізвище, ім'я, по батькові)

група ІПЗ-41

Тема кваліфікаційної роботи: розробка інтернет-порталу на ігрову тематику з використанням WebPack і Swiper.

Керівник к.т.н., доц. Ящук А. А.

Актуальність теми. У сучасному світі інтернет-ресурси мають важливе значення у багатьох аспектах нашого життя, зокрема у сфері розваг. Ігрова індустрія є однією з найбільш швидкозростаючих та популярних галузей, яка приваблює мільйони користувачів по всьому світу. Розробка інтернет-порталу на ігрову тематику є важливим завданням, яке дозволяє створити зручну платформу для гравців, розробників та ентузіастів, де вони можуть обмінюватися інформацією, обговорювати новинки та підтримувати один одного.

Об'єкт дослідження. Методи і алгоритми сучасних програмних продуктів для розробки ігрових веб-сайтів.

Характеристика теоретичного рівня, наявності самостійних розробок і практичної значимості роботи. Було проведено аналіз предметної області, що включає в себе дослідження ігрової індустрії, аналіз ринку та потреб користувача. Також було вибрано сучасні технології для розробки, такі як HTML5, SCSS, JS, WebPack, Swiper. Що до самостійних розробок, то було створено дизайн сайту та розроблено модуль з розділом новин. Практична значимість інтернет-порталу це насамперед інформаційний ресурс, тобто він надає актуальну інформацію про гру, новини по ній, огляди, тощо.

Загальний висновок

Роботу виконано у повному обсязі у відповідності до поставлених завдань, а здобувач Грицюк В. А. заслуговує на високу оцінку.

Керівник _____ **Ящук А. А.**
(підпис) (прізвище, ім'я, по батькові)

«7» червня 2024 р.

АНОТАЦІЯ

Грицюк В. А. Розробка Розробка інтернет-порталу на ігрову тематику з використанням WebPack і Swiper. Рукопис.

Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2024.

Кваліфікаційна робота бакалавра складається зі вступу, трьох розділів, висновків і пропозицій, списку використаних джерел та додатків.

У першому розділі здійснено аналіз сучасного стану проблеми розробки інтернет-порталів і сформульовано завдання. В другому розділі визначено вимоги до розроблюваної системи, а також засоби, методи і алгоритми розробки. У третьому розділі розроблено інтернет-портал з використанням WebPack і Swiper. У висновках узагальнено інформацію, відображену у попередніх частинах. Результати розробки демонструють можливості JS,SCSS,HTML та таких інструментів як WebPack і Swiper.

Ключові слова: веб-сайт, CSS, SCSS, JS, WebPack, Swiper, HTML.

ABSTRACT

Hrytsyuk V. A. Development of a Game-themed internet portal using WebPack an Swiper. Manuscript.

Bachelor's qualifying thesis of the educational program "Software Engineering". Lutsk National Technical University. Lutsk, 2024.

The bachelor's qualifying thesis consists of an introduction, three sections, conclusions and proposals, a list of used sources and annexes.

In the first chapter, an analysis of the current state of the problem of developing computer games was carried out and the task was formulated. The second section defines the requirements for the developed system, as well as means, methods and development algorithms. In the third chapter, we developed an Internet portal using WebPack and Swiper. The conclusions summarize the information presented in the previous chapters. The development results demonstrate the capabilities of JS, SCSS, HTML and such tools as WebPack and Swiper.

Keywords: website, CSS, SCSS, JS, WebPack, Swiper, HTML.

ЗМІСТ

ВСТУП 7
РОЗДІЛ 1 АНАЛІЗ РОЗРОБКИ ІНТЕРНЕТ-ПОРТАЛУ 9
1.1 Аналіз сучасного стану проблеми 9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра 15
Висновки до розділу 1 16
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ 17
2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання 17
2.2 Функціонально-структурна схема роботи об’єкта проектування 21
Висновки до розділу 2 24
РОЗДІЛ 3 РОЗРОБКА ІНТЕРНЕТ-ПОРТАЛУ ІЗ ВИКОРИСТАННЯМ СУЧАСНИХ ФРЕМВОРКІВ 25
3.1 Практична реалізація веб-сайту 25
3.2 Підключення стилів для сайту 29
3.3 Підключеня JavaScript 31
3.4 Інструкція для користувача 34
Висновки до розділу 3 37
ВИСНОВКИ 38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 39
ДОДАТКИ 41

ВСТУП

Актуальність роботи. У сучасному світі інтернет-ресурси мають важливе значення у багатьох аспектах нашого життя, зокрема у сфері розваг. Ігрова індустрія є однією з найбільш швидкозростаючих та популярних галузей, яка приваблює мільйони користувачів по всьому світу. Розробка інтернет-порталу на ігрову тематику є важливим завданням, яке дозволяє створити зручну платформу для гравців, розробників та ентузіастів, де вони можуть обмінюватися інформацією, обговорювати новинки та підтримувати один одного.

Ігрова індустрія охоплює широкий спектр продуктів та послуг, від мобільних ігор до масштабних багатокористувацьких онлайн-ігор (MMORPG), від віртуальної реальності (VR) до кіберспорту. За останні десятиліття цей сектор зазнав значних змін завдяки інноваціям у технологіях, що дозволило розробникам створювати більш реалістичні та захоплюючі ігрові світи. За даними досліджень, світовий ринок відеоігор у 2023 році перевищив 180 мільярдів доларів США, і цей показник продовжує зростати.

Метою цієї кваліфікаційної роботи є розробка інтернет-порталу на ігрову тематику з використанням сучасних інструментів та технологій, таких як WebPack і Swiper. WebPack [1] є потужним інструментом для збірки модулів JavaScript, що дозволяє ефективно керувати залежностями та оптимізувати продуктивність веб-додатків. Swiper, у свою чергу, є популярною бібліотекою для створення веб-слайдерів, яка забезпечує інтерактивний та динамічний користувацький інтерфейс.

Для досягнення мети необхідно виконати наступні завдання:

– визначити актуальність проблеми та ігрового інтернет-порталу;

– визначити специфіку сучасних ігрових інтернет-порталів;

– проаналізувати та дослідити вже існуючі ігрові інтернет-портали;

– провести детальний аналіз сучасних методів та програмних продуктів для розробки ігрових інтернет-порталів;

– обгрунтувати та провести поетапний детальний аналіз вже існуючих технологій;

– визначити етапи створення, розробки, дизайну та програмної реалізації ігрового інтернет-порталу.

Об’єктом дослідження є методи і алгоритми сучасних програмних продуктів для розробки ігрових веб-сайтів

Предметом дослідження є інтернет-портал створений з використанням інноваційних методів

Результатом роботи стане функціональний інтернет-портал на ігрову тематику, який надасть користувачам зручний та привабливий інтерфейс для взаємодії з контентом, а також продемонструє можливості використання WebPack і Swiper у розробці сучасних веб-додатків.

РОЗДІЛ 1
АНАЛІЗ РОЗРОБКИ ІНТЕРНЕТ-ПОРТАЛУ

1.1 Аналіз сучасного стану проблеми

У наш час практично кожна успішна організація, компанія, або спільнота має власний веб-сайт, що є важливим інструментом для взаємодії з аудиторією. В умовах сучасної конкуренції використання новітніх інформаційних технологій є необхідним фактором існування, що дозволяє значно розширити можливості реклами та залучити нових користувачів. Це особливо актуально для інтернет-порталів на ігрову тематику, де якісний веб-сайт може стати центром геймерської спільноти [2].

Створення та розробка будь-якого інтернет-порталу включає кілька основних етапів:

– технічне завдання;

– структурна схема (розділи, контент, навігація);

– графічні елементи стилів, навігації, макету, елементів порталу;

– інші елементи веб-сайту;

– тестування та розміщення порталу в мережі.

Інтернет-портал (рис. 1.1) – це комплекс програмних засобів, через який можна надавати різноманітні послуги та інформацію про ігри, новини індустрії, огляди ігор, а також забезпечити спілкування між користувачами. Такі портали поєднують елементи інформаційних ресурсів та соціальних мереж, створюючи інтерактивний простір для геймерів.

