

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

**«Вебзастосунок UA Hammer для керування
армійськими списками та турнірами з
Warhammer 40 000»**

Виконав студент групи ПЗп-22-1
КРАВЦОВ Ілля Миколайович

Керівник роботи:
БАКАЛЮК Тетяна Анатоліївна

Рецензент:
АНТОНЮК Дмитро Сергійович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії програмного
забезпечення

Тетяна Вакалюк

«16» липня 2026 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **КРАВЦОВ Ілля Миколайович**

Керівник роботи: **ВАКАЛЮК Тетяна Анатоліївна**

Тема роботи: **«Вебзастосунок UA Hammer для керування армійськими списками та турнірами з Warhammer 40 000»**,

затверджена Наказом закладу вищої освіти від **«16» липня 2026 р., №396/с**

Вихідні дані для роботи: **процес створення армійських ростерів, керування ігровими даними та організації турнірів у межах настільної гри Warhammer 40,000, який передбачає використання вебзастосунку для збереження користувачів, юнітів, зброї, армійських списків, турнірів, учасників, раундів і результатів матчів.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Вакалюк Т.А.	16.07.2026р.	16.07.2026р
2	Вакалюк Т.А.	16.07.2026р	16.07.2026р
3	Вакалюк Т.А.	16.07.2026р	16.07.2026р

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з опису вебзастосунку UA Hammer, а також пояснювальної записки. Пояснювальна записка до випускної роботи містить 98 сторінок, 39 ілюстрацій та 18 таблиць.

Метою роботи є розробка вебзастосунку, який забезпечує користувачам зручну платформу для створення армійських списків, перегляду доступних бойових одиниць, роботи з турнірами та адміністрування довідкових даних системи.

Основна ідея полягає у створенні простого та зрозумілого вебінтерфейсу, що підтримує рольову модель користувачів, у якій звичайний користувач може створювати власні ростери та працювати з турнірами, а адміністратор має доступ до редакторів загонів і видалення турнірів.

У рамках проєкту було визначено ключові завдання розробки, проведено аналіз предметної області та існуючих рішень, що вже існують. Було спроектовано структуру бази даних MySQL/MariaDB для зберігання користувачів, ростерів, юнітів, зброї, здібностей, турнірів, учасників, раундів і матчів. Була спроектована клієнт-серверна архітектура, у якій клієнтська частина на основі HTML, CSS та JavaScript відповідає за відображення вебінтерфейсу у браузері користувача, а серверна частина на PHP забезпечує обробку запитів, взаємодію з базою даних, перевірку прав доступу та роботу з основними функціями системи: створенням ростерів, переглядом юнітів, турнірами й адміністративним керуванням.

КЛЮЧОВІ СЛОВА: UA HAMMER, WARHAMMER 40,000, ARMI БІЛДЕР, АРМІЙСЬКИЙ РОСТЕР, ТУРНІРНИЙ МОДУЛЬ, PHP, MYSQL, MARIADB, XAMPP, PDO, ВЕБЗАСТОСУНОК.

					ДУ «Житомирська політехніка».22.121.17.000 - ПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.		Кравцов І.М.			Вебзастосунок UA Hammer для керування армійськими списками та турнірами з Warhammer 40 000			Літ.	Арк.	Аркушів	
Керівник		Вакалюк Т.А.								4	98
								Житомирська політехніка, група ІПЗп-22-1			
Н. контр.											
Зав. каф.		Вакалюк Т.А.									

ABSTRACT

This bachelor's degree final project consists of a description of the UA Hammer web application, as well as an explanatory note. The explanatory note accompanying the final project comprises 98 pages, 39 illustrations and 18 tables.

The aim of the project is to develop a web application that provides users with a convenient platform for creating army lists, viewing available combat units, managing tournaments and administering the system's reference data.

The main idea is to create a simple and intuitive web interface that supports a role-based user model, in which a standard user can create their own rosters and manage tournaments, whilst an administrator has access to unit editors and the ability to delete tournaments.

As part of the project, key development tasks were identified, and an analysis of the subject area and existing solutions was carried out. A MySQL/MariaDB database structure was designed to store users, rosters, units, weapons, abilities, tournaments, participants, rounds and matches. A client-server architecture was designed, in which the client-side, based on HTML, CSS and JavaScript, is responsible for rendering the web interface in the user's browser, whilst the server-side component, written in PHP, handles request processing, interacts with the database, checks access rights and manages the system's core functions: creating rosters, viewing units, organising tournaments and administrative management.

KEYWORDS: UA HAMMER, WARHAMMER 40,000, ARMY BUILDER, ARMY ROSTER, TOURNAMENT MODULE, PHP, MYSQL, MARIADB, XAMPP, PDO, WEB APPLICATION.

		<i>Кравцов І.М</i>			<i>ІПЗ.КР.Б – 121 – 26 – ПЗ</i>	<i>Арк.</i>
		<i>Вакалюк Т.А..</i>				
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		<i>4</i>

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ.....	10
1.1 Постановка задачі.....	10
1.2 Аналіз аналогів програмного забезпечення.....	12
1.3 Вибір архітектури.....	17
1.4 Обґрунтування вибору технологічних рішень та вимоги до апаратного забезпечення.....	20
Висновки до першого розділу	24
РОЗДІЛ 2 ПРОЕКТУВАННЯ WEBЗАСТАСУНКУ	25
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи	25
2.2 Розробка бази даних системи.....	30
2.3 Проектування та реалізація алгоритмів роботи системи.....	40
2.4 Реалізація функціоналу системи.....	44
Висновки до другого розділу	53
РОЗДІЛ 3 ІНТЕРФЕЙС, ПОРЯДОК РОБОТИ ТА ТЕСТУВАННЯ СИСТЕМИ «UA HAMMER».	55
3.1 Порядок встановлення та налаштування параметрів системи.....	55
3.2 Структура вебінтерфейсу системи UA Hammer.....	58
3.3 Тестування роботи системи.....	70
Висновки до третього розділу	73
ВИСНОВКИ	75

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	78
ДОДАТКИ.....	80

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		6

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

UI – користувацький інтерфейс

URL – уніфікований показчик ресурсу, адреса вебсторінки

SMTP – протокол передавання електронної пошти

Warhammer 40,000 – настільна стратегічна гра, для якої розроблено вебзастосунок UA Hammer

Ростер – армійський список, що містить обраних користувачем юнітів

Юніт – бойова одиниця в межах армійського списку

Парінг – формування пари гравців у межах турнірного раунду

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. У сучасному цифровому світі вебзастосунки стали важливою частиною повсякденного життя користувачів. Вони використовуються для обробки даних, організації інформації, автоматизації різних процесів та забезпечення зручної взаємодії між користувачем і системою. Особливої актуальності набувають спеціалізовані вебплатформи, які орієнтовані на конкретні спільноти та дозволяють вирішувати практичні задачі у певній сфері.

