

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики

(інститут)

Факультет інформаційних технологій

(факультет)

Кафедра Програмного забезпечення комп'ютерних систем

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
магістра

(назва освітньо-кваліфікаційного рівня)

студента	<i>Сергієнка Микити Ігоровича</i> (ПІБ)
академічної групи	<i>121М-22-2</i> (шифр)
спеціальності	<i>121 Інженерія Програмного Забезпечення</i> (код і назва спеціальності)
освітньої програми	<i>«121 Інженерія Програмного Забезпечення»</i> (назва освітньої програми)
на тему:	<i>Розробка та дослідження ефективності впровадження веб-додатку адміністратора кіберклуба.</i>

_____ *Сергієнко М. І.*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинг овою	інституційною	
розділів кваліфікаційної роботи				
спеціальний	<i>Проф. Бердник М.Г.</i>			

Рецензент				
-----------	--	--	--	--

Нормоконтролер				
----------------	--	--	--	--

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

Завідувач кафедри
Програмного забезпечення комп'ютерних
систем

(повна назва)

_____ М.О. Алексєєв
(підпис) (прізвище,
ініціали)

« » _____ 20 ____ Року

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності _____ *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

студента _____ *Сергієнка Микити Ігоровича*
(група) (прізвище та ініціали)

Тема кваліфікаційної роботи _____ *Розробка та дослідження ефективності*
провадження веб-додатка адміністратора кіберклуба

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 09.10.2023 р. № 1227-с

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – процес розробки та впровадження веб-додатку для управління кіберклубом.

Предмет досліджень – програмне забезпечення для управління кіберклубом.

Мета НДР – підвищення ефективності управління кіберклубом.

Вихідні дані для проведення роботи – базуються на основних принципах системного аналізу, функціонального аналізу, теорії баз даних. Також були використані методи емпіричного узагальнення на основі даних.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Новизна запропонованих рішень полягає в обґрунтуванні та розробки веб-додатку для управління сучасних комп'ютерних клубів.

Практична цінність полягає в розробці зручного користувацького інтерфейсу для більш ефективної роботи та управління управлінням комп'ютерного клубу.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє побачити та оцінити безпосереднє актуальність та ефективність веб-додатку, а також зробити актуальний аналіз розвитку комп'ютерних клубів.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Аналіз теми та постановка задачі	12.09.2023-30.09.2023
Розробка веб-додатку та вивчення роботи комп'ютерних клубів	01.10.2022-31.10.2023
Дослідження актуальності веб-додатку та аналіз розвитку комп'ютерних клубів	01.11.2023-12.12.2023

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект від результатів роботи очікується позитивним завдяки розробленому веб-додатку процеси пов'язані з управлінням та продажу послуг будуть значно простішими та зручнішими для адміністраторів.

Соціальний ефект від реалізації результатів роботи очікується позитивним, завдяки веб-додатку та дослідженню ринку комп'ютерних клубів, що в свою чергу зробить для людей більш зручною для управління та зручного використання.

Завдання видав

_____ (підпис)

Бердник М. Г.

_____ (прізвище, ініціали)

Завдання прийняв до виконання

_____ (підпис)

Сергієнко М.І

_____ (прізвище, ініціали)

Дата видачі завдання: 12.09.2023 р.

Термін подання кваліфікаційної роботи до ЕК 12.12.2023

РЕФЕРАТ

Пояснювальна записка: 93 стор., 23 рис., 3 додатка, 25 джерел.

Об'єкт дослідження: процес розробки та впровадження веб-додатку для управління кіберклубом.

Предмет дослідження: програмне забезпечення для управління кіберклубом.

Мета магістерської роботи: підвищення ефективності управління кіберклубом.

Методи дослідження: методи дослідження базуються на основних принципах системного аналізу, функціонального аналізу, теорії баз даних. Також були використані методи емпіричного узагальнення на основі даних.

Наукова новизна: полягає в обґрунтуванні та розробки веб-додатку для управління сучасних комп'ютерних клубів.

Прогнози щодо розвитку досліджень: дослідження, описані в кваліфікаційні роботі, можуть стати базисом для впровадження ефективного веб-додатку для управління кіберклубом.

Практична цінність полягає в розробці зручного користувацького інтерфейсу для більш ефективної роботи та управління управлінням комп'ютерного клубу.

Список ключових слів: кіберклуб, веб-додаток, управління, покупка, комп'ютерний клуб.

ABSTRACT

Explanatory note: 93 pages, 23 figures, 3 appendices, 25 sources.

Research object: the process of development and implementation of a web application for managing a cyber club.

Research subject: software for cyber club management.

The purpose of the master's thesis: increasing the efficiency of cyber club management.

Research methods: research methods are based on the basic principles of system analysis, functional analysis, database theory. Empirical data-based generalization methods were also used.

Scientific novelty: consists in the justification and development of a web application for the management of modern computer clubs.

Predictions for the development of research: the research described in the qualification work can become the basis for the implementation of an effective web application for managing a cyber club.

The practical value lies in the development of a convenient user interface for more efficient work and management of the computer club.

List of keywords: cyber club, sales, management, purchase, computer club.

Перелік умовних позначень

HTML –	Hyper Text Markup Language;
JS –	JavaScript;
UI –	User Interface;
JSX –	Javascript XML;
CSS –	Cascading Style Sheets;
SPA –	Single Page Application;
DOM –	Document Object Model;
API –	Application Programming Interface;
SSE –	Server-Sent Events;
TS –	TypeScript;
CSR –	Client Side Rendering;
W3C –	World Wide Web Consortium;

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ	11
1.1. Відомості про сучасні технології для створення веб-додатку	11
1.2. Характеристика та загальні відомості про кіберклуб.....	14
1.3. Кіберклуби та їх особливості	18
1.4. Розвиток кіберклубу в сучасному світі.....	19
1.5. Постановка задачі.....	20
1.6. Висновок	21
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ДОДАТКІВ ДЛЯ КОНТРОЛЮ КОМП'ЮТЕРНИХ КЛУБІВ ТА УПРАВЛІННЯ	23
2.1. Додатки для контролю комп'ютерних клубів та їх особливості...23	
2.2. Аналіз росту та розвитку комп'ютерних клубів	28
2.3. Особливості управління комп'ютерним клубів	32
2.4. Дослідження додатку для управлінням комп'ютерним клубом ...	33
2.5. Висновок	34
РОЗДІЛ 3 РОЗРОБКА УНІВЕРСАЛЬНОГО КРИПТОГAMANЦЯ ТА АНАЛІЗ ЙОГО ФУНКЦІОНАЛУ	36
3.1. Розробка веб-додатка адміністратора кіберклуба	36
3.2. Технології використані в розробці веб-додатку	36
3.3. Кроссбраузерність	46
3.4. Архітектура додатку	47
3.5. Інтерфейс додатку	49
3.6. Висновок	70
ВИСНОВКИ	72
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	75
ДОДАТОК Б ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ	93

ВСТУП

Кіберклубів в сучасному світі є надзвичайно актуальним та важливим в контексті стрімкого розвитку кіберспорту. Кожен день ця індустрія привертає все більше уваги та зацікавленості, виростаючи з нішового хобі в повноцінну галузь електронного спорту. Розмаїття та величезна кількість кіберспортивних заходів свідчать про стрімке зростання популярності цього виду розваг.

Актуальність дослідження полягає, насамперед, в необхідності розуміти, як саме кіберклуби взаємодіють з цим розвитком, як забезпечують інфраструктуру для гравців та організаторів ігор, і як можуть оптимізувати та поліпшувати свою діяльність.

Однією з ключових аспектів актуальності є економічний вимір. Кіберспорт став важливою економічною галуззю з значними інвестиціями, спонсорськими угодами та рекламними можливостями. Дослідження кіберклубів дозволить виявити найбільш ефективні стратегії для залучення інвестицій та створення прибуткових моделей у цьому секторі.

Також важливо розглядати аспекти кіберклубів у контексті соціальних та культурних тенденцій. Кіберспорт вже давно вийшов за межі простої гри та став важливою складовою сучасної культури. Розуміння та вивчення ролі кіберклубів у формуванні цих тенденцій визначить їхнє місце в суспільстві та вплив на молодше покоління.

Аспекти безпеки та кіберзахисту є ще однією актуальною темою у дослідженні кіберклубів. З ростом популярності кіберспорту збільшується й інтерес кіберзлочинців до цього сектору. Розуміння вразливостей та можливостей захисту кіберклубів важливо для забезпечення безпеки гравців, організаторів і технічних інфраструктур.

Загалом, дослідження кіберклубів відкриває нові можливості для розуміння та оптимізації цього важливого елемента кіберспортивного

екосистеми. Знаходження відповідей на актуальні питання стосовно їхньої ролі та впливу допоможе покращити якість кіберспорту та визначити його місце в сучасному світі.

Мета дослідження полягає в розробці та дослідження ефективного веб-додатку для адміністраторів кіберклубом.

Завдання дослідження. Для досягнення поставленої мети в роботі сформульовані та вирішені такі завдання:

1. Визначити існуючі стандарти веб-додатків для кіберклубів;
2. Визначити архітектуру додатку;
3. Розробка зручного інтерфейсу користувача;
4. Провести дослідження ефективності роботи даного додатку.

Об'єкт дослідження: процес розробки та впровадження веб-додатку для управління кіберклубом для адміністраторів.

Предмет дослідження: ефективного веб-додатку для управління кібер клубом для адміністраторів.

Методи дослідження: методи дослідження базуються на основних принципах системного аналізу, функціонального аналізу.

Наукова новизна: Веб-додаток для управління кіберклубом для адміністраторів зручним інтерфейсом.

Практична цінність полягає в тому, що результат дослідження можна використовувати в сфері бізнесу комп'ютерних клубів задля ефективного управління та забезпечення високого рівня управління.

Особистий внесок автора:

1. Наукові результати роботи отримані автором самостійно;
2. Вибір методів досліджень і технологій реалізацій;
3. Імплементация веб-додатку управління кіберклубом;
4. Розробка теоретичної частини роботи, в якій досліджені і систематизовані знання про додатки для управління кіберклубом;
5. Оцінка отриманих результатів.

Структура і обсяг роботи. Робота складається з вступу, чотирьох розділів, висновків, списку використаних джерел і двох додатків. Загальний обсяг роботи становить 93 сторінок, із них 60 сторінка основного тексту, в тому числі 23 рисунків і список використаних джерел із 25 найменувань на 2 сторінках.

РОЗДІЛ 1

ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАННЯ

1.1. Відомості про сучасні технології для створення веб-додатку

React.js, часто просто називаний React, є бібліотекою для розробки інтерфейсів користувача (UI) веб-додатків. React [1] був створений компанією Facebook і використовується для створення динамічних та ефективних веб-додатків з високопродуктивним та зручним інтерфейсом.

Основні особливості React включають:

— компонентний підхід React розбиває інтерфейс на невеликі незалежні компоненти. Кожен компонент може мати свій стан (state) та методи, що дозволяють легко організувати та управляти кодом;

— віртуальний DOM React використовує віртуальний DOM для оптимізації оновлення елементів на сторінці. Замість того, щоб змінювати реальний DOM при кожному оновленні, React створює віртуальний представлення і визначає, які зміни потрібно внести до реального DOM, щоб відобразити новий стан;

— JSX (JavaScript XML) React використовує JSX, який є розширенням синтаксису JavaScript і дозволяє писати код UI, надаючи вигляд XML/HTML. Це полегшує роботу з компонентами та їх відображенням;

— односторінкові додатки React дозволяє легко створювати односторінкові додатки (Single Page Applications), де весь контент завантажується один раз, і зміни сторінок відбуваються без перезавантаження сторінки;

— стан (State) та властивості (Props) React використовує концепції стану і властивостей. Стан представляє змінні, які можуть змінюватися в

процесі виконання програми. Властивості передаються в компоненти і використовуються для передачі даних від батьківського компонента до дочірнього.

React використовується багатьма компаніями та розробниками для створення високопродуктивних та ефективних користувацьких інтерфейсів у веб-додатках.

Next.js - це фреймворк для розробки веб-додатків на основі React. Він надає набір інструментів та конвенцій для створення реактивних, високопродуктивних та SEO-дружніх веб-додатків. Next.js [2] є високорівневим рішенням, яке дозволяє розробникам швидко створювати універсальні (universal) React-додатки, які можна рендерити як на сервері, так і на клієнті.

Деякі ключові особливості Next.js включають:

- універсальний (Universal) рендерінг можливість використовувати як серверний, так і клієнтський рендерінг для забезпечення оптимальної швидкості завантаження та SEO-оптимізації;

- автоматична оптимізація Next.js надає автоматичну оптимізацію для завантаження тільки необхідних ресурсів та побудови ефективних веб-додатків;

- робота з маршрутами легке визначення маршрутів та управління навігацією в додатках;

- підтримка для стилів і компонентів Next.js легко інтегрується з різними інструментами для роботи зі стилями, включаючи CSS, SASS, та інші;

- збірка та оптимізація є вбудовані засоби для збірки та оптимізації коду, що полегшує деплоймент веб-додатків;

- розширюваність можливість використовувати різні плагіни та розширювати функціонал Next.js.

Next.js допомагає вирішити багато завдань, пов'язаних з розробкою веб-додатків, і є популярним вибором серед розробників React для створення універсальних та ефективних додатків.

SignalR - це технологія, розроблена Microsoft, яка дозволяє створювати веб-додатки з підтримкою реального часу. Головна ідея SignalR [3] полягає в тому, щоб дозволити серверу спілкуватися з клієнтами асинхронно. Це означає, що сервер може надсилати повідомлення клієнту, і клієнт може реагувати на ці повідомлення без необхідності активного запитування сервера.

SignalR підтримує кілька транспортних протоколів, таких як WebSockets, Server-Sent Events (SSE), Long Polling і деякі інші, в залежності від доступності та характеристик браузера та сервера.

Основні можливості SignalR включають:

- реальний час забезпечує можливість миттєвого оновлення вмісту на сторінці без перезавантаження;
- двосторонній обмін повідомленнями клієнти та сервер можуть взаємодіяти один з одним, надсилаючи повідомлення в обидва напрямки;
- підтримка різних платформ SignalR може використовуватися як в ASP.NET, так і в незалежних від сервера JavaScript-додатках;
- спрощення розробки дозволяє розробникам зосередитися на логіці додатку, а не на деталях роботи з сеансами та зв'язками.

SignalR широко використовується для створення різноманітних веб-додатків, таких як чати в реальному часі, онлайн-ігри, системи моніторингу та інші додатки, де важливо мати миттєву взаємодію між сервером і клієнтом.

TypeScript - це мова програмування, яка є надмножиною (супермножиною) JavaScript. Розроблена і підтримується компанією Microsoft, TypeScript [4] додає статично типізовані змінні, класи, інтерфейси та інші функції до синтаксису JavaScript. Основна ідея

TypeScript - це полегшити розробку великих та складних веб-додатків, забезпечуючи переваги статичної типізації та інших концепцій, що поліпшують якість коду та його підтримку.

Основні особливості TypeScript:

- статична типізація TypeScript дозволяє вказувати типи для змінних, параметрів функцій, та інших об'єктів в коді. Це дозволяє виявляти та усувати помилки на етапі розробки, а не під час виконання програми;

- класи та інтерфейси TypeScript надає підтримку для об'єктно-орієнтованого програмування за допомогою класів та інтерфейсів;

- екосистема JavaScript TypeScript є розширенням JavaScript, тобто весь синтаксис JavaScript є валідним в коді TypeScript. Ви можете використовувати існуючий JavaScript-код у ваших TypeScript-проектах;

- компіляція код на TypeScript компілюється у вихідний JavaScript-код, який можна виконувати в будь-якому середовищі, що підтримує JavaScript;

- інструменти розробки TypeScript інтегрується з популярними інструментами розробки, такими як Visual Studio, Visual Studio Code, та іншими.

TypeScript є популярним вибором для розробників, які хочуть використовувати потужні можливості статичної типізації та інших сучасних функцій в їхніх проектах, особливо великих та складних веб-додатках.

1.2. Характеристика та загальні відомості про кіберклуб

Термін "кіберклуб" може вказувати на кілька різних речей в контексті інформаційних технологій та комп'ютерних ігор. Ось деякі загальні відомості та характеристики, які можуть бути пов'язані з цим терміном.

Комп'ютерний клуб:

— кіберклуб може бути місцем, де люди збираються для гри в комп'ютерні ігри, обміну досвідом, спільного програмування або просто спілкування про теми, пов'язані з комп'ютерами та технологіями;

— комп'ютерні клуби можуть надавати доступ до високопродуктивного обладнання для гри, організовувати турніри з комп'ютерних ігор, сприяти навчанню програмування або інших технічних навичок, а також створювати сприятливу атмосферу для обміну ідеями та співпраці.

Кіберспорт:

— кіберклуб може також вказувати на команду або групу гравців у кіберспорті;

— кіберклуби у цьому контексті можуть бути організаціями, що підтримують професійні або аматорські команди, надають їм устаткування та інфраструктуру для тренувань і участі в турнірах.

Інформаційна безпека:

— в інших випадках термін "кіберклуб" може вказувати на групу або організацію, що спеціалізується на інформаційній безпеці та кіберзахисті;

— кіберклуби у цьому контексті можуть проводити навчання з кібербезпеки, аудити безпеки, аналізувати загрози та вразливості, а також надавати консультації з питань кіберзахисту.

В кожному з цих випадків кіберклуб може бути спільнотою або організацією, яка об'єднує людей зі спільним інтересом до технологій, комп'ютерів, ігор чи кібербезпеки. Організації цього типу можуть бути важливими для сприяння обміну знань, підтримки і розвитку у відповідних галузях.

Використання кіберклубу має численні переваги для гравців і фанатів комп'ютерних ігор. Ось кілька ключових переваг:

— кіберклуби надають гравцям можливість об'єднуватися в спільноту з людьми, які мають схожі інтереси. Це створює місце для спілкування, обміну досвідом та встановлення нових дружб;

— в багатьох кіберклубах організуються турніри та змагання, які надають гравцям можливість випробувати свої навички та конкурувати в іграх. Це сприяє розвитку навичок і створенню конкурентного середовища;

— деякі кіберклуби можуть мати високоякісне геймерське обладнання, яке можна використовувати членами клубу. Це може включати потужні комп'ютери, геймерські консолі та інші пристрої;

— кіберклуби можуть бути платформою для гравців, які прагнуть розвиватися в професійному геймінгу. Тут можна отримати підтримку, тренуватися, приймати участь у турнірах та розвивати свої геймерські навички;

— деякі кіберклуби сприяють творчості своїх членів, підтримуючи стрімінг, створення відеоконтенту, арти та іншого матеріалу. Це допомагає гравцям поділитися своїми досягненнями та враженнями;

— кіберклуби можуть організовувати навчальні заходи, майстер-класи та інші ініціативи, щоб покращити геймерські та технічні навички своїх членів;

— клуби можуть брати участь в організації та підтримці громадських подій, таких як локальні лан-паті, турніри чи благодійність на користь геймерської спільноти.

Взяти участь у кіберклубі може бути чудовим способом покращити геймерський досвід, знаходячи нових друзів, конкуруючи та ділитися своєю страстю до комп'ютерних ігор.

Хоча кіберклуби мають багато переваг, їх також можуть супроводжувати деякі недоліки або обмеження. Деякі з можливих недоліків використання кіберклубу включають:

— деякі гравці можуть зіткнутися з проблемою обмеженості доступу до кіберклубу через фізичну відстань. Якщо клуб знаходиться далеко від місця проживання, це може ускладнити регулярну участь;

— деякі кіберклуби можуть вимагати вартості членства або внесків для участі. Це може стати бар'єром для тих, хто не може собі це дозволити;

— не всі кіберклуби мають однакові ресурси та устаткування. Деякі можуть бути добре обладнані, тоді як інші можуть мати обмежені можливості, що може вплинути на загальний досвід користувачів;

— у певних кіберклубах, де багато людей використовують одне обладнання, може виникати конфлікт за доступом або затримки викликані перевищенням завантаження мережі чи комп'ютерів.