Рисунок 1.1 – Інтернет-портал [3]

Основними відмінностями інтернет-порталу від традиційних сайтів є інтерактивність, велика кількість інформації, багатий асортимент контенту та персоналізований підхід до кожного відвідувача. Одним з найважливіших інструментів для створення сучасного та ефективного інтернет-порталу є використання WebPack (рис. 1.2) – популярного збирача модулів для JavaScript-додатків. WebPack дозволяє оптимізувати ресурси, забезпечуючи швидке завантаження сторінок та високу продуктивність.

Рисунок 1.2 – Логотип WebPack [4]

Окрім WebPack, при розробці порталу використовуються різні технології, які можна поділити на кілька категорій:

– фронт-енд мови програмування HTML5, CSS3, JavaScript, JavaScript-фреймворки (React, Angular, Vue.js), бібліотеки JavaScript (jQuery, Bootstrap);

– бек-енд мови програмування (PHP, Python, Java, Ruby on Rails), бази даних (MySQL, PostgreSQL, MongoDB), фреймворки для бек-енду (Laravel, Django, Ruby on Rails), API;

– та інші: CMS (WordPress, Drupal), інструменти для керування зображеннями (Cloudinary, Imgur), сервіси аналітики (Google Analytics).

Конкретний набір технологій буде залежати від потреб та особливостей вашого ігрового порталу.

Приклади популярних ігрових порталів:

– IGN (рис. 1.3), який використовує JavaScript, PHP, MySQL, React, Cloudinary;

– GameSpot, який використовує JavaScript, PHP, MySQL, Angular, Imgur;

– PC Gamer, який використовує JavaScript, Python, PostgreSQL, Vue.js, Google Analytics.

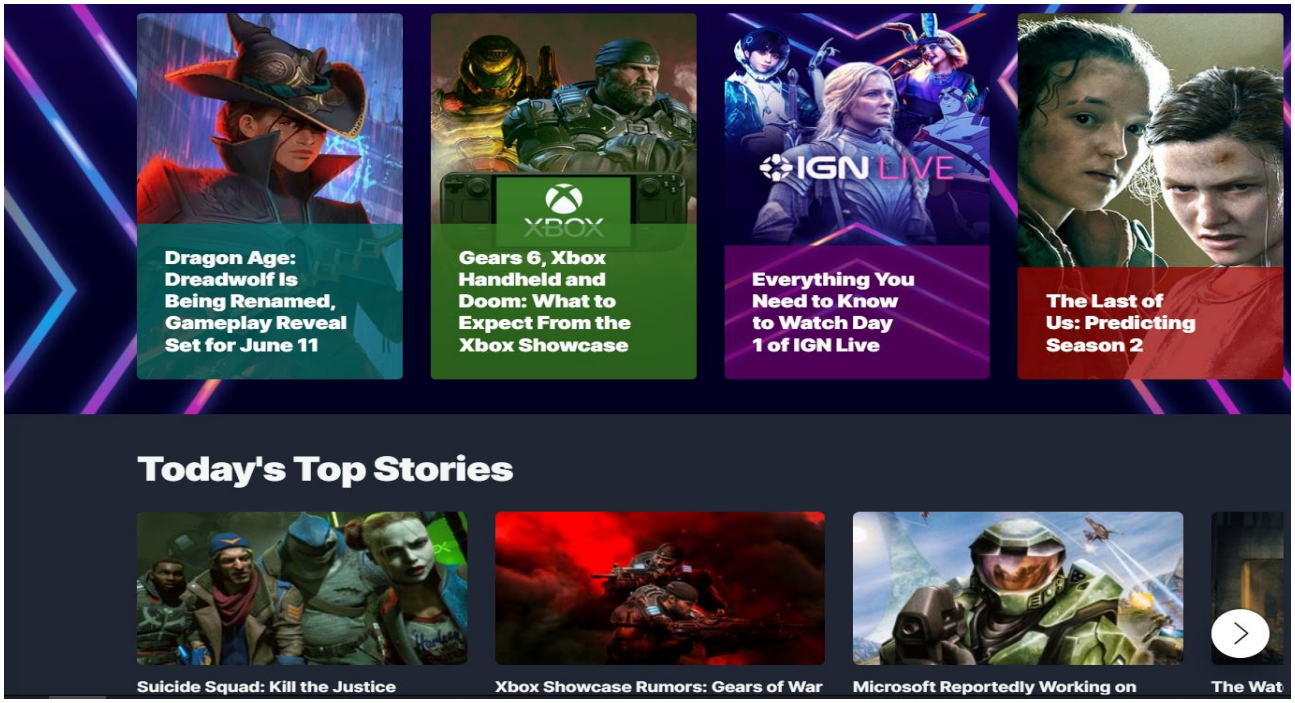

Рисунок 1.3 – Інтернет-портал IGN [5]

Хоча геймери не можуть фізично взаємодіяти з продуктами, що обговорюються на порталі, великий обсяг актуальної інформації, огляди та рекомендації допомагають компенсувати цей недолік, надаючи користувачам можливість отримати детальну технічну інформацію про ігри та їх особливості, що не завжди можна отримати від продавців у традиційних магазинах. Цей підхід дозволяє створити унікальний і цінний ресурс для всіх ентузіастів ігрової індустрії.

React.js (або просто React) (рис. 1.4) – це JavaScript бібліотека для побудови інтерфейсів користувача (UI). Її розробила компанія Facebook (тепер Meta) і вперше випустила у 2013 році. React дозволяє розробникам створювати великі веб-додатки, які можуть оновлюватися і рендеритися без перезавантаження сторінки. Головною особливістю React є його декларативний підхід до програмування інтерфейсів та використання компонентів.

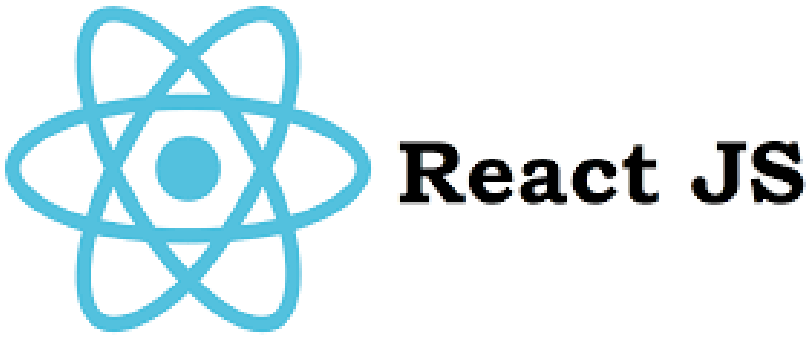

Рисунок 1.4 – Бібліотека React [6]

React.js використовується для розробки динамічних та інтерактивних веб-сторінок. Зокрема, він часто застосовується для:

– односторінкових додатків (Single Page Applications, SPAs) забезпечує швидку навігацію без перезавантаження сторінки;

– розробки інтерфейсів, він дозволяє створювати складні інтерфейси користувача, використовуючи компоненти;

– з React Native також можна створювати мобільні додатки для iOS та Android.

React використовує віртуальний DOM для оптимізації рендерингу, що дозволяє зменшити кількість взаємодій з реальним DOM і підвищити продуктивність.

Завдяки дифінгу (порівнянню) віртуальних DOM, оновлення інтерфейсу відбувається швидше. Також компоненти можна легко повторно використовувати в різних частинах додатку. Кожен компонент є незалежним і може бути легко підтримуваним та тестованим.

Розробники описують, як інтерфейс має виглядати у певному стані, а React самостійно оновлює інтерфейс при зміні стану.

Присутня велика кількість готових рішень, бібліотек та інструментів, а також багата документація та ресурси для навчання і можливість розробки мобільних додатків з використанням того ж підходу та бібліотеки.

Також React має і декілька недоліків:

– react швидко розвивається, і це може вимагати частих оновлень та адаптації коду;

– розробники повинні постійно слідкувати за новими підходами та інструментами;

– для ефективної роботи з React часто потрібно знати інші технології, такі як JSX, Redux, Webpack тощо;

– хоча JSX зручний, для новачків він може здатися складним;

– оскільки React генерує динамічний контент на стороні клієнта, пошукові системи можуть мати проблеми з індексацією таких сторінок. Це можна виправити за допомогою серверного рендерингу (наприклад, Next.js).