Однією з таких сфер є настільні стратегічні ігри, зокрема Warhammer 40,000. Ця гра має велику кількість правил, фракцій, підрозділів, характеристик та обмежень і оновлень, які необхідно враховувати під час створення армійських списків. Через це ручне формування ростерів може бути незручним, займати багато часу та призводити до помилок у складання армії.

Також актуальною є проблема організації турнірів і взаємодії між учасниками ігрової спільноти. Новачкам часто складно знайти інформацію про події, записатися на турнір або зрозуміти, де саме проходять ігрові заходи.

Розрізненість спільноти та відсутність єдиної зручної системи можуть ускладнювати участь у таких подіях. Саме тому створення вебзастосунку, який поєднує функції армійського білдера та турнірної системи, є актуальним і має практичну цінність.

Метою даної роботи є розробка вебзастосунку UA Hammer для керування армійськими списками та турнірами з Warhammer 40,000. Система повинна забезпечувати можливість реєстрації та авторизації користувачів, створення і редагування ростерів, перегляду доступних юнітів, збереження даних у базі даних, а також створення турнірів і участі в них.

Для досягнення поставленої задачі необхідно провести аналіз предметної області та визначити основні проблеми, пов'язані із керуванням настільних стратегічних ігор. Також потрібно розглянути існуючі програмні рішення, які використовуються для створення армійських списків, роботи з правилами та проведення турнірних подій, щоб визначити їхні переваги й недоліки. На основі

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		8

цього необхідно сформувати вимоги до вебзастосунку UA Hammer, обґрунтувати вибір архітектури системи та технологічних засобів реалізації. Спроектувати структуру бази даних для зберігання користувачів, ростерів, юнітів, зброї, здібностей, турнірів, учасників, раундів і матчів. Після цього необхідно реалізувати серверну частину вебзастосунку з використанням PHP та MySQL/MariaDB, створити користувацький інтерфейс для роботи з ростерами, юнітами, турнірами й адміністративними функціями, а також перевірити працездатність системи в локальному середовищі XAMPP.

Результатом виконання роботи має стати функціональний вебзастосунок для гравців Warhammer 40,000, який дозволяє зручно створювати армійські списки, керувати збереженими ростерами, переглядати доступні підрозділи, а також створювати турніри та брати участь у них. Такий застосунок сприятиме спрощенню процесу підготовки до гри, зменшенню кількості помилок під час формування армій та покращенню взаємодії між учасниками ігрової спільноти.

Таким чином, дана робота спрямована на вирішення актуальної задачі автоматизації процесів, пов'язаних із керуванням армійськими списками та організацією турнірів у Warhammer 40,000, шляхом створення сучасного, зручного та функціонального вебзастосунку.

Об'єктом дослідження є процес автоматизації створення армійських списків та організації турнірів у настільній стратегічній грі Warhammer 40,000 із використанням прикладних вебтехнологій.

Предметом дослідження є методи, засоби та програмні інструменти розробки вебзастосунку UA Hammer, який забезпечує реєстрацію та авторизацію користувачів, створення і збереження армійських списків, перегляд доступних юнітів, а також створення турнірів і запис користувачів на них.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

1.1. Постановка задачі

У сучасному світі зростає потреба у створенні зручних і ефективних інструментів, для керування настільних стратегічних ігор, зокрема Warhammer 40,000. Традиційний процес підготовки і записів на турнір потребує значних витрат часу і знань, оскільки включає пошук актуальних правил, інформації про юніти, перевірку ігрових обмежень, підрахунок очок, формування ростера та турнірних подій.

Warhammer 40,000 є однією з найпопулярніших стратегічною настільною грою у світі, яка об'єднує велику кількість гравців по всьому світу. Гра вже багато років активно розвивається і все більше гравців залучається до неї. В ній є багато різних механік, фракцій та їх бойових одиниць, що постійно оновлюються відповідно до нових редакцій правил та змін балансу. Підготовка до гри вимагає ретельного вивчення актуальних правил і обмежень, за якими інколи важко стежити через велику кількість супровідних матеріалів. Ручна перевірки всіх обмежень часто призводить до помилок, які погіршують ігровий процес. Крім того, інформація про правила, характеристики бойових одиниць, зміни балансу та інші ігрові дані часто розміщується в різних джерелах, через що користувачам доводиться витрачати додатковий час на їх пошук і перевірку. Відсутність єдиного інформаційного середовища ускладнює підготовку до гри та підвищує ймовірність використання неактуальних даних. Не менш важливим є питання організації турнірів, яке включає реєстрацію учасників, формування парінгів, ведення результатів матчів і турнірної таблиці. Виконання цих процесів вручну потребує значних часових витрат і збільшує ризик організаційних помилок.

Розробка вебзастосунку для керування армійськими ростерами та турнірами є актуальним завданням, що спрямоване на спрощення процесу створення армійських списків, зручний перегляд ігрових даних, зменшення

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		10

кількості помилок під час формування ростера та покращення організації турнірних подій для гравців Warhammer 40,000.

Розроблений вебзастосунок забезпечить зручний спосіб керування армійськими ростерами, юнітами та турнірами. Система дозволить користувачам створювати армійські списки, переглядати доступні бойові одиниці, контролювати загальну кількість очок в ростері, організувати і приймати участь в турнірах.