— гравці, які шукають спокій та приватність, можуть відчутти дефіцит цих аспектів в загальному просторі кіберклубу, де може бути багато людей;

— існує потенційна загроза для безпеки даних та конфіденційності в інтернет-клубах через можливість вірусів, кібератак та інших кіберзагроз;

— деякі кіберклуби можуть мати обмежений час роботи або специфічний графік, що може не влаштовувати всіх гравців, особливо тих, хто працює або навчається за незвичайним графіком;

— залежно від кіберклубу, може виникнути проблема культурних відмінностей або конфліктів між членами спільноти.

Не зважаючи на ці недоліки, багато гравців вважають кіберклуби важливими та приємними місцями для соціалізації та спільної гри.

1.3. Кіберклуби та їх особливості

Термін "кіберклуб" вказує на спеціальний простір, де люди збираються для гри в комп'ютерні ігри, спілкування та інших комп'ютерно-орієнтованих активностей.

Ось деякі загальні особливості кіберклубів:

— комп'ютери та геймінг-обладнання кіберклуби зазвичай обладнані потужними комп'ютерами та ігровою технікою, такою як геймпади, гарнітури, інші аксесуари, що дозволяють гравцям насолоджуватися іграми в повній мірі;

— ігрові турніри та змагання кіберклуби можуть організовувати ігрові турніри та змагання між гравцями. Це може включати як локальні турніри, так і участь у великих глобальних ігрових заходах;

— інтернет-з'єднання кіберклуби забезпечують швидке та стабільне Інтернет-з'єднання для гравців, оскільки для багатьох ігор важливе низьке пінгове значення і швидкість завантаження;

— спільнота та спілкування кіберклуби часто стають місцем для спілкування гравців. Люди можуть обмінюватися досвідом, грати разом у великих мультиплеєрних іграх або просто спостерігати за грою інших;

— трансляції та спостереження за грою деякі кіберклуби можуть надавати можливість транслювати гру чи спостерігати за грою інших на екранах у клубі або через Інтернет;

— навчання та розвиток деякі кіберклуби пропонують курси та тренування для гравців, які бажають покращити свої навички у конкретних іграх або галузях;

— кав'ярні та зона відпочинку багато кіберклубів мають кав'ярні або зони відпочинку, де гравці можуть відпочити, випити каву чи перекусити між ігровими сесіями;

— використання VR-технологій деякі кіберклуби можуть бути обладнані віртуальною реальністю (VR) для гри в ігри, які підтримують цю технологію.

Важливо відзначити, що конкретні особливості кіберклубів можуть варіюватися в залежності від їхньої локації, розміру та спеціалізації. Вони можуть бути як комерційними підприємствами, так і ініціативами громадських організацій або ентузіастів.

1.4. Розвиток кіберклубу в сучасному світі

Розвиток кіберклубів у сучасному світі визначається багатьма факторами, пов'язаними із зростанням популярності кіберспорту та онлайн-геймінгу загалом [5].

Ось кілька ключових аспектів цього розвитку:

— збільшення популярності кіберспорту за останні роки кіберспорт став серйозною галуззю розваг і залучає велику увагу гравців та глядачів. Це призвело до збільшення попиту на кіберклуби як місця для тренувань, підготовки та участі в турнірах;

— професійний рівень кіберспорту з'явлення професійних кіберспортивних ліг і турнірів, які привертають увагу великих спонсорів та інвесторів, підтримує розвиток кіберклубів як платформ для тренувань професійних гравців;

— глобальні турніри та змагання участь у глобальних турнірах і лігах надає кіберклубам можливість знати своїх суперників, вдосконалювати стратегії та підвищувати рівень гри;

— електронні спортивні ігри з'явлення нових електронних ігор та жанрів створює різноманітні можливості для кіберклубів і гравців;

— технологічні зміни постійне вдосконалення технологій, які використовуються в ігровій індустрії, дозволяє кіберклубам надавати гравцям доступ до останніх ігрових платформ та обладнання;

— співпраця із спонсорами та брендами залучення спонсорів та партнерів дозволяє кіберклубам отримувати фінансову підтримку для розвитку та підтримки своїх команд;

— освітні та тренінгові програми деякі кіберклуби ведуть освітні та тренінгові програми для гравців, що допомагають підготувати нове покоління ігрових талантів;

— спільнота та спілкування кіберклуби стають не лише місцем для гри, але й соціальною платформою, де гравці можуть спілкуватися, ділитися досвідом і взаємодіяти;

— інтеграція з онлайн-платформами велика кількість ігор та турнірів відбуваються в онлайн-режимі, що розширює можливості кіберклубів для залучення гравців з різних частин світу.

Разом з цим, важливо відзначити, що розвиток кіберклубів визначається індивідуальними стратегіями та підходами кожного клубу, його лідерства та спільноти.

1.5. Постановка задачі

Для розробки та дослідження ефективності веб-додатка адміністратора кіберклуба необхідно:

1. Визначити основні функціональні та нефункціональні вимоги до веб-додатка адміністратора кіберклуба.

2. Вивчити потреби адміністраторів управління кіберклубом та визначити їх очікування від веб-додатка.

3. Розробити структуру та архітектуру веб-додатка, враховуючи вимоги адміністраторів та забезпечуючи зручний та ефективний інтерфейс користувача.

4. Визначити технологічний стек для розробки веб-додатка, включаючи мову програмування, фреймворки та інші технічні компоненти;

5. Розробити основні модулі та функції веб-додатка адміністратора, включаючи управління хостами, гравцями, ігровими заходами та інші аспекти кіберклуба.

6. Забезпечити безпеку та конфіденційність інформації, що обробляється веб-додатком.

7. Здійснити моніторинг роботи веб-додатка під час його реального використання адміністраторами.

8. Зібрати зворотний зв'язок від адміністраторів щодо зручності використання та корисності функціоналу.

9. Оцінити вплив веб-додатка на ефективність управління кіберклубом.

10. Зрозуміти, як новий інструмент впливає на швидкість реакції адміністраторів та загальну продуктивність клубу.

Це комплекс завдань спрямований на створення високоефективного та зручного інструменту для адміністрування кіберклубу, а також на його подальше вдосконалення.

1.6. Висновок

В першому розділі роботи було вивчено сучасні технології, які застосовуються для створення веб-додатків. Аналіз вказав на те, що на сьогоднішній день існують різноманітні інструменти та підходи, які дозволяють розробникам ефективно створювати функціональні та інноваційні веб-додатки.

Далі в розділі була надана характеристика кіберклубів, яка включала загальні відомості про їхню природу та функціональність. Особливий акцент був зроблений на важливості кіберклубів як платформи для обміну знаннями, взаємодії та спільної роботи в області інформаційної безпеки.

Розділ також розглянув різні типи кіберклубів та їх особливості. Аналіз показав, що кожен кіберклуб має свої унікальні риси, що визначають його специфіку та завдання.

Одним із важливих аспектів розділу був огляд розвитку кіберклубів у сучасному світі. Висвітлено фактори, які впливають на їхню динаміку, такі як технологічний прогрес, зміни в кіберзагрозах та підходи до кібербезпеки. Це вказало на постійну потребу у вдосконаленні та адаптації кіберклубів до змін у сучасному інформаційному середовищі.

У завершенні розділу була поставлена задача дослідження, яка визначає основні напрямки подальших досліджень. Визначено ключові аспекти, що будуть розглядатися в наступних розділах для отримання глибшого розуміння феномену кіберклубів та їхнього впливу на сучасні технології та кібербезпеку.

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ДОДАТКІВ ДЛЯ КОНТРОЛЮ КОМП'ЮТЕРНИХ КЛУБІВ ТА УПРАВЛІННЯ

2.1. Додатки для контролю комп'ютерних клубів та їх особливості

В сучасному світі, де технології невпинно вдосконалюються, комп'ютерні клуби стають не лише простором для розваг і відпочинку, але і важливим елементом управління та контролю. Щоб забезпечити ефективну роботу і надати користувачам високоякісний сервіс, важливо використовувати сучасні додатки для контролю та оптимізації роботи комп'ютерних клубів. Ці інноваційні рішення не лише спрощують управління, а й забезпечують безпеку, ефективність та зручність використання, відкриваючи нові можливості для розвитку індустрії розваг та інтерактивного спілкування. Давайте розглянемо, як сучасні додатки можуть перетворити комп'ютерні клуби у центри сучасної інтерактивної культури та розваг.

Детально розглянемо кілька програм для управління комп'ютерними клубами.

Smartlaunch:

- управління доступом керування часом, коли користувачі можуть використовувати комп'ютери;
- контроль витрат відстеження використання ресурсів, зокрема часу, друку і інтернет-трафіку;
- моніторинг користувачів відображення інформації про активність ігор і використання ресурсів;
- статистика ігор збір статистики щодо використання ігор у комп'ютерному клубі;

— оплата послуг можливість налаштування платіжних систем і контроль оплати за послуги.

Antamedia Internet Cafe:

— керування доступом встановлення обмежень на робочий час і контроль доступу;

— фільтрація контенту блокування небажаного веб-контенту для користувачів;

— звітність збір інформації про використання комп'ютерів і інтернету;

— облік використання інтернету відстеження часу, витраченого на веб-сайти;

— підтримка бездротового інтернету можливість обслуговування клієнтів з підтримкою Wi-Fi.

NandyCafe:

— моніторинг клієнтів слідування за активністю користувачів і перегляд їхніх дій;

— облік інтернет-часу збір інформації про використання інтернет-ресурсів;

— відстеження витрати паперу контроль над друкуванням для обмеження витрат;

— захист від несанкціонованого доступу забезпечення безпеки даних і захист від несанкціонованого використання.

CyberCafePro:

— фільтрація веб-контенту блокування небажаного веб-контенту і захист від загроз;

— звітність створення звітів щодо використання комп'ютерів і інтернету;

— управління годинами роботи керування часом роботи і доступу для користувачів;

— моніторинг користувачів слідування за активністю користувачів і їхніми діями.

Ці програми дозволяють власникам комп'ютерних клубів ефективно управляти ресурсами, контролювати доступ користувачів і забезпечувати високий рівень безпеки та зручності для клієнтів. Перед вибором програми важливо врахувати конкретні потреби і вимоги вашого клубу, а також перевірити актуальні відгуки користувачів і технічну підтримку.

Smartlaunch - це високоефективне програмне забезпечення, яке визначається своєю спрямованістю на управління комп'ютерними клубами та ігровими центрами. Ця інноваційна платформа не просто полегшує організацію робочих процесів, але й надає власникам і операторам клубів потужний інструментарій для забезпечення безперебійного функціонування ігрових зон.

Smartlaunch [6] дозволяє ефективно керувати доступом, встановлювати часові обмеження та моніторити активність користувачів. Забезпечуючи контроль за ресурсами, програма дозволяє оптимізувати використання часу, інтернет-ресурсів та інших послуг.

Однією з ключових переваг Smartlaunch є можливість аналізу статистики ігор, що дозволяє власникам клубів зрозуміти уподобання своєї аудиторії та пристосувати асортимент ігор відповідно. Це не лише сприяє задоволенню клієнтів, але й підвищує прибутковість бізнесу.

Захист конфіденційності і безпека даних - важливі аспекти Smartlaunch, що забезпечують впевненість користувачів у їхній особистій та фінансовій інформації.

Завдяки регулярним оновленням і високоякісній технічній підтримці, Smartlaunch залишається невід'ємною частиною сучасних комп'ютерних клубів, де ефективність та зручність використання є відзначеними показниками.

Antamedia Internet Cafe - це потужне програмне забезпечення, спеціально створене для ефективного управління інтернет-кафе та геймінг-центрами. Завдяки своїй високотехнологічній інфраструктурі, Antamedia надає власникам бізнесу інструменти для оптимізації процесів та забезпечення найвищого рівня обслуговування клієнтів.

Завдяки Antamedia Internet Café [7] можна легко налаштовувати та контролювати доступ користувачів, встановлювати гнучкі часові обмеження та вести моніторинг їхньої активності. Функціонал фільтрації веб-контенту дозволяє власникам контролювати вміст, що переглядається користувачами, і забезпечувати безпечне середовище.

Antamedia також забезпечує облік використання інтернет-ресурсів, включаючи час та обсяги трафіку, що є важливим для оптимізації бізнес-процесів. Зручні інструменти створення звітів дозволяють власникам аналізувати роботу свого закладу та вчасно приймати стратегічні рішення.

Однією з особливостей Antamedia Internet Cafe є можливість підключення до бездротової мережі, що підвищує зручність обслуговування клієнтів.

Безпека та конфіденційність даних - це важливі аспекти роботи Antamedia, що забезпечують високий рівень довіри користувачів до цього програмного забезпечення.

Antamedia Internet Cafe - це не просто інструмент для управління, а справжній партнер для власників інтернет-кафе, які прагнуть надати своїм клієнтам найкращий досвід користування та забезпечити успішне функціонування свого бізнесу.

NandyCafe — це інноваційне програмне забезпечення, розроблене спеціально для управління інтернет-кафе та геймінг-центрами. Його виокремлює висока ефективність і багатофункціональність, завдяки чому власники бізнесу можуть з легкістю керувати та оптимізувати роботу свого закладу.

Однією з ключових особливостей HandyCafe [8] є зручний інтерфейс, який робить його використання простим та доступним навіть для тих, хто не має глибоких технічних знань. Програма дозволяє легко налаштовувати та контролювати доступ користувачів, встановлювати гнучкі години роботи та вести моніторинг їхньої активності.

Завдяки HandyCafe, власники інтернет-кафе можуть ефективно контролювати облік використання інтернет-ресурсів, моніторити час, проведений користувачами в закладі, та забезпечувати безпеку у вигляді фільтрації веб-контенту.

Однак HandyCafe не лише інструмент для контролю, але й засіб для забезпечення зручності користувачів. Програма підтримує роботу з різними платіжними системами та забезпечує надійний захист від несанкціонованого доступу.

Висока швидкість реакції та система підтримки дають впевненість в тому, що HandyCafe — це не просто програмне забезпечення, а невід'ємна частина успішного функціонування інтернет-кафе та геймінг-центрів. HandyCafe — це інструмент, який допомагає бізнесу рости та надає неперевершений досвід користувачам.

CyberCafePro - це потужне і надійне програмне забезпечення, створене для управління інтернет-кафе та ігровими центрами. Його надзвичайна функціональність та простота використання роблять його вибором для власників бізнесу, які прагнуть забезпечити ефективне управління та високий рівень обслуговування клієнтів.

Основні переваги CyberCafePro [9] включають можливість зручного управління годинами роботи та контролю доступу користувачів. Програма дозволяє налаштовувати часові обмеження і контролювати доступ до комп'ютерів, щоб забезпечити оптимальне використання ресурсів.

Завдяки фільтрації веб-контенту, CyberCafePro допомагає власникам клубів забезпечити безпеку та зручність для своїх користувачів, блокуючи

небажаний вміст та створюючи безпечне середовище для перегляду веб-сторінок.

Програмне забезпечення також забезпечує ведення докладних звітів щодо використання комп'ютерів та інтернет-ресурсів. Це допомагає власникам бізнесу аналізувати та оптимізувати роботу свого інтернет-кафе.

CyberCafePro володіє інтуїтивним інтерфейсом та легко інтегрується з різними платіжними системами, що робить його відмінним вибором для бізнесу в сфері розваг та геймінгу. Завдяки своїй надійності та високій продуктивності, CyberCafePro є справжнім союзником для успішного ведення інтернет-кафе та ігрових центрів.

2.2. Аналіз росту та розвитку комп'ютерних клубів

З появою та поширенням персональних комп'ютерів і Інтернету комп'ютерні клуби перейшли від місць для гри до справжніх центрів інтерактивного спілкування та розваг. Розуміння та вивчення факторів, що впливають на їх ріст і розвиток, стає надзвичайно важливим для бізнесу в цій галузі.

Одним із ключових факторів є технологічний прогрес. Зростання потужності обчислювальної техніки, розвиток графічних технологій та введення в гру віртуальної реальності розширюють можливості комп'ютерних клубів у створенні захоплюючих ігрових вражень.

Перехід до епохи електронних спортивних змагань (esports) робить комп'ютерні клуби центрами для тренувань та конкурентних ігор. Споживачі більше цінують соціальну взаємодію та спільноти, що вимагає від клубів створювати привабливе співтовариство для гравців.

Існуюча потреба в стабільному та швидкому Інтернет-з'єднанні стає критичною для онлайн-ігор та стрімінгових послуг. Клуби, що

використовують хмарні технології, можуть надавати нові рівні доступу до ігор та інших інтерактивних забав.

У світі, переповненому конкуренцією, ефективний маркетинг та брендування стають визначальними. Клуби повинні створити унікальний образ та пропонувати привабливі акції, а також сприяти розвитку ідентичності та спільноти серед своїх клієнтів.

Аналізуючи ріст та розвиток комп'ютерних клубів (рис 2.1.), ми бачимо, що ключовими чинниками є технологічний прогрес, врахування змін в споживчому попиті, використання широкосмугового інтернету, та ефективне маркетингове стратегію. Успішні клуби активно впроваджують інновації та надають своїм клієнтам не лише гру, але і цілковитий ігровий досвід.

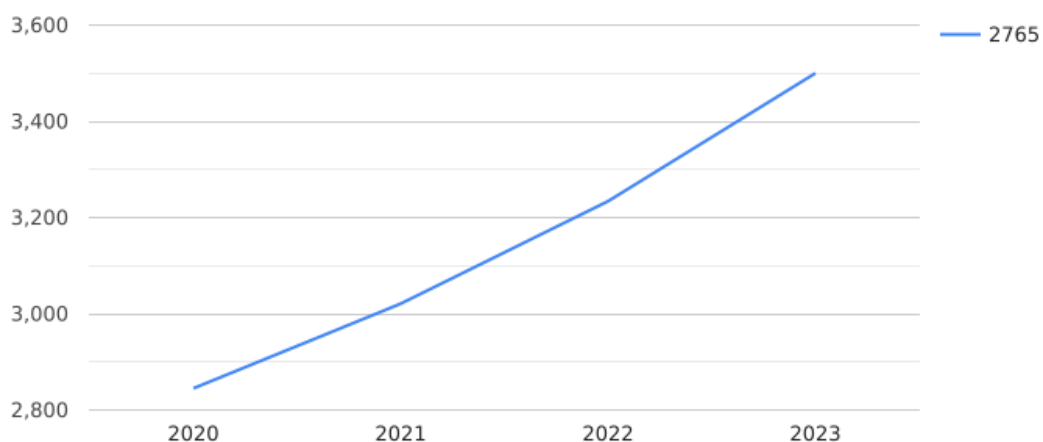


Рис. 2.1. Зростання комп'ютерних клубів в Україні за останнім часом

В останні роки спостерігається помітний ріст інтересу до комп'ютерних клубів, який може бути пояснений декількома факторами. Цей ріст свідчить про зміну у сприйнятті геймінгу та комп'ютерної культури, а також про розширення функціоналу клубів для задоволення різноманітних потреб гравців.

Одним із ключових факторів, що сприяють зростанню відвідування комп'ютерних клубів, є прагнення людей до соціальної взаємодії та входження в спільноту. Клуби стали не лише місцями для гри, але і центрами, де гравці можуть знайти однодумців, поділитися враженнями та приєднатися до електронних спортивних команд.

Розвиток електронних спортивних змагань (esports) став значущим стимулом для відвідування клубів. Багато клубів організують або підтримують турніри з різних ігор, привертаючи гравців, які бажають взяти участь в конкуренції та покращити свої навички.

Комп'ютерні клуби також надають доступ до високоякісного геймінг-обладнання, яке може бути вартим значної інвестиції для окремих гравців. Це стає привабливим фактором для тих, хто хоче насолоджуватися високоякісним геймінгом без необхідності купувати дороге обладнання.

Багато клубів розширюють свою функціональність, надаючи різноманітні розважальні події. Це може включати тематичні ігрові вечори, лекції, відгуки на нові ігри та інші заходи, які роблять відвідування клубів захоплюючим та різноманітним дозвіллям.

Загалом, інтерес до геймінгу у світі зростає. Графіка, сюжети та механіка ігор стають все більше захоплюючими, привертаючи нових гравців та спонукаючи вже існуючих до більш активної участі в геймінговій культурі.

Усі ці фактори взаємодіють, створюючи позитивний тренд у зростанні відвідування комп'ютерних клубів і підкреслюючи їхню роль у формуванні спільнот та сприянні соціальній взаємодії серед гравців.

За статистичними даними, більше третини аудиторії (32.7%) належать до вікової категорії від 18 до 24 років. Це період активного пошуку себе, вибору кар'єри та насолоди свободою. Молодь в цьому віці

знаходить у комп'ютерних клубах не лише місце для гри, але й платформу для соціальної взаємодії та створення нових дружб.

Вікова група (рис 2.2) від 25 до 34 років (28.5%) також є значущою частиною аудиторії. Це часто молоді професіонали, які використовують клуби для відпочинку після робочого дня та участі в ігрових турнірах. Деякі з них, можливо, знаходять в комп'ютерних іграх вихід від реальності та можливість виявити свої лідерські та стратегічні навички.

Інша частка аудиторії (17.6%) – це молодь віком менше 18 років. Деякі з них можуть бути новачками в геймінгу, шукаючи в клубі місце для розвитку своїх ігрових талантів та соціальної інтеграції.

Останній сегмент аудиторії (21.2%) належить до різних категорій. Це можуть бути люди старше 34 років, які також відчувають привабливість геймінгу та шукають в клубах можливість відпочинку чи участі в ігрових івентах.

Таким чином, аудиторія комп'ютерних клубів формує динамічне та різноманітне співтовариство, де кожен може знайти щось для себе – чи то нових друзів, конкурентів в іграх або просто місце для цікавого часу.

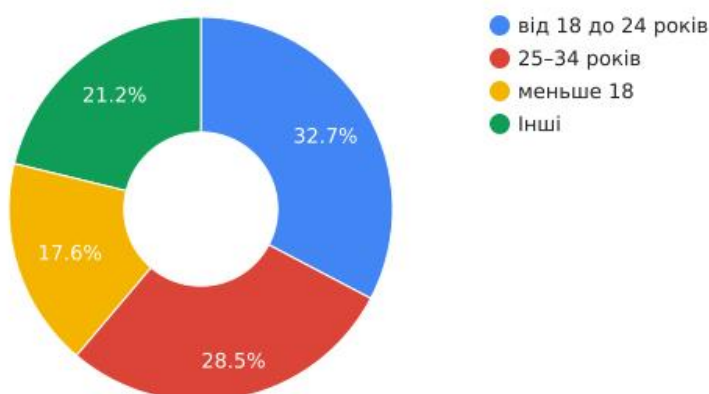


Рис. 2.2. Відсоткове аудиторія комп'ютерних клубів

2.3. Особливості управління комп'ютерним клубів

Управління комп'ютерним клубом – це виклик, який вимагає від організаторів не лише технічної експертизи, але і розвинених навичок управління бізнесом та взаємодії з різними аспектами геймінгової спільноти [10].

Ось деякі особливості управління комп'ютерним клубом:

— технічне забезпечення та інфраструктура впевнене та сучасне технічне забезпечення є ключовим аспектом успішного управління комп'ютерним клубом. Гравці шукають клуби, де є потужні геймінг-ПК, високоякісні монітори та швидкий Інтернет. Також важливо мати зручне місце для гравців і атмосферу, яка сприяє поглибленому зануренню в геймінговий світ;

— організація турнірів та подій управління комп'ютерним клубом часто пов'язане з організацією геймінгових турнірів та івентів. Клуби, які активно проводять або беруть участь у турнірах, залучають більше гравців та стають центром уваги в геймінговій спільноті;

— командна робота та персонал ефективно управління клубом передбачає наявність досвідченого та пристосованого персоналу. Важливо, щоб команда була здатна вирішувати технічні питання, організувати події, надавати клієнтам допомогу та забезпечувати безперебійну роботу клубу;

— співпраця з геймерськими спільнотами управління клубом включає у себе роботу з геймерськими спільнотами та партнерами. Збудувати партнерські стосунки з виробниками геймінгового обладнання, розробниками ігор та іншими гравцями геймінгової індустрії може забезпечити клубу доступ до новітніх технологій та подій;

— маркетинг та спілкування з гравцями ефективний маркетинг – ще один важливий компонент. Залучення нових гравців та збереження

існуючої аудиторії вимагає рекламних кампаній, акцій, розробки інтерактивного контенту та активної участі в соціальних мережах.

Управління комп'ютерним клубом – це баланс між геймінговими враженнями та бізнес-аспектами, де кожен елемент відіграє важливу роль у створенні привабливого середовища для гравців та розвитку геймінгової спільноти.

2.4. Дослідження додатку для управління комп'ютерним клубом

В сучасному світі, де геймінг та електронні розваги займають велику частину культурного простору, управління комп'ютерним клубом стає важливим аспектом. Щоб спростити та оптимізувати цей процес, виникають різноманітні додатки для управління комп'ютерними клубами, спеціально розроблені для задоволення потреб геймерських спільнот та бізнесу.

Одним з таких інструментів є "Гейм-Менеджер". Цей додаток розроблений для ефективного управління всіма аспектами комп'ютерного клубу, починаючи від технічного обладнання і закінчуючи організацією ігрових турнірів та подій.

Дослідження функцій додатку:

— управління технічним забезпеченням додаток дозволяє моніторити та керувати технічним обладнанням в реальному часі. Оптимізована робота геймінгових ПК, перевірка наявності оновлень та вирішення технічних проблем – все це робиться за допомогою інтуїтивно зрозумілого інтерфейсу;

— організація турнірів та ліг можливість створювати та керувати геймінговими турнірами – це ключова функція. Від керування реєстрацією

учасників до автоматизованої підготовки розкладу турнірів – додаток спрощує весь процес організації ігрових заходів;

— керування фінансами та акціями вбудована система фінансового обліку дозволяє відстежувати прибутки, витрати та прибутковість клубу. Також, додаток дозволяє організовувати різноманітні акції та програми лояльності для постійних клієнтів;

— аналітика та звітність важливий компонент – це аналітичні інструменти для вивчення ефективності клубу. Збір та аналіз даних про відвідування, участь у турнірах та інші параметри допомагають у прийнятті обґрунтованих управлінських рішень;

— взаємодія з гравцями можливість спілкування та отримання фідбеку від гравців – це ще одна важлива функція. За допомогою спеціальних чатів та форумів, гравці можуть обговорювати нові ігри, події та висловлювати свої побажання.

Дослідження додатку для управління комп'ютерним клубом підкреслює важливість інновацій та технологій у розвитку геймінгової індустрії. Гнучкість, надійність та зручність використання дозволяють таким додаткам стати необхідними інструментами для успішного управління клубами та забезпечення найвищого рівня задоволення гравців.

2.5. Висновок

У другому розділі роботи було проведено глибокий аналіз додатків, спрямованих на контроль та управління комп'ютерними клубами, виокремлено їхні унікальні особливості та функціонал. Це дозволило отримати чітке уявлення про те, які можливості надають такі програмні засоби для ефективного керування клубними структурами.

Одним із ключових аспектів дослідження став аналіз росту та розвитку комп'ютерних клубів. Поглиблене вивчення цього питання

виявило, що динаміка розвитку клубних установ визначається не лише збільшенням їхньої кількості, але й розширенням функціональних можливостей, що відповідає сучасним технологічним вимогам та потребам користувачів.

Особлива увага була приділена аналізу особливостей управління комп'ютерними клубами. Визначено, що успішне управління вимагає не лише технічних знань, але й уміння ефективно взаємодіяти з користувачами, забезпечуючи їм якісний сервіс та безпеку.

Окремий блок досліджень був присвячений вивченню конкретного додатку для управління комп'ютерним клубом. Аналіз його функціоналу та впливу на роботу клубу надав можливість зрозуміти, як цей інструмент може впливати на ефективність управління та сприяти покращенню роботи клубу в цілому.

Загалом, другий розділ роботи сприяв глибокому розумінню сучасних тенденцій у сфері управління комп'ютерними клубами, а також визначенню перспектив для подальших досліджень та вдосконалення існуючих підходів до цього питання.

РОЗДІЛ 3

РОЗРОБКА УНІВЕРСАЛЬНОГО КРИПТОГАМАНЦЯ ТА АНАЛІЗ ЙОГО ФУНКЦІОНАЛУ

3.1. Розробка веб-додатка адміністратора кіберклуба

Додаток має надавати користувачу наступні можливості:

- вхід у систему;
- створення віртуальної карти кіберклуба;
- список тарифів та створення;
- вивід логів;
- створення та оновлення користувача (CRUD);
- відображення заказів;
- каталог (CRUD);
- відображення сесій та зміни в режимі реального часу.

3.2 Технології використані в розробці веб-додатку

React.js (рис 3.1.) - це бібліотека JavaScript, яка використовується для розробки інтерактивних і високоефективних користувацьких інтерфейсів. Створений компанією Facebook, React дозволяє розробникам побудувати складні веб-додатки з великою кількістю компонентів, які взаємодіють один з одним.

Магія React [1, 11] полягає в тому, що він використовує концепцію віртуального DOM (Document Object Model). При зміні даних React не перерендерює весь документ, а тільки ті частини, які фактично змінилися. Це робить додатки на React високоефективними та швидкодіючими.

Основною одиницею React є компонент. Компонент - це невеликий ізольований елемент інтерфейсу, який може мати власний стан (state) та

отримувати дані від інших компонентів через властивості (props). Стан дозволяє вам зберігати і відстежувати зміни даних, а властивості передають дані від батьківських компонентів до дочірніх.

React також використовує JSX (JavaScript XML), розширення JavaScript, що дозволяє описувати структуру інтерфейсу, схожу на XML або HTML, без виклику методів для створення DOM-елементів. JSX робить код React більш зрозумілим та лаконічним.

Однією з сильних сторін React є його активна спільнота та екосистема. Багато бібліотек та інструментів (наприклад, Redux для керування станом, React Router для навігації) створені спеціально для розширення можливостей React та полегшення розробки.

Використання React важливо не тільки для створення веб-додатків, але і для розвитку мобільних додатків за допомогою React Native. Це дозволяє розробникам використовувати знання React для створення нативних додатків для платформ Android і iOS.

Загалом, React є потужним інструментом для створення сучасних, ефективних та швидких веб-додатків, який продовжує привертати увагу та підтримку розробників у всьому світі.



Рис 3.1. Reactjs

Next.js (рис. 3.2.) - це фреймворк для розробки веб-додатків на основі React.js. Розроблений компанією Vercel, Next.js надає розробникам зручний і потужний інструментарій для створення високопродуктивних та масштабованих додатків.

Однією з ключових особливостей Next.js [2, 12] є можливість рендерингу на стороні сервера (Server-Side Rendering - SSR) та на стороні клієнта (Client-Side Rendering - CSR). Це дозволяє оптимізувати завантаження сторінок, забезпечуючи користувачам швидший ініціальний доступ до контенту. Крім того, Next.js підтримує статичне генерування (Static Site Generation - SSG), що сприяє ще більшій швидкодії та ефективності додатків.

Next.js також надає вбудовану підтримку маршрутизації, що полегшує організацію сторінок та навігацію в додатку. Компоненти, які розробляються за допомогою Next.js, можуть автоматично завантажувати необхідні дані перед рендерингом, що робить їх більш динамічними та гнучкими.

Next.js включає також низку інших функціональностей, таких як вбудована підтримка CSS-in-JS (наприклад, Styled Components), автоматичне код-розделення (Code Splitting), інтеграція з серверним API та обробка зображень на стороні сервера.

Фреймворк ідеально підходить для розробки як невеликих лендінгів, так і великих веб-додатків. Його популярність визначається не лише зручністю використання та потужністю функцій, але й великою спільнотою розробників та активністю в удосконаленні фреймворку.

Next.js інтегрується з платформою Vercel, що дозволяє легко розгортати та управляти веб-додатками. Завдяки цьому взаємодії, розробники можуть швидко публікувати свої проекти та зосереджуватися на їхньому вдосконаленні, користуючись всіма перевагами Next.js.



Рис 3.2. Nextjs

TypeScript (рис. 3.3.) - це мова програмування, яка представляє сучасний інтерфейс для JavaScript, додавши статичну типізацію та інші функції, які полегшують розробку великих та складних проєктів. Розроблений Microsoft, TypeScript стає все більш популярним інструментом серед програмістів.

Однією з ключових особливостей TypeScript [4, 13] є можливість визначення типів для змінних, параметрів функцій, об'єктів та інших елементів коду. Це дозволяє виявляти помилки на етапі розробки і полегшує рефакторинг коду. Статична типізація допомагає покращити надійність програми та робить її більш зрозумілою для розробників, що сприяє зниженню кількості помилок та полегшує спільну роботу великих команд.

TypeScript є сумісним із стандартним JavaScript, що дозволяє поступово переходити від однієї мови до іншої. Розробники можуть починати з додавання типів до окремих файлів у вже існуючому проєкті, поступово розширюючи цей підхід на весь код.

Ще однією важливою рисою TypeScript є підтримка сучасних функцій JavaScript та технологій. TypeScript інтегрується з популярними

фреймворками, такими як React, Angular та Vue, дозволяючи розробникам використовувати типізацію в існуючих та нових проектах.

Крім того, TypeScript має велику спільноту та активний розвиток. Вона постійно оновлюється, додаючи нові можливості та поліпшення. Регулярні випуски та добре документована API роблять TypeScript привабливим вибором для багатьох розробників, які шукають ефективний інструмент для розробки якісного, масштабованого та надійного коду.

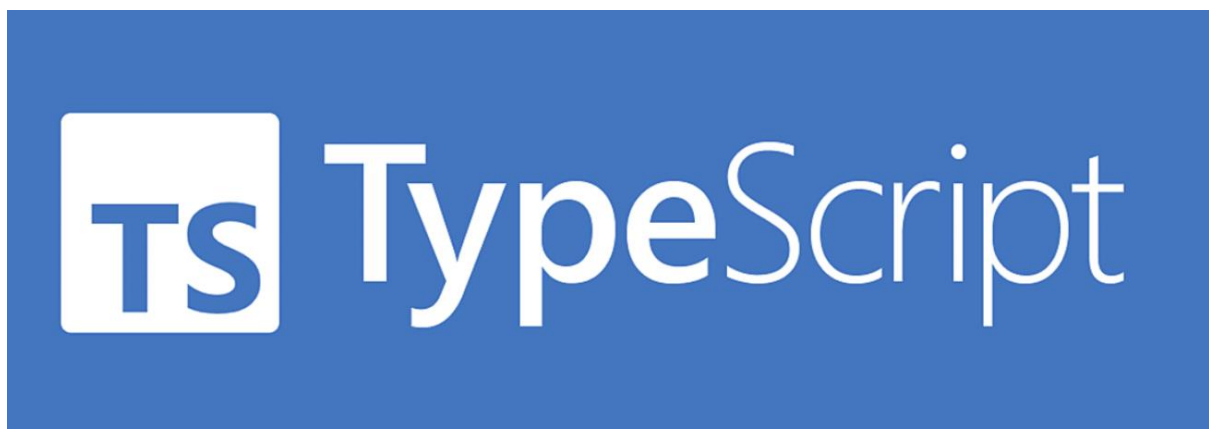


Рис 3.3. TypeScript

Sockets представляють собою механізм комунікації між програмами або процесами, які працюють на різних комп'ютерах в мережі. Сокети дозволяють встановлювати з'єднання між двома чи більше вузлами і передавати дані в обох напрямках.

Щоб краще зрозуміти сокети [14], давайте уявимо, що це схоже на телефонний дзвінок. Кожен абонент має свій телефон (сокет), і вони можуть викликати один одного, встановлюючи двосторонній канал зв'язку. Спілкування відбувається через цей канал, і обидві сторони можуть надсилати та отримувати повідомлення.

Сокети широко використовуються в різних областях програмування, зокрема в мережевому програмуванні, веб-розробці та інших сценаріях, де необхідно передавати дані між вузлами мережі. Одним із популярних

використань сокетів є реалізація клієнт-серверних додатків, де клієнти та сервери можуть обмінюватися інформацією в режимі реального часу.

Основна ідея сокетів полягає в тому, що вони дозволяють програмам взаємодіяти на різних рівнях, надаючи спосіб для передачі даних через мережу. Це може бути використано для обміну текстовими повідомленнями, передачі файлів або навіть стрімінгу мультимедійних даних.

Сокети виступають важливою складовою для багатьох технологій, таких як WebSocket для реалізації двостороннього зв'язку між клієнтом та сервером у веб-додатках. У світі програмування сокети - це ключовий інструмент для забезпечення комунікації і взаємодії між програмами, що розташовані на різних пристроях або вузлах мережі.

Styled Components (рис. 3.4.) - це бібліотека для реактивного стилізації компонентів в React-додатках. Ця технологія надає розробникам потужний і зручний спосіб управління стилями, використовуючи JavaScript або TypeScript.

Styled Components [15] може розпочатися із зазначення того, що традиційний підхід до стилізації React-компонентів часто використовує CSS-файли або inline-стили, що може вести до збільшення складності та важкості управління кодом. Відповідь на це виглядає так: Styled Components дозволяють вписувати стилі безпосередньо в код компонентів, використовуючи потужні можливості JavaScript-шаблонів та тегованих шаблонів строк.

Однією з ключових переваг Styled Components є можливість використання динамічних стилів, що базуються на змінних, пропсах компонентів чи умовах виконання. Це полегшує створення адаптивних та змінних інтерфейсів, а також спрощує управління стилізацією в залежності від умов застосування.

Ще однією важливою характеристикою Styled Components є можливість використовувати вложеність та композицію стилів. Компоненти можна вкладати один в одного, а їх стилі будуть автоматично комбінуватися. Це дозволяє писати чистий та організований код, розділяючи логіку компонента від його оформлення.

Завдяки Styled Components, розробники можуть отримати більше контролю над стилізацією та зменшити можливість конфліктів у стилях. Ця бібліотека стала популярною у реакт-спільноті і продовжує допомагати розробникам ефективно впоратися з завданнями стилізації у сучасних веб-додатках.

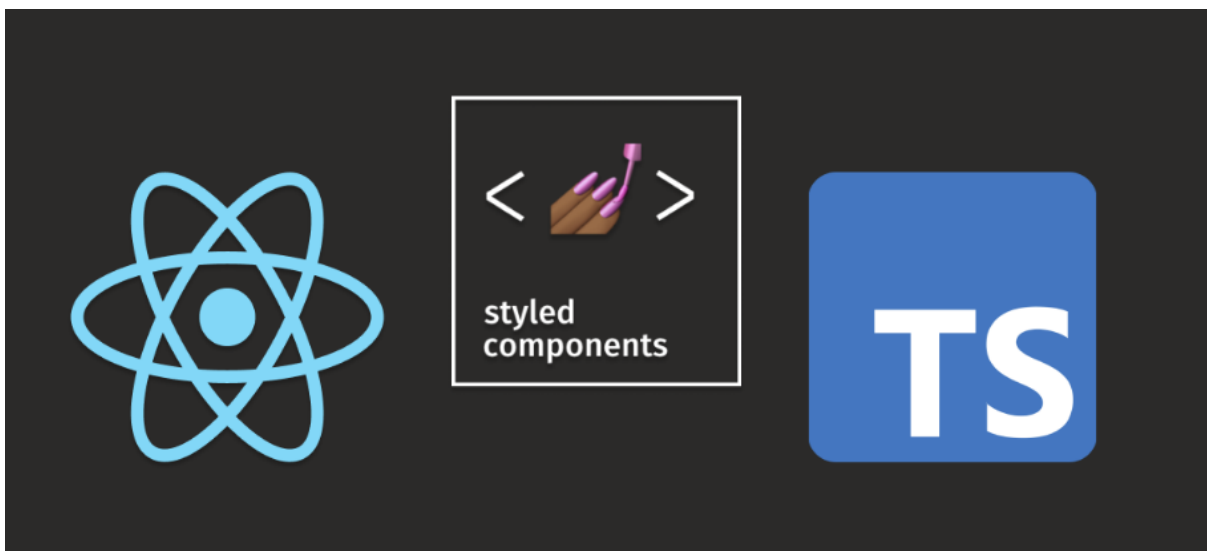


Рис 3.4. Styled Components

SWR (State-While-Revalidate) (рис. 3.5.) - це бібліотека для управління станом даних та кешування в реактивних веб-додатках. Розроблена компанією Vercel, SWR [16] робить взаємодію з сервером та кешування даних більш простою та ефективною, особливо в контексті React-додатків.

Розповідь про SWR може розпочатися із зазначення, що управління станом і отримання даних з сервера є неодмінною частиною багатьох веб-

додатків. Традиційний підхід може включати в себе використання стейт-менеджерів, ефектів та ряду інших реактивних конструкцій. SWR призначений для спрощення цього процесу.

Однією з головних переваг SWR є його простота використання. Він пропонує простий API для визначення джерела даних та автоматичного кешування результатів. SWR автоматично керує поновленням даних, дозволяючи підтримувати інтерфейс користувача завжди актуальним та швидкодіючим.

Ще однією цікавою функцією є можливість використання власних функцій для обробки даних, які повертаються з сервера, або для визначення умов поновлення кешу. Це дає більшу гнучкість та контроль над тим, як дані обробляються та відображаються в інтерфейсі.

SWR також добре інтегрується з React та може бути використаний разом із популярними бібліотеками для стейт-менеджменту, такими як Redux або MobX.

Використання SWR дозволяє розробникам покращити ефективність та простоту управління станом даних у їхніх React-додатках, забезпечуючи при цьому оптимальну продуктивність та високу якість користувацького досвіду.



SWR

React Hooks for Data Fetching

Рис 3.5. SWR

Framer Motion - це бібліотека для додатків React, яка дозволяє розробникам створювати динамічні та анімовані інтерфейси швидко та ефективно. Розроблена командою Framer, вона надає вдосконалені засоби для керування анімаціями, що робить їх більш простими та потужними.

Початок розповіді про Framer Motion [17] може розпочатися з акценту на те, що анімації в інтерфейсах важливі для створення привабливих та високоефективних веб-додатків. Framer Motion дозволяє розробникам досягати цього завдяки простому та декларативному API для створення анімацій.

Ця бібліотека дозволяє застосовувати анімації до компонентів React за допомогою динамічно змінюваних властивостей, які визначають, як об'єкт або компонент повинен виглядати під час різних станів анімації. Також Framer Motion підтримує анімації переходів між різними сторінками чи компонентами, що полегшує створення зручного та інтуїтивного користувацького досвіду.

Бібліотека підтримує велику кількість різних типів анімацій, включаючи переміщення, зміну розміру, повороти, нахил, прозорість, колір та багато інших. Користувачі можуть використовувати ключові рамки, щоб точно визначити розташування об'єкта під час різних етапів анімації.

Framer Motion також пропонує зручний спосіб обробки подій анімації, що дозволяє виконувати певні дії під час відтворення або завершення анімацій, що робить бібліотеку гнучкою та потужною у використанні.

Завдяки Framer Motion, розробники можуть легко додавати анімації до своїх React-додатків, покращуючи візуальний досвід користувачів та створюючи привабливі та динамічні інтерфейси.

Токен - це символ або код, який представляє деяку вартість чи право та може бути використаний для отримання доступу або виконання певних операцій. Токени широко застосовуються в різних контекстах, таких як

криптовалюти, інформаційна безпека, веб-розробка та багато інших областей.

Історія токенів в інформаційній безпеці пов'язана з аутентифікацією та авторизацією. Токени використовуються для підтвердження ідентичності користувачів або систем та для визначення їхніх прав доступу. Наприклад, токени можуть бути використані для отримання доступу до ресурсів або послуг після успішної аутентифікації.

В контексті криптовалют токени є цифровими активами, які представляють конкретні речі або послуги. Токени можуть бути використані для участі в різних проектах або обмінюватися на інші криптовалюти. Наприклад, токени ERC-20 на блокчейні Ethereum визначають стандарт для створення токенів, які можуть бути взаємозамінні між різними додатками та платформами.

У веб-розробці токени використовуються для авторизації та захисту ресурсів. Наприклад, токени доступу OAuth використовуються для авторизації користувачів та надання їм доступу до сторонніх сервісів без необхідності введення свого пароля.

Токени стали важливим елементом сучасних технологій, забезпечуючи безпеку та зручність в області аутентифікації та авторизації. Їхнє використання в різних сферах показує, наскільки вони універсальні та ефективні в забезпеченні безпеки та обміні ресурсами.

Cookies, або файли cookie, — це невеликі файли, які зберігаються на комп'ютері користувача та містять невелику кількість інформації, пов'язаної з веб-сайтом. Вони слугують засобом для збереження даних між різними сторінками та візитами, роблячи взаємодію із веб-сайтами більш ефективною та приємною для користувачів.

Історія cookies почалася в 1994 році, коли програміст Луї Монталто розробив їх для платформи Netscape Navigator. Вони були призначені для

того, щоб зберігати обраний мовний варіант користувача, але впродовж часу ця ідея розширилася.

Зараз cookies грають важливу роль у веб-розробці та інтернет-маркетингу. Вони можуть зберігати інформацію про користувачів, таку як персональні налаштування, предпочтєння та дані для авторизації. Це дозволяє веб-сайтам створювати більш персоналізований та зручний досвід для кожного користувача.

Однак, треба відзначити, що cookies також стали об'єктом обговорення щодо приватності користувачів. У світі, де питання захисту особистих даних набуває все більшого значення, технології, що забезпечують конфіденційність та контроль над персональною інформацією, стають важливими.

Загалом, cookies є важливою складовою інтернет-екосистеми, яка забезпечує зручність та персоналізацію взаємодії користувачів із веб-сайтами. Їхній розвиток відображає постійне прагнення до поліпшення інтернет-технологій для кращого досвіду використання мережі. Кінець форми

3.3. Кроссбраузерність

Кроссбраузерність — це концепція в розробці веб-сайтів та програмного забезпечення, яка означає здатність продукту працювати на різних веб-браузерах та забезпечувати однаковий функціонал та вигляд для користувачів незалежно від браузера, який вони використовують. Кроссбраузерність [18] є важливою складовою успішного та доступного веб-досвіду. Дозвольте мені поділитися деякими ключовими аспектами цієї теми.

Коли ми говоримо про кроссбраузерність, ми стикаємося з різноманітністю веб-браузерів, таких як Google Chrome, Mozilla Firefox,

Safari, Microsoft Edge, та інші. Кожен з цих браузерів має свої власні особливості та різні інтерпретації стандартів веб-технологій. Тому розробники повинні узгоджувати свої продукти так, щоб вони працювали оптимально на будь-якому з них.

Одним з основних принципів кроссбраузерності є використання валідних та стандартних кодів. Розробники повинні слідувати специфікаціям та стандартам, визначеним організаціями, такими як W3C (World Wide Web Consortium), щоб забезпечити зручну роботу своїх продуктів на будь-якому браузері.

Додатково, важливо використовувати тести на різних браузерах, щоб впевнитися в тому, що ваш веб-сайт чи додаток функціонує правильно та виглядає однаково на всіх популярних платформах. Автоматизовані тести дозволяють виявити проблеми та виправити їх ще до випуску продукту.

Запам'ятовуючи про кроссбраузерність, розробники не лише забезпечують гладку та однакову роботу свого продукту для всіх користувачів, але також покращують його доступність та роблять його більш конкурентоспроможним на ринку веб-технологій.

3.4. Архітектура додатку

Компонентна архітектура [19] — це підхід до розробки програмного забезпечення, який базується на концепції розділення системи на окремі компоненти чи модулі. Кожен компонент відповідає за виконання конкретної функції або обов'язку і може бути розроблений та підтримуваний незалежно від інших компонентів. Цей підхід спрощує розробку, підтримку та масштабування програмного забезпечення. Ось кілька ключових аспектів компонентної архітектури:

— розділення на компоненти у компонентній архітектурі система поділяється на малий набір незалежних компонентів, кожен з яких виконує конкретну функцію чи набір функцій. Це може включати в себе веб-компоненти, серверні сервіси, або інші елементи системи;

— незалежність та переахтовність компоненти розробляються та тестуються незалежно один від одного, що дозволяє змінювати, вдосконалювати або видаляти конкретні компоненти, не торкаючись інших частин системи. Це сприяє переахтовній системі та полегшує впровадження нового функціоналу;

— гнучкість та масштабованість компоненти можуть бути використані багаторазово та комбіновані для створення різних конфігурацій системи. Це забезпечує гнучкість та можливість масштабування системи відповідно до зростання потреб;

— інтерфейси та взаємодія компоненти взаємодіють один з одним через визначені інтерфейси. Це означає, що спосіб, яким компоненти обмінюються інформацією, чітко визначено та документовано, що сприяє стабільності системи та спрощує співпрацю між командами розробників;

— підтримка реінжинірингу компоненти можуть бути перероблені, вдосконалені чи замінені без впливу на інші частини системи. Це робить архітектуру готовою до реінжинірингу та адаптації до змін вимог;

— узагальнення та спрощення тестування компонентна архітектура значно спрощує тестування. Через те, що кожен компонент може бути тестований окремо, можна впевнено перевіряти його функціональність та вплив на систему в цілому. Це робить тестування більш ефективним та забезпечує високий рівень надійності програмного продукту;

— спільне використання та відновлення у компонентній архітектурі розробники можуть використовувати сторонні компоненти або

бібліотеки, щоб прискорити розробку та використання вже перевірених та оптимізованих рішень. Це зменшує час розробки та ризик появи помилок;

- модульність та розширюваність. компоненти можуть бути розглянуті як модулі, які можна легко розширювати та замінювати. Нові функції можуть бути додані, а система може адаптуватися до змін без необхідності перебудови усього продукту;

- захист від збоїв та зручність підтримки оскільки кожен компонент працює незалежно, при збоях або проблемах в одному компоненті, інші можуть продовжувати працювати. Це забезпечує вищу стійкість до збоїв та полегшує виявлення, виправлення та підтримку системи;

- поділ робочих обов'язків компонентна архітектура полегшує розподіл робочих обов'язків серед команд розробників. Кожен може працювати над власним компонентом, не втручаючись у роботу інших, що підвищує продуктивність та ефективність розробки.

Загалом, компонентна архітектура є потужним інструментом для створення та управління складним програмним забезпеченням. Вона дозволяє ефективно організовувати та управляти функціональністю, спрощує розробку та тестування, і робить продукт гнучким та готовим до майбутніх змін.

3.5. Інтерфейс додатку

Веб-додаток працює на більшості сучасних комп'ютерах та пристроях з відповідною підтримкою браузерів. Ось мінімальні вимоги, які можуть забезпечити стабільну роботу веб-додатку:

- Центральний процесор (ЦП): ЦП: Intel, AMD 4-го покоління або вище;

- Графічний адаптер: 3D адаптер nVidia, Intel, AMD/ATI;

- Відеопам'ять: Мінімум 128 МБ;
- Оперативна пам'ять (RAM): Мінімум 2 ГБ.

Веб-додаток коректно працює на останніх версіях браузерів таких як Google Chrome, Mozilla Firefox, Microsoft Edge, Safari для macOS та Safari.

Екран авторизації – це важлива частина будь-якого веб-додатка або сервісу, що забезпечує безпечний та контрольований доступ користувачів. Зазвичай цей екран використовується для перевірки ідентичності та надання доступу до облікового запису.

На екрані авторизації (рис. 3.6.) користувачі зазвичай вводять свої облікові дані, такі як ім'я користувача та пароль. Забезпечуючи захист конфіденційності, такі екрани також можуть використовувати різноманітні методи автентифікації, наприклад, двофакторну автентифікацію або використання токенів.

Для впровадження ефективного екрану авторизації важливо дотримуватися сучасних практик безпеки. Це включає в себе використання захищених протоколів передачі даних, таких як HTTPS, а також обробку та зберігання облікових даних користувачів у безпечний спосіб.

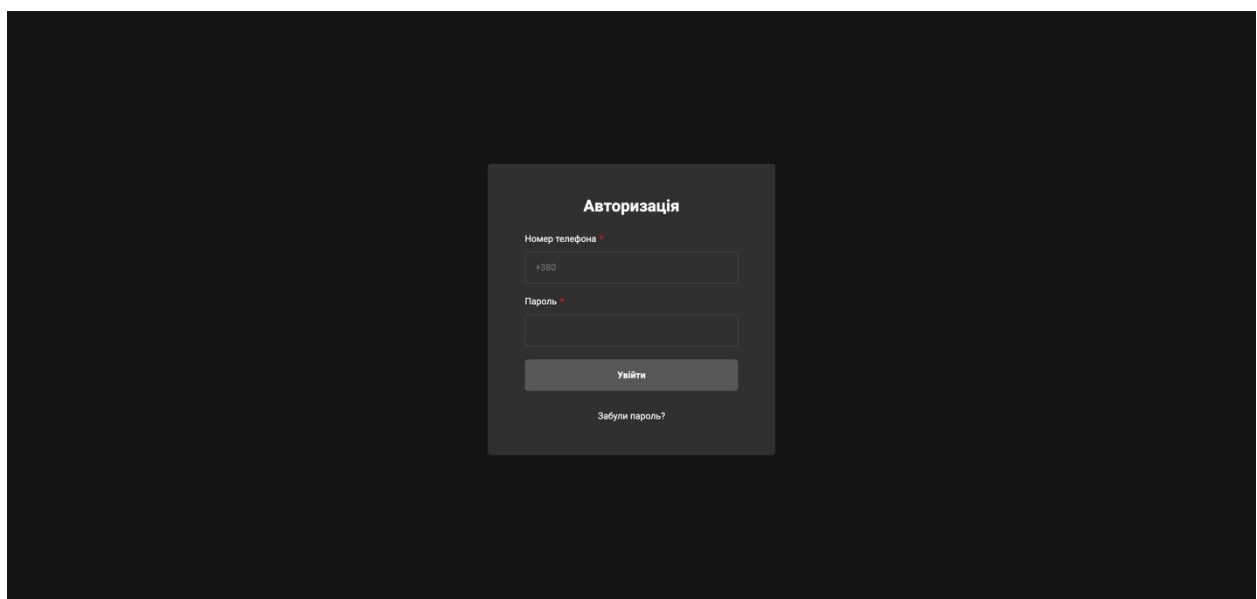


Рис 3.6. Екран авторизації

Екран відновлення паролю (рис. 3.7.) відіграє важливу роль в веб-додатках, надаючи користувачам можливість відновити свій доступ у випадку, якщо вони забули пароль або хочуть змінити його з причин безпеки.

На екрані відновлення паролю зазвичай присутні поля для введення електронної пошти чи імені користувача, пов'язаного з обліковим записом. Після введення цих даних користувачам надсилається лист із посиланням для відновлення паролю або ж код для підтвердження їхньої особи.

Важливо забезпечити безпеку цього процесу, використовуючи шифрування та заходи захисту, щоб уникнути можливих загроз для конфіденційності користувачів. Додатково, екран відновлення паролю може містити підказки чи інструкції для користувачів, які можуть виникнути під час процесу відновлення.

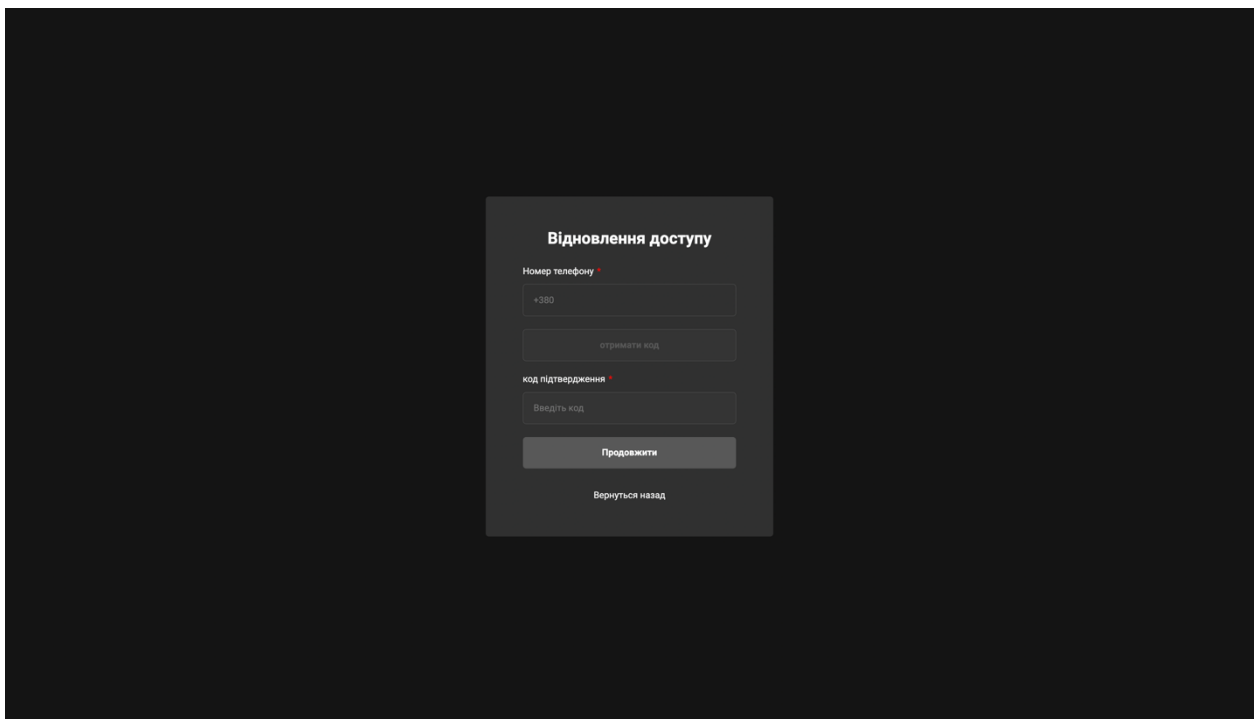


Рис 3.7. Екран відновлення паролю

Екран з функціоналом відображення карти клубу (рис. 3.8.) — це важливий інструмент для адміністраторів або користувачів, які відповідають за управління хостами та їх розташуванням у конкретних зонах клубу чи об'єкта. Давайте розглянемо основні функції цього екрану.

Екран надає інтерактивну карту приміщення чи клубу, де кожен хост та його розташування позначені на карті. Це забезпечує зручний та візуальний спосіб відслідковування розташування обладнання чи ресурсів.

Функція Drag and Drop (DnD) дозволяє користувачам легко та швидко змінювати місцезнаходження хостів на карті. Це може бути корисним для оптимізації розміщення обладнання чи для реорганізації приміщення залежно від потреб.

Екран з функціоналом відображення стану хостів надає важливу інформацію щодо доступності та статусу окремих хостів чи систем в мережі. Це корисний інструмент для адміністраторів та користувачів, які хочуть візуально відслідковувати стан об'єктів або ресурсів.

Статус "Неактивний" вказує на те, що хост не активний або вимкнений. Це може свідчити про те, що система вимкнена або не функціонує у даний момент.

Статус "Включений" показує, що хост активний і готовий до роботи, але не має активних сесій або специфічних завдань.

Статус "Занят" вказує на те, що хост використовується чи виконує якісь завдання, але не має активних сесій.

Статус "Забронирован" вказує на те, що хост або ресурс має зарезервовану часову або ресурсну квоту для майбутніх використань.

Елемент може містити додаткову інформацію про стан хоста, таку як помилки, повідомлення або інші деталі, які можуть бути корисними для аналізу або управління.

Екран із відображенням карти клубу та вбудованим функціоналом надає зручний та ефективний інструмент для управління та моніторингу розташування хостів, їх стану та параметрів у реальному часі.