MySQL (рис. 1.5) – це система керування базами даних (RDBMS), що використовує SQL (Structured Query Language) як основну мову для взаємодії з ними. MySQL було розроблено компанією MySQL AB у 1995 році і згодом придбано компанією Sun Microsystems, яка пізніше була куплена Oracle Corporation. MySQL є одним із найпопулярніших і найширше використовуваних RDBMS у світі, особливо для веб-додатків.

Рисунок 1.5 – База даних MySQL [7]

MySQL використовується для управління та зберігання даних у різних типах додатків, включаючи:

– багато популярних веб-платформ, таких як WordPress, Drupal, Joomla, використовують MySQL;

– інтернет-магазини часто використовують його для зберігання даних про продукти, замовлення, клієнтів і т.д.;

– він використовується в різних корпоративних програмах для управління даними.

MySQL має досить багато переваг:

– він відомий своєю високою швидкістю обробки запитів, це робить його придатним для додатків, яким потрібний швидкий доступ до даних;

– здатний ефективно обробляти великі обсяги даних і масштабуватися відповідно до потреб бізнесу;

– має відкритий вихідний код і може використовуватися безкоштовно під ліцензією GNU General Public License (GPL);

– велика спільнота розробників та користувачів, яка активно підтримує та розвиває MySQL;

– MySQL підтримує транзакції, що гарантує надійність і цілісність даних;

– може працювати на різних операційних системах, таких як Windows, Linux, macOS;

– підтримує інтеграцію з різними мовами програмування, такими як PHP, Python, Java тощо;

– забезпечує різні механізми аутентифікації та управління доступом до даних.

Але він має також і декілька недоліків:

– проблеми з масштабуванням на вертикальних рівнях, для дуже великих та високонавантажених систем можуть виникати труднощі з масштабуванням і продуктивністю;

– має менше вбудованих аналітичних функцій порівняно з іншими системами, такими як PostgreSQL або Oracle;

– деякі складні SQL-запити можуть працювати повільніше в MySQL або вимагати оптимізації;

– хоча MySQL є відкритим, його комерційне використання може вимагати ліцензування від Oracle.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

У процесі виконання кваліфікаційної роботи ми досліджуємо сучасний стан розвитку області розробки веб-сайту, зокрема, ігрового інтернет-порталу, та проводимо короткий аналіз комп'ютерного програмного середовища, яке використовується для створення веб-сайтів різного типу.

Безперечно, будь-який ігровий інтернет-портал має бути розроблений як веб-сайт, доступний в Інтернеті. Веб-сайт має складатися з взаємозалежних компонентів, функції яких мають бути розділені функціональними обмеженнями. Враховуючи це, ми повинні вибрати інструменти для реалізації нашого ігрового інтернет-порталу, який ми розробимо відповідно до найновіших актуальних тенденцій.

Для досягнення поставленої мети, ми повинні виконати наступні завдання:

– визначити актуальність проблеми та ігрового інтернет-порталу;

– визначити специфіку сучасних ігрових інтернет-порталів;

– проаналізувати та дослідити вже існуючі ігрові інтернет-портали;

– провести детальний та покроковий аналіз сучасних інноваційних методів та програмних продуктів для розробки ігрових інтернет-порталів;

– обгрунтувати та провести поетапний детальний аналіз вже існуючих технологій, програмного забезпечення, вибрати раціонально-оптимальний спосіб реалізації ігрового інтернет-порталу відповідно до вищезазначених вимог;

– визначити етапи створення, розробки, дизайну та програмної реалізації ігрового інтернет-порталу.

Таким чином, однією з необхідних умов для якісної та ефективної реалізації даного дипломного проекту та всіх відповідних до нього поставлених вимог є наявність технічного завдання.

Висновки до розділу 1

Було здійснено аналіз існуючих ігрових порталів, а також вивчено сучасні технології та інструменти веб-розробки. На основі проведеного дослідження було зроблено висновок, що для створення ефективного та сучасного ігрового порталу рекомендується використовувати JavaScript, SCSS, WebPack та Swiper. Використання WebPack та Swiper при розробці ігрового порталу дозволить створити сучасний, ефективний та зручний для користувачів веб-сайт, який буде центром тяжіння для геймерської спільноти.

РОЗДІЛ 2
СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Обґрунтування вибору шляхів, технологій (алгоритмів) і засобів вирішення поставленого завдання

Для створення веб-сайту мною було використано середовище розробки Visual Studio code (рис. 2.1) і не просто так, адже воно має численні переваги.

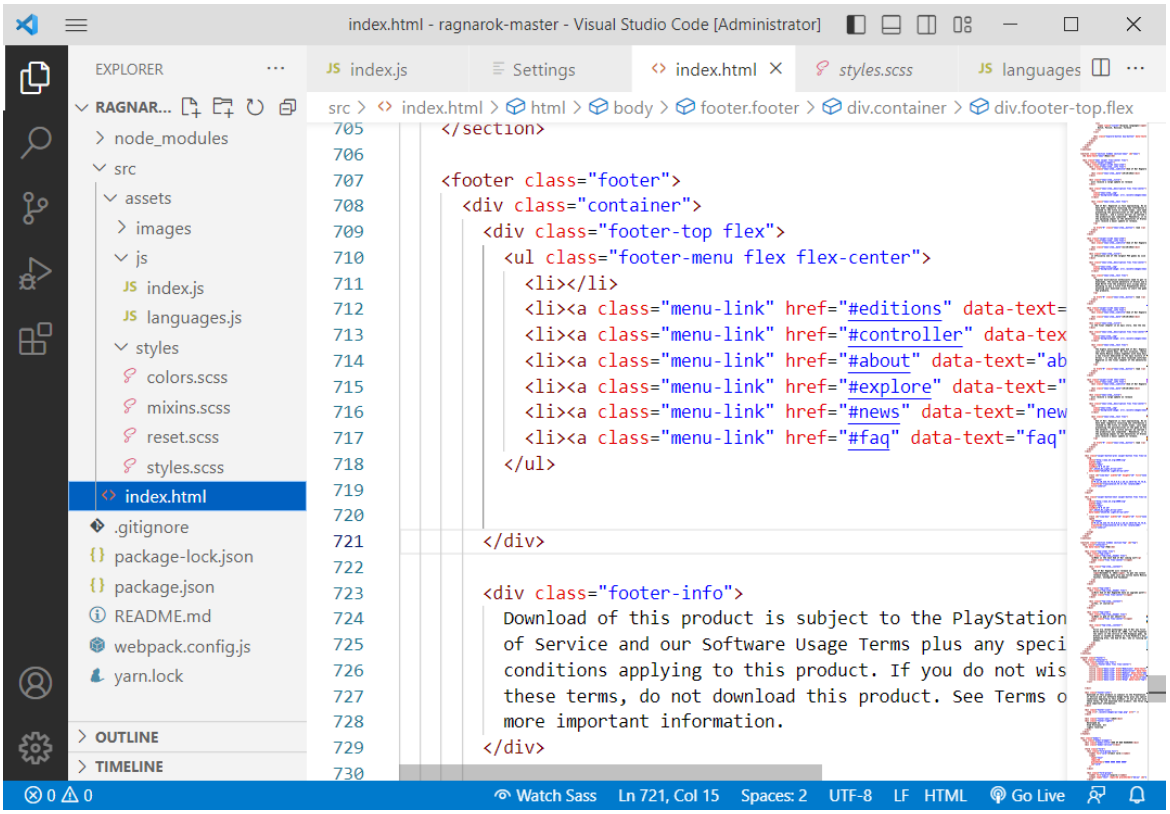

Рисунок 2.1 – Середовище розробки VS code

VS Code є безкоштовним програмним забезпеченням з відкритим кодом, що робить його доступним для розробників з будь-яким бюджетом. Це особливо важливо для студентів та початківців, які не мають коштів на придбання ліцензійних програм.