Система буде реалізована у вигляді вебзастосунку, який дозволить користувачам працювати з армійськими ростерами, юнітами та турнірами в єдиному середовищі. Такий підхід спростить процес підготовки до гри, зменшить кількість ручних дій під час формування армійського списку та зробить організацію турнірних подій більш зручною.

Результатом роботи є вебзастосунок UA Hammer, призначений для керування армійськими ростерами, юнітами та турнірами у межах настільної гри Warhammer 40,000. Основною метою системи є спрощення процесу створення армійських списків, перегляду доступних бойових одиниць, запису користувачів на турніри, формування парінгів і збереження результатів матчів.

Основними завданнями, які має забезпечувати система, є створення та збереження армійських ростерів, вибір фракції і підфракції, керування юнітів в армійських списках, автоматичний підрахунок загальної кількості очок та перегляд характеристик бойових одиниць. Також вебзастосунок повинен забезпечувати пошук і фільтрацію юнітів, щоб користувач міг швидше знаходити потрібні бойові одиниці та формувати ростер відповідно до обраної фракції. Окремим напрямом роботи системи є турнірний модуль, який дозволяє створювати турніри, записувати користувачів на події, формувати парінги між учасниками, вносити результати матчів і зберігати турнірні показники гравців.

Впровадження такої системи дозволить зробити процес підготовки до гри більш структурованим і зручним для користувача. Замість використання кількох окремих джерел інформації гравець зможе працювати з ростерами, юнітами та турнірами в межах одного вебзастосунку. Це зменшить кількість ручних дій,

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		11

спростить пошук потрібної інформації та знизить ризик помилок під час створення армійського списку.

Крім того, система повинна бути побудована таким чином, щоб у майбутньому її можна було розширювати. До можливих напрямів розвитку належать додавання нових фракцій, юнітів, зброї, покращення турнірного модуля та створення модуля наративних кампаній. Отже, поставлена задача полягає не лише у створенні окремого армійського білдера, а у розробці вебзастосунку, який поєднує роботу з армійськими списками, ігровими даними та турнірними подіями.

1.2. Аналіз аналогів програмного забезпечення.

На даний момент існує багато інформаційно-пошукових та керувальних вебсистем, призначених для роботи з ігровими даними та створенням армійських ростерів у настільних стратегічних іграх. Подібні платформи забезпечують користувачам можливість структурувати інформацію, здійснювати пошук доступних ігрових одиниць та формувати власні армії відповідно до ігрових обмежень.

Для ефективного проєктування вебзастосунку UA Hammer доцільно розглянути наявні аналоги, які вже використовуються гравцями для створення ростерів, перегляду характеристик юнітів і роботи з турнірними подіями. Аналіз таких систем дозволяє визначити їхні основні функціональні можливості, переваги та недоліки, а також сформулювати вимоги до розроблюваного програмного продукту.

Армі білдери для настільних стратегічних ігор

Одним із основних типів програмного забезпечення, близького до тематики розроблюваного вебзастосунку, є армі білдери. Такі системи призначені для створення армійських ростерів у настільних стратегічних іграх. Вони дозволяють користувачам обирати ігрову систему, фракцію, підрозділи, спорядження та контролювати загальну вартість армії в очках.

До таких рішень можна віднести New Recruit, Tabletop Admiral, Warlord Games та інші подібні вебзастосунки. Вони значно спрощують процес підготовки

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		12

до гри, оскільки автоматично підраховують очки, показують доступні бойові одиниці та забезпечують відсутність помилок при зборі армії. Користувачу не потрібно вручну рахувати вартість кожного юніта або перевіряти склад армії вручну. Замість цього він може швидко сформувати ростер через вебінтерфейс і за потреби зберегти.

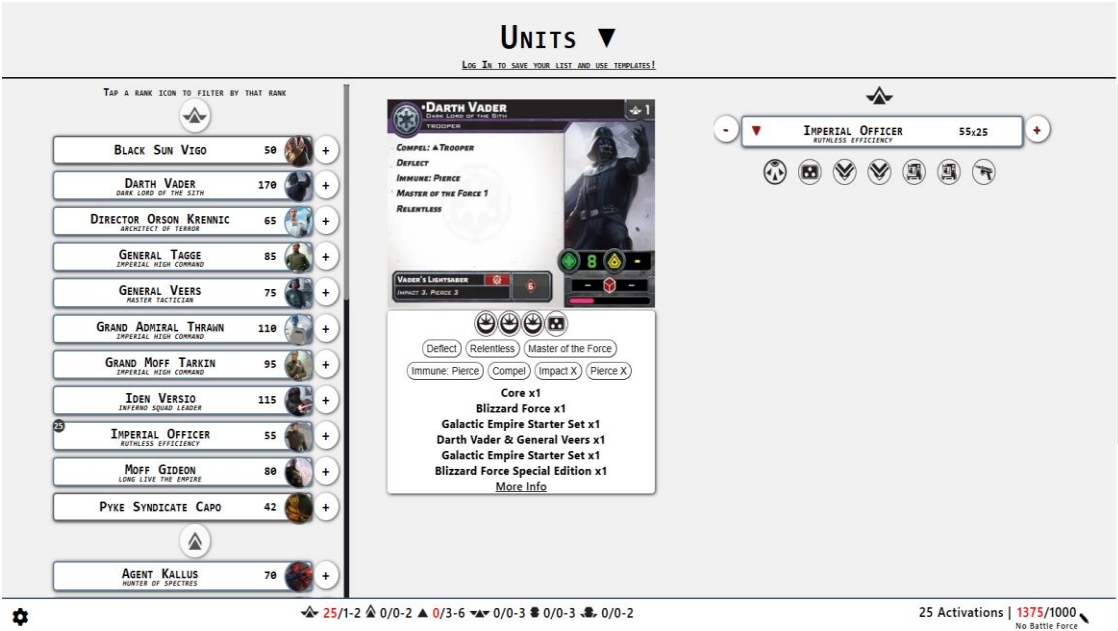


Рис. 1.1. Інтерфейс армі білдера Tabletop Admiral

Армі білдери, які використовується для створення ростерів у настільних стратегічних іграх є вебзастосунками, що дають змогу користувачам формувати списки армій, підраховувати очки та перевіряти відповідність ростера правилам гри. Сервіси дозволяють користувачу обрати фракцію та сформувати армійський список із доступних юнітів. Під час створення ростера система автоматично підраховує загальну кількість очок, що дає змогу контролювати відповідність списку встановленому ліміту.

