

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**  
Кафедра інженерії програмного забезпечення

**Пояснювальна записка**

до бакалаврської кваліфікаційної роботи  
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА МОБІЛЬНОЇ ОНЛАЙН ГРИ В ЖАНРІ MMORPG НА  
БАЗІ PHP ТА JS»**

Виконав: студент 4 курсу, групи ПД-44

---

спеціальності 121 Інженерія програмного  
забезпечення

---

(шифр і назва спеціальності)

Хом'як Д.М.

---

(прізвище та ініціали)

Керівник

Яскевич В.О.

---

(прізвище та ініціали)

Рецензент

---

(прізвище та ініціали)

Нормоконтроль

---

(прізвище та ініціали)

Київ – 2021

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ**  
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Інженерії програмного забезпечення

\_\_\_\_\_ О.В. Негоденко

« \_\_\_\_ » \_\_\_\_\_ 2021 року

**З А В Д А Н Н Я**  
**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**  
**ХОМ'ЯКУ ДМИТРУ МИКОЛАЙОВИЧУ**

1. Тема роботи: «Розробка мобільної онлайн гри в жанрі MMORPG на базі PHP та JS»

Керівник роботи Яскевич Владислав Олександрович, кандидат технічних наук затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Середовище розробки Visual Studio Code;

3.2. Науково-технічна література.

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

4.1. Аналіз ринку мобільних ігор;

4.2. Вимоги та оцінка якості гри;

4.3. Опис використаних технологій;

- 4.4. Опис проектування гри.
5. Перелік графічного матеріалу (назва основних слайдів)
- 5.1. Популярність мобільних онлайн ігор;
- 5.2. Переваги та недоліки найпопулярніших мобільних онлайн ігор;
- 5.3. Програмні засоби реалізації;
- 5.4. Огляд функціоналу гри та тестування;
- 5.5 Апробація результатів дослідження.
6. Дата видачі завдання 19.04.2021

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 21.04.21	
2	Аналіз ринку	21.04.21 – 25.04.21	
3	Розробка вимог та оцінка якості	26.04.21 – 05.05.21	
4	Підбір технологій для реалізації	05.05.21 – 06.05.21	
5	Розробка гри	06.05.21 – 20.05.21	
6	Вступ, висновки, реферат	20.05.21 – 23.05.21	
7	Розробка презентації	23.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент \_\_\_\_\_  
( підпис )                      ( прізвище та ініціали )

Керівник роботи \_\_\_\_\_  
( підпис )                      ( прізвище та ініціали )





## РЕФЕРАТ

Текстова частина бакалаврської роботи 61 с., 37 рис., 21 джерело.

Ключові слова: РОЗРОБКА, ОНЛАЙН, MMORPG.

*Об'єкт дослідження* – розробка мобільних онлайн ігор.

*Предмет дослідження* – мобільна онлайн гра в жанрі MMORPG, розроблена на основі PHP та JS.

*Мета роботи* – розробка гри для портфолію, отримання досвіду у розробці мобільних онлайн ігор.

*Методи дослідження* – емпіричні методи, спостереження та опис. Теоретичні методи, такі як аналіз, узагальнення, пояснення та кваліфікація.

У роботі проведено аналіз вже існуючих мобільних онлайн MMORPG ігор. Розглянуто які сучасні технології використовуються для написання. Проаналізовані їх плюси та мінуси використання. Обрано найбільш простий та зручний спосіб розробки мобільної онлайн гри з елементами MMORPG.

Описаний базовий функціонал гри, з використанням яких технологій це реалізовано, чому саме так, а не інакше.

Описана база даних, її таблиці та взаємодія з API онлайн гри.

*Галузь використання* – так як гра буде розповсюджуватись безкоштовно через платформу для Android додатків Google Play, її можна використовувати будь-яка людина, в якій є смартфон на базі OS Android або через браузер на IOS.

## ЗМІСТ

ВСТУП.....	10
1 АНАЛІЗ РИНКУ МОБІЛЬНИХ ІГОР .....	12
1.1 Пошук та вибір платформи для гри.....	12
1.2 Пошук популярних онлайн ігор.....	12
1.3 Аналіз популярності.....	14
2 ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ.....	16
2.1 Середовище для розробки.....	16
2.2 Обрані технології розробки .....	17
2.2.1 Linux, Ubuntu .....	17
2.2.2 Apache.....	19
2.2.3 Nginx.....	21
2.2.4 MySQL та PMA.....	23
2.2.5 PHP.....	26
2.2.6 HTML.....	27
2.2.7 CSS.....	29
2.2.8 JavaScript та JQuery .....	30
2.2.9 C# .....	31
3 РОЗРОБКА, ОГЛЯД ТА ТЕСТУВАННЯ ГРИ.....	32
3.1 Розробка основної частини гри .....	32
3.1.1 Налаштування сервера.....	32
3.1.2 Створення бази даних .....	36
3.1.3 Серверна частина .....	38
3.1.4 Клієнтська частина.....	40
3.2 Тестування та розробка додатку для Google Play .....	41
3.2.1 Розробка Android додатку .....	41
3.2.2 Google Play Console.....	42
3.2.3 Тестування гри.....	42
3.3 Огляд можливостей гри .....	44

3.4 Огляд рейтингу гри.....	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
Додаток А.....	56
Додаток Б.....	59



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GUI – Guide User Interface

ПЗ – програмне забезпечення

СУБД – Система Управління Базами Даних

IDE – Інтегроване середовище розробки

VDS – Virtual Dedicated Server

DDOS – Атака на сервер з метою зробити його недоступним

HTTP – Протокол передачі даних

HTTPS – Захищений протокол передачі даних

БД – База даних

PMA - PhpMyAdmin

JS - JavaScript

API - Application Programming Interface

SSL - Secure Sockets Layer

MMORPG - Massively Multiplayer Online Role-Playing Game

DOM - Document Object Model

## ВСТУП

Сьогодні важко представити наше життя без смартфонів. Месенджери, відео, соціальні мережі, онлайн ігри. Здавалось би, як можна було жити без цього всього в кишені? Смартфони замінили нам повноцінні комп'ютери. Більше не потрібно вставати з дивану щоб написати повідомлення своєму другу або пограти з ним в онлайн гру. Все це доступно тепер із смартфона.

Завдяки швидкому збільшенню функціоналу та можливостей смартфонів ми спостерігаємо стрімкий ріст кількості мобільних додатків. Всі соціальні мережі в першу чергу орієнтуються на мобільну версію. Новини тепер онлайн, будь-яке відео на вибір. Навіть найпопулярніші ігри, які потребують далеко не самий простий комп'ютер, тепер доступні на смартфоні.

На даний момент існує величезна кількість мобільних онлайн ігор. Але, більшість з них це простенькі, нічим не привабливі проекти, які створені для викачки грошей з людей, де за кожну дію потрібно заплатити або подивитись рекламу. Але є і багато перспективних проектів, в які хочеться грати, які чудово підходять для витрачання вільного часу.

*Об'єкт дослідження* - розробка мобільних онлайн ігор.

*Предмет дослідження* - мобільна онлайн гра в жанрі MMORPG, розроблена на основі PHP та JS.

*Мета роботи* – аналіз ринку мобільних ігор, розробка гри для портфоліо, отримання досвіду у розробці мобільних онлайн ігор.

Відповідно до мети роботи можна відмітити основні завдання:

1. Розглянути які сучасні технології використовуються для написання ігор;
2. Проаналізувати їх плюси та мінуси використання;
3. Обрано найбільш простий та зручний спосіб розробки мобільної онлайн гри з елементами MMORPG;
4. Описати базовий функціонал гри, з використанням яких технологій це реалізовано, чому саме так, а не інакше і т.д.;
5. Описати базу даних, її таблиці та взаємодію з API онлайн гри.

*Методи дослідження* – емпіричні методи, спостереження та опис. Теоретичні методи, такі як аналіз, узагальнення, пояснення та кваліфікація.

Перш за все потрібно проаналізувати ринок, знайти прямих конкурентів, дослідити їх плюси та мінуси як в самій грі, так і у використаних для її реалізації технологіях. Далі, взяти з них найкращі ідеї та найпростіші технології для реалізації свого проекту.

**Наукова новизна** роботи полягає в наступному:

1. Розроблено конкурентоспроможну гру в своєму жанрі;
2. Встановлено, що для реалізації великого та успішного проекту достатньо використати PHP та JS;
3. Проаналізовано способи реалізації, їх плюси та мінуси;
4. Проведено аналіз ринку мобільних ігор;

**Практична значущість** дослідження полягає у вирішенні важкої практичної задачі – розробка мобільної онлайн MMORPG гри з використанням найпростіших технологій в короткий термін та з мінімальними витратами по грошам.

Результати досліджень бакалаврської роботи апробовані:

1. На всеукраїнській науково-технічній конференції: «Застосування програмного забезпечення в ІКТ»;
2. На всеукраїнській науково-технічній конференції: «Сучасні інтелектуальні інформаційні технології в науці та освіті».

## 1 АНАЛІЗ РИНКУ МОБІЛЬНИХ ІГОР

### 1.1 Пошук та вибір платформи для гри

На сьогоднішній день існує чотири платформи для завантаження додатків зі смартфона. Для Android це Google Play та Amazon Store, для Windows Phone - Windows Store, а для IOS – APP Store. На таблиці 1 зображене порівняння всіх популярних платформ для вибору найкращих з них.

Таблиця 1 – Порівняння платформ [1]

	Google Play	APP Store	Amazon Store	Windows Store
Кількість додатків	2.7 млн.	1.82 млн.	450 тис.	669 тис.
Кількість завантажень	28.3 млрд.	8.2 млрд.	2.2 млрд.	3.6 млрд.
Дохід (доларів)	17.3 млрд.	32 млрд.	5 млрд.	7.1 млрд.

Як ми бачимо, найбільш популярними є Google Play та APP Store. Незважаючи на те, що кількість додатків та завантажень у першої набагато більше, ніж у другої – ми спостерігаємо цікаву картину. Дохід від APP Store майже в два рази більше, ніж від Google Play [2]. Візьмемо їх за основу для майбутнього аналізу ринку та випуску своєї гри саме на цих платформах.

### 1.2 Пошук популярних онлайн ігор

Проведемо пошук по запиту “MMORPG online” на кожній з платформ. Першою на черзі буде Google Play. На рисунку 1 зображений пошук на системі Android.

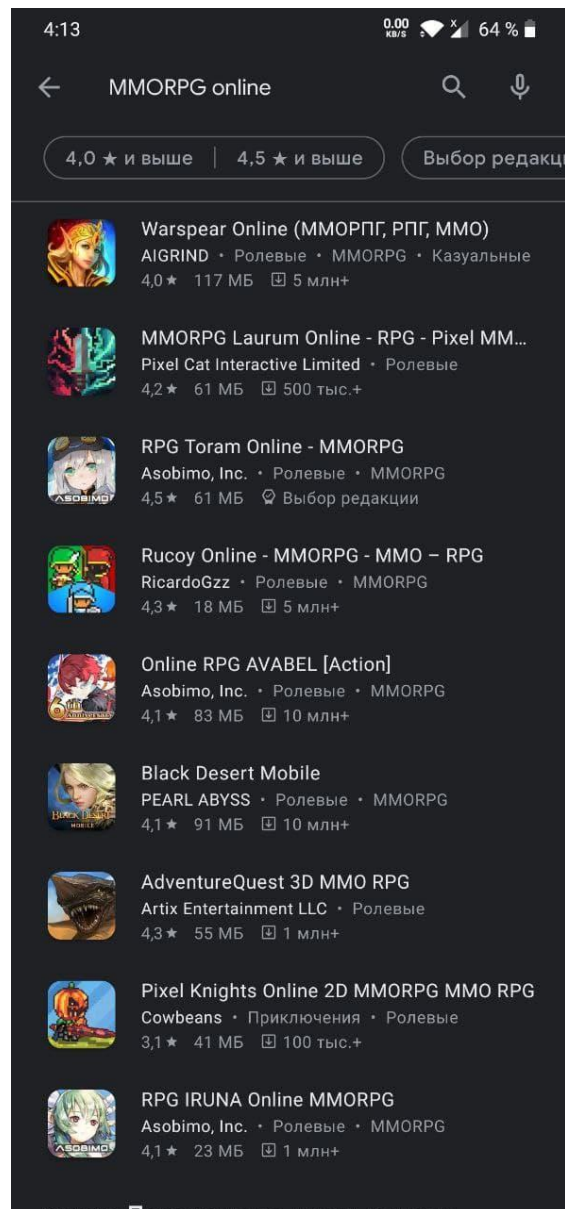


Рисунок 1 – Пошук в Google Play

Виділимо для себе такі додатки:

1. Warspear Online;
2. Black Desert Mobile;
3. MMORPG Laurum Online.

Далі зробимо теж саме, але використовуючи APP Store. На рисунку 2 зображений пошук в системі MacOS.



Рисунок 2 – Пошук в APP Store

Також виділимо для себе основні додатки:

1. Warspear Online;
2. Pixel Starships;
3. Pirates of Everseas.

Як ми бачимо, основним конкурентом, який доступний на двох платформах буде Warspear Online. Далі розберемо популярність цієї гри, її основну аудиторію та можливості.

### 1.3 Аналіз популярності

Основаючись на пошук ми знайшли популярний додаток для аналізу та створення своєї гри базуючись на всіх його перевагах. Провівши декілька годин в грі та пройшовши навчання, можна зробити висновки і описати знайдені переваги та недоліки.

Основними перевагами будуть:

1. Графіка;
2. Геймплей;
3. Атмосфера;
4. Free-to-play;
5. Велика кількість завдань, предметів і босів.

Основні недоліки:

1. Потребує вкладання грошей;
2. Часткова відсутність балансу ігрових предметів;
3. Потрібен постійний онлайн;
4. Технічна підтримка.

Основною аудиторією гри є підлітки віком 10-14 років, але так як в грі присутня торгівля між гравцями майже всіх предметів, то є велика кількість людей віком 18+, які займаються продажом ігрових ресурсів за реальні гроші. Таким чином гра бере на себе аудиторію від 10 до 22 років.

В грі є величезна кількість кланів, земель, за володіння якими можна воювати, можливість взаємодіяти з іншими гравцями, битв з босами, як командних, там і одиночних. Загалом, все зроблено для максимального охоплення аудиторії.

Також, гра доступна на всіх сучасних платформах. Таких як:

1. Android;
2. IOS;
3. Windows.

Це дає змогу охопити як можна більше людей, що і потрібно.

Також потрібно звернути увагу на утримання гравців у грі. Для цього існує реферальна програма, постійні бонуси для гравців, які постійно онлайн та оновлення контенту, щоб постійно було куди рости. Також, раз в декілька років випускається глобальна обнова, які кардинально змінює або додає чогось нового в геймплей.

Візьмем основну механіку, геймплей та переваги для нашої гри.

## 2 ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

### 2.1 Середовище для розробки

Вибір впав на Visual Studio Code. Це чудове, а саме головне “легке” середовище для багатоплатформної розробки додатків [3]. Воно містить величезну кількість плагінів, які дозволяють додати будь-який відсутній за замовчуванням функціонал. Використовувати Visual Studio Code можна не тільки на Windows, а й на Linux і OS X.

Його особливостями є:

1. Можливість розробляти додатки як з графічним інтерфейсом, таким як Windows Forms, так і консольним;
2. Підтримка величезної кількості мов програмування. Наприклад: PHP, JavaScript, Python, C++, C#;
3. Присутня за замовчуванням інтеграція з сервісами GIT та GitHub, підказки та автодоповнення стандартних конструкцій, вмонтований debugger, який працює з більшістю мов програмування;
4. Підсвітка синтаксису;
5. Можливість встановлювати плагіни безпосередньо із самого середовища для розробки.

Visual Studio Code – це не повноцінна IDE, але дуже зручний та простий редактор коду, який ідеально підходить для початківців у програмуванні. До того ж його можливості постійно розширюються як компанією-розробником, так і ентузіастами, які його використовують і хочуть зробити більш функціональним.

На рисунку 3 зображений GUI Visual Studio Code.