Він має велику екосистему плагінів, які дозволяють розширювати його функціональність та налаштовувати його відповідно до ваших потреб. Це робить його не лише універсальним інструментом, але й дає можливість адаптувати його до конкретних проектів та завдань.

VisualStudio Code підтримує широкий спектр мов програмування, включаючи JavaScript, HTML, SCSS та інші. Це робить його зручним інструментом для веб-розробників, які можуть використовувати його для роботи з різними частинами проекту, не перемикаючись на інші програми.

Також він має вбудовану підтримку Git, що робить роботу з системами контролю версій зручною та ефективною. Це важливо для командної розробки, оскільки дозволяє синхронізувати код та відстежувати зміни, внесені різними учасниками проекту.

Крім цього у VS Code простий та зрозумілий інтерфейс для користувача, що робить його легким для новачків. Це дозволяє швидко освоїти програму та приступити до розробки, не витрачаючи багато часу на вивчення її складних функцій.

Окрім цих переваг, він також пропонує такі можливості:

– підсвічування синтаксису та автозаповнення коду, що полегшує написання та читання коду;

– віддалений доступ до проектів, що дозволяє працювати над ними з будь-якого місця;

– налагодження коду, що дає можливість виявляти та виправляти помилки;

– інтеграція з іншими інструментами розробки, такими як linters та formatters.

Для фронт-енду було вибрано HTML, CSS, SCSS та JS (рис. 2.2). HTML – це універсальна мова розмітки, яка підтримується усіма браузерами. Її елементи є головними блоками у всіх сторінок. За допомогою конструкцій HTML, зображення та інших об’єкти, можуть бути вбудовані у сторінку. Він надає засоби для створення документів, виділяючи структурну семантику тексту, таку як абзаци, посилання, заголовки, цитати та інші. Елементи HTML мають теги, які пишуться з кутовими дужками, це теги на подобі <img /> чи <input /> , що виводять вміст на сторінку. Також є інші теги, такі як <p>, що надають інформацію про текст, та можуть включати інші теги, піделементи. Браузери не

можуть показати теги HTML, але можуть використовувати їх для для того, щоб описати вміст сторінки.

Рисунок 2.2 – Логотип JS, HTML, CSS [8]

JavaScript – це мова програмування, за допомогою якої можна створювати динамічні веб-сторінки, оновлювати окремі блоки сторінки, діалогові карти, анімувати графіку, прокручувати відео в медіапрогравачі і так далі.

JS є мовою, що інтерпретується. Це означає, що для коректної роботи їй потрібен інтерпретатор [9]. У теперішній час дуже важко уявити інноваційний сайт без участі JavaScript, бо коли у компанії є сайт, то їй так чи інакше потрібен фахівець з цієї мови програмування. Вона є обов'язковою для більшості вакансій у веб сфері. Коли з'явився Node.js, то стало можливим розробляти на JavaScript не лише клієнтську частину, але й серверну. Через це розробники зараз мають змогу писати проект всього лиш на одній мові, що є величезним плюсом. Використовуючи різні інструменти, такі як Electron, React Native і так далі, вона дозволяє створювати програми для комп'ютерів, мобільні додатки, веб-додатки і т.д.

JavaScript має багато переваг, таких як:

– код виконується безпосередньо у браузері, що забезпечує швидкий відгук на дії користувача;

– обробка частини даних на стороні клієнта знижує навантаження на сервер;

– підтримка усіма браузерами, JavaScript підтримується всіма сучасними браузерами без необхідності додаткових плагінів;

– може використовуватися як на клієнтській, так і на серверній стороні;

– має велику кількість бібліотек (наприклад, jQuery) і фреймворків (наприклад, React, Angular, Vue.js), які спрощують розробку;

– пакети та модулі npm (Node Package Manager) забезпечує доступ до безлічі пакетів для різних задач;

– забезпечує можливість створювати складні, інтерактивні та реальні інтерфейси користувача;

– постійні оновлення та вдосконалення, мова постійно розвивається, що додає нові можливості та підвищує продуктивність.

Вибір цих технологій ґрунтується на наступних міркуваннях, відповідно до мети поставленого завдання: HTML, JavaScript, SCSS та Swiper (рис. 2.3) є добре підібраними технологіями для створення інтернет-порталу. HTML використовується для структури та контенту, JavaScript – для інтерактивності, SCSS – для стилізації, а Swiper – для слайд-шоу та каруселей. Ці технології є сучасними та широко використовуються у веб-розробці. Це робить їх надійними та гарантує, що ви зможете знайти багато інформації та ресурсів для їх вивчення та використання. HTML, JavaScript, SCSS та Swiper мають порівняно простий синтаксис та логіку роботи, що робить їх доступними для вивчення навіть для початківців. Ці технології є гнучкими та масштабованими, що дозволяє створювати веб-сайти будь-якої складності та розміру [10].

Рисунок 2.3 – Swiper [11]

Було створено діаграму прецедентів, яка показує взаємодію користувача з сайтом (рис. 2.4).

Рисунок 2.4 – UML-діаграма прецедентів

2.2 Функціонально-структурна схема роботи об’єкта проектування

Першочергово, потрібно зрозуміти, що таке структура сайту – це спосіб організації сторінок веб-сайту, включаючи підкатегорії, категорії, товари, або, іншими словами, це своєрідний план, який чітко відображає взаємозв’язок між сторінками [12].

Структури веб-сайту можуть бути:

– зовнішніми (макет сторінки з вказівкою на розташування блоків);

– внутрішніми (категорії, належність окремих сторінок і матеріалів до них);

Існують такі типи структур:

– алфавітна організація (інформація розташована в алфавітному порядку);

– хронологічний (використовується для розміщення новин, прес-релізів);

– деревоподібні (різні категорії виходять з головної сторінки);

– тематичний (інформація впорядкована за темами);

– лінійний (інформація розташована у вигляді ланцюга з компонентами);

– картатий (компоненти розташовані в окремих гілках);

– павутина (комбінація кількох типів);

– гібридний (комбінація кількох типів структур);

– ієрархічна (допомагає користувачам швидко знаходити товар, який їх цікавить).

Кожен тип структури має свої переваги та недоліки, тому найкращим вибором є гібридний тип структур, який є комбінацією багатьох найкращих типів. Це допоможе досягти найкращого результату [13].

Створення функціональної схеми – це тривалий і складний процес, який включає інформацію про способи реалізації заданих функцій. Схема визначає, як відбуваються перетворення, які функціональні елементи потрібні для цього, і містить лише ті входи і виходи, які необхідні для правильної роботи.

Вона розробляється на основі структурної схеми для кожного блоку, і з окремих функціональних елементів складається загальна функціональна схема об’єкту [14].

Функціональні схеми можуть бути представлені у таких формах:

– укрупнені (відображають найважливіші блоки системи, зв’язки між ними);

– розгорнуті (відображають більш детально, що полегшує їх читання, ілюструє принцип роботи).

Веб-сайт (інтернет-портал) складається з структурних розділів, що відповідають за інформацію, котра призначена майбутнім користувачам різного контингенту (рис. 2.5). Користувачі також можуть швидко переходити на потрібні їм розділи та переглянути час до релізу.

Рисунок 2.5 – Меню сайту

Розділ «Видання» призначений для надання інформації про версії гри які можна придбати, та про перевагу кожної з них.

Розділ «Джойстик» містить інформацію про ексклюзивний джойстик, який можна отримати при купівлі гри для PS5.

Розділ «Про гру» призначений для того, щоб дізнатися більше про гру, також в цьому розділі буде прикріплений трейлер, який можна одразу ж переглянути.

Розділ «Дослідити» містить інформацію про системні вимоги гри для PC та PS,в ньому також буде використаний Swiper для анімації показу мінімальних та рекомендованих вимог.

Розділ «Новини» призначений для того, щоб відвідувачі сайти змогли переглянути останні новини про гру, все це буде виконано також за допомогою Swiper для анімації скролу новин.

Розділ «Допомога» містить інформацію з найбільш поширеними питаннями про гру і відповіддю на них.

Також в меню буде змога зміни української на англійську при наведенні на стрілку біля назви поточної мови (рис. 2.6).