Важливою особливістю них є можливість перегляду карток юнітів, їхнього спорядження. Завдяки цьому користувач може не лише додати підрозділ до списку, а й оцінити його роль у складі армії. Такий підхід спрощує планування ростера та допомагає швидше підібрати потрібні бойові одиниці.

Водночас Tabletop Admiral має певні обмеження. Інтерфейс сервісу виглядає досить простим і місцями застарілим, що може впливати на зручність роботи. Також функціонал збереження та подальшого керування ростерами є

менш розвиненим порівняно з деякими іншими армі білдерами. Крім того, сервіс орієнтований переважно на створення армійських списків і не надає повноцінного турнірного модуля для формування парінгів, внесення результатів матчів та ведення турнірної таблиці.

Інформаційні бази правил та довідкові ресурси

Окрему групу аналогів становлять інформаційні ресурси, які надають доступ до правил, характеристик юнітів, зброї, спеціальних здібностей і фракційних особливостей. Такі системи не завжди дозволяють створювати повноцінні ростери, однак вони є важливими для гравців, оскільки допомагають швидко знайти потрібну інформацію під час підготовки до гри або безпосередньо під час партії.

Подібні ресурси зазвичай містять структуровані сторінки з описом бойових одиниць, їх характеристик, доступної зброї та спеціальних правил. Це зручно, оскільки правила настільних стратегічних ігор часто мають значний обсяг, а пошук потрібного юніта або правила в паперових книгах займає багато часу.

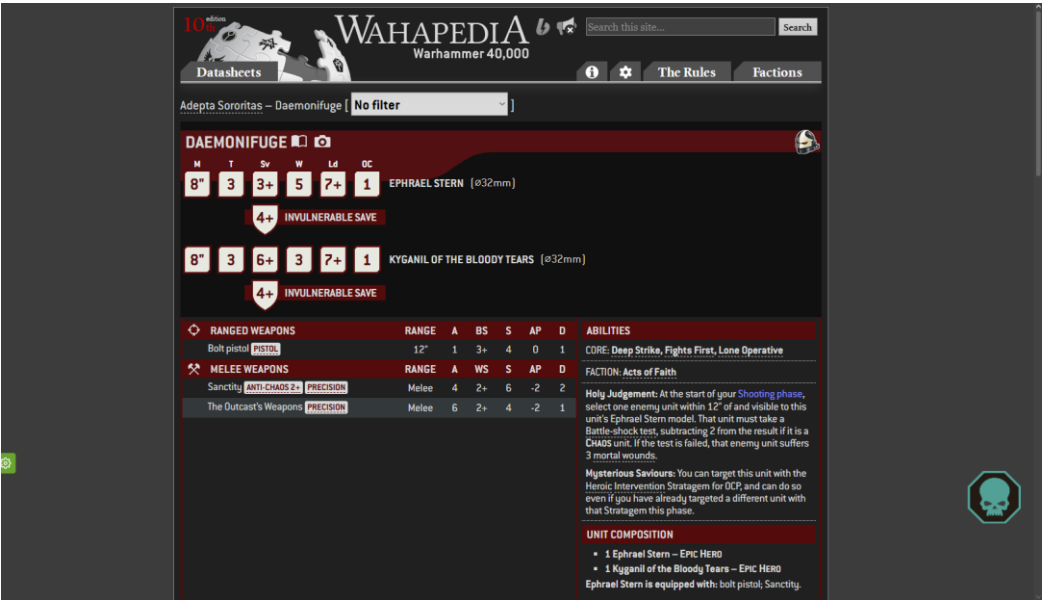


Рис. 1.2. Інтерфейс ресурсу з правилами Wahapedia

Перевагою таких систем є швидкий доступ до ігрової інформації та можливість переглядати дані без необхідності використовувати друковані матеріали. Однак їхнім недоліком є те, що вони часто виконують лише довідкову

функцію. Вони не завжди дозволяють створювати власні армійські списки, зберігати їх у профілі користувача або використовувати дані для турнірного модуля. Саме тому під час розробки UA Hammer було вирішено поєднати перегляд інформації про юнітів із можливістю додавання цих юнітів до ростера.

Турнірні платформи та системи організації подій

Ще одним важливим напрямом є системи для організації турнірів. Такі платформи дозволяють створювати події, реєструвати учасників, формувати раунди, створювати пари гравців і вносити результати матчів. Для настільних ігор турнірний функціонал є важливим, оскільки організатору потрібно швидко керувати списком учасників, контролювати результати та оновлювати турнірну таблицю після кожного раунду.

У подібних системах користувач зазвичай може переглянути список доступних турнірів, ознайомитися з датою проведення, місцем, форматом, кількістю учасників і правилами участі. Організатор, у свою чергу, має доступ до додаткових інструментів: створення раундів, парінгів, введення результатів і перегляду статистики. Це дозволяє зменшити кількість ручної роботи та мінімізувати помилки під час проведення події.

