

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

Інститут електроенергетики

(навчально-науковий інститут)

факультет інформаційних технологій

(факультет)

Кафедра інформаційних технологій та комп'ютерної інженерії

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня бакалавра

(бакалавра, магістра)

Здобувача вищої освіти Чабаненка Євгена Вікторовича

(ПІБ)

академічної групи 126-21-2

(шифр) спеціальності 126 «Інформаційні системи та технології»

(код і назва спеціальності)

спеціалізації за освітньо-професійною (освітньо-науковою) програмою _____

(за наявності)

(офіційна назва) на тему Розробка вебзастосунку моніторингу активності користувачів ігрового серверу

(назва за наказом ректора)

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	доц. Соколова Н. О.			
розділів:				

Рецензент				
-----------	--	--	--	--

Нормоконтролер	д.т.н., проф. Коротенко Г.М.			
----------------	---------------------------------	--	--	--

Дніпро
2025

ЗАТВЕРДЖЕНО:
завідувач кафедри
інформаційних технологій
та комп'ютерної інженерії
(повна назва)

_____ В.В. Гнатушенко
(підпис) (ініціали та прізвище)

«_____» 2025 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня бакалавра
(бакалавра, магістра)

здобувача вищої освіти Чабаненко Є.В. академічної групи 126-21-2
(прізвище та ініціали) (шифр)
спеціальності 126 «Інформаційні системи та технології»
спеціалізації за освітньою-професійною програмою _____
(за наявності)

на тему Розробка вебзастосунку моніторингу активності користувачів ігрового серверу
затверджену наказом ректора НТУ «Дніпровська політехніка» від 05.05.2025 № 336-с

Розділ	Зміст	Термін виконання
Розділ 1. Аналіз стану області рішення завдання	Обґрунтування вибіру технології для розробки веб додатку.	21.04.2025– 04.05.2025
Розділ 2. Проектні рішення	Основні завдання, які повинен вирішувати сайт і описуються основні кроки створення дизайну та верстки програми.	05.05.2025– 03.06.2025

Завдання видано _____
(підпис керівника) (ініціали та прізвище)

Дата видачі 03.02.2025

Дата подання до екзаменаційної комісії _____

Прийнято до виконання _____ Чабаненко Є.В.
(підпис здобувача вищої освіти) (ініціали та прізвище)

РЕФЕРАТ

Пояснювальна записка: містить 87 сторінок тексту, рисунків - 55, таблиць - 4, використаних джерел - 23.

ВЕБСТОРІНКА, РИСУНОК, САЙТ, АДМІНІСТРАТОР, ДАНІ, КОРИСТУВАЧ, ЛЕДЕР, ЗАСТУПНИК, ТАБЛИЦЯ, ДОСТУП, РОЛЬ.

Об'єкт кваліфікаційної роботи: Процес адміністрування та контролю активності керівного складу (лідерів і заступників) на ігровому сервері багатокористувацької рольової онлайн-гри «Arizona RP».

Предмет кваліфікаційної роботи: система моніторингу активності користувачів, зокрема лідерів та їхніх заступників.

Мета роботи: створення вебпрограми для моніторингу активності користувачів ігрового сервера, що дозволяє спростити роботу адміністратора та забезпечити його надійним та зручним місцем зберігання даних.

У вступі наведено інформацію про стан проблеми, актуальність теми дослідження та методи її вирішення, а також можливі результати дослідження.

У першому розділі було закладено міцну теоретичну та технічну основу для створення вебзастосунку. Детальне дослідження предметної області дозволило визначити актуальні проблеми, сформулювати вимоги до майбутньої системи та обрати доцільні інструменти й технології для її реалізації.

Другий розділ продемонстрував, як ці теоретичні знання було застосовано на практиці під час розробки функціонального вебзастосунку для автоматизації адміністративної діяльності на ігровому сервері «Arizona RP». Запропоновані рішення орієнтовані на підвищення ефективності керування, зниження навантаження на адміністративний персонал та забезпечення зручності взаємодії з системою для всіх категорій користувачів.

ABSTRACT

Explanatory note: contains 87 pages of text, 55 figures, 4 tables, 23 sources used.

WEBPAGE, FIGURE, SITE, ADMINISTRATOR, DATA, USER, LEADER, DEPUTY, TABLE, ACCESS, ROLE.

Object of qualification work: The process of administration and control of the activity of the management team (leaders and deputies) on the game server of the multiplayer online role-playing game "Arizona RP".

Subject of qualification work: a system for monitoring the activity of users, in particular leaders and their deputies.

Purpose of work: creation of a web program for monitoring the activity of users of the game server, which allows to simplify the work of the administrator and provide him with a reliable and convenient place to store data.

The introduction provides information on the state of the problem, the relevance of the research topic and methods for its solution, as well as possible results of the research.

The first section laid a solid theoretical and technical foundation for creating a web application. A detailed study of the subject area allowed us to identify current problems, formulate requirements for the future system, and select appropriate tools and technologies for its implementation.

The second section demonstrated how this theoretical knowledge was applied in practice during the development of a functional web application to automate administrative activities on the Arizona RP game server. The proposed solutions are aimed at increasing management efficiency, reducing the burden on administrative staff, and ensuring the convenience of interacting with the system for all categories of users.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, СКОРОЧЕНЬ І ТЕРМІНІВ

SAMP - San Andreas Multiplayer;

Arizona RP - мультиплеєрний сервер з елементами рольової гри;

Лідер - керівник фракції в грі;

Заступник - помічник лідера;

Black List (ЧС) - перелік гравців, які порушили правила;

Фракція - внутрішньоігрова організація (МВС, Армія, мафія тощо).

БД - База даних

Статік(Statik) - це постійний ідентифікатор гравця, який закріплений за певним обліковим записом

СУБД - Система управління базами даних.

Роль - набір прав, які визначають, що конкретний користувач може робити на вебсайті.

Ник(Никнейм) - Це ім'я вашого ігрового персонажа.

ролеплей(RP) - елементи рольової гри

UX (User Experience) - досвід користувача

UI (User Interface) - зовнішній вигляд (кольори, кнопки, іконки).

Профіль - персональна сторінка користувача

Гільдії - організовані спільноти гравців у багатокористувацьких онлайн-іграх

Онлайн - це загальна кількість часу, який гравець провів у грі на сервері

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАВДАННЯ	11
1.1 Аналіз подібних рішень	12
1.2 Сучасні технології розробки вебдодатків	13
1.3 Огляд бібліотек і сервісів для розробки вебсайтів	16
1.4 Вибір інструментів для створення вебпрограми	21
1.5 Огляд сучасних інструментів та фреймворків для створення вебзастосунків	22
1.6 Основні етапи розробки вебзастосунку. Визначення завдань і створення моделі вебзастосунку.	23
1.7 Висновки з першого розділу	26
РОЗДІЛ 2. ПРОЄКТНІ РІШЕННЯ	28
2.1 Розробка прототипу та визначення основного функціоналу	28
2.1.1 Створення макету спільних елементів вебзастосунку	33
2.2 Створення дизайну вебсайту та верстка	37
2.2.1 Головна сторінка сайту, с статистикою лідерів та заступників	42
2.2.2 Бічна панель для зручного переходу між сторінками	46
2.2.3 Сторінка профілю лідера чи заступника з докладною інформацією	49
2.2.4 Сторінка додавання лідера/заступника	56
2.2.5 Сторінка архіву лідерів/заступників з інформацією коли і хто зняв лідера/заступника	58

2.2.6	Сторінка Чорного списку лідерів/заступників	62
2.2.7	Сторінка Архіву чорного списку лідерів/заступників	64
2.2.8	Сторінка управління правами користувачів на сайті	65
2.2.9	Сторінка власного профілю користувача	66
2.2.10	Сторінка магазину з зняттям доганів та попередження	67
2.2.11	Створення скрипту для автоматичного підрахунку онлайн	71
2.3	Розробка баз даних	72
2.4	Висновки з другого розділу	74
	ВИСНОВКИ	76
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	79
	ДОДАТОК А	81

ВСТУП

У сучасних багатокористувацьких іграх дедалі важливішим стає ефективне адміністрування внутрішніх процесів, особливо дану тенденцію можна помітити на великих ігрових серверах із великою кількістю користувачів. Тому, обрана тема кваліфікаційної роботи є актуальною на теперішній час, бо попит на вебсайти зростає з геометричною прогресією, а розробка вебзастосунку є не тільки технічним завданням, а ще пріоритетом для будь-якого успішного багатокористувацького ігрового проєкту.

Одним із таких проєктів є сервер Arizona RP, що працює на платформі San Andreas Multiplayer (SAMP). У рамках цього серверу реалізована система фракцій, де гравці можуть обіймати керівні посади (лідери, заступники), а адміністрація виконує функції контролю за їхньою активністю та відповідністю до встановлених правил.

Протягом трьох років існування ігрового серверу всі адміністративні процеси виконувалися вручну, а дані зберігалися в електронних таблицях. Це призводило до ряду труднощів: ускладнене сортування, відсутність швидкого пошуку, неможливість спільного доступу з обмеженими правами, а також витрати великої кількості часу на рутинні перевірки. Відсутність централізованої системи обліку створювала передумови для помилок, дублювання інформації або її втрати.

Обґрунтування вибору теми та її актуальність

На підставі викладеного матеріалу тема кваліфікаційної роботи актуальна. Адже розробка інструментів для автоматизації контролю за лідерами фракцій не просто оптимізує процеси, а й безпосередньо сприяє створенню більш стабільної та безпечної ігрової екосистеми, що є життєво важливою для успіху будь-якого багатокористувацького ігрового проєкту.

У процесі контролю за лідерами фракцій модератор стикається з великою кількістю рутинних завдань, а саме: підрахунок щоденного часу перебування в мережі конкретного лідера чи заступника; перевірка статистики прийому та звільнення з фракцій; контроль за балансом фракцій; ведення системи покарань (Black List).

Отримання вищезгаданих даних вручну займає багато часу та є неефективним. Тому обрана тема кваліфікаційної роботи дає у створенні вебпрограми, яка дозволить автоматизувати ці процеси та надати зручний інтерфейс для доступу до структурованих даних.

Об'єкт дослідження

Ігровий сервер Arizona RP платформи SAMP.

Предмет дослідження

Система моніторингу активності користувачів, зокрема лідерів та їхніх заступників.

Мета дослідження

Створення вебзастосунку для моніторингу активності користувачів серверу, що дозволяє автоматизувати перевірку фракцій, структурувати дані та забезпечити багаторівневий контроль доступу.

Основні завдання дослідження:

1. Аналіз існуючих аналогів.
2. Вибір технологій реалізації вебзастосунку.
3. Проектування структури БД та інтерфейсу.
4. Розробка системи багаторівневого доступу.
5. Реалізація функціоналу для перегляду активності, анкет, архіву та Black List.
6. Тестування розробленої системи.
7. Визначення необхідного набору функцій.

Методи дослідження

В даній кваліфікаційній роботі використовуються методи аналізу, порівняння, логічного синтезу, структурного та об'єктно-орієнтованого програмування.

Користувачі системи:

Старша адміністрація (куратори, Головний адміністратор та Заступник Головного адміністратора) - повний доступ до всіх даних і можливість редагування.

Модератори (Адміністратори які стежать за організаціями) - доступ до анкет лідерів, статистики, Black List.

Лідери та заступники - обмежений доступ до перегляду власного профілю та активності.

Теоретична та практична значущість роботи

Розроблений вебзастосунок буде використано в адміністративних цілях старшою адміністрацією серверу. Він дозволяє оптимізувати та автоматизувати рутинні процеси, покращити контроль за фракціями та забезпечити надійне збереження інформації. Це має позитивний вплив на ефективність управління віртуальною організаційною структурою.

Очікуваний результат

Результатом дослідження стане розроблена вебпрограма, що відповідає всім функціональним вимогам, має зручний інтерфейс та забезпечує швидкий доступ до потрібної інформації залежно від ролі користувача.

РОЗДІЛ 1

АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАВДАННЯ

1.1 Аналіз подібних рішень

Моніторинг активності гравців є важливою складовою адміністрування багатокористувацьких ігрових серверів, особливо в проєктах з розвиненою внутрішньою ієрархією, як-от рольові фракції, гільдії, команди. У таких середовищах ефективність ігрового процесу значною мірою залежить від дисципліни учасників, прозорості звітності та можливості адміністрації контролювати ключових гравців, зокрема лідерів і їхніх заступників.

Сучасні ігрові спільноти дедалі частіше використовують вебзастосунки або аналітичні панелі для збору статистики та управління користувачами.

Більшість таких рішень мають схожі риси:

- використання бази даних для зберігання інформації про користувачів;
- наявність авторизації;
- облік активності за періодами (день, тиждень, місяць);
- розмежування прав доступу (звичайний користувач, лідер, адміністратор);
- зручний графічний інтерфейс.

Нижче наведено приклади реалізації подібних систем у популярних ігрових проєктах:

1. Minecraft (наприклад, MineStat, Plan Player Analytics)

У грі Minecraft використовуються плагіни, такі як Plan (Player Analytics), які генерують детальні вебінтерфейси з інформацією про гравців: кількість годин у грі, активність за днями, країна користувача, IP-адреса, історія входів. Дані подаються у вигляді діаграм, таблиць і графіків. Вебінтерфейс є загальнодоступним або обмеженим за роллю.

2. Rust (наприклад, BattleMetrics)

BattleMetrics - сторонній сервіс, який надає вебмоніторинг для серверів Rust, ARMA, Ark тощо. Він дозволяє відслідковувати, скільки часу гравець провів у грі, коли останній раз був онлайн, які дії виконував, та дозволяє створювати фільтри для сповіщень (наприклад, про вхід певного гравця), [1].

3. ARMA 3 (наприклад, ArmA3Sync + вебпанелі спільнот)

У багатьох ролеплей-проектах на базі ARMA 3 використовуються власні панелі керування підрозділами, що інтегруються з серверами. Вони дають змогу відстежувати виконання завдань, ролі гравців у підрозділах, а також формувати звітність по місіях.

4. World of Warcraft (WoW Armory / Raider.IO)

Raider.IO та WoW Armory - приклади глобального моніторингу, що включають рейтинг гравців, активність у підземеллях, етапи прокачки персонажів. Ці сервіси мають публічні API, через які можна створювати власні панелі лідерів або рейтинги гільдій, [2].

Таб.1.1 Порівняльний аналіз

Платформа	API/Плагін	Рівень аналітики	Гнучкість під адаптацію	Призначення
Plan (Minecraft)	Плагін	Високий	Середня	Активність гравців
BattleMetrics	Сторонній API	Високий	Висока	Адміністрування серверів
ARMA 3 панелі	Кастомні	Високий	Висока	Рольова структура
Raider.IO	API	Високий	Низька	Глобальна статистика

Хоча більшість з описаних рішень реалізовані на потужних платформах або у вигляді плагінів до ігрових клієнтів, вони об'єднані спільною метою - надання структурованої, наочної інформації про гравців.

Жодне з наведених рішень не адаптоване під специфіку серверу Arizona RP на платформі SAMP, який має унікальну внутрішньоігрову ієрархію, ролі, правила та фракції.

Для забезпечення точного, зручного та безпечного контролю за лідерами фракцій необхідно створити власну систему моніторингу, яка враховуватиме як структуру ігрового процесу, так і організаційні потреби адміністрації.

Саме це стало передумовою до розробки індивідуального вебзастосунку, що стане ефективним інструментом автоматизації адміністративних процесів на сервері.

1.2 Сучасні технології розробки вебдодатків

Сучасні технології розробки вебдодатків це великий спектр інструментів, фрейворків, методологій. Тому, це не просто написання коду, а створення гнучких та масштабованих вебсайтів, які відповідають високим критеріям сучасності. Саме технології дозволяють створювати динамічні платформи, які адаптуються до вимог користувачів. Отже, саме мови програмування дають комп'ютеру "інтелект" і дозволяють виконувати мільярди операцій за секунду, перетворюючи його на найпотужніший інструмент сучасності. В кваліфікаційній роботі наведено найпоширеніші технології в сьогоденні, адже саме вони є ключовим інструментом для створення та розробки вебдодатків.

Python - це мова програмування, яка часто використовується в вебдодатках, в тому числі при розробці програмного забезпечення, в науці про дані та навчанні нейронних мереж. Розробники віддають перевагу цій мові, оскільки вона дозволяє збільшити швидкість написання, простоту

читання та якість коду. Кожен елемент мови - це об'єкт, що робить її повністю об'єктно-орієнтованою, а виділення блоків коду здійснюється через пробіли.

Синтаксис мови простий у використанні та вивченні, завдяки чому вона популярна серед початківців програмістів. Python використовується для написання скриптів, має велику кількість різноманітних бібліотек і підтримує методи асинхронного програмування. При написанні цієї роботи був використаний інший, але не менш функціональний мова програмування PHP, [3].

JavaScript домінує у веббраузерах, адже це не просто сервіс, а повноцінна мова програмування, яка є основною технологією веброзробки поряд з HTML і CSS. Ця мова програмування має всі атрибути повноцінної мови: змінні, типи даних, функції, об'єкти, класи, успадкування, цикли, умовні оператори, механізми обробки помилок, асинхронність тощо.

Підсумовуючи, можна дійти висновку, що JavaScript є основою інтерактивності будь-якого сучасного вебсайту, адже

- роль JavaScript у клієнтській логіці (динамічне оновлення інтерфейсу, валідація форм);
- взаємодію з DOM (зміна атрибутів, зміна вмісту);
- AJAX-запити (асинхронність даних на сервер);
- застосування JavaScript у фреймворках (React, Vue, Angular),
- роботу з подіями, [4].

Згідно з статистикою, саме PHP є найпоширенішою мовою програмування загального призначення з відкритим початковим кодом. PHP сконструйована спеціально для ведення веброзробок, а її код може впроваджуватися безпосередньо в HTML. PHP має синтаксис, схожий на мову програмування C, і підтримується більшістю хостинг-провайдерів. Це робить PHP надзвичайно потужним і гнучким інструментом для створення динамічних вебсайтів. Хоча сучасні фреймворки часто відходять від прямого

вбудовування PHP в HTML на користь шаблонних рушіїв та MVC-архітектури, основна ідея генерації динамічного HTML на стороні сервера залишається центральною для PHP. Це робить його одним з найефективніших інструментів для швидкої та масштабованої веброзробки, [5].

CMS - це конструктори сайтів, які дозволяють створювати сайт із заготовлених блоків, що не дає можливості додати будь-який код від себе. В основному підходить для створення простих інформаційних сайтів-візиток, [6].

При створенні динамічних вебдодатків часто виникає необхідність у зберіганні даних. Для цього використовуються бази даних, які зв'язуються з додатком за допомогою мови програмування SQL.

SQL - це стандартна мова запитів для роботи з реляційними базами даних. SQL дозволяє легко отримувати доступ до потрібної інформації та виконувати складні аналітичні запити. Завдяки їм можна вставляти, оновлювати та видаляти інформацію в БД також ефективно працювати з великими обсягами даних і використовується в різних галузях, включаючи веброзробку, бізнес-аналіз та науку про дані.