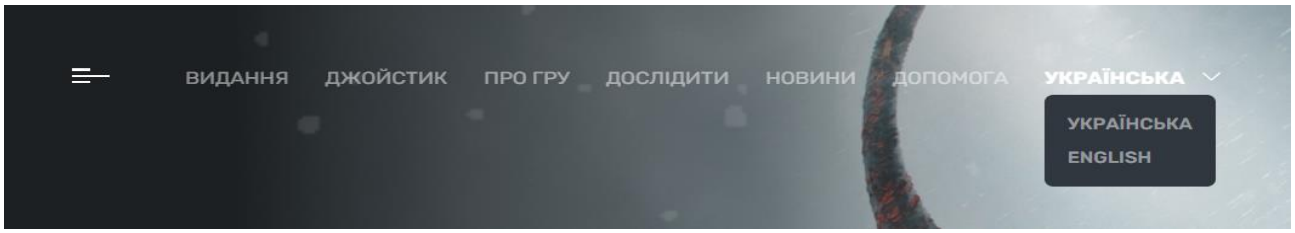

Рисунок 2.6 – Меню зміни мови

Що ж, схему роботи майбутнього веб-сайту можна зобразити так, як це показано на рисунку 2.7.

Рисунок 2.7 – Схема сайту

Отже, структура веб-сайту більш ніж зрозуміла та чітко структуризована, всі розділи мають достаньо-повну інформацію про головну сторінку, меню та всі інші елементи.

Висновки до розділу 2

В даному розділі, ми розглянули та проаналізували технології створення веб-сайту (інтернет-порталу). Після детального розгляду середовища для написання коду, мов програмування, їх складових, можливостей, переваг доведено, чому саме було вибрано Visual Studio, Webpack, Swiper, JS, SCSS та визначено майбутній дизайн-макет веб-сайту.

РОЗДІЛ 3
РОЗРОБКА ІНТЕРНЕТ-ПОРТАЛУ ІЗ ВИКОРИСТАННЯМ СУЧАСНИХ ФРЕМВОРКІВ

3.1 Практична реалізація веб-сайту

Першим кроком у процесі створення та розробки нашого сайту є відкриття консолі у нашому редакторі коду та створення кореневої папки за допомогою команди npm init.

Далі ми встановлюємо пакети для нашого WebPack [15]. Після чого створимо файл index.html і напишемо основний код для нашого сайту (ліст. 3.1).

Лістинг 3.1 – Початок написання коду

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
/>

    <link rel="preconnect" href="https://fonts.googleapis.com" />
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin
/>
    <link

href="https://fonts.googleapis.com/css2?family=Rubik:wght@300;400;500;600
;700&display=swap"
      rel="stylesheet"
    />
    <link
      rel="apple-touch-icon"
      sizes="180x180"
      href="/assets/images/favicon/apple-touch-icon.png"
    />
    <link
      rel="icon"
      type="image/png"
      sizes="32x32"
    />
```

Кінець лістингу 3.1

Одразу ж підключаємо наші шрифти та іконки і пишемо основну структуру нашого сайту (рис. 3.1).

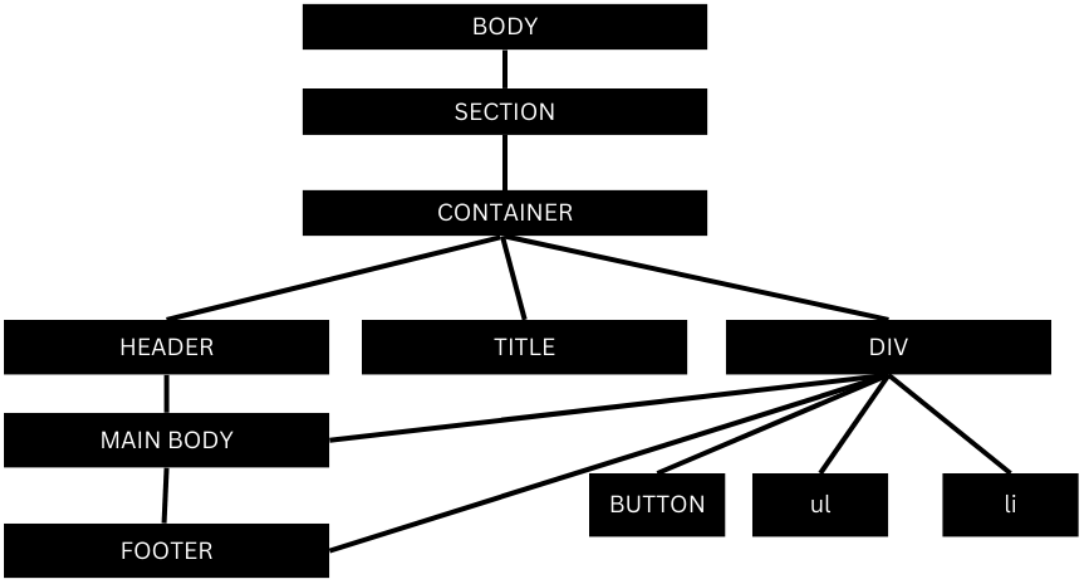

Рисунок 3.1 – Структура html

Переваги написання HTML коду через тег <section> включають:

– використання <section> допомагає упорядковувати вміст сторінки за логічними розділами, що полегшує розуміння коду.

– відображення конкретного розділу сторінки за допомогою <section> надає більшу семантичну зрозумілість коду, що корисно для пошукових систем та адаптивних технологій.

– використання <section> допомагає удосконалювати доступність веб-сайту для користувачів з обмеженими можливостями, так як це надає додатковий контекст та структуру.

– дозволяє легко визначати та стилізувати різні розділи сторінки за допомогою CSS, а також обробляти їх окремо за допомогою JavaScript.

– сприяє локалізації та підтримці мови, оскільки цей тег дозволяє створювати структуровані блоки для різних мовних версій веб-сайту.

Спочатку створюємо верхній блок нашого сайту, тобто меню (ліст. 3.2).

Лістинг 3.2 – Меню сайту

```
<section class="section section-main">
     <div class="container">
       <header class="header flex">
         <div class="header-menu flex">
           <div class="header-menu__button">
             <span></span>
           </div>

           <nav class="header-menu__list">
             <ul class="flex flex-center">
               <li>
                 <a
                   data-text="editions"
                   class="header-menu__link menu-link"
                   href="#editions"
                 >
                   editions
                 </a>
               </li>
               <li>
                 <a
                   data-text="controller"
                   class="header-menu__link menu-link"
                   href="#controller"
                   >controller</a
                 >
               </li>
```

Кінець лістингу 3.2

Після чого почнемо створювати блоки title, editions, controller, about, explore, new, faq. Знизу можна буде побачити створення блоків на прикладі блоку title та editions (ліст. 3.3).

Лістинг 3.3 – Блок title

```
<div class="description">
          <div class="description-title">
            <h1>God of War Ragnarök</h1>
            <div class="description-subtitle" data-text="title1">
              Taking place three years following the events of the
previous game, Fimbulwinter, a great winter that spans three summers, is
drawing to a close which will begin the prophesied Ragnarök.
            </div>
```

```
            <div class="description-button buy-button" data-value="0"
data-text="button">
              buy now
            </div>
          </div>
        </div>
      </div>
    </section>

    <section class="section hidden section-editions" id="editions">
      <div class="container">
        <div class="editions">
          <h2 data-text="editions">Editions</h2>
```

Кінець лістингу 3.3

Коли ми закінчили написання коду для основної частини нашого сайту, можна перейти до нижньої частини, або ж footer (ліст. 3.4).

Лістинг 3.4 – Блок footer