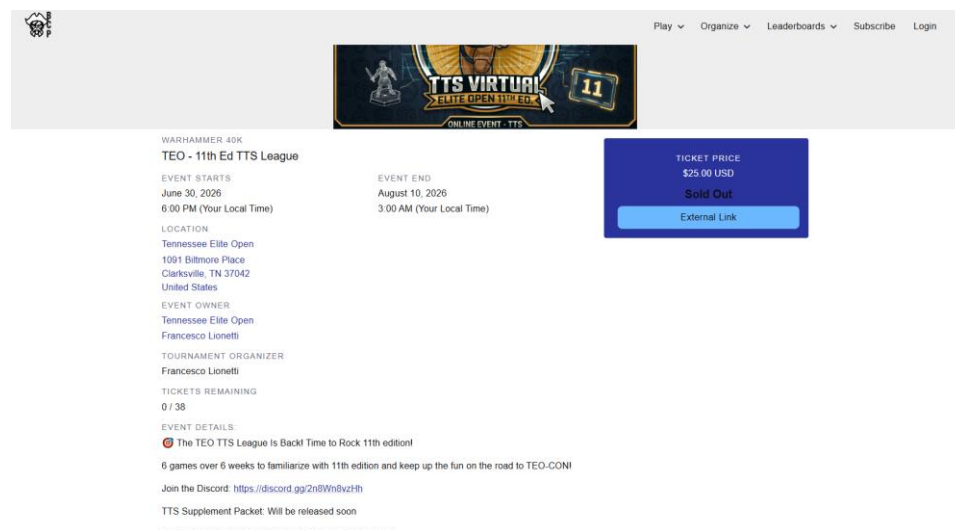


Рис. 1.3. . Інтерфейс турнірної платформи ВСР

Перевагою турнірних платформ є автоматизація організаційних процесів і зручність для учасників. Такі системи дозволяють швидше створювати події, реєструвати гравців, формувати раунди та зберігати результати матчів. Проте

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		15

їхнім недоліком є те, що вони часто орієнтовані саме на проведення турнірів і не завжди мають зручний зв'язок з армійськими ростерами користувачів. У багатьох випадках гравець створює армійський список в одному сервісі, а для участі в турнірі використовує іншу платформу. Через це дані про армію, учасника та результати змагання зберігаються окремо.

Для порівняння вище описаних вебзастосунків було створено порівняльну таблицю, у якій наведено основні функціональні можливості розглянутих аналогів.

Таблиця 1.1

Порівняння функціональних можливостей аналогів

Характеристика	Армі білдери	Інформаційні бази правил	Турнірні платформи
Створення армійських ростерів	Підтримується створення ростерів із автоматичним підрахунком очок	Не підтримується, ресурс виконує інформаційну функцію	Не є основною функцією системи
Перегляд характеристик юнітів	Доступний перегляд карток юнітів і пов'язаного спорядження	Доступний детальний перегляд характеристик, зброї та спеціальних правил	Не є основною функцією системи
Автоматичний підрахунок очок	Підтримується	Не підтримується	Не застосовується
Пошук і фільтрація юнітів	Частково підтримується за назвою	Підтримується пошук інформації за фракціями та правилами	Не орієнтований на пошук юнітів
Збереження ростерів	Підтримується	Не підтримується	Не є основною функцією
Турнірний модуль	Відсутній	Відсутній	Підтримується

Аналізуючи табл. 1.1, можна зробити висновок, що кожен із розглянутих вебзастосунків має власну спеціалізацію. Tabletop Admiral орієнтований переважно на створення армійських ростерів і автоматичний підрахунок очок.

Він є зручним для швидкого формування армійського списку, однак не надає повноцінного турнірного модуля та має обмежені можливості подальшого керування ростерами.

Wahapedia виконує роль інформаційно-довідкового ресурсу. Його головною перевагою є швидкий доступ до характеристик юнітів, зброї, спеціальних правил і фракційних особливостей. Водночас цей ресурс не призначений для створення власних ростерів, збереження армійських списків або організації турнірних подій.

ВСР, навпаки, орієнтований передусім на організацію турнірів. Така платформа дозволяє створювати події, реєструвати учасників і працювати з результатами матчів. Проте її функціональність не зосереджена на детальній роботі з юнітами, зброєю та створенням армійських списків у межах одного середовища.

Отже, проведений аналіз показує, що наявні рішення здебільшого закривають лише окремі потреби користувачів: створення ростерів, перегляд правил або організацію турнірів. Саме тому розробка UA Hammer є доцільною, оскільки система має поєднувати ці напрями в одному вебзастосунку. Користувач повинен мати можливість створювати армійські ростери, переглядати інформацію про юнітів, працювати з турнірами, а адміністратор - керувати ігровими даними системи.

1.3. Вибір архітектури

Під час розробки вебзастосунку UA Hammer важливо правильно обрати архітектуру системи, оскільки від цього залежить зручність використання, швидкість роботи, безпека даних та можливість подальшого розширення функціоналу. Розроблювана система повинна забезпечувати роботу з користувачами, армійськими ростерами, юнітами, фракціями та турнірами, тому доцільним є використання клієнт-серверної архітектури.

Клієнт-серверна архітектура передбачає поділ системи на дві основні частини: клієнтську та серверну. Клієнтська частина відповідає за відображення

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		17

інтерфейсу користувача у браузері, а серверна частина обробляє запити, виконує перевірку даних, працює з базою даних та формує відповідь для користувача.

Використання клієнт-серверної архітектури дозволяє розмежувати функції між окремими компонентами системи. Клієнтська частина відповідає за взаємодію з користувачем і відображення інформації, тоді як серверна частина забезпечує виконання бізнес-логіки, перевірку коректності введених даних та взаємодію з базою даних. Такий розподіл спрощує супровід вебзастосунку, підвищує його надійність і полегшує подальшу модернізацію.

Під час вибору архітектури було враховано, що система повинна бути доступною користувачу через браузер без встановлення додаткового програмного забезпечення. Це є важливим для вебзастосунку, оскільки користувачі можуть працювати з ростерами та турнірами з різних пристроїв, використовуючи лише інтернет-браузер. Такий підхід також спрощує оновлення системи, оскільки основна логіка зосереджена на серверній частині, а користувач отримує доступ до актуальної версії вебзастосунку через сторінки сайту.

У межах обраної архітектури клієнтська частина забезпечує взаємодію користувача із системою. Через вебінтерфейс користувач може зареєструватися, авторизуватися, перейти до особистого кабінету, створити армійський ростер, обрати фракцію, переглянути доступних юнітів, додати їх до списку або записатися на турнір. Клієнтська частина також відповідає за зручне відображення форм, списків, таблиць і повідомлень, які допомагають користувачу орієнтуватися в системі та розуміти результат виконаних дій.