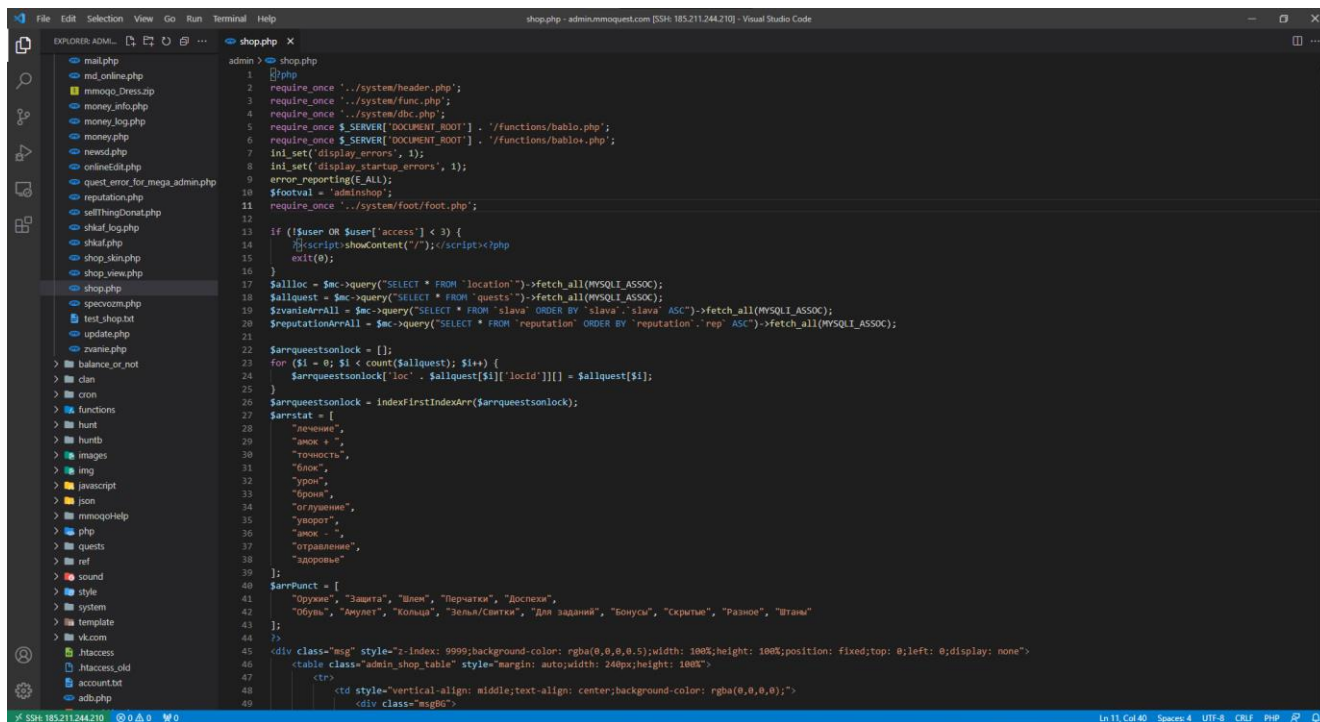


Рисунок 3 – Інтерфейс Visual Studio Code

## 2.2 Обрані технології розробки

### 2.2.1 Linux, Ubuntu

Спочатку потрібно обрати на чому буде розвертатись сервер. Оскільки для обробки великої кількості запитів потрібні великі потужності, було вирішено орендувати VDS. Це дає змогу уникнути проблем з навантаженням завдяки потужним серверам та підтримувати доступність гри в будь-який час. Також вирішальним фактором став канал доступу до мережі інтернет, адже хостинг надає фіксовану швидкість, безлімітний трафік та примітивний захист від DDOS-атак.

В якості операційної системи для сервера був обраний Linux. Він кардинально відрізняється від Windows, до якого ми всі звикли. В Linux є багато дистрибутивів, які базуються на ядрі Linux [5]. В той час як у Windows є основна платформа з декількома версіями: XP, 7, 10, як приклад.

Розглянемо основні переваги Linux систем перед Windows:

1. Безкоштовність. Якщо ми беремо будь-яку версію Windows – за неї потрібно заплатити, в той час як всі дистрибутиви Linux розповсюджуються абсолютно безкоштовно;
2. Кастомізація. Відкритий вихідний код дає змогу повністю редагувати систему під себе. Можна змінити все, навіть вигляд графічного інтерфейсу;
3. Швидка і проста установка. Будь-який дистрибутив можна запустити з флешки перед установкою і протестувати;
4. Безпека. Дистрибутиви на стільки відрізняються, що зробити якийсь універсальний вірус неможливо, саме це зумовлено відсутністю їх на таких системах;
5. Швидка робота на базових комп'ютерах. Для роботи Linux достатньо самого примітивного відра, який запускається і видає картинку;
6. Вмонтовані драйвера. В ядрі зберігаються всі необхідні налаштування для правильної роботи будь-яких пристроїв без зайвих установок;
7. Термінал. Це консоль для управління всією системою, в ній можна абсолютно все. Як змінити якісь параметри системи, так і встановити додатковий функціонал за допомогою пакетів та команд.

В Linux дуже багато переваг і мало недоліків, саме тому вона була обрана як операційна система для сервера.

Серед усіх дистрибутивів був обраний Ubuntu. Він базується на основі Debian. Це також один з дистрибутивів. Розповсюджується Ubuntu абсолютно безкоштовно [4]. Його особливостями є:

1. Орієнтованість на простих користувачі. Багато речей спрощено;
2. Чудова апаратна підтримка;
3. Незвичний, як для Linux-подібних систем, графічний інтерфейс, який нагадує Windows;
4. Безпека.

На рисунку 4 зображений GUI системи Ubuntu.

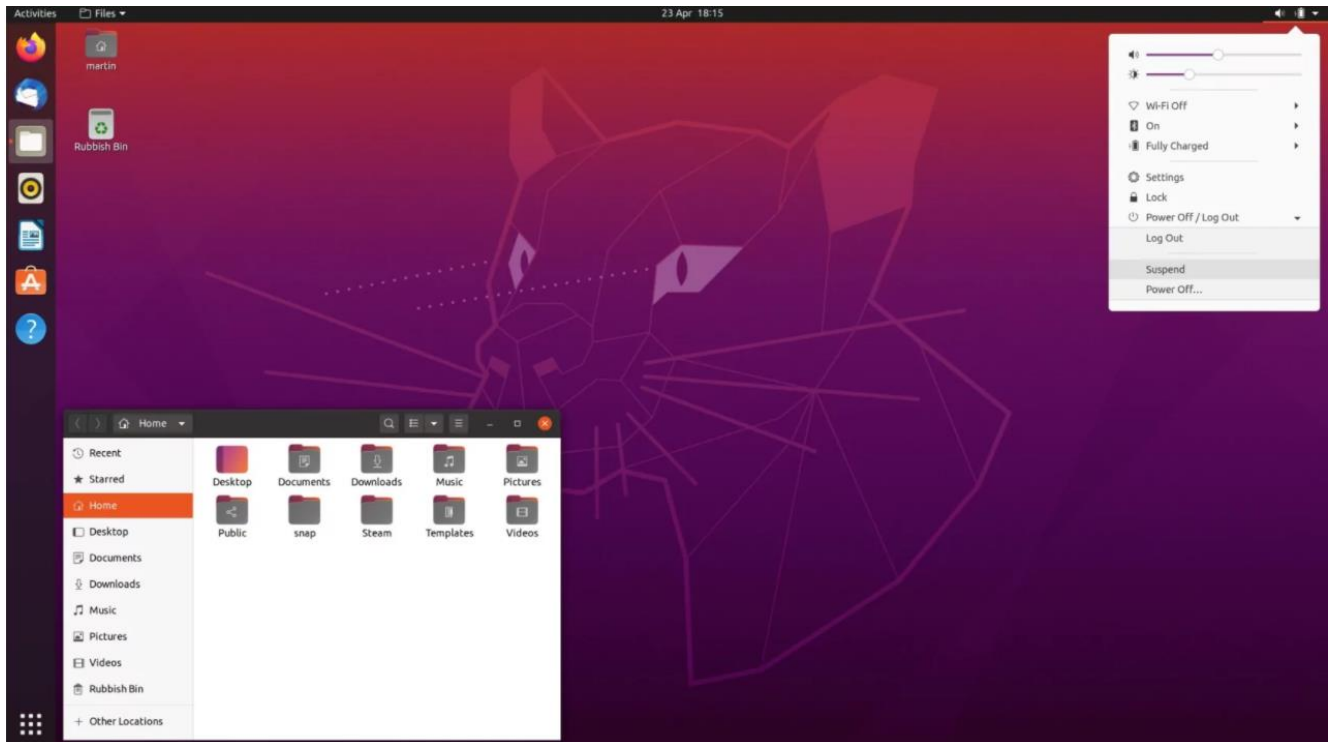


Рисунок 4 – Інтерфейс дистрибутиву Ubuntu

### 2.2.2 Apache

Далі нам знадобиться програмне забезпечення для розміщення веб-сервера. Apache гарно себе зарекомендував у масштабних проектах, до того ж він являється найпопулярнішим ПО в своїй сфері [6]. Він має величезну кількість налаштувань, що дозволяє реалізувати всі особливості онлайн гри.

Apache складається з динамічної модульної системи та ядра. Налаштувати його можна за допомогою config-файлу.

Ядро може працювати без будь-яких модулів, але тоді його функціонал буде обмежений. Воно служить для обробки запитів HTTP/HTTPS, config-файлів та завантаження модулів, якщо вони підключені. Ядро написано на мові програмування C.

За допомогою модульної системи можна розширювати функціонал роботи ядра. Модулі можуть підключатись як під час, так і після установки всього ПО, адже вони йдуть в комплекті з інсталятором. В основному вони відповідають за певні аспекти обробки клієнтських запитів. Такі як: підтримку різних мов програмування, вебкешування даних, авторизацію та безпеку. Існує понад 500

різних модулів. Розглянувши найпопулярніші з них було вирішено використати ISPmanager.

Це так званий “комбайн” для управління веб-сервером [7]. Він призначений облегшити процес налаштування та адміністрування сервера. Його основними можливостями є:

1. Можливість організувати доступ декількох людей до сервера з різними правами доступу;
2. Швидке налаштування web-доменів, можна додати SSL сертифікати та налаштувати перенаправлення;
3. Управління DNS серверами;
4. Можливість створити поштову систему. А саме поштовий домен та ящик. А також такі особливості як поштові аліаси, перенаправлення, захист від спаму та фільтри;
5. Створення та повне налаштування бази даних;
6. Можливість керування ftp користувачами. Їх налаштування;
7. Повний доступ до всіх файлів на сервері через стандартний менеджер файлів;
8. Налаштування мережевого екрану;
9. Система резервних копіювань даних на випадок аварійних ситуацій;
10. Повний моніторинг ресурсів серверу.

Основним мінусом даного ПО я вважають ціну. Це повністю платний сервіс, який не має безкоштовних аналогів або версій.

Приклад графічного інтерфейсу ISPmanager можна побачити на рисунку 5.

Загалом, Apache має велику кількість переваг, але з них випливають і мінуси. Одним з головних мінусів є продуктивність, адже швидкість обробки користувацьких запитів у сервера буде повільною, саме тому було вирішено використати Apache в парі з більш продуктивним Nginx.

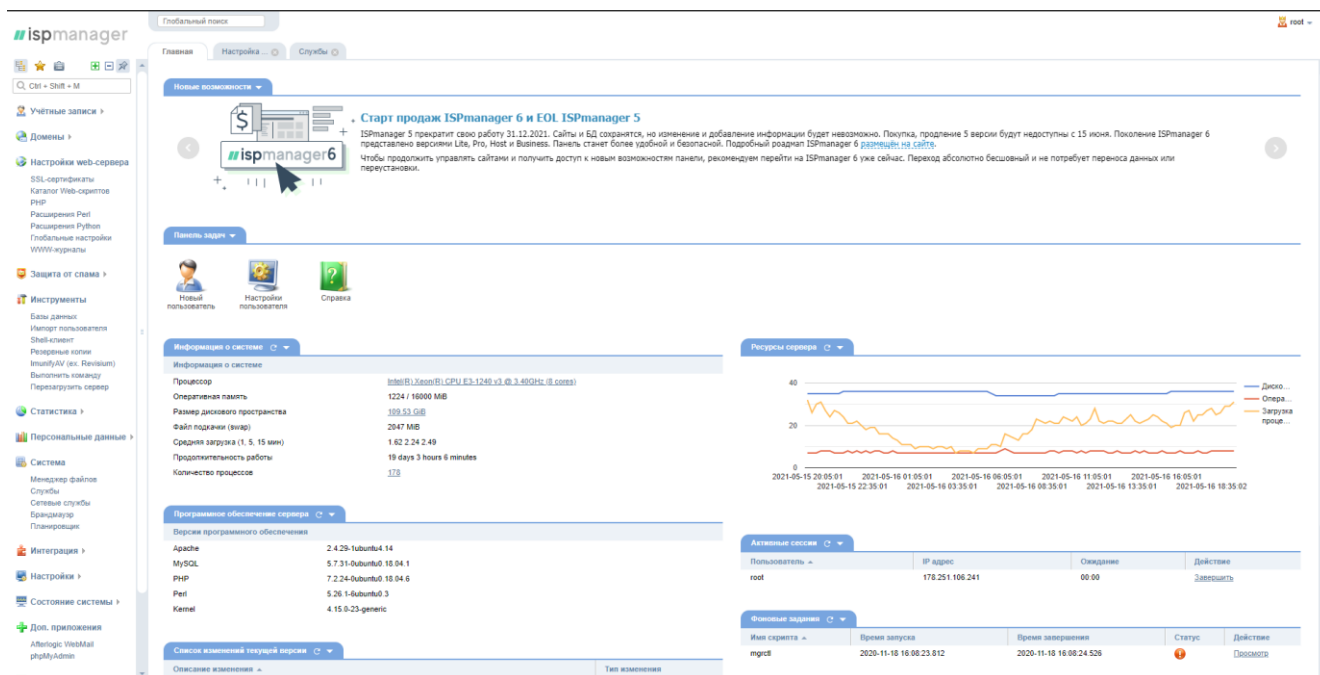


Рисунок 5 – Интерфейс ISPmanager

### 2.2.3 Nginx

Це також HTTP-сервер для обробки запитів користувачів [8]. Його основною перевагою перед Apache є швидкість роботи. Він може чудово кешувати зображення, статичні файли та скрипти, розвантажуючи повільне HTTP-з'єднання для швидкої відправки користувачам. В нашому випадку Nginx буде виконувати саме таку, важку роботу, пов'язану з HTTP запитами. В свою чергу Apache буде виконувати код сервера в безпечному та захищеному середовищі.

Саме таким чином ми досягнемо швидкої роботи гри та позбавимось від головного недоліку Apache – швидкості. По різних тестах, використання такої зв'язки ПО дозволяє підняти швидкість роботи додатку більш ніж в два рази.

На рисунку 6 можна побачити схему роботи Apache разом з Nginx.

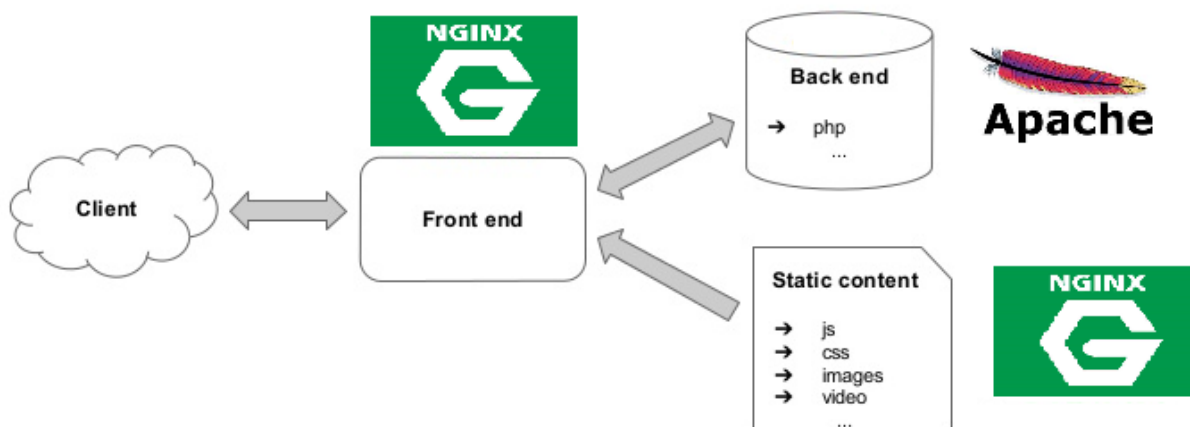


Рисунок 6 – Взаємодія Apache та Nginx

Також, створимо наглядну блок-схему роботи сервера з використання такого ПО. Її можна побачити на рисунку 7.