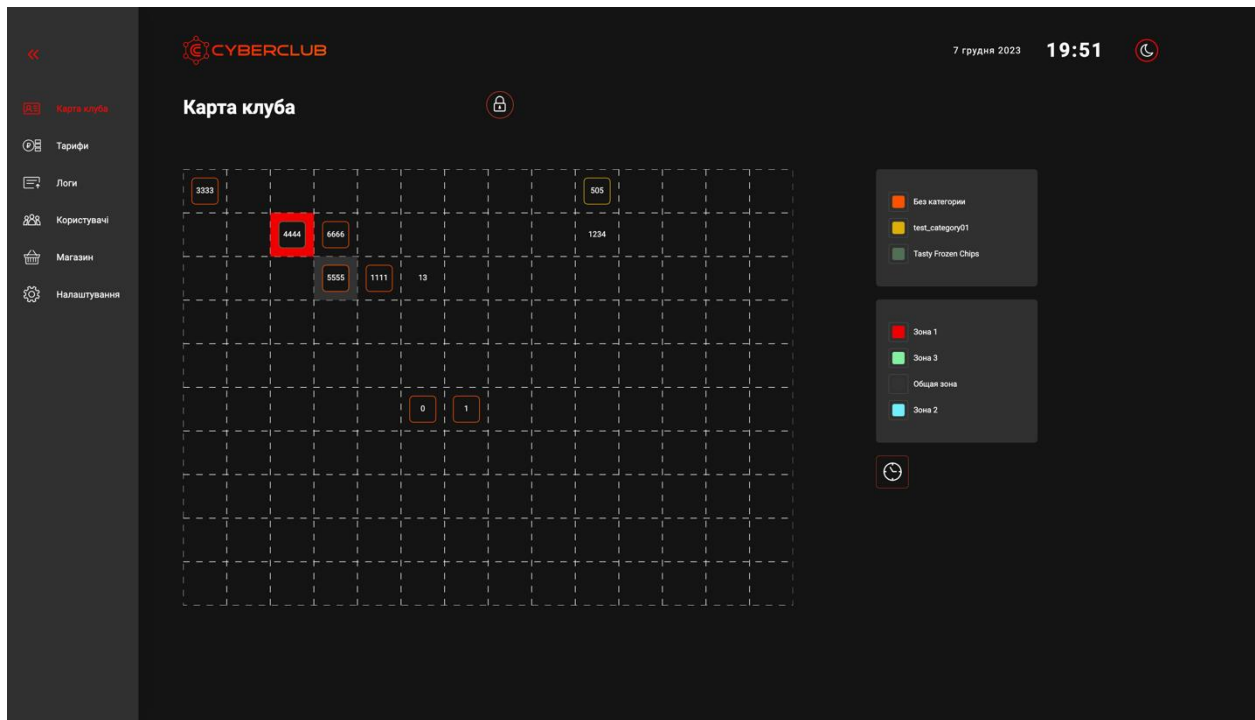


Рис 3.8. Карта клубу

Функція "Додати хост" (рис. 3.9.) дозволяє користувачам легко вводити нові пристрої або ресурси до системи. Це може включати в себе визначення параметрів нового хоста, таких як ім'я, характеристики та призначення зони на карті клубу чи об'єкта. Цей функціонал робить процес розширення системи більш простим та інтуїтивно зрозумілим.

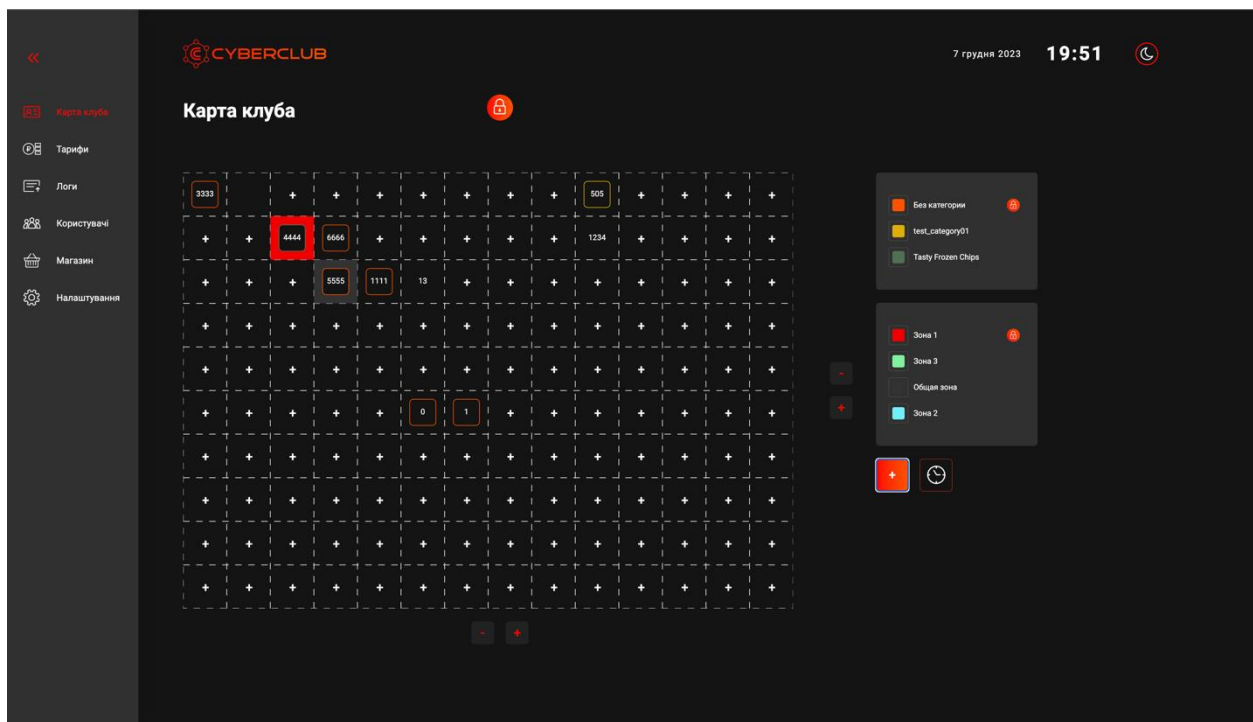


Рис 3.9. Карта клубу додати хост

Aside (рис. 3.10.) з можливістю додавання зон та груп — це інструмент для управління просторами та організації об'єктів чи ресурсів у великих системах.

Функція дозволяє користувачам створювати нові зони чи області на карті клубу чи об'єкта. Зони можуть визначати різні функціональні частини приміщення та використовуватися для організації та класифікації пристроїв чи ресурсів.

Дозволяє групувати хости, ресурси або об'єкти для зручності управління. Користувач може створювати групи за певними критеріями, такими як призначення, функціональність чи розташування, щоб швидше та легше керувати великим обсягом об'єктів.

Структурувати та організовувати простір, забезпечуючи ефективне управління та легкий доступ до інформації. Додавання зон та груп стає важливим інструментом для адміністраторів та користувачів, які

відповідають за оптимізацію та структурування просторів або об'єктів у великих системах.

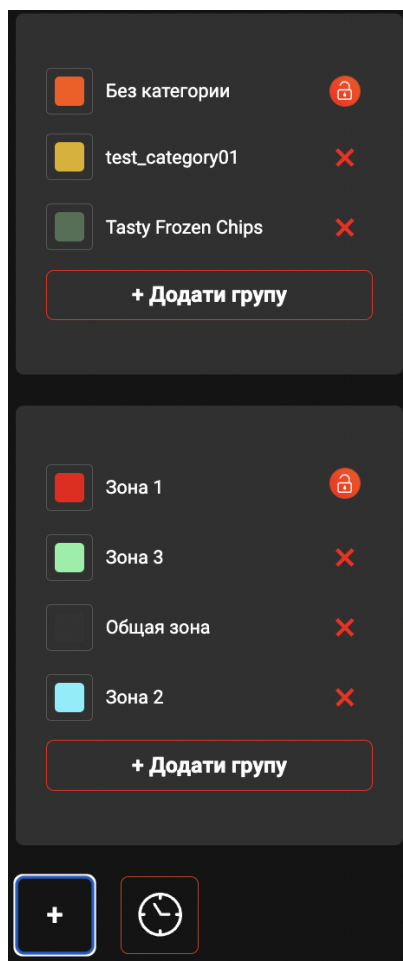


Рис 3.10. Редагування зони та групи

Користувачі можуть вносити зміни в параметри хостів (рис. 3.11.), такі як назва, статус, характеристики та інше. Це дозволяє актуалізувати та налаштовувати інформацію відповідно до потреб.

Дозволяє призначати хостам конкретні зони або розділи на карті клубу. Такий функціонал може бути корисним у випадках, коли клуб або приміщення поділяється на різні області з різними функціональними завданнями.

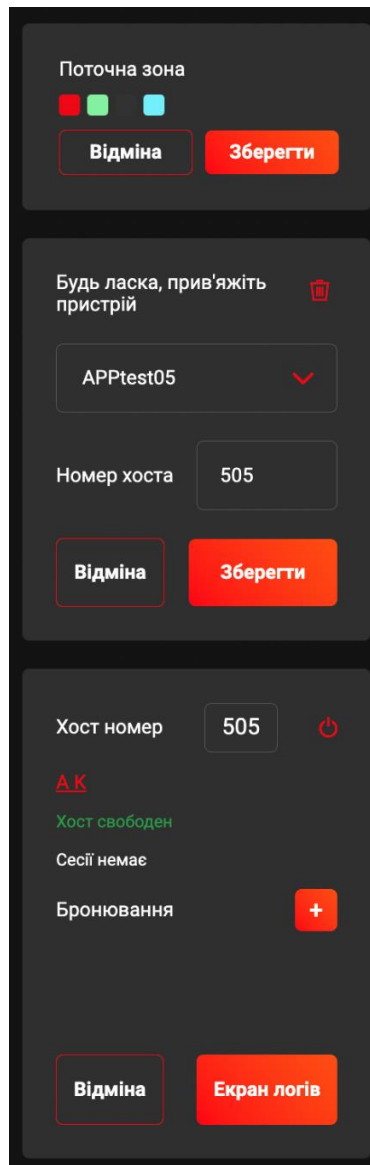


Рис 3. 11. Редагування хоста

Екран логів для хостів (рис. 3.12.) - це інструмент, який надає важливу інформацію про події та взаємодії, що відбуваються з конкретними хостами у системі. Цей екран відіграє ключову роль у відслідковуванні та діагностиці подій для підтримки та удосконалення функціонування обладнання чи ресурсів.

Екран логів показує історію важливих подій, таких як включення/вимкнення, зміни статусів, помилки чи інші взаємодії з конкретним хостом.

Логи містять повну інформацію про кожну подію, включаючи дату, час, тип події, ідентифікатор хоста, та можливі додаткові деталі.

Загалом, екран логів для хостів є необхідним елементом для ефективного управління та моніторингу хостів, забезпечуючи доступ до детальної інформації про їхню діяльність та функціонування.

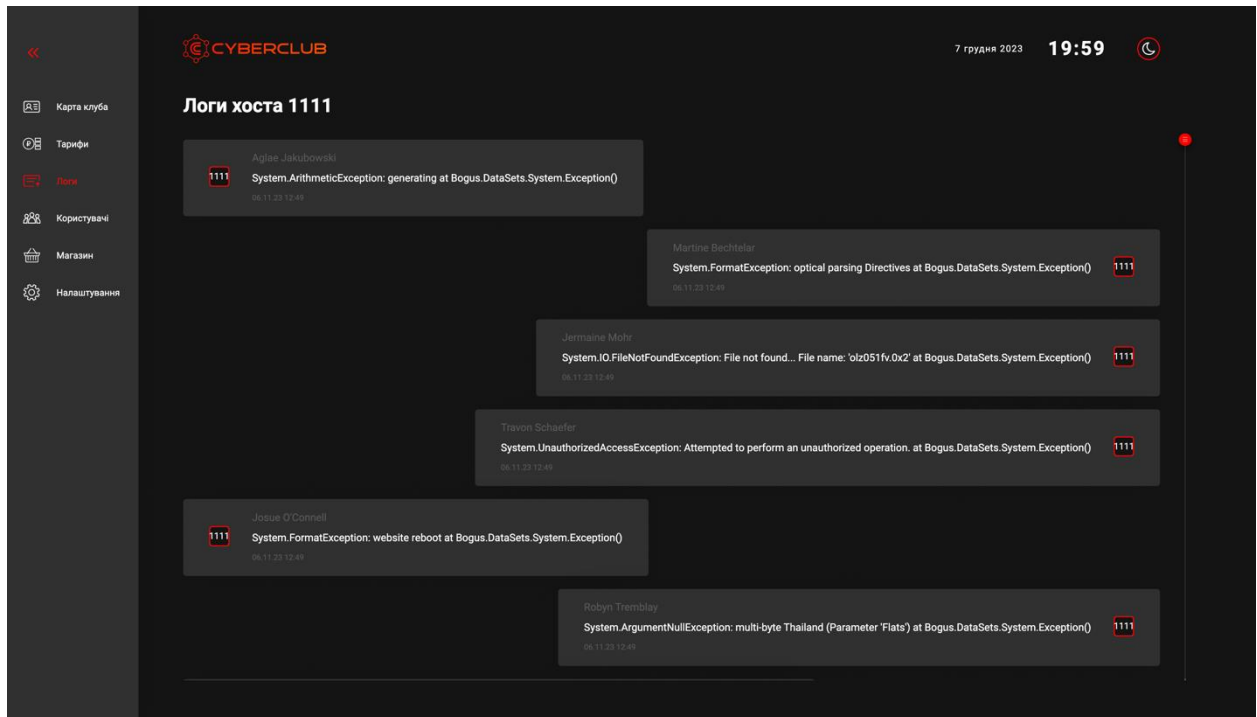


Рис. 3.12. Екран логів

Екран користувачів кіберклубу (рис. 3.13.) - це інтерфейс, який надає зручний та організований доступ до інформації про користувачів, що використовують послуги кіберклубу. Такий екран може включати різноманітні функції для ефективного управління та спілкування з членами клубу. Основні характеристики екрану користувачів кіберклубу.

Екран надає візуальний огляд активних користувачів клубу, включаючи їхні імена, статуси та можливо, іншу основну інформацію.

Управління Правами Доступу: Адміністратори кіберклубу можуть використовувати цей екран для надання та керування правами доступу

користувачів, забезпечуючи безпечний та контрольований доступ до різних ресурсів.

Збір та відображення статистичних даних щодо активності користувачів, таких як тривалість сеансу, витрати часу на різних ресурсах та інші параметри, може допомагати адміністрації клубу в аналізі та вдосконаленні обслуговування.

Можливість надсилання повідомлень та сповіщень користувачам клубу про події, акції або важливі повідомлення.

Користувачі можуть мати можливість встановлювати та змінювати особисті налаштування, які пов'язані з їхніми акаунтами, такі як пароль, електронна адреса чи інші персональні дані.

Екран користувачів кіберклубу є ключовим інструментом для зручного управління та спілкування з аудиторією клубу, забезпечуючи як адміністраторам, так і користувачам, зручний доступ до важливої інформації та функціоналу.

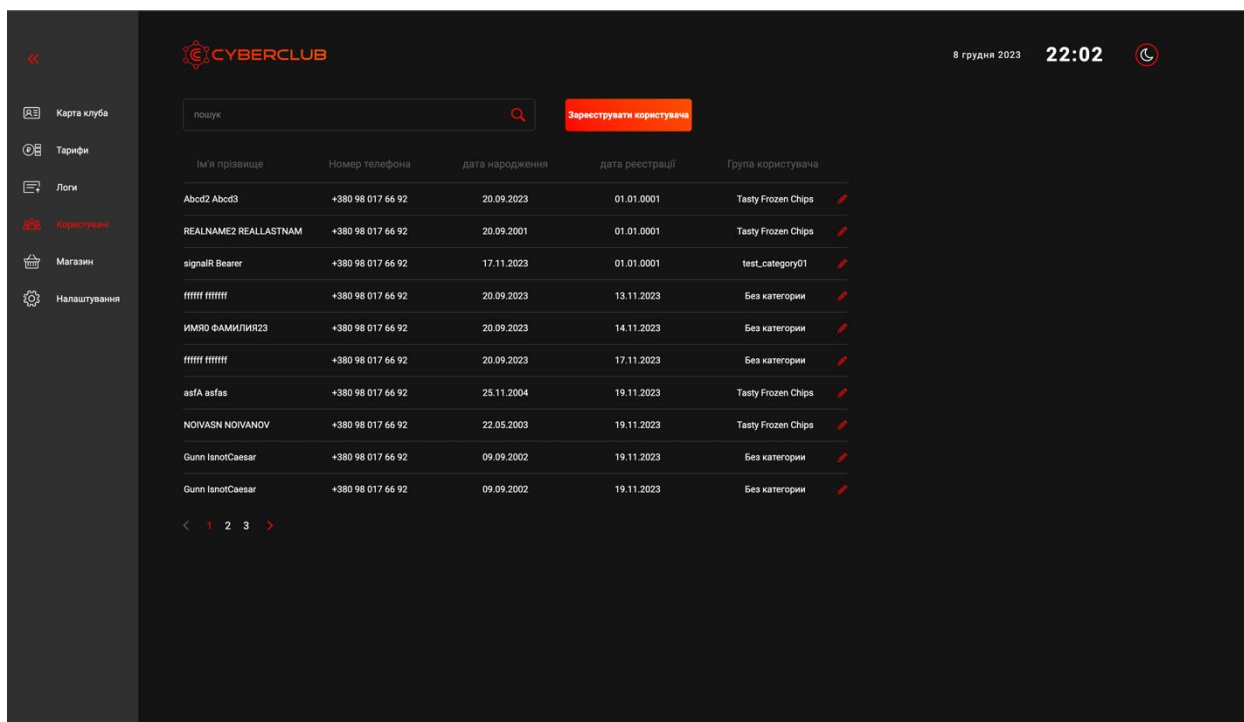


Рис. 3.13 Екран користувачів

Модальне вікно реєстрації користувача (рис. 3.14.) - це відмінний спосіб надати зручний та зосереджений інтерфейс. Ось кілька ключових аспектів цього модального вікна:

- зручний дизайн та введення інформації модальне вікно зазвичай має привабливий та інтуїтивно зрозумілий дизайн, який полегшує користувачеві введення основної інформації, такої як ім'я, електронна пошта та пароль;

- підказки та інструкції для полегшення процесу реєстрації, модальне вікно може містити корисні підказки та інструкції, щоб користувач відчував себе впевнено та інформовано;

- моментальна перевірка доступності деякі вікна реєстрації можуть миттєво перевіряти доступність обраного імені користувача або електронної адреси, щоб уникнути конфліктів та спростити вибір;

- безпека та підтвердження важливим елементом є введення елементів безпеки, таких як пароль, та можливість підтвердження облікових даних, щоб убезпечити процес реєстрації;

- візуальне підтвердження реєстрації після успішної реєстрації модальне вікно може надати візуальне підтвердження та, можливо, надіслати ласкаве повідомлення привітання.

Модальне вікно реєстрації користувача має за мету зробити процес приєднання до платформи простим та приємним, створюючи перший позитивний враження від взаємодії з веб-сайтом.

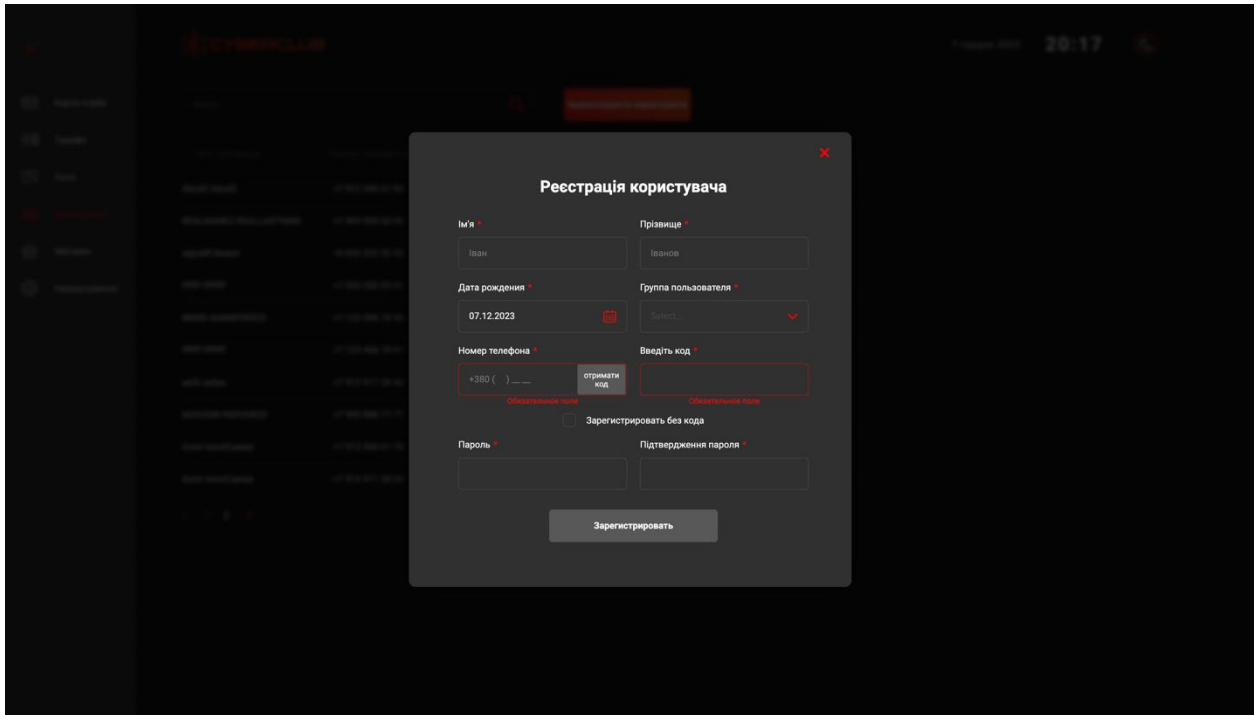


Рис. 3.14. Реєстрація користувача

Екран користувача з інформацією (рис. 3.15.) - це інтерфейс, який надає повний обзор основної інформації та можливостей кожного користувача в системі. Ось деякі ключові елементи цього екрану:

— телефон відображає номер телефону користувача, що є основним контактним засобом;

— дата реєстрації вказує на дату, коли користувач зареєстрував свій обліковий запис;

— дата народження відображає дату народження користувача, яка може бути важливою для вікових обмежень або персоналізації обслуговування;

— група користувача показує приналежність користувача до конкретної групи, що може визначати його права та можливості;

— баланс відображає фінансовий баланс користувача, який може включати в себе кредити, знижки чи інші фінансові показники.