Серверна частина відповідає за основну логіку роботи вебзастосунку. Вона приймає запити від клієнтської частини, перевіряє введені дані, працює із сесіями користувачів, звертається до бази даних і повертає результат виконання операції. Саме на серверній частині реалізовано реєстрацію та авторизацію, створення і видалення ростерів, додавання юнітів до армійського списку, підрахунок очок, створення турнірів, запис учасників і внесення результатів матчів.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Оскільки всі основні операції виконуються на сервері, користувач не має прямого доступу до бази даних, що підвищує рівень захисту інформації. Сервер здійснює перевірку прав доступу до функцій системи, контролює коректність виконання операцій та забезпечує цілісність даних під час роботи з ростерами, юнітами й турнірами.

Окремим важливим елементом архітектури є база даних. Вона використовується для збереження інформації про користувачів, армійські ростери, юнітів, фракції, турніри, учасників, раунди та результати матчів. Завдяки використанню бази даних система може зберігати створені користувачами ростери, відображати їх після повторного входу, контролювати участь у турнірах і забезпечувати зв'язки між окремими сутностями вебзастосунку.

Взаємодія між частинами системи відбувається послідовно. Користувач виконує дію у браузері, після чого клієнтська частина надсилає запит до серверної частини. Сервер перевіряє отримані дані, за потреби звертається до бази даних, виконує відповідну операцію та повертає результат користувачу. Така схема дозволяє централізовано контролювати доступ до даних, перевіряти права користувачів і зменшувати ризик некоректної роботи системи.

Нижче в таблиці (табл. 1.2) наведено опис основних компонентів клієнт-серверної архітектури, реалізованої в межах цього вебзастосунку .

Таблиця 1.2

Компонент	Функціонал
Клієнтська частина	Відображає вебінтерфейс у браузері користувача, форми, списки ростерів, юнітів і турнірів
Серверна частина	Обробляє запити користувача, виконує авторизацію, перевіряє права доступу та реалізує основну логіку системи

Компонент	Функціонал
База даних	Зберігає інформацію про користувачів, ростери, юнітів, турніри, учасників, раунди та результати матчів
Взаємодія з базою даних	Забезпечує виконання SQL-запитів через PDO та передавання даних між серверною частиною і базою даних
Сесії користувача	Зберігають стан авторизованого користувача та дозволяють обмежувати доступ до особистих ростерів і турнірних дій

Важливою перевагою клієнт-серверної архітектури є можливість подальшого розширення функціоналу. У майбутньому до вебзастосунку можна додавати нові фракції, юнітів, зброю, спеціальні правила, розширені функції турнірного модуля або модуль наративних кампаній. Оскільки система поділена на клієнтську частину, серверну логіку та базу даних, нові можливості можна впроваджувати поступово, не змінюючи повністю всю структуру вебзастосунку.

Таким чином, клієнт-серверна архітектура є доцільною для реалізації UA Hammer, оскільки вона забезпечує зручну роботу користувача через браузер, централізоване збереження даних, перевірку прав доступу та обробку основних функцій на серверній частині. Такий підхід відповідає вимогам системи та створює основу для її подальшого розвитку.

1.4 Обґрунтування вибору технологічних рішень та вимоги до апаратного забезпечення

Для реалізації вебзастосунку UA Hammer було обрано набір технологій, який дозволяє створити клієнтську частину, серверну логіку, базу даних і локальне середовище для розробки та тестування. Оскільки система працює через браузер, обробляє користувацькі запити, зберігає дані та використовує

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		20

авторизацію, доцільним є застосування HTML, CSS, JavaScript, PHP, MySQL/MariaDB, PDO та XAMPP.

Для реалізації серверної частини вебзастосунку було обрано мову програмування PHP. Вона підходить для створення динамічних вебсторінок, обробки HTTP-запитів, роботи з формами, сесіями користувачів і взаємодії з базою даних. У межах проєкту PHP використовується для реєстрації та авторизації користувачів, створення армійських ростерів, додавання і видалення юнітів, роботи з турнірами, перевірки прав доступу та виконання адміністративних дій.

Важливою перевагою PHP для даного проєкту є можливість швидко реалізувати серверну логіку вебзастосунку без використання складної додаткової інфраструктури. Це є доцільним для навчального проєкту, оскільки дозволяє зосередитися на реалізації основних функцій системи: роботі з користувачами, ростерами, юнітами, турнірами та базою даних. Також PHP підтримує механізми хешування паролів, що важливо для безпечного зберігання облікових даних користувачів. Крім того, він має велику кількість документації та добре інтегрується із сучасними системами керування базами даних. Це спрощує розробку і подальше розширення функціональних можливостей вебзастосунку, а також забезпечує можливість швидкого усунення виявлених недоліків.

Для створення клієнтської частини використовуються HTML, CSS та JavaScript. HTML відповідає за структуру сторінок вебзастосунку, зокрема головної сторінки, форм авторизації та реєстрації, особистого кабінету, армійського білдера, сторінки ростера і турнірного модуля. CSS застосовується для оформлення зовнішнього вигляду системи, налаштування кольорів, відступів, розташування блоків і загального стилю інтерфейсу. Завдяки цьому вебзастосунок має зрозумілий вигляд і відповідає тематиці Warhammer 40,000.

JavaScript використовується для покращення взаємодії користувача з вебзастосунком. У межах системи він може застосовуватися для пошуку юнітів, фільтрації за категоріями, оновлення списку доданих бойових одиниць та

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		21

зручнішої роботи з конструктором армій. Використання JavaScript дозволяє зробити інтерфейс більш інтерактивним і зменшити кількість зайвих дій користувача під час роботи з ростерами та турнірами. Його використання також дозволить виконувати частину перевірок введених користувачем даних прямо в веб-браузері. Це забезпечує швидший відгук інтерфейсу і зменшує навантаження на сервер та в цілому покращує роботу з системою.