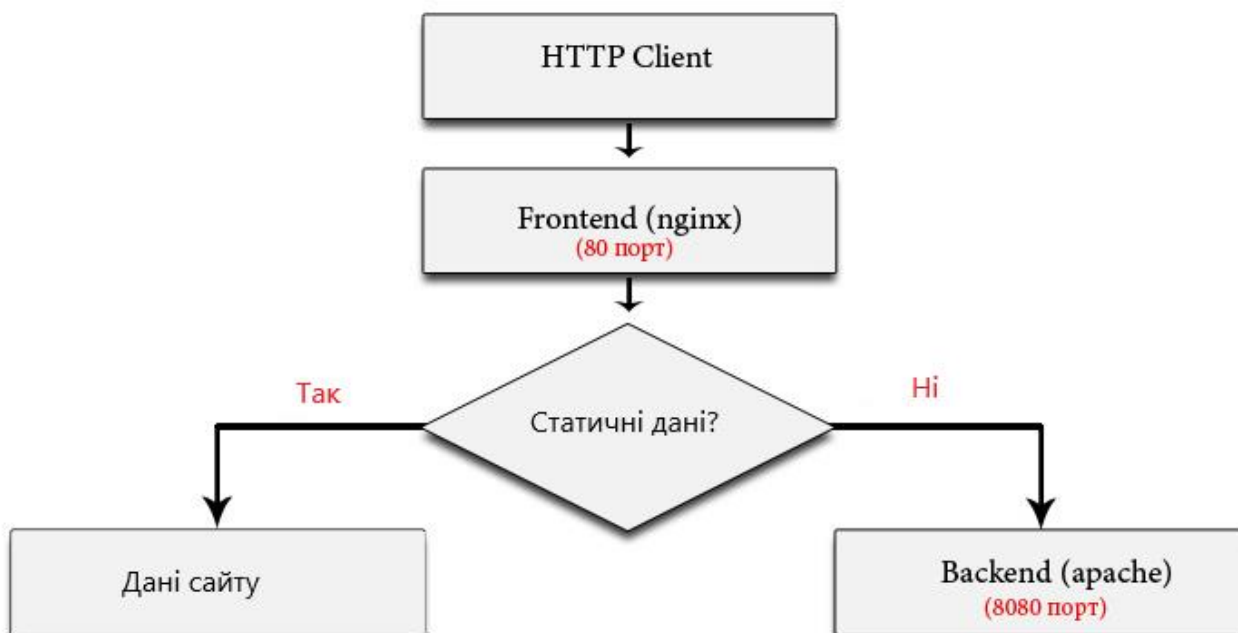


Рисунок 7 – Блок-схема обробки запитів сервером

#### 2.2.4 MySQL та PMA

Так як в грі буде реєстрація, велика кількість даних гравця, предмети, локації – це все потрібно десь зберігати. Для цього ми використаємо БД. В якості бази даних слугує MySQL. Це безкоштовне програмне забезпечення, яке слугує для взаємодії з БД використовуючи мову SQL [9]. СУБД MySQL ідеально підходить для серверів з обмеженими ресурсами. Також вона має високу ступінь надійності, що дозволяє грі працювати цілодобово.

Основні переваги:

1. Багатопотоковість. Це дозволяє робити декілька запитів у БД одночасно;
2. Безкоштовна. MySQL розповсюджується без будь-якої плати за використання;
3. Швидка робота в купі з масштабуванням;
4. Гнучке налаштування форматів чисел, тексту, міток часу та дати;
5. Оптимізація зв'язків між таблицями;
6. Швидка робота транзакцій через InnoDB;
7. Велика кількість типів таблиць.

Максимальної швидкості обробки запитів MySQL досягає завдяки правильному налаштуванню сервера, розподілення ресурсів та вибору правильного типу таблиць. В дані СУБД є сім основних типів таблиць, але ми будемо використовувати тільки два. Це InnoDB для швидкої обробки запитів до динамічних даних та MyISAM для зберігання великої кількості статичних даних, таких як статистика користувачів, їхні дії, авторизації і т.д.

InnoDB – це найновіший тип таблиць, MySQL рекомендує використовувати саме його, адже в ньому є підтримка транзакцій, відкатів та захист від втрати даних. Також в ньому використовується блокування на рівні запису та неблокуюче читання, що дозволяє підняти продуктивність БД та пришвидшити обробку запитів до неї. InnoDB використовує первинні ключі для пошуку даних по запиту в таблиці, це дозволяє виконувати прості махінації з таблицею за мінімальну кількість часу, адже не потрібно зрівнювати величезні тексти чи дані.

MyISAM – це тип таблиці, який ідеально підходить для зберігання та нечастого доступу до таблиці. Зробимо порівняння цих двох типів для кращого розуміння їх різниці, яке зображене в таблиці 2.

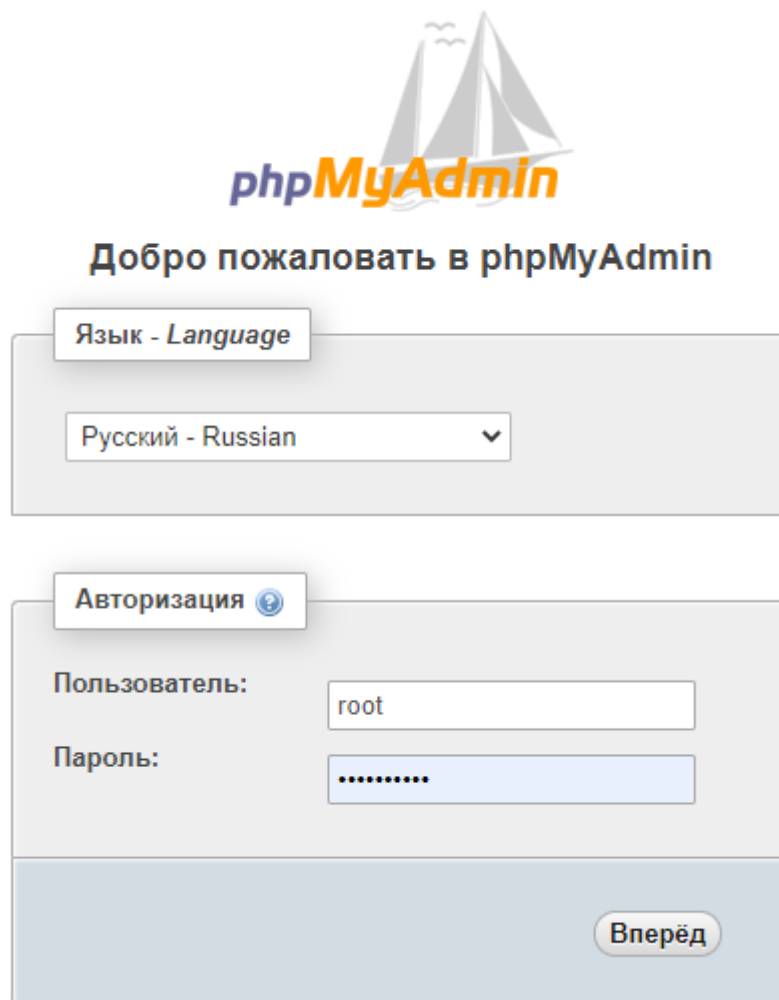
Таблиця 2 – Порівняння типів таблиць [10]

<b>Функція</b>	<b>InnoDB</b>	<b>MyISAM</b>
Максимальний розмір таблиці	64 ТБ.	256 ТБ.
Транзакції	Так	Ні
Блокування	Запис	Таблиця
MVCC	Так	Ні
Хеш-індекси	Ні	Ні
Індекси повнотекстового пошуку	Ні	Так
Кластерні індекси	Так	Ні
Кешування даних	Так	Ні
Кешування індексів	Так	Так
Стиснення даних	Так	Так
Шифрування даних	Так	Так
Підтримка кластерних БД	Ні	Ні
Реплікація	Так	Так
Підтримка зовнішніх ключів	Так	Ні
Бекапи	Так	Так
Кешування запитів	Так	Так

Далі настало питання вибору оболонки для адміністрування БД. Так як може не бути доступу до сервера в потрібний момент – PhpMyAdmin добре підходить для швидкої зміни параметрів БД, або створення та редагування таблиць через браузер комп'ютера або мобільного пристрою. Це чудовий GUI для адміністрування, яких полегшує весь процес роботи з БД [11]. РМА підтримує



створення декількох користувачів з різним рівнем доступу для багатьох цілей. Вхід в систему можна побачити на рисунку 8. На рисунку 9 зображений GUI оболонки.



phpMyAdmin

Добро пожаловать в phpMyAdmin

Язык - *Language*

Русский - Russian

Авторизация

Пользователь: root

Пароль: .....

Вперёд

Рисунок 8 – Вхід до PhpMyAdmin

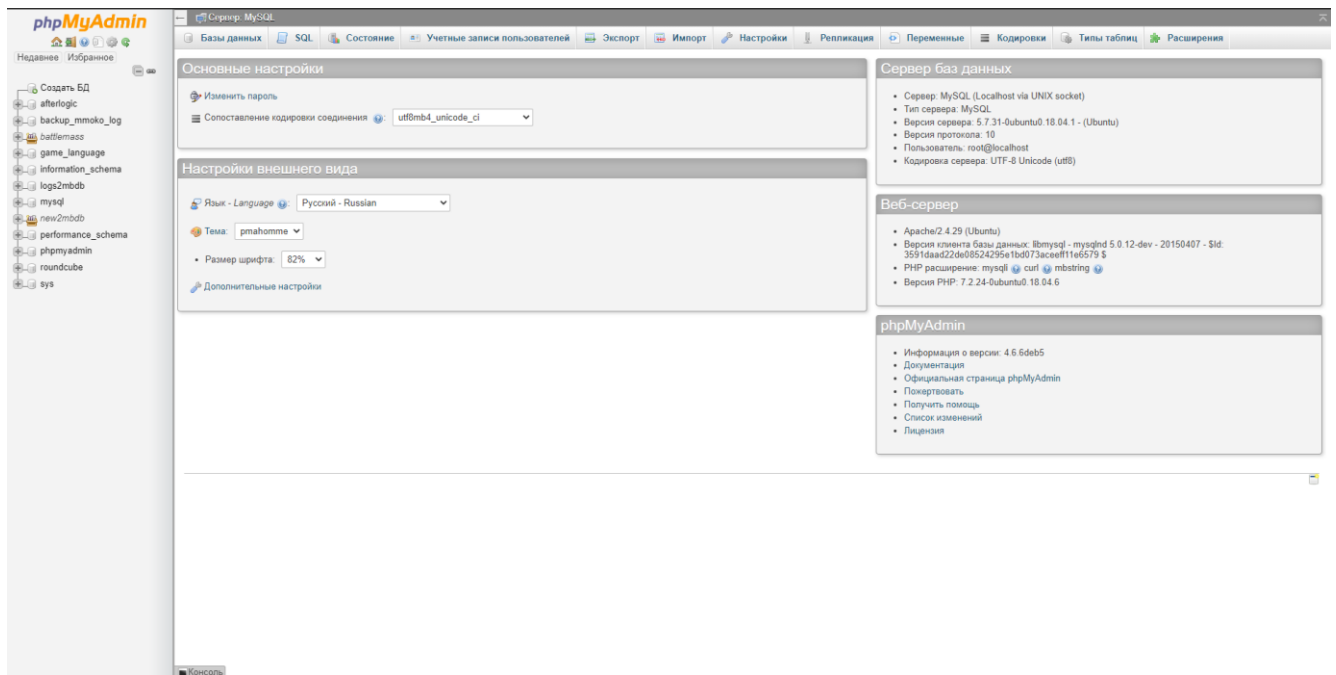


Рисунок 9 – Интерфейс PhpMyAdmin

Вибір MySQL та PhpMyAdmin виправданий ще тим, що даний набір ПО створений спеціально для PHP, який буде в якості серверної мови програмування.

### 2.2.5 PHP

Це серверна мова програмування, яка була однією із самих перших в світі. PHP зарекомендував себе як швидка та перспективна мова, адже при правильному використанні за допомогою неї можна отримати гарні результати у продуктивності [12].

На даний момент її використовують такі гіганти як соціальна мережа Facebook, електронний майданчик об'яв OLX, енциклопедія Wikipedia та багато інших не менш відомих сервісів.

На сьогоднішній момент найновішою версією PHP являється 7, тому саме її ми і будемо використовувати.

Розглянемо основні переваги:

1. Безкоштовно. PHP розповсюджується повністю без будь-якої плати за використання. Вся документація та велика кількість літератури для вивчення також є безкоштовною;

2. Багатоплатформність. PHP доступний на всіх популярних ОС, компіляція коду не буде відрізнятися від операційної системи. Таким чином можна написати все на Windows, а потім без будь-яких проблем запустити це на Linux;
3. Легко у вивченні. PHP має набагато нижчий поріг входу в розробку ніж його конкуренти. Принципи розробки не сильно відрізняються від інших мов програмування, тому вивчивши PHP можна буде з легкістю перейти на іншу мову;
4. Орієнтованість на веб-розробку. PHP створювався та розвивався в першу чергу для створення веб-сервісів, але його багатий функціонал дозволяє використовувати цю мову як API.

Окрім переваг PHP має також і недоліки. Розглянемо самий основний.

Це синтаксис. Так як PHP вже дуже стара мова програмування, вже вийшла велика кількість версій, які змінюють назву базових функцій та методів. Через це виникає велика кількість проблем, адже старі функції нікуди не зникають, а нові продовжують додаватись та дублювати роботу своїх попередників.

Написаний код на PHP можна умовно поділити на наступні типи:

1. Компоненти – це бібліотеки, які збираються за допомогою менеджера пакетів;
2. Фреймворки – це вже готовий набір, дуже часто звичайних компонентів, який спрямований на розробку сайтів будь-якого типу. Наприклад, Laravel, Yii, Symfony;
3. Повноцінні додатки – це вже готові до використання сервіси. Наприклад WordPress або PhpMyAdmin, який ми будемо використовувати.

### **2.2.6 HTML**

Для відображення елементів гри буде використовуватись найпопулярніша мова розмітки. Найголовніше призначення HTML – це відображення тексту та

придання йому логічного значення [13]. За допомогою нього можна розділити сторінку на потрібну структуру для гри, а також відображати текстові елементи.

Розглянемо основний синтаксис. Для того щоб відобразити будь-який текст за допомогою HTML потрібно “обгорнути” його в тег.

Тег – це назва елемента, яка означає якість логічне значення свого вмісту. Він позначається з обох боків символами “<” та “>”. Ім’я не чутливе до реєстру, тому може бути записане будь-яким чином.

Деякі теги повинні закриватись, тому вони складаються з двох частин:

1. Відкриваючий тег. Наприклад: “<div>”;
2. Закриваючий тег, який позначається символом “/” перед назвою. Наприклад: “</div>”.

В конструкцію тегу можна додати атрибут, який буде використовуватись іншими елементами гри для виконання певних маніпуляцій з елементом.

Атрибут – конструкція типу “назва='значення’”, яка знаходиться в середині HTML-елемента.

На рисунку 10 зображений приклад конструкції тегу.

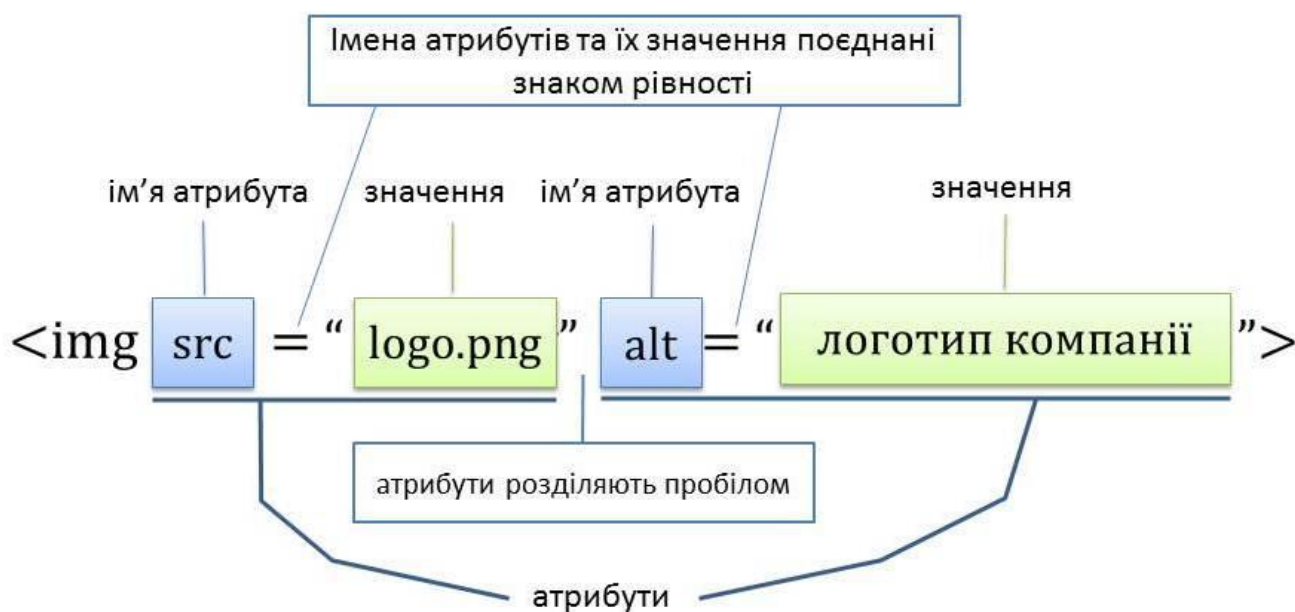


Рисунок 10 – Приклад тегу з атрибутами

### 2.2.7 CSS

Для придання красивого вигляду гри буде використовуватись мова стилів. CSS – це в першу чергу таблиця стилів, яку ми описуємо для стилізації контенту в грі [14]. За допомогою неї описуються окремі елементи HTML розмітки.