У цій роботі ми використовуємо вільну реляційну систему управління базами даних (СУБД) MySQL.

Розглянемо її основні переваги:

- Швидкість роботи.
- Безкоштовна ліцензія.
- Підтримка кількох типів таблиць: MyISAM, InnoDB.
- Можливість створення плагінів.
- Надійна система безпеки.
- Таблиці дозволяють зберігати кілька мільйонів записів.
- Вебдодаток "phpMyAdmin". На рисунку 1.1. наглядно наведено приклад інтерфейсу цього вебдодатку.

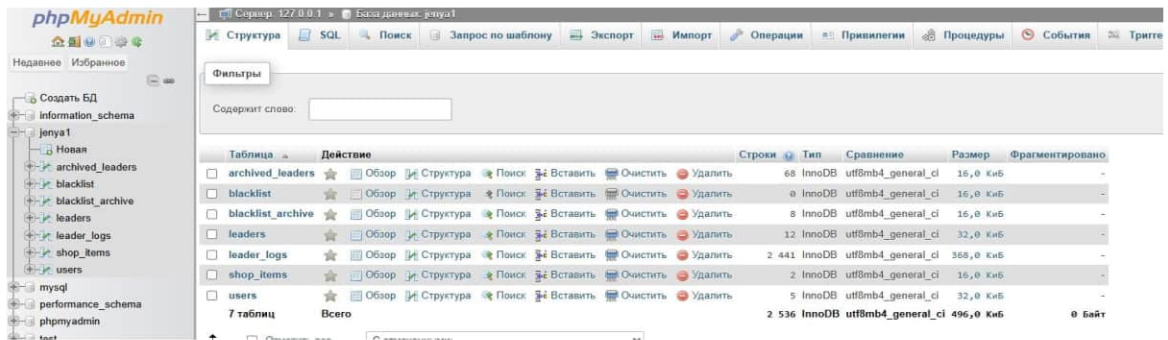


Рис. 1.1.- Интерфейс "phpMyAdmin"

"phpMyAdmin" - це засіб, який можна встановити на сервер з СУБД MySQL. За допомогою нього можна змінювати та візуалізувати дані, використовуючи спеціальний вебсайт, [7].

1.3 Огляд бібліотек і сервісів для розробки вебсайтів

Під час розробки вебзастосунків часто виникають задачі, більшість з яких уже колись були вирішені. Часто такі рішення оформлюються у вигляді спеціальних бібліотек, якими може скористатися кожен охочий. Найчастіше ці бібліотеки мають відкритий код. В даній роботі наведено бібліотеки, які були використані під час створення вебзастосунку в практичній частині кваліфікаційної роботи.

«jQuery» - це набір функцій мовою JavaScript, який використовується для взаємодії з HTML. Ця бібліотека надає багато корисних функцій для роботи з елементами DOM, дозволяючи легко звертатися до атрибутів та вмісту елементів. «jQuery» також містить зручний API для роботи з AJAX, [8].

«jQuery UI» (Користувацький інтерфейс) - це готові до використання елементи, створені на основі бібліотеки «jQuery» JavaScript, які можна додавати на вебсторінки, такі як кнопки, елементи форм, випадаючі списки.

Вони виконують роль "прикраси" та покращують взаємодію користувача з сайтом [9]. Приклад інтерфейсу «jQuery» наведено на рисунку 1.2.

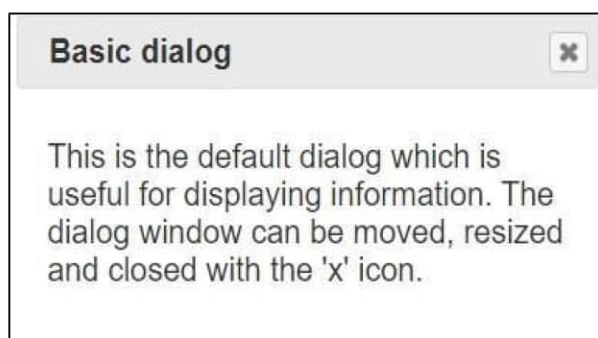


Рис. 1.2. — Приклад діалогового вікна, створеного за допомогою «jQuery UI»

Bootstrap 5.3 - це остання стабільна версія одного з найпопулярніших фреймворків для розробки вебінтерфейсів. Його головна мета - пришвидшити створення адаптивних, зручних для користувача інтерфейсів з використанням готових компонентів та сіткової системи.

Завдяки широкій функціональності, постійним оновленням та активній спільноті розробників, Bootstrap 5.3 не лише зберігає свою актуальність, а й впевнено адаптується до сучасних вимог веброботки.

Покращена структура CSS і JS дозволяє швидше завантажувати сторінки, що особливо важливо для мобільних користувачів.

Адаптивний дизайн - Bootstrap побудований на системі "мобільний спочатку" (mobile-first), що дозволяє сайтам добре виглядати на будь-яких пристроях: від смартфона до десктопа .

Темна тема (Dark Mode) - У версії 5.3 з'явилася підтримка темної теми за допомогою CSS-змінних. Можна легко перемикатися між світлим та темним інтерфейсом.

Покращена документація - Bootstrap має чудову офіційну документацію з прикладами коду, що спрощує навчання та використання фреймворку, [10].

Отже, Bootstrap 5.3 продовжує бути лідером серед CSS-фреймворків, пропонуючи збалансоване поєднання простоти використання, гнучкості, продуктивності та можливості створювати сучасні, адаптивні та візуально привабливі вебінтерфейси з високою швидкістю. Він ідеально підходить як для швидкого прототипування, так і для розробки великих проєктів.

Google Fonts - це безкоштовний сервіс від компанії Google, який надає доступ до великої колекції вебшрифтів з відкритою ліцензією. Його основна мета - полегшити використання красивих та сучасних шрифтів у веброботці без потреби в ручному завантаженні файлів.

- Оптимізація швидкості. Шрифти завантажуються з розподіленої мережі серверів Google (CDN), що забезпечує швидке завантаження на будь-якому континенті.
- Адаптація під мови та стилі. Google Fonts підтримує десятки мов (у тому числі українську) та різноманітні начертання: тонкий (thin), звичайний (regular), напівжирний (medium), жирний (bold), курсив тощо.
- Інструменти попереднього перегляду. На сайті fonts.google.com можна одразу переглядати текст зі шрифтами, комбінувати їх і копіювати код для вбудовування, [11].

Будь-який сервер, що зберігає файли застосунку, може піддаватися DDoS-атакам. Для захисту застосунку існує безліч способів та сервісів. Особливо на теперішній час захист застосунків є критично важливим фактором в Україні. Адже з початку повномасштабного вторгнення Україна перебуває під постійними та інтенсивними кібератаками.

Отже, захист вебзастосунку є важливим при розробці, адже це безпека, стабільність та підтримка суспільства.

Cloudflare - слугує потужним захисним бар'єром, поглинаючи основний удар DDoS-атак ще до того, як вони досягнуть серверів. Його глобальна мережа та інтелектуальні системи фільтрації є надзвичайно ефективними проти атак, що походять з різних джерел.

«CloudFlare» - це американська компанія, яка надає послуги CDN (мережа доставки контенту), захисту від DDoS-атак і безпечного доступу до ресурсів. Основною перевагою для початківців-розробників є повністю безкоштовне підключення доданих функцій. Сервіс дозволяє в автоматичному режимі відбивати атаки, а також вручну створювати різні фільтри. Наприклад, у моніторингу можна помітити велику кількість запитів з Китаю, у той час як сайт належить до сфери доставки їжі в Дніпрі. У такому випадку можна налаштувати фільтр на користувачів із певного регіону або країни чи застосувати інші правила. При спрацюванні фільтра можна вибрати дію: повністю заблокувати доступ або попросити користувача пройти перевірку на робота. Також система самостійно може ініціювати перевірку користувача, якщо її алгоритми виявлять підозрілу активність, [12].

Google reCaptcha v3 - система для захисту вебсайтів від спам-ботів. Третя версія сервісу працює у фоновому режимі на сторінці вона аналізує поведінку користувача та повертає оцінку, від значення якої залежить відображення вікна перевірки користувача робота (рис. 1.3.), [13].

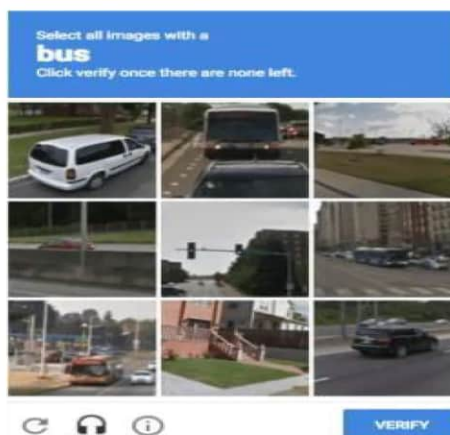


Рис. 1.4. - Приклад роботи системи Google ReCaptcha v3, [13].

Отже, саме Google reCaptcha v3 допомагає боротися з автоматизованими ботами, які можуть використовуватися для спаму, брутфорсу облікових записів або інших зловмисних дій на рівні застосунку. Це особливо важливо для захисту реєстрацій, форм зворотного зв'язку та інших інтерактивних елементів, які можуть бути атаковані ботами.

Розробники мають гнучку настройку і можуть визначати дії кожної оцінки залежно від своїх потреб. Наявність системи на сторінці можна визначити за логотипом у нижньому правому кутку

1.4 Вибір інструментів для створення вебпрограми

Розглянемо програмне забезпечення, необхідне для створення вебсайту. Для написання будь-якого коду необхідні текстові редактори. Існує велика кількість різних середовищ, призначених для написання коду. Розглянемо одне з них.

Notepad++ - універсальний інструмент розробки, що підтримує багато мов програмування (PHP, CSS, HTML, JS та інші).

Основні переваги

- Заміна фрагментів тексту за результатами пошуку вибірково або в усьому файлі
- Зручний пошук з використанням регулярних виразів
- Можливість встановлення додаткових плагінів
- Зручний механізм згортання блоків
- Підсвічування синтаксису

Для створення сайту необхідно розробити дизайн, а також мати логотип та інші необхідні зображення. Без продуманого дизайну та візуального контенту, навіть найфункціональніший сайт може виглядати непрофесійно та відштовхувати користувачів. Для вирішення цього завдання потрібні графічні редактори. Найпростішим з них є Paint. Для створення складних дизайнів та логотипів часто використовуються Adobe Photoshop.

Якщо виникає потреба створювати файли з векторною графікою, ідеально підходять Inkscape та Adobe Illustrator, [14].

1.5 Огляд сучасних інструментів та фреймворків для створення вебзастосунків

У сучасних умовах розробка вебзастосунків є одним із ключових напрямів у сфері інформаційних технологій, адже це динамічна галузь, яка постійно розвивається. Така тенденція пов'язана зі зростанням попиту на онлайн-сервіси, що забезпечують інтерактивну взаємодію користувача з системою в режимі реального часу. Саме для розробки вебсистем існує велика кількість інструментів, фреймворків та технологій, які дозволяють створювати функціональні, масштабовані та зручні у використанні застосунки.

Серед найпопулярніших сучасних рішень можна виділити:

Laravel - потужний фреймворк для розробки вебзастосунків мовою PHP. Забезпечує зручну архітектуру MVC, має вбудовані механізми аутентифікації, авторизації, роботи з базами даних через ORM (Eloquent), маршрутизацію, систему шаблонів Blade. Laravel часто використовується для створення адміністративних панелей, CRM-систем, API-серверів, [15].

Переваги:

- Висока швидкість розробки;
- Велика спільнота та кількість пакетів;
- Добре підходить для створення панелей адміністратора.

React.js (JavaScript) - бібліотека для побудови інтерфейсів від компанії Meta (Facebook). Широко використовується у SPA (Single Page Application) для динамічної взаємодії без повного перезавантаження сторінки.

Переваги:

- Висока швидкість інтерфейсу;
- Компонентна структура;

- Підтримка численних інструментів розробки, [16].

Node.js - середовище виконання JavaScript на сервері. Дозволяє створювати високопродуктивні сервери для обробки великої кількості одночасних з'єднань.

Застосування: бекенд для ігрових серверів, чатів, аналітики в реальному часі[17].

MySQL / PostgreSQL - Системи управління реляційними базами даних, які забезпечують надійне зберігання інформації, швидкий пошук і підтримку складних запитів. Ідеально підходять для зберігання інформації про користувачів, статистику активності, логування подій, [18].

Bootstrap / Tailwind CSS - Фреймворки для швидкої розробки адаптивного дизайну. Bootstrap забезпечує компоненти інтерфейсу, а Tailwind дає змогу гнучко кастомізувати стилі через класи, [19].

Сучасні вебзастосунки орієнтовані на зручність, швидкість і масштабованість. Їх інтеграція в систему моніторингу активності користувачів ігрового серверу дозволяє створити ефективний інструмент для адміністрації. При розробці власного рішення використовуються перевірені технології - PHP, MySQL, Bootstrap, що забезпечує надійність і простоту підтримки системи.

При виборі технологій для розробки вебзастосунку важливо враховувати специфіку проєкту, навантаження на систему, потребу в масштабуванні та зручність подальшої підтримки.

1.6 Основні етапи розробки вебзастосунку. Визначення завдань і створення моделі вебзастосунку

Перед початком розробки вебзастосунку було проведено аналіз функціональних та нефункціональних вимог, які визначають основні характеристики, можливості та обмеження системи. Аналіз вимог дозволяє

сформувати чітке бачення майбутньої системи, зменшити ризики та забезпечити відповідність очікуванням користувачів.

В таблиці 1.2 наведено приклад основних функціональних вимог вебзастосунку.

Таблиця 1.2 - Основні функціональні вимоги вебзастосунку

№ з/п	Основні функціональні вимоги	Визначення завдань
1.	Система автентифікації	Користувачі повинні мати змогу входити в систему з використанням логіна та пароля.
2.	Ролі користувачів	Система передбачає щонайменше дві ролі - звичайний користувач та адміністратор, які мають різні рівні доступу до функцій.
3.	Панель керування	Адміністратор має можливість: <ul style="list-style-type: none"> - додавати нових лідерів та їх заступників; - редагувати або архівувати інформацію; - переглядати чорний список і архів ЧС; керувати правами доступу.
4.	Відображення лідерів	Система відображає список лідерів державних і нелегальних організацій, а також міністрів/суддів з їх онлайн-активністю, дисциплінарними записами, балами та іншими показниками
5.	Профілі користувачів	Кожен користувач може переглядати свій профіль, та змінювати аватар.
6.	Адаптивність інтерфейсу	Вебінтерфейс повинен коректно відображатися на пристроях з різними розмірами екранів.

Продовж. Таб.1.2

7.	Безпека	Доступ до функцій адміністрування можливий лише після автентифікації та перевірки прав.
8.	Масштабованість	Структура бази даних дозволяє у майбутньому додати нові типи організацій, показників або ролей.
9.	Продуктивність:	Сайт повинен швидко завантажуватись навіть при великій кількості даних у таблицях.

На цьому етапі розробник повинен визначити основні напрями функціональності сайту, а також створити макет, який визначатиме приблизне розташування елементів на сторінках та загальну архітектуру.

Прототип сторінок і структуру сайту можна створити у Paint або намалювати на аркуші паперу.

1. На основі макету в Adobe Photoshop або будь-якому іншому зручному графічному редакторі створюється остаточний зовнішній вигляд майбутнього сайту, опрацьовуються всі деталі та нюанси. Цей етап є одним із ключових.
2. Верстка HTML - сторінок. На цьому етапі дизайн розбивається на окремі блоки, які описуються за допомогою HTML, CSS та JavaScript. Фактично зображення сайту перетворюються на справжні вебсторінки застосунку.

Розробка вебзастосунку - це багатоетапний процес, який вимагає системного підходу. Основні етапи розробки вебзастосунку:

- Етап планування та аналізу
- Етап проєктування
- Етап розробки
- Етап тестування
- Етап запуску
- Етап підтримки та розвитку

Також на етапі проєктування створюється макет, який визначає приблизне розташування елементів на сторінках та загальну структуру сайту.

На етапі плануванні та аналізу важливо мати чітке формулювання цілей, які має вирішити застосунок (наприклад, збільшення продажів, автоматизація процесів, надання інформації). Також детальний опис майбутніх користувачів, їхніх потреб, очікувань та сценаріїв взаємодії. Визначення функціональних вимог (наприклад, реєстрація користувачів, пошук товарів, обробка замовлень), а також нефункціональних вимог (продуктивність, безпека, масштабованість, юзабіліті, сумісність).

Основні завдання на етапі проєктування вебзастосунку:

1. Визначення загальної архітектури системи (а саме: тип застосунку, фронтенд/бекенд)
2. Побудова схеми навігації сайту
3. Визначення ролей користувачів.
4. Визначення мов програмування, баз даних та інших інструментів
5. Проєктування баз даних (визначення таблиць, зв'язків,полів)
6. Розробка UX (User Experience)
7. Розробка UI (User Interface)
8. Дизайн-макет сайту

На основі прототипу в Adobe Photoshop або іншому графічному редакторі створюється остаточний зовнішній вигляд сайту. Опрацьовуються всі деталі та нюанси. Цей етап є одним із ключових у процесі створення.

На етапі розробки - а це безпосередньо написання коду відповідно до технічного завдання та дизайн-макетів, що були створені на попередніх етапах.

Отже, кожен з цих етапів є важливим і взаємопов'язаним, а їх якісне виконання забезпечує успішну реалізацію вебзастосунку.

1.7 Висновки до першого розділу

Проаналізувавши розділ 1 можна дійти висновку, що аналіз стану області рішення завдання є критично важливим початковим етапом у розробці вебзастосунку. Адже саме аналіз закладає фундамент для всього проєкту, забезпечуючи чітке розуміння потреб, обмежень та можливостей. Завдяки дослідженню існуючих аналогів або конкурентів можна виявити як і кращі практики, так і типові помилки, яких варто уникати.

Тому в розділі першому було розглянуто та обрано основні технології, що використовуються для створення вебзастосунків у рамках цієї роботи, а саме:

- Мова розмітки - HTML.
- Мова опису зовнішнього вигляду - CSS.
- Мова сценаріїв - JavaScript.
- Мова для написання серверної частини застосунку - PHP.
- Система управління базами даних - MySQL.

Також розглянуто бібліотеки та сервіси, що дозволяють швидко вирішувати низку завдань для проєктних рішень в даній кваліфікаційній роботі:

- Набір функцій на мові JavaScript для взаємодії з елементами сайту - jQuery.
- Набір користувацьких інтерфейсів на основі jQuery - jQuery UI.
- Кнопки, форми, модальні вікна, навігацію Bootstrap 5.3 1. Обрані основні інструменти для створення вебзастосунку:
- Текстовий редактор - Notepad++.
- Примітивний графічний редактор - Paint.
- Професійний графічний редактор - Adobe Photoshop.

- Засоби для створення векторної графіки - Inkscape та Adobe Illustrator.

Визначені основні етапи розробки вебсайту:

- Постановка задач, які повинен вирішувати застосунок.
- Створення дизайну.
- Створення серверної частини.
- HTML-верстка.

Отже, проведений аналіз стану області рішення завдання є не просто формальністю, а стратегічною інвестицією, яка дозволяє створити чітке бачення майбутнього вебзастосунку. Він забезпечує системний підхід до вибору технологій, визначення функціоналу та планування етапів, значно зменшуючи ризики та гарантуючи, що кінцевий продукт буде відповідати очікуванням користувачів та бізнес-цілям, працюючи при цьому надійно та безпечно в умовах сучасних загроз.

Тому, саме аналіз допомагає сформувати єдине та однозначне розуміння того, яким має бути кінцевий продукт, які функції він виконуватиме і які проблеми вирішуватиме. Це усуває двозначності та розбіжності між усіма зацікавленими сторонами.

РОЗДІЛ 2

ПРОЄКТНІ РІШЕННЯ

2.1 Розробка прототипу та визначення основного функціоналу

Arizona RP - це один з найбільших проєктів у мультиплеєрній грі San Andreas Multiplayer (SAMP). Це рольова онлайн-гра, де гравці беруть на себе ролі персонажів і взаємодіють один з одним у віртуальному світі на основі певних правил та сценаріїв. У грі симулюється життя, зокрема існують державні організації, бізнеси, злочинні структури, суди, правоохоронні органи тощо.

У межах гри існують організації (державні, нелегальні, мафіозні, медіа, тощо), якими керують призначені гравці:

Лідер - це гравець, що очолює організацію (наприклад, поліцію, армію або мафію). Він відповідає за дисципліну, набір учасників, внутрішні події, звітність тощо.

Заступник лідера - допомагає лідеру виконувати обов'язки та може його замінювати у разі відсутності. Призначення лідерів та контроль їхньої діяльності здійснює адміністрація серверу.

Онлайн гравця - це загальна кількість часу, який гравець провів у грі на сервері.

Онлайн враховується з моменту входу гравця на сервер і до його виходу, і часто використовується для:

- перевірки активності лідера фракції;
- контролю виконання нормативів (наприклад, мінімальний онлайн на посаді);
- видачі нагород або штрафів;
- прийняття рішень щодо зняття/призначення гравця на посаду.

Старша адміністрація - це працівники серверу (не гравці в звичайному розумінні), які виконують наглядові та організаційні функції:

- Призначають/знімають лідерів;
- Видають попередження та догани за порушення правил;
- Перевіряють онлайн, звіти та активність;
- Забезпечують дотримання RP-стандартів.

Старша адміністрація зазвичай має доступ до адміністративної панелі або внутрішніх інструментів для моніторингу і керування організаціями.

У такому складному ігровому процесі виникає потреба в автоматизованому моніторингу та керуванні лідерами, оскільки все це неефективно вести вручну через Google-таблиці або Discord

Можна зробити висновок, що система повинна:

- Відображати лідерів, їхній онлайн, кількість доган, балів.
- Дозволяти старшій адміністрації додавати, редагувати або знімати лідерів;
- Вести чорні списки (Black List), архіви, статистику.
- Підтримувати облік, попереджень та інших адміністративних дій.

Таким чином, вебзастосунок - це внутрішня CRM-система для модераторів та адміністрації ігрового серверу, яка спрощує контроль за гравцями, підвищує ефективність керування та зменшує кількість людських помилок.

Основне завдання розроблюваного сайту - спрощення роботи старшої адміністрації ігрового сервера «Arizona RP». Головний адміністратор та його помічники стикаються з такими задачами:

- Зберігання анкет лідерів та заступників лідерів.
- Зберігання архіву лідерів та заступників лідерів.
- Організація системи штрафів і заохочень.
- Ведення системи чорного списку.

Розроблюваний сайт має допомагати у вирішенні цих задач.

Структура сайту передбачається як багатосторінкова, при цьому перехід між сторінками вільний і доступний з будь-якої сторінки на будь-яку іншу. Така структура дозволяє легко додавати нові розділи на сайт, а також ізолювати код, що забезпечує працездатність інших сторінок у разі помилки на одній із них.

Для кращого розуміння логіки навігації та взаємодії користувача з вебзастосунком було створено мапу сайту (наведено на рисунку 2.1) (структурну діаграму сторінок). Вона відображає основні сторінки та зв'язки між ними, а також розмежування доступу залежно від ролі користувача. Така структурна діаграма є важливим етапом у проектуванні вебзастосунку, особливо для систем з розширеною ієрархією користувачів і функціоналом модерації.

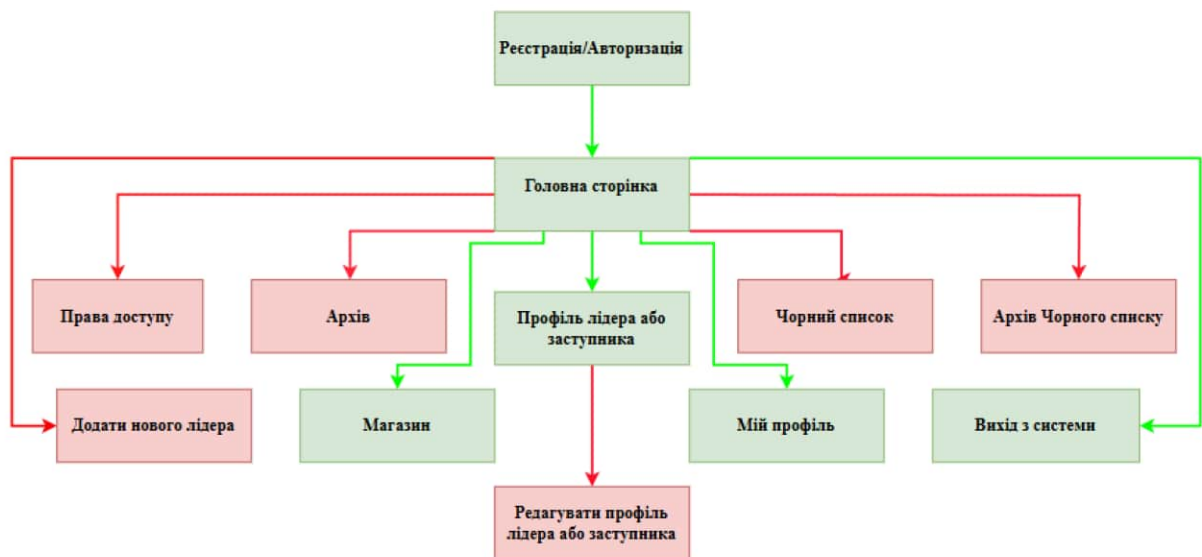


Рис.2.1 - Мапа сайту

Мапа починається з точки входу - Реєстрація/Авторизація, після чого користувач потрапляє на Головну сторінку, яка є центральним вузлом усієї системи.

З головної сторінки користувач має доступ до таких сторінок:

1) Загальнодоступні сторінки (зелений колір):

- Профіль лідера або заступника - відображення детальної інформації про конкретного керівника.
- Магазин - перегляд придбаних предметів або винагород.
- Мій профіль - персональна інформація користувача.
- Вихід з системи - завершення сесії користувача

Також с цих сторінок, користувач може повернутись на будь яку доступну йому сторінку.

2) Адміністративні сторінки (червоний колір) - доступні лише користувачам з роллю адміністратора або модератора:

- Додати нового лідера - створення нового запису у системі.
- Права доступу - управління ролями та дозволами користувачів.
- Архів - історія змін, старі дані про лідерів.
- Редагувати профіль лідера або заступника - можливість змінити дані користувача.
- Чорний список - список порушників або небажаних осіб.
- Архів Чорного списку - історія внесених та видалених записів зі списку.

Модератор або адміністратор сайту, також має вільний доступ переходу між цими сторінками.

Сайт замислюється як допоміжник для модерації сервера за стеження за активністю лідерів та заступників.

Всього на сайті має бути 12 сторінок. Сторінки відображені в таблиці 2.1.

Таблиця 2.1 Сторінки вебсайту

№	Сторінка	Назва файлу
1.	Авторизація	login.php
2.	Сторінка реєстрації	register.php
3.	Сторінка лідерів та заступників	index.php
4.	Сторінка профілю лідерів чи заступників	leader_profile.php
5.	Архів лідерів	archive.php
6.	Форма додавання лідерів та заступників	add_leader_form.php
7.	Чорний список	blacklist.php
8.	Архів чорного списку	archive_blacklist.php
9.	Редагування профіль лідера чи заступника	edit_leader.php
10.	Сторінка надання доступу користувачу до додаткових функцій модерації	manage_users.php
11.	Магазин	shop.php
12.	Сторінка профілю користувача	profile.php

Усі файли вказані з розширенням «.php», оскільки файли цього формату дозволяють об'єднувати в собі елементи HTML та серверних скриптів, що спрощує розробку нескладних сторінок, створюючи їх без використання технології асинхронних викликів.

2.1.1 Створення макету спільних елементів вебзастосунку

У кожного багатосторінкового сайту є повторювані області: заголовок, підвал, різні меню. Макет спільних елементів розроблюваного вебзастосунку зображено на рисунку 2.2.

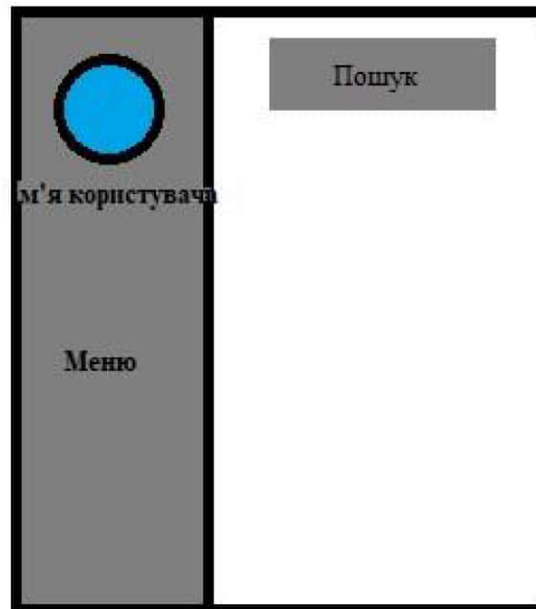


Рис. 2.2 - Макет загальних для всіх сторінок елементів

Загальними елементами для всіх сторінок є меню навігації, фотографія користувача (блакитне коло) та ім'я користувача. Також передбачено блок із формою пошуку.

Біла область - це поле відображення різних елементів сайту. Доступ до даних, а особливо до їх редагування, повинен бути обмежений авторизацією. Для цього необхідна спеціальна сторінка. Розглянемо її макет на рисунку 2.3.

На сторінці повинні бути розміщені:

- форма для введення даних для входу
- кнопка реєстрації.

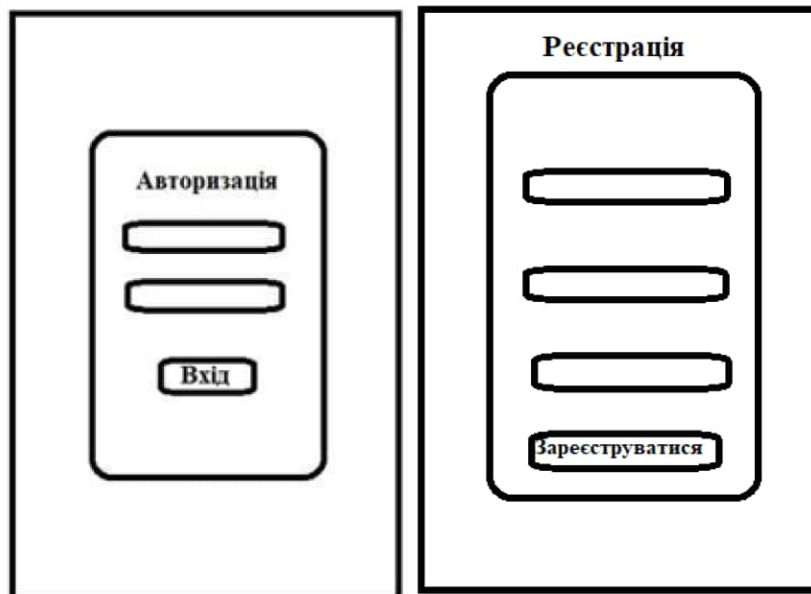


Рис. 2.3 - Макет сторінки реєстрації та авторизації

При призначенні нового лідера чи заступника, модератор створює анкету користувача. Для цього потрібний інструмент для додавання, подальшого перегляду та редагування даних. У зв'язку з цим мають бути створені дві сторінки, які відображені на рисунку 2.4. - сторінка створення користувача (ліворуч) та сторінка профілю з меню редагування (праворуч).

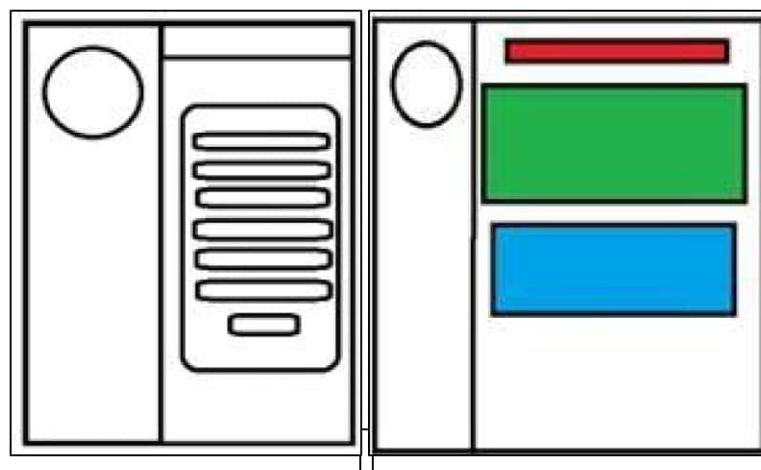


Рис. 2.4 - Макет сторінок

На рисунку 2.4 відображено праворуч різними кольорами розмежовані зони на сторінці профілю: червона - нік лідера чи заступника, форум; синя - лог дій, в ньому відображаються всі операції, здійснені з профілем: нарахування балів, видача покарань, та інше; зелена - блок з інформацій про лідера чи заступника, який включає показники активності, бали, спосіб призначення, кількість годин, проведених в мережі по днях; жовта - форма редагування, яка дозволяє змінити будь-який пункт із зеленої зони.

Загальні списки лідерів, архіви відображаються таблицями на окремих сторінках, при цьому таблиці будуть розміщені у білій зоні макета з рисунку 2.1.

Для організації на сайті системи штрафів та заохочень створюється система внутрішніх балів, яка дозволяє купувати за внутрішню валюту зняття доганів та попереджень.

Система чорного списку (ЧС) повинна включати можливість створення, перегляду та редагування заявок, список усіх користувачів, занесених до ЧС, а також архів та інформацію щодо використання, мати систему зручного пошуку.

При реалізації системи чорного списку використовуватиметься одна сторінка, яка в залежності від зазначених параметрів відобразатиме потрібний розділ та функціонал.

Можна дійти висновку, що саме використання однієї сторінки з динамічним відображенням функціоналу зробить інструмент дуже зручним та ефективним для модераторів, дозволяючи швидко перемикатися між завданнями без зайвих перезавантажень сторінок. Також вся інформація та функціонал, пов'язані з ЧС, зосереджені в одному місці, тому модераторам та адміністраторам не потрібно переходити між різними сторінками для створення заявки, перегляду списку чи аналізу архіву. Саме таке рішення робить робочий процес більш ефективним.

Також на сайті є достатня кількість php скриптів, які відображені в таблиці 2.2.

Таблиця 2.2 - php скрипти вебзастосування

№	Назва файлу.	За що відповідає.
1.	edit_leader.php	Редагування інформації про лідера в базі даних.
2.	log_processor.php	Обробка та відображення логів з файлу
3.	remove_leader.php	Видаляє лідера з основної таблиці та бд та переносить його до архіву.
4.	update_leader_notes.php	Оновлює примітки та покарання для лідера.
5.	update_leader_schedule.php	Змінює графік активності лідера по днях тижня.
6.	upload_avatar.php	Завантажує аватар користувача на сервер з перевіркою типу та розміру файлу.
7.	update_archived_leader.php	Редагує дані лідера в архівній таблиці.
8.	db.php	Налаштування підключення до бази даних MySQL. Використовується для взаємодії з СУБД.

2.2 Створення дизайну вебсайту та верстка

До дизайну сайту немає великих вимог. Його головне завдання - полегшення роботи, а не залучення клієнтів та інше. На рисунку 2.6. відображено наглядно сторінку авторизації.

Вона має простий інтерфейс, який дозволяє користувачеві без зайвих дій здійснити вхід у систему. Такий підхід до дизайну забезпечує:

- мінімальне відволікання уваги;
- швидке виконання дій;

- зрозумілу структуру елементів (форма входу, кнопки, підказки).

Ключовими характеристиками сторінки авторизації є простота, лаконічність та інтуїтивність, що відповідає загальній меті вебзастосунку — забезпечення ефективної роботи з мінімальними зусиллями з боку користувача.

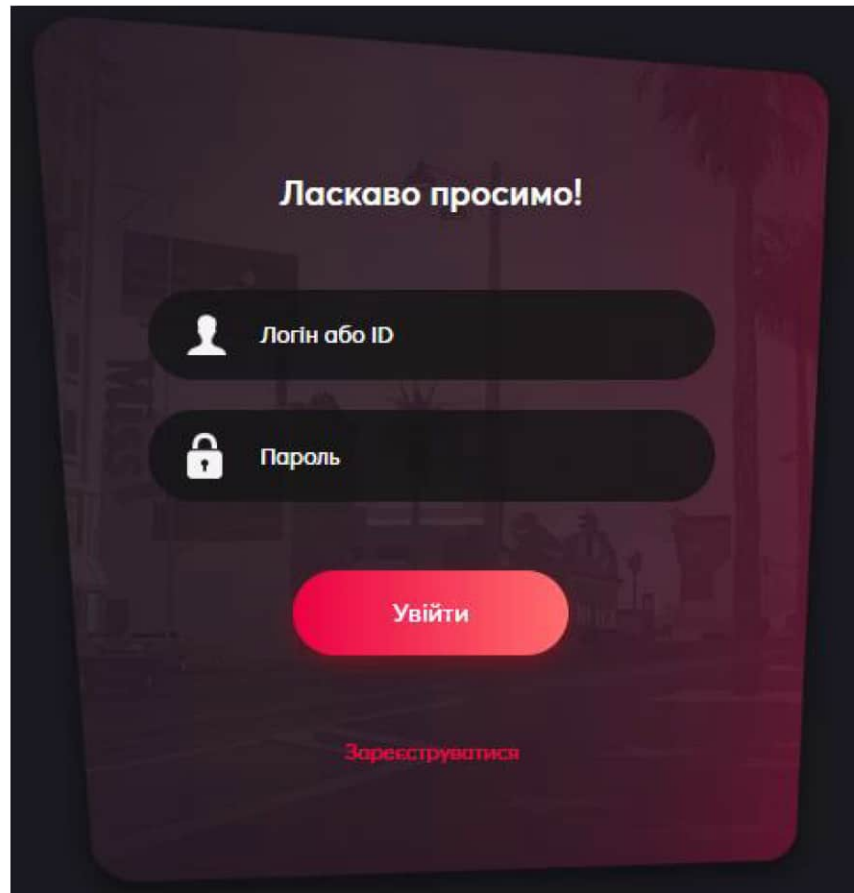


Рис. 2.6. - Сторінка авторизації

Сторінка login.php реалізує форму авторизації з інтуїтивно зрозумілим інтерфейсом, обробкою даних на сервері та базовими механізмами безпеки.

Файл складається з двох основних частин:

1. Backend (PHP) - обробка даних форми, перевірка логіна/пароля, робота з базою даних.

2. Frontend (HTML/CSS/JS) - візуальне відображення форми, інтерактивні елементи.

Frontend - Для початку у файлі «login.php » додамо службову інформацію (підключимо стилі та додаткові файли). На рис. 2.6. зображено код, який описує заголовок сайту.

- Форма авторизації (<form>) - Має поля для логіна (username) та пароля (password), кнопку входу.
- Відображення помилок - Блок warning-pass показує повідомлення про помилки (наприклад, "Невірний пароль").
- Кнопка "Показати пароль" - Перемикає тип поля password ↔ text (реалізовано на JS).
- Посилання на реєстрацію - Перенаправляє на сторінку register.php.
- Стилi (CSS) Використовує класи(inputgroup,authorization,centered) для адаптивного дизайну.

```

<html lang="uk">
<head>
  <meta charset="utf-8" />
  <link rel="shortcut icon" href="include/images/logo.png" type="image/png">
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <link rel="stylesheet" href="include/css/app.css" />
  <script src="include/js/jquery.js"></script>
</head>
<body>

```

Рис. 2.7. - Код опису службової інформації вебсторінки Авторизації

Backend (PHP) :

- Підключення до БД (require 'db.php') - Використовує окремий файл для з'єднання з MySQL.
- Старт сесії (session_start()) -Забезпечує збереження даних користувача після авторизації.

- Обробка POST запиту - Перевіряє, чи форма відправлена, і витягує username та password.
- Валідація даних - Перевіряє, чи заповнені обидва поля.
- Запит до БД - Шукає користувача за логіном через підготовлений запит (захист від SQL-ін'єкцій).
- Перевірка пароля - Порівнює хеш пароля з бази даних з введеним значенням (password_verify).
- Збереження в сесії - Записує username, user_id та roles у глобальний масив \$_SESSION.
- Перенаправлення - Після успішної авторизації відправляє користувача на index.php.

Якщо користувача не має в системі, в нього буде повідомлення "Користувача не знайдено" (Зовнішній вигляд показано на рисунку 2.8.)

Повідомлення "Користувача не знайдено" призначене для інформування користувача про те, що введені дані (логін) не співпадають з жодним записом у базі даних.

- Повідомлення розміщується по центру екрану, використовуючи напівпрозоре затемнення фону.
- Кнопка "Повернутись" має яскравий червоний колір, що привертає увагу користувача і дозволяє швидко повернутись на попередню сторінку.
- Застосовується напівпрозоре затемнення фону, що візуально виділяє повідомлення серед іншого контенту та створює ефект модального вікна.
- Інтерфейс передбачає мінімальну кількість дій для користувача — натиснувши на кнопку, він одразу повертається на сторінку авторизації.

Такий підхід дозволяє не лише забезпечити зрозумілий зворотний зв'язок, але й підвищити зручність користування системою навіть у випадках помилки.

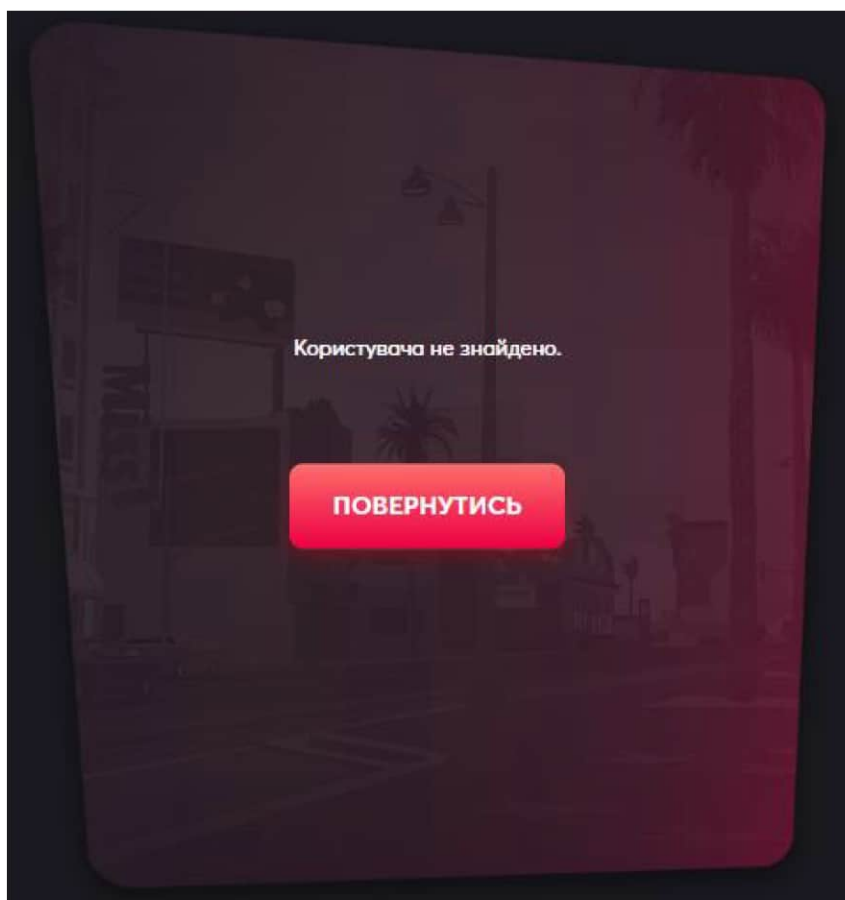


Рис. 2.8 - Зовнішній вигляд сторінки, коли користувач не зареєстрований

У випадку, якщо обліковий запис не знайдено, користувач має можливість створити новий обліковий запис за допомогою кнопки “Зареєструватися”. Після натискання кнопки, він перенаправляється на сторінку реєстрації, інтерфейс якої показано на рисунку 2.9.

Після успішного заповнення форми та натискання кнопки “Зареєструватися”, дані передаються на сервер для обробки. У разі успішної реєстрації користувач потрібно перейти на сторінку авторизації та авторизуватися. Якщо під час перевірки виникають помилки (наприклад,

логін уже існує, паролі не співпадають, або поля не заповнені), на екрані виводиться відповідне повідомлення. Це забезпечує зручність користування та захист від помилкових введень.



Рис. 2.9 - Повідомлення що під час перевірки виникла помилка(логін уже існує, паролі не співпадають, або поля не заповнені)

За обробку даних форми відповідає файл register.php, який:перевіряє унікальність логіна;

- здійснює валідацію введених даних (довжина, співпадіння паролів); зберігає облікові дані користувача в базі даних (з хешуванням пароля).
- зберігає облікові дані користувача в базі даних (з хешуванням пароля).

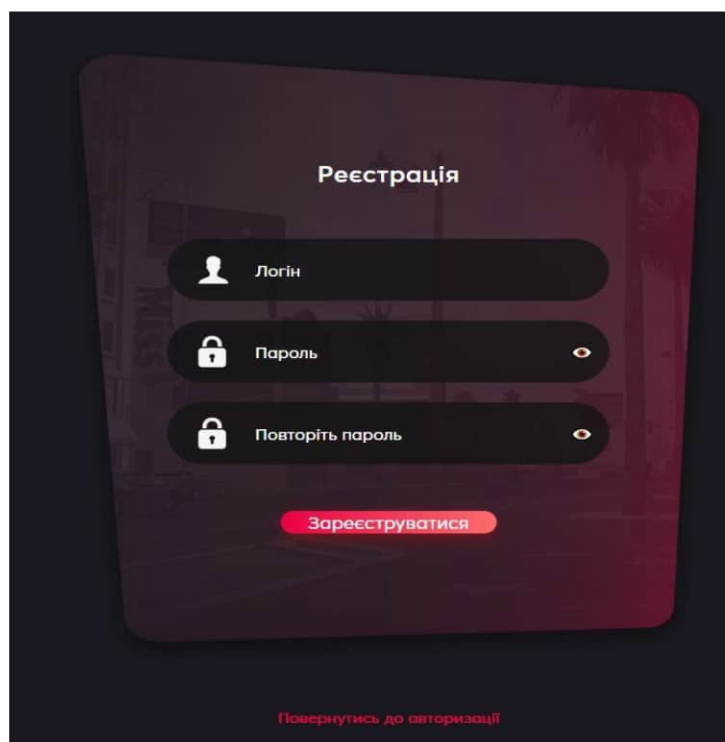


Рис. 2.10 - Зовнішній вигляд сторінки Реєстрації

Службова інформація (стислі та додаткові файли), ті ж самі, як і на сторінці Авторизації

При натисканні кнопки "Зареєструватися" виконується:

- валідація;
- чи всі поля заповнені;
- чи довжина паролю від 6 до 30 символів;
- чи співпадає пароль і підтвердження;
- Перевірка, чи логін вже зайнятий (запит у БД);

Якщо все гаразд:

- хешується пароль (password_hash);
- створюється запис у таблиці users;
- Встановлення повідомлення \$message про результат

2.2.1 Головна сторінка сайту, с статистикою лідерів та заступників

Після авторизації, користувач перенаправляється на сторінку "index.php". Відповідно до пункту 2.1 вона містить таблицю: список лідерів та заступників.

Користувач має змогу візуально оцінити активність та дисципліну кожного лідера завдяки зручному представленню даних. Підсвітка червоним кольором використовується для позначення критичних значень (наприклад, 3/3 догани), що дозволяє швидко ідентифікувати проблемні ситуації. Усі дані в таблиці оновлюються автоматично або за допомогою адміністратора.

Інтерфейс є адаптивним та зручним для перегляду з різних пристроїв. Завдяки структурованості таблиць, користувачі легко знаходять потрібну інформацію. Додатково реалізована можливість переходу до профілю лідера через натискання на його нікнейм.

Організація	Нікнейм	Онлайн	Discord	Догани	Попередження	Бали	Останнє оновлення
Уряд	Denis_Shmigal	04:19:36	DenisShmigal	3 / 3	2 / 5	0	2025-06-09 09:38:20
СБУ	Vasil_Malyuk	02:57:09	VasilMalyuk	1 / 3	2 / 5	500	2025-06-09 19:55:33
СБУ (Гастулінки)	Sergey_Andrushchenko	04:42:18	SergeyAndrushchenko	0 / 3	0 / 5	0	2025-06-07 23:18:10
SWAT	Serj_Rosati	03:08:44	SerjRosati	0 / 3	0 / 5	0	2025-06-07 23:17:10
Національна Гвардія	Oleksandr_Pivnenko	03:16:07	OleksandrPivnenko	0 / 3	0 / 5	0	2025-06-07 23:19:38
Поштові Служби	Mykola_Oleschuk	01:57:23	MykolaOleschuk	0 / 3	0 / 5	0	2025-06-07 23:20:29

Фракція	Нікнейм	Онлайн	Discord	Догани	Попередження	Бали	Останнє оновлення
Українська Мафія	Arsena_Allyev	03:41:22	ArsenaAllyev	5 / 5	0 / 5	0	2025-06-09 11:59:56
Winlock MC	Xavier_Cooper	01:41:28	XavierCooper	0 / 5	0 / 5	0	2025-06-07 23:21:57

Рис. 2.11. - Зовнішній вигляд сторінки "index.php"

Структура таблиць однакова для всіх секцій:

1. Організація / Фракція - назва фракції;
2. Нікнейм - ім'я лідера, яке може бути посиланням на його профіль;
3. Онлайн - час онлайн (перетворюється функцією secondsToTime);
4. Discord - Discord контакт лідера;
5. Догани / Попередження - відображаються числово та з підсвіткою, якщо перевищено норму;
6. Бали - оцінка/рейтинг лідера;
7. Останнє оновлення - дата останнього оновлення інформації.

Код побудовано на вкладених циклах foreach, які проходять по організаціях серверу:

```
<?php foreach ($gov_organizations as $key => $name): ?>
  <?php if (isset($leaders[$name])): ?>
    <?php foreach ($leaders[$name] as $leader): ?>
```

Рис. 2.12 - Цикл foreach

Цикл foreach перебирає масив \$gov_organizations, де:

- \$key - це умовне скорочення (наприклад, 'gov');
 - \$name - це повна назва організації (наприклад, 'Уряд').
1. Лідери державних організацій (Уряд, Центр Ліцензування, СБУ, SWAT, RCSD, LSPD, SFPD, В'язниця Суворого Режиму, LSMC, LVMC, Національна Гвардія, Повітряні Сили, ЗМІ ЛС, Пожежний Департамент) - \$gov_organizations
 2. Лідери нелегальних організацій(Ukrainian Mafia, Warlock MC, Grove Street Families, The Rifa Gang, The Ballas Gang, Los-Santos Vagos) -\$illegal_organizations
 3. Міністри та судді(Міністр Юстицій, Міністр Оборони, Міністр Охорони здоров'я, Суддя) - \$gov_ministr

```
$gov_organizations = [
    'gov' => 'Уряд',
    'gcl' => 'Центр Ліцензування',
    'fbi' => 'СБУ',
    'swat' => 'SWAT',
    'rcsd' => 'RCSD',
    'lspd' => 'LSPD',
    'sfpd' => 'SFPD',
    'tcr' => 'В'язниця Суворого Режиму',
    'lsmc' => 'LSMC',
    'lvmc' => 'LVMC',
    'als' => 'Національна Гвардія',
    'asf' => 'Повітряні Сили',
    'smils' => 'ЗМІ ЛС',
    'fire' => 'Пожежний Департамент'
];

$illegal_organizations = [
    'rm' => 'Ukrainian Mafia',
    'wmc' => 'Warlock MC',
    'grove' => 'Grove Street Families',
    'rifa' => 'The Rifa Gang',
    'ballas' => 'The Ballas Gang',
    'vagos' => 'Los-Santos Vagos'
];

$gov_ministr = [
    'minjust' => 'Міністр Юстицій',
    'minobr' => 'Міністр Оборони',
    'minzdr' => 'Міністр Охорони здоров'я',
    'sud' => 'Суддя'
```

Рис. 2.13. - Представлені масиви даних із організаціями серверу

У таблицях передбачена візуальна індикація проблемних записів. Якщо кількість доган ≥ 3 (або ≥ 5 для нелегалів). Якщо кількість попереджень \geq допустимого значення ($\$max_warnings$), то біля нікнейму з'являється червоний знак (!). Приклад відображено на рисунку 2.14.

Організація	Нікнейм	Онлайн	Discord	Догани
Уряд	Denis_Shmigal !	04:19:36	DenisShmigal	3 / 3

Рис. 2.14. - Зовнішній вигляд, якщо кількість попереджень \geq допустимого значення ($\$max_warnings$)

На рисунку 2.16 відображено реалізація проблемних записів, якщо доганів ≥ 3 , адже ця функція стане потужним інструментом у системі модерації, допомагаючи підтримувати порядок на сервері та відстежувати лідерів чи заступників які на межі зняття з посади.

```
<?php if ($leader['strikes'] >= 3 || $leader['warnings'] >= $max_warnings): ?>
    <span class="badge-custom ms-2">!</span>
<?php endif; ?>
```

Рис. 2.15 - Реалізація проблемних записів, якщо доганів ≥ 3

А також застосовується jQuery-скрипт, який зображений на рисунку 2.18.

```
$(document).ready(function() {
    $('tbody tr').each(function() {
        const strikes = parseInt($(this).find('td:nth-child(5)').text().split('/')[0]);
        const warnings = parseInt($(this).find('td:nth-child(6)').text().split('/')[0]);
        const maxWarnings = parseInt($(this).find('td:nth-child(6)').text().split('/')[1]);

        if (strikes >= 3 || warnings >= maxWarnings) {
            $(this).addClass('bg-danger bg-opacity-10');
        }
    });
});
```

Рис. 2.16 - Застосування jQuery-скрипта

Усі таблиці обгорнуті в блок “table-responsive”

1. дозволяє прокручувати таблиці горизонтально на мобільних пристроях
2. забезпечує адаптивну верстку без спотворення елементів.

Цей блок є центральною частиною інтерфейсу моніторингу, що надає адміністраторам повну інформацію про:

1. активність лідерів;
2. поточні попередження і догани;
3. загальний стан організацій.

Інтерфейс поєднує захищену серверну логіку, адаптивний дизайн та інтерактивну підсвітку проблем, що робить систему зручною та ефективною для аналізу ситуації на сервері.

2.2.2 Бічна панель для зручного переходу між сторінками

У вебзастосунку реалізовано зручну бічну панель, яка розміщується зліва на всіх основних сторінках. Вона служить для навігації, відображення інформації про користувача та надає швидкий доступ до адміністративних функцій (у разі наявності прав доступу). Сторінки " blacklist.php", "archive.php", "archive_blacklist.php" будуються аналогічно, відмінність тільки в назву файлів і даних, які вони відображають.

У верхній частині панелі розташовано блок з даними поточного користувача:

Аватар - завантажується з серверу. Реалізована можливість зміни аватара: при кліку на зображення відкривається діалог вибору файлу, і після вибору він автоматично завантажується через форму (автоматична відправка `form.submit()` при зміні `input:file`).

Після блоку користувача йде вертикальне меню з такими пунктами:

1. Лідери та заступники - основна сторінка з даними про адміністрацію ігрового сервера.
2. Мій профіль - сторінка перегляду особистої інформації користувача.
3. Магазин - перехід до вбудованого ігрового магазину.

Якщо поточний користувач має відповідні права (перевірка змінною `$has_admin_access`), відображається випадаюче меню "Адмін-панель" з такими можливостями:

1. Додати лідера - перехід на форму додавання нового лідера чи заступника сервера.
2. Архів лідерів - список колишніх лідерів та заступників.
3. Чорний список - користувачі, які отримали бан та не мають право знову очолювати роль лідера чи заступника.
4. Архів ЧС - Історія лідерів чи заступників, які колись були внесені в чорний список, але з якихось причин було винесено
5. Права доступу - модуль керування користувачами та їх ролями (адмін, модератор, користувач).

Меню реалізоване з використанням компонента Bootstrap Accordion, що дозволяє зручно згортати/розгортати розділ.

Вся інформація у панелі динамічно підставляється з PHP-перемінних, які формуються на основі активної сесії.

Виведення аватара та нікнейму захищене через `htmlspecialchars()`, що запобігає XSS-атакам.

Панель адаптована під Bootstrap, що забезпечує кросбраузерність і респонсивний дизайн.

На рис. 2.17 представлено три варіанти зовнішнього вигляду бічної панелі (sidebar), які змінюються відповідно до ролі користувача в системі:

Наприклад звичайний користувач (Xavier_Cooper) має доступ лише до основних розділів: "Лідери та заступники", "Мій профіль", "Магазин" та функцію виходу з системи.

А ось користувач з роллю "moderator" (Jesus_Cooper, середня панель) бачить додатковий пункт меню - "Адмін-панель", яка дозволяє розгорнути адміністративні функції.

Такий підхід до відображення інтерфейсу забезпечує зручність навігації та дозволяє приховувати адміністративні елементи від користувачів, які не мають відповідних прав.

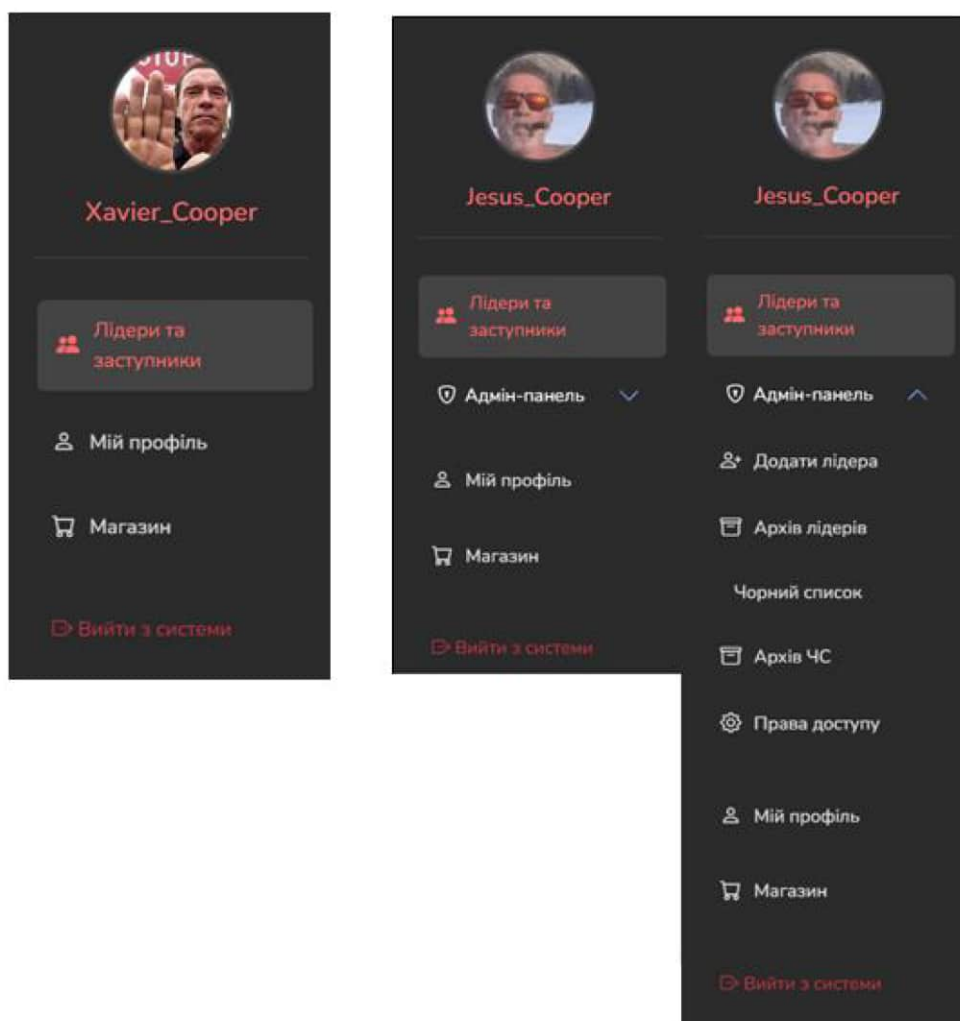


Рис. 2.17. - Зовнішній вигляд "sidebar" для звичайних користувачів та с правого боку для користувачів з роллю "moderator" та "admin"(с правого боку вигляд розгорнутої адмін-панелі)

Функціонал "Адмін-панелі" реалізовано у вигляді випадаючого меню, що дозволяє зберігати компактність інтерфейсу та уникати перевантаження елементами навігації при стандартному перегляді.

Бічна панель реалізована на основі HTML, PHP і Bootstrap, що забезпечує інтерактивність, адаптивність та динамічну генерацію вмісту залежно від ролі користувача.

```
<div class="user-name"><? = htmlspecialchars($admin_nickname); ?></div>
```

Виводиться нік користувача, отриманий із бази даних.

Функція `htmlspecialchars()` використовується для запобігання скриптовим атакам (XSS)

Стилі адаптовані під темну тему (`bg-transparent`, `text-white`, `shadow-none`), що відповідає сучасному дизайну вебзастосунку.

2.2.3 Сторінка профілю лідера чи заступника з докладною інформацією

Найскладнішими сторінками є «`leader_profile.php`» та «`profile.php`». Сторінки схожі між собою, але відображають різні поля якщо ви не є лідером чи заступником. Побудуємо структуру сторінки профілю лідера чи заступника, а потім на її прикладі зробимо сторінку профілю користувача. Для цього нам знадобиться стандартна сторінка, на яку буде вставлено гнучкі Bootstrap-класи (`card`, `mb-3`, `d-flex`, `justify-content-between`), які дозволяють адаптувати зовнішній вигляд під різні типи користувачів без дублювання структури. Такий підхід дозволяє створити єдину сторінку профілю з динамічним вмістом, що забезпечує зручність в обслуговуванні та масштабуванні.

На рівні бекенду реалізовано перевірку прав доступу до сторінки та окремих блоків редагування, що унеможливорює перегляд або зміну даних неавторизованими користувачами.

Завдяки гнучкій структурі та модульному підходу сторінка легко масштабується для нових ролей або додавання нових блоків, не порушуючи існуючу архітектуру.

Інформація на сторінці профілю розділена на логічні блоки: Основна інформація, Графік активності, Примітки/Покарання, Лог операцій. Це спрощує навігацію, сприйняття даних та покращує UX.

The screenshot displays a user profile page for 'Vasi_Malyuk'. The page is organized into several sections:

- Profile Header:** Includes a profile picture of 'Jesus_Cooper' and the name 'Vasi_Malyuk — Профіль'.
- Statistics Table:**

Організація:	СБУ
Discord:	VasiMalyuk
Онлайн:	24:28:29
Днів на посаді:	1
Догани:	0 / 3
Попередження:	0 / 5
Бали:	0
Останнє оновлення:	2025-06-07 23:16:08
- Activity Graph:** A bar chart showing online time for each day of the week:

Понеділок	Вівторок	Середа	Четвер	П'ятниця	Субота	Неділя
05:36:42	05:26:48	03:08:57	00:00:00	05:05:57	03:16:07	01:53:58
- Notes/Punishments:** A section titled 'Примітки/Покарання' with a 'Редагувати' button and a sub-section for 'Інформація відсутня'.
- Operation Log:** A list of system events:
 - 2025-06-09 19:55:33: Змінено догани: 0 → 1 (Причина: test) - Jesus_Cooper
 - 2025-06-09 19:55:33: Змінено попередження: 0 → 2 (Причина: test) - Jesus_Cooper
 - 2025-06-09 19:55:33: Змінено бали: 0 → 500 (Причина: test) - Jesus_Cooper
 - 2025-06-08 14:31:31: Автоматичне оновлення: онлайн +05:05:57 (Friday) - system
 - 2025-06-08 14:31:31: Автоматичне оновлення: онлайн +03:16:07 (Saturday) - system
 - 2025-06-08 14:31:31: Автоматичне оновлення: онлайн +01:53:58 (Sunday) - system

Рис. 2.18 - Зовнішній вигляд “leader_profile.php”

Розглянемо кожен розділ окремо: Клас «card» кожен логічний блок (наприклад, інформація про користувача, групу, активність) обгортається в

компонент `card`, що забезпечує послідовне оформлення з відступами, фоном і рамкою.

Клас «`mb-4`» Забезпечує відступи між блоками `card`, завдяки чому сторінка не виглядає перевантаженою.

Використання «`table table-dark table-striped`» Дані про організацію, Discord, онлайн, догани, попередження, бали та інше представлені у форматі таблиці з темним стилем і чергуванням рядків. Це не тільки відповідає загальному стилю сторінки, а й покращує читабельність. Зовнішній вигляд представлений на рис. 2.19.

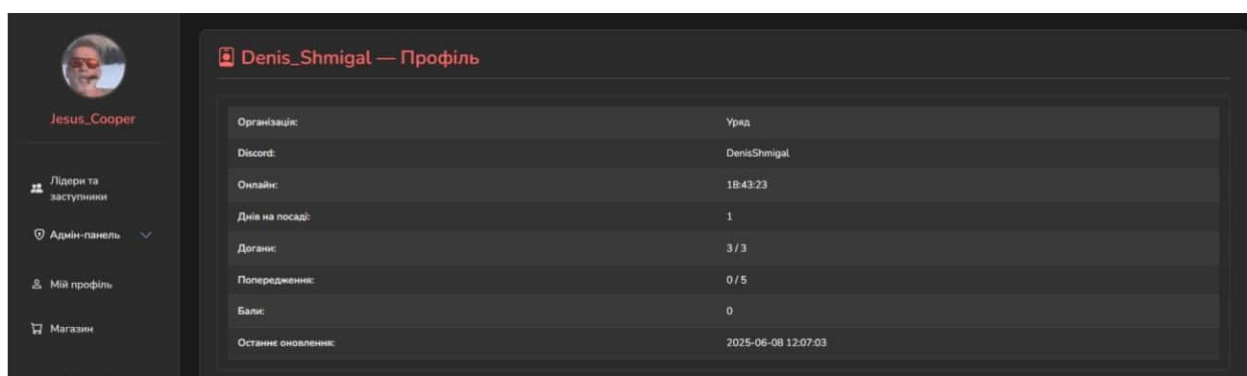


Рис. 2.19. - Зовнішній вигляд блоку, який містить дані про організацію, Discord, онлайн, догани, попередження

Клас «`card`» Блок «Графік активності» оформлений як окрема картка. Це дозволяє візуально відокремити графік від іншої інформації профілю, створюючи чітку структуру сторінки. Компонент «`card`» забезпечує фон, рамку та внутрішні відступи, що підвищує загальну читабельність.

Клас «`card-header`» Використовується для заголовка картки з піктограмою календаря. Додає контекст до вмісту та підсилює зручність навігації.

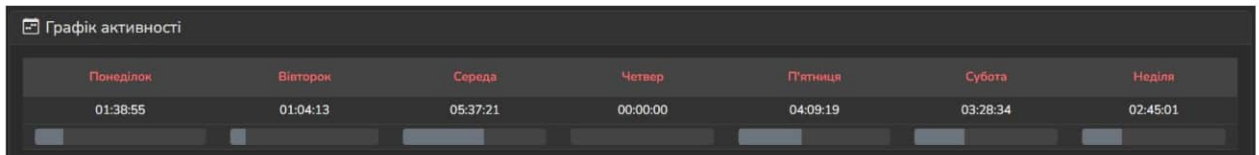


Рис. 2.20. - Зовнішній вигляд блоку часу проведене у грі по дням.

Клас «card-body» та «notes-content» Основний вміст блоку виводиться у «card-body», а «notes-content» використовується як обгортка для динамічного тексту. Текст приміток обробляється через `htmlspecialchars` і `nl2br`, що забезпечує безпеку та збереження форматування (переноси рядків).

Поводження з відсутніми даними: Якщо поле `notes` не заповнено, виводиться повідомлення «Інформація відсутня», що покращує UX - користувач не бачить порожнього місця.

Цей блок показує приклад контекстно-залежного виведення контенту з можливістю редагування лише для визначених ролей. Його структура дозволяє легко розширити функціональність - наприклад, додати історію змін або лог редагувань.

Кнопка редагування (`btn btn-sm btn-outline-warning`) відображається лише для користувачів з ролями `admin` або `moderator` через перевірку:

```
<?php if (in_array('admin', $user_roles) || in_array('moderator', $user_roles)): ?>
```

Рис. 2.21 - Перевірка на роль для редагування блоку Примітки/Покарання

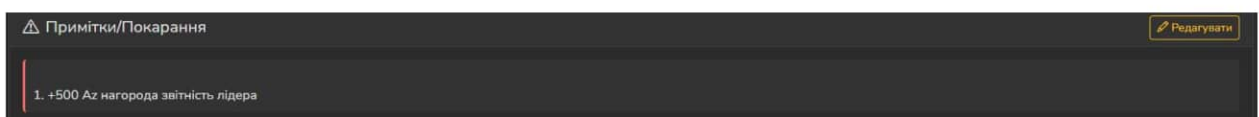


Рис. 2.22 - Зовнішній вигляд Примітки/Покарання

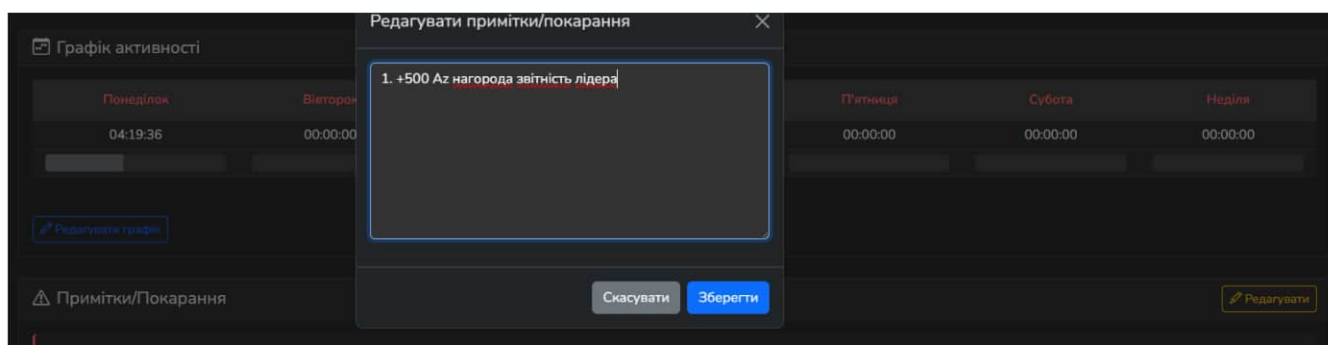


Рис. 2.23 - Модальне вікно редагування блоку примітки/покарання.

Клас «card» Блок журналу операцій реалізований як окрема картка, яка логічно відокремлює історію дій над профілем. Це дозволяє відстежувати змінення, які були здійснені адміністраторами або модераторами, не змішуючи їх з основною інформацією про користувача.

Клас «card-header» Заголовок блоку містить піктограму «bi-journal-text» це підсилює семантику елемента, одразу вказуючи користувачу, що тут міститься хронологічний лог подій.

Запит до бази даних (leader_logs) Виводиться журнал змін для конкретного лідера на основі leader_id, відсортований за датою додавання (ORDER BY created_at DESC), що гарантує актуальність записів згори.

Клас «list-group list-group-flush» Кожен запис логів виводиться як окремий елемент списку, без зайвих меж між елементами (flush). Це створює компактний та читабельний вигляд.

Використання «d-flex justify-content-between» в елементі «list-group-item» Кожен запис логів має чітке розділення дати/часу та імені адміністратора по краях блоку, що покращує сприйняття інформації.

Цей блок дозволяє адміністраторам переглядати дії в профілі у зручному форматі, забезпечуючи як прозорість змін, так і можливість аудиту.

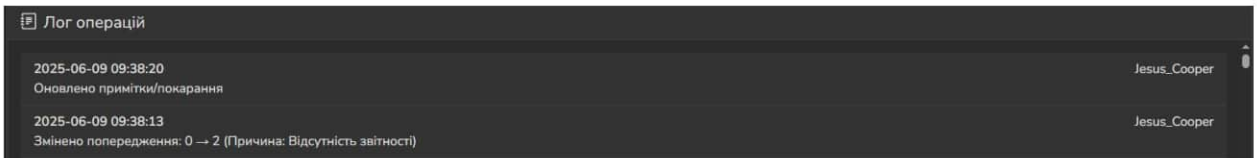


Рис. 2.24 - Зовнішній вигляд лог операцій.

Також на профілі лідера чи заступника з роллю модератора чи адміністратора є кнопки Редагувати та Зняти. Зовнішній вигляд представлений на рис. 2.25/2.26. Якщо натиснути на кнопку редагувати, відкриється окрема сторінка редагування (`edit_leader.php`)

Зовнішній вигляд представлений на рис. 2.27. Все це, дозволяє дуже швидко та зручно редагувати анкету. За допомогою SELECT-запиту завантажуються поточні значення лідера. Та порівнюються нові та старі дані - якщо є зміни, необхідно вказати причини.

Для кожної зміни потрібно вказувати причину, що підвищує відповідальність та прозорість дій адміністрації, з автоматичним записом у лог операцій.

А якщо натиснути на кнопку зняти, відкриється модальне вікно, де обов'язково вказати причину, а також поставити галочку “Підтверджую, що хочу зняти цього лідера”, тільки після цього автоматично лідер чи заступник буде видалений з таблиці лідерів, та занесений в архівний лист.(Рис. 2.25)

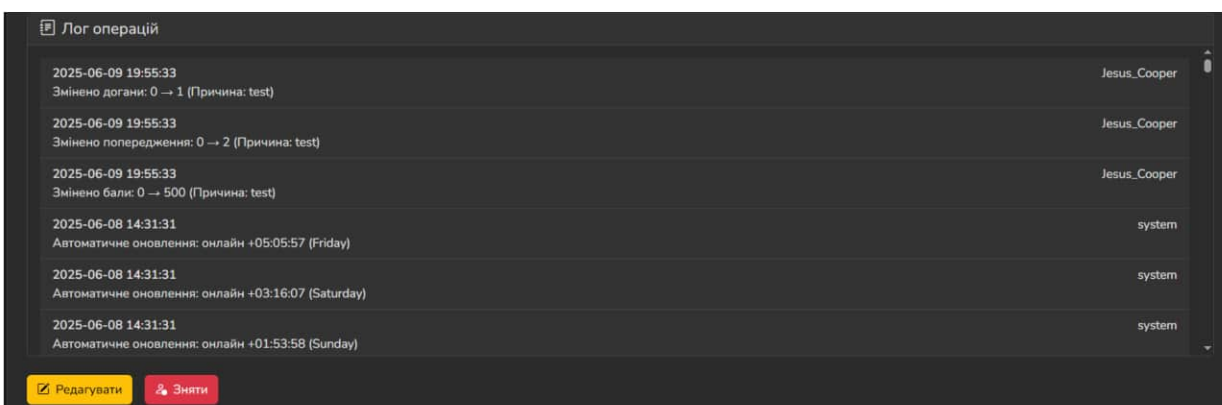


Рис. 2.25 - Кнопки редагувати дані лідера чи заступника та кнопка зняти

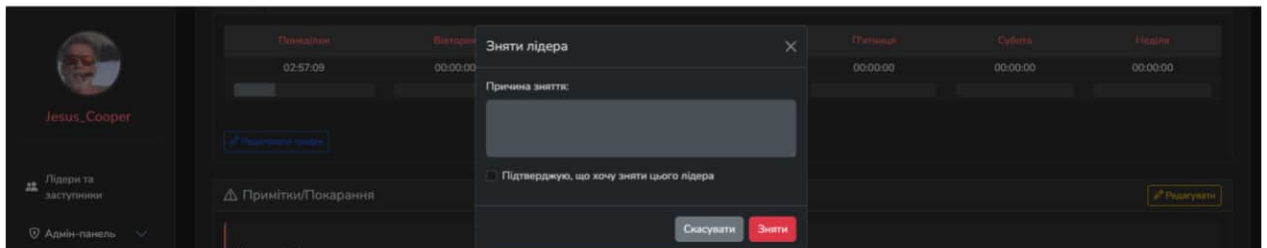


Рис. 2.26- Модальне вікно зняття лідера чи заступника

Редагувати: Denis_Shmigal

Фракція
 Уряд
 Причина зміни фракції

Роль
 Лідер

Discord
 DenisShmigal

Онлайн (години)
 67403

Догани
 3
 Причина зміни доган

Попередження
 2
 Причина зміни попереджень

Бали
 0
 Причина зміни балів

← Назад до профілю **Зберегти зміни**

Рис. 2.27-Сторінка редагування лідера чи заступника.

2.2.4 Сторінка додавання лідера/заступника

Сторінка додавання лідера/заступника представляє зручну вебформу, яка дозволяє адміністраторам чи модераторам вносити основну інформацію про нового керівника організації. Після заповнення форми відбувається створення нового профілю, що згодом буде відображатися на сторінках `leader_profile.php`, `index.php` та `profile.php`. Форма реалізована за допомогою класичних елементів `<form>`, з використанням Bootstrap-класів для стилізації:

- «`form-group`», «`mb-3`» - забезпечують акуратні відступи між полями.
- «`form-control`» - уніфікований вигляд для всіх полів вводу.
- «`input`», «`textarea`», «`select`» - різні типи полів для збирання інформації.

Форма включає наступні обов'язкові для заповнення поля:

- Нікнейм (`nickname`)
- Організація (`organization`)
- Discord (`discord`)
- Посада (`position`, зазвичай "лідер" або "заступник")

На стороні клієнта реалізована базова HTML5-валідація. Поля містять атрибут `required`, що унеможлиблює відправку порожньої форми.

Форма надсилається методом POST. Далі скрипт:

- Отримує дані з полів за допомогою `$_POST`
- Виконує додаткову перевірку та фільтрацію (наприклад, через `htmlspecialchars`, `trim`)
- Перевіряє, чи існує вже введений нікнейм у таблиці `leaders`, використовуючи SQL-запит типу `SELECT`. Якщо такий запис вже є, користувачу виводиться повідомлення про помилку та запобігається повторному додаванню Рис. 2.31.

- Якщо нікнейм унікальний, формується SQL-запит для додавання нового запису в таблицю leaders.

Після обробки форми користувачу може бути показано:

- Повідомлення про успішне створення профілю
- Повідомлення про помилку (наприклад, дублювання нікнейму, некоректні дані)

Після створення профілю, новий лідер зберігається у базі даних і одразу доступний для перегляду через leader_profile.php. Це дозволяє адміністраторам у реальному часі керувати складом керівництва, оновлювати інформацію та вести облік активності.

Рис.2.28 - Зовнішній вигляд add_leader_form.php (Додавання нового лідера чи заступника).

Рис. 2.29 - Помилка валідації

2.2.5 Сторінка архіву лідерів/заступників з інформацією коли і хто зняв лідера/заступника

Сторінка `archive.php` є функціональним компонентом вебзастосунку, що відповідає за перегляд та обробку даних колишніх лідерів і заступників організації. Вона надає адміністраторам можливість зручно управляти історією керівного складу, зберігаючи повну інформацію про попередніх учасників.

Головною метою сторінки є відображення списку користувачів, які були переведені в архів, з можливістю перегляду, фільтрації.

Завдяки файлу `remove_leader.php`, здійснюється видалення та автоматичне перенесення лідера чи заступника в архів.

Він також веде журнал дій адміністратора. Сценарій виконується після підтвердження видалення через POST-запит з відповідними даними.

Усі дані користувача переносяться в таблицю `archived_leaders` для подальшого зберігання. Таким чином реалізується "м'яке видалення", що дозволяє зберегти важливу історичну інформацію.

```
$stmt = $conn->prepare("INSERT INTO archived_leaders
(nickname, organization, ld, discord, last_update, online_hours, strikes, warnings, points, reason, deleted_by, deleted_at)
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
$stmt->bind_param(
"ssssiiiiiss",
$leader['nickname'],
$leader['organization'],
$leader['ld'],
$leader['discord'],
$leader['last_update'],
$leader['online_hours'],
$leader['strikes'],
$leader['warnings'],
$leader['points'],
$reason,
$deleted_by,
$deleted_at
);
$stmt->execute();
```

Рис. 2.30 - Перенесення даних в архів

```
$stmt = $conn->prepare("DELETE FROM leaders WHERE id = ?");
$stmt->bind_param("i", $leader_id);
$stmt->execute();
```

Рис. 2.31 - Видалення лідера чи заступника

Цей PHP-скрипт виконує багатоступеневу обробку видалення користувача:

- Перевірка прав доступу.
- Отримання та перевірка даних.
- Запис дій у лог.
- Перенесення запису до архіву.
- Фізичне видалення з основної таблиці.

Такий підхід гарантує збереження даних, аудит подій, і безпечну роботу з системою керування лідерами.

Вивід реалізовано за допомогою стандартної HTML-таблиці, стилізованої за допомогою Bootstrap (table, table-bordered, table-striped).

Таблиця відображає:

- нікнейм користувача;
- назву організації;
- контакт у Discord;
- посаду (лідер або заступник);
- дату архівації.

The screenshot shows a web interface for 'Архів лідерів' (Leaders Archive). On the left is a sidebar with the user profile 'Jesus_Cooper' and navigation links: 'Лідери та заступники', 'Адмін-панель', 'Мій профіль', 'Магазин', and 'Вийти з системи'. The main content area has a search bar and filters for 'Організація' (Organization) and 'Роль' (Role), both set to 'Всі' (All). Below the filters is a table with the following data:

Нікнейм	Організація	Роль	Discord	Онлайн	Догани	Попередження	Бали	Причина	Видалення(на)	Дата видалення	Дії
123	SWAT	Лідер	123	0 год.	0	0	0	123	Serj_Rosati	2025-06-08 19:07:40	Відновити Редагувати
123	Центр Ліцензування	Лідер	132	0 год.	0	0	0	123	Serj_Rosati	2025-06-08 19:06:50	Відновити Редагувати
Serj_Rosati	RCSD	Лідер	Serj_Rosati	0 год.	6	0	0	123	Jesus_Cooper	2025-06-08 19:03:24	Відновити Редагувати

Рис. 2.32 - Зовнішній вигляд сторінки archive.php

Сторінка отримує дані з бази даних за допомогою SQL-запитів типу SELECT, з умовою вибору лише записів зі статусом архівних. Параметри фільтрації можуть бути передані через GET.

Для виконання операцій редагування або відновлення профілів доступ надається лише авторизованим адміністраторам. Перевірка здійснюється через сесію або спеціальний механізм автентифікації. Можливість редагування або відновлення запису

Для кожного профілю передбачено кнопку «Редагувати» або «Відновити», яка перенаправляє адміністратора на форму зміни даних.

Відновлення профілю реалізується або через UPDATE, або через переміщення запису з архівної таблиці до основної.

Користувачеві виводиться повідомлення про успішне виконання дії (наприклад, «Профіль відновлено»)

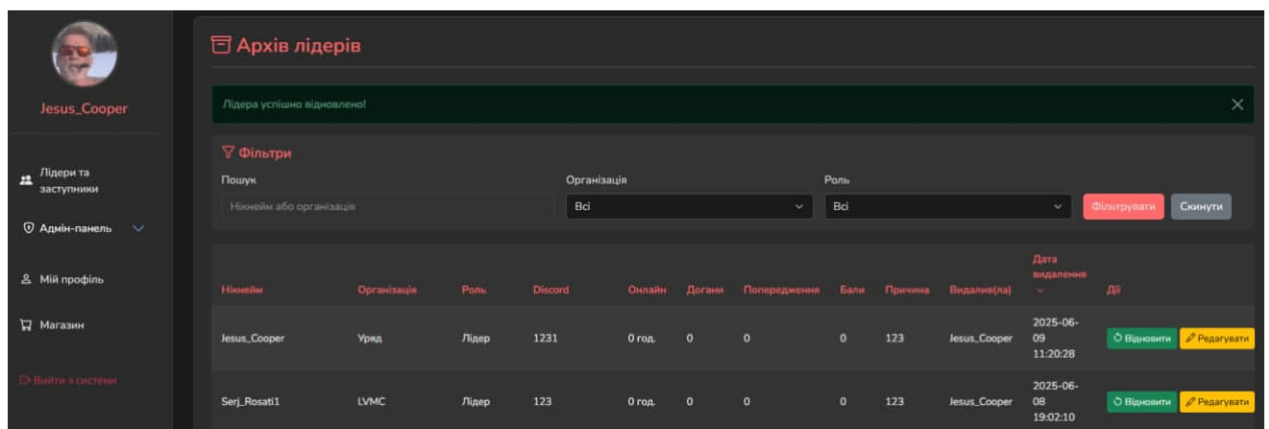


Рис. 2.33 - Повідомлення при успішном відновленю лідера чи заступника

Такий підхід дозволяє адміністраторам зручно керувати архівними записами, не порушуючи логіки роботи основного застосунку.

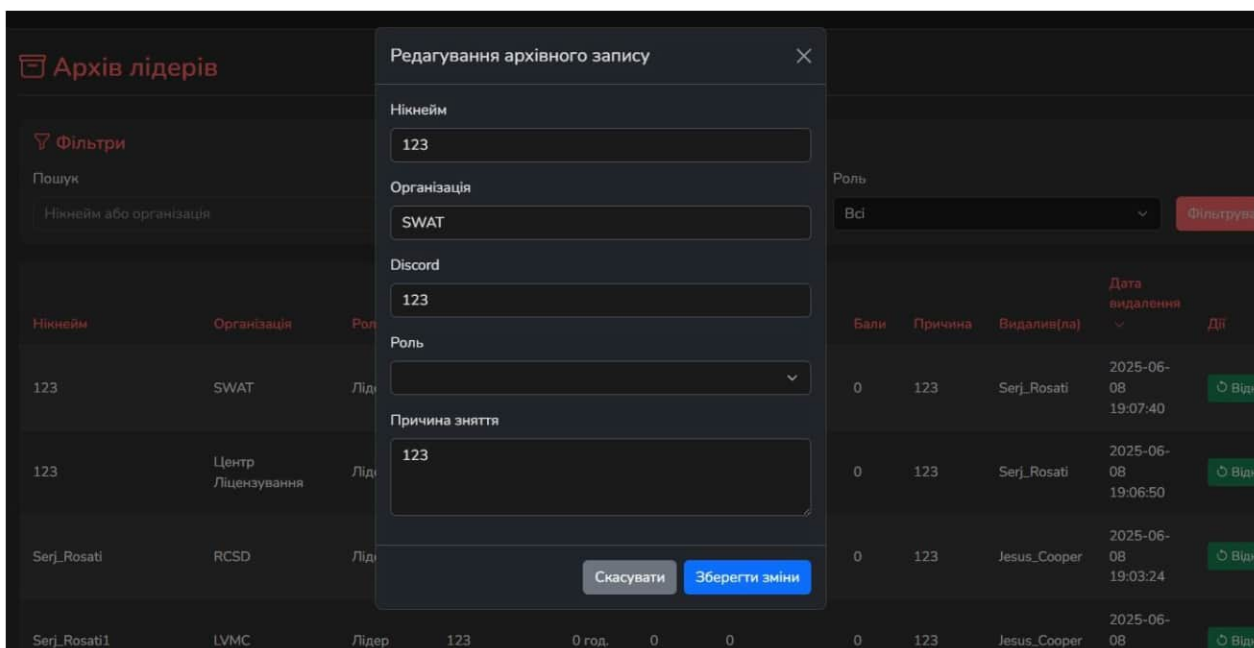


Рис. 2.34 - Модальне вікно редагування архівного запису.

Функціонал:

- Сортування та фільтрація записів у таблиці.
- Пошук за нікнеймом або організацією.
- Адаптивність інтерфейсу - сторінка підтримує перегляд як на десктопах, так і на мобільних пристроях завдяки використанню Bootstrap.

Сторінка “archive.php” відіграє важливу роль у забезпеченні повноцінного управління життєвим циклом користувачів системи. Вона дозволяє вести облік попередніх керівників, зберігаючи історичну інформацію про зміни у структурі організації. Завдяки реалізації зручного інтерфейсу та перевірок доступу, архівна система є безпечною та ефективною в експлуатації.

2.2.6 Сторінка Чорного списку лідерів/заступників.

Сторінка “blacklist.php” забезпечує адміністративне управління чорним списком користувачів(). Доступ до сторінки мають лише користувачі з ролями “admin” або “moderator”.

- Якщо користувач не авторизований - виконується редирект на login.php.
- Якщо користувач не має ролі admin або moderator - доступ забороняється.

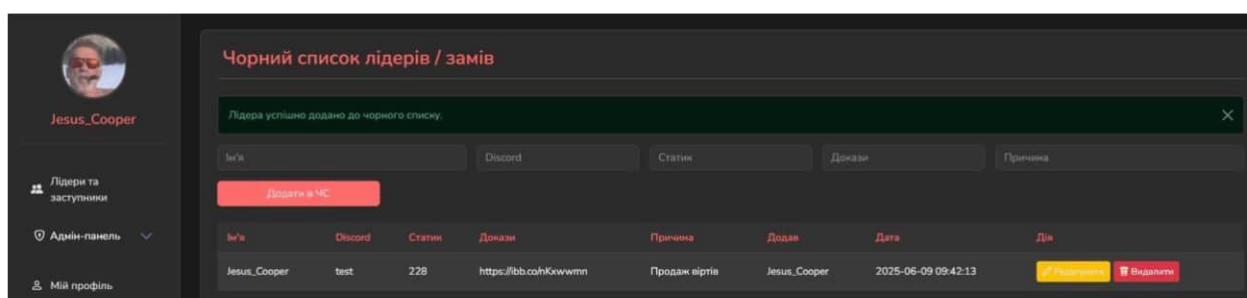


Рис. 2.35 - Зовнішній вигляд сторінки blacklist.php

Після успішного додавання або редагування запису виводиться повідомлення, збережене у сесії (\$_SESSION['success']).

Через POST-запит з параметром add_blacklist можна додати нового учасника чорного списку.

Зберігається наступна інформація:

- Ім'я (name)
- Discord-нік (discord)
- Статик (інше поле ідентифікації, statik)
- Докази (proof)
- Причина (reason)
- Хто додав (added_by, із сесії)

Також на цій сторінці, є можливість редагувати існуючі записи, завдяки файлу edit_blacklist.php(рис.2.36). Поля аналогічні, як і при створенні.

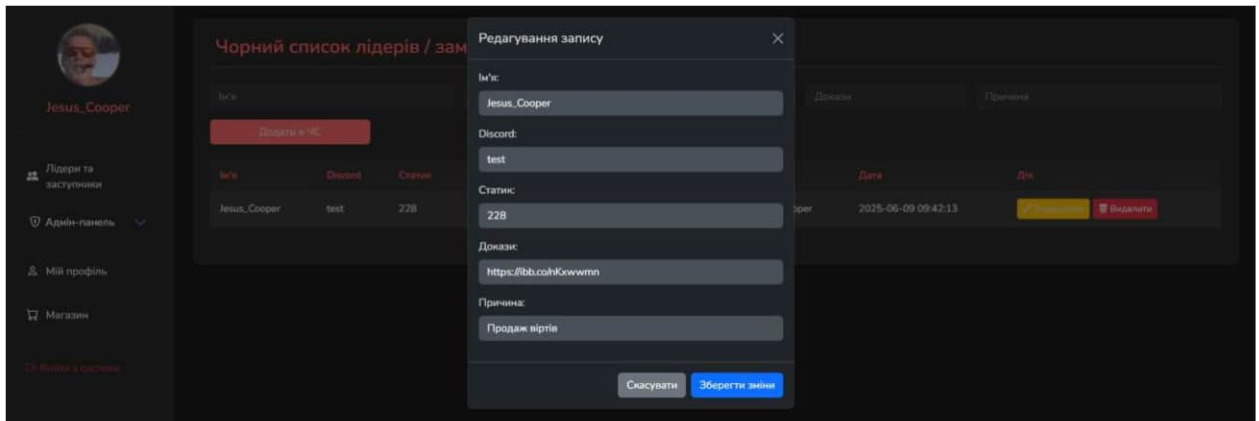


Рис. 2.36 - Редагування анкети чорного списку.

Також на цій сторінці, є можливість видалити з чорного списку, та перенесення в архів чорного списку(Зовнішній вигляд представлено на Рис. 2.37).

Якщо надійшов POST-запит з параметром “remove_blacklist”, система:

- Знаходить запис по “id”
- Копіює дані цього запису в таблицю blacklist_archive, додаючи:
- Хто видалив (removed_by)
- Причину видалення (removal_reason)
- Потім видаляє цей запис із основної таблиці “blacklist”

Після дії користувачу показується повідомлення про успішне переміщення в архів.

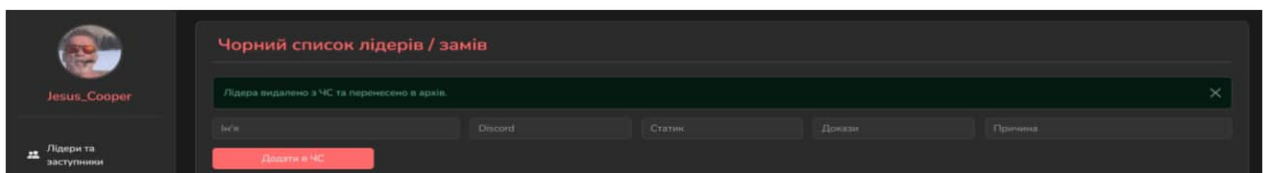


Рис. 2.37 - Повідомлення про успішне переміщення в архів чорного списку.

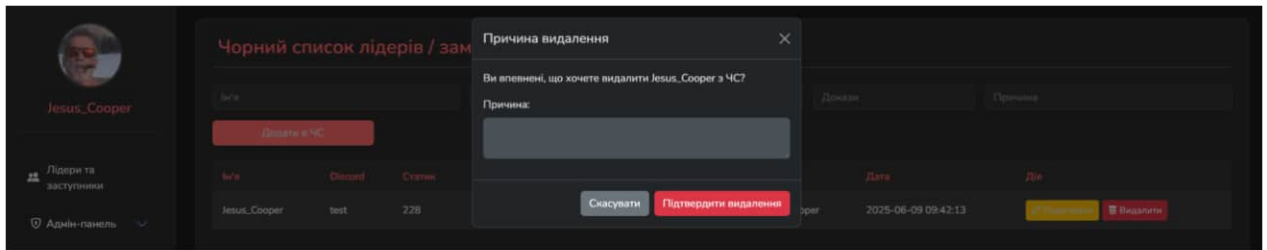


Рис. 2.38 - Модальне вікно видалення с таблиці чорного списку.

2.2.7 Сторінка Архіву чорного списку лідерів/заступників

Сторінка “archive_blacklist.php” призначена для перегляду архіву користувачів, які раніше були внесені до чорного списку і пізніше видалені з нього. Сторінка є частиною адміністративної панелі вебзастосунку моніторингу активності користувачів ігрового серверу. Вона доступна лише для користувачів з ролями адміністратора або модератора.

Дана сторінка забезпечує безпечний, зручний і функціональний перегляд історії чорного списку користувачів з можливістю контролю доступу до сторінки та приємним інтерфейсом.

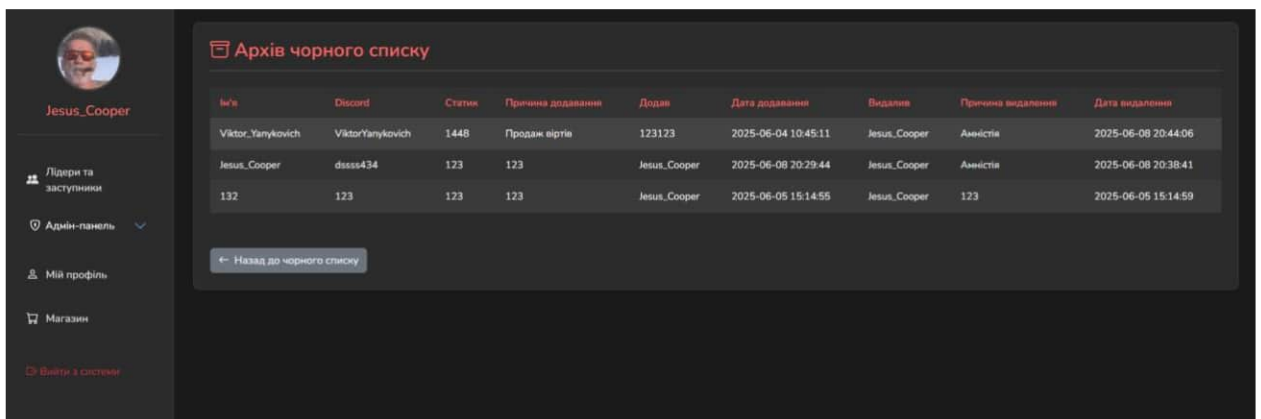


Рис. 2.39 - Зовнішній вигляд сторінки “archive_blacklist.php”

Таблиця містить такі ж самі поля, як і в таблиці чорного списку.

2.2.8 Сторінка управління правами користувачів на сайті

Також для зручності на сайті є сторінка “Управління правами користувачів” `manage_users.php`.

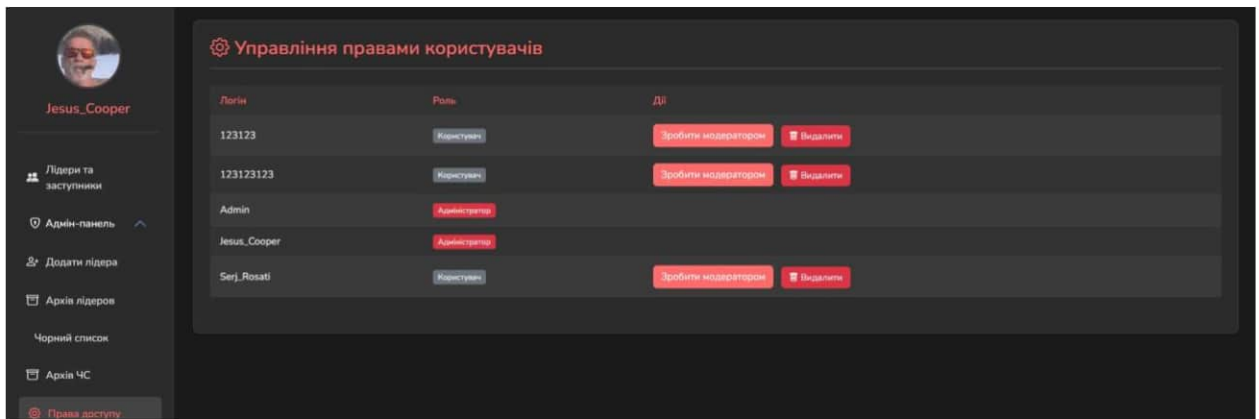


Рис. 2.40 - Зовнішній вигляд сторінки `manage_users.php` ”

Сторінка доступна лише користувачам із роллю `admin`. Усі інші отримують повідомлення: « У вас не має доступу до керування користувачами.»

На сторінці відображається таблиця з усіма зареєстрованими користувачами. Для кожного користувача показується:

- логін;
- поточна роль (`admin`, `moderator`, `user`);
- доступні дії.

Адміністратор може:

- надати роль модератора користувачу;
- забрати права модератора (зробити звичайним користувачем).

Зміна ролей реалізована через POST-форми, що надсилають відповідну дію (`grant`, `revoke`) на сервер. Серверна частина обробляє запит і оновлює поле `roles` у таблиці `users`.

Адміністратор може видалити будь-якого користувача, крім інших адміністраторів.

2.2.9 Сторінка власного профілю користувача

В вебзастосунку є можливість переглянути свій же профіль.

- Отримується ім'я користувача та список ролей із сесії.
- Перевіряється, чи є користувач у таблиці leaders за його нікнеймом.
- Якщо користувач знайдений у списку лідерів, його автоматично перенаправляють на детальну сторінку лідера leader_profile.php.
- Якщо користувача не знайдено, тоді відкривається сторінка профілю з короткою інформацією:
 - Ім'я користувача
 - Ролі
 - Статус(Зовнішній вигляд сторінки представлений на рис. 2.41)

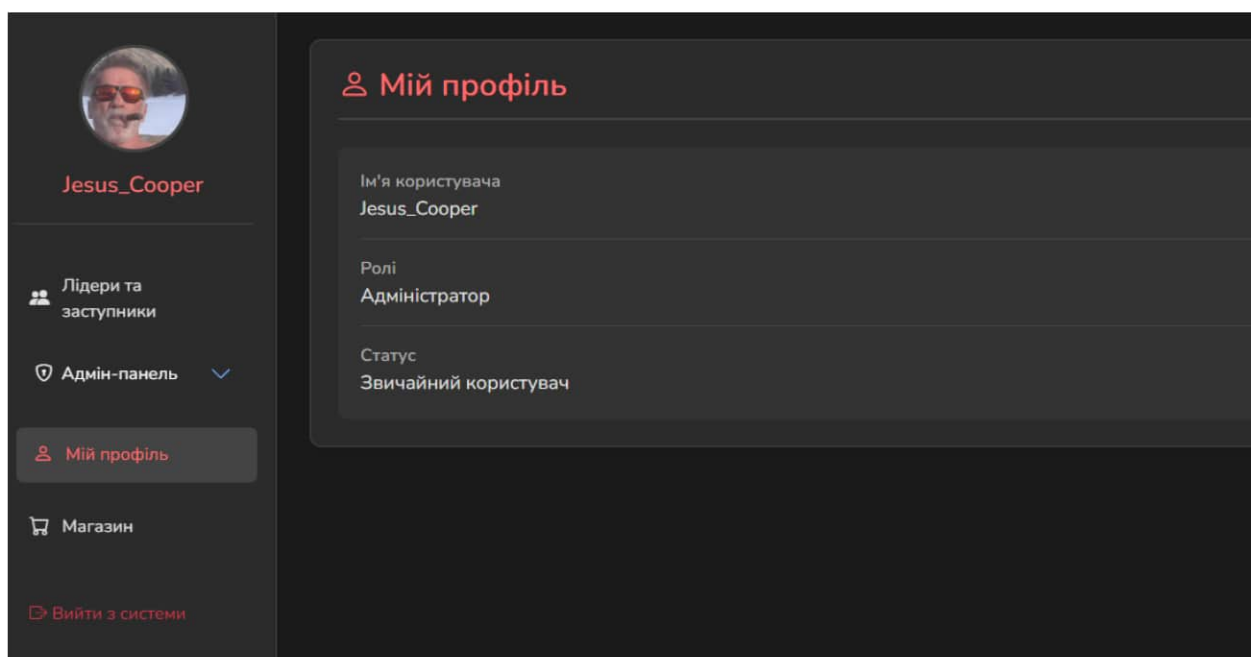


Рис. 2.41 - Зовнішній вигляд “profile.php”

2.2.10 Сторінка магазину з зняттям доганів та попередження

Сторінка “Магазин” створена для зручності лідерам та заступникам знімати догани та попередження. Є можливістю додавання, редагування, видалення товарів (для адміністраторів та модераторів), а також їх перегляду для звичайних користувачів.

- Ролі користувача отримуються із сесії та розбиваються в масив.
- Зберігається нікнейм користувача для подальших операцій.
- Доступ до адмін-функцій мають лише ролі admin та moderator.

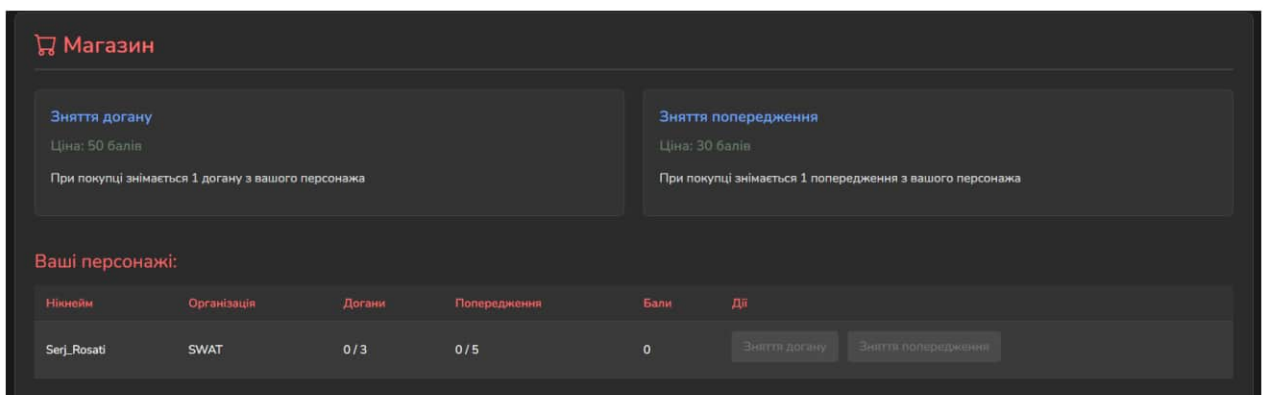


Рис. 2.42 - Сторінка магазину(зовнішній вигляд для лідерів та заступників)

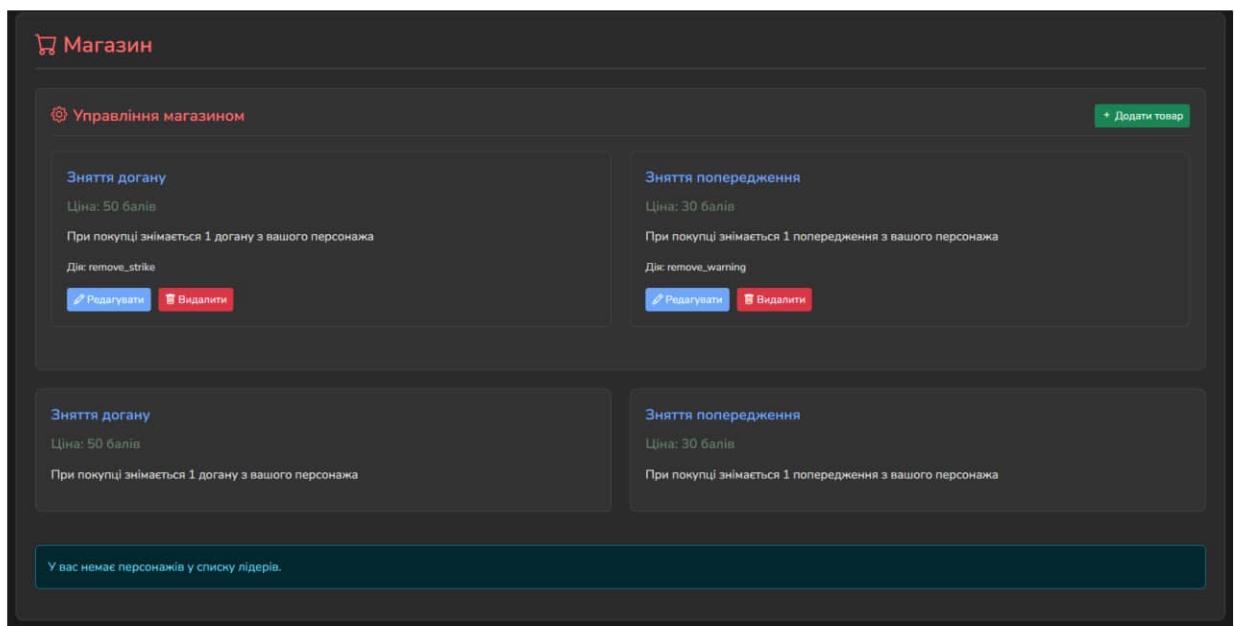


Рис. 2.43 - Сторінка магазину(зовнішній вигляд з роллю admin та moderator)

- Якщо користувач має права адміністратора або модератора і надіслав форму, запускається обробка введених даних.
- Перевіряються обов'язкові поля: назва товару, ціна, дія.

Якщо дані коректні:

- Якщо `item_id = 'new'` - додається новий товар у таблицю `shop_items`.
- Інакше - оновлюється вже існуючий товар.
- Якщо у користувача немає в таблиці лідерів та заступників то буде повідомлення “У вас немає персонажів у списку лідерів”

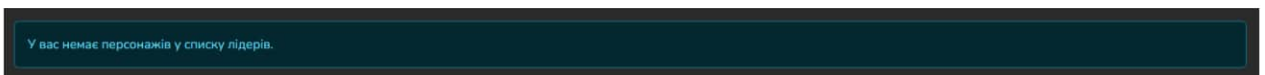


Рис. 2.44 - Повідомлення що користувача немає у списку лідерів.

 A dark-themed modal window titled "Редагувати товар" (Edit item) with a close button (X) in the top right corner. It contains four input fields:

- "Назва товару *" (Item name) with the value "Зняття догану" (Strike removal).
- "Ціна (в балах) *" (Price in points) with the value "50".
- "Опис товару" (Item description) with the value "При покупці знімається 1 догану з вашого персонажа" (1 strike is removed from your character upon purchase).
- "Дія (системне ім'я) *" (Action/system name) with the value "remove_strike".

 At the bottom, there are two buttons: "Скасувати" (Cancel) and "Зберегти зміни" (Save changes).

Рис. 2.45 - Модальне вікно редагування товару

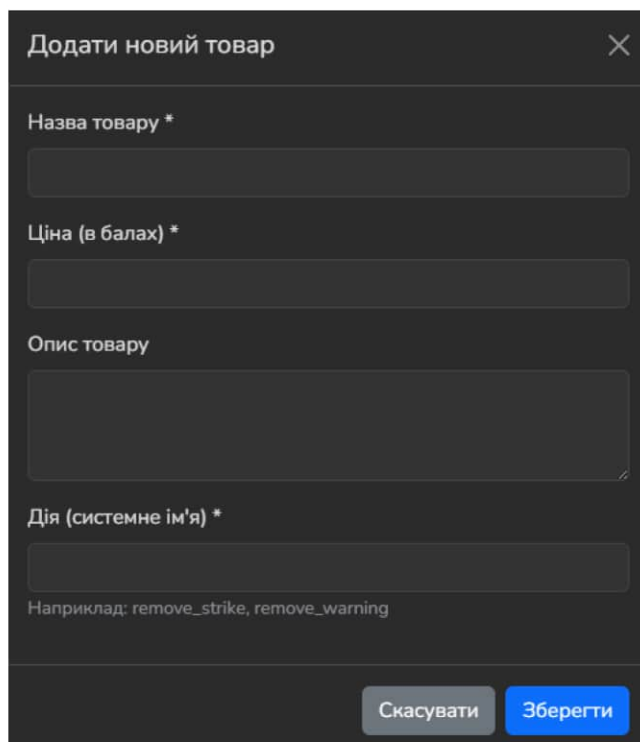
Після натискання кнопки "Зберегти" форма надсилається методом POST, де на сервері обробляється:

- Перевірка прав (чи має користувач роль admin/moderator),
- Валідація введених даних,
- Оновлення або додавання товару до бази даних.

Вікно з'являється поверх сторінки, не перезавантажуючи її повністю.

Після збереження користувача повертає назад на ту ж сторінку.

- Використовується Bootstrap-модальне вікно (`<div class="modal fade" id="editModal">`).
- Перед відкриттям - через JS вставляються дані в поля форми.
- Форма містить приховане поле `item_id` для ідентифікації товару.
- Після надсилання форму обробляє PHP-код у верхній частині `shop.php`.



Додати новий товар

Назва товару *

Ціна (в балах) *

Опис товару

Дія (системне ім'я) *

Наприклад: remove_strike, remove_warning

Скасувати Зберегти

Рис. 2.46 - Модальне вікно додавання товару

2.2.11 Створення скрипту для автоматичного підрахунку онлайн

Для обробки лог-файлів було створено простий парсер на мові PHP.

Основною задачею парсеру є зчитування файлу журналу подій, розбір кожного рядка та виділення з нього ключових елементів: підрахування онлайн

На сайті реалізована автоматична система підрахунку онлайн, це реалізовано за допомогою файлу “log_processor.php”

Цей файл робить:

- Скрипт постійно перевіряє наявність нового файлу з логами.
- Після появи лог-файлу скрипт відкриває його та читає вміст построчно.
- Зчитує лог-файл, який містить інформацію про гравців, що виходять із сервера.
- Парсить (розбирає) кожен рядок лог-файлу.
- Визначає нікнейм гравця, час сесії в секундах та дату.
- Зберігає дані у вигляді масиву для подальшого використання.
- Має додаткову функцію для перетворення часу з секунд у формат гг:хх:сс

Парсер реалізовано у вигляді функції parseLogFile, яка приймає шлях до файлу як параметр, перевіряє його існування, а далі опрацьовує кожен рядок за допомогою регулярного виразу. Логіка побудована на припущенні, що кожен рядок журналу має структуру [Дата] [НикНейм] [Час] [День Тиждня] повідомлення

На приклад: 2025-05-01 14:33:01 Гравець ****_**** вийшов з сервера, час сесії: 5:13:11 Monday

```
function parseLogFile($logfile) {
    if (!file_exists($logfile)) {
```

Рис. 2.47 - Функція яка виконує основну логіку обробки лог-файлу

```
$logs = file($logfile, FILE_IGNORE_NEW_LINES | FILE_SKIP_EMPTY_LINES);
```

Рис. 2.48 - Перевіряється наявність файлу. Якщо його немає - повертається порожній масив.

```
foreach ($logs as $log) {
    if (preg_match('/^\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}) Гравець (\w+)_\w+ вийшов з сервера, час сесії: (\d+):(\d+):(\d+) (\w+)$/', $log, $matches
```

Рис. 2.49 - Зчитуються всі рядки з файлу в масив \$logs.

```
"/^\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}) Гравець (\w+)_\w+ вийшов з сервера, час сесії: (\d+):(\d+):(\d+) (\w+)$/', $log, $matches
```

Рис. 2.50 - Регулярний вираз

Цей вираз витягує:

- дату та час (\$matches[1])
- нікнейм гравця (об'єднання \$matches[2] та \$matches[3])
- години, хвилини, секунди
- день тижня

Потім ці значення конвертуються у загальну кількість секунд.

```
$totalSeconds = $hours * 3600 + $minutes * 60 + $seconds;
```

Рис. 2.51 - конвертування у загальну кількість секунд

secondsToTime(\$seconds) - конвертує загальну кількість секунд у формат годин:хвилин:секунд

На основі цього парсера можна:

1. Записувати дані у базу даних.
2. Видаляти оброблений лог-файл після обробки.
3. Очікувати на появу нового лог-файлу (наприклад, в циклі або через стоп-скрипт).

4. Видаляти застарілі записи з бази та записувати нові, забезпечуючи актуальність даних.

2.3 Розробка бази даних

Вся структура бази даних побудована таким чином, щоб забезпечити гнучкість, надійність зберігання історичних даних, контроль прав доступу та можливість масштабування системи без зміни основної логіки. Завдяки поділу на активні та архівні таблиці, інформація ніколи не втрачається, а можливість ведення логів усіх змін забезпечує прозорість у роботі модераторів та адміністраторів.

Таблиця `leaders`:

Містить інформацію про поточних лідерів організацій з такими полями:

- `id` - унікальний ідентифікатор
- `organization` - організація/фракція
- `nickname` - нікнейм лідера
- `online_hours` - кількість годин онлайн
- `discord` - Discord лідера
- `strikes` - кількість доганів
- `warnings` - кількість попереджень
- `points` - бали лідера
- `last_update` - дата останнього оновлення
- `ld` - тип лідера (`leader/zam`)
- `notes` - примітки
- Поля для обліку годинника по днях тижня (`monday_hours`, `tuesday_hours` и т.д)

Таблиця `archived_leaders`.

Містить архів віддалених лідерів з аналогічними полями як і таблиця `leaders` плюс:

- deleted_by - хто видалив
- deleted_at - коли видалено
- reason - причина видалення

Таблиця blacklist.

Для чорного списку з полями:

- name - ім'я
- discord - Discord
- statik - статик
- reason - причина
- added_by - хто додав
- date_added - дата додавання
- proof - докази
- edited_by - хто редагував
- edit_date - дата редагування

Таблиця blacklist_archive.

Архів видалених із чорного списку з полями:

- removed_by - хто видалив
- removal_reason - причина видалення
- date_removed - дата видалення

Таблиця leader_logs.

Лог змін лідерів із полями:

- leader_id - ID лідера
- admin_username - хто вніс зміни
- action - опис дії
- created_at - дата зміни

Таблиця users:

Містить інформацію про зареєстрованих користувачів сайту.

- username - Ім'я користувача

- password - пароль користувача
- created_at - дата створення акаунту
- roles - Роль користувача на сайті

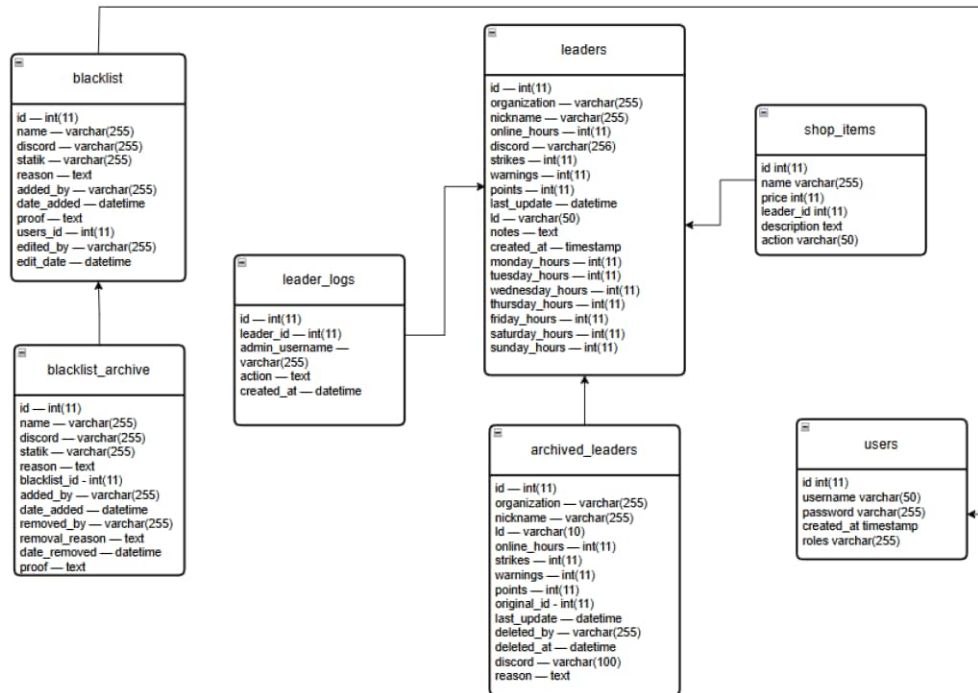


Рис 2.52 - Схема бази даних

Переваги:

1. Гнучкість: Просте додавання нових ролей, полів або функціоналу без зміни загальної логіки.
2. Масштабованість: Розмежування даних дає змогу розподіляти навантаження між таблицями.
3. Прозорість: Повне логування змін створює систему звітності та довіри.
4. Збереження історії: Архівні таблиці забезпечують зберігання всіх даних навіть після видалення з активного реєстру.

2.4 Висновки до другого розділу

У другому розділі було розглянуто ключові проектні рішення щодо розробки вебзастосунку для моніторингу активності лідерів і заступників на ігровому сервері «Arizona RP». Основною метою створення цього застосунку є оптимізація та автоматизація адміністративних процесів, що дозволяє зменшити кількість помилок, підвищити ефективність керування та забезпечити зручність у щоденній модерації. Система забезпечує наступні переваги:

Спрощення адміністративних процесів:

1. Автоматизований моніторинг активності лідерів, ведення архівів, управління чорним списком та система штрафів/заохочень значно зменшують навантаження на адміністраторів.
2. Відмова від ручного ведення Google-таблиць або Discord-каналів підвищує ефективність роботи.

Зручний інтерфейс:

1. Використання Bootstrap забезпечує адаптивність та кросбраузерність.
2. Інтуїтивна навігація, бічна панель та візуальні індикатори (наприклад, червоні значки для проблемних записів) роблять систему зручною для користувачів.

Безпека та контроль доступу:

1. Реалізована авторизація та розмежування прав за ролями (адмін, модератор, користувач).
2. Захист від XSS-атак через функції типу `htmlspecialchars()`.
3. Логування дій адміністраторів для прозорості та аудиту.

Функціональність:

1. Система балів та магазин дозволяють лідерам знімати догани, що мотивує до активності.
2. Архіви зберігають історичні дані, що важливо для аналізу та відстеження змін.

3. Можливість швидкого редагування профілів, додавання нових лідерів та управління чорним списком.

Технічна реалізація:

1. Використання PHP та MySQL забезпечує стабільну роботу з даними.
2. Динамічне оновлення інформації (автоматичний підрахунок онлайн через `log_processor.php`).
3. Модульна структура сайту дозволяє легко розширювати функціонал.

ВИСНОВКИ

У цій роботі розглядається завдання автоматизації процесу роботи головного адміністратора на ігровому сервері "Arizona RP". Як рішення проблеми розроблено вебдодаток, який допоможе спростити безліч рутинних дій.

За результатами роботи досягнуто наступної мети: розроблений сайт, має багатосторінкову структуру, що дозволяє уникнути електронних таблиць та зберігати списки та інформацію про адміністраторів. У функціоналі сайту реалізована бальна система заохочень та покарань.

Розроблено скрипт перевірки активності лідерів та заступників.

Були виконані такі завдання:

1. Проаналізовано наявні ІТ рішення, які б могли допомогти у вирішенні проблеми. Відповідних продуктів, які могли б закрити всі потреби, знайдено не було.
2. Проаналізовано різноманітні інструменти розробки вебдодатків. Для написання програми було вибрано такі засоби: HTML, JavaScript, PHP, CSS, jQuery, jQuery UI, MySQL, Bootstrap

Визначено основні функції вебдодатку:

- Автоматизований моніторинг активності.
 - Сховище анкет лідерів та заступників.
 - Багаторівнева система доступу до даних.
 - Система чорного списку.
 - Архів чорного списку.
 - Інтерактивне представлення даних.
3. Спроектовано та розроблено інтерфейс, який забезпечує інтуїтивну взаємодію з системою, незалежно від пристрою користувача.

4. Реалізовано фронтенд. Для кожної сторінки написано HTML код, підключений дизайн та налаштовані необхідні бібліотеки та JS скрипти.
5. Зроблений бекенд. Розроблено бази даних. Для сторінки лідерів. Створено власний парсер логів, що забезпечує автоматизований підрахунок онлайн-активності користувачів.

Запропоноване рішення дозволяє значно зменшити навантаження на адміністративний персонал, уникнути помилок при веденні обліку вручну, а також покращити прозорість і контроль за активністю лідерів та заступників на ігровому сервері.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Battlemetric [Електрон.ресурс].Спосіб доступу:URL:
<https://www.battlemetrics.com/>
2. Raider.io [Електрон.ресурс].Спосіб доступу:URL:
<https://raider.io/UK/addon>
3. Python [Електрон.ресурс].Спосіб доступу:URL:
<https://foxminded.ua/de-vykorystovuietsia-python/>,
<https://goit.global.ua/articles/shcho-take-python-i-dlia-choho-vin-potriben>
3. JavaScript [Електрон.ресурс].Спосіб доступу:URL:
<https://goit.global.ua/articles/shcho-take-javascript-i-dlia-choho-vin-potriben/>
4. Керівництво PHP [Електрон.ресурс].Спосіб доступу: URL:
<https://www.php.net/manual/uk/index.php>,
<https://uk.wikipedia.org/wiki/PHP>
5. CMS [Електрон.ресурс].Спосіб доступу:URL:
<https://tutost.ua/uk/blog/cms>
6. SQL [Електрон.ресурс].Спосіб доступу:URL:
<https://hub.kyivstar.ua/articles/yak-sql-dopomagaye-keruvaty-biznes-danymy>, <https://www.phpmyadmin.net/>
8. jQuery [Електрон. ресурс]. Спосіб доступу: URL: <https://jquery.com>
9. jQueryUI [Електрон. ресурс]. Спосіб доступу: URL: <https://jqueryui.com>
10. Bootstrap5.3 [Електрон.ресурс].Спосіб доступу:URL:
https://w3schoolsua.github.io/bootstrap5/bootstrap_get_started.html
11. Google Fonts [Електрон.ресурс].Спосіб доступу:URL:
<https://webfonts.com.ua/fonts/google-fonts/>
12. Найкращі програми та додатки для малювання [Електрон. ресурс].
Спосіб доступу: URL: <https://www.komarov.design/naikrashchi-proghrami-ta-dodatki-dlia-maliuvannia-u-2024/>

13. Google reCaptcha v3 [Електрон. ресурс]. Спосіб доступу: URL:
<https://www.dreamhost.com/blog/uk/povnii-posibnik-z-recaptcha/>
14. Notepad++ [Електрон. ресурс]. Спосіб доступу: URL:
<https://notepadok.com/uk/>
15. Laravel (PHP) [Електрон. ресурс]. Спосіб доступу: URL:
<https://foxminded.ua/php-freimvorky/>
16. React.js (JavaScript) [Електрон. ресурс]. Спосіб доступу: URL:
<https://uk.legacy.reactjs.org/>
17. Node.js [Електрон. ресурс]. Спосіб доступу: URL:
<https://www.dreamhost.com/blog/uk/shcho-take-nodejs/>
18. MySQL / PostgreSQL [Електрон. ресурс]. Спосіб доступу: URL:
<https://www.dreamhost.com/blog/uk/postgresql-proti-mysql/>
19. Bootstrap / Tailwind CSS [Електрон. ресурс]. Спосіб доступу: URL:
<https://www.dreamhost.com/blog/uk/tailwind-proti-bootstrap/>

Додаток А. Програмний код

А.1 Програмний код login.php

```
<?php
session_start();
require 'db.php';

$message = '';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = trim($_POST['username'] ?? '');
    $password = $_POST['password'] ?? '';

    if (empty($username) || empty($password)) {
        $message = "Будь ласка, заповніть усі поля.";
    } else {
        $stmt = $conn->prepare("SELECT id, password, roles FROM users WHERE
username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $stmt->store_result();

        if ($stmt->num_rows === 1) {
            $stmt->bind_result($id, $password_hash, $roles);
            $stmt->fetch();

            if (password_verify($password, $password_hash)) {
                $_SESSION['username'] = $username;
                $_SESSION['user_id'] = $id;
                $_SESSION['user_roles'] = $roles;
                header("Location: index.php");
                exit();
            } else {
                $message = "Невірний пароль.";
            }
        } else {
            $message = "Користувача не знайдено.";
        }
    }
}
```

```

        $stmt->close();
    }
}
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="utf-8" />
    <link rel="shortcut icon" href="include/images/logo.png" type="image/png">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="include/css/app.css" />
    <script src="include/js/jquery.js"></script>
</head>
<body>
    <div class="container centered">
        <form class="authorization sign-in" action="login.php" id="login"
method="POST">
            <div class="background">
                
            </div>

            <?php if (!empty($message)): ?>
                <div class="warning-pass">
                    <span><?= $message ?></span>
                    <button type="button" onclick="history.back()">Повернутись</button>
                </div>
            <?php else: ?>
                <h2>Ласкаво просимо!</h2>
                <div class="input-group">
                    <div class="icon">
                        
                    </div>
                    <input type="text" name="username" placeholder="Логін або ID" required
/>
                </div>
                <div class="input-group">
                    <div class="icon button cursor">
                        
                    </div>

```

```

        <input class="input" type="password" name="password"
placeholder="Пароль" required />
    </div>
    <button type="submit">Увійти</button>
    <p><a href="register.php">Зареєструватися</a>
    <?php endif; ?>
</form>
</div>

<script>
    const button = document.querySelector('.button');
    const input = document.querySelector('.input');
    if (button && input) {
        button.addEventListener('click', () => {
            input.type = input.type === "password" ? "text" : "password";
        });
    }
</script>
</body>
</html>

```

A.2 Програмний код register.php

```

<?php
session_start();
require 'db.php';

$message = '';

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $username = trim($_POST['username'] ?? '');
    $password = $_POST['password'] ?? '';
    $password_confirm = $_POST['password_confirm'] ?? '';

    if (empty($username) || empty($password) || empty($password_confirm)) {
        $message = "Будь ласка, заповніть усі поля.";
    } elseif (strlen($password) < 6 || strlen($password) > 30) {
        $message = "Пароль повинен містити від 6 до 30 символів.";
    } elseif ($password !== $password_confirm) {
        $message = "Паролі не співпадають.";
    }
}

```

```

} else {
    $stmt = $conn->prepare("SELECT id FROM users WHERE username = ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        $message = "Користувач з таким логіном вже існує.";
    } else {
        $hashedPassword = password_hash($password, PASSWORD_DEFAULT);
        $stmt = $conn->prepare("INSERT INTO users (username, password) VALUES
(?, ?)");
        $stmt->bind_param("ss", $username, $hashedPassword);

        if ($stmt->execute()) {
            $message = "Реєстрація пройшла успішно. <a
href='login.php'>Увійти</a>";
        } else {
            $message = "Помилка під час реєстрації.";
        }
    }
    $stmt->close();
}
?>

<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Реєстрація</title>
    <link rel="shortcut icon" href="include/images/logo.png" type="image/png">
    <link rel="stylesheet" href="include/css/app.css" />
    <style>
        .password-group {
            position: relative;
        }
        .toggle-password {
            position: absolute;

```

```

    right: 14px;
    top: 50%;
    transform: translateY(-50%);
    cursor: pointer;
    font-size: 18px;
    color: #ccc;
    user-select: none;
}
.warning-container {
    margin-top: 16px;
    margin-bottom: 10px;
    text-align: center;
}
.warning-pass {
    background-color: rgba(255, 0, 0, 0.15);
    color: #ff4c4c;
    font-size: 14px;
    padding: 8px 16px;
    border-radius: 12px;
    display: inline-block;
    opacity: 0;
    transition: 0.3s ease;
}
.warning-pass.active {
    opacity: 1;
}
</style>
</head>
<body>
    <div class="container centered">
        <form class="authorization sign-in" action="register.php" method="POST"
autocomplete="off">
            <div class="background">
                
            </div>

            <h2>Реєстрація</h2>

            <div class="input-group">
                <div class="icon">

```

```

        
    </div>
    <input type="text" name="username" placeholder="Логін" required />
</div>

<div class="input-group password-group">
    <div class="icon">
        
    </div>
    <input id="password" class="input" type="password" name="password"
placeholder="Пароль" required />
    <span class="toggle-password" data-target="password">🔒</span>
</div>

<div class="input-group password-group">
    <div class="icon">
        
    </div>
    <input id="confirm_password" class="input" type="password"
name="password_confirm" placeholder="Повторіть пароль" required />
    <span class="toggle-password" data-target="confirm_password">🔒</span>
</div>

<button type="submit" class="btn-register">Зареєструватися</button>

<div class="warning-container">
    <div class="warning-pass<? = !empty($message) ? ' active' : '' ?>">
        <span><? = $message ?></span>
    </div>
</div>

<p><a href="login.php">Повернутись до авторизації</a></p>
</form>
</div>

<script src="include/js/app.min.js"></script>
<script>
    document.querySelectorAll('.toggle-password').forEach(toggle => {
        toggle.addEventListener('click', () => {

```

```
const input = document.getElementById(toggle.dataset.target);
if (input.type === 'password') {
  input.type = 'text';
  toggle.textContent = '👁';
} else {
  input.type = 'password';
  toggle.textContent = '🙈';
}
});
});
</script>
</body>
</html>
```