Для збереження даних у системі використовується MySQL/MariaDB. Такий вибір є доцільним, оскільки вебзастосунок працює з великою кількістю пов'язаних сутностей: користувачами, армійськими ростерами, юнітами, фракціями, зброєю, турнірами, їх учасниками, раундами та результатами матчів. Реляційна база даних дозволяє організувати ці зв'язки, забезпечити збереження інформації та швидке отримання потрібних записів під час роботи користувача із системою.

У межах UA Hammer база даних використовується для збереження облікових записів користувачів, створених ростерів, доданих до них юнітів, інформації про турніри та участь користувачів у них. Завдяки цьому користувач може після авторизації переглядати власні армійські списки, відкривати їх для редагування, записуватися на турніри та працювати з результатами матчів. Для адміністратора база даних також є основою для керування довідковими даними системи, зокрема юнітами, зброєю та спеціальними правилами.

Для взаємодії PHP із базою даних використовується PDO. Це рішення дозволяє виконувати підготовлені SQL-запити та безпечніше обробляти дані, які вводить користувач. У межах системи це особливо важливо під час авторизації, створення ростера, додавання юнітів, запису на турнір і внесення результатів матчів. Використання PDO також спрощує обробку помилок і робить роботу з базою даних більш структурованою.

Для локальної розробки та тестування було обрано XAMPP. Це середовище містить Apache, PHP, MySQL/MariaDB і phpMyAdmin, що дозволяє швидко запустити вебзастосунок на локальному комп'ютері без оренди хостингу або складного налаштування сервера. XAMPP використовується для перевірки

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		22

роботи сторінок, взаємодії з базою даних, тестування авторизації, армійського білдера, турнірного модуля та адміністративних функцій.

Використання phpMyAdmin у складі XAMPP спрощує роботу з базою даних під час розробки. За його допомогою можна створювати таблиці, переглядати збережені записи, виконувати SQL-запити та перевіряти коректність зв'язків між даними. Це є зручним для навчального проєкту, оскільки дозволяє швидко контролювати стан бази даних і перевіряти, чи правильно система зберігає користувачів, ростери, юнітів і турнірну інформацію. Використання локального середовища розробки також дає змогу проводити тестування функціональних можливостей вебзастосунку без необхідності розгортання проєкту на віддаленому сервері. Це прискорює процес розробки, полегшує виправлення помилок та дозволяє перевіряти роботу нових функцій до їх впровадження в основну версію системи.

Для розроблення та використання вебзастосунку не потрібне спеціалізоване апаратне забезпечення. Система може працювати на персональному комп'ютері або ноутбучі із сучасним веббраузером, а локальна розробка виконується за допомогою середовища XAMPP. Це забезпечує доступність системи для широкого кола користувачів без додаткових вимог до технічного оснащення.

Після завершення розроблення система може бути розгорнута на віддаленому вебсервері, який підтримує PHP та MySQL/MariaDB. Такий підхід забезпечує можливість одночасної роботи багатьох користувачів через мережу Інтернет, централізоване зберігання даних та спрощує подальше супроводження і оновлення вебзастосунку.

Таким чином, для реалізації UA Hammer було обрано PHP, MySQL, JavaScript, HTML, CSS, XAMPP та PDO. Таке поєднання технологій є доцільним для створення вебзастосунку з функціями авторизації, керування армійськими ростерами, роботи з юнітами та організації турнірів.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		23

Висновки до першого розділу.

У першому розділі було проведено аналіз предметної області, пов'язаної зі створенням армійських ростерів, переглядом ігрових даних та організацією турнірів у настільних стратегічних іграх. У результаті було визначено основну проблему, яка полягає у необхідності зручного об'єднання роботи з ростерами, юнітами та турнірними подіями в межах одного вебзастосунку.

У процесі дослідження існуючих рішень було розглянуто декілька груп аналогів: армі білдери, інформаційні бази правил та турнірні платформи. Було встановлено, що кожна з цих систем має власне призначення і вирішує лише частину задачі. Армі білдери допомагають створювати ростери, довідкові ресурси надають доступ до характеристик юнітів і правил, а турнірні платформи спрощують організацію подій. Водночас такі рішення не завжди поєднують ці можливості в одному середовищі.

На основі проведеного аналізу було сформовано основні вимоги до вебзастосунку UA Hammer. Система повинна забезпечувати реєстрацію та авторизацію користувачів, створення і редагування армійських ростерів, перегляд доступних юнітів, пошук і фільтрацію, створення турнірів, запис учасників, формування парінгів та збереження результатів матчів. Також було визначено вимоги до зручності інтерфейсу, безпеки даних, продуктивності та можливості подальшого розширення функціоналу.

Відповідно до сформованих вимог було обґрунтовано вибір клієнт-серверної архітектури та основних технологічних засобів реалізації. Для розробки вебзастосунку було обрано PHP, MySQL, HTML, CSS, JavaScript, PDO та локальне середовище XAMPP. Такий набір технологій є достатнім для реалізації основних функцій системи, роботи з базою даних, обробки користувацьких запитів та подальшого розвитку вебзастосунку.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		24

РОЗДІЛ 2. ПРОЕКТУВАННЯ ВЕБЗАСТАСУНКУ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Вебзастосунок «UA Hammer» надає користувачам вебінтерфейс, за допомогою якого вони можуть створювати армійські ростери, редагувати їх, переглядати доступні юніти, та їх характеристики, а також створювати турніри та брати участь у них. Функціональність системи залежить від дій користувачів.

Користувачі можуть бути поділені на три основні групи: Звичайні користувачі мають можливість зареєструватися, авторизуватися, створювати власні армійські ростери, додавати та видаляти в них юніти, переглядати характеристики підрозділів, записуватися на турніри, переглядати список учасників, результати парінгів і рейтинг учасників. Організатори турнірів є користувачами, які створюють турніри вказувати їх основні параметри та керують їми, формувати парінги та вносити результати ігор. Адміни можуть видаляти турніри, додавати та редагувати нові юніти.