Для цього потрібно взяти селектор та прописати для нього потрібні нам стилі.

Селектор – це конструкція для вибору певних елементів у HTML розмітці. Це може бути як назва елемента, так і його клас або атрибуту.

Клас – один з атрибутів елемента, який дозволяє застосувати однакові стилі для багатьох HTML-елементів водночас.

На рисунку 11 приведений приклад опису стилів у елемента.

Також одним з основних атрибутів є `id`. Це унікальний номер або назва для одного елемента, за допомогою якого можна швидко звернутись до самого HTML-елементу. В основному, `id` використовується в JavaScript, адже інтерпретатор браузера виносить посилання на такі елементи в самий верх, що дозволяє швидко і без будь-яких проблем знайти та отримати значення елемента.

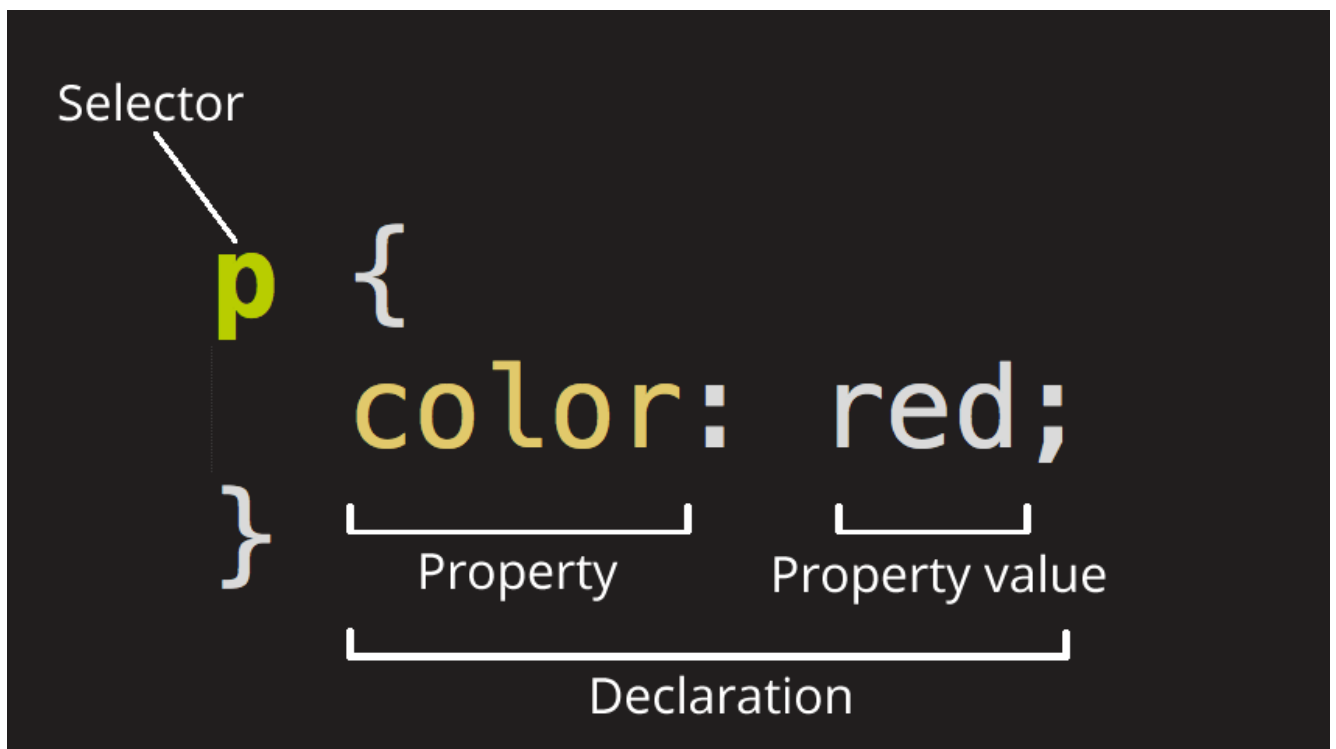


Рисунок 11 – Приклад опису стилів за допомогою CSS

### 2.2.8 JavaScript та JQuery

Це легка, об'єктно-орієнтовна, інтерпритуєма мова програмування з динамічною типізацією [15]. Найчастіше JS використовується для створення функціоналу для web-сторінки. В даному випадку він буде слугувати frontend мовою, як інструмент додавання функціоналу елементами гри.

Розглянемо основні переваги:

1. Швидкість роботи та продуктивність. JS дозволяє частково обробляти веб-сторінки без серверу, чим економить ресурси як клієнта, так і сервера;
2. Величезна кількість фреймворків. Останнім часом їх все більше і більше, адже це одна з найбільш популярних мов програмування;
3. Простота у використанні. Будь-яку задачу можна вирішити за допомогою базового набору функціоналу, без використання додаткових бібліотек;
4. Простота у вивченні. Синтаксис JS дуже легкий та схожий з іншими мовами, тому його вивчення як першої мови дасть величезний потенціал для інших;
5. Підтримка. JS постійно оновлюється, додається новий функціонал, який підтримується усіма сучасними браузерами.

Тепер розглянемо його мінуси, адже їх також не мало:

1. Динамічна типізація. JS може повністю проігнорувати деякі помилки, пов'язані із типами даних. Виявити їх на етапі розробки майже нереально;
2. Безпека. В інтерпретатор у вигляді браузера можна дуже просто вмонтувати вірусний код, який може нашкодити користувачеві;
3. Відсутність можливості завантажити або зчитати якийсь файл. Це зумовлено саме попереднім пунктом;
4. Абсолютна відсутність віддаленого доступу. Якщо завантажити код користувачу в браузер, то він там і буде жити поки користувач не оновить сторінку і не завантажиться нова версія файлу з кодом.

Для розробки гри JS цікавий тим, що він достатньо простий у вивченні, швидко розвивається та виправляє всі свої недоліки. Його популярність говорить

сама за себе. На рисунку 12 зображений рейтинг популярних мов за версією ресурсу Stack Overflow [16].



Рисунок 12 – Рейтинг популярних мов програмування та розмітки

Для спрощення загального синтаксису буде використовуватись бібліотека jQuery. Це набір функціоналу, який сформований на взаємодію з JavaScript та HTML [17]. JQuery дозволяє отримати більш швидкий та простий доступ до DOM, що поліпшує динамічне редагування інтерфейсу. Також до його плюсів можна віднести об'єднання всіх реалізацій DOM між браузерами та виправлення відомих багів у CSS та HTML.

### 2.2.9 C#

Щоб реалізувати клієнт для ОС Android потрібно використати об'єктно-орієнтовну мову програмування C# [18]. Це одна з найбільш популярних C-подібних мов програмування, яка містить всі необхідні функції для створення Android-додатку.

## 3 РОЗРОБКА, ОГЛЯД ТА ТЕСТУВАННЯ ГРИ

### 3.1 Розробка основної частини гри

#### 3.1.1 Налаштування сервера

Перш за все орендуємо сервер на будь-якому популярному хостингу. Вибір впав на cloud-shield. Це сервіс, який надає VDS сервера, а також розширений захист від DDOS-атак з використанням тунельного захисту. Також основним плюсом даного сервісу є надання безлімітного інтернет каналу шириною в 1 GB в секунду.

Замовляємо базову конфігурацію на процесорі Intel Xeon E3-1240 v3, з 16 GB оперативної пам'яті та 128 GB SSD накопичувачем та вказуємо, що потрібен ISPmanager.

Після здійснення оплати на сервер на пошту приходить логін та пароль для доступу на сервер. Входимо по отриманій ip-адресі та вводимо наші дані. Після авторизації бачимо GUI з рисунку 5.

Перш за все нам потрібно виконати початкову установку базових компонентів. Таких як PHP, Nginx, Apache, MySQL, PhpMyAdmin. Після закінчення установки у вікні сповіщень видно, що всі обрані компоненти успішно встановлені. Можна приступати до подальшого налаштування.

Далі створимо та налаштуємо домени. Перед цим їх потрібно купити, тому потрібно звернутись до ринку та обрати найбільш вдалу назву та адекватну ціну. Для цього звернемося до сервісу з продажу доменів GoDaddy. Це один з найкрупніших реєстраторів, який підтримує домену зону першого рівня, тобто:

1. Com;
2. Org;
3. Net;
4. Info;
5. Biz.

Процес створення домену зображений на рисунку 13. Як виглядають вже створені можна побачити на рисунку 14.



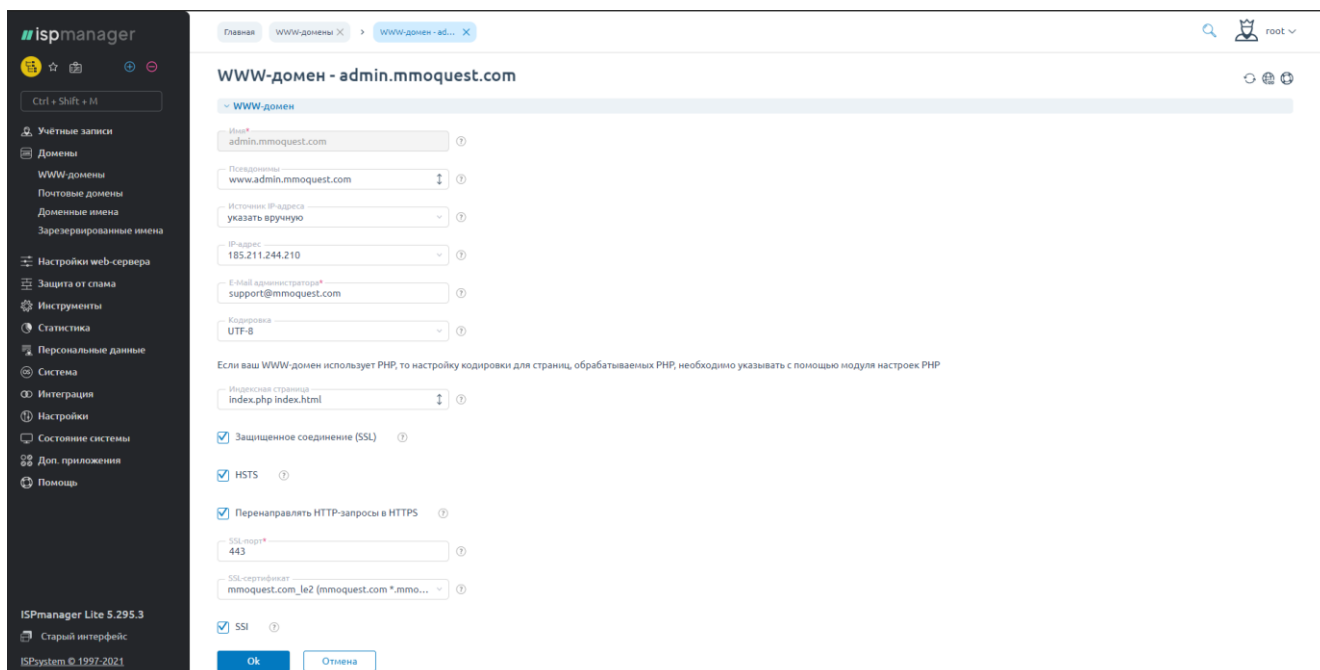


Рисунок 13 – Створення доменів

Имя	Владелец	Корневая директория	IP-адреса	Параметры	Режим работы PHP	Версия PHP
admin.mmoquest....	www-root	/var/www/www-root/data/admin.mmoquest.com	185.211.244.210		модуль Apache	7.2.24 (native)
battlemass.com	www-root	/var/www/www-root/data/battlemass.com	185.211.244.210		модуль Apache	7.2.24 (native)
mmoquest.com	www-root	/var/www/www-root/data/mmoquest.com	185.211.244.210		модуль Apache	7.2.24 (native)
mmoquest.online	www-root	/var/www/www-root/data/mmoquest.online	185.211.244.210		модуль Apache	7.2.24 (native)
mmoquest.ru	www-root	/var/www/www-root/data/mmoquest.ru	185.211.244.210		модуль Apache	7.2.24 (native)
mobitva2.online	www-root	/var/www/www-root/data/mobitva2.online	185.211.244.210		модуль Apache	7.2.24 (native)

Рисунок 14 – Огляд створених та налаштованих доменів

Щоб з'єднання було конфіденційним – використаємо SSL сертифікат. Він також є платним та потребує підключення на кожен домен окремо. На рисунку 15 зображений процес створення SSL. Таким чином отримуємо сертифікат на кожен із доменів. Для цього використаємо сервіс Let's Encrypt. Це автоматизований центр сертифікації, за допомогою якого можна отримати SSL [19]. Список сертифікатів можна побачити на рисунку 16.

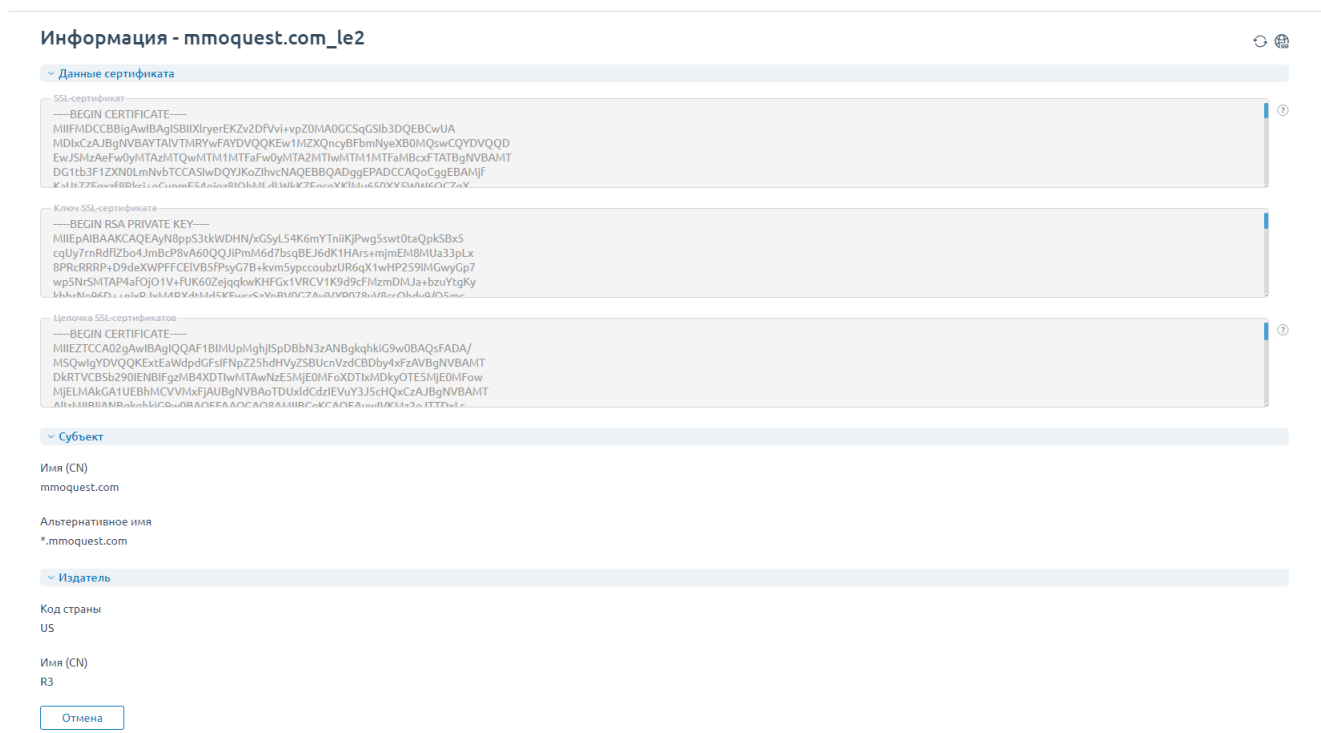


Рисунок 15 – Створення SSL сертификата

Имя SSL-сертификата	Владелец	Действителен до	Тип	Используется
battlemass.com_le2	www-root	2021-06-12	Существующий	...
mmoquest.com_le2	www-root	2021-06-12	Существующий	...
mmoquest.online_le2	www-root	2021-06-12	Существующий	...
mmoquest.ru_le2	www-root	2021-06-12	Существующий	...
mobitva2.online_le2	www-root	2021-06-12	Существующий	...

Рисунок 16 – Список SSL сертифікатів

Також не забуваємо про журнал помилок, в який будуть записуватись помилки у виконанні певних сторінок гри. На рисунку 17 зображений список вже створених журналів.