```
<footer class="footer">
      <div class="container">
        <div class="footer-top flex">
          <ul class="footer-menu flex flex-center">
            <li></li>
            <li><a class="menu-link" href="#editions" data-
text="editions">editions</a></li>
            <li><a class="menu-link" href="#controller" data-
text="controller">controller</a></li>
            <li><a class="menu-link" href="#about" data-
text="about">ABOUT GAME</a></li>
            <li><a class="menu-link" href="#explore" data-
text="explore">Explore</a></li>
            <li><a class="menu-link" href="#news" data-
text="news">news</a></li>
            <li><a class="menu-link" href="#faq" data-
text="faq">FAQ</a></li>
          </ul>
        </div>
```

Кінець лістингу 3.4

3.2 Підключення стилів для сайту

Наступним етапом розробки нашого сайту є створення файлів з стилями, адже без них він не буде мати ніякого вигляду. Створюємо такі файли: styles.scss, reset.scss, mixins.scss, colors.scss.

У файлі colors.scss будуть зберігатися основні кольори для інтернет-порталу (ліст. 3.5).

Лістинг 3.5 – Налаштування базових кольорів

```
$main: #1c2023;
$dark: #30373f;
$violet: #8856d9;
$sea: #a9c7c7;
$grey: #8c96a0;
$orange: #ff4136;
$white: #fff;
$black: #000;
```

Кінець лістингу 3.5

У файлі reset.scss я вказую всі загальні стилі тексту та відступи для всього сайту, а також анулюю деякі стандартні стилі (ліст. 3.6).

Лістинг 3.6 – Налаштування стилів

```
body {
  background: $main;
  font-family: "Rubik", Roboto, sans-serif;
  font-size: 16px;
  padding: 0;
  margin: 0;
  overflow-x: hidden;
  color: $white;
}

h1,
h2,
h3,
h4,
h5,
h6,
p {
```

```
  padding: 0;
  margin: 0;
  font-size: 16px;
  line-height: 24px;
}

* {
  box-sizing: border-box;
}
```

Кінець лістингу 3.6

У файлі styles.scss (ліст. 3.7) буде записаний головний код для всіх стилів нашого сайту, починаючи від відступів кожного елемента і закінчуючи їх анімацією.

Лістинг 3.7 – Стиль для деяких елементів

```
&-main {
    padding-top: 40px;
    position: relative;
    max-width: 1920px;
    margin: 0 auto;
    min-height: 700px;

    &:after,
    &:before {
      height: 764px;
      width: 1363px;
      content: "";
      display: block;
      position: absolute;
      top: 0;
      right: 0;
      z-index: -1;
    }

    &:after {
      background: url(../images/snow.gif) no-repeat center/cover;
      opacity: 0.1;
    }

    &:before {
      background: url(../images/bg.png) no-repeat center/cover;
    }
  }
```

Кінець лістингу 3.7

CSS є важливим елементом коду, оскільки він відповідає за стилізацію та вигляд веб-сторінок. Використання CSS дозволяє розміщувати та оформлювати різні елементи веб-сторінки, забезпечуючи їм зовнішній вигляд, розміщення на сторінці, анімацію та інші важливі аспекти.

Це допомагає покращити користувацький досвід, зробити сторінку більш привабливою та функціональною, а також забезпечити її сумісність з різними пристроями та розмірами екранів.

3.3 Підключеня JavaScript

Останнім етапом в створенні сайту є надання змогу користувачу контактувати з різними елементами які ми створили, а для цього потрібно підключити JS. Отже створюємо файл index.js та підключаємо Swiper (ліст. 3.8).

Лістинг 3.8 – Початок коду JS

```
import "swiper/swiper.min.css";
import "../styles/reset.scss";
import "../styles/styles.scss";

import Swiper, { Navigation } from "swiper";
import { languages } from "./languages";
Swiper.use([Navigation]);

const checkboxes = {
  requirements: ["minimum", "recommended"],
  versions: ["standard", "limited"],
};
let isPlay = false;
const classes = {
  opened: "opened",
  hidden: "hidden",
  active: "active",
};
```

Кінець лістингу 3.8

JavaScript є важливим елементом веб-розробки з кількох причин. Він дозволяє створювати інтерактивні веб-сторінки, забезпечує можливість

взаємодії користувачів з веб-сайтом, а також додає динамічність до веб-сторінок. JavaScript використовується для валідації форм, анімації, роботи з асинхронними запитами до сервера (AJAX), розробки ігор та багато іншого. Він є важливим елементом для створення сучасних та функціональних веб-додатків.

Створюємо функцію для «випадання» меню (ліст. 3.9):

Лістинг 3.9 – Функція для роботи меню

```
const toggleMenu = () => header.classList.toggle(classes.opened);
const scrollToSection = (e) => {
  e.preventDefault();
  const href = e.currentTarget.getAttribute("href");

  if (!href && !href.startsWith("#")) return;

  const section = href.slice(1);
  const top = document.getElementById(section)?.offsetTop || 0;
  window.scrollTo({ top, behavior: "smooth" });
};
```

Кінець лістингу 3.9

Наступним кроком потрібно створити функцію для роботи таймера, який буде показувати час виходу гри з точністю до секунд (ліст. 3.10).

Лістинг 3.10 – Функція таймеру

```
const formatValue = (value) => (value < 10 ? `0${value}` : value);

const getTimerValues = (diff) => ({
  seconds: (diff / 1000) % 60,
  minutes: (diff / (1000 * 60)) % 60,
  hours: (diff / (1000 * 3600)) % 24,
  days: (diff / (1000 * 3600 * 24)) % 30,
});

const setTimerValues = (values) => {
  Object.entries(values).forEach(([key, value]) => {
    const timerValue = document.getElementById(key);
    timerValue.innerText = formatValue(Math.floor(value));
  });
};
const startTimer = (date) => {
```

```
  const id = setInterval(() => {
    const diff = new Date(date).getTime() - new Date().getTime();

    if (diff < 0) {
      clearInterval(id);
      return;
    }

    setTimerValues(getTimerValues(diff));
  }, 1000);
};
```

Кінець лістингу 3.10

Також на сайті є змога вибрати англійську або українську мову, щоб це реалізувати створимо функцію toggleLanguage (ліст. 3.11) і файл languages.js у якому ми будемо зберігати наші змінні.

Лістинг 3.11 – Функція зміни мови

```
const setTexts = () => {
  const lang = localStorage.getItem("lang") || "en";
  const content = languages[lang];

  Object.entries(content).forEach(([key, value]) => {
    const items = document.querySelectorAll(`[data-text="${key}"]`);
    items.forEach((item) => (item.innerText = value));
  });
};

const toggleLanguage = ({ target }) => {
  const { lang } = target.dataset;

  if (!lang) return;

  localStorage.setItem("lang", lang);
  setTexts();
};
```

Кінець лістингу 3.11

За допомогою JavaScript створюємо і інші функції, які будуть відповідати за анімацію слайдів, появу меню покупки при нажаті на кнопку, включення трейлеру та інші, що представлені у додатку А.

3.4 Інструкція для користувача

Після написання всього коду давайте запустимо наш сайт, та глянемо на його зовнішній вигляд. Відкривши інтернет-портал одразу бачимо меню, таймер з релізом гри та текст з назвою і невеликим описом (рис. 3.2).

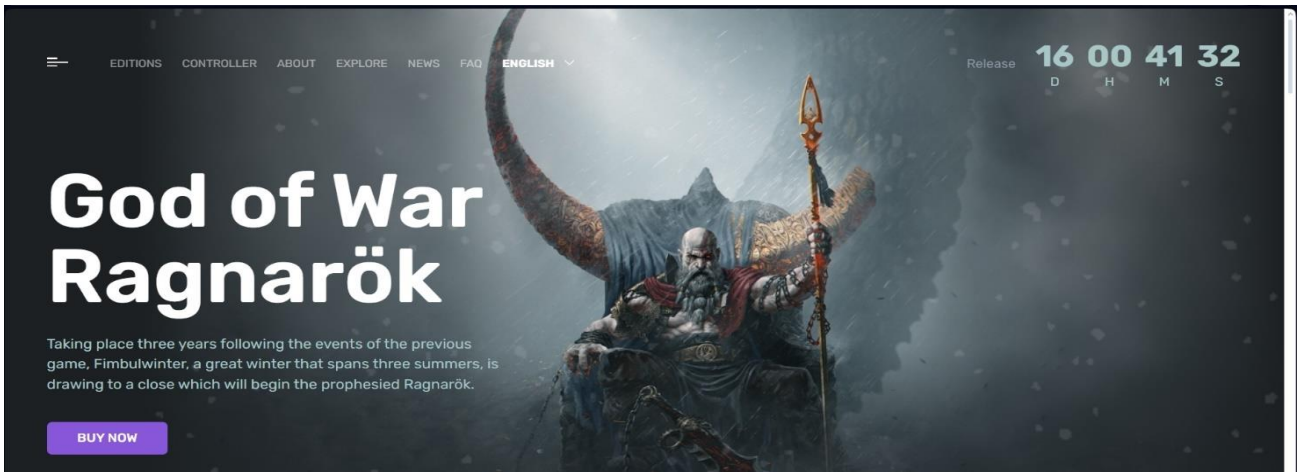

Рисунок 3.2 – Головна сторінка

Прогортавши вниз можна побачити блок видання, у якому вказано тип видання, система для котрої воно доступне гру, ціну та різні переваги, наприклад при купівлі Digital Deluxe Edition у подарунок ви отримаєте також оригінальний звуковий супровід до гри (рис. 3.3).

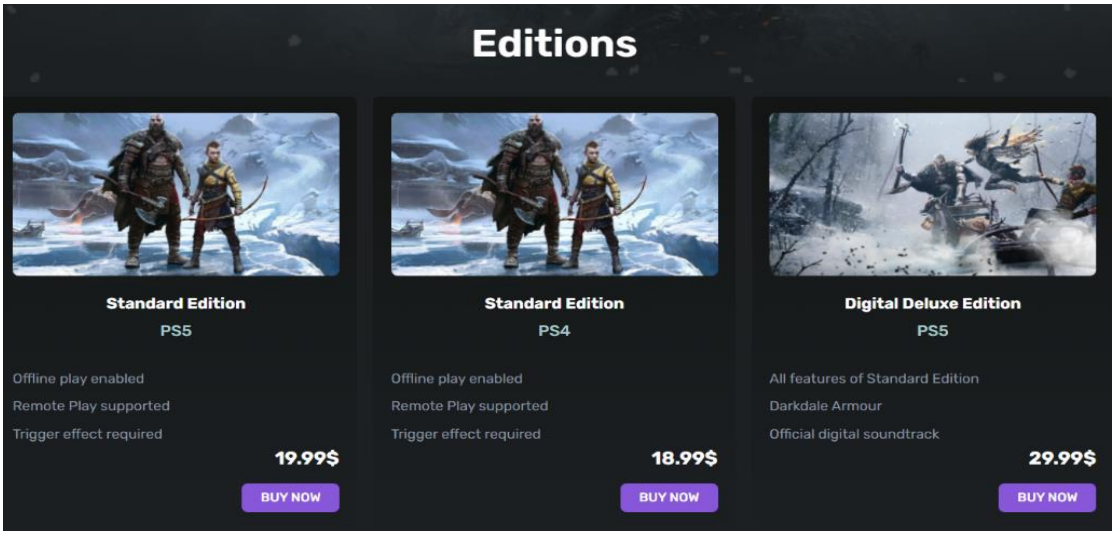

Рисунок 3.3 – Блок видання

Далі зображений блок з інформацією про ексклюзивний джойстик, який можна отримати для PS5. Також у цьому блоці присутня анімація зображення (рис. 3.4).

Рисунок 3.4 – Блок DualSense

Одразу ж після цього можна побачити блок у якому користувачі мають змогу переглянути трейлер до гри (рис. 3.5).

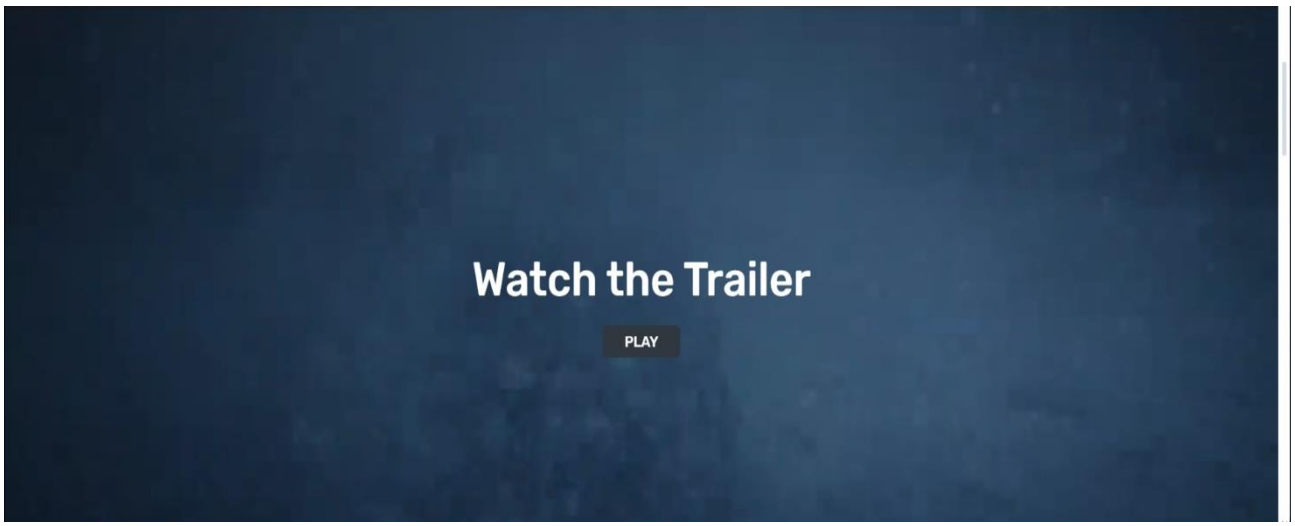

Рисунок 3.5 – Блок з трейлером

Прогорнувши вниз ми можемо побачити інформацію про мінімальні та рекомендовані системні вимоги до гри, також видно використання Swiper при натисканні на кнопку і його анімацію (рис. 3.6).

У цьому блоці вказані вимоги як для PS, так і для PC. Також є змога викликати вікно у якому потрібно ввести данні для здійснення покупки.

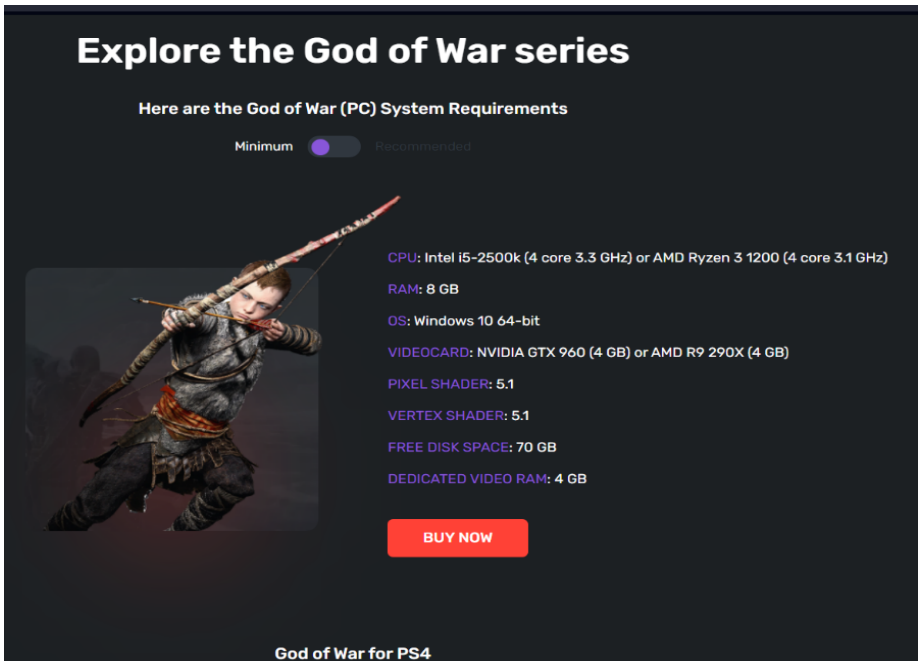

Рисунок 3.6 – Системні вимоги

Далі зображений блок з новинами про гру, у якому теж використовується Swiper для створення анімації прогортування (рис. 3.7).

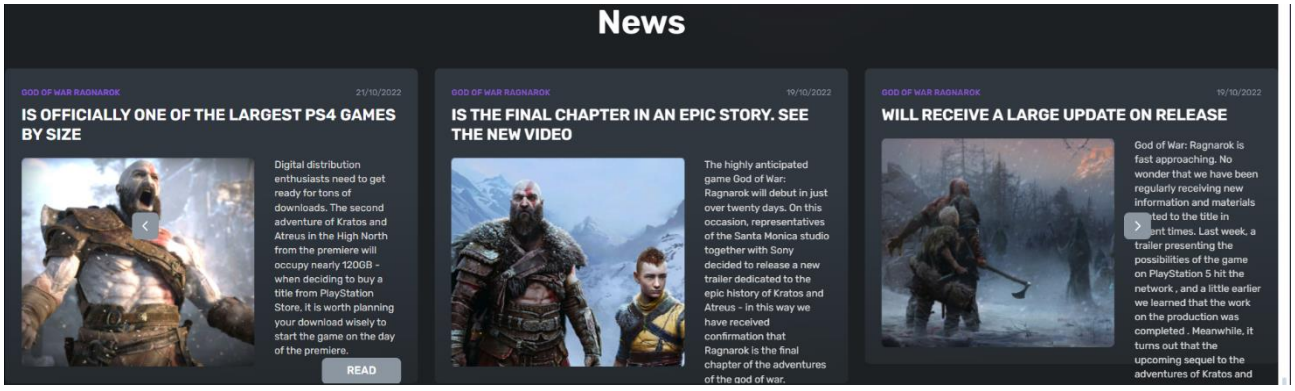

Рисунок 3.7 – Блок з новинами

І останнім блоком сайту є footer, блок у якому вказані відповіді на найпоширеніші питання та розташоване меню навігації (рис. 3.8). Він також був створений за допомогою Swiper.

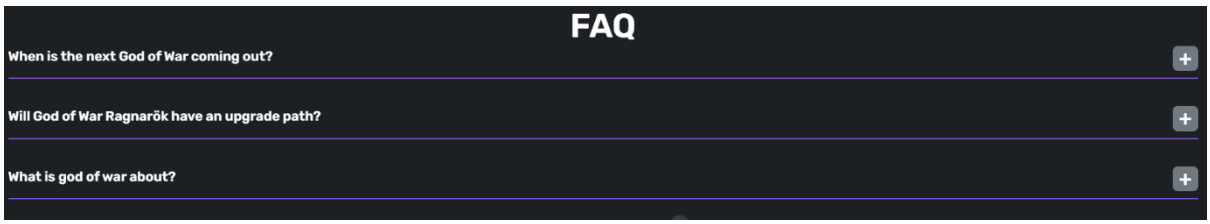

Рисунок 3.8 – Footer

Висновки до розділу 3

Підсумовуючи проведену роботу у розділі 3 зроблено: поетапний опис, побудова веб-сайту, підключення стилів та JS, проведено налагодження роботи веб-сайту, тестування усіх рівнів.

ВИСНОВКИ

У ході виконання даної кваліфікаційної роботи бакалавра, було розроблено інтернет-портал з використанням WebPack та Swiper для ігрової тематики, з можливістю публікації новин, оглядів та іншого контенту про гру.

В ході виконання кваліфікаційної роботи було виконано:

– визначено актуальність поставленої проблеми розробки інтернет-порталів на ігрову тематику у сучасному світі, а саме проблеми високої конкуренції, контенту, локалізації та зміни в технологіях;

– визначено специфіку сучасних веб-сайтів, методи та технології їх розробки, такі як хронологічний, тематичний, лінійний, гібридний і т.д.;

– проаналізовано аналоги уже існуючих інтернет-порталів IGN, GameSpot та PC Gamer;

– обґрунтовано та проведено поетапний детальний аналіз уже існуючих технологій, програмного забезпечення JavaScript, MySQL, React, WebPack, Swiper і обрано раціонально-оптимальний спосіб реалізації веб-сайту (інтернет-порталу) відповідно до зазначених вище вимог;

– детально описано усі етапи створення, розробки, дизайну, програмну реалізацію веб-сайту (інтернет-порталу);

Отже, наразі інтернет-портал є досить функціональним та адаптивним, у зв’язку з чим відвідувачі веб-сайту у будь-який час можуть ознайомитися з новинами, оглядами та іншим контентом про гру God of War.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Офіційний сайт WebPack. URL: https://webpack.js.org (дата звернення: 20.02.2024).

2. Стаття про інтернет-портали та їх зручність і переваги використання. URL: https://goldwebsolutions.com/uk/blog/shho-take-internet-portal-ta-v-chomy-i (дата звернення: 03.03.2024).

3. Що таке інтернет-портал. URL: https://www.castcom.ru/publications/web/chto-takoe-internet-portal.html (дата звернення: 05.03.2024).

4. Webpack – Wikipedia. URL: https://uk.wikipedia.org/wiki/Webpack (дата звернення: 06.03.2024).

5. IGN: Video game news. URL: https://www.ign.com (дата звернення: 07.03.2024).

6. Основи роботи з React. URL: http://www.itshop.ru/Osnovy-raboty-s-bibliotekoy-ReactJS/l4t3i413157 (дата звернення: 08.03.2024).

7. MYSQL. URL: https://servergate.ru/articles/mysql-preimushchestva-i-nedostatki/ (дата звернення: 12.03.2024).

8. HTML5 CSS3 JS icon set. URL: http://surl.li/ujuow (дата звернення: 15.03.2024).

9. Документація по встановленню та використанню Swiper. URL: https://refine.dev/blog/swiper-js/#introductio (дата звернення: 25.03.2024).

10. Using WebPack with TypeScript. URL: https://blog.logrocket.com/using-webpack-typescript/ (дата звернення: 08.04.2024).

11. What is Swiper and how does it work. URL: https://medium.com/@alicienes99/what-is-swiper-and-how-does-it-work-e729645ed5e8 (дата звернення: 12.04.2024).

12. Документація Jest про тестування. URL: https://jestjs.io/docs/webpack (дата звернення: 17.04.2024).

13. 11 types of web portals. URL: https://www.intelivita.com/blog/types-of-web-portals/ (дата звернення: 02.05.2024).

14. How to create a portal. URL: https://accessally.com/blog/how-to-create-a-portal-website/ (дата звернення: 12.05.2024).

15. Installation | webpack. URL: https://webpack.js.org/guides/installation/ (дата звернення: 22.05.2024).

ДОДАТКИ

Додаток А – лістинг коду

```
const handleCheckbox = ({ currentTarget: { checked, name } }) => {
  const { active } = classes;
  const value = checkboxes[name][Number(checked)];
  const list = document.getElementById(value);
  const tabs = document.querySelectorAll(`[data-${name}]`);
  const siblings = list.parentElement.children;

  for (const item of siblings) item.classList.remove(active);
  for (const tab of tabs) {
    tab.classList.remove(active);
    tab.dataset[name] === value && tab.classList.add(active);
  }

  list.classList.add(active);
};

const initSlider = () => {
  new Swiper(".swiper", {
    loop: true,
    slidesPerView: 3,
    spaceBetween: 20,
    initialSlide: 2,
    navigation: {
      nextEl: ".swiper-button-next",
      prevEl: ".swiper-button-prev",
    },
  });
};

const handleFaqItem = ({ currentTarget: target }) => {
  target.classList.toggle(classes.opened);
  const isOpened = target.classList.contains(classes.opened);
  const height = target.querySelector("p").clientHeight;
  const content = target.querySelector(".faq-item__content");

  content.style.height = `${isOpened ? height : 0}px`;
};

const handleScroll = () => {
  const { scrollY: y, innerHeight: h } = window;
  sections.forEach((sec) => {
    if (y > sec.offsetTop - h / 1.5)
sec.classList.remove(classes.hidden);
  });
};

const setTexts = () => {
  const lang = localStorage.getItem("lang") || "en";
  const content = languages[lang];
```

```
  Object.entries(content).forEach(([key, value]) => {
    const items = document.querySelectorAll(`[data-text="${key}"]`);
    items.forEach((item) => (item.innerText = value));
  });
};
```