— займає хост номер вказує на номер хоста, який в даний момент використовується користувачем;

— активна сесія інформує про те, чи користувач має активну сесію або віддалено підключений до системи;

— логи користувача забезпечує доступ до логів, де можна переглядати історію дій та подій, пов'язаних з обліковим записом користувача;

— відправити SMS надає можливість відправити текстове повідомлення на телефон користувача;

— надіслати повідомлення на ПК Забезпечує можливість відправити повідомлення користувачеві на комп'ютер;

— редагувати дозволяє змінювати основні дані користувача, такі як контактна інформація або групові приналежності;

— завершити сесію завершення активної сесії користувача, що може бути корисним для безпеки та управління доступом;

— заблокувати надає можливість заблокувати обліковий запис користувача, щоб уникнути невірної використання чи несанкціонованого доступу.

Цей екран надає повний спектр інструментів та інформації для ефективного управління та спілкування з кожним користувачем системи.

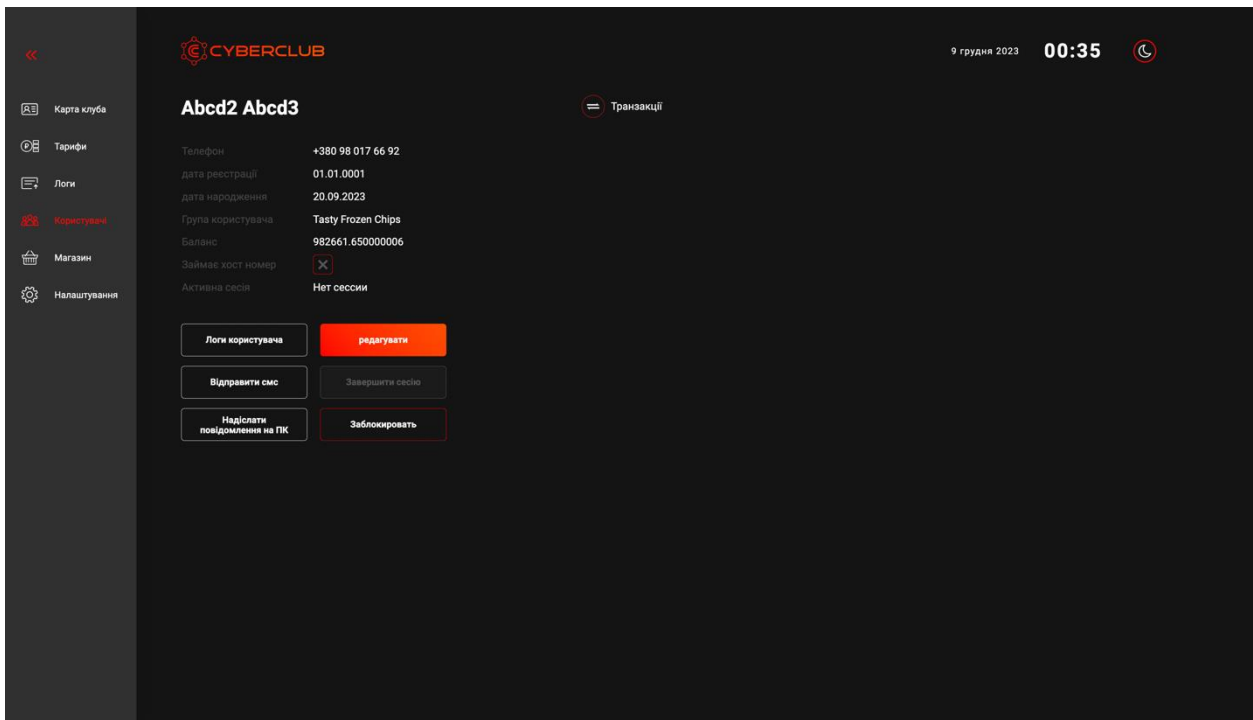


Рис. 3.15. Інформація користувача

Модальне вікно з транзакціями (рис. 3.16.) - це інтерфейс, який забезпечує користувача зручним та структурованим способом переглядати та керувати своїми фінансовими операціями.

Ось кілька ключових елементів такого модального вікна:

— фільтр "Усе/Поповнення/Списання" можливість вибрати тип транзакцій для відображення, що дозволяє користувачеві швидко фільтрувати та знаходити необхідні операції;

— таблиця відображення транзакцій у вигляді таблиці для зручного та структурованого перегляду. Кожен рядок містить інформацію про конкретну транзакцію;

— дата та час інформація про точний момент виконання транзакції, що допомагає користувачеві відслідковувати часові рамки подій;

— статус показник стану транзакції, такий як "Успішно," "В обробці," або інший, що надає інформацію про поточний стан операції;

— сума вказує на фінансову вартість транзакції.

Комбінація цих елементів робить модальне вікно з транзакціями потужним інструментом для управління фінансовими операціями. Користувач може легко відфільтрувати та знаходити необхідні транзакції, переглядати їхні деталі та здійснювати контроль над фінансами.

Транзакції		
Усе	Поповнення	Списання
Дата і время	Статус	Сумма
14.11.2023 07:00	Возрат	+999 999 999 грн
14.11.2023 07:00	Списання	-1 000 грн
14.11.2023 07:06	Возрат	+9 999 999 999 грн
14.11.2023 07:08	Списання	-26 488,7 грн
14.11.2023 07:13	Списання	-99 999,99 грн
14.11.2023 07:16	Списання	-99 999,99 грн
14.11.2023 07:21	Списання	-118 888,5 грн
16.11.2023 12:17	Списання	-628 грн
16.11.2023 12:22	Списання	-102 грн
16.11.2023 12:27	Списання	-100 грн
16.11.2023 13:09	Списання	-100 грн
17.11.2023 08:15	Списання	-97 342,5 грн
17.11.2023 09:34	Списання	-40 грн
20.11.2023 08:08	Списання	-110 000 грн
20.11.2023 08:10	Списання	-110 000 грн
20.11.2023 08:13	Списання	-770 000 грн
20.11.2023 08:28	Списання	-110 000 грн

Рис. 3.16. Інформація транзакцій користувача

Екран каталогу (рис. 3.17.) - це важлива частина системи, яка надає користувачам зручний та організований доступ до товарів чи послуг. Ось кілька ключових елементів цього екрану:

— aside "Усі Категорії" бічна панель "Усі категорії" надає швидкий доступ до різних категорій товарів. Вона спрощує навігацію та допомагає користувачам швидко знаходити та переглядати товари, які їх цікавлять;

— пошук функція пошуку дозволяє користувачам швидко знаходити конкретні товари або фільтрувати їх за певними критеріями. Це полегшує процес пошуку та вибору продуктів.

— додати товар кнопка "Додати товар" надає можливість адміністраторам чи працівникам магазину швидко вводити нові товари в каталог. Це робить процес оновлення асортименту простим та ефективним;

— видалити товар опція "Видалити товар" дозволяє видаляти продукти, які вже не є актуальними чи вибрані з каталогу. Це сприяє підтримці актуальності та чистоти каталогу;

— редагувати товар кнопка "Редагувати товар" дозволяє вносити зміни до існуючих записів про товари. Це корисно для корегування цін, описів або інших деталей;

— інформація про товар відкривши конкретний товар, користувач отримує доступ до детальної інформації про нього, включаючи зображення, характеристики, ціну та інші важливі дані.

Екран каталогу допомагає оптимізувати процеси додавання, видалення та редагування товарів, забезпечуючи зручний інтерфейс для користувачів та управлінців бізнесу.

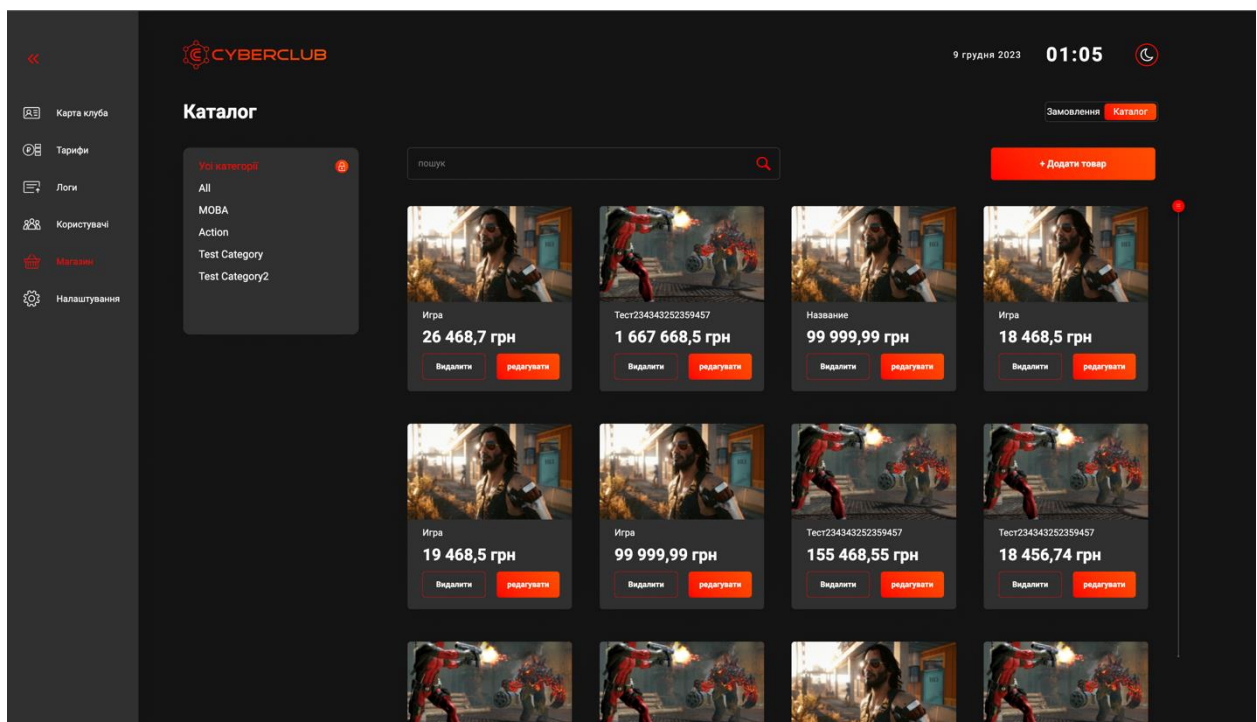


Рис. 3.17. Екран каталогу

Модальне вікно для додавання товару (рис. 3.18.) - це інтерфейс, який дозволяє користувачам чи адміністраторам зручно вводити нові товари в каталог. Ось кілька ключових елементів цього модального вікна:

- назва поле для введення назви товару. Назва ідентифікує товар та надає користувачам чітку інформацію про його характеристики;

- ціна введення вартості товару. Ця інформація дозволяє встановити ціну товару та управляти фінансовими аспектами каталогу;

- кількість вказівка кількості одиниць товару, які доступні для продажу. Це поле важливе для ведення обліку запасів та управління наявністю товару;

- категорія вибір категорії, до якої відноситься товар. Категоризація полегшує організацію каталогу та допомагає користувачам знаходити необхідні товари;

- завантаження зображення можливість завантажити зображення товару, що дозволяє надати візуальний аспект та допомагає клієнтам легше розпізнавати продукт.

Це модальне вікно створене для максимальної простоти та зручності при додаванні нових товарів до каталогу. Воно надає компактний та інтуїтивно зрозумілий інтерфейс, спрощуючи процес введення та оновлення товарів.

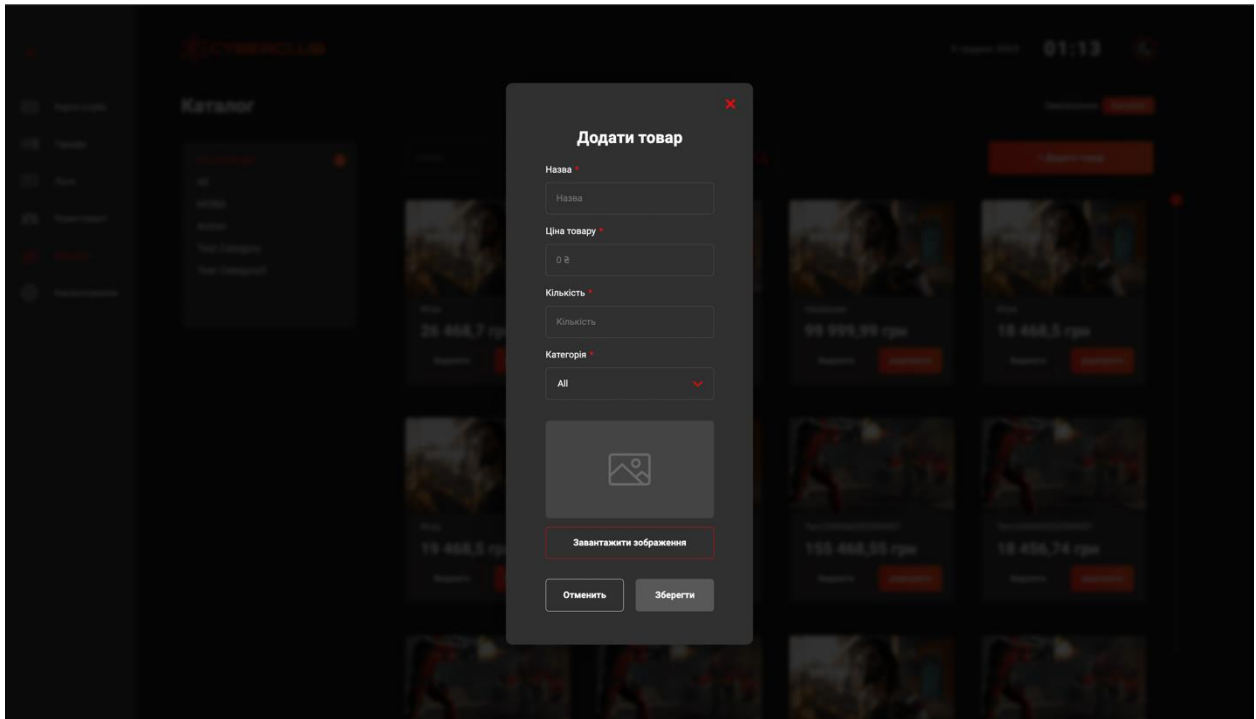


Рис. 3.18. Модальне вікно для додавання товару

Екран "Замовлення" (рис. 3.19.) - це інтерфейс, що надає інформацію та контроль над процесом обробки та управління замовленнями. Ось кілька ключових елементів цього екрану:

— номер хоста це поле ідентифікує конкретного хоста або клієнта, який здійснює замовлення. Номер хоста може служити унікальним ідентифікатором для легкої ідентифікації та відстеження замовлень;

— номер замовлення кожне замовлення має свій унікальний номер, який використовується для внутрішнього відстеження та управління;

— номер телефону контактний номер телефону забезпечує можливість зв'язку з клієнтом у випадку необхідності уточнень чи підтвердження деталей замовлення;

— дата та час інформація про час та дату, коли було здійснено замовлення, що важливо для відстеження та організації робочого процесу;

- позиції перелік товарів чи послуг, які входять до замовлення. Це дозволяє операторам та персоналу точно знаходити та готувати необхідні продукти;
- загальна ціна відображення загальної суми, яку клієнт має розрахувати за своє замовлення;
- коментарі місце для додаткових коментарів або вказівок від клієнта щодо замовлення;
- виділення маркування чи виділення особливих аспектів замовлення, таких як пріоритет чи особливі вимоги клієнта;
- замовлення закрите індикація того, що замовлення успішно виконане та закрите, що є важливою інформацією для внутрішнього контролю та аналізу.

Екран "Замовлення" є ефективним інструментом для координації та обробки замовлень, надаючи користувачам та персоналу системи повний обзор та контроль над кожним замовленням.

Номер хоста	Номер замовлення	Номер телефона	Дата та час	Позиці	Загальна ціна	Коментарі
505	43	+380 98 017 66 92	24.11.2023 17:47	1	9 101 грн	order from app
3333	29	+380 98 017 66 92	22.11.2023 13:24	1	10 грн	order from app
3333	30	+380 98 017 66 92	22.11.2023 13:49	0		order from app
4444	8	+380 98 017 66 92	14.11.2023 15:19	6	158 812,2 грн	ZAKAZ
3333	21	+380 98 017 66 92	20.11.2023 12:51	7	2 065 515,07 грн	order from app
3333	31	+380 98 017 66 92	22.11.2023 13:50	0		order from app

Рис. 3.19. Екран замовлень

Модальне вікно деталі замовлення (рис. 3.20.) - це інтерфейс, який надає докладну та структуровану інформацію про конкретне замовлення, дозволяючи операторам або адміністраторам отримати повний обзор та приймати необхідні рішення. Ось кілька ключових елементів такого модального вікна:

- номер замовлення унікальний ідентифікатор, який ідентифікує конкретне замовлення;
- дата та час інформація про момент часу, коли було здійснено замовлення, що допомагає визначити його хронологію;
- номер хоста ідентифікація хоста або клієнта, який здійснює замовлення, щоб забезпечити внутрішнє відстеження та зручну взаємодію;
- позиції докладний перелік товарів чи послуг, що входять до замовлення, включаючи їхні характеристики та кількості;
- загальна ціна відображення суми, яку клієнт має розрахувати за весь замовлення;
- коментарі можливість додавання коментарів або вказівок до замовлення, які можуть бути важливими для виконання чи обслуговування;
- статус замовлення індикація поточного стану замовлення, такого як "В обробці," "Виконане," або інший;
- контактна інформація включення номера телефону чи інших контактних даних замовника для зручного зв'язку;
- виділення та підсумок маркування особливих аспектів замовлення та підсумок, який може містити додаткову інформацію, яка спрощує роботу замовлення.

Модальне вікно деталі замовлення спроектоване для того, щоб надати користувачам максимально деталізовану інформацію та забезпечити зручний інтерфейс для управління та взаємодії з кожним конкретним замовленням.

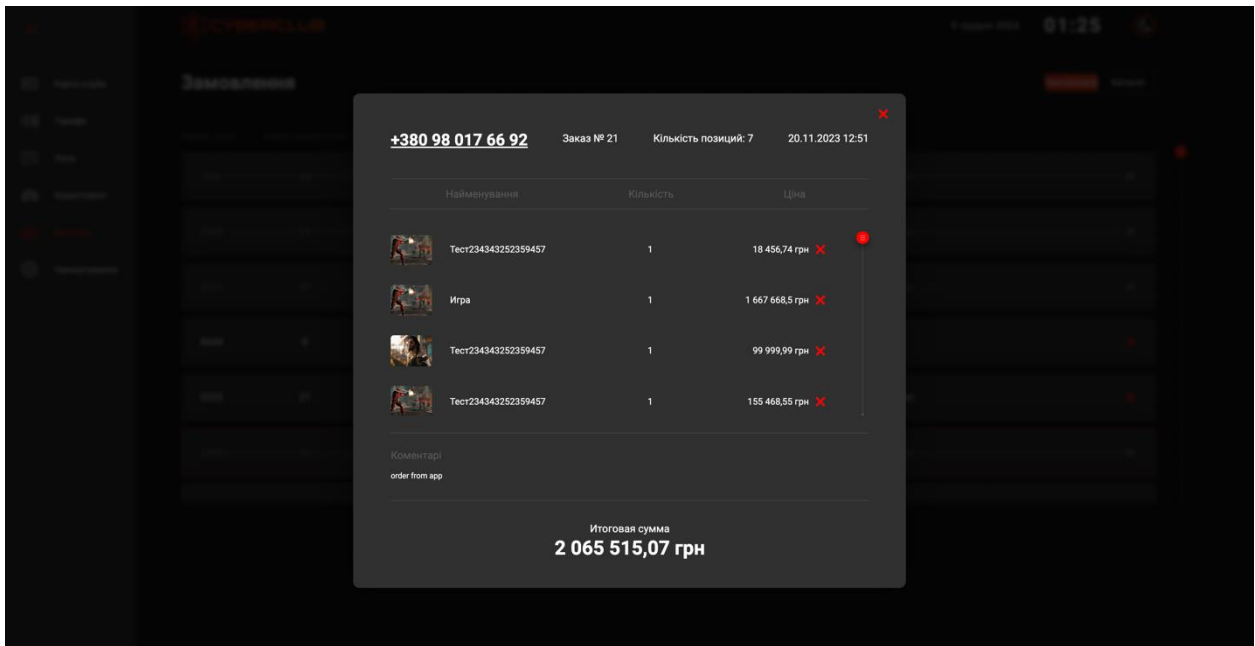


Рис. 3.20. модальне вікно замовлення

Світла тема веб-додатку кіберклубу (рис 3.21.) - це не тільки естетичний вибір, але й стратегічно обрана атмосфера, яка створює комфортну та дружню атмосферу для гравців та адміністраторів. Вона впливає на загальний вигляд і відчуття користувача, надаючи позитивний емоційний фон для кіберспортивного середовища.

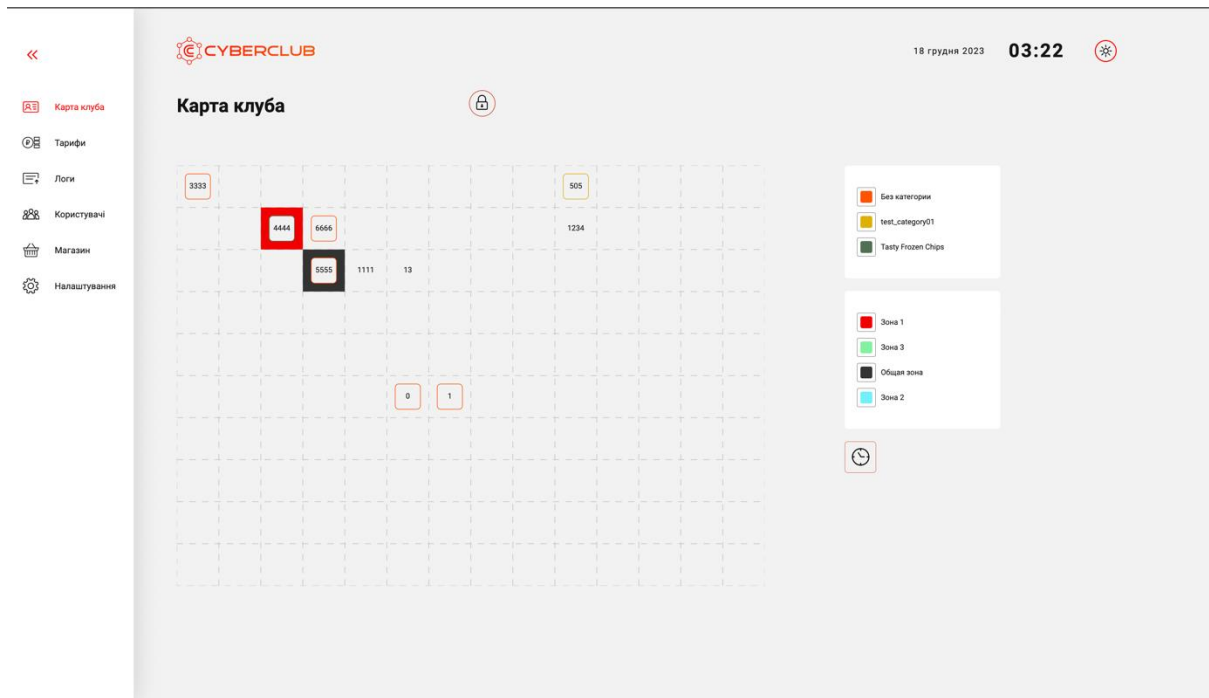


Рис. 3.21. світла тема веб-додатку

Вибір світлої теми для веб-додатку кіберклубу створює не лише зручне робоче середовище, але і сприяє позитивному враженню користувачів, допомагаючи відзначити його серед інших сучасних рішень у сфері кіберспорту.

3.6. Висновок

У третьому розділі роботи було ретельно розглянуто розробку веб-додатку, спрямованого на адміністрування кіберклубу. В процесі вивчення цього аспекту враховано використані технології, особливості кроссбраузерності, архітектури та інтерфейсу веб-додатку.

Технічна сторона веб-додатка отримала ретельний аналіз. Зокрема, в розробці використовувалися передові технології, такі як React, Next.js, TypeScript, Styled Components, SWR та sockets. Використання цих інструментів не лише забезпечило високий технічний рівень додатку, а й

забезпечило швидку та ефективну розробку з урахуванням сучасних стандартів веб-програмування.

Однією з ключових характеристик веб-додатку є його кроссбраузерність. Заходи, призначені для забезпечення коректної роботи на різних веб-переглядачах, враховувались та імплементувались з урахуванням різноманітних особливостей браузерів.

Архітектура додатку була ретельно спроектована з метою оптимізації продуктивності та можливості його масштабування в майбутньому. Це дозволило створити стабільний та надійний інструмент для адміністрування кіберклубу.

Ще однією важливою частиною розгляду був інтерфейс веб-додатку. Його користувацька дружба та ергономіка були високо оцінені, що вказує на забезпечення комфортного та ефективного використання адміністраторами.

Висновок підкреслює вдалу розробку веб-додатку адміністратора кіберклубу, високий рівень використаних технологій, кроссбраузерність, архітектурну стабільність та ефективний інтерфейс, що в сукупності гармонійно взаємодіють для досягнення поставлених цілей.

ВИСНОВКИ

Для веб-додатка адміністрування кіберклубу можна визначити ключові аспекти, які визначають його важливість та ефективність. Створення такого інструменту є актуальним у контексті швидкого технологічного розвитку.

В роботі використані технології, такі як React, Next.js, TypeScript, Styled Components, SWR та sockets. Ці інструменти дозволили створити не лише функціональний, але й технічно передовий інструмент для керування кіберклубом.

Додаток надає адміністраторам широкий спектр функцій для управління хостами, гравцями, ігровими сесіями та іншими важливими аспектами кіберспортивної діяльності. Інтуїтивний інтерфейс сприяє зручності використання, забезпечуючи швидкий доступ до необхідних функцій та інформації.

Додатково, розроблений інструмент дозволяє вести детальну статистику та аналіз, виходячи за межі простого управління. Це робить можливим прийняття обґрунтованих стратегічних рішень та оптимізацію діяльності кіберклубу.

Одним із ключових плюсів є впровадження заходів безпеки та конфіденційності, що гарантує захист важливих даних адміністраторів та гравців. Це стає додатковим стимулом для сприйняття веб-додатку як надійного та професійного інструменту.

В цілому, розроблений додаток не лише відповідає сучасним вимогам управління кіберклубом, але і встановлює новий стандарт ефективності та функціональності в цій сфері. Його використання призначено для спрощення рутинних операцій, покращення координації та підвищення загального рівня організації кіберспортивних подій.

СПИСОК ВИКОРАСТАНИХ ДЖЕРЕЛ

1. <https://react.dev/learn>
2. <https://nextjs.org/>
3. <https://learn.microsoft.com/ru-ru/aspnet/signalr/overview/getting-started/introduction-to-signalr>
4. <https://www.typescriptlang.org/>
5. https://www.moyo.ua/ua/news/istoriya_kybersporta_kak_prostye_st_relyalki_prevratilis_v_otdelnyy_vid_sporta.html
6. <https://smartlaunch.com/>
7. <https://antamediacafe.com/>
8. <https://www.handycafe.com/features.php>
9. <https://www.cybercafe.pro/features.php>
10. <https://senet.cloud/blog/gaming-lounge-business-plan-2023>
11. <https://web-developer.in.ua/assets/articles/react/react-introduction/react-introduction.html>
12. <https://brander.ua/technologies/nextjs>
13. <https://foxminded.ua/typescript/>
14. <https://uk.javascript.info/websocket>
15. <https://styled-components.com/>
16. <https://swr.vercel.app/>
17. <https://www.framer.com/motion/>
18. <https://www.hostzealot.com.ua/blog/how-to/shho-take-krosbrauzernist-i-yak-cyogo-domogtisya>
19. <https://www.dstu.dp.ua/Portal/Data/3/19/3-19-kl34.pdf>
20. <https://wezom.com.ua/ua/blog/kak-sozdat-veb-prilozhenie>
21. <https://blog.ithillel.ua/articles/web-application-architecture>
22. <https://dev.ua/news/chto-takoe-api-prostym-yazykom>

23. <https://www.w3.org/>
24. <https://medium.com/@natalia.grigorenko.05>
25. <https://nodejs.org/docs/latest/api/>

ЛІСТИНГ ПРОГРАМИ

```
// react
import type { ReactElement, ReactNode } from "react";
import React from "react";

// next
import type { NextPage } from "next";
import { AppProps } from "next/app";

// utils
import { Providers, shouldForwardProp } from "@/utils";

// lib
import { GlobalStyle } from "@/styles/global-styles";
import { StyleSheetManager } from "styled-components";
import { SWRConfig } from "swr";
import NextNProgress from "nextjs-progressbar";

// styles
import "@/styles/app.css";
import "react-datepicker/dist/react-datepicker.css";
import "react-toastify/dist/ReactToastify.css";

// contexts
import { AuthProvider, SidebarProvider } from "@/contexts";

// components
import { ProtectRoute, StyledToastContainer } from "@/components";

// core
import { fetcher } from "@/core/axios";

export type TNextPageWithLayout<Props = object> = NextPage<Props> & {
  getLayout?: (page: ReactElement) => ReactNode;
  requireAuth?: boolean;
};

type TAppPropsWithLayout = AppProps<{ fallback: object }> & {
  Component: TNextPageWithLayout;
};
```

```

export default function App({ Component, pageProps }: TAppPropsWithLayout) {
  const getLayout = Component.getLayout ?? ((page) => page);
  const { fallback } = pageProps;

  return (
    <AuthProvider>
      <StyleSheetManager shouldForwardProp={shouldForwardProp}>
        <Providers>
          <GlobalStyle />
          <SidebarProvider>
            <NextNProgress color={"#FF0000"} />
            <SWRConfig
              value={{
                fetcher,
                fallback: fallback || {},
                revalidateIfStale: false,
              }}
            >
              {Component.requireAuth ? (
                <ProtectRoute>
                  {getLayout(<Component {...pageProps} />)}
                </ProtectRoute>
              ) : (
                getLayout(<Component {...pageProps} />)
              )}
            </SWRConfig>
            <StyledToastContainer
              position="bottom-center"
              autoClose={10000}
              hideProgressBar={true}
              pauseOnHover={true}
              draggable={true}
              icon={() => null}
            />
          </SidebarProvider>
        </Providers>
      </StyleSheetManager>
    </AuthProvider>
  );
}

```

```

import { Html, Head, Main, NextScript } from "next/document";

```

```

export default function Document() {
  return (
    <Html lang="ru">
      <Head />
      <body>

```

```

    <Main />
    <div id="dialog" />
    <NextScript />
  </body>
</Html>
);
}

// components
import {
  Button,
  Input,
  PageLoader,
  PhoneNumberInput,
  Typography,
} from "@components";

// custom hooks
import { useSignInForm } from "./useSignInForm";

// constants
import { EMPTY_STRING } from "@constants";
import { Paths } from "@types";

// lib
import { Controller } from "react-hook-form";

// react
import React from "react";

// hoc
import { withTruthyAuthRedirect } from "@hoc";

// styles
import * as S from "./sign-in.styles";

export const SignIn = withTruthyAuthRedirect(() => {
  const { register, control, onSubmit, errors, isValid, loading } =
    useSignInForm();

  if (loading) return <PageLoader isBlackout />;

  return (
    <S.Overlay>
      <S.Wrapper>
        <S.Form noValidate onSubmit={onSubmit}>
          <Typography variant={"headline2"} className={"title"}>
            Авторизація
          </Typography>
        </S.Form>
      </S.Wrapper>
    </S.Overlay>
  );
});

```

```

</Typography>

<Controller
  name={"login"}
  control={control}
  render={({ field: { onChange, value } }) => {
    return (
      <PhoneNumberInput
        label={"Номер телефона"}
        isRequired
        value={value}
        onChange={onChange}
        specialLabel={EMPTY_STRING}
        autoFormat
        placeholder={"+380"}
        errorMessage={errors.login?.message}
        country={"uk"}
      />
    );
  }}
/>

<Input
  {...register("password")}
  type={"password"}
  label={"Пароль"}
  isRequired
  className={"password"}
  errorMessage={errors.password?.message}
/>

<Button className={"submitBtn"} disabled={!isValid || loading}>
  Увійти
</Button>

<S.AccessRecoveryLink href={Paths.AccessRecoveryCode}>
  Забули пароль?
</S.AccessRecoveryLink>
</S.Form>
</S.Wrapper>
</S.Overlay>
);
});

import { useCallback, useEffect, useMemo, useRef, useState } from "react";
import { Sheet, SheetEmpty } from "../sheet/sheet.component";
import { Grid } from "../grid/grid.component";
import { Space } from "../space/space.component";

```

```

import * as S from "./clubMap.styled";
import { Groups, Host, Zones } from "../organisms";
import { Button } from "./button";
// api
import {
  ISession,
  ISessionDeleted,
  ISheetWithSession,
  useMapConfig,
  usePlaySession,
  useSheets,
} from "../hooks/api/axios";
import { mutate } from "swr";
import { https } from "@core/axios";
import { ColorZoneCurrencySheetForm, HostForm } from "@components/modules";
import { useSheetContext } from "@contexts";
import { SignalRContext } from "@pages/club-map";
import { ResponseSocket } from "@types/socket.types";
import { useTheme } from "styled-components";
import { Dialog, TDialogRef } from "@components/molecules";

type ICoordination = {
  x: number;
  y: number;
};

const useSheetMapHook = () => {
  const { xMax } = useSheetContext();
  const { sheets } = useSheets();
  const { playSesssion } = usePlaySession();
  const sheetWithSessionMap = useMemo(() => {
    if (!sheets || !playSesssion) return new Map<string, ISheetWithSession>();

    const sessionMap = new Map<string, ISession>();
    const sheetWithSessionMap = new Map<string, ISheetWithSession>(new Map());

    playSesssion.result.forEach((session) =>
      sessionMap.set(session.user, session),
    );

    sheets.result.forEach(({ host, id, x, y, zone }) => {
      if (host) {
        const playSessionFound = sessionMap.get(host.userId) || null;

        sheetWithSessionMap.set(id, {
          id,
          zone,
          x: x * xMax,

```

```

    y: y * xMax,
    host: {
      ...host,
      play_session: playSessionFound,
    },
  });
} else {
  sheetWithSessionMap.set(id, {
    host,
    id,
    x: x * xMax,
    y: y * xMax,
    zone,
  });
}
});

return sheetWithSessionMap;
}, [playSesssion, sheets, xMax]);

return { sheets: sheetWithSessionMap };
};

export const ClubMap = () => {
  const dialog = useRef<TDialogRef>(null);
  const { text, colors } = useTheme();
  const { sheets } = useSheetMapHook();
  const { setSheet, sheet, xMax } = useSheetContext();
  const { mapConfig } = useMapConfig();
  const grid = useMemo(
    () => ({
      x_max: mapConfig?.result?.find(({ key }) => key === "x_max")?.value || 0,
      y_max: mapConfig?.result?.find(({ key }) => key === "y_max")?.value || 0,
    }),
    [mapConfig],
  );

  const [data, setData] = useState<Map<string, ISheetWithSession>>(new Map());

  const [currencyDrop, setCurrencyDrop] = useState<ICoordination>();
  const [isDragging, setIsDragging] = useState(false);
  const [draggedElement, setDraggedElement] = useState<ICoordination | null>(
    null,
  );
  const [offset, setOffset] = useState<ICoordination>({ x: 0, y: 0 });

  const [isLock, setIsLock] = useState<boolean>(true);
  const [isAdd, setIsAdd] = useState<boolean>(false);

```



```

const [isShowTime, isSetIsShowTime] = useState<boolean>(false);

const handleToggleLock = useCallback(() => {
  setIsLock((prev) => !prev);
  setSheet(null);
  setIsAdd(false);
  isSetIsShowTime(false);
}, [setSheet]);

const handleToggleAdd = useCallback(() => setIsAdd((prev) => !prev), []);
const handleShowTime = useCallback(
  () => isSetIsShowTime((prev) => !prev),
  [],
);

const handleActionLine = async (
  method: "POST" | "DELETE",
  direction: "right" | "down",
) => {
  try {
    await https("/line", { method, data: { direction } });
    Promise.all([mutate("/map-config"), mutate("/sheet")]);
  } catch (error) {
    dialog.current?.open();
  }
};

// evenets for D&D started

const handleMouseDown = useCallback(
  (item: ISheetWithSession) => (event: React.MouseEvent<SVGSVGElement>) => {
    event.preventDefault();
    setIsDragging(true);

    const svgRect = event.currentTarget.getBoundingClientRect();
    const x = event.clientX - svgRect.left;
    const y = event.clientY - svgRect.top;

    setOffset({ x, y });
    setDraggedElement({ x: item.x, y: item.y });
    setCurrencyDrop({
      x: item.x,
      y: item.y,
    });
    setSheet(item);
  },
  [setSheet],
);

```

```

const handleMouseMove = useCallback(
  (event: React.MouseEvent<SVGSVGElement>) => {
    event.preventDefault();
    if (isDragging && draggedElement && sheet) {
      const rect = event.currentTarget.getBoundingClientRect();
      const svgX = event.clientX - rect.left;
      const svgY = event.clientY - rect.top;

      const maxX = (grid.x_max + 1) * xMax - xMax;
      const maxY = (grid.y_max + 1) * xMax - xMax;

      const x = Math.min(Math.max(0, svgX - offset.x), maxX);
      const y = Math.min(Math.max(0, svgY - offset.y), maxY);

      setDraggedElement({ x, y });

      setCurrencyDrop({
        x: Math.floor((x + offset.x) / xMax) * xMax,
        y: Math.floor((y + offset.y) / xMax) * xMax,
      });

      setData((prevMap) => {
        const item = prevMap.get(sheet.id) as ISheetWithSession;
        return new Map(prevMap).set(sheet.id, { ...item, x, y });
      });
    }
  },
  [
    isDragging,
    draggedElement,
    sheet,
    grid.x_max,
    grid.y_max,
    xMax,
    offset.x,
    offset.y,
  ],
);

const handleMouseUp = useCallback(
  async (event: React.MouseEvent<SVGSVGElement>) => {
    event.preventDefault();
    if (!sheet) return;

    setIsDragging(false);

    if (draggedElement !== null) {

```

```

let sheetOnDrop: ISheetWithSession = {} as ISheetWithSession;

const x = Math.floor((draggedElement.x + offset.x) / xMax) * xMax;
const y = Math.floor((draggedElement.y + offset.y) / xMax) * xMax;

data.forEach((findSheet) => {
  if (findSheet.x === x && findSheet.y === y) {
    sheetOnDrop = findSheet;
  }
});

if (!sheetOnDrop.id)
  return setData((sheets) => {
    const mapSheets = new Map(sheets);
    mapSheets.set(sheet.id, { ...sheet });
    return mapSheets;
  });

setData((sheets) => {
  const mapSheets = new Map(sheets);

  mapSheets.set(sheet.id, {
    ...sheet,
    host: { ...sheetOnDrop.host },
    zone: { ...sheetOnDrop.zone },
  });

  mapSheets.set(sheetOnDrop.id, {
    ...sheetOnDrop,
    zone: { ...sheet.zone },
    host: { ...sheet.host },
  });

  return mapSheets;
});

setSheet({
  ...sheetOnDrop,
  zone: { ...sheet.zone },
  host: { ...sheet.host },
});

await https.patch(`/host-location/${sheet.host.id}`, {
  from: sheet.id,
  to: sheetOnDrop.id,
});
}

```

```

    setDraggedElement(null);
  },
  [data, draggedElement, offset.x, offset.y, setSheet, sheet, xMax],
);

// evenets for D&D ended

// evenets for sheets started

const handleClickSheet = useCallback(
  (sheet: ISheetWithSession) => () => setSheet(sheet),
  [setSheet],
);

// evenets for sheets ended

useEffect(() => setData(sheets), [sheets]);

const renderSheets = useCallback(() => {
  const elementJSX: JSX.Element[] = [];

  const eventsSheet = (sheet: ISheetWithSession) => {
    if (isLock)
      return {
        onClick: handleClickSheet(sheet),
      };

    return {
      onClick: handleClickSheet(sheet),
      onMouseDown: handleMouseDown(sheet),
    };
  };

  const eventsSheetEmpty = (sheet: ISheetWithSession) => {
    if (!isAdd) return {};

    return {
      onClick: handleClickSheet(sheet),
    };
  };

  data?.forEach(({ host, x, y, id, zone }) => {
    if (host) {
      elementJSX.push(
        <Sheet
          key={id}
          id={id}
          x={x}

```

```

    y={y}
    zone={zone}
    isTime={isShowTime}
    host={host}
    active={id === sheet?.id}
    size={xMax}
    {...eventsSheet({ host, x, y, id, zone })}
  />,
);
} else {
  elementJSX.push(
    <SheetEmpty
      size={xMax}
      key={id}
      host={host}
      x={x}
      y={y}
      id={id}
      zone={zone}
      isAdd={isAdd}
      isActive={id === sheet?.id}
      {...eventsSheetEmpty({ host, x, y, id, zone })}
    />,
  );
}
});

return elementJSX;
}, [
  data,
  handleMouseDown,
  handleOnClickSheet,
  isAdd,
  isLock,
  isShowTime,
  sheet,
  xMax,
]);

const socketPlaySessionCreated = useCallback(
  (playSessionCreated: ISession) => {
    setData((prev) => {
      return new Map(
        [...prev].map(([id, sheet]) => {
          if (sheet?.host?.id === playSessionCreated.host) {
            return [
              id,
              {

```

```

        ...sheet,
        host: { ...sheet.host, play_session: playSessionCreated },
    },
    ];
}

return [id, sheet];
}),
);
});
},
[],
);

```

```

const socketPlaySessionDeleted = useCallback(
(playSessionDeleted: ISessionDeleted) => {
  setData((prev) => {
    return new Map(
      [...prev].map(([id, sheet]) => {
        if (sheet?.host?.id === playSessionDeleted.host) {
          return [
            id,
            {
              ...sheet,
              host: { ...sheet.host, play_session: null },
            },
          ];
        }
      })

      return [id, sheet];
    }),
  );
});
},
[],
);

```

```

SignalRContext.useSignalREffect(
  "map",
  (data: ResponseSocket) => {
    if (data.event === "session-create")
      return socketPlaySessionCreated(data.object);
    if (data.event === "session-delete")
      return socketPlaySessionDeleted(data.object);
  },
  [],
);

```

```

return (
  <S.Main>
    <S.Content>
      <S.ButtonLock onClick={handleToggleLock}>
        {isLock ? (
          <svg
            width="42"
            height="42"
            viewBox="0 0 42 42"
            fill="none"
            xmlns="http://www.w3.org/2000/svg"
          >
            <rect
              x="0.5"
              y="0.5"
              width="41"
              height="41"
              rx="20.5"
              fill={colors.primaryBGC}
              stroke="#D23C1B"
            />
            <path
              d="M13.971 17.7548C13.971 17.3938 13.9723 17.0266 13.971 16.6597C13.9672 15.7883
13.9592 16.5168 13.9582 15.6453C13.9576 15.0987 13.9589 14.5522 13.9729 14.0056C14.0065 12.6981
14.3464 11.4834 15.0545 10.3832C15.7523 9.29854 16.715 8.54013 17.8636 8.00236C18.7569 7.58383
19.6915 7.37815 20.6727 7.40187C21.5951 7.42396 22.4792 7.62705 23.3288 7.99781C24.0707 8.32145
24.7257 8.77636 25.2855 9.35996C26.3696 10.4904 27.0744 11.8168 27.2109 13.4155C27.2329 13.6735
27.2329 13.9341 27.2332 14.1934C27.2351 15.8451 27.2361 15.8968 27.231 17.5481C27.2307 17.6953
27.2613 17.7558 27.4169 17.7665C27.9643 17.8032 28.4675 17.9673 28.8902 18.341C29.4864 18.8677
29.796 19.541 29.7976 20.3358C29.8037 23.7145 29.7966 25.4933 29.789 28.872C29.7861 30.1178
28.9991 31.0991 27.8051 31.3617C27.6764 31.3899 27.5419 31.4003 27.4099 31.4C23.772 31.3951
20.134 31.3873 16.4961 31.3805C15.5558 31.3786 14.6142 31.4059 13.6751 31.3669C12.8399 31.3321
12.1731 30.9454 11.7491 30.1857C11.6734 30.0499 11.6085 29.9072 11.5466 29.7643C11.4357 29.5085
11.3996 29.2431 11.3999 28.9613C11.4057 25.5066 11.406 23.6518 11.4012 20.197C11.4005 19.6394
11.6143 19.1683 11.9274 18.7397C12.422 18.0625 13.1019 17.7551 13.9275 17.7723C13.9371 17.7723
13.9467 17.7649 13.971 17.7548ZM20.5944 29.773C21.0449 29.773 21.4958 29.7734 21.9463
29.773C23.0841 29.7717 24.2222 29.7704 25.3599 29.7682C26.0376 29.7669 26.7156 29.774 27.3933
29.7578C27.8515 29.7467 28.1409 29.4513 28.186 28.9886C28.1994 28.8492 28.2058 28.7082 28.2058
28.5682C28.2071 25.8825 28.2068 24.7969 28.2068 22.1113C28.2068 21.5589 28.2087 21.0068 28.2061
20.4544C28.2045 20.1363 28.1346 19.8425 27.8825 19.6245C27.6668 19.438 27.4045 19.4022 27.1367
19.4019C22.7495 19.4006 18.3627 19.4 13.9755 19.4019C13.3674 19.4019 12.9971 19.7899 12.9971
20.4096C12.9971 23.7288 12.9971 25.4478 12.9971 28.767C12.9971 28.8102 12.9965 28.8538 12.9978
28.897C13.0096 29.4075 13.3144 29.7467 13.818 29.7597C14.5538 29.7786 15.2903 29.7724 16.0264
29.773C17.5492 29.7743 19.0713 29.773 20.5944 29.773ZM20.581 17.7756C21.029 17.7756 21.4772
17.7756 21.9252 17.7756C23.1119 17.7756 24.2988 17.7733 25.4855 17.7785C25.6219 17.7792 25.6561
17.7346 25.6529 17.603C25.6408 17.1218 25.6353 16.6406 25.6341 16.159C25.6312 15.309 25.6334
16.0589 25.6334 15.2089C25.6334 15.1601 25.6331 15.1114 25.6357 15.0627C25.6743 14.3484 25.6743
13.6368 25.505 12.9359C25.2056 11.6966 24.5142 10.7205 23.5096 9.98449C22.2837 9.08636 20.9076

```

```

8.83355 19.4397 9.16434C18.3512 9.40967 17.4789 10.0397 16.7402 10.8687C16.1408 11.5406 15.7465
12.3273 15.6366 13.2277C15.5602 13.8526 15.5516 14.4882 15.5443 15.1192C15.5328 16.0986 15.5481
15.4783 15.5516 16.458C15.5532 16.8531 15.5542 17.2482 15.5542 17.6433C15.5542 17.7187 15.5503
17.7727 15.6609 17.7723C16.4875 17.7684 17.3144 17.774 18.1409 17.7749C18.9541 17.7759 19.7679
17.7756 20.581 17.7756Z"
    fill={text.primary}
  />
  <path
    d="M19.6529 23.27C19.6529 23.038 19.649 22.8061 19.6537 22.5744C19.6634 22.1163
20.0337 21.8218 20.6198 21.8008C21.1168 21.7829 21.549 22.0851 21.5809 22.511C21.6102 22.9013
21.5852 23.2938 21.5961 23.6851C21.602 23.8964 21.5307 24.1206 21.7185 24.309C21.7867 24.3773
21.8592 24.4457 21.9442 24.4995C22.7205 24.9898 22.9216 26.4163 21.9157 27.0677C21.5287 27.3186
21.0626 27.4039 20.5715 27.3999C19.9175 27.3943 19.3532 27.218 18.9962 26.7665C18.426 26.0448
18.4147 25.2211 19.3349 24.5877C19.5563 24.4355 19.6447 24.2372 19.649 24.0121C19.6537 23.7646
19.6502 23.5172 19.6502 23.27C19.651 23.27 19.6517 23.27 19.6525 23.27H19.6529Z"
    fill={text.primary}
  />
</svg>
): (
<svg
  width="40"
  height="40"
  viewBox="0 0 40 40"
  fill="none"
  xmlns="http://www.w3.org/2000/svg"
>
  <rect
    width="40"
    height="40"
    rx="20"
    fill="url(#paint0_linear_926_2147)"
  />
  <path
    d="M12.971 16.7547C12.971 16.3937 12.9723 16.0265 12.971 15.6596C12.9672 14.7881
12.9592 15.5166 12.9582 14.6452C12.9576 14.0986 12.9589 13.5521 12.9729 13.0055C13.0065 11.6979
13.3464 10.4833 14.0545 9.38307C14.7523 8.29842 15.715 7.54001 16.8636 7.00223C17.7569 6.58371
18.6915 6.37802 19.6727 6.40174C20.5951 6.42384 21.4792 6.62693 22.3288 6.99768C23.0707 7.32133
23.7257 7.77624 24.2855 8.35983C25.3696 9.4903 26.0744 10.8167 26.2109 12.4154C26.2329 12.6734
26.2329 12.934 26.2332 13.1933C26.2351 14.845 26.2361 14.8967 26.231 16.548C26.2307 16.6952
26.2613 16.7556 26.4169 16.7664C26.9643 16.8031 27.4675 16.9672 27.8902 17.3409C28.4864 17.8676
28.796 18.5409 28.7976 19.3357C28.8037 22.7144 28.7966 24.4931 28.789 27.8719C28.7861 29.1177
27.9991 30.099 26.8051 30.3616C26.6764 30.3898 26.5419 30.4002 26.4099 30.3999C22.772 30.395
19.134 30.3872 15.4961 30.3804C14.5558 30.3784 13.6142 30.4057 12.6751 30.3667C11.8399 30.332
11.1731 29.9453 10.7491 29.1856C10.6734 29.0498 10.6085 28.9071 10.5466 28.7641C10.4357 28.5084
10.3996 28.2429 10.3999 27.9612C10.4057 24.5065 10.406 22.6517 10.4012 19.1969C10.4005 18.6393
10.6143 18.1682 10.9274 17.7396C11.422 17.0624 12.1019 16.755 12.9275 16.7722C12.9371 16.7722
12.9467 16.7647 12.971 16.7547ZM19.5944 28.7729C20.0449 28.7729 20.4958 28.7732 20.9463
28.7729C22.0841 28.7716 23.2222 28.7703 24.3599 28.768C25.0376 28.7667 25.7156 28.7739 26.3933

```



```

28.7576C26.8515 28.7466 27.1409 28.4512 27.186 27.9885C27.1994 27.8491 27.2058 27.7081 27.2058
27.568C27.2071 24.8824 27.2068 23.7968 27.2068 21.1111C27.2068 20.5587 27.2087 20.0067 27.2061
19.4543C27.2045 19.1362 27.1346 18.8424 26.8825 18.6244C26.6668 18.4379 26.4045 18.4021 26.1367
18.4018C21.7495 18.4005 17.3627 18.3998 12.9755 18.4018C12.3674 18.4018 11.9971 18.7898 11.9971
19.4094C11.9971 22.7287 11.9971 24.4476 11.9971 27.7669C11.9971 27.8101 11.9965 27.8537 11.9978
27.8969C12.0096 28.4074 12.3144 28.7466 12.818 28.7596C13.5538 28.7784 14.2903 28.7723 15.0264
28.7729C16.5492 28.7742 18.0713 28.7729 19.5944 28.7729ZM19.581 16.7755C20.029 16.7755 20.4772
16.7755 20.9252 16.7755C22.1119 16.7755 23.2988 16.7732 24.4855 16.7784C24.6219 16.779 24.6561
16.7345 24.6529 16.6029C24.6408 16.1217 24.6353 15.6405 24.6341 15.1589C24.6312 14.3088 24.6334
15.0588 24.6334 14.2088C24.6334 14.16 24.6331 14.1113 24.6357 14.0625C24.6743 13.3483 24.6743
12.6367 24.505 11.9358C24.2056 10.6965 23.5142 9.72036 22.5096 8.98437C21.2837 8.08623 19.9076
7.83343 18.4397 8.16422C17.3512 8.40955 16.4789 9.03961 15.7402 9.86853C15.1408 10.5405 14.7465
11.3272 14.6366 12.2276C14.5602 12.8525 14.5516 13.488 14.5443 14.1191C14.5328 15.0984 14.5481
14.4781 14.5516 15.4578C14.5532 15.853 14.5542 16.2481 14.5542 16.6432C14.5542 16.7186 14.5503
16.7725 14.6609 16.7722C15.4875 16.7683 16.3144 16.7738 17.1409 16.7748C17.9541 16.7758 18.7679
16.7755 19.581 16.7755Z"
    fill="white"
  />
  <path
    d="M18.6529 22.2698C18.6529 22.0378 18.649 21.8058 18.6537 21.5741C18.6634 21.1161
19.0337 20.8216 19.6198 20.8006C20.1168 20.7826 20.549 21.0848 20.5809 21.5107C20.6102 21.9011
20.5852 22.2936 20.5961 22.6849C20.602 22.8961 20.5307 23.1204 20.7185 23.3087C20.7867 23.3771
20.8592 23.4455 20.9442 23.4993C21.7205 23.9895 21.9216 25.4161 20.9157 26.0675C20.5287 26.3183
20.0626 26.4037 19.5715 26.3997C18.9175 26.3941 18.3532 26.2178 17.9962 25.7662C17.426 25.0446
17.4147 24.2209 18.3349 23.5874C18.5563 23.4353 18.6447 23.237 18.649 23.0118C18.6537 22.7644
18.6502 22.5169 18.6502 22.2698C18.651 22.2698 18.6517 22.2698 18.6525 22.2698H18.6529Z"
    fill="white"
  />
  <defs>
    <linearGradient
      id="paint0_linear_926_2147"
      x1="40"
      y1="20"
      x2="8.79967e-08"
      y2="20"
      gradientUnits="userSpaceOnUse"
    >
      <stop stopColor="#FE5301" />
      <stop offset="1" stopColor="#FF0000" />
    </linearGradient>
  </defs>
</svg>
)}
</S.ButtonLock>
{!isLock && (
  <
  <S.Buttons
    diraction="vertical"

```

```

    onAdd={() => handleActionLine("POST", "right")}
    onDelete={() => handleActionLine("DELETE", "right")}
    position="right"
  />
  <S.Buttons
    diraction="horizontal"
    onAdd={() => handleActionLine("POST", "down")}
    onDelete={() => handleActionLine("DELETE", "down")}
    position="bottom"
  />
</>
)}
<S.ContentScroll>
  <svg
    onMouseMove={handleMouseMove}
    onMouseUp={handleMouseUp}
    viewBox={`0 0 ${grid.x_max + 1} * xMax } ${
      (grid.y_max + 1) * xMax
    }` }
    width={(grid.x_max + 1) * xMax}
    height={(grid.y_max + 1) * xMax}
    fill="none"
    xmlns="http://www.w3.org/2000/svg"
  >
    <Space x_max={grid.x_max} y_max={grid.y_max} size={xMax} />
    <Grid {...grid} size={xMax} />
    {renderSheets()}
    {isDragging && (
      <rect
        x={currencyDrop?.x}
        y={currencyDrop?.y}
        opacity="0.05"
        width={xMax}
        height={xMax}
        fill="#FF0000"
      />
    )}
  </svg>
</S.ContentScroll>
</S.Content>
<S.Aside>
  {!sheet && (
    <>
      <Groups isEdit={!isLock} />
      <Zones isEdit={!isLock} />
    </>
  )}
</S.Aside>

```

```

    })

    {!isLock && (
      <
        {sheet?.host && <ColorZoneCurrencySheetForm />}
        {sheet && <HostForm />}
      </>
    )}

    {sheet?.host && <Host />}

    <S.ButtonsWapper>
      {!isLock && (
        <Button isActive={isAdd} icon="add" onClick={handleToggleAdd} />
      )}
      <Button isActive={isShowTime} icon="time" onClick={handleShowTime} />
    </S.ButtonsWapper>
  </S.Aside>
  <Dialog ref={dialog}>
    <S.DialogTitle variant="text1">
      Будь ласка, видаліть зайвий вміст, перш ніж зменшувати таблицю
    </S.DialogTitle>
    <S.DialogIcon>
      <svg
        width="64"
        height="64"
        viewBox="0 0 64 64"
        fill="none"
        xmlns="http://www.w3.org/2000/svg"
      >
        <path
          fill-rule="evenodd"
          clip-rule="evenodd"
          d="M44.8769 52.013C41.1651 54.4061 36.7447 55.7949 32 55.7949C27.0733 55.7949 22.4964
          54.2976 18.6994 51.7333L31.6483 38.7844L44.8769 52.013ZM51.3348 45.8725L37.9475
          32.4852L51.7333 18.6994C54.2976 22.4964 55.7949 27.0733 55.7949 32C55.7949 37.1766 54.1418
          41.9672 51.3348 45.8725ZM45.4594 12.3749L31.6483 26.186L18.1275 12.6652C22.0328 9.85818
          26.8234 8.20513 32 8.20513C36.9955 8.20513 41.6315 9.74454 45.4594 12.3749ZM11.987
          19.1231C9.59386 22.8349 8.20513 27.2553 8.20513 32C8.20513 36.9955 9.74454 41.6315 12.3749
          45.4594L25.3491 32.4852L11.987 19.1231ZM64 32C64 49.6731 49.6731 64 32 64C14.3269 64 0
          49.6731 0 32C0 14.3269 14.3269 0 32 0C49.6731 0 64 14.3269 64 32Z"
          fill="url(#paint0_linear_1689_15549)"
        />
        <defs>
          <linearGradient
            id="paint0_linear_1689_15549"
            x1="64"
            y1="32"

```

```
x2="1.40795e-07"  
y2="32"  
gradientUnits="userSpaceOnUse"  
>  
  <stop stop-color="#FE5301" />  
  <stop offset="1" stop-color="#FF0000" />  
</linearGradient>  
</defs>  
</svg>  
</S.DialogIcon>  
</Dialog>  
</S.Main>  
};
```

Всі інші файли з програмним кодом можна буде переглянути на магнітному носії

ДОДАТОК Б

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Кваліфікаційна робота Сергієнка М.І.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна робота Сергієнка М.І.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Сергієнка М.І.rar	Архів. Містить коди програми і скомпільовану програму
Презентація	
Сергієнко М.І.ppt	Презентація кваліфікаційної роботи