Журналы WWW-запросов - www-root

Просмотреть Очистить Архив Ctrl + Shift + F

Имя файла	Размер файла	Дата изменения	Количество архивов	Общий размер архивов
admin.mmoquest.com.access.log	0 B	2021-05-17 06:25:01	5	3.27 KiB
admin.mmoquest.com.error.log	0 B	2021-05-17 06:25:01	5	2.65 KiB
battlemass.com.access.log	361.90 MiB	2021-05-17 16:27:32	10	321.85 MiB
battlemass.com.error.log	83.50 KiB	2021-05-17 16:00:24	10	75.10 KiB
mmoquest.com.access.log	599.75 MiB	2021-05-17 16:27:32	5	423.10 MiB
mmoquest.com.error.log	197.38 KiB	2021-05-17 16:16:37	5	247.61 KiB
mmoquest.online.error.log	0 B	2021-05-17 06:25:34	5	100 B
mmoquest.ru.error.log	0 B	2021-05-17 06:25:34	5	100 B
mobitva2.online.error.log	0 B	2021-05-17 06:25:34	5	100 B

Рисунок 17 – Список журналів помилок

Не забуваємо про налаштування таких компонентів як Nginx та Apache. В першому випадку налаштування знаходяться за шляхом `/etc/nginx/nginx.conf`. Приклад налаштування зображений на рисунку 18. В другому випадку файл налаштування знаходиться в `/etc/apache2/apache2.conf`. Приклад його налаштування зображений на рисунку 19.

Редактировать файл - nginx.conf

Имя файла `/etc/nginx/nginx.conf`

Кодировка `UTF-8`

Шрифт `12px` Режим `text` Тема `Chrome`  Нумерация Перенос `Off`

```

1 user www-data;
2 worker_processes 8;
3
4 error_log /var/log/nginx/error.log crit;
5 pid /var/run/nginx.pid;
6
7
8
9 events {
10     worker_connections 4096;
11     use epoll;
12     multi_accept on;
13 }
14
15 http {
16     open_file_cache max=200000 inactive=20s;
17     open_file_cache_valid 30s;
18     open_file_cache_min_uses 2;
19     open_file_cache_errors on;
20
21     include /etc/nginx/mime.types;
22     default_type application/octet-stream;
23
24     log_format main '$remote_addr - $remote_user [$time_local] "$request" '
25                   '$status $body_bytes_sent "$http_referer" '
26                   '"$http_user_agent" "$http_x_forwarded_for"';
27
28     access_log off;
29
30     sendfile on;
31     tcp_nopush on;
32     tcp_nodelay on;
33
34     reset_timedout_connection on;
35
36     client_body_timeout 10;
37     send_timeout 2;
38
39     keepalive_timeout 60;
40     keepalive_requests 100000;
41
42     gzip on;
43     gzip_min_length 10240;
44     gzip_comp_level 1;
45

```

Сохранить

Рисунок 18 – Файл налаштування Nginx

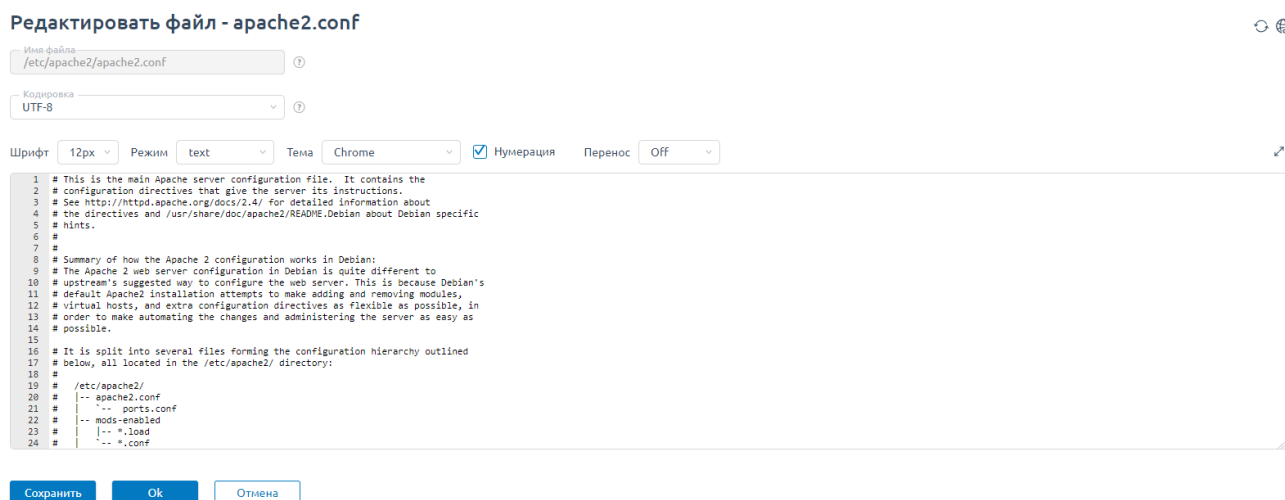


Рисунок 19 – Файл налаштування Apache

Основне налаштування сервера проведено. Можна перейти до розгортання бази даних.

### 3.1.2 Створення бази даних

За допомогою РМА створимо базу даних та всі необхідні таблиці. На рисунку 20 зображений процес створення БД.

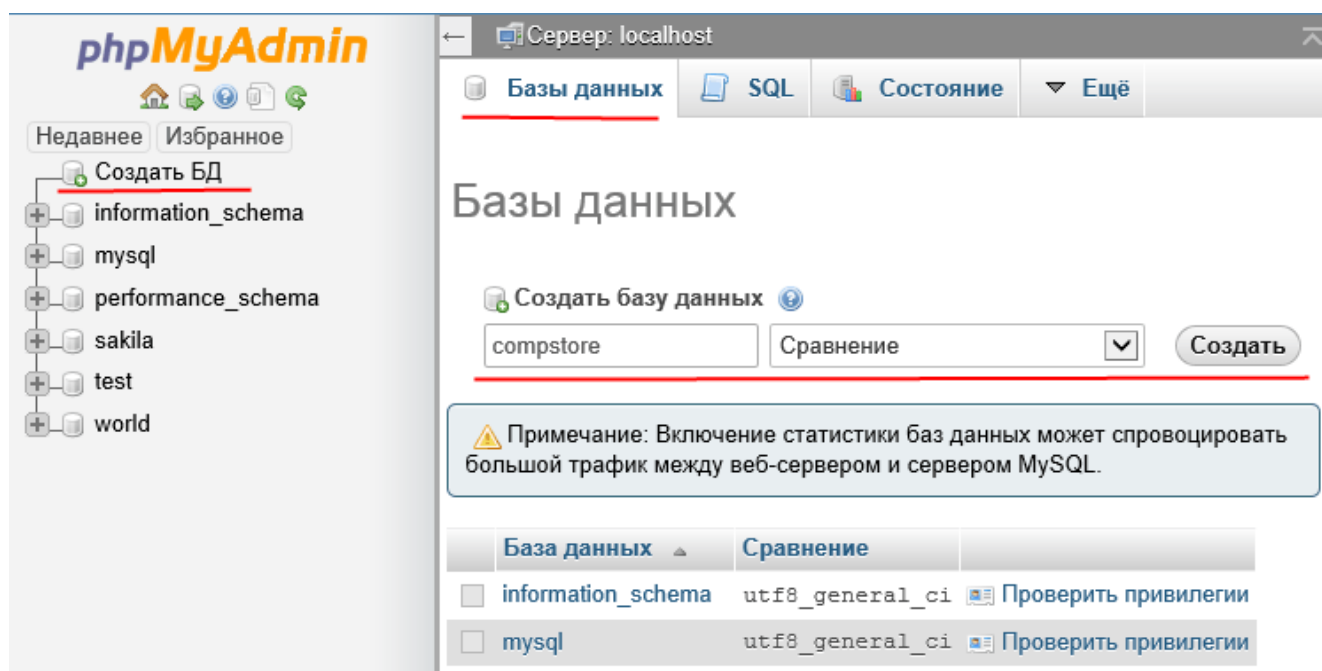


Рисунок 20 – Створення бази даних

Далі необхідно створити початкові таблиці в грі, беручи до уваги всі аспекти, функціонал та логіку роботи гри. На рисунку 21 зображений процес створення таблиці в БД.

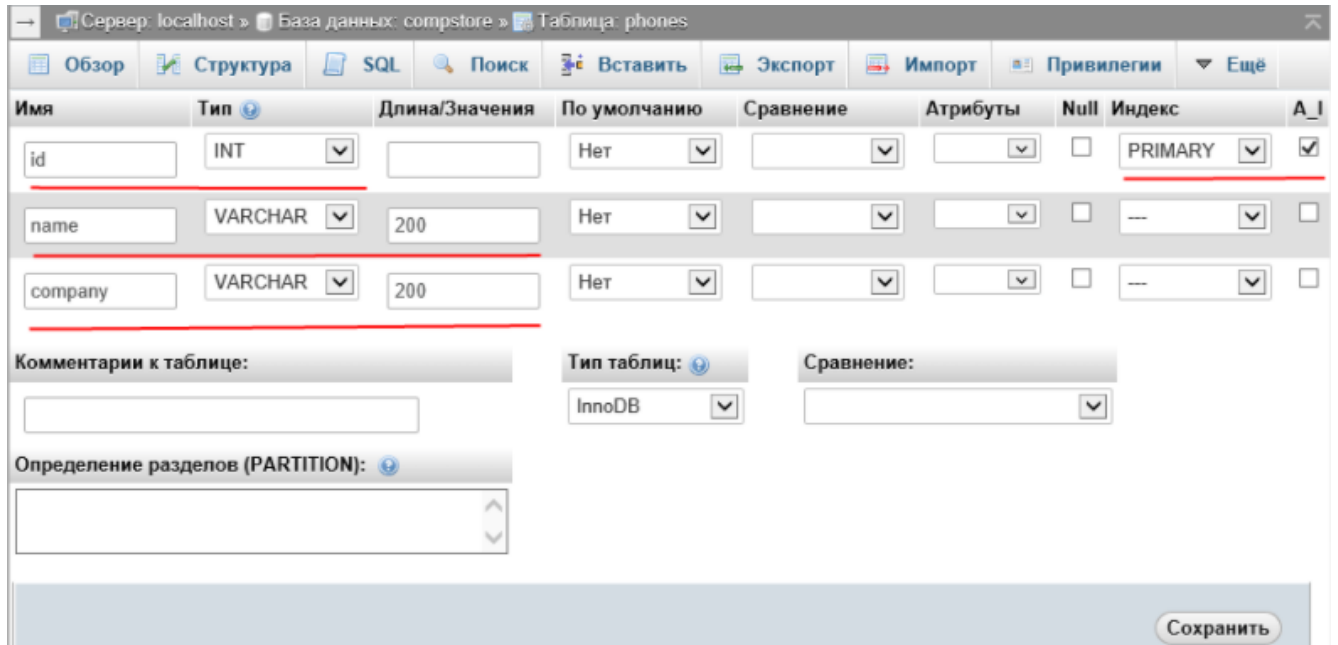


Рисунок 21 – Створення таблиці в РМА

На рисунку 22 зображена діаграма загальної картини вигляду БД уже після створення всіх необхідних таблиць та їх налаштування.

Після повного налаштування всіх таблиць ми можемо приступити до розробки серверної частини гри.

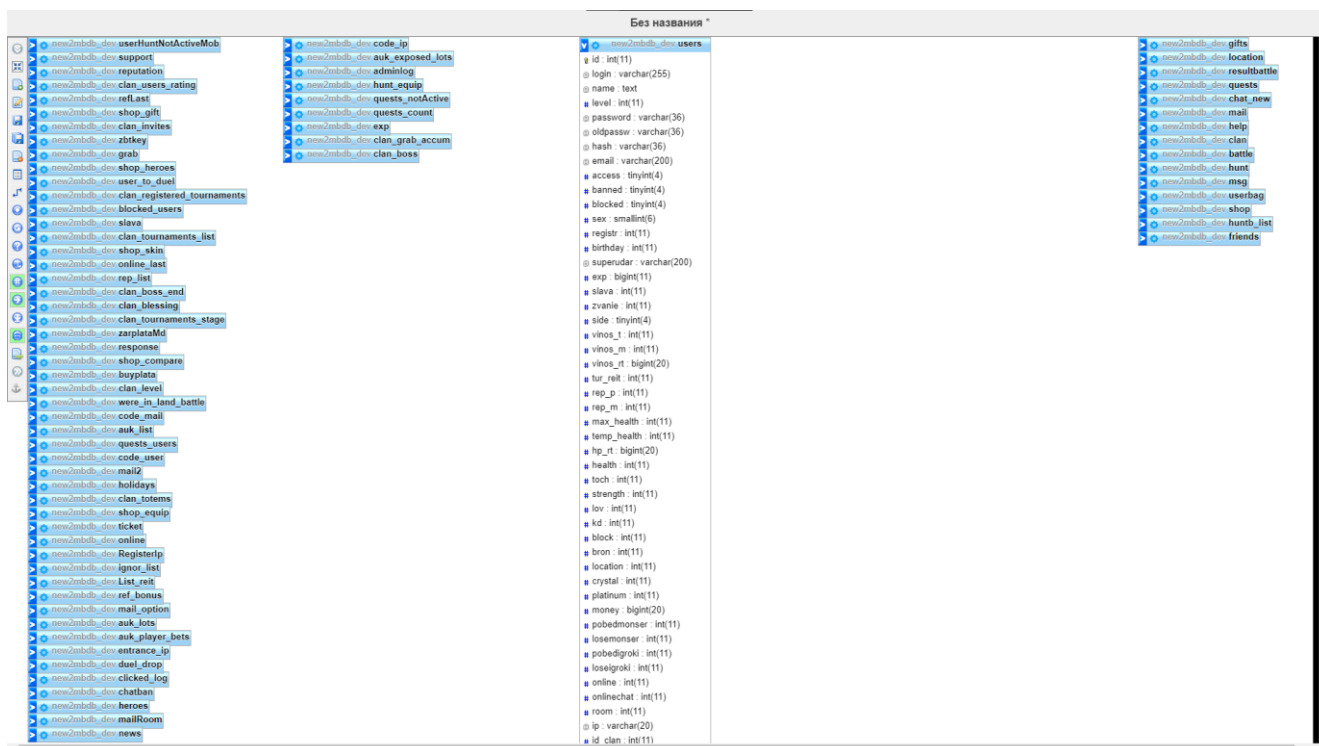


Рисунок 22 - Діаграма структури БД

### 3.1.3 Серверна частина

В першу чергу потрібно створити файл для підключення та використання БД в коді гри. Для цього створюємо папку “system”, в якій буде міститись файл “connect.php”. Тепер, якщо потрібно використати десь базу даних – просто використовуємо include “connect.php”. Це дає змогу використовувати у всіх файлах водночас. Приклад підключення БД зображений на рисунку 23.

```

connect.php
system > connect.php
1  <?php
2  $par1 = "localhost";
3  $par2 = "admin"; //login
4  $par3 = "password"; //pass
5  $par4 = "db_name"; //db
6
7  ini_set('error_reporting', E_ALL);
8  ini_set('display_errors', 1);
9  ini_set('display_startup_errors', 1);
10
11 $mc = new mysqli($par1, $par2, $par3) or die(header("Location: /disconnect.php?error=2"));
12 $mc->set_charset("utf8mb4");
13 $mc->select_db($par4) or die(header("Location: /disconnect.php?error=3"));

```

Рисунок 23 – Підключення БД в файлі

Далі створюємо потрібні файли для гри. Для цього будемо використовувати раніше створене підключення до БД. Таким чином будується структура серверної частини гри. На рисунку 24 наглядний приклад створених файлів та папок.

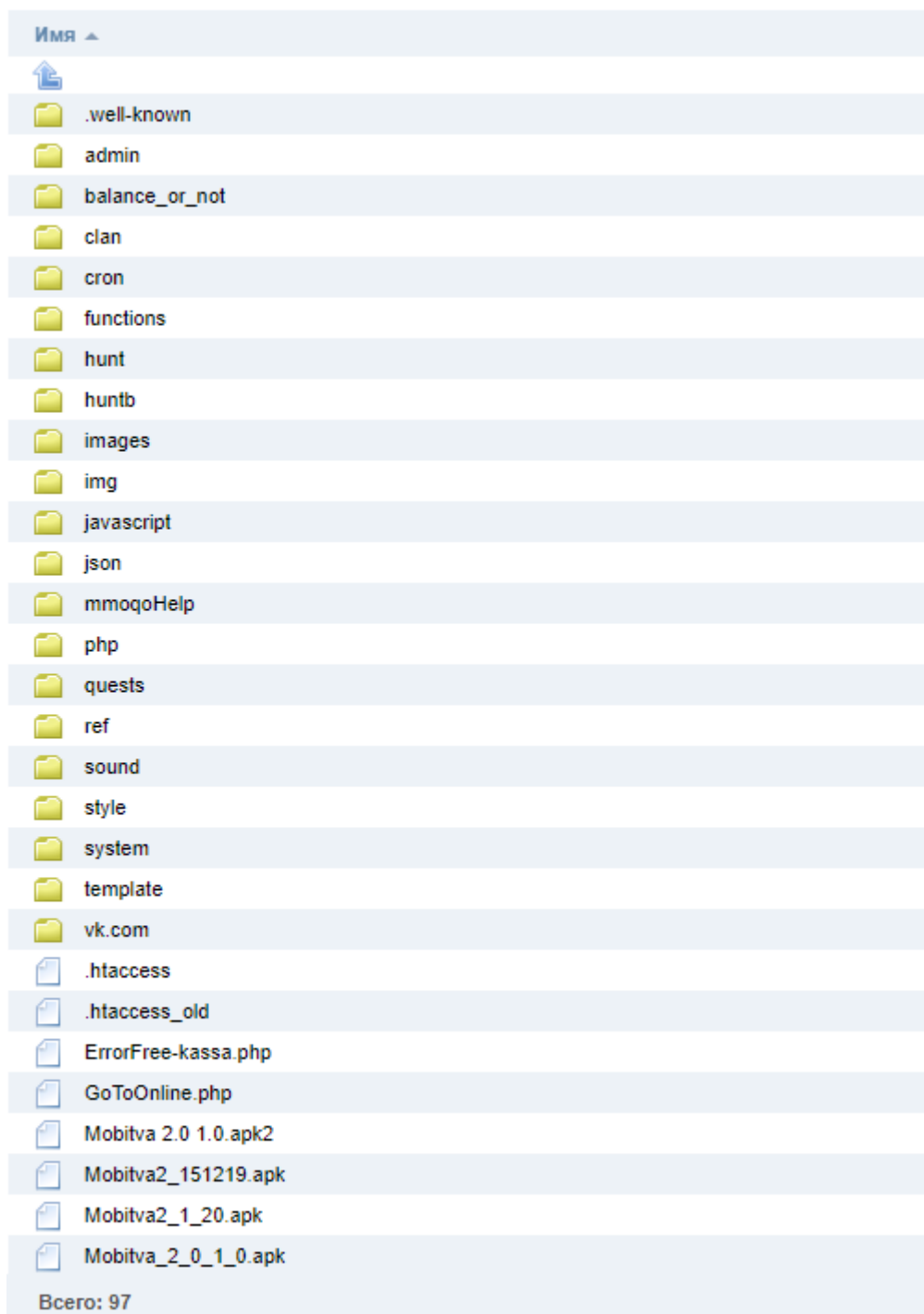


Рисунок 24 – Список створених файлів

### 3.1.4 Клієнтська частина

Для відображення картинки створюємо основну функцію, яка буде перемикати сторінки, елементи меню та відкривати потрібні вкладки. На рисунку 25 зображена саме така функція.

```

header.js
javascript > header.js > ...
474
475 showContent = function (link, isAdd, custom_url) {
476
477     if (link !== 'back') {
478         if (isAdd === true) {
479             if (typeof (custom_url) !== "undefined") {
480                 goBack('add', custom_url);
481             } else {
482                 goBack('add', link);
483             }
484         }
485     }
486
487     if (MyLib.loaded === 0) {
488         MyLib.loaded = 1;
489         MyLib.loaded1 = 1;
490         if (MyLib.startapp != 0 && !$("#img").is(".imgLoading")) {
491             document.body.appendChild(imgLoading);
492         } else {
493             MyLib.startapp = 1;
494         }
495     }
496
497     setTimeout(function () {
498         clearTimeout(MyLib.setTimeoutFoot);
499         clearTimeout(MyLib.setTimeoutHuntB);
500         MyLib.intervaltimer.forEach(clearInterval);
501         MyLib.setTimeout.forEach(clearTimeout);
502         MyLib.battleIntervalTimer.forEach(clearInterval);
503         MyLib.battleSetTimeout.forEach(clearTimeout);
504         $.ajax({
505             type: "POST",
506             url: "" + link,
507             dataType: "text",
508             data: {
509                 glbool: 1,
510             },
511             success: function (data) {
512                 if (link.match(/main/ig)) {
513                     MyLib.links = [];
514                 } else {
515                     var regex = new RegExp('chat.php*');
516                     if (regex.test(link) && regex.test(MyLib.links[MyLib.links.length - 1])) {
517                         showContent("/main.php");
518                     } else {
519                         if (MyLib.links.length > 10) {
520                             MyLib.links = [];
521                         } else {
522                             MyLib.links.push(link)
523                         }
524                     }
525                 }
526             }
527         });
528     });
529 }

```

Рисунок 25 – Функція для перемикання сторінок гри



Також верстаємо всі потрібні сторінки, оформлюємо їх за допомогою стилів та додаємо функціонал, використовуючи JavaScript. В грі доступний аукціон, де гравці можуть торгуватись один з один (Додаток А). Також, одним з головних аспектів гри є клани (Додаток Б).

## 3.2 Тестування та розробка додатку для Google Play

### 3.2.1 Розробка Android додатку

Для запуску додатку в Google Play потрібно написати Android-додаток. Для цього розробляється за допомогою Android Studio невеликий клієнт, за допомогою якого буде проходити підключення до серверу гри. Код зображений на рисунку 26.

```
package com.mmoquest.mmoquest; Untitled-1 ●
1 package com.mmoquest.mmoquest;
2 import androidx.appcompat.app.AppCompatActivity;
3 import android.content.Intent;
4 import android.graphics.Bitmap;
5 import android.net.Uri;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.view.WindowManager;
9 import android.webkit.CookieManager;
10 import android.webkit.RenderProcessGoneDetail;
11 import android.webkit.WebSettings;
12 import android.webkit.WebView;
13 import android.webkit.WebViewClient;
14 import android.widget.ImageView;
15 import android.widget.ProgressBar;
16 import android.widget.RelativeLayout;
17 import android.widget.Toast;
18 import com.google.android.gms.tasks.OnSuccessListener;
19 import com.google.firebase.iid.FirebaseInstanceId;
20 import com.google.firebase.iid.InstanceIdResult;
21
22 public class MainActivity extends AppCompatActivity{
23
24     public String uri = "https://mmoquest.com/";
25     public String apiUri = "android";
26     public CheckVersion checkVersion;
27     public RelativeLayout relativeLayout;
28
29     private WebView webView;
30     private ImageView logo;
31     private ProgressBar progressBar;
32     private boolean webViewStart = true;
33     private String fireToken;
34     private CookieManager cookieManager;
35
36     @Override
37     protected void onCreate(Bundle savedInstanceState) {
38         super.onCreate(savedInstanceState);
39         setContentView(R.layout.activity_main);
40
41         webView = (WebView) findViewById(R.id.webview);
42         logo = (ImageView) findViewById(R.id.logo);
43         progressBar = (ProgressBar) findViewById(R.id.progressBar);
44         relativeLayout = (RelativeLayout) findViewById(R.id.frame);
45
46         progressBar.setVisibility(View.GONE);
47
48         getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
49         getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_ADJUST_PAN);
50         new FullscreenKeyboard(this, this);
```

Рисунок 26 – Код Android-додатку

### 3.2.2 Google Play Console

Далі, за допомогою інструментів розробника, а саме Google Play Console, здійснюється додавання гри в Google Play. Це кабінет, в якому зберігаються всі додатки [20]. Потрібно очікувати декілька днів поки додаток пройде модерацію та почне індексуватись в глобальному пошуку площадки. На рисунку 27 зображена панель Google Play Console з успішно доданою грою та мінімальною статистикою по завантаженням з відгуками.

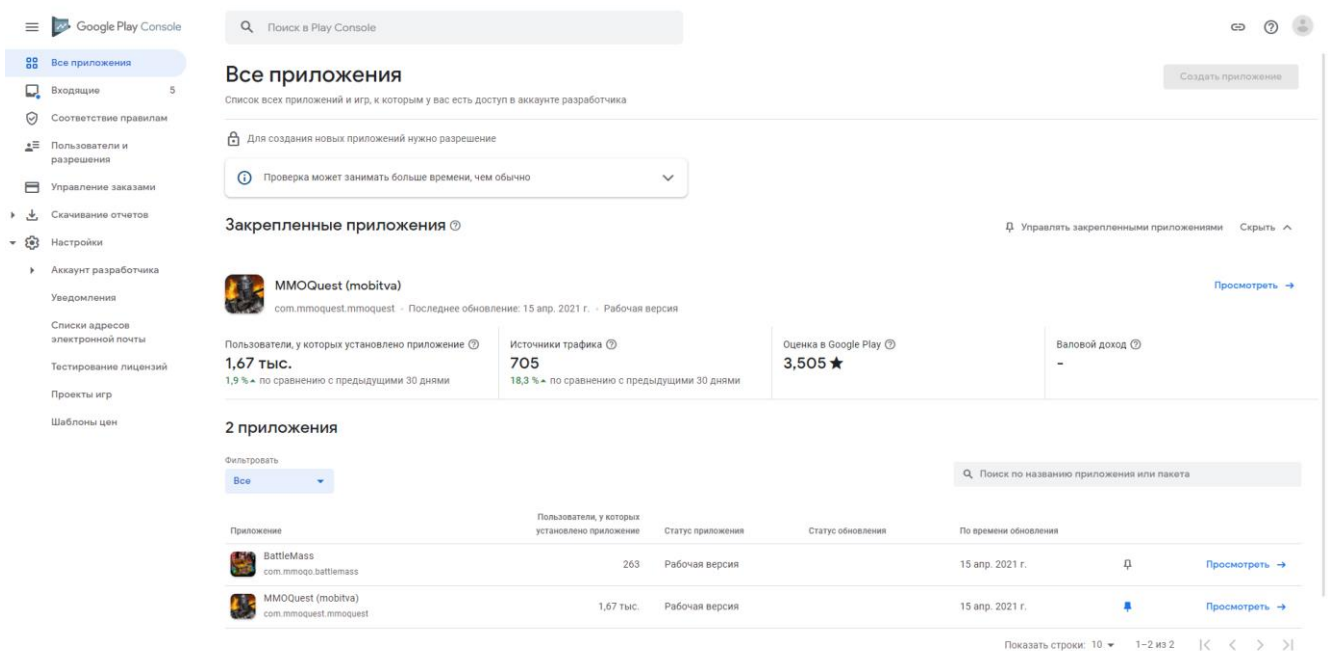


Рисунок 27 – GUI Google Play Console

### 3.2.3 Тестування гри

Тестування грає велику роль в життєвому циклі будь-якого додатку. Не виключенням є і онлайн гра, де важливо перевірити кожен деталь, щоб кінцевий клієнт був задоволений. Потрібно зауважити, що рейтинг додатку формується на основі відгуків користувачів, які можуть бути не задоволені роботою програми.

Існує безліч методів тестування додатків:

1. Тестування верстки;
2. Функціональне тестування;
3. Usability-тестування;

4. Тестування сумісності;
5. Тестування продуктивності;
6. Тестування безпеки;
7. Регресійне тестування.

Тестування верстки відповідає за візуальну частину, до якої відноситься невірне з'єднання блоків або елементів гри, масштабування, зміна ширини екрану.

Далі потрібно провести тестування гри.

1. Масштабування (50% - 150%). На рисунках 27 та 28 зображено тестування верстки;
2. Багатолатформність. На рисунку 29 зображений вигляд з мобільного пристрою на базі Android через додаток гри. В свою чергу, на рисунках 27 та 28 продемонстрована робота в системі Windows через браузер Chrome;
3. Правильна робота всіх сторінок гри, переходів між ними, перевірка базового функціоналу та елементів;
4. Зручність використання та взаємодії з GUI;
5. Продуктивність на різних пристроях;
6. Безпека даних, серверної частини та особистої інформації користувачів.

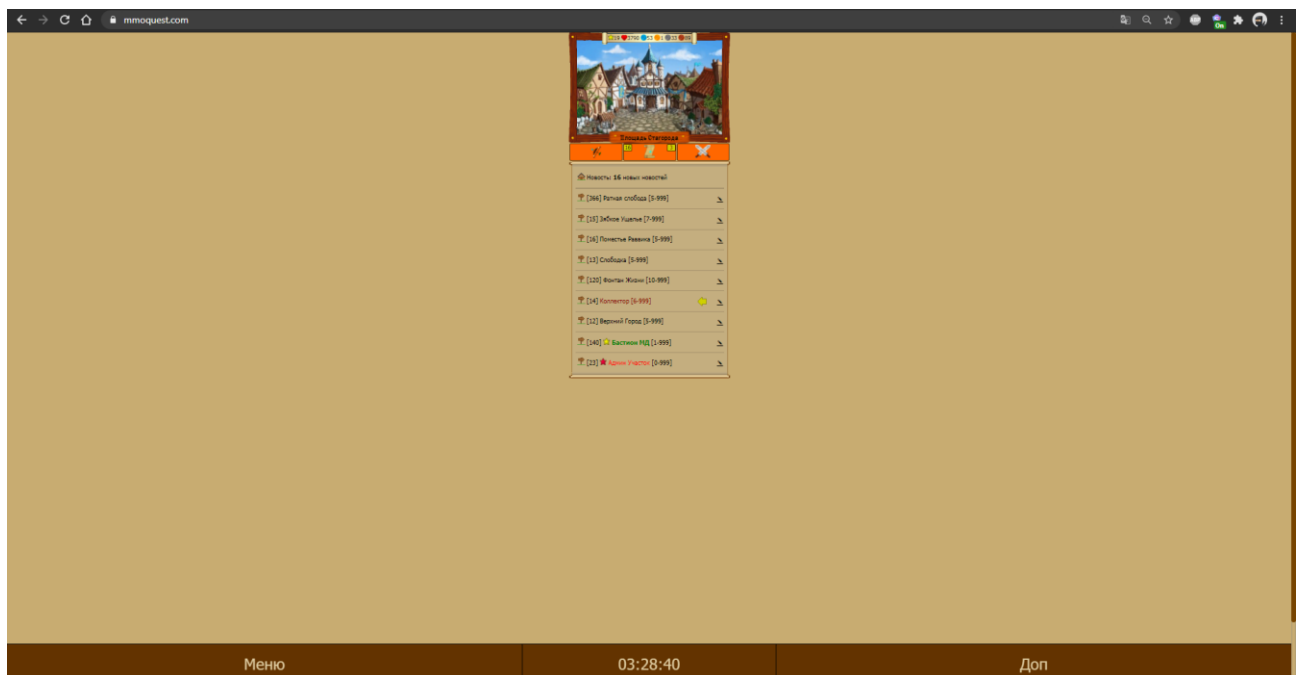


Рисунок 27 – Тестування масштабування 50%



Рисунок 28 – Тестування масштабування 150%



Рисунок 29 – Тестування додатку гри на операційній системі Android

### 3.3 Огляд можливостей гри

Використовуючи сервіс MindMeister створимо карту гри. Це сервіс по візуалізації мап у вигляді “хмарок” [21]. Він чудово підходить для відображення сторінок гри та базового опису їх можливостей. На рисунку 30 зображено карту гри.

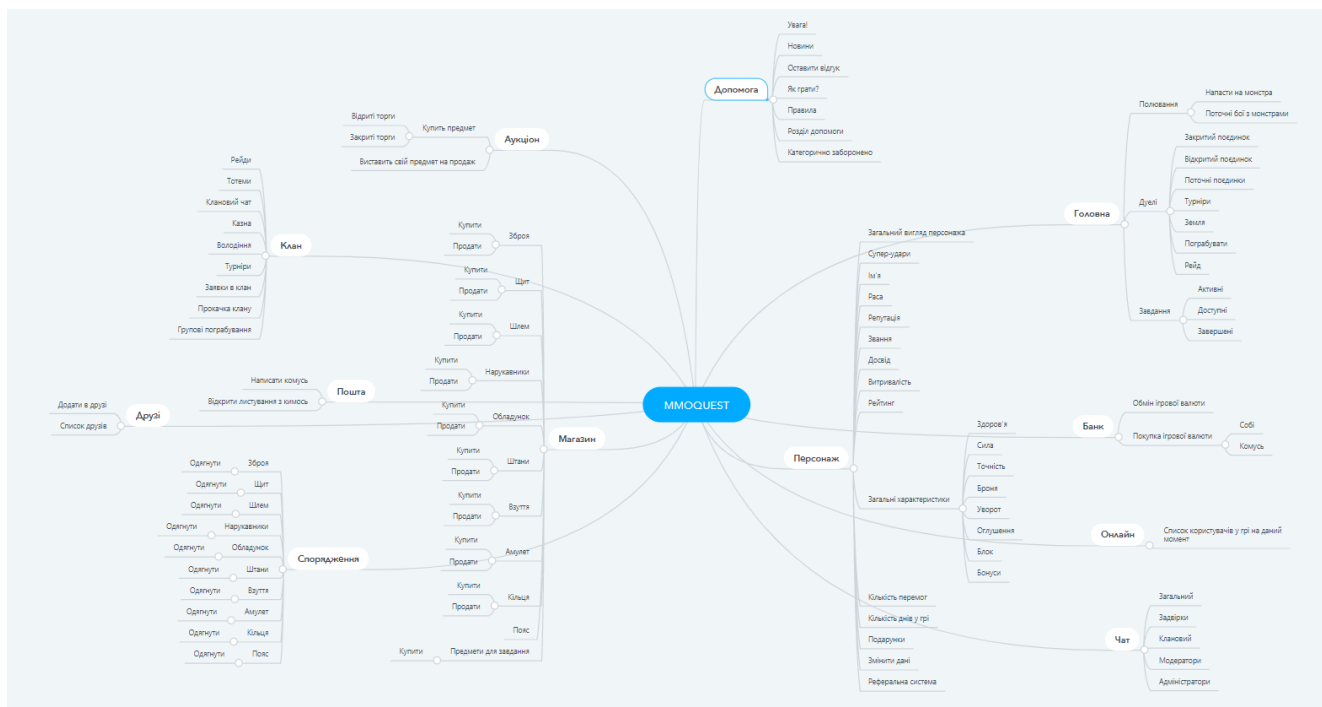


Рисунок 30 – Карта гри

При запуску гри відображається сторінка входу. На рисунку 31 зображено сторінку авторизації.

Якщо відсутній акаунт, то потрібно його створити. Для цього потрібно натиснути кнопку “Почати гру”. Далі будемо запропоновано обрати націю та стать. По сюжету гри існує дві ворогуючі нації, які воюють за землі. На рисунку 32 зображений вибір персонажа.

Після успішного проходження навчання користувачеві буде запропоновано змінити ім'я, тимчасовий логін та пароль персонажа, а також вказати поштову адресу. На рисунку 33 зображений процес реєстрації.

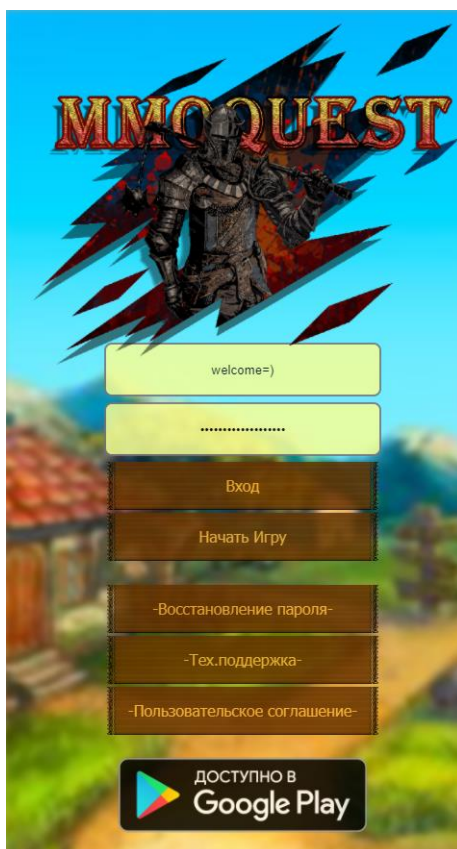


Рисунок 31 – Сторінка авторизації

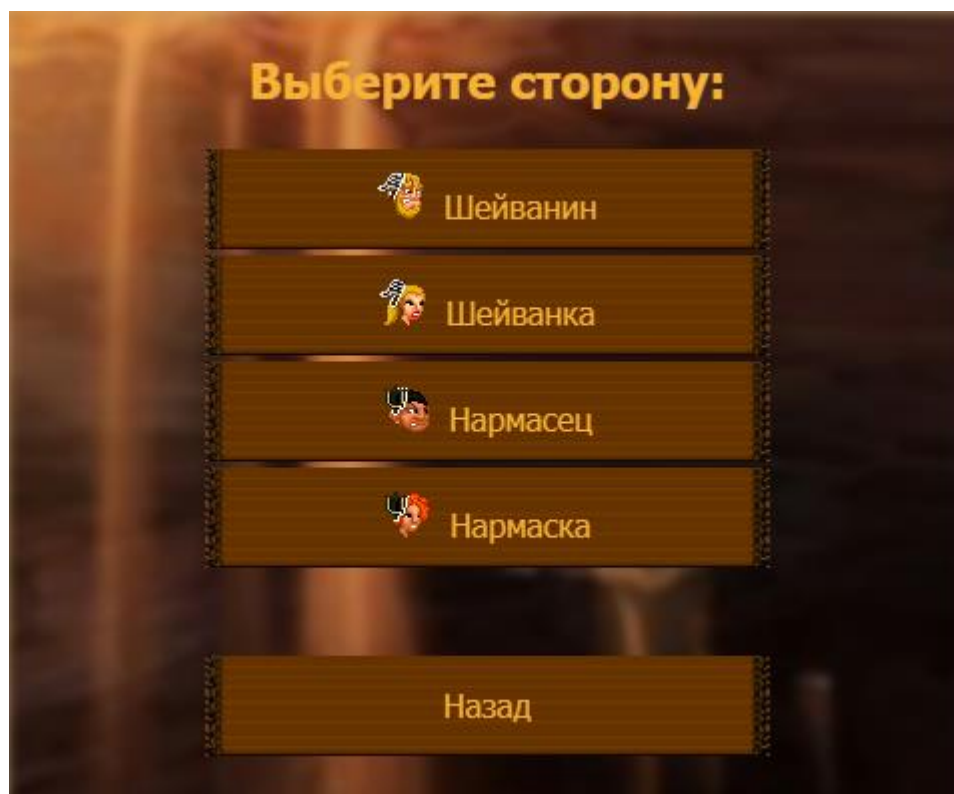
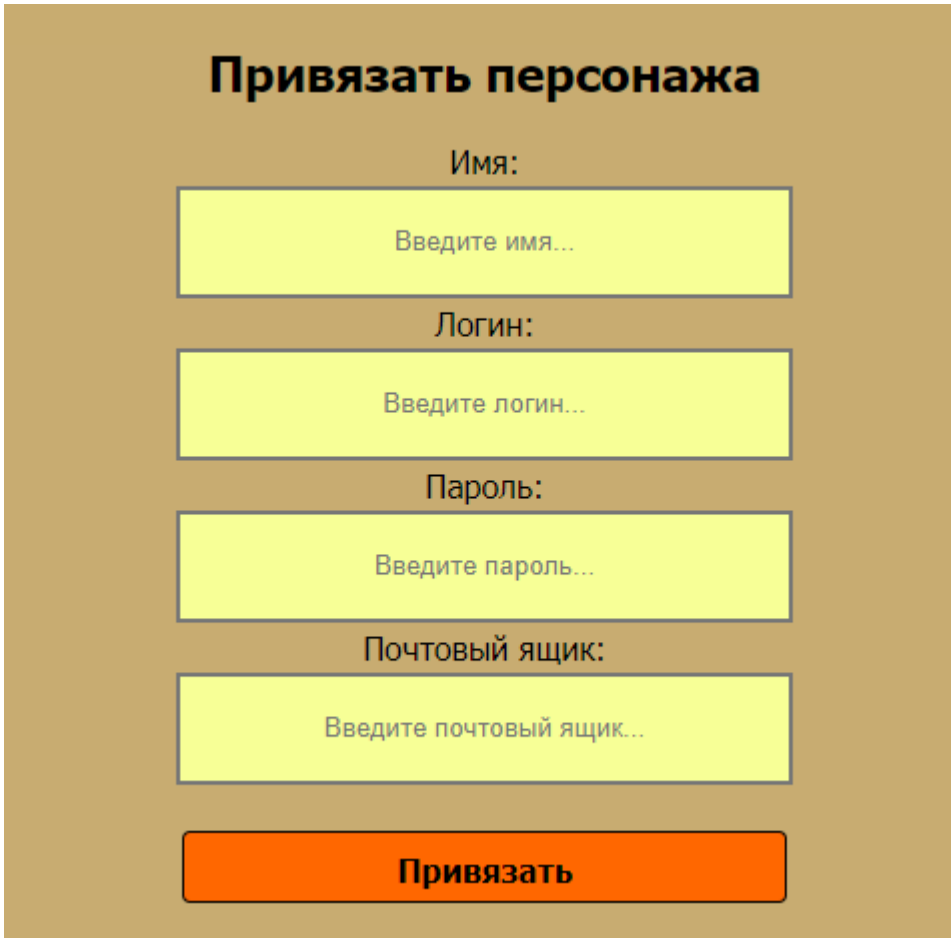


Рисунок 32 – Вибір нації та статі персонажа



**Привязать персонажа**

Имя:  
Введите имя...

Логин:  
Введите логин...

Пароль:  
Введите пароль...

Почтовый ящик:  
Введите почтовый ящик...

**Привязать**

Рисунок 33 – Процесс реєстрації

Після успішного завершення реєстрації користувачеві буде запропоновано знову увійти в гру щоб перевірити достовірність введених даних.

Якщо введені дані при авторизації були вірні – користувач попадає на головну сторінку гри, де йому доступне головне вікно локації, кнопки управління, список доступних локацій та кількість ігрової валюти. На рисунку 34 зображено головну сторінку гри.

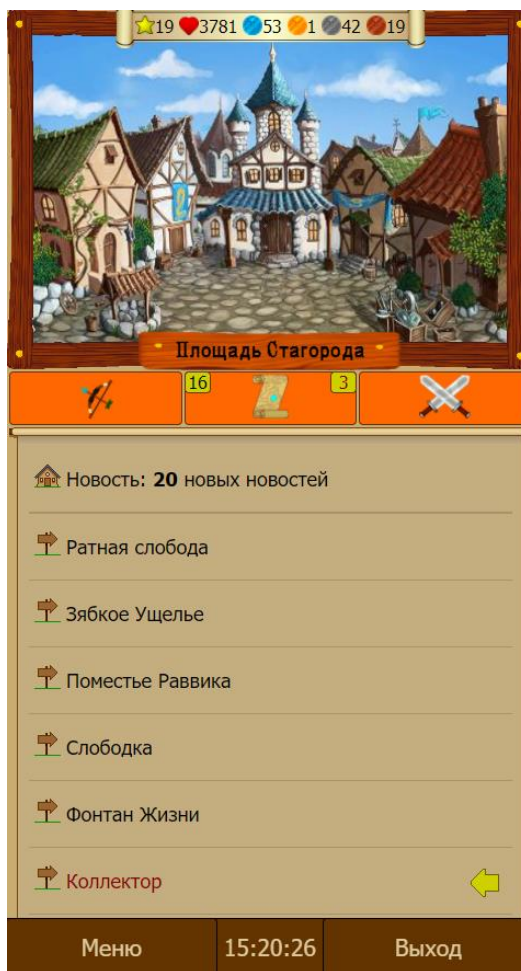


Рисунок 34 – Головна сторінка гри

Далі користувачеві доступні наступні розділи гри:

1. Клан. Це об'єднання гравців у групу для виконання спільних завдань, проходження важких мобів та війни за землі;
2. Топ. Це список найкращих гравців по їх рівню;
3. Аукціон. Це торгова площадка для обміну гравцями предметами між собою;
4. Магазин. Це місце для покупки або продажу ігрових предметів за ігрову валюту;
5. Банк. Це система обміну та покупки ігрової валюти;
6. Онлайн. Це список гравців, який в даний момент знаходяться в мережі;
7. Пошта. Це особистий обмін повідомленнями між гравцями;
8. Друзі. Це сторінка зі списком друзів, яких гравець відмітив;



9. Чат. Це загальне місце для обміну повідомленнями. Існує декілька видів:
  - 9.1 Загальний. Чат для спілкування по ігровим моментам;
  - 9.2 Задвірки. Чат, в якому є можливість спілкуватись на будь-які теми і діють свої правила;
  - 9.3 Клановий. Чат для обміну повідомленнями між гравцями в клані;
  - 9.4 Модератори. Прихований чат для спілкування між модераторами, який видно тільки для них.
10. Обладунки. Місце для переодягнення предметів персонажа. Доступно декілька розділів, а саме:
  - 10.1 Зброя;
  - 10.2 Щит;
  - 10.3 Шлем;
  - 10.4 Нарукавники;
  - 10.5 Штани;
  - 10.6 Взуття;
  - 10.7 Амулет;
  - 10.8 Кільця;
  - 10.9 Пояс.
11. Персонаж. Розділ із загальною інформацією про персонажа, а саме:
  - 11.1 Вигляд персонажа;
  - 11.2 Супер-удари;
  - 11.3 Ім'я;
  - 11.4 Нація;
  - 11.5 Репутація;
  - 11.6 Звання;
  - 11.7 Досвід;
  - 11.8 Витривалість;
  - 11.9 Рейтинг;
  - 11.10 Загальні характеристики:
    - 11.10.1 Здоров'я;

- 11.10.2 Сила;
- 11.10.3 Точність;
- 11.10.4 Броня;
- 11.10.5 Пригучість;
- 11.10.6 Оглушення;
- 11.10.7 Блок;
- 11.10.8 Бонуси.
- 11.11 Кількість перемог;
- 11.12 Кількість днів у грі;
- 11.13 Подарунки;
- 11.14 Змінити дані;
- 11.15 Реферальна система.

## 12. Допомога. Розділ із загальною інформацією про гру та підказками:

- 12.1 Увага!;
- 12.2 Новини;
- 12.3 Залишити відгук;
- 12.4 Як грати?;
- 12.5 Правила;
- 12.6 Розділ допомоги;
- 12.7 Категорично заборонено.

Управління боєм відбувається завдяки натисканню на три головні кнопки:

1. Вгору – удар по голові;
2. По-середині – удар по тілу;
3. Вниз – удар по ногам.

Завдяки комбінуванню цих кнопок формуються супер-удари, які наносять в декілька раз більше шкоди противнику. На рисунку 35 зображений бій з мобом. На рисунку 36 зображений приклад бою з іншим гравцем.



Рисунок 35 – Бій з мобом



Рисунок 36 – Бій з іншим гравцем

### 3.4 Огляд рейтингу гри

За допомогою Google Play Console, оцінимо скільки людей завантажило додаток, який його загальний рейтинг та як сторінка гри виглядає в Google Play. На рисунку 37 зображені всі ці дані.

**MMOQuest (mobitva)**  
 ММОQO Ролевые  
 10+  
 Это приложение можно скачать на все ваши устройства.  
 Вы можете открыть доступ к этому контенту членам вашей семьи. Подробнее о Семейной библиотеке.

Обновлено: 15 апреля 2021 г.  
 Размер: 5,0M  
 Текущая версия: 0.27  
 Требуемая версия: Android 4.0 и выше  
 Количество установок: 5 000+  
 Возрастные ограничения: 10+  
 Фэнтези с насилием  
 Подробнее...

Установлено



Рисунок 37 – Відображення рейтингу додатку

## ВИСНОВКИ

На сьогодні кількість онлайн ігор постійно зростає величезними темпами, це зумовлене збільшенням “гаджефікації” населення. Все більше і більше людей проводить свій вільний час в смартфонах, саме тому потрібно збільшувати кількість контенту, який вони можуть споживати.

В рамках випускний кваліфікаційної роботи були виконані поставлені завдання.

По-перше, проведений аналіз ринку мобільних ігор, знайдено найдена найпопулярніша гра для дослідження всіх її переваг та недоліків. Детально розписаний вибір технологій для розробки, а саме PHP, JavaScript, CSS, HTML, MySQL, Ubuntu, чому саме вони та їх основні можливості.

По-друге, проведений детальний опис розробки гри з додаванням рисунків, таблиць та діаграм.

По-третє, здійснене тестування за основними критеріями якості.

Таким чином, результатом випускної кваліфікаційної роботи є повністю реалізована мобільна онлайн гра в жанрі MMORPG, написана з використанням технологій PHP та JS.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Моніторинг мобільних додатків [Електронний ресурс] – Ринок мобільних додатків 2020 - Режим доступу: <https://asomobile.net/blog/digest2020/>
2. Betting apps and services [Електронний ресурс] - Google Play vs App Store - Режим доступу: <https://www.safebettingsites.com/2020/10/13/google-play-hit-28-3bn-downloads-in-q3-2020-3x-more-than-app-store/>
3. Visual Studio Code [Електронний ресурс] – Why VS Code? - Режим доступу: <https://code.visualstudio.com/learn>
4. Ubuntu [Електронний ресурс] – About - Режим доступу: <https://ubuntu.com/appliance/about>
5. Все про пристрої та технології [Електронний ресурс] – Що таке Linux - Режим доступу: <http://ipkey.com.ua/faq/924-linux.html>
6. Apache Http Server Project [Електронний ресурс] – About - Режим доступу: [https://httpd.apache.org/ABOUT\\_APACHE.html](https://httpd.apache.org/ABOUT_APACHE.html)
7. Панель управління веб-сервером [Електронний ресурс] – Можливості - Режим доступу: <https://www.ispsystem.ru/software/ispmanager>
8. NGINX [Електронний ресурс] – Про - Режим доступу: <https://nginx.org/ru/>
9. MySQL [Електронний ресурс] – Why MySQL? - Режим доступу: <https://www.mysql.com/why-mysql/>
10. Tech blog [Електронний ресурс] – MySQL - Режим доступу: <https://dizballanze.com/ru/mysql-obzor-tipov-tablits/>
11. PhpMyAdmin [Електронний ресурс] – Docs - Режим доступу: <https://docs.phpmyadmin.net/en/latest/>
12. PHP [Електронний ресурс] – Documentation - Режим доступу: <https://www.php.net/manual/ru/index.php>
13. MDN Web Docs [Електронний ресурс] – HTML - Режим доступу: <https://developer.mozilla.org/ru/docs/Web/HTML>
14. MDN Web Docs [Електронний ресурс] – CSS - Режим доступу: <https://developer.mozilla.org/ru/docs/Web/CSS>

- 15.MDN Web Docs [Электронный ресурс] – JavaScript - Режим доступа:  
<https://developer.mozilla.org/ru/docs/Web/JavaScript>
- 16.Stack Overflow [Электронный ресурс] – Рейтинг - Режим доступа:  
<https://ru.stackoverflow.com/>
- 17.JQuery [Электронный ресурс] – About - Режим доступа:  
<https://learn.jquery.com/about/>
- 18.C# [Электронный ресурс] – About - Режим доступа:  
<https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- 19.Let's Encrypt [Электронный ресурс] – Про нас - Режим доступа:  
<https://letsencrypt.org/ru/about/>
- 20.Google Play Console [Электронный ресурс] – About - Режим доступа:  
<https://support.google.com/googleplay/android-developer/answer/6112435?hl=ru>
- 21.MindMeister [Электронный ресурс] – Про нас - Режим доступа:  
<https://www.mindmeister.com/ru>

## Додаток А

### Основний набір функціоналу аукціону клієнтської частини

```

shopsort1 = function (elem) {
    $(".shopminiblock").css({fontWeight:          "normal",
fontSize: "16px", textDecoration: "none"});
    $(elem).css({fontWeight:  "bold",   fontSize:  "14px",
textDecoration: "underline"});
    $(".shopblock").each(function () {
        $(this).show();
    });
    MyLib.ticks = 0;
};

shopsort2 = function (elem) {
    $(".shopminiblock").css({fontWeight:          "normal",
fontSize: "16px", textDecoration: "none"});
    $(elem).css({fontWeight:  "bold",   fontSize:  "14px",
textDecoration: "underline"});
    $(".shopblock").each(function () {
        $(this).attr("money")    <=    MyLib.userMoney    &&
$(this).attr("platina")        <=    MyLib.userPlatina    &&
$(this).attr("level") <= MyLib.userLevel ? $(this).show() :
$(this).hide();
    });
    MyLib.ticks = 1;
};

shopsort3 = function (elem) {
    $(".shopminiblock").css({fontWeight:          "normal",
fontSize: "16px", textDecoration: "none"});
    $(elem).css({fontWeight:  "bold",   fontSize:  "14px",
textDecoration: "underline"});
    $(".shopblock").each(function () {
        $(this).attr("level")    !=    MyLib.userLevel    ?
$(this).hide() : $(this).show();
    });
    MyLib.ticks = 2;
};

shopsort4 = function (elem) {

```



```

    $(".shopminiblock").css({fontWeight:          "normal",
fontSize: "16px", textDecoration: "none"});
    $(elem).css({fontWeight:  "bold",  fontSize:  "14px",
textDecoration: "underline"});
    $(".shopblock").each(function () {
        $(this).attr("myFriends") == 1 ? $(this).show() :
$(this).hide();
    });
    MyLib.ticks = 3;
};

shopsort5 = function (elem) {
    $(".shopminiblock").css({fontWeight:          "normal",
fontSize: "16px", textDecoration: "none"});
    $(elem).css({fontWeight:  "bold",  fontSize:  "14px",
textDecoration: "underline"});
    $(".shopblock").each(function () {
        $(this).attr("myBets") == 1 ? $(this).show() :
$(this).hide();
    });
    MyLib.ticks = 4;
};

shopsort6 = function (elem) {
    $(".shopminiblock").css({fontWeight:          "normal",
fontSize: "16px", textDecoration: "none"});
    $(elem).css({fontWeight:  "bold",  fontSize:  "14px",
textDecoration: "underline"});
    $(".shopblock").each(function () {
        $(this).attr("myLots") == 1 ? $(this).show() :
$(this).hide();
    });
    MyLib.ticks = 5;
};

shopsortStyle = function (elem) {
    $(".sortStyle").css({fontWeight:  "normal",  fontSize:
"16px", textDecoration: "none"});
    $(elem).css({fontWeight:  "bold",  fontSize:  "14px",
textDecoration: "underline"});
    MyLib.style = $(elem).attr("value");
    if (MyLib.ticks == 0) {
        $(".shopblock").each(function () {
            $(this).attr("stail") == MyLib.style ||
MyLib.style == 5 ? $(this).show() : $(this).hide();
        });
    }
};

```

```

    }
    if (MyLib.ticks == 1) {
        $(".shopblock").each(function () {
            $(this).attr("stail") == MyLib.style    &&
$(this).attr("money") <= MyLib.userMoney    &&
$(this).attr("platina") <= MyLib.userPlatina    &&
$(this).attr("level") <= MyLib.userLevel ||
                MyLib.style == 5    &&
$(this).attr("money") <= MyLib.userMoney    &&
$(this).attr("platina") <= MyLib.userPlatina    &&
$(this).attr("level") <= MyLib.userLevel ? $(this).show() :
$(this).hide();
        });
    }
    if (MyLib.ticks == 2) {
        $(".shopblock").each(function () {
            $(this).attr("stail") == MyLib.style    &&
$(this).attr("level") == MyLib.userLevel || MyLib.style ==
5    && $(this).attr("level") == MyLib.userLevel    ?
$(this).show() : $(this).hide();
        });
    }

    if (MyLib.ticks == 3) {
        $(".shopblock").each(function () {
            $(this).attr("stail") == MyLib.style &&
                $(this).attr("myFriends") == 1 ||
                MyLib.style == 5 &&
                $(this).attr("myFriends") == 1
                ? $(this).show() : $(this).hide();
        });
    }

    if (MyLib.ticks == 4) {
        $(".shopblock").each(function () {
            $(this).attr("stail") == MyLib.style &&
                $(this).attr("myBets") == 1 ||
                MyLib.style == 5 &&
                $(this).attr("myBets") == 1
                ? $(this).show() : $(this).hide();
        });
    }
}
};

```

## Додаток Б

### Основний код роботи кланової системи на клієнтській частині

```

var clanImgs = [];
function AddClanImg(jsons, img) {
    if (typeof(clanImgs[JsonToArray(jsons)]) ===
"undefined") {
        clanImgs[JsonToArray(jsons)]=[img];
    }
}

function JsonToArray(jsony) {
    return
    "id"+jsony[0]["id"]+"r"+jsony[0]["r"]+"g"+jsony[0]["g"]+"b"
    +jsony[0]["b"]
    +jsony[1]["id"]+"r"+jsony[1]["r"]+"g"+jsony[1]["g"]+"b"
    +jsony[1]["b"]
    +jsony[2]["id"]+"r"+jsony[2]["r"]+"g"+jsony[2]["g"]+"b"
    +jsony[2]["b"];
}

function CreateClanLogo(json, idHtmlElement, clanId) {
    var imgClan = [];

    try {
        imgClan = JSON.parse(json);
    } catch(e) {
        json
        ' [{"id":0,"r":0,"g":0,"b":0}, {"id":0,"r":0,"g":0,"b":0}, {"i
d":0,"r":0,"g":0,"b":0}]';
        imgClan = JSON.parse(json);
    }

    if (typeof(clanImgs[JsonToArray(imgClan)]) ===
"undefined") {
        var imgArr = [];
        var imgloadcouner = 1;

        for (var i = 0; i < imgClan.length; i++) {
            imgArr[i] = new Image();
            var idImg = imgClan[i]["id"] + 1;
            var bg = "";

```

```

        if (i == 0) {
            bg = "background";
        } else if (i == 1) {
            bg = "badge";
        } else {
            bg = "frame";
        }

        imgArr[i].src = "/images/clan_icons/" + bg
+ "/" + idImg + ".png?1.0";
        imgArr[i].onload = function () {
            if (imgloadcouner >= imgClan.length) {
                ShowClanLogo(imgArr,          imgClan,
idHtmlElement, json);
            } else {
                imgloadcouner++;
            }
        };
    }
} else {

    $(".clanLogo")[$(".clanLogo").length - 1].width
= 256;
    $(".clanLogo")[$(".clanLogo").length -
1].height = 256;
    var LoadGerb = new Image();
    LoadGerb.src =
clanImgs[JsonToArray(imgClan)][0];

    LoadGerb.onload = function () {
        $(".clanLogo")[$(".clanLogo").length -
1].getContext('2d').drawImage(LoadGerb, 0, 0);
    }
}

function ShowClanLogo(imgArr, json, idHtmlElement,
startJson) {
    var canv = $(".class:eq(-
1)").context.getElementById(idHtmlElement);
    var imageData = {};
    var ctx = canv.getContext('2d');
    var canArr = [];

```

```

        for (var i = 0; i < imgArr.length; i++) {
            canArr[i] = $(".class:eq(-
1) ").context.createElement('canvas');
            //recolor
            canArr[i].width = imgArr[i].naturalWidth;
            canArr[i].height = imgArr[i].naturalHeight;
            canArr[i].getContext('2d').drawImage(imgArr[i],
0, 0);
            imageData =
canArr[i].getContext('2d').getImageData(0, 0,
canArr[i].width, canArr[i].height);

            for (var j = 0; j < imageData.data.length; j +=
4) {
                if (imageData.data[j + 3] != 0) {
                    var graycolor = (imageData.data[j] +
imageData.data[j + 1] + imageData.data[j + 2]) / 3;
                    if (graycolor > 0) {
                        imageData.data[j] = json[i]["r"] *
(imageData.data[j] / 255);
                        imageData.data[j + 1] = json[i]["g"]
* (imageData.data[j + 1] / 255);
                        imageData.data[j + 2] = json[i]["b"]
* (imageData.data[j + 2] / 255);
                    }
                    imageData.data[j + 3] = 255 *
(imageData.data[j + 3] / 255);
                }
            }

            canArr[i].getContext('2d').putImageData(imageData, 0, 0);
        }

```



---

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



## Розробка мобільної онлайн гри в жанрі MMORPG на базі PHP та JS

Виконав студент 4 курсу  
групи ПД-44  
Хом'як Дмитро Миколайович  
Керівник роботи  
К.т.н Яскевич Владислав Олександрович

Київ – 2021

---

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи:** розробка гри для портфолію, отримання досвіду у розробці мобільних онлайн ігор.
- **Об'єкт дослідження:** розробка мобільних онлайн ігор.
- **Предмет дослідження:** мобільна онлайн гра в жанрі MMORPG, розроблена на основі PHP та JS.

---

## АНАЛОГИ

Warspear Online



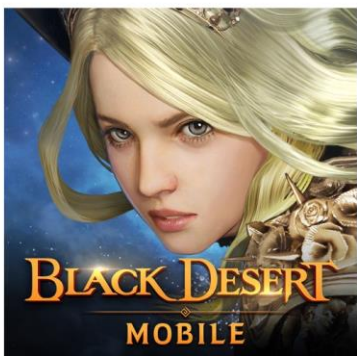
Переваги	Недоліки
Графіка	Потребує вкладень грошей
Геймплей	Часткова відсутність балансу ігрових предметів
Атмосфера	Потрібен постійний онлайн
Free-to-play	Погана технічна підтримка
Велика кількість завдань, предметів і босів	

3

---

## АНАЛОГИ

Black Desert Mobile



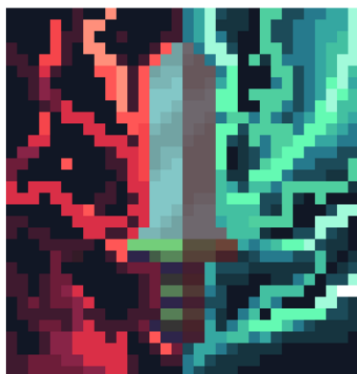
Переваги	Недоліки
Відкритий світ	Оптимізація
Управління персонажем	Відсутність ігрових завдань
Графіка	Потребує вкладень грошей
Геймплей	Потрібно виділяти в день не менше однієї години
	Багато нечесних гравців

4

---

# АНАЛОГИ

MMORPG Laurum Online



Переваги	Недоліки
Цікавий, продуманий сюжет	Відсутність вибору мови інтерфейсу
Цікаві завдання	Оптимізація
Величезна кількість монстрів, локацій та предметів	Багато нечесних гравців
Відкритий світ	Примітивна графіка
Можливість повноцінного обміну між гравцями	Не завжди зрозуміло що потрібно робити

5

---

## ТЕХНІЧНІ ЗАВДАННЯ

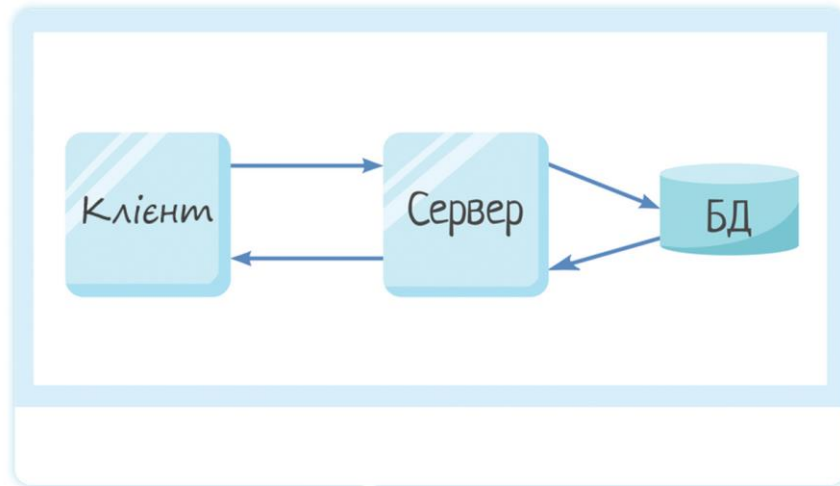
- *Провести аналіз ринку мобільних ігор;*
- *Сформувати перелік технологій, які будуть використовуватись для розробки гри;*
- *Розробити та провести базове тестування гри.*

6



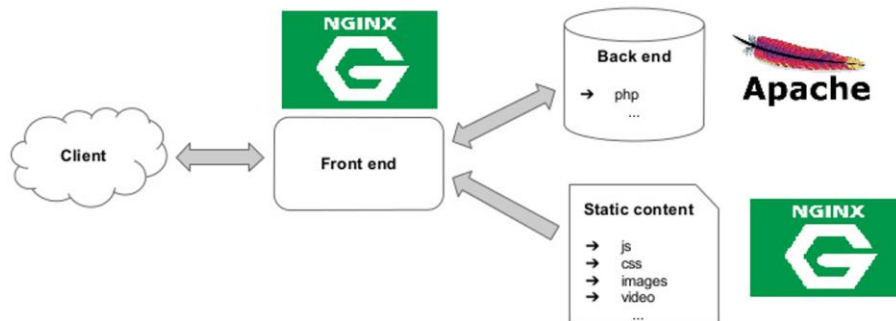


## КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА



9

## ВЗАЄМОДІЯ АРАСНЕ ТА NGINX



10



---

# АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- Хом'як Д.М. // Застосування програмного забезпечення в ІКТ: Матеріали першої всеукраїнської науково-технічної конференції. Збірник тез. 12.02.2021, ДУТ, м. Київ — К.: ДУТ, 2021. — С. 109.
- Хом'як Д.М. // Сучасні інтелектуальні інформаційні технології в науці та освіті /

12

---

## ВИСНОВКИ

1. Проведено аналіз ринку мобільних ігор;
2. Розглянуто технології для розробки гри;
3. Розроблено та протестовано гру;
4. Отримано досвід у розробці мобільних онлайн ігор;
5. Доданий +1 проект для портфолію.

### **Перспективи подальших досліджень:**

- Розширення існуючого та відкриття наступного серверу гри;
- Розробка повноцінного IOS-додатку для гри;
- Новий контент та функціонал гри;
- Вихід на міжнародний ринок.

13

---

**ДЯКУЮ ЗА УВАГУ!**