Основними об'єктами системи є користувач, адмін, ростер, юніт, фракція, турнір та учасник турніру. Користувач може мати декілька створених ростерів та турнірів. Кожен ростер належить конкретному користувачу та містить набір доданих юнітів. Юніти мають власні параметри, категорії, вартість в очках та належать до певних фракцій та підфракцій. Турнір містить інформацію про назву, дату проведення, місце, вартість участі, максимальну кількість учасників та список зареєстрованих гравців.

Для наочного відображення взаємодії користувачів із функціями вебзастосунку було створено діаграму варіантів використання (рис. 2.1.).

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		25

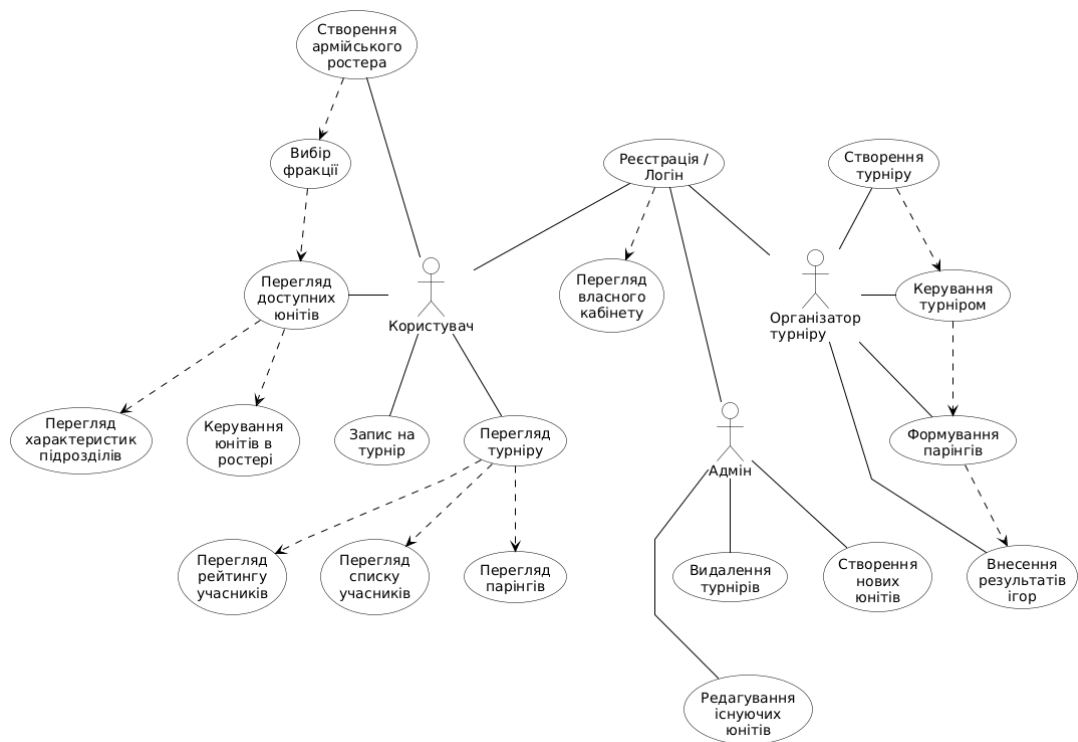


Рис. 2.1. Діаграма варіантів використання

Бізнес-вимоги

Вебзастосунок UA Hammer має забезпечити зручну цифрову платформу для створення армійських ростерів та організації турнірів з Warhammer 40,000. Система повинна спростити процес формування армійських списків, зменшити кількість ручних розрахунків під час вибору юнітів та надати користувачам єдину точку доступу до основних функцій: роботи з ростерами, фракціями, юнітами та турнірами.

Окремою бізнес-вимогою є можливість централізованого керування ігровими даними. Для цього у вебзастосунку передбачена адміністративна панель, яка дозволяє адміністратору створювати нові юніти та редагувати вже існуючі. Це дозволяє зручно підтримувати актуальність даних без прямого редагування бази даних.

Вимоги користувачів

Користувач очікує інтуїтивно зрозумілий інтерфейс, за допомогою якого можна зареєструватися, авторизуватися, створити армійський ростер, вибрати фракцію та додати до ростера необхідні юніти. Також користувач повинен мати

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		26

можливість переглядати характеристики підрозділів, доступні турніри, список учасників, парінги та рейтинг.

Організатор турніру ,який теж є користувачем, потребує можливості створювати турніри, вказувати їх основні параметри, формувати парінги та вносити результати ігор в створених їм турнірах.

Адміністратор системи повинен мати доступ до функцій керування юнітами та турнірами, зокрема створення нових юнітів, редагування існуючих і видалення турнірів.

Для забезпечення виконання наведеного вище функціоналу система повинна відповідати наступним вимогам, які представлені у таблиці (табл. 2.1.).

Таблиця 2.1

Функціональні вимоги

Функція	Опис
Авторизація та реєстрація	Система повинна забезпечувати можливість створення облікового запису, виходу та входу до системи.
Перегляд власного кабінету	Після входу користувач, організатор турніру або адміністратор повинен мати можливість переглядати власний кабінет та доступні функції відповідно до своєї ролі.
Створення армійського ростера	Авторизований користувач повинен мати можливість створити армійський ростер, вказати його назву, вибрати фракцію та задати ліміт очок.
Перегляд доступних юнітів	Система повинна відображати список юнітів, доступних для вибраної фракції. Для кожного юніта повинні відображатися назва, категорія та вартість в очках.
Перегляд характеристик підрозділів	Авторизований користувач повинен мати можливість переглядати характеристики конкретного підрозділу після перегляду списку доступних юнітів.

Функція	Опис
Керування юнітів в ростері	Авторизований користувач повинен мати можливість додавати доступних юнітів до створеного ростера, а також видаляти їх зі складу ростера. Після зміни складу система повинна оновлювати загальну вартість ростера в очках.
Перегляд турніру	Користувач повинен мати можливість переглядати основну інформацію про турнір: назву, дату проведення, місце, вартість участі, категорію та максимальну кількість учасників.
Запис на турнір	Авторизований користувач повинен мати можливість записатися на турнір, якщо кількість учасників не перевