

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет імені Петра Могили**

**Факультет комп'ютерних наук**

**Кафедра інженерії програмного забезпечення**

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри \_\_\_\_\_ Є. О. Давиденко  
*підпис*

«\_\_» \_\_\_\_\_ 2022 р.

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**Вебсервіси клієнтської підтримки онлайн-гравців**

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21810226

**Студент**

\_\_\_\_\_ К. Є. Трушевський  
*підпис*

«\_\_» \_\_\_\_\_ 2022 р.

**Керівник**

ст. викладач

\_\_\_\_\_ І. О. Кандиба  
*підпис*

«\_\_» \_\_\_\_\_ 2022 р.

**Консультант**

канд. техн. наук, доцент

\_\_\_\_\_ А. О. Алексєєва  
*підпис*

«\_\_» \_\_\_\_\_ 2022 р.

**Миколаїв - 2022**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет імені Петра Могили**

**Факультет комп'ютерних наук**

**Кафедра інженерії програмного забезпечення**

ЗАТВЕРДЖУЮ

Зав. Кафедри

\_\_\_\_\_ Є.О. Давиденко

« \_\_\_ » \_\_\_\_\_ 2022р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи бакалавра**

**Видано студенту групи 409 факультету комп'ютерних наук**

**Трушевському Кірілу Євгеновичу**

*(прізвище, ім'я, по батькові студента)*

1. Тема кваліфікаційної роботи:

Вебсервіси клієнтської підтримки онлайн-гравців

Затверджена наказом по ЧНУ ім. П. Могили від «01» грудня 2021р. № 314

2. Строк представлення кваліфікаційної роботи «\_\_» \_\_\_\_\_ 202\_\_р

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікувальним результатом є створення вебсайту для перегляду кіберспортивних статистичних даних

4. Перелік питань, що підлягають розробці:

- аналіз предметної області та існуючих аналогів;
- розробка програмного забезпечення;
- тестування роботи програмного забезпечення;
- аналіз результатів розробки

5. Перелік графічних матеріалів:

презентація

6. Завдання до спеціальної частини

Дослідження питань охорони праці, які безпосередньо пов'язані з діяльністю розробника програмного забезпечення.

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
Алексєєва А.О.	Кафедра екології	Спеціальна частина з охорони праці

Керівник роботи ст. викладач Кандиба Ігор Олександрович  
(посада, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Завдання прийнято до виконання

Трушевський Кіріл Євгенович  
(прізвище, ім'я, по батькові студента)

\_\_\_\_\_  
(підпис)

Дата видачі завдання «\_\_» \_\_\_\_\_ 2022р.

**КАЛЕНДАРНИЙ ПЛАН**  
**Виконання кваліфікаційної роботи**

Тема: «Вебсервіси клієнтської підтримки онлайн-гравців»

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	20.12.2021	21.12.2021	Виконано
2.	Огляд літератури за темою роботи	26.02.2022	28.02.2022	Виконано
3.	Складання календарного плану КРБ	31.02.2022	02.02.2022	Виконано
4.	Аналіз сучасних засобів реалізації процесу дистанційного навчання	05.02.2022	15.02.2022	Виконано
5.	Розробка вимог до системи дистанційного навчання	16.02.2022	29.03.2022	Виконано
6.	Аналіз ринку наявних технологій і рішень для вирішення завдань	01.03.2022	06.03.2022	Виконано
7.	Проектування структури бази	06.03.2022	12.03.2022	Виконано
8.	Реалізація, тестування та налагодження web-застосунку	15.03.2022	29.05.2022	Виконано
9.	Розробка спеціальної частини з охорони праці	30.05.2022	12.06.2022	Виконано
10.	Відгук керівника КРБ	14.06.2022	14.06.2022	Виконано
11.	Оформлення КРБ та презентації	24.01.2022	19.05.2022	Виконано
12.	Попередній захист	20.05.2022	08.06.2022	Виконано
13.	Рецензування	15.06.2022	15.06.2022	Виконано
14.	Завершення оформлення КРБ та презентації	15.06.2022	22.06.2022	Виконано
15.	Захист кваліфікаційної роботи	30.06.2022	30.06.2022	Виконано

Розробив студент Трушевський Кіріл Євгенович  
(прізвище, ім'я, по батькові студента)(підпис)

Керівник роботи ст.викладач Кандиба Ігор Олександрович  
(посада, прізвище, ім'я, по батькові)(підпис)

«\_\_» \_\_\_\_\_ 2022 р.

## АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

« Вебсервіси клієнтської підтримки онлайн-гравців »

Студент 409 гр.: Трушевський Кіріл Євгенович

Керівник: ст. викладач Кандиба Ігор Олександрович

З різким ростом популярності комп'ютерних ігор та кіберспорту, багато користувачів почало приділяти більше уваги своїм досягненням в грі. Бажаючи стати професійним кіберспортсменом, гравці почали переглядати свою статистику, але багато інформації вони не бачили в самому клієнті гри. Через це з'явилося достатньо сервісів які дозволяють переглянути свою статистику і навіть запропонувати, що можна покращити використовуючи для цього штучний інтелект

Метою кваліфікаційної роботи є вдосконалення процесу підтримки онлайн-гравців шляхом створення вебсервісу з використанням API, на базі PHP та Vue JS.

У вступі визначається актуальність теми, що приймається за мету та невеликий огляд поставленої задачі, предмет дослідження та об'єкт дослідження.

У першому розділі описується аналітична частина, тобто огляд існуючих аналогів. Визначається основна особливість вебсервісів, завдяки чому було сформовано загальне розуміння предметної області.

У другому розділі описується процес розробки та вибір мови програмування, середовище розробки, логіка веб-сайту. Розробка UML-діаграм та опис інтерфейсів.

У третьому розділі демонструється проведена робота з кодування та тестування, крім того описується розробка налаштувань для зручного використання.

Об'єкт кваліфікаційної роботи: статистичні дані кіберспортивних дисциплін та гравців.

Предмет кваліфікаційної роботи: вебсервіси аналізу та обробки кіберспортивних даних.

У висновках проводиться аналіз роботи та отриманих результатів.

Кваліфікаційна робота бакалавра викладена на 65 сторінок, вона містить 4 розділи, 35 ілюстрацій, 3 таблиць, 20 джерел в переліку посилань.

## **ABSTRACT**

of the Bachelor's Thesis

« Web services for customer support of online players »

Student 409 gr.: Trushevskiy Kiril

Supervisor: Senior Lecturer Kandyba Ihor

With the sharp rise in popularity of computer games and eSports, many users have begun to pay more attention to their achievements in the game. Wanting to become a professional e-athlete, players began to view their statistics, but they did not see much information in the client of the game. Because of this, there are enough services that allow you to view your statistics and even suggest that you can improve using artificial intelligence

The purpose of the qualification work is to improve the process of supporting online players by creating a web service using the API, based on PHP and Vue JS.

The introduction determines the relevance of the topic to be taken as a goal and a brief overview of the task, the subject of research and the object of research.

The first section describes the analytical part, ie an overview of existing analogues. The main feature of web services is determined, thanks to which the general understanding of the subject area was formed.

The second section describes the development process and the choice of programming language, development environment, website logic.

The third section demonstrates the work done on coding and testing, in addition, describes the development of settings for easy use.

Object of qualification work: statistics of e-sports disciplines and players.

Subject of qualification work: web services of analysis and processing of e-sports data.

The conclusions analyze the work and the results obtained.

The qualifying work of the bachelor is presented on 65 pages, it contains 4 sections, 35 illustrations, 3 tables, 20 sources in the list of references.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	4
ВСТУП .....	5
1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ ФОРМУВАННЯ КІБЕСПОРТСМЕНІВ В КІБЕРСПОРТІ .....	8
1.1 Дослідження ролі та місця кіберспорту серед напрямків сучасного спорту	8
1.2 Аналіз взаємозв'язку команд, позицій та гравців у кіберспорті.....	11
1.3 Аналіз сучасних напрямків кіберспорту .....	12
1.4 Проблеми розвитку гравців в кіберспорті.....	16
1.5 Аналіз аналогічних рішень .....	17
1.6 Специфікації вимог до програмного забезпечення .....	20
Висновки до розділу 1 .....	21
2 МОДЕЛЮВАННЯ ТА РОЗРОБКА КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....	23
2.1 Інформаційна модель обробки кіберспортивних статистичних даних....	23
2.2 Функціональна модель обробки кіберспортивних статистичних даних та за стосунку .....	26
2.3 Сценарії використання системи.....	30
Висновки до розділу 2 .....	32
3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕПЕЧЕННЯ .....	33
3.1 Аналіз функціоналу та формування стеку технологій проекту .....	33
3.2 Проектування бази даних.....	38
3.3 Моделювання структури .....	42
Висновки до розділу 3 .....	45
4 РОЗРОБКА ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ.....	46
4.1 Розробка функціоналу отримання та обробки даних.....	46
4.2 Тестування інформаційної технології.....	59



Висновки до розділу 4 .....	62
ВИСНОВКИ.....	64
СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ.....	65

## **ПЕРЕЛІК СКОРОЧЕНЬ**

ДР	–	Дипломна робота
ІС	–	Інформаційна система
СКБД	–	Система керування базами даних
БД	–	База даних

## ВСТУП

Робота присвячена розробці вебсервісу для перегляду статистики користувача системи Steam. Для авторизації використовується акаунт користувача та його дані відображуються на вебсайті шляхом взаємодії з сервісом Steam, який дозволяє отримати всю інформацію про користувача після авторизації. З певними обмеженнями, розроблене програмне забезпечення може бути використана для взаємодії з штучним інтелектом.

*Актуальність теми.* Індустрія комп'ютерних ігор виникла в середині 1970-х років як рух ентузіастів і за кілька десятиліть виросла з невеликого ринку у величезну індустрію з щорічним прибутком в 9.5 мільярдів доларів у США в 2017 році. На ринку працюють як великі гравці, так і невеликі фірми, а також незалежні розробники та спільноти (напр. Indiegogo, Kickstarter та ін.). З розвитком комп'ютерів у розробників ігор з'явилося більше можливостей та свободи і творчості. Ігри пройшли довгий шлях становлення від найпростіших двоколірних 2D до сучасних 3D ігор з таким рівнем графіки, що людина, яка грає в цю гру, забуває, що вона в грі і сприймає її віртуальний світ як реальний. Також в даний час йде розробка ігор для шоломів віртуальної реальності і вже зовсім скоро на ринку з'являться ігри, які дозволять гравцеві повністю зануритися у віртуальний світ.

На сьогодні, комп'ютери стали доступними через це багато людей мають можливість насолоджуватись іграми та деякі мають заробляють на цьому завдяки кіберспорту.

Кіберспорт поєднує в собі характеристики традиційного спорту (змагальна складова), сфери розваг (емоційна складова) та ІТ-сфери (технологічна складова). Таке поєднання охоплює як унікальні, так і споріднені риси кожної сфери. Аналогічно з традиційним спортом, кіберсфера передбачає існування команд, гравців, правил турніру, проведення змагань і прагнення до перемоги. Це забезпечує високий рівень залученості вболівальників до процесу.

Як і індустрія розваг, кіберспорт робить ставку на емоційній складовій, що дозволяє приваблювати велику кількість вболівальників і становить інтерес для медіасфери. З точки зору спорідненості з ІТ-сферою в кіберспорті ключові події відбуваються у віртуальному середовищі, що обумовлює потребу у спеціальних технологіях і програмному забезпеченні. Стрімкий розвиток кіберспорту значно вплинув і на власне ігрову індустрію. Розробники випускають ігри, в яких гравці об'єднуються в команди і змагаються між собою.

Виникли турніри, професійні команди та фанати. Найбільш поширеними кіберспортивними дисциплінами є Dota 2, Counter-Strike : Global Offensive, League of Legends, Hearthstone, Call-of-Duty : Warzone. Ці ігри були розроблені компаніями VALVe (Dota 2, Counter-Strike : Global Offensive), Blizzard Activision (Hearthstone, Call-of-Duty : Warzone) та Riot Games (League of Legends).

Steam – це сервіс Valve, відомого розробника відеоігор, який надає послуги цифрового розповсюдження, мультиплеєру та комунікації гравців. Steam також використовується для отримання автоматичних оновлень і новин як для продуктів Valve, так і для сторонніх розробників, зокрема Activision, Codemasters, Eidos Interactive, Epic Games, GSC Game World, id Software, Sega, THQ, Atari, Rockstar Games, Ubisoft, NCSoft та багато інших. Окрім відеоігор, сервіс пропонує інструменти для розробників ігор, музику, фільми та ігрове обладнання. За допомогою API Steam дозволяє взаємодіяти зі своїми серверами та отримувати деталі стосовно кожного аккаунту.

API – це система інструментів і ресурсів вбудована в застосунок, яка дозволяє розробникам створювати програмні продукти, які взаємодіють з іншими службами. API – це те, що дозволяє двом окремим додаткам взаємодіяти один з одним. API дозволяють користувачам взаємодіяти з іншими службами або сайтами, не залишаючи ваш сайт. API важливі, тому що без них було б дуже складно зрозуміти синтаксис всіх додатків і сервісів, з якими

взаємодіє ваш інструмент. API-інтерфейси пояснюють, як називається кожен об'єкт і які дії можна виконати з кожним з них.

**Об'єкт кваліфікаційної роботи** – статистичні дані кіберспортивних дисциплін та гравців.

**Предмет кваліфікаційної роботи** – вебсервіси аналізу та обробки кіберспортивних даних.

**Мета кваліфікаційної роботи** – вдосконалення процесу підтримки онлайн-гравців шляхом створення вебсервісу з використанням API, на базі PHP та Vue JS.

**Досягнення мети можливо шляхом вирішення поставлених задач:**

1. Дослідити існуючі сервіси кіберспортивної статистики.
2. Розробити специфікацію вимог для проекту.
3. Проаналізувати сучасні технології з розробки веб застосунків.
4. Дослідити методи зберігання та обробки великої кількості даних.
5. Визначити можливості взаємодії з сервісом Steam.

# **1 АНАЛІЗ СУЧАСНОГО СТАНУ ПРОБЛЕМИ ФОРМУВАННЯ КІБЕСПОРТСМЕНІВ В КІБЕРСПОРТІ**

## **1.1 Дослідження ролі та місця кіберспорту серед напрямків сучасного спорту**

Інформаційні технології, поруч із використанням в традиційних процесах та галузях, забезпечують формування нових секторів економіки та суспільного життя. До таких секторів належить і кіберспорт – ігрові змагання, засновані на використанні інформаційних і комп'ютерних технологій, які відбуваються у віртуальному просторі. Наразі кіберспорт є глобальною індустрією, розмір якої перевищує оборот в 150 млрд дол. США, а загальна кількість учасників і прихильників – 2,5 млрд. осіб. У 2015 році обсяг ринку кіберспорту становила \$655 млн., у 2018 році ж досягла \$865 млн.

На сьогодні ринок кіберспорту є одним із найдинамічніших. Кількість кібергравців та глядачів постійно зростає, як і їх купівельна спроможність. Відтак, вкладання ресурсів у цей ринок є доцільним. Спонсорські кошти дозволяють прискорити розвиток галузі і збільшують об'єм ринку. Тому можна з упевненістю стверджувати, що спонсори – це не виключно альтруїсти і їхні вкладення, є нічим іншим, як інвестиціями, що забезпечують стрімкий ріст ринку.

В теперішній час все більшої популярності набирає спорт, але за умов карантину популярним все більше стає і кіберспорт. В загальному, спорт є організованою за певними правилами діяльністю людей, що полягає в зіставленні їхніх фізичних та інтелектуальних здібностей. Це змагання за певними правилами, зазвичай це специфічний вид фізичної або інтелектуальної активності, яку здійснюють з метою змагання, а також поліпшення здоров'я, отримання морального і матеріального задоволення, прагнення до удосконалення і слави, покращення навичок.

У загальному спорт поділяють на види, зокрема:

- олімпійські (легка та важка атлетика, плавання, гімнастика, різні види єдиноборств та ін.);
- неолімпійські (бальні танці, крикет, регбі, американський футбол, гольф та ін.);
- технічні (мото- та автоперегони, класичне ралі та ін.); – єдиноборства;
- ігрові (теніс, бадмінтон, боулінг, різноманітні різновиди більярду тощо); – розумові (настільні ігри);
- кіберспорт або е-спорт.

Через карантин та стрімкий розвиток ІТ, спорт переходить в віртуальну площину, тобто все швидше розвивається кіберспорт. Відтак, кіберспорт – це вид інтерактивної спортивної діяльності, результатом якої є спортивні змагання в комп'ютерні ігри, в яких гравці розвивають свої фізичні та інтелектуальні навички.

Ще донедавна кіберспорт не вважали офіційним видом спорту. Проте, беручи до уваги масштаб який охоплює е-спорт, 7 вересня 2020 року його було офіційно визнано видом спорту в Україні. Віртуальний, електронний спорт можна умовно поділити на такі види:

- шутери («Counter-Strike», «Warface»);
- стратегії («StarCraft», «WarCraft»);
- командні рольові ігри, в яких гравці керують своїм персонажем, постійно його розвиваючи («League of Legends», «Dota 2»);
- спортивні симулятори футболу, хокею, баскетболу (FIFA, NHL);
- технічні симулятори, в яких люди грають за водія танка, літака або автомобіля («World of Tanks»).

Комп'ютерні ігри – це звичайне програмне забезпечення, мета якого забезпечити цікаве дозвілля з використанням комп'ютера. Основними компонентами ігрового 13 виду діяльності є: наявність правил; відсутність

практичної мети; специфічна мотивація, суть якого складається в так названому функціональному задоволенні, що забезпечується як самим процесом гри, так і її результатами; нематеріальні результати діяльності.

Класифікацій комп'ютерних ігор є чимало, розглянемо одну з найбільш поширених:

- ігри типу «action», у тому числі і «RPG»;
- пригодницькі, типу «quest»;
- стратегічні; – ігри, що імітують транспорт;
- віртуальне казино.

У змаганнях з кіберспорту використовують ті комп'ютерні ігри, які є найбільш видовищними:

- шутери – жанр відеоігор, основу ігрового процесу якого складає стрільба зі зброї по цілях, зазвичай вороже налаштованих;
- стратегії реального часу, у яких гравець і противник виконують дії одночасно, а не по черзі;
- спортивні симулятори – жанр відеоігор, основою ігрового процесу якого є правила реального або вигаданого виду спорту.

Конкретні ігри називаються дисциплінами, серед яких FIFA, CounterStrike, Dot 2, Starcraft, World of Tanks, League of Legends та інші. Таким чином, набираючи все більшої популярності, кіберспорт отримує більшу кількість гравців та нові види ігор. При цьому, на сучасному етапі, особливо за посилення карантинних заходів, розмиваються межі між традиційним спортом та віртуальним, кіберспортом. Наприклад, збірна України у першому матчі суперфіналу Світової Ліги з шахів перемогла команду Сербії, при цьому внаслідок карантинних обмежень поєдинок турніру відбувався дистанційно, на платформі Chess.com. Тому можна стверджувати не тільки про збільшення ролі кіберспорту, а й про поступове поглинання ним деяких видів традиційного спорту або їх складових.



## 1.2 Аналіз взаємозв'язку команд, позицій та гравців у кіберспорті

У кіберспорті команди гравців часто називаються кланами. Деякі види відеоігор, наприклад Counter-Strike, створені саме для командного змагання, інші ж дозволяють грати як в режимі один на один, так і команда проти команди.

Десять років тому в Україні було більше десятка напівпрофесійних команд, зараз таких з постійним складом немає. Зате професійних стало більше - в 2010 до таких можна було віднести тільки Natus Vincere. Сьогодні в Україні базуються чотири професійні команди: Na'Vi, HellRaisers, Gambit і нещодавно з'явилася pro100.

Підготовка до змагань потребує регулярних тренувань, зазвичай, мінімум дві-три години на день, а перед самими змаганнями – шість-вісім. Напередодні змагань кіберспортсмени, як і атлети, тренуються двічі на день. Кіберспортсменами можуть бути не всі. В основному це молоді люди, в середньому трохи за 20. Вони проводять багато часу, тренуючись в улюбленій кібердисципліні. Це означає, що їм не обійтися без якісного комп'ютера і потужного інтернет-зв'язку.

Учасники змагань повинні бути фізично витривалими, оскільки пульс гравців часом досягає 160-180 ударів за хвилину. Кіберспортсмени іноді травмуються. Страждають в першу чергу руки, через одноманітність рухів, які людина виконує сидячи перед комп'ютером, та зір, що падає через тривале перебування перед монітором.

Окрім професійного кіберспорстмена, є багато вакансій для організаторів подій, агентів, економістів та статистів, девелоперів, менеджерів з продажу, журналістів, PR-менеджерів тощо.

Однією з популярних ігор зараз є League of Legends. В даній грі кожен гравець має свою позицію. Позиція – це атрибут чемпіона (єдиний керований

гравцем персонаж), який позначає найбільш популярну або переважну лінію для нього

Позиція не є жорстко закріпленою за характеристикою чемпіона і протягом існування чемпіона може змінюватися. За гравцем залишається вільний вибір позиції для обраного чемпіона.

Кіберспортсмени, як і традиційні спортсмени, мають тренерів, які допомагають швидше досягти високих результатів. Ігри в кіберспорті змінюються швидко. Тренери в них грають стільки ж часу, скільки і гравці. Тому, фактично вони не можуть навчити спортсменів чогось нового, але вони стежать за командами конкурентів, за трендами, переглядають записи ігор, можуть помічати звички, заготовки суперника. Тренери спостерігають за процесом гри та можуть визначити, хто з учасників команди в певній ситуації мав рацію, а хто зробив помилку, і як її уникнути на майбутніх змаганнях.

Таким чином, визначено основними сутностями у кіберспорті команду, позицію для гравця та власне гравця – людину й персонажа як узагальненої сутності; ці елементи перебувають в складному взаємозв'язку й грають основну роль за автоматизованого формування команд в кіберспорті.

### **1.3 Аналіз сучасних напрямків кіберспорту**

Еволюція комп'ютерних ігор відбулась через їх технологічні вдосконалення, а завдяки появі Інтернет вони стали багатокористувацькими. Вперше кіберспортивні риси та змагання по мережі з'явилися в Quake: Arena, Starcraft. Ігри захоплювали не графікою, а відмінним динамічним геймплеєм, вимагали від гравців швидкої реакції і багатьох годин тренувань. Online кіберзмаганнями не мали обмежень, люди почали збиратися на LAN-турніри в Інтернет-клубах. Активно кіберспорт почав розвиватись на протязі останніх 5–10 років із-за доступності Інтернет.

Кіберспорт поєднує в собі характеристики традиційного спорту (змагальна складова), сфери розваг (емоційна складова) та ІТ-сфери (технологічна складова). Таке поєднання охоплює як унікальні, так і споріднені риси кожної сфери. Аналогічно із традиційним спортом, кіберсфера передбачає проведення кіберзмагань, існування кіберкоманд, кібергравців, правил турніру і прагнення до перемоги. Це забезпечує високий рівень залученості вболівальників до процесу. Так само, як індустрія розваг, кіберспорт робить ставку на емоційній складовій, що дозволяє приваблювати велику кількість прихильників і становить інтерес для засобів медіа. З точки зору спорідненості з ІТ-сферою, в кіберспорті ключові події відбуваються у віртуальному середовищі, що обумовлює потребу у спеціальних технологіях і програмному забезпеченні.

Стрімкий розвиток кіберспорту значно вплинув на ігрову індустрію. Розробники випускають ігри, в яких гравці об'єднуються в команди і змагаються між собою. Виникли турніри, професійні команди, і фанати. Виникла потреба формувати ефективні команди цільового призначення, для чого потрібно було враховувати велику кількість факторів та їх взаємозв'язків. Оцінка ж вже сформованих команд із врахуванням тих же факторів та їх взаємозв'язків є основою для прогнозування результатів змагань у кіберспорті.

Розробники щодня вдосконалюють свої ігри, додають нові інструменти, нових героїв та можливостей. Зараз на піку популярності знаходяться такі ігри: Counter-Strike, DotA, FIFA, Halo 2, Heroes of Newerth, League of Legends, Quake, Starcraft, Warcraft, World of Tanks, Hearthstone. На даний час є чимало напрямків кіберспорту, розглянемо кілька з них.

Одним з напрямків є спортивні симулятори футболу, хокею, баскетболу. Як і раніше, коли тільки розпочинався розвиток комп'ютерних ігор, так і зараз, даний напрямок є популярним.

Прикладом даного напрямку може слугувати гра «FIFA». FIFA – серія відеоігор жанру футбольних симуляторів, що розробляється студією EA Canada, яка входить в корпорацію Electronic Arts(рис 1.1).



Рисунок 1.1 – FIFA 22

FIFA – футбольний симулятор, дисципліна, що завжди має свого глядача. Саме завдяки цьому дана гра і не втрачає свою популярність та актуальності. Для досягнення більшого реалізму розробники «FIFA 2015» звернулись до «живих» ігрових моментів, пов’язаних з емоціями і контактами віртуальних персонажів, тобто до повсякденності футбольних матчів. Емоції гравців поширюються як на суперників, так і на членів однієї команди. Гравці реагують на старання та помилки партнерів, а також на грубу гру суперників. Крім того, гра є більш реалістичною завдяки тому, що тепер у ній реалізовані різні живі контакти, як-от штовхання руками та ліктями, хапання за майку тощо. Тобто стратегія реалістичності «FIFA 2015» є стратегією відображення «моментів поза грою», які не пов’язані з ігровою логікою і правилами, а є наслідком виключно людського чинника.

Наступним напрямком можна розглянути – технічні симулятори. Даний напрямок не втрачає своєї популярності вже багато років. Прикладом є гра «World of Tanks» – клієнтська масова багатокористувальницька онлайн-гра в реальному часі в жанрі аркадного танкового симулятора в історичному сеттінгу Другої світової війни, розроблена білоруською студією Wargaming.net(рис 1.2).



Рисунок 1.2 – World of Tanks

Дана гра пройшла багато модифікацій та оновлень, і розробники її досі вдосконалюють, цим самим привертають увагу користувачів до гри.

Концепція World of Tanks базується на командних танкових боях в режимі PvP. Онлайн-реліз російської версії гри відбувся 12 серпня 2010 року. Онлайн-реліз World of Tanks на територіях Європи та Північної Америки відбувся 12 квітня 2011 року. Гра базується на схемі Free-2-Play; самі розробники декларують тільки безкоштовне завантаження клієнта і безкоштовний вхід в гру. 2016 року відбувся чемпіонат світу з World of Tanks, в якому виграла українська команда Natus Vincere. Приз за перше місце склав 150 000 дол. США.



Ще одним сучасним напрямком є командні рольові ігри, в яких гравці керують своїм персонажем. Яскравим прикладом даного напрямку є гра «League of Legends» – відеогра жанру МОБА (багатокористувацька онлайн бойова арена), створена командою розробників Riot Games для Microsoft Windows та Macintosh(рис 1.3).



Рисунок 1.3 – League of Legends

У жовтні 2012 року Riot Games у своєму прес-релізі оголосила, що League of Legends стала найпопулярнішою відеоурою в світі: у ній було зареєстровано 70 мільйонів гравців, з яких 32 мільйони були активними впродовж місяця, а 12 мільйонів – щоденно. Середній постійний онлайн гри становив 3 мільйони гравців. В вересні 2016 року Riot Games заявили про понад 100 мільйонів активних гравців щомісяця.

#### **1.4 Проблеми розвитку гравців в кіберспорті**

Через стрімке зростання популярності та доступності кіберспорту, багато користувачів намагаються потрапити до професійної команди. В залежності від

дисципліни, потрібно розвивати різні навички. Основною проблемою розвитку гравців є те, що їм нікуди звернутись за порадою. Наразі не існує академії, яка б змогла зробити зі звичайного гравця – професіонала, окрім академій при кіберспортивних командах, але до них вкрай складно потрапити. Звичайно ж талановитим гравцям легше потрапити до команди, але це не дає гарантій що його візьмуть до команди.

### **Постановка задачі**

Ціль кваліфікаційної роботи полягає у розробці інформаційної системи для забезпечення гравців отриманням статистики відносно гри в яку вони грають, задля дослідження практичної ефективності обраної методики розвитку. Дослідження практичної ефективності обраної методики розвитку слід зробити шляхом аналізу попередніх результатів, які були занесені до системи.

Для досягнення цієї мети, необхідно виконати наступні задачі:

1. Провести аналіз відомих систем, які надають послуги щодо отримання статистики користувача.
2. Розробити метод оцінки ефективності розвитку професійних навичок користувача в обраній дисципліні.
3. Розробити веб-сайт, який буде отримувати дані з серверу Steam та відображати отриману інформацію.

Розробити базу даних, яка будет отримувати та зберігати інформацію про користувача.

### **1.5 Аналіз аналогічних рішень**

В результаті дослідження було проаналізовано аналогічні рішення. Сервіс GOSU-AI (рис 1.4) дозволяє користувачам отримувати статистичні відносно кожного матчу. Також є елементи аналізу штучного інтелекту. Цей сервіс може аналізувати тільки деякі дисципліни, а саме Dota 2, League of Legends, PUBG. Великою перевагою є те, що достатньо ввести посилання на аккаунт

користувача та необхідна авторизація. Недоліком є те, що сервіс доступний тільки англійською мовою та мала кількість дисциплін.



Рисунок 1.4 - GOSU-AI

Сервіс Game-Stat (рис 1.5) дає можливість переглянути скільки людей зараз грають в гру , отримати свій поточний ранк для обмеженої кількості дисциплін. Перевагою є дуже простий інтерфейс та відсутність необхідності авторизації. Недоліки цього сервісу полягають в тому що неможливо переглянути детальну статистику.





## Статистика онлайн ігор на сайті Game-Stat.com

### Онлайн в іграх - статистика Steam

На нашому сайті Ви можете знати ТОП ігор по онлайн на даний момент. В таблиці представлені самі популярні онлайн-ігри і кількість одночасно ігрованих гравців.

#### Рейтинг по кількості гравців на сервері

Гра	Гравців онлайн	Максимум онлайн
Counter-Strike: Global Offensive	580888	923996
Dota 2	400268	695841
Lost Ark	248681	885379
PUBG: BATTLEGROUNDS	208583	432920
Apex Legends	206399	360428
ARK: Survival Evolved	111998	247292
Grand Theft Auto V	82358	133299
Team Fortress 2	82269	111575
Wallpaper Engine	77176	105360
Rust	55937	165120

Кожному гравцю хочеться знати свій рейтинг і ігрову статистику. Наш сайт надає можливість знати свою статистку, досягнення і рейтинги в іграх. На даний момент ми надаємо [статистику по Fortnite](#) - набираючій популярності онлайн-іграх. В майбутньому плануємо підключити можливість перевірити статистику і по інших іграх.

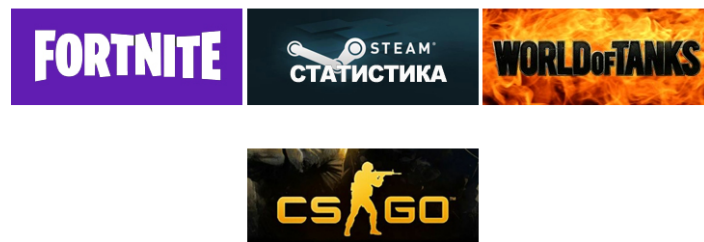


Рисунок 1.5 – GameStat

Сервіс Cod-Tracker (рис 1.6) дає можливість отримати статистику великої кількості дисциплін, можливість авторизації за допомогою сервісів Steam, Battle.net, Xbox Live. Користувач може отримати досить детальну статистику відносно обраної дисципліни, також у цього сервісу існує мобільний застосунок. Недоліком є достатньо складний користувацький інтерфейс та те, що сервіс доступний тільки англійською та постійна реклама у діалогових вікнах.

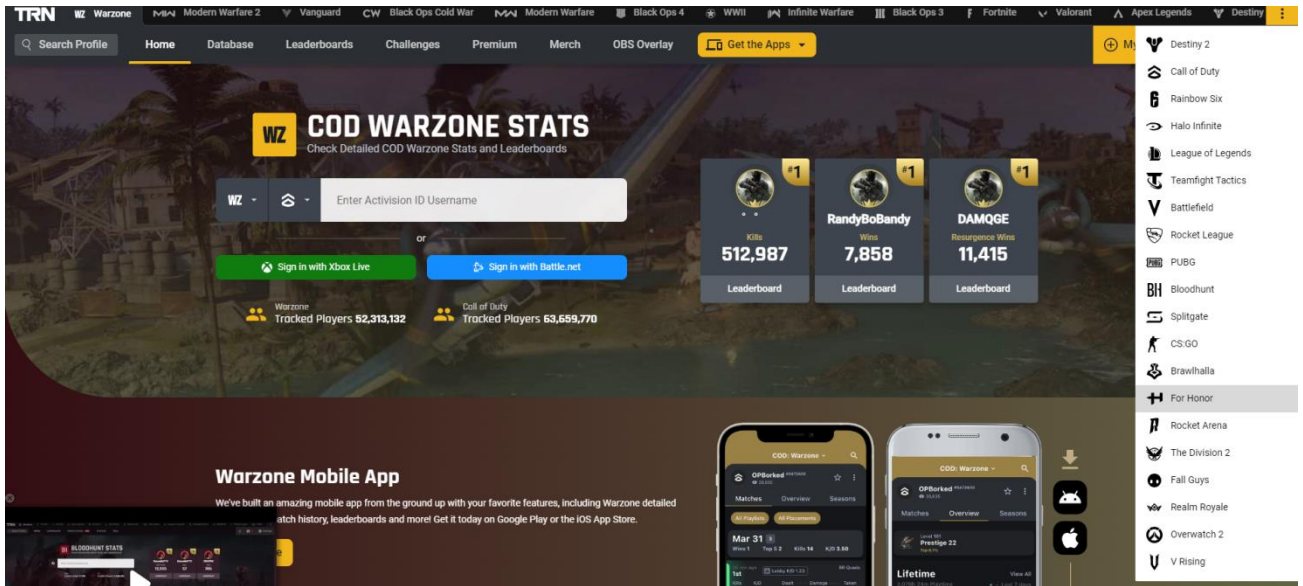


Рисунок 1.6 - Cod-Tracker

## 1.6 Специфікації вимог до програмного забезпечення

### ПРИЗНАЧЕННЯ ТА МЕЖІ ПРОЄКТУ

### ЗАГАЛЬНИЙ ОПИС

Даний застосунок буде використовуватись гравцями та користувачами сервісу Steam. Кожен користувач має мати аккаунт Steam. Користувачі матимуть обмежений функціонал аналізу штучного інтелекту

### ФУНКЦІЇ СИСТЕМИ

#### 1. Авторизація

1.1.1 Ця функція надає можливість користувачу авторизуватись використовуючи свій аккаунт Steam

1.1.2 Вхідна інформація – аккаунт Steam, вихідна інформація – масив даних

1.1.3 Функціональні вимоги – наявність активного API ключа

#### 2. Отримання статистики

1.2.1 Ця функція надає можливість користувачу продивитись статистику по обраній грі

1.2.2 Вхідна інформація – аккаунт Steam та ID обраннях гри, вихідна інформація - масив відформатованих даних

1.2.3 Функціональні вимоги – наявність активного API ключа

### **ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ:**

- Вхідні дані – Steam аккаунт користувача.
- Всі дані будуть зберігатись у базі даних MySQL на боці сайту.

### **ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ**

Наявність комп'ютеру та інтернету

### **ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

- Для забезпечення доступу до сайту необхіден веб-браузер.
- Мережне програмне забезпечення – наявність інтернету.
- Для ведення інформаційної бази використовується веб-сервер.
- Мова розробки – PHP, HTML, CSS, JS з використанням фреймворку

Vue JS.

### **ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ**

1. Веб-сайт матиме видиму частину для користувача.
2. Всі редагування коду виконані у середовищі Visual Studio Code.
3. Для комунікації з сервером, необхідна бібліотека OpenID.

### **ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

- Кросс-браузерна доступність та адаптивність.
- Веб-сайт повинен бути захищеним.

Для забезпечення надійності, потрібно обрати правильний веб-хостинг.

### **Висновки до розділу 1**

В розділі за результатом дослідження ролі та місця кіберспорту серед напрямків сучасного спорту визначено, що на сучасному етапі, особливо за посилення карантинних заходів, розмиваються межі між традиційним спортом та віртуальним, кіберспортом. Можна стверджувати не тільки про збільшення

ролі кіберспорту, а й про поступове поглинання ним деяких видів традиційного спорту або їх складових. Аналогічно з традиційним спортом, кіберсфера передбачає існування команд, гравців, правил турніру, проведення змагань і прагнення до перемоги. За результатом аналізу сучасного стану проблеми автоматизації формування команд в кіберспорті визначено, що основними сутностями у кіберспорті команду, позицію для гравця та власне гравця – людину й персонажа як узагальненої сутності; ці елементи перебувають в складному взаємозв'язку й грають основну роль за автоматизованого формування команд в кіберспорті.

Встановлено, що для формування професійного гравця у кіберспорті слід враховувати широкий ряд параметрів. Ці параметри стосуються як людей-гравців, так і їхніх персонажів у кіберспорті, причому множина цих параметрів у різних випадках може дуже відрізнятися. Також слід враховувати статистику всіх проведених ігор та досвід гри кібергравця на певній позиції команди.

Тому метою роботи поставлено розробку інформаційної системи для відображення статистики користувача. Ця система допоможе гравцю зрозуміти, що необхідно покращити задля поставленої мети. Для дослідження ефективності системи, необхідно розробити алгоритм, який дозволяє відслідкувати прогрес.

## 2 МОДЕЛЮВАННЯ ТА РОЗРОБКА КОМПОНЕНТІВ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

### 2.1 Інформаційна модель обробки кіберспортивних статистичних даних

Основними сутностями які впливають на роботу програмного забезпечення, що розроблюється, є гравець та його аккаунт який прив'язаний до Steam. В залежності від того, які ігри є у користувача на аккаунті вихідні дані можуть змінюватись. Наприклад, користувач грає в Dota2, йому буде відображено статистику відносно саме цієї гри та з її унікальними значеннями відносно ігрового процесу, так само і з іншими іграми. Враховуючи це, ми будемо мати різну кількість даних для обробки стосовно кожної гри.

При цьому основні параметри самого користувача будуть статичними та їх необхідно обробляти лише один раз при авторизації нового користувача та додавання його перших даних до БД. З урахуванням додаткових параметрів які впливають на процес відображення та аналізу даних, необхідно уніфікувати їх. Для цього необхідно виявити спільні параметри для всіх ігор, які прив'язані до користувача.

Розглянемо параметри, які необхідно обробляти при першій авторизації користувача. Самим важливим параметром є *steamid* – він дає ID користувача в сервісі Steam, після цього він записується до бази та являється ідентифікатором користувача, *personaname* – цей параметр визначає ім'я користувача, *avatar* – аватар користувача який встановлено на даний момент. *Personastate* – цей параметр відповідає за статус аккаунту, а саме чи онлайн користувач, існує декілька статусів аккаунту, а саме: 0 або Offline – це означає що користувач зараз офлайн, 1 або Online – користувач онлайн, 2 або Busy – користувач онлайн, але зайнятий та у нього не відображаються повідомлення, 3 або Away –

користувач онлайн але не проявляє активність мінімум 5 хвилин, 4 або Snooze – користувач онлайн, але не проявляє активність протягом однієї години, 5 або looking to trade – користувач онлайн і проводить обмін внутріігрових речей з іншим користувачем, 6 або looking to play - користувач онлайн і запустив гру і ця гра відобразилась в його профілі.

Не менш важливим параметром на рівні з *SteamID* є *communityvisibilitystate*-цей параметр відповідає за можливість зчитування даних з аккаунту. Наприклад, якщо цей статус має значення Private, то це означає що акант доступний для перегляду тільки для самого власника акаунту, якщо має значення Friends only, то сам власник аккаунту може переглядати його, та інші користувачі які є у списку його друзів, та якщо має значення Public, то тоді всі користувачі можуть переглядати деталі стосовно його аккаунту.

За допомогою Steam можна також отримати персональні дані користувача, якщо в нього відкритий профіль, а саме:

- *realname* – справжнє ім'я користувача;
- параметр *timecreated* – дата коли був створений його аккаунт;
- параметр *cityid* – ідентифікатор його міста, параметр;
- *loccountrycode*-ідентифікатор країни користувача;
- параметр *loccityid* – ідентифікатор області міста.

Необхідно сформулювати параметри, які спільні для всіх ігг Steam. Ці параметри мають бути загальними, а саме: параметр *game\_count* – відповідає за загальну кількість ігор на аккаунті користувача, включаючи безкоштовні та платні ігри, які можна придбати у сервісі Steam; параметр *appid* – цей параметр відповідає за ідентифікацію ігри а також для отримання більш детальної статистики відносно обранної гри; параметр *playtime\_2weeks* -кількість годин яку зіграв користувач у гру протягом двох тижнів; параметр *playtime\_forever* – відповідає за отримання загальної кількості годин, які зіграв користувач; параметр *img\_logo\_url* – необхіден для отримання логотипу обранної гри; ще 2022р.

один важливий параметр це *has\_community\_visible\_stats*-він необхіден для отримання інформації стосовно досягнень користувача у обраній грі.

Ці все параметри є загальними для всіх аккаунтів Steam, та ігр які існують в цьому сервісі, якщо необхідно отримати більш специфічні деталі стосовно кожної гри, то необхідно аналізувати та робити запити до окремих API, які відображають масив даних притаманні для обраного проекту. Найбільш популярні проекти це Dota 2, Team Fortress 2, Counter-Strike : Global Offensive, Rust, Tom Clancy's Rainbow Six Siege, Player Unknown Battlegrounds.

Використовуючи отриманні дані, користувач інформаційної системи має можливість аналізувати їх та розуміти що відбувається з його навичками: чи покращуються вони чи ні. Це можна відобразити на рисунку 2.1.

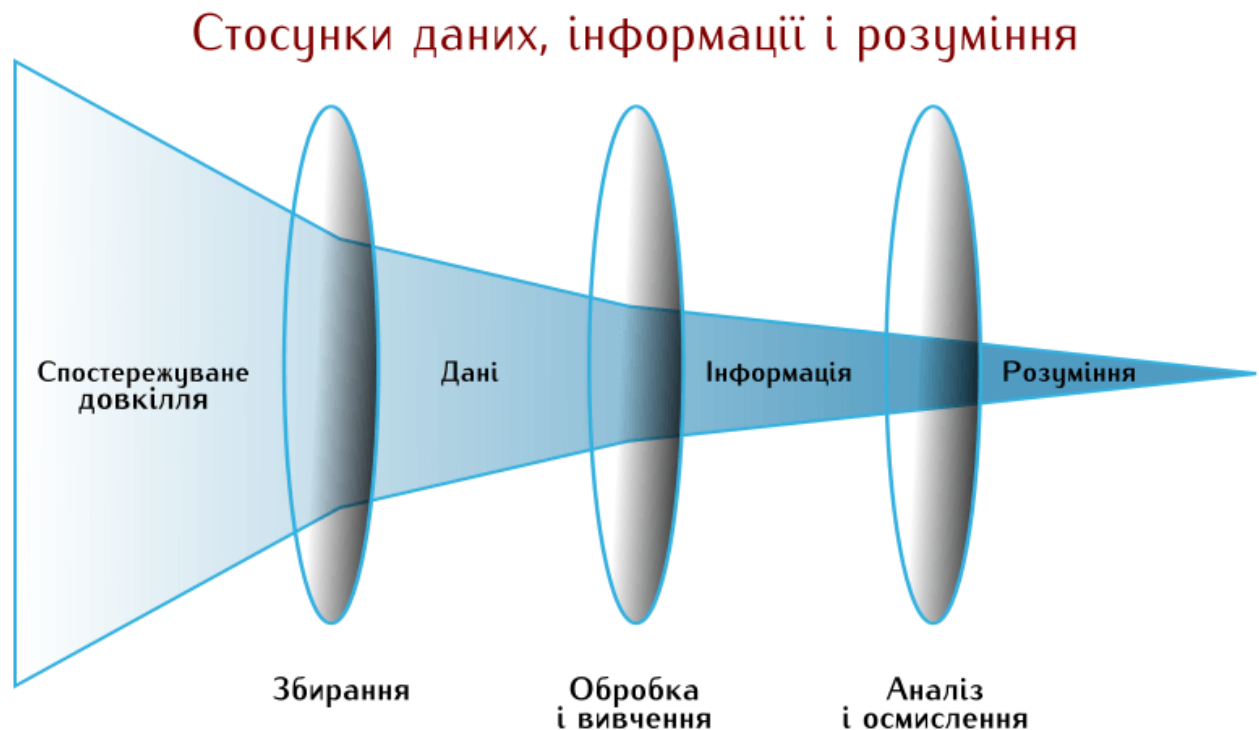


Рисунок 2.1 – Зображення моделі аналізу даних

На цьому рисунку видно, що користувач має змогу отримати дані з спостережувального довкілля, а саме з сервісу Steam. Вебзастосунок, який було

розроблено збирає всі дані, відображає користувачу, після цього користувач переглядає статистику та вивчає дані відносно кожної гри і врешті решт робить висновки стосовного того, що треба покращити задля отримання результату або досягнення поставленої мети.

Застосування розробленого програмного забезпечення відкриває можливості для підрахунку прогнозованої потенційної ефективності гри певних гравців на певних позиціях, формування оптимального для визначеної ситуації методики тренування, а також планування розвитку у кіберспорті.

## **2.2 Функціональна модель обробки кіберспортивних статистичних даних та за стосунку**

Для забезпечення отримання даних, необхідно розробити алгоритм взаємодії з сервісом Steam. Для цього існує бібліотека OpenID яка розроблена компанією VALVE. Ця бібліотека має всі необхідні функції, які дозволяють звернутися до сервісу та отримати відповідь у вигляді масиву. Ця бібліотека написана на мові програмування php і достатньо зручна для використання в розробці вебзастосунку.

Базовою функцією у вебзастосунку має бути авторизація. Саме ця функція дає можливість отримувати дані з сервісу Steam, оскільки при авторизації користувач використовує діючий аккаунт до якого прив'язані всі ігри користувача(рис 2.2). Після авторизації, необхідно перевірити чи існує аккаунт. Якщо таких аккаунт існує, тоді робиться запит до бази для ініціалізації користувача а також до Steam задля отримання деталей щодо користувача.



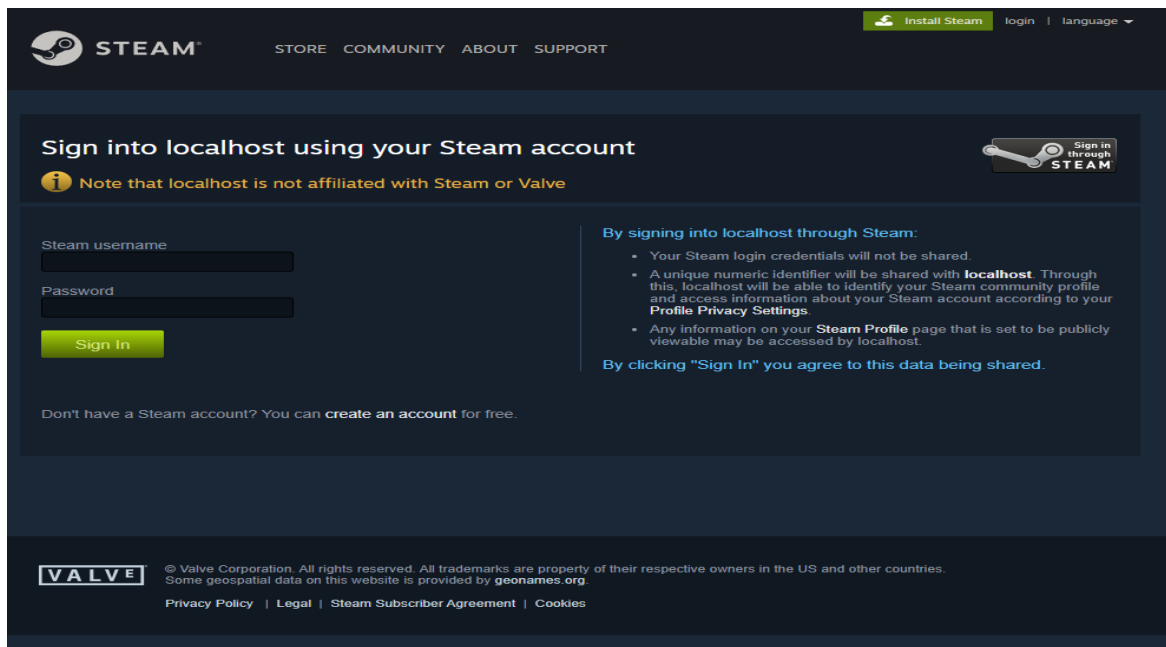


Рисунок 2.2 – Сторінка для авторизації з сервісом Steam

Якщо у користувача немає аккаунту Steam авторизація не може бути виконана, в такому випадку користувачу необхідно створити аккаунт Steam на сайті сервісу, оскільки бібліотека для взаємодії з даним сервісом не передбачає можливість створення аккаунту.

Після авторизації користувачу надається можливість перегляд статистики у спеціальному розділі вебсайту куди будуть відображені всі ігри які прив'язані до користувача та уніфіковані дані щодо обраної гри. Системою передбачені отримання різних статусів користувачів. Це означає, що у користувачів існує можливість отримання VIP-статусу. Цей статус дає можливість дивитися більш детальну статистику найпопулярніших ігор сервісу Steam якщо ці проекти прив'язані до аккаунту користувача.

Щоб отримати статистику для цих проектів необхідно використовувати окремі API, які розроблені для безпосередньої взаємодії з клієнтом гри. Також інформаційною системою передбачено отримання порад базуючись на аналіз штучного інтелекту. Для отримання цього VIP-статусу необхідно його купити на сайті.

За допомогою API, робиться запит до бази даних Steam(рис 2.3). Потім, Steam видає масив даних де написано про всі ігри,які є у користувача на аккаунті.

Після того, як було отримано всі ігри користувача, необхідно зробити ще один запит по обраній грі.

Після отримання відповідей від бази по цим двом запитам, можна:

- переглянути статистику по обраній грі;
- переглянути інформацію про користувача Steam

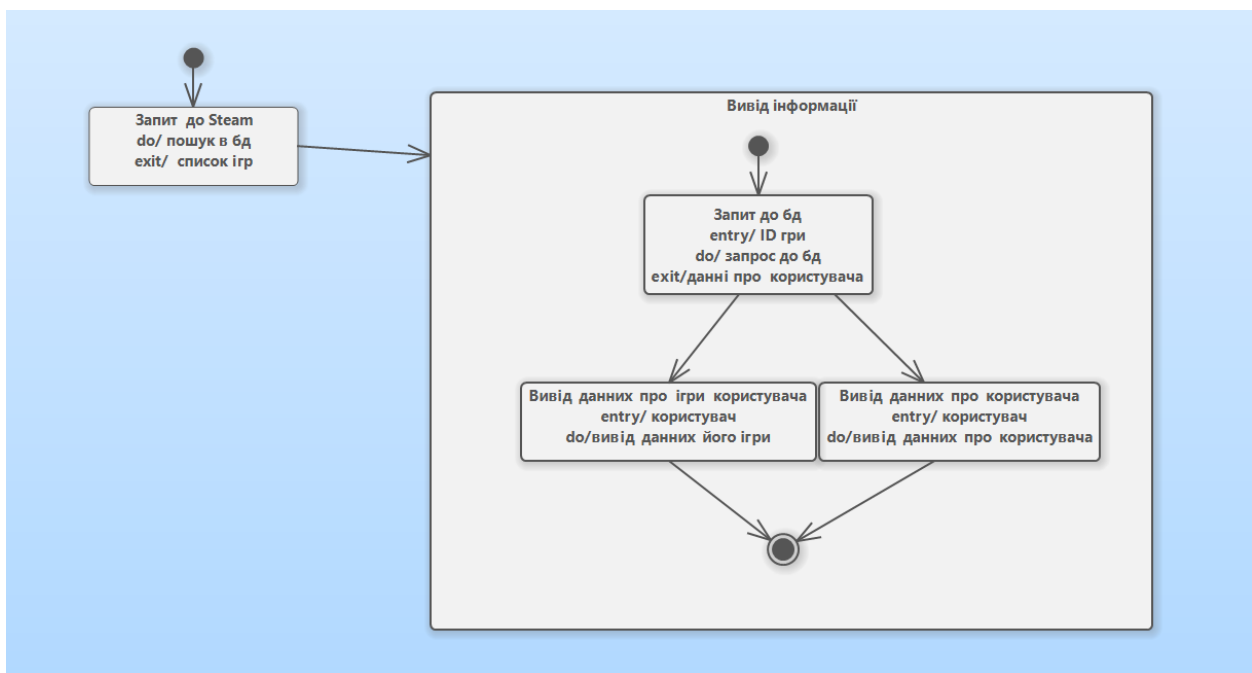


Рисунок 2.3 - Діаграма станів для «Отримання статистики»

Крім перегляду статистики, користувач має можливість переглядати свій профіль. Цей профіль можливо змінювати на сайті, але ці зміни будуть зроблені тільки на стороні за стосунку, тобто дані у самому сервісі Steam не будуть зміненні.

Для модерації інформаційної системи необхідна наявність адміністратора. У нього буде можливість змінювати дані на стороні серверу, а саме видаляти користувача, додавати його або робити зміни для поточних даних. Для

забезпечення зручності користування системою повинна бути створена панель адміністратора с графічним інтерфейсом.

Адміністратор має усю повноту прав: є права VIP-користувача, за відмінності що він бачить усіх користувачів, та може редагувати інших користувачів системи, а саме видалити чи додати їх.

Функціональна модель може бути зображена діаграмою(рис 2.4)

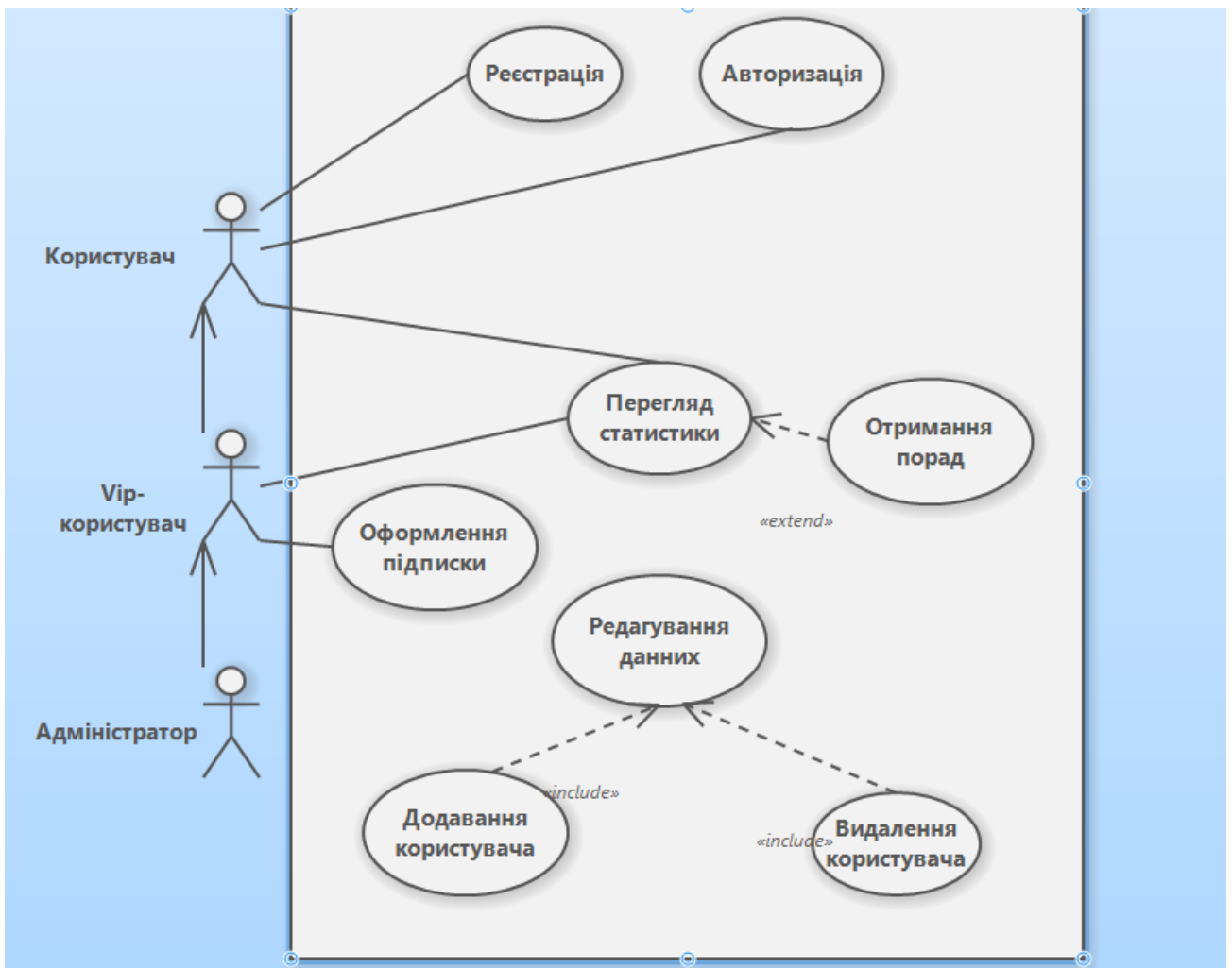


Рисунок 2.4 – Діаграма використання

Також на рисунку 2.5 наведено можливі стани для користувача інформаційної системи.

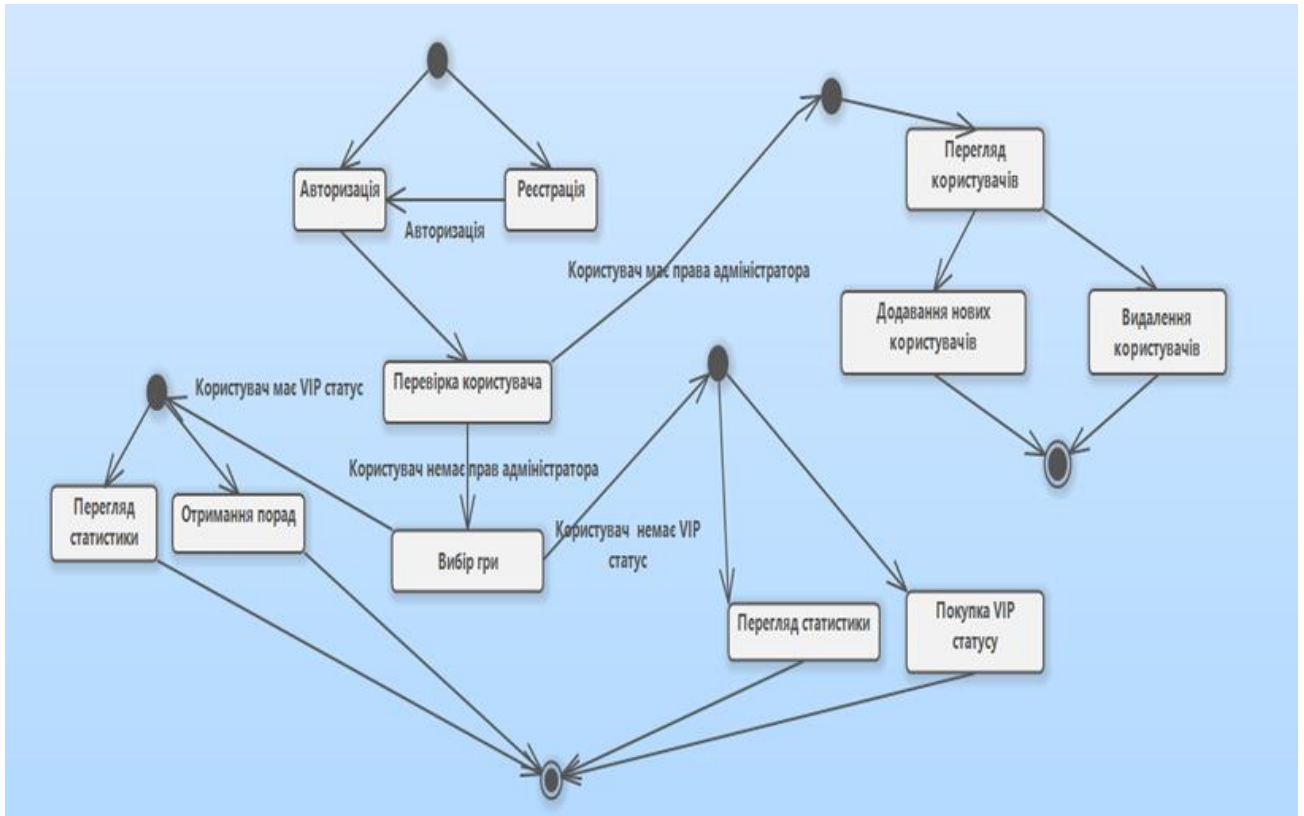


Рисунок 2.5 – Діаграма станів

## 2.3 Сценарії використання системи

### Сценарій 1. Вхід до системи.

Відвідувач потрапляє на сайт і йому одразу пропонують увійти під своїм логіном та паролем. Після введення, система звіряється з базою даних на предмет збігу. Якщо така пара логін-пароль існує, він входить у систему, якщо ні - повідомляється що логін та пароль невірні.

### Сценарій 2. Перегляд статистики.

*Головний сценарій. (Успішний)*

Користувач переходить на свою сторінку та обирає гру. Після цього користувачу відображається список, в якому написана вся статистика обраної гри.

*Альтернативні сценарії.*

1. Користувач не має ігри на акаунті Steam. Статистика не відображається та виводить повідомлення що «Внесених результатів не має».

2. Помилка системи. З невідомих причин не має доступу до даних або сторінка недоступна.

Таблиця 2.1 – Отримання даних

<b>Scope</b>	Система перегляду статистики
<b>Level</b>	Відобразити статистику користувача
<b>Primary Actor</b>	Користувач
<b>Stakeholders and interests:</b>	<ol style="list-style-type: none"> <li>1. Звичайний користувач: Зацікавлений перегляді базової статистики</li> <li>2. VIP-користувач: Зацікавлений у отриманні порад</li> </ol>
<b>Preconditions</b>	Користувач виконав вхід до системи
<b>Main Success Scenario</b>	<ol style="list-style-type: none"> <li>1. Користувач здійснює вхід до системи</li> <li>2. Обирає гру яка його цікавить</li> <li>3. Продивляється статистику</li> </ol>
<b>Extensions</b>	<ol style="list-style-type: none"> <li>1. Користувач немає акаунту Steam</li> <li>2. Помилка системи.</li> </ol>
<b>Special Requirements</b>	1. Пристрій з доступом до інтернету та який задовольняє мінімальні системні вимоги браузера. Акаунт Steam
<b>Frequency of Occurrence</b>	Система може працювати 24/7.

## **Висновки до розділу 2**

В розділі розроблено компоненти інформаційної технології аналізу кіберспортивних статистичних даних - інформаційну модель кіберспортивного гравця, функціональну модель обробки статистичних даних, наведена діаграма можливостей використання системи.

Використання в межах інформаційних технологій розробленої інформаційної моделі відкриває можливості для аналізу ефективності гравця в обраній дисципліні, отримання даних щодо поточної ефективності та загальні статистичні дані відносно обраної кіберспортивної дисципліни.

Розроблена функціональна модель дає змогу визначити основні функціональні можливості інформаційної технології. Це надає розуміння відносно того як і які дані користувача обробляти та відображати у застосунку. Було визначено всі можливості користувача у взаємодії з інформаційною технологією.

## 3 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕПЕЧЕННЯ

### 3.1 Аналіз функціоналу та формування стеку технологій проекту

У зв'язку тим, що переважна більшість бібліотек для взаємодії з Steam розроблено на мові програмування PHP, то в рамках цього проекту буде використана мова що може взаємодіяти з ним, тобто PHP (рис 3.1).



Рисунок 3.1 – Мова програмування PHP

*PHP* — це мова програмування загального призначення з відкритим кодом. PHP розроблений для веб-розробки, і його код можна реалізувати безпосередньо в HTML.

Замість того, щоб регулярно виводити HTML-код за допомогою команд (як, наприклад, у Perl або C), PHP-скрипти містять HTML із вбудованим кодом. PHP-код позначений спеціальними відкриваючими та закриваючими тегамі `<?Php?>`, які дозволяють «перейти» в «режим PHP» і вийти з нього.

PHP відрізняється від JavaScript тим, що PHP-скрипти виконуються на сервері і генерують HTML, який надсилається клієнту. Якщо у вас на сервері є сценарій, подібний до наведеного вище, клієнт отримає лише результат його виконання. Ви навіть можете налаштувати свій сервер так, щоб процесор PHP

обробляв звичайні файли HTML, щоб клієнти навіть не могли знати, чи отримують вони звичайні файли HTML або результати сценаріїв.

PHP має три основні області застосування.

1. Створення сценарію, який виконується на стороні сервера. PHP традиційно найбільше використовувався таким чином. Для цього потрібно зробити три речі. Інтерпретатори PHP (у вигляді програм CGI або серверних модулів), веб-сервери та браузері. Щоб мати можливість переглядати результати PHP-скриптів у браузері, потрібен робочий веб-сервер і встановлений PHP. Ви можете переглянути результат роботи програми PHP у своєму браузері, отримавши сторінку PHP, згенеровану сервером. Якщо ви лише експериментуєте, ви можете використовувати свій домашній комп'ютер замість сервера.

2. Створення сценарію для виконання в командному рядку. Ви можете створювати PHP-скрипти, які запускаються без сервера або браузера. Все, що вам потрібно, це аналізатор PHP. Цей спосіб використання PHP чудово підходить для сценаріїв, які потрібно періодично запускати, наприклад, за допомогою cron (на платформах \*nix або Linux) або планувальника завдань на платформах Windows. Ці сценарії також можна використовувати для простих завдань обробки текстів.

3. Створення програми, яка запускається на клієнті. PHP може бути не найкращою мовою для створення такого типу, але якщо добре знаєте PHP і хочете використовувати деякі з його функцій у своїй клієнтській програмі, ви можете використовувати PHP-GTK для створення такої програми. Подібним чином можна створювати кросплатформні програми. PHP-GTK — це розширення PHP, яке не постачається з основним дистрибутивом PHP.

Для забезпечення візуальної частини вебсайту використано CSS (скорочення від Cascading Style Sheets, що означає Cascading Style Sheets) – це спеціальна мова (мова стилів), яка описує вигляд документа (як і де відобразити 2022р.



елементи веб-сторінки), написаного мовою розмітки. CSS найчастіше використовується для документів, позначених як HTML, XHTML і XML(рис 3.2).



Рисунок 3.2 – Мова стилю сторінки css

Однією з головних переваг використання CSS є можливість відокремити вміст сторінки від її дизайну. Такий поділ покращує сприйняття та доступність контенту, забезпечує більшу гнучкість та контроль над відображенням вмісту за різних умов, робить контент більш структурованим та простішим, усуває дублювання тощо. CSS дає можливість: Відображати один і той же документ у різних стилях.

Багатосторінковий дизайн для різних пристроїв. Наприклад, на екрані дизайн буде мати велику ширину, і меню не відобразатиметься під час друку, тоді як на смартфоні меню буде внизу під вмістом.

Скоротити час завантаження сторінки веб-сайту, перенісши правила відображення в окремі файли CSS. У цьому випадку браузер завантажує лише структуру документа та дані, що зберігаються на сторінці, а правила стилю для цих даних завантажуються браузером і кешуються лише один раз.

Легкість подальших змін дизайну. Не потрібно редагувати кожен сторінку, необхідна зміна файлу CSS.

Інші варіанти оформлення. Наприклад, використовуючи розмітку CSS, реалізує можливість зробити меню завжди видимим під час прокручування або видалити підкреслення з посилань.

Дозволяє створювати складні та витончені дизайнерські прийоми.

Власне, це основна каскадна робота metRules в CSS (пріоритети, ваги). Це дозволяє отримати очікувані результати, коли до елемента одночасно застосовуються кілька правил стилю. і створення цієї технології.

Разом з CSS було використано HTML(рис 3.3) – Стандартизована мова розмітки для перегляду вебсторінок. Веббраузер отримує HTML-документ із сервера через HTTP/HTTPS або відкриває його з локального диска та інтерпретує код в інтерфейс, який буде відображатися на екрані монітора.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою структури HTML зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані в відтворену сторінку. HTML надає інструменти для створення структурованих документів, які представляють структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, посилання та інші елементи. Елементи HTML описуються тегами, записаними в кутових дужках. Такі теги, як `<img />` або `<input />` відображають вміст безпосередньо на сторінці. Інші теги, такі як `<p>`, оточують і надають інформацію про текст, а також можуть включати інші теги як дочірні елементи. Замість відображення тегів HTML браузери використовують їх для інтерпретації вмісту сторінки.



Рисунок 3.3 – Мова розмітки HTML

Для забезпечення адаптивності було використано Vue.js — це платформа JavaScript, яка використовує шаблони MVVM для створення користувацьких інтерфейсів на основі моделі даних, а також є прогресивною платформою для розробки інтерфейсів користувача за допомогою реактивного зв'язування даних(рис 3.4). На відміну від інших монолітних фреймворків, Vue був розроблений з нуля і його можна постійно налаштовувати. Основна бібліотека зосереджена лише на шарі відображення і може бути легко інтегрована з іншими бібліотеками та існуючими проектами. З іншого боку, Vue також готовий стати основою для простих SPA і складних додатків у поєднанні з сучасними інструментами та допоміжними бібліотеками.



Рисунок 3.4 – Vue.JS

### 3.2 Проектування бази даних

Задля забезпечення збереження інформації про користувача, необхідно розробити базу даних та визначити сутності предметної області та їх властивості. На рисунку 3.5 зображено модель бази даних ІС, яка передбачає такі таблиці: “Users” (користувачі), “Order” (замовлення користувача), “UserGame” (Всі ігри користувача), “OwnedGames” (Окрема вибрана гра), “UserLocation” (Локація користувача), “Merchants” (Метод оплати).

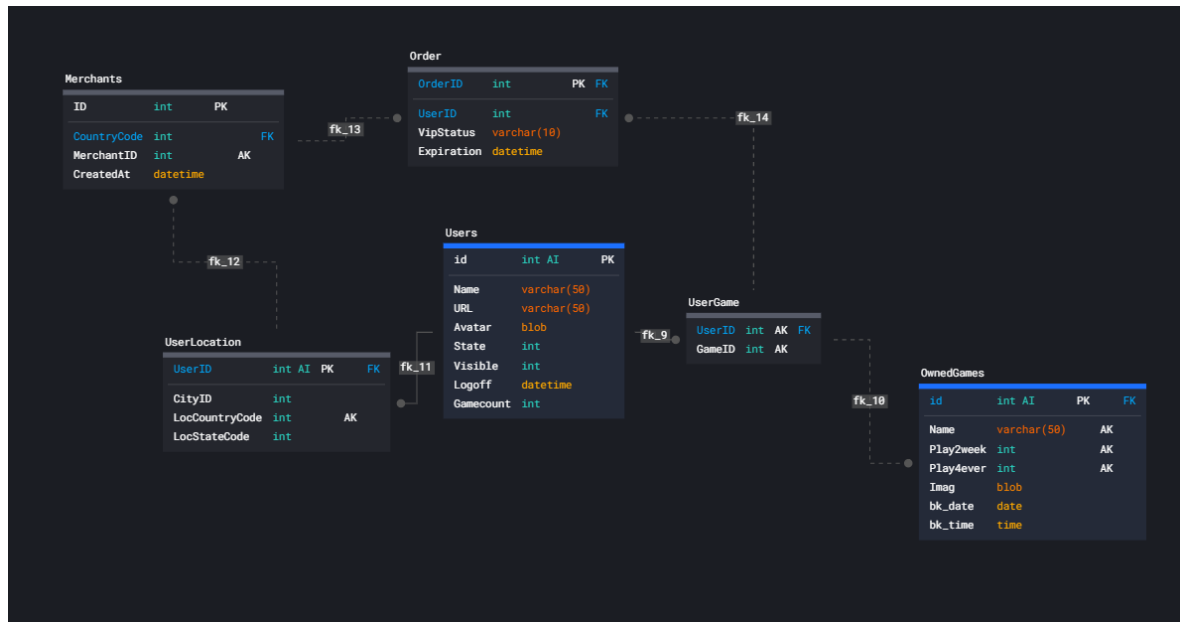
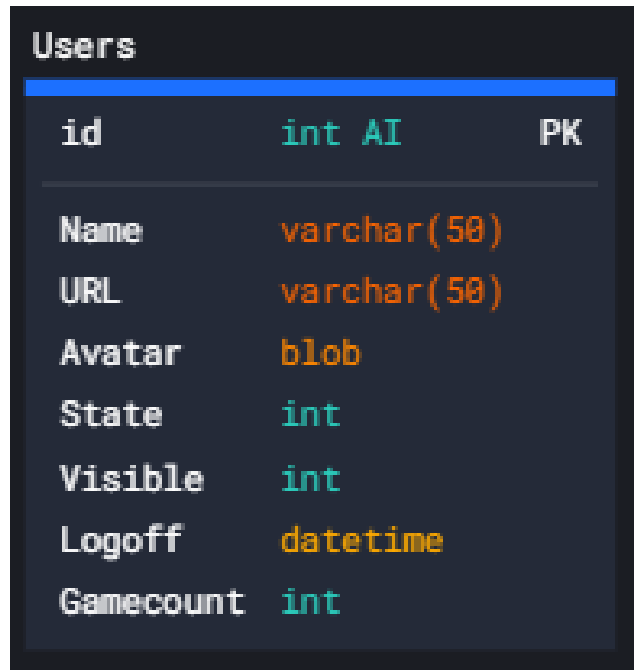


Рисунок 3.5 – Модель бази даних

Кожна таблиця БД призначена для збереження даних окремої сутності. Властивості сутностей є атрибутами для кожної таблиці.

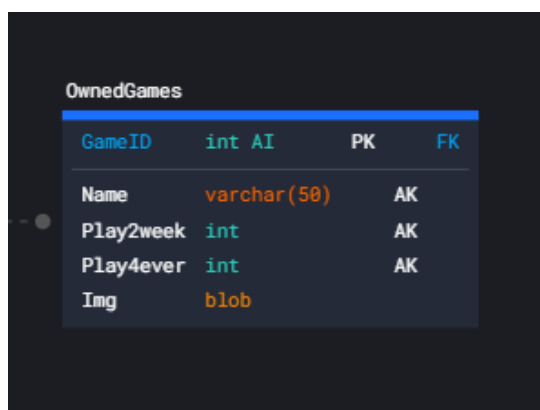
Таблиця “Users” відповідає за зберігання даних про гравців та має наступні атрибути: ID, Name, URL, Avatar, State, Visible, Logoff, Gamecount. Поле ID відповідає за присвоєння кожному користувачу унікального ідентифікатора яких співпадає з ідентифікатором користувача в сервісі Steam, має тип даних int. Поле Name зберігає нікнейм користувача, тип даних – varchar(50). Поле URL відповідає за отримання посилання на аккаунт користувача, тип даних- varchar(50). Поле Avatar отримує зображення аватару(картинки) користувача, тип даних - blob. Поле State відповідає за поточний мережевий статус користувача у сервісі Steam, тип даних – int. Поле Visible отримує дані стосовно того, чи можна зчитувати інформацію з акканта за допомогою якого, була зроблена авторизація, тип даних – int. Поле Logoff відповідає за збереження дати коли останній раз користувач був у системі, тип даних – datetime. Поле Gamecount отримує кількість ігор, які прив’язані до аканту користувача, тип даних – int(рис 3.6).



Users		
id	int AI	PK
Name	varchar(50)	
URL	varchar(50)	
Avatar	blob	
State	int	
Visible	int	
Logoff	datetime	
Gamecount	int	

Рисунок 3.6 – Таблиця Users

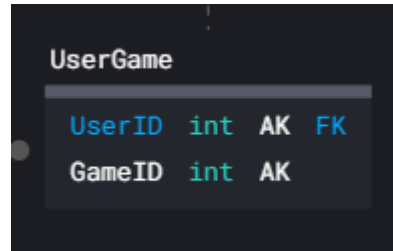
Таблиця “OwnedGames ” відповідає за зберігання даних щодо ігор, які є в бібліотеці Steam та має наступні атрибути: GameID, Name, Play2week, Play4ever, Img. Поле GameID зберігає в собі ідентифікатор гри в сервісі Steam, тип даних- int. Поле Name відповідає за назву гри, тип даних- varchar(50). Поле Play2week відповідає кількість годин проведених у грі протягом двох тижнів, тип даних- int. Поле Play4ever відповідає кількість годин проведених у грі з першого запуску та до сьогодні, тип даних- int. Поле Play4ever зберігає хеш-код логотипу гри для подальшого виведення на сторінку, тип даних- blob(рис 3.7).



OwnedGames			
GameID	int AI	PK	FK
Name	varchar(50)	AK	
Play2week	int	AK	
Play4ever	int	AK	
Img	blob		

Рисунок 3.7 – Таблиця OwnedGames

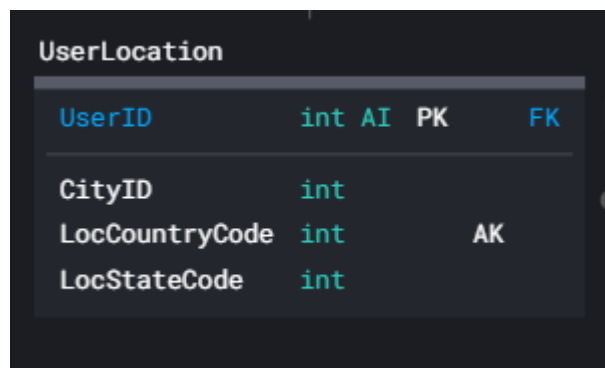
Таблиця “UserGames” відповідає за взаємодію таблиці OwnedGames, Users та Orders та має наступні атрибути: UserID та GameID(рис 3.8).



UserGame			
UserID	int	AK	FK
GameID	int	AK	

Рисунок 3.8 – Таблиця UserGames

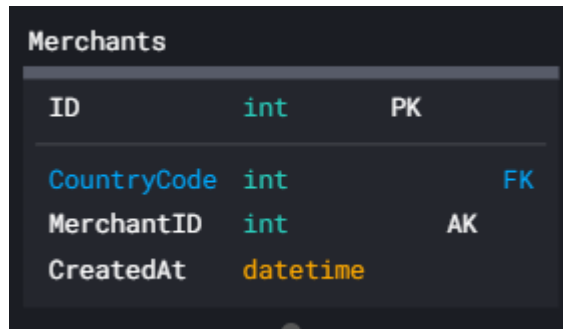
Таблиця “ UserLocation” відповідає за зберігання даних щодо місця проживання користувача, та має наступні атрибути: UserID, CityID, LocCountryCode, LocStateCode. Поле UserID зберігає дані про поточного користувача та має зв’язок з таблицею Users. Поле CityID зберігає дані про місто користувача. Поле LocCountryCode зберігає дані про країну користувача. Поле LocStateCode зберігає дані про область(регіон) користувача(рис 3.9).



UserLocation			
UserID	int	AI	PK FK
CityID	int		
LocCountryCode	int		AK
LocStateCode	int		

Рисунок 3.9 – Таблиця UserLocation

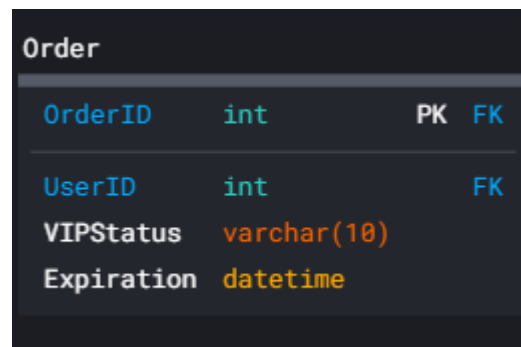
Таблиця “ Merchants” відповідає за зберігання даних щодо методи оплати користувача, та має наступні атрибути: ID, CountryCode, MerchantID, CreatedAt. Поле ID є ідентифікатором кількості транзакцій користувача. Поле CountryCode зберігає дані про країну користувача до якої прив’язані методи оплати та має зв’язок з таблицею UserLocation. Поле MerchantID зберігає дані про транзакцію. Поле CreatedAt зберігає дані про дату створення замовлення(рис 3.10).



Merchants			
ID	int	PK	
CountryCode	int		FK
MerchantID	int	AK	
CreatedAt	datetime		

Рисунок 3.10 – Таблиця Merchants

Таблиця “ Order ” відповідає за зберігання даних щодо оплати замовлення користувача, та має наступні атрибути: OrderID, UserID, VIPStatus, Expiration. Поле OrderID є ідентифікатором замовлення користувача. Поле UserID зберігає дані користувача який здійснив платіж та має зв’язок з таблицею Users. Поле VIPStatus зберігає дані про надання користувачу VIP-статусу. Поле Expiration зберігає дані про дату закінчення VIP-статусу. (рис 3.11).



Order			
OrderID	int	PK	FK
UserID	int		FK
VIPStatus	varchar(10)		
Expiration	datetime		

Рисунок 3.11 – Таблиця Order

### 3.3 Моделювання структури

Для забезпечення створення об’єктів, необхідно розробити класи(рис 3.12)



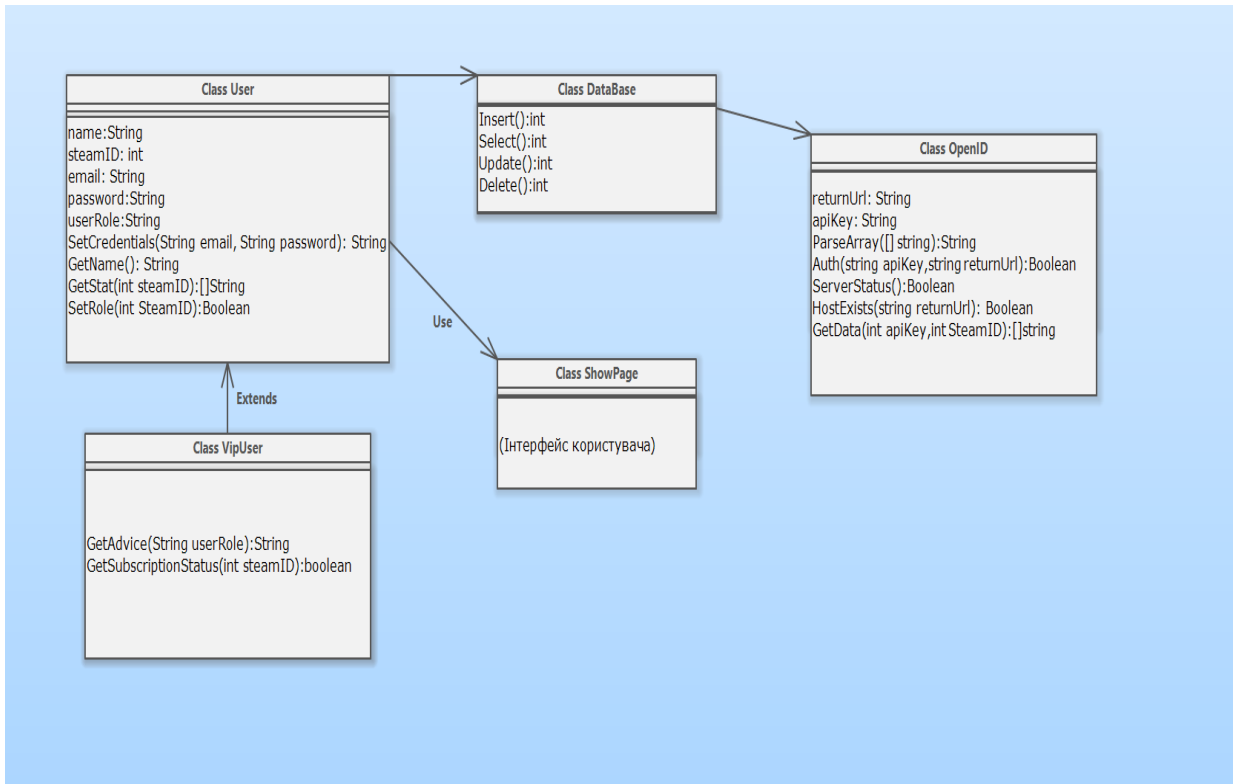


Рисунок 3.12 – Діаграма класів

### Опис класу User

User – клас, який відповідає за дані користувача;

Name – поле, відповідає за імя користувача

SteamID – поле, відповідає за ідентифікатор аккаунту користувача

email – поле, відповідає за пошту користувача

UserRole- поле, відповідає за роль користувача

String GetName- метод, відповідає за отримання імені користувача

Boolean SetRole - метод, відповідає за зміну ролі користувача

String GetStat - метод, відповідає за відображення статистики

### Опис класу VipUser

VipUser – наслідує клас User, але має додаткові методи

String GetAdvice - метод, відповідає за отримання поради

Boolean GetSubscriptionStatus - метод, відповідає за перегляд активності підписки

Опис класу DataBase

int insert() – метод, за допомогою якого виконується запис даних до БД;

int select() - метод, за допомогою якого виконується отримання даних з БД

int update() - метод, за допомогою якого виконується зміна даних у БД;

int delete() - метод, за допомогою якого виконується видалення даних з БД;

Опис класу ShowPage

ShowPage – клас, який виводить шаблони інтерфейсу для користувача;

Опис класу OpenID

OpenID – клас, відповідає за комунікацію з Steam

string returnUrl– поле, містить посилання на домашню сторінку після авторизації

String apiKey - поле, містить Арі ключ для взаємодії з Steam

String ParseArray – Метод, парсить масив відповіді

Boolean Auth - метод, відповідає за успішність авторизації

Boolean ServerStatus – метод, перевіряє доступність серверу Steam

Boolean HostExists - метод, перевіряє чи існує сторінка на яку буде перенаправлено після авторизації

String GetData – метод, сортує дані з масиву та виводить на сторінку

Для організації коду, логічного розміщення компонентів використовують діаграму компонентів.

Нічого не заважає розмістити усі файли в одному -двох пакетах або директоріях, але це негативно позначиться на якості супроводу системи та при наявності різних прав доступу до коду може порушити конфіденційність.

Також, наразі робота майже усіх фреймворків організована пакетно, модульно, і при необхідності необхідно під'їднати лише його(рис 3.13).

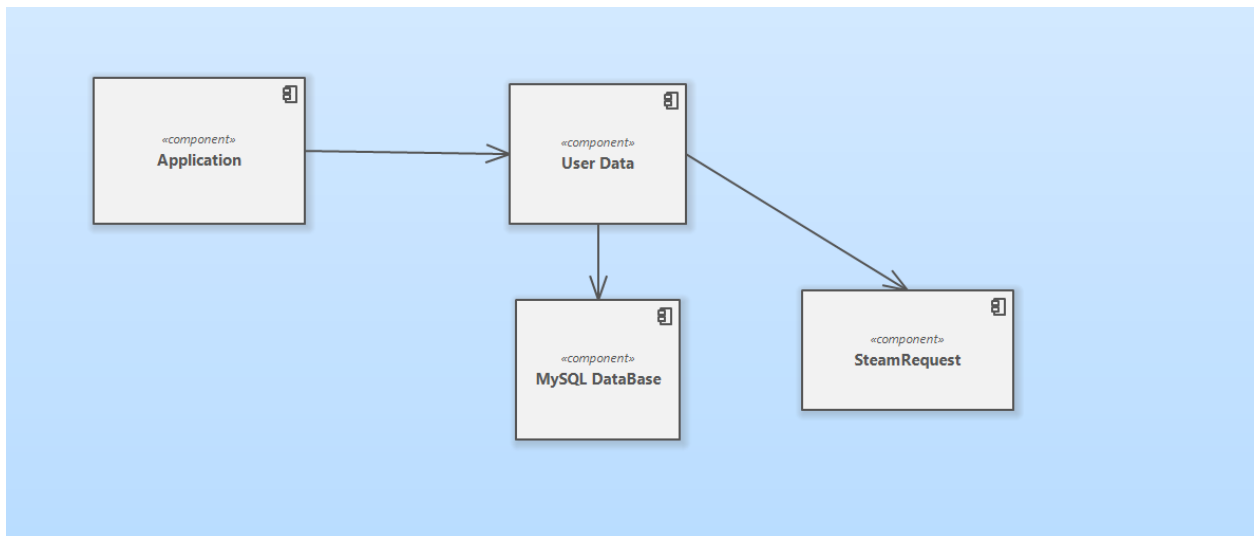


Рисунок 3.13 - Діаграма компонентів

### Висновки до розділу 3

Використовуючи інформаційну та функціональну моделі, було сформовано стек технологій для проекту, спроектовано базу даних, яка буде обробляти всі необхідні вхідні дані, розроблено структуру класів та їх функціональні можливості.

Для розробки інформаційної системи було застосовано фреймворк Vue JS й мову програмування PHP, СКБД MySQL та середовище розробки Visual Studio Code.

## 4 РОЗРОБКА ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ

### 4.1 Розробка функціоналу отримання та обробки даних

Для забезпечення взаємодії з сервісом Steam необхідно використовувати SteamAPI та бібліотеку OpenID. Для того щоб забезпечити цю взаємодію слід сгенерувати API ключ (рис 4.1), це можна зробити на сторінці розробника сервісу, авторизувавшись в свій аккаунт та вказавши доменне ім'я на якому буде працювати система.

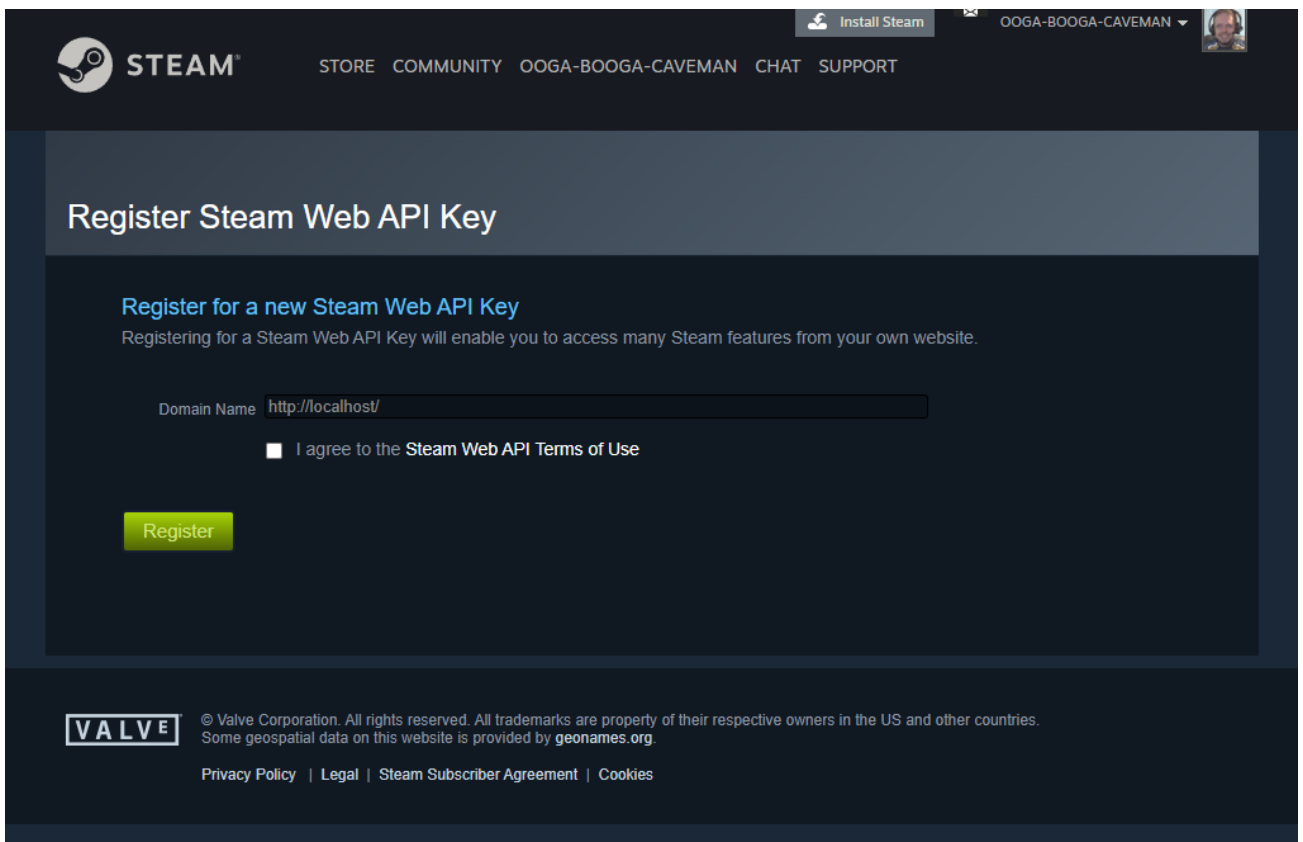


Рисунок 4.1 - Створення API ключа

Після того як створено API ключ, необхідно виконати ініціалізацію ключових змінних, а саме `apikey`, `domainname`, `logoutpage`, `loginpage`. Для змінної `apikey` слід вказати той API ключ, який отримано, `domainname` отримує інформацію про основну сторінку, з якої виконується перенаправлення на

авторизацію, `logoutpage` – сторінка на яку буде перенаправлено користувача в разі виходу з аккаунта, `loginpage` – сторінка до якої буде перенаправлено користувача у випадку успішної авторизації. Для змінних `loginpage` та `logoutpage` можна не вказувати значення, в такому випадку буде використовуватись значення змінної `domainname` (рис 4.2). Також після ініціалізації виконується перевірка значень.

```
<?php
//Version 4.0
$steamauth['apikey'] = "90B26B0398D2C5C9B5CA62DC1265B21D";
$steamauth['domainname'] = "http://localhost/index.php";
$steamauth['logoutpage'] = "";
$steamauth['loginpage'] = "";

// System stuff
if (empty($steamauth['apikey'])) {die("<div style='display: block; width: 100%; background-color: red; text-a
if (empty($steamauth['domainname'])) {$steamauth['domainname'] = $_SERVER['SERVER_NAME'];}
if (empty($steamauth['logoutpage'])) {$steamauth['logoutpage'] = $_SERVER['PHP_SELF'];}
if (empty($steamauth['loginpage'])) {$steamauth['loginpage'] = $_SERVER['PHP_SELF'];}
?>
```

Рисунок 4.2 - Ініціалізація змінних для авторизації

Коли ці змінні ініціалізовано та перевірено їх значення, користувач потрапляє на домашню сторінку вебсайту (рис 4.3) та йому не будуть доступні сторінки «Статистика, Личный кабинет» оскільки авторизація не виконана та дані з сервісу Steam не отримано. Для того, щоб користувач мав можливість переглянути ці сторінки йому необхідно натиснути на кнопку у правому верхньому куті «Sing in through STEAM» та виконати авторизацію. Ініціалізація цієї кнопки виконана за допомогою наступної функції -

```
function loginbutton($buttonstyle = "square") {
    $button['rectangle'] = "01";
    $button['square'] = "02";
    $button = "<a href='?login' class='col-1 header__login'><img
src='/img/steamauth.png'></a>";
```

```
echo $button;  
}
```

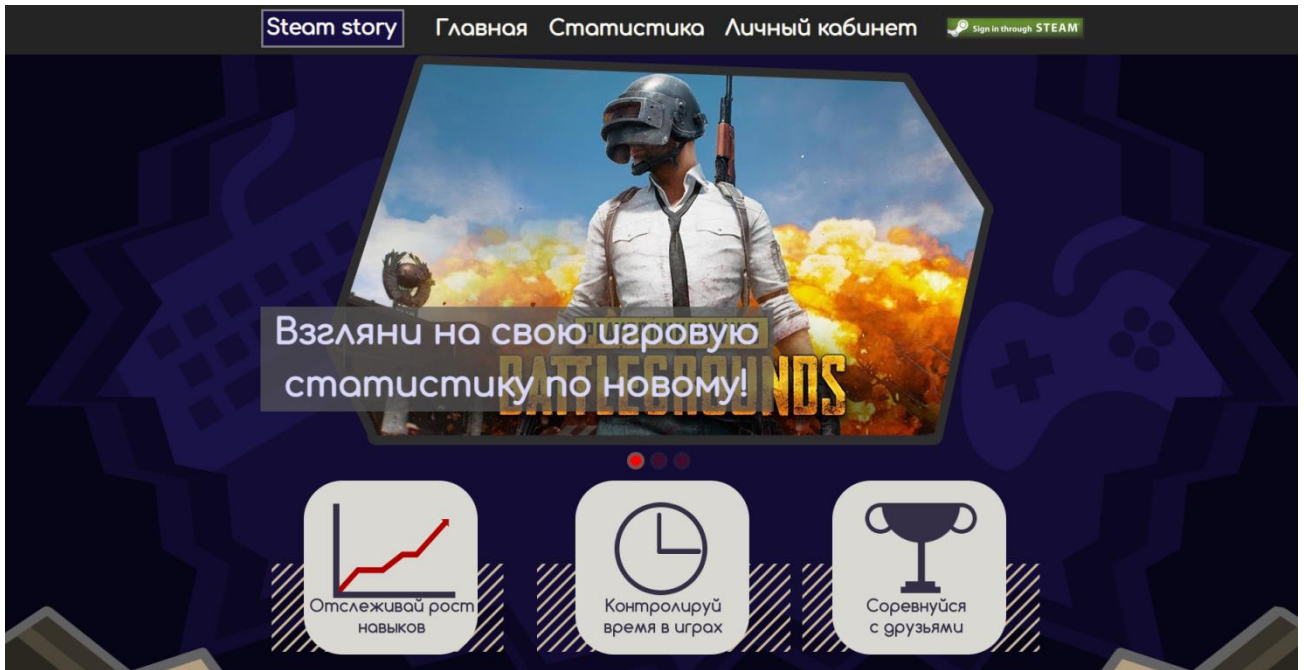


Рисунок 4.3 – Дизайн головної сторінки за стосунку

При натисканні на кнопку авторизації, відбувається створення об'єкту класу LightOpenID та використовуються методи цього класу і перевіряється чи є помилки при виконанні наступного коду -

```
if (isset($_GET['login'])) {  
    require 'authentication/openid.php';  
    try {  
        require 'SteamConfig.php';  
        $openid = new LightOpenID($steamauth['domainname']);  
  
        if (!$openid->mode) {  
            $openid->identity = 'https://steamcommunity.com/openid';  
            header('Location: ' . $openid->authUrl());  
        }  
    }  
}
```

```
} elseif ($openid->mode == 'cancel') {
    echo 'User has canceled authentication!';
} else {
    if($openid->validate()) {
        $id = $openid->identity;
        $ptn = "/^https?:\\\/steamcommunity\\.com\\/openid\\/id\\(7[0-9]{15,25}+\\)$/";
        preg_match($ptn, $id, $matches);

        $_SESSION['steamid'] = $matches[1];
        if (!headers_sent()) {
            header('Location: '.$steamauth['loginpage']);
            exit;
        } else {
            ?>
            <script type="text/javascript">
                window.location.href="<?=$steamauth['loginpage']?>";
            </script>
            <noscript>
                <meta http-equiv="refresh"
content="0;url=<?=$steamauth['loginpage']?>" />
            </noscript>
            <?php
            exit;
        }
    } else {
        echo "User is not logged in.\n";
    }
}
}
```

```
} catch(Exception $e) {  
    echo $e->getMessage();  
}  
}
```

Після виконання цього коду кнопка авторизації була замінена на кнопку виходу та було додано аватар користувача (рис 4.4)

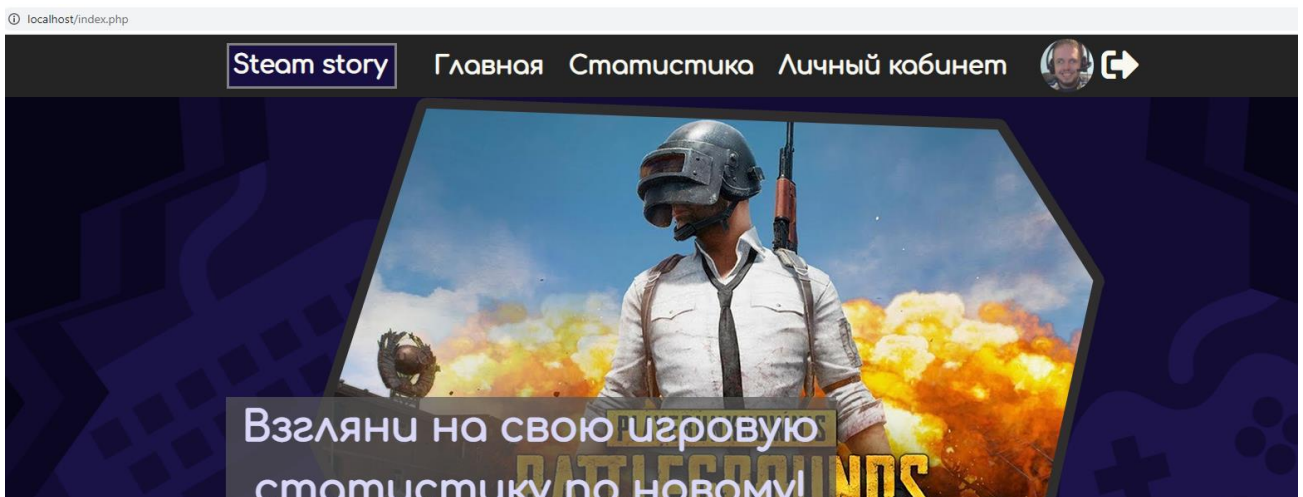


Рисунок 4.4 – Результат роботи функції авторизації

У разі успішної авторизації відбувається запит до сервісу Steam щоб отримати дані користувача за допомогою функції `file_get_contents`. Відповіддю сервісу є json масив, який необхідно разпарсити за допомогою функції `json_decode`. Ці дані записуються у поточну сесію.

```
if (empty($_SESSION['steam_uptodate']) or  
empty($_SESSION['steam_personaname'])) {  
    require 'SteamConfig.php';  
    $url =  
file_get_contents("http://api.steampowered.com/ISteamUser/GetPlayerSummaries/v0  
002/?key=".$_SESSION['apikey']."&steamids=".$_SESSION['steamid']);  
    $content = json_decode($url, true);
```



```
$_SESSION['steam_steamid'] = $content['response']['players'][0]['steamid'];
$_SESSION['steam_communityvisibilitystate'] =
$content['response']['players'][0]['communityvisibilitystate'];
$_SESSION['steam_profilestate'] = $content['response']['players'][0]['profilestate'];
$_SESSION['steam_personaname'] =
$content['response']['players'][0]['personaname'];
$_SESSION['steam_lastlogoff'] = $content['response']['players'][0]['lastlogoff'];
$_SESSION['steam_profileurl'] = $content['response']['players'][0]['profileurl'];
$_SESSION['steam_avatar'] = $content['response']['players'][0]['avatar'];
$_SESSION['steam_avatarmedium'] =
$content['response']['players'][0]['avatarmedium'];
$_SESSION['steam_avatarfull'] = $content['response']['players'][0]['avatarfull'];
$_SESSION['steam_personastate'] =
$content['response']['players'][0]['personastate'];
if (isset($content['response']['players'][0]['realname'])) {
    $_SESSION['steam_realname'] = $content['response']['players'][0]['realname'];
} else {
    $_SESSION['steam_realname'] = "Не указано";
}
$_SESSION['steam_primaryclanid'] =
$content['response']['players'][0]['primaryclanid'];
$_SESSION['steam_timecreated'] =
$content['response']['players'][0]['timecreated'];
$_SESSION['steam_uptodate'] = time();
$_SESSION['steam_gameextrainfo'] =
$content['response']['players'][0]['gameextrainfo'];
$_SESSION['gameserverip'] = $content['response']['players'][0]['gameserverip'];}
```

В результаті отримання відповіді від серверу, необхідно вивести ці дані на сторінку, щоб користувач зміг проглянути відформатовану та структуровану інформацію. Для цього використовується функція `echo`, яка дає можливість виводити на сторінку HTML теги, стилі та бекенд змінні. Для сторінки «Личный кабинет» (рис 4.5) виконується наступний код -

`echo"`

```
<div class='col-lg-6 account__name'>
  <h1>
    Никнейм: $name
  </h1>
</div>
<div class='col-lg-6 account__avatar'>
  <img src=' $avatarfull '>
</div>
<span class='col-lg-12 account__id'>
  ID: $id
</span>
<span class='col-lg-12 account__visibitily'>
";
if($visible==1){
  $visible="Закрытый";
}
else if($visible==2){
  $visible="Закрытый";
}
else{
  $visible="Открытый";
```

```
}  
echo"  
    Аккаунт: $visible  
</span>  
<span class='col-lg-12 account__state'>  
";  
switch ($state) {  
    case '0':  
        $state="Сейчас не в сети";  
        break;  
    case '1':  
        $state="Онлайн";  
        break;  
    case '2':  
        $state="Занят";  
        break;  
    case '3':  
        $state="Нет на месте";  
        break;  
    case '4':  
        $state="Спит";  
        break;  
    case '5':  
        $state="Обмениваются с кем-то";  
        break;  
    case '6':  
        $state="Собирается играть";  
        break;
```

default:

```
$state="Профиль приватный,не удалось получить данные";  
break;    };
```

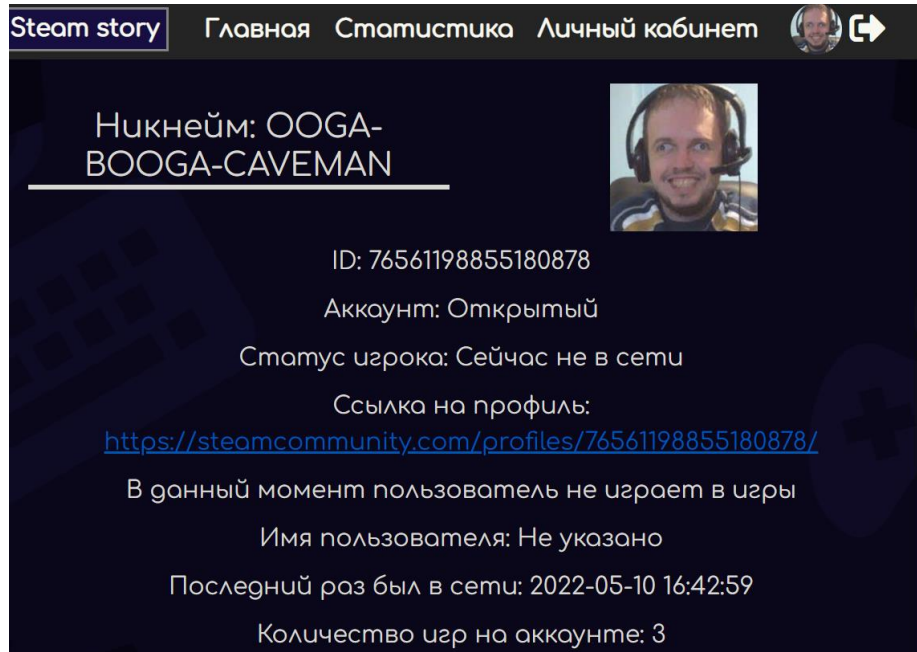


Рисунок 4.5 – Сторінка «Личный кабинет»

Для виведення інформації на сторінку «Статистика» необхідний наступний код -

```
for($i = 0; $i < $length; $i++){  
    $gIMG = $games[$i]['img_icon_url'];  
    $g4EVER =  
round($games[$i]['playtime_forever']/59.99653078924545,0,PHP_ROUND_HALF_  
DOWN);  
    $gNAME = $games[$i]['name'];  
    $gID = $games[$i]['appid'];  
    echo"<div class='col-lg-4' style='text-align: center;'>
```

```
<img  
src='http://media.steampowered.com/steamcommunity/public/images/apps/"$gID."/".  
$gIMG.".jpg' alt = 'error' class='col-lg-4'>  
  <span class='col-lg-12'>  
    Название игры: $gNAME  
  </span>  
  <span class='col-lg-12'>  
    ID игры: $gID  
  </span>  
  <span class='col-lg-12'>  
    Часов сыгранно: $g4EVER  
  </span>  
</div>";  
}
```

Результатом роботи даного коду є статистичні дані користувача, які прив'язані до його аккаунту Steam (рис 4.6)

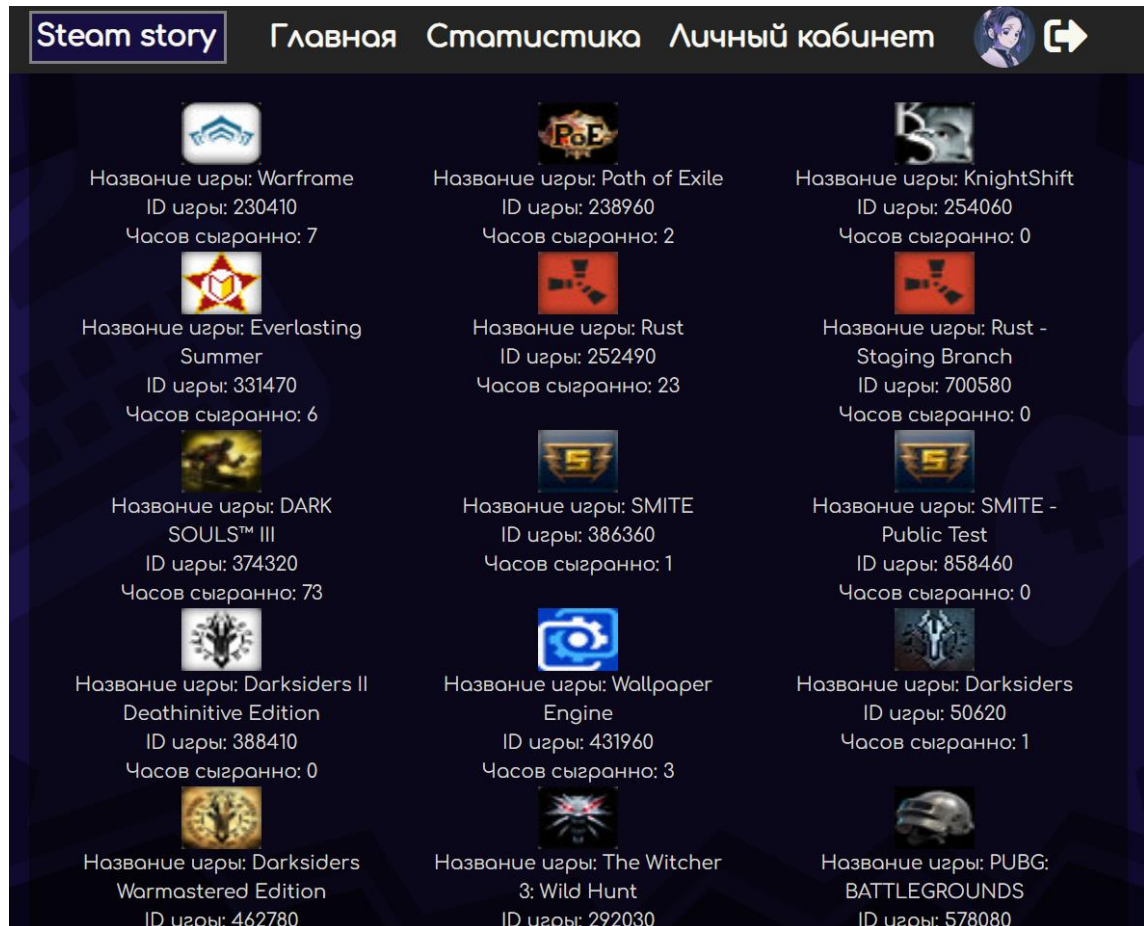


Рисунок 4.6 – Сторінка «Статистика»

Після отримання цих даних з масиву, необхідно додати їх до бази даних. Для цього створено об'єкт класу `mysqli`. За допомогою методу `query` та `fetch_array` виконується взаємодія за БД

```
$link=new mysqli("localhost","root","root","steam");  
if(!$link){  
    die("server doesn't exist");  
}  
$query1="select max(ID) from users";  
$res = $link->query($query1);  
$incr = $res->fetch_array(MYSQLI_NUM);
```

```
$query="insert
users(ID,SteamID,Name,Url,Avatar,State,Visible,Logoff,Gamecount,Interface)
values(($incr[0]+1),'$id','$name','$url','$avatar',$state,$visible,'$logoff',$gamecount,$i
nterface)";
$link->query($query);
$querydata="select * from users where ID=($incr[0]+1)";
$res1 = $link->query($querydata);
$data = $res1->fetch_array(MYSQLI_ASSOC);
$query1="select max(ID) from ownedgames";
$gres = $link->query($query1);
$gincr = $gres->fetch_array(MYSQLI_NUM);
for($i = 0; $i < $length; $i++){
    $gIMG = $games[$i]['img_logo_url'];
    $gID = $games[$i]['appid'];
    $g4EVER = $games[$i]['playtime_forever'];
    $gNAME = $games[$i]['name'];
    $icrem = ++$gincr[0];
    $querygame = "insert into ownedgames(ID,GameID,Name,Play4ever,Imag)
values($icrem,'$gID','$gNAME','$g4EVER','$gIMG)";
    $link->query($querygame);
}
$link->close();
```

Для забезпечення адаптивності використано фреймворк Bootstrap - це відкритий та безкоштовний HTML, CSS та JS фреймворк, який використовується веб-розробниками для швидкого створення адаптивних дизайнів сайтів. Основна сфера його застосування – це розробка фронтенд складових сайтів та інтерфейсів адміністраційних панелей. Серед аналогічних

систем (Foundation, UIKit, Semantic UI, InK та ін) фреймворк Bootstrap є найпопулярнішим.

Для динамічності застосовується фреймворк Vue JS. Для цього необхідно виконати ініціалізацію об'єкту Vue.

```
new Vue({
  el: '#vue-header-js , #vue-main-js',
  data: {} ,
  methods: {
    restore_main() {
      $('.swiper-container').show();
      $('.steam_option').show();
      $('.info').show();
    },
    clear_main() {
      $('.swiper-container').hide();
      $('.steam_option').hide();
      $('.info').hide();
    },
    show_account(){
      $('.account').show();
    },
    hide_account(){
      $('.account').hide();
    },
    show_statistic(){
      $('.statistic').show();
    },
    hide_statistic(){
```



```
    $('.statistic').hide();  
  }  
}  
});
```

Після ініціалізації об'єкту доступні всі методи об'єкту та в меню було додано обробник подій для кліків на пункти меню в верхній частині сайту.

```
<ul class="header__navigation col-8">  
  <li>  
    <a href="#"  
      @click="hide_account(),hide_statistic(),restore_main()">Главная</a>  
  </li>  
  <li>  
    <a href="#"  
      @click="clear_main(),hide_account(),show_statistic()">Статистика</a>  
  </li>  
  <li>  
    <a href="#"  
      @click="clear_main(),show_account(),hide_statistic()">Личный кабинет</a>  
  </li>  
</ul>
```

## 4.2 Тестування інформаційної технології

Для дослідження функціональності розробленої інформаційної системи перегляду кіберспортивної статистики було розроблено два тестових випадки(тест-кейси). Для першого випадку перевіряється процес авторизації користувача у сервісі Steam (таблиця 4.1).

Таблиця 4.1 – Тест-кейс T01

Тест-кейс ID: T01	Пріоритет: 1	Створено: 09.06.2022
-------------------	--------------	----------------------

Назва: перевірка коректності авторизації користувача	
Вхідні дані: API ключ	
Кроки	Очікуваний результат
1.Запустити вебсайт 2.Натистути на кнопку авторизації 3.Ввести ім'я користувача та пароль на сторінці сервісу Steam 4.Повернутись на домашню сторінку	Відображення кнопки «Вийти з аккаунту» та аватару користувача на домашній сторінці
Результат виконання: пройдено успішно	

У випадку успішної авторизації користувач отримує відображення свого аватару на сторінці (рис 4.7). Якщо користувач не вводить коректні деталі свого аккаунту, на сторінці Steam буде відображена помилка (рис 4.8).

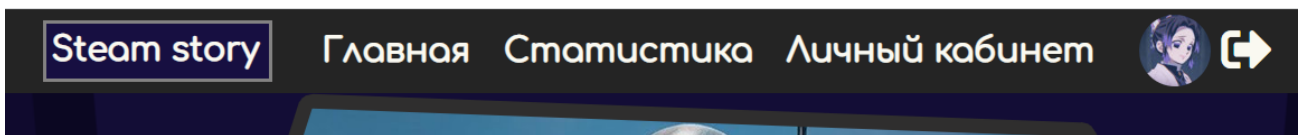
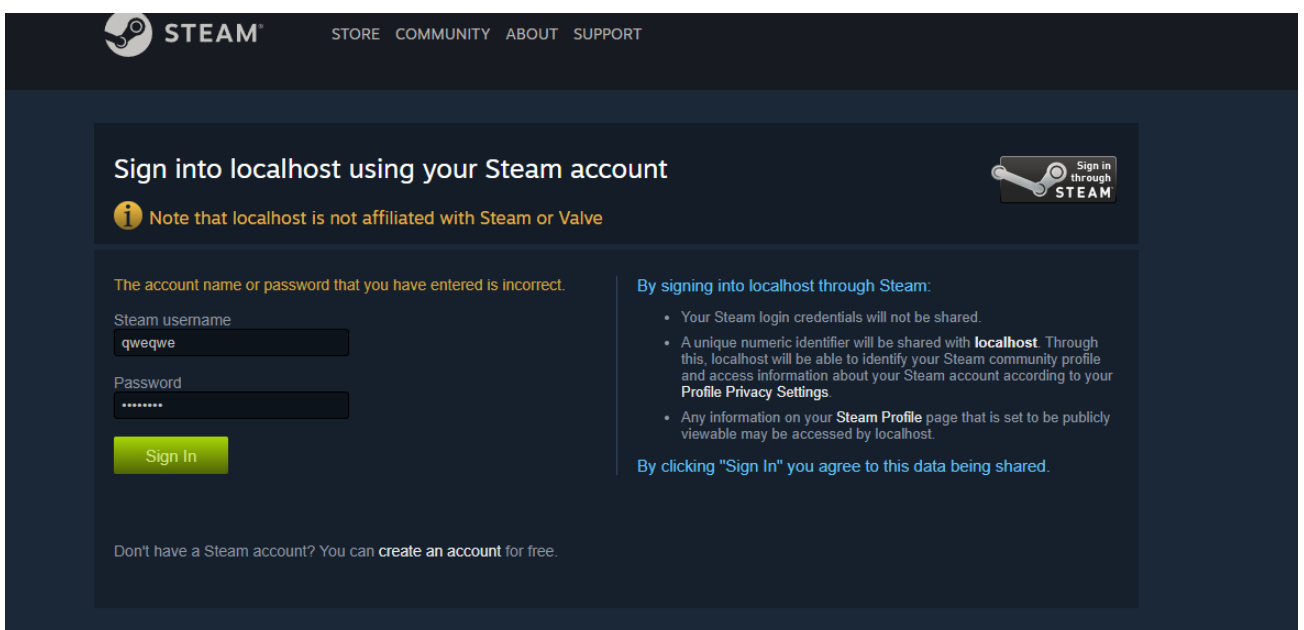


Рисунок 4.7 – Успішна авторизація



### Рисунок 4.8 – Користувач вказав не коректні дані

Другий тестовий випадок перевіряє функціональну можливість відображення статистичних даних (таблиця 4.2).

Таблиця 4.2 – Тест-кейс T02

Тест-кейс ID: T02	Пріоритет: 2	Створено: 10.06.2022
Назва: перевірка коректності відображення статистики користувача		
Вхідні дані: Steam аккаунт користувача		
Кроки	Очікуваний результат	
1.Запустити вебсайт 2.Натиснути на кнопку авторизації 3.Ввести ім'я користувача та пароль на сторінці сервісу Steam 4.Зробити запит до сервісу Steam 5. Відобразити статистичні дані користувача	Відображення статистичних даних користувача на сторінці вебсайту	
Результат виконання: пройдено успішно		

У випадку успішного отримання даних, вони будуть відображені на сторінках вебсайту (рис 4.9). Якщо виникає помилка у запиті до сервісу Steam то виникає помилка (рис 4.10).



Рисунок 4.9 – Успішний запит та відображення даних

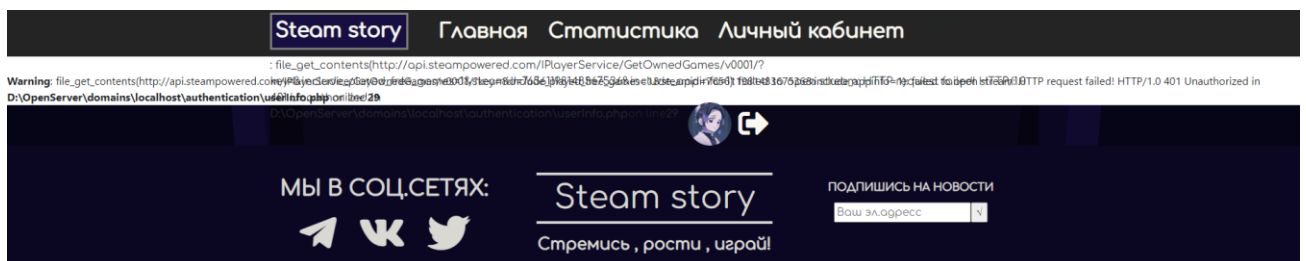


Рисунок 4.10 – Помилка у запиті

#### Висновки до розділу 4

Використовуючи інформаційну та функціональну моделі, було розроблено інформаційну систему. Розроблено візуальний інтерфейс 2022р. Трушевський К.Є. 121 – КРБ.1 – 409.21810226

користувача та об'єднано його з серверною частиною системи. Для забезпечення можливості кастомізації коду, було створено документацію та додано проект на GitHub.

Напрямами практичного використання розробленої інформаційної технології визначено наступні: автоматизоване відображення статистичних даних, автоматизоване зберігання даних у базі.

Застосунок було протестовано на можливість отримання потенційних помилок та усунення них. Було проаналізовано основні функціональні сценарії.

## ВИСНОВКИ

Кваліфікаційна робота бакалавра розв'язує технічну задачу створення інформаційної технології перегляду та аналізу статистичних даних в кіберспорті. В результаті виконання роботи були поставлені та вирішені такі завдання:

- Проведено аналіз ролі кіберспорту у сучасному спорті та визначенно найпопулярніші дисципліни.
- Проведено аналіз існуючих рішень та операючись на переваги та недолікі було сформовано вимоги до системи.
- Розроблено інформаційну та функціональну моделі для перегляду та аналізу кіберспортивних статистичних даних.
- Розроблено інформаційну технологію перегляду та аналізу кіберспортивних статистичних даних з використанням створених моделей.
- Проведено дослідження ефективності інформаційної технології аналізу кіберспортивних статистичних даних.

В результаті проведенного дослідження було виявлено, що розроблена інформаційна технологія коректно працює на всіх пристроях та у різних браузерах. Функціональні можливості системи були також протестовані та всі помилки виправлені.

Напрямами практичного використання розробленої інформаційної технології наступне: автоматизований аналіз даних, перегляд користувачем відформатованих статистичних кіберспортивних даних в обраній дисципліні.

## СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кіберспорт як вид спорту і нова індустрія: проблеми росту. URL: <https://yurgazeta.com/publications/practice/sportivne-pravo/kibersport-yak-vid-sportu-i-novaindustriya-problemi-rostu.html> (дата звернення: 27.04.2022)
2. Що таке кіберспорт та як ця культура розвинена в Україні. URL: <https://www.radiosvoboda.org/a/29189982> (дата звернення: 28.04.2022)
3. Ознаки кіберспорту як спортивної дисципліни. URL: <http://journals.uran.ua/itfcs/article/viewFile/171290/170943> (дата звернення: 28.04.2022)
4. World of Tanks. URL: [https://uk.wikipedia.org/wiki/World\\_of\\_Tanks](https://uk.wikipedia.org/wiki/World_of_Tanks) (дата звернення: 29.04.2022)
5. FIFA (серія відеоігор). URL: <https://uk.wikipedia.org/wiki/FIFA> (дата звернення: 29.04.2022)
6. League of Legends. URL: [https://leagueoflegends.fandom.com/ru/wiki/League\\_of\\_Legend](https://leagueoflegends.fandom.com/ru/wiki/League_of_Legend) (дата звернення: 29.04.2022)
7. Мова програмування PHP. URL: <https://www.php.net/> (дата звернення: 01.05.2022)
8. Swiper слайдер. URL: <https://swiperjs.com/> (дата звернення: 01.05.2022)
9. Середовище програмування Microsoft VisualStudio Code. URL: <https://code.visualstudio.com/> (дата звернення: 01.05.2022)
10. Фреймворк Bootstrap. URL: <https://getbootstrap.com/docs/5.2/getting-started/introduction/> (дата звернення: 02.05.2022)
11. Steam API documentation. URL: <https://steamcommunity.com/dev> (дата звернення: 02.05.2022)

12. Steam API functions. URL: [https://developer.valvesoftware.com/wiki/Steam\\_Web\\_API](https://developer.valvesoftware.com/wiki/Steam_Web_API) (дата звернення: 02.05.2022)
13. Фреймвок VueJS. URL: <https://vuejs.org/guide/introduction.html> (дата звернення: 03.05.2022)
14. HTML tutorials. URL: <https://www.w3schools.com/html/> (дата звернення: 04.05.2022)
15. CSS tutorials. URL: <https://www.w3schools.com/css/> (дата звернення: 04.05.2022)
16. MySQL documentation. URL: <https://dev.mysql.com/doc/> (дата звернення: 06.05.2022)
17. GOSU AI. URL: <https://gosu.ai/> (дата звернення: 07.05.2022)
18. Cod Tracker. URL: <https://cod.tracker.gg/warzone> (дата звернення: 07.05.2022)
19. Game-Stat. URL: <https://game-stat.com/> (дата звернення: 07.05.2022)
20. Dota 2 API. URL: <https://docs.opendota.com/> (дата звернення: 10.05.2022)



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет імені Петра Могили**

**Факультет комп'ютерних наук**

**Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**Вебсервіси клієнтської підтримки онлайн-гравців**

**СПЕЦІАЛЬНА ЧАСТИНА З ОХОРОНИ ПРАЦІ**

Спеціальність «Інженерія програмного забезпечення»

121 – КРБ.1 – 409.21810226

*Студент*

\_\_\_\_\_ **К.Є.Трушевський**  
*підпис*  
«\_\_» \_\_\_\_\_ 2022 р.

*Консультант кан.тех.наук., доцент*

\_\_\_\_\_ **А. О. Алексєєва**  
*підпис*  
«\_\_» \_\_\_\_\_ 2022 р.

**Миколаїв – 2022**

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1. НОРМАТИВНА БАЗА БЕЗПЕКИ РОБОТИ З ПЕРСОНАЛЬНИМ КОМП'ЮТЕРОМ.....	5
2. ВИМОГИ БЕЗПЕКИ ПЕРЕД ПОЧАТКОМ РОБОТИ .....	9
3. ВИМОГИ БЕЗПЕКИ ПІД ЧАС ВИКОНАННЯ РОБОТИ.....	10
4. ВИМОГИ БЕЗПЕКИ В АВАРІЙНІЙ СИТУАЦІЇ.....	11
ВИСНОВКИ.....	14
СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ.....	15

## **ПЕРЕЛІК СКОРОЧЕНЬ**

МОЗ – Міністерство охорони здоров'я

ДСН – державні санітарні норми

ДСанПН – державні санітарні правила та норми

МЮУ – Міністерство юстиції України

## ВСТУП

У зв'язку з розвитком інформаційних технологій кожне підприємство вирішило їх впроваджувати для полегшення та автоматизації роботи за допомогою технічних пристроїв. Як приклад, можна навести використання ноутбуків та персональних комп'ютерів, які мають на меті збір, аналіз та обмін інформацією всередині та за межами підприємств або установ. Використання комп'ютера полегшує процеси на підприємствах.

Взявши до уваги переваги інтеграції інформаційних технологій за допомогою технічних пристроїв, треба ще враховувати також і недоліки їх інтеграції у робочий процес. Головним недоліком є загострення суспільного та власного здоров'я, яке потребує створення та удосконалення існуючих робочих місць, які матимуть більш сприятливі до здоров'я умови робочого місця. Також треба виділити проведення заходів спрямованих на запобігання розвитку негативних наслідків впливу персонального комп'ютера на здоров'я користувача.

Велика кількість проблем зі здоров'ям виникає через використання персонального комп'ютера або ноутбука. Наприклад, проблеми зі спиною, проблеми із зором або синдром зап'ястного каналу.

Вплив на зір дисплея залежить від його положення. Одним із негативних факторів є частота динамічного оновлення дисплею, яка при маленькій частоті оновлення супроводжується мерехтінням, яке має негативні вплив на зір. За низької частоти оновлення дисплею очні м'язи перебувають у великому навантаженні, що може викликати розвинення короткозорості або втрати зору.

## **1. НОРМАТИВНА БАЗА БЕЗПЕКИ РОБОТИ З ПЕРСОНАЛЬНИМ КОМП'ЮТЕРОМ**

Дія інструкції поширюється на всі підрозділи підприємства, де виконують роботи з персональним комп'ютером (ПК).

Інструкція розроблена відповідно до Положення про розробку інструкцій з охорони праці, затвердженого наказом Держнаглядохоронпраці від 29.01.1998 № 9, Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Держнаглядохоронпраці від 26.01.2001 № 11, Правил охорони праці під час експлуатації електронно-обчислювальних машин, затверджених наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 26.03.2010 № 61, Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 № 7, Загальних вимог стосовно забезпечення роботодавцями охорони праці працівників, затверджених наказом Міністерства надзвичайних ситуацій України від 21.01.2012 № 67 (НПАОП 0.00-7.11-12) [1].

За цією інструкцією працівника, який використовує персональний комп'ютер (*далі* – користувач), інструктують перед початком роботи (первинний інструктаж), а потім через кожні 6 місяців (повторний інструктаж). Результати інструктажу заносять до Журналу реєстрації інструктажів з питань охорони праці на робочому місці (у журналі має бути підпис особи, яка інструктує, та користувача).

Користувач зобов'язаний дбати про особисту безпеку і здоров'я, а також про безпеку і здоров'я довколишніх при виконанні будь-яких робіт, а також під час перебування на території підприємства.

До роботи на персональному комп'ютері допускають осіб, які пройшли інструктажі з питань охорони праці та пожежної безпеки.

Користувач зобов'язаний:

- виконувати правила внутрішнього трудового розпорядку;
- не допускати за своє робоче місце сторонніх осіб;
- не виконувати вказівок, які суперечать правилам охорони праці та пожежної безпеки;
- знати правила надання домедичної допомоги;
- знати розташування та вміти користуватись первинними засобами пожежогасіння;
- вміти працювати з ПК.

Основні небезпечні та шкідливі виробничі фактори, що можуть впливати на користувача:

- підвищений рівень статичної електрики;
- нерівномірність розподілу яскравості в полі зору;
- підвищена яскравість світлового зображення;
- ураження електричним струмом;
- напруга зору та уваги;

У приміщеннях із ПК має бути природне і штучне освітлення(таблиця 1).

Таблиця 1 - Норми мікроклімату для приміщень з ВТД ЕОМ та ПЕОМ

Пора року	Категорія робіт	Температура повітря, град. С не більше	Відносна вологість повітря, %	Швидкість руху повітря, м/с
Холодний	легка-1а	22–24	40–60	0,1
	легка-1б	21–23	40–60	0,1
Теплий	легка-1а	23–21	40–60	0,1
	легка-1б	22–24	40–60	0,2

При розміщенні робочих місць необхідно унеможливити пряме засвічування екрана природним освітленням. При природному освітленні слід передбачити наявність сонцезахисних засобів (плівка, жалюзі, штори тощо). Світлові відблиски із клавіатури, екрана та інших частин ПК у напрямку очей користувача неприпустимі.

Основним обладнанням робочого місця є ПК або ноутбук, монітор, клавіатура, маніпулятор, робочий стіл, стілець (крісло).

При розміщенні елементів робочого місця слід враховувати:

- робочу позу користувача;
- простір для розміщення користувача;
- можливість огляду елементів робочого місця;
- можливість огляду простору поза межами робочого місця;
- можливість робити записи, розміщувати на робочому столі документацію та матеріали, які використовує користувач.

Розміщення елементів робочого місця не має заважати рухам та переміщенню для експлуатування ПК. Монітор встановлюють так, щоб відстань від поверхні екрана до очей користувача була 600-700 мм залежно від розміру екрана. Клавіатуру розміщують на робочому або окремому столі на відстані 100-300 мм від краю з боку користувача. Положення клавіатури та кут її нахилу залежить від побажання користувача (як правило, в межах 1-11°). Не допускати хитання клавіатури [7].

Конструкція робочого столу має бути такою, щоб оптимально розмістити на робочій поверхні обладнання, що використовують, з урахуванням кількості, розмірів, конструктивних особливостей і характеру його роботи.

Крісло має забезпечувати підтримування раціональної робочої пози під час виконання основних виробничих операцій та можливість зміни пози. Тип робочого крісла обирають залежно від характеру та тривалості роботи.

Раціональна поза користувача:

- ступні розташовані на підлозі або на підставці для ніг;
- стегна зорієнтовані у горизонтальній площині;
- верхні ділянки рук вертикальні;
- кут ліктьового суглоба у межах 70-90°;
- зап'ястя зігнуті під кутом не більше ніж 20°;
- нахил голови у межах 11-20°, а часті її повороти виключені.

Для забезпечення оптимальної робочої пози користувача необхідно засоби праці, з якими користувач має тривалий або найбільш частий зоровий контакт, розмістити у центрі зони зорового спостереження та моніторного поля;

Забезпечити відстань близько 100 мм між найважливішими засобами праці, з якими користувач працює найчастіше.

ПК встановлювати на рівній твердій поверхні (столі). Не дозволено встановлювати ПК та оргтехніку на хитких підставках чи на похилій поверхні.

ПК не встановлювати впритул до стіни, перегородки тощо. Не допускати загородження вентиляційних отворів ПК сторонніми предметами.

Розетка біля ПК має бути в доступному місці, щоб в аварійних випадках можна було своєчасно його відімкнути. Не рекомендовано використовувати подовжувачі.

Під час переміщення ПК, периферійних пристроїв витягти вилку живлення з розетки.

Не допускати ушкодження чи модифікування шнура живлення. Заборонено ставити важкі речі на шнур живлення, тягнути чи надмірно перегинати його, скручувати та перев'язувати шнур живлення вузлом. ПК під'єднувати до електромережі лише за допомогою справних штепсельних з'єднань та електророзеток заводського виробництва [3].



## **2. ВИМОГИ БЕЗПЕКИ ПЕРЕД ПОЧАТКОМ РОБОТИ**

Оглянути робоче місце і навести на ньому лад; впевнитись, що на ньому немає сторонніх предмети, все обладнання і блоки ПК з'єднані з системним блоком з'єднувальними шнурами.

Перевірити надійність встановлення апаратури на робочому столі. Монітор не має стояти на краю стола. Повернути монітор так, щоб було зручно дивитися на екран – під прямим кутом (а не збоку) і трохи зверху вниз; при цьому екран має бути трохи нахиленим – нижній край ближче до користувача.

Перевірити загальний стан апаратури, справність електропроводки, з'єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана. Вставити вилку в розетку і впевнитися, що вона міцно тримається. Заборонено вставляти і виймати вилку мокрими руками.

Відрегулювати та зафіксувати висоту крісла та зручний для користувача нахил спинки.

Комп'ютерне обладнання повинні підключатися до електромережі лише за допомогою справних штепсельних з'єднань і електророзеток заводського виготовлення. У штепсельних з'єднаннях та електророзетках, крім контактів фазового та нульового робочого провідників, мають бути спеціальні контакти для підключення нульового захисного провідника. Їх конструкція має бути такою, щоб приєднання нульового захисного провідника відбувалося раніше, ніж приєднання фазового та нульового робочого провідників. Порядок роз'єднання при відключенні має бути зворотним. Не допускається підключати комп'ютерну техніку до звичайної двопровідної електромережі, зокрема з використанням перехідних пристроїв. За потреби приєднати до комп'ютера необхідну апаратуру (принтер, сканер тощо). Усі кабелі, що з'єднують системний блок із іншими пристроями, вмикати та вимикати лише при вимкненому комп'ютері. Відрегулювати яскравість свічення, контрастивість монітора [4].

### 3. ВИМОГИ БЕЗПЕКИ ПІД ЧАС ВИКОНАННЯ РОБОТИ

Під час роботи на ПК:

- стійко встановити клавіатуру на робочому столі, не допускаючи її хитання, водночас передбачити можливість її поворотів та переміщень;
- якщо в конструкції клавіатури не передбачено простору для упору долонь, клавіатуру розміщують на відстані не менше 100 мм від краю столу в оптимальній зоні моніторного поля;
- під час роботи на клавіатурі сидіти рівно, не напружуватися;
- щоб зменшити несприятливе навантаження на користувача при роботі з комп'ютерною мишею (вимушена поза, необхідність постійно контролювати якість дій), забезпечити велику вільну поверхню столу для переміщення комп'ютерної миші та зручного упору ліктьового суглоба;

При роботі з ПК заборонено самостійно розбирати та ремонтувати системний блок (корпус ноутбука), монітор, клавіатуру, комп'ютерну мишу тощо, встромляти сторонні предмети до вентиляційних отворів ПК, ноутбука або монітора, ставити на системний блок ПК та периферійні пристрої металеві предмети, ємкості з водою (вази, горщики для квітів, склянки), оскільки через потрапляння води у середину апарата може виникнути пожежа або ураження електрострумом. Тривалість безперервної роботи за ПК не має перевищувати 2 год. Після цього необхідно зробити 11-хвилинну перерву. Якщо виник зоровий дискомфорт або інші неприємні відчуття, необхідно зробити коротку перерву.

Для зниження нервово-емоційного напруження, стомлення зорового аналізатора, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільно під час декількох перерв виконувати комплекс вправ. Зберегти інформацію. Вимкнути ПК, монітор чи ноутбук. Вимкнути стабілізатор, якщо комп'ютер під'єднаний до мережі через нього. Прибрати робоче місце [2].

#### **4. ВИМОГИ БЕЗПЕКИ В АВАРІЙНІЙ СИТУАЦІЇ**

Аварійні та небезпечні ситуації під час виконання роботи на ПК можуть виникнути у разі: короткого замикання, перевантаження блоку живлення системного блоку, перегрівання, пожежі, поломки крісла тощо.

У разі виникнення аварії або ситуації, що може привести до аварії, нещасного випадку, негайно від'єднати ПК від електромережі, повідомити інцидент керівникові.

Не допускати в небезпечну зону сторонніх осіб.

Якщо стався нещасний випадок, зберегти обстановку в робочій зоні та устаткування у такому стані, в якому вони були на момент події (якщо це не загрожує життю і здоров'ю інших працівників і не призведе до більш тяжких наслідків). Поінформувати про подію керівника робіт (іншу відповідальну особу підприємства) та в подальшому керуватися його вказівками. Вжити заходів, щоб запобігти подібним випадкам у подальшому.

У разі виникнення пожежі (ознак горіння), повідомити керівнику та, за потреби, викликати оперативно-рятувальну службу за телефоном 101 або 112 (назвати адресу та місце виникнення пожежі, наявність людей, повідомити своє прізвище) та вжити можливих заходів для евакуювання людей, гасіння (локалізації) пожежі наявними засобами пожежогасіння. Пам'ятати, що гасіння електротехнічних пристроїв, які перебувають під напругою, виконувати лише після їх попереднього від'єднання від електромережі. Гасити за допомогою вуглекислотних або порошкових вогнегасників, а в окремих випадках – сухим піском.

За потреби надати потерпілому домедичну допомогу згідно з інструкцією, що діє на підприємстві. У разі подальшого погіршення самопочуття потерпілого, не припиняючи надання домедичної допомоги, викликати за телефоном 103 швидку медичну допомогу. Виконувати вказівки керівника робіт для ліквідування небезпеки.

Якщо на металевих частинах обладнання виявлено напругу (відчуття струму), заземлюючий провід обірваний, необхідно вимкнути обладнання, негайно доповісти керівникові про несправності електрообладнання і без його вказівки до роботи не приступати. При припиненні подавання електроенергії, вимкнути обладнання.

При появі незвичного звуку, запаху паленого, мимовільного відключення комп'ютера та оргтехніки, негайно припинити роботу і поставити до відома керівника. При виникненні пожежі негайно вимкнути обладнання, знеструмити електромережу за винятком освітлювальної мережі, повідомити про пожежу всім працюючим і приступити до гасіння осередку пожежі наявними засобами пожежогасіння.

При нещасному випадку необхідно, насамперед, звільнити потерпілого від травмуючого фактора, звернутися до медпункту, зберегти, по можливості, місце травмування в тому стані, в якому воно було на момент травмування. При звільненні потерпілого від дії електроструму слідкуйте за тим, щоб самому не опинитися в контакті з токоведучою частиною та під напругою [5-6].

У випадку аварії:

- негайно припинити роботу;
- залишити небезпечну зону і вжити заходів щодо запобігання подальшого розвитку аварії (відключити електроенергію шляхом виключення загального вимикача або пакетного вимикача на електрощиті приміщення, а у випадку пожежі і приточно-витяжну вентиляцію).

Повідомити про те, що сталося, власнику або безпосередньо керівнику ділянки, на якій сталася аварія.

При нещасних випадках надати першу (долікарську) медичну допомогу потерпілому. При ураженні електричним струмом, якщо потерпілий дихає не часто і судорожно або у нього відсутні ознаки життя (не прослуховується дихання, немає пульсу, розширені зіниці очей) зробити штучне дихання «рот-у-

рот» і непрямий масаж серця (при цьому необхідно викликати швидку допомогу). При пораненні накласти стерильну пов'язку. При кровотечі рану закрити стерильною пов'язкою і тримати в такому стані 3-5 хв, якщо кровотеча припинилась, пов'язку закріпити бинтом. При переломах, вивихах і розтягненні забезпечити спокійне і зручне положення пошкоджено: частини тіла, у всіх випадках після надання першої (долікарської) медичної допомоги потерпілого направити в медпункт або викликати лікаря [2].

## ВИСНОВКИ

Під час виконання спеціальної частини з охорони праці було проаналізовано умови праці людини на підприємствах та в установах, а також роботу при користуванні електронними пристроями.

Правила, викладені в нормативно-правових актах, сьогодні дуже актуальні і важливі, оскільки майже на кожному підприємстві є співробітники, які працюють з електронними пристроями. Коли людина працює за персональним комп'ютером, вона сильно втомлюється, страждає від погіршення зору, нервозності та в деяких випадках психічного здоров'я.

Сьогодні існує багато шкідливих впливів на діяльність людини, таких як шум, вібрація та інші фактори, тому кожен працівник та роботодавець повинен слідувати нормативних вимогам охорони праці, що максимально мінімізувати шкідливий вплив цих факторів на кожного працівника.

Під час діяльності людини на підприємствах необхідно дотримуватись вимог і стандартів захисту працівників в організаціях. Дотримання вимог до працівників та власників підприємств зменшить шкідливий вплив на здоров'я людини.

## СПИСОК ДЖЕРЕЛ ПОСИЛАННЯ

1. Про затвердження санітарного та епідемічного благополуччя населення. URL: <https://zakon.rada.gov.ua/laws/show/4004-12#Text> (дата звернення: 30.01.2022).
2. Порядок проведення медичних оглядів працівників певних категорій. URL: <https://zakon.rada.gov.ua/laws/show/z0846-07#Text> (дата звернення: 30.01.2022).
3. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин ДСанПІН 3.3.2.007-98. URL: <https://mozdocs.kiev.ua/view.php?id=2441> (дата звернення: 30.01.2022).
4. Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: <https://www.sop.com.ua/article/718-vimogi-shchodo-bezpeki-ta-zahistu-zdorovya-pratsvnikv-pd-chas-roboti-z-ekrannimi-pristroyami> (дата звернення: 30.01.2022).
5. Закон України за 21 квітня 2018р №108/31960 про затвердження вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями. URL: <https://zakon.rada.gov.ua/laws/show/z0108-18#Text> (дата звернення: 30.01.2022).
6. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. URL: <http://www.ohranatruda.in.ua/pages/1180/> (дата звернення: 30.01.2022).
7. Робота в офісі: основні санітарно-гігієнічні вимоги. URL: <https://te.dsp.gov.ua/roboata-v-ofisi-osnovni-sanitarno-gigiyenichni-vymogy/> (дата звернення: 30.01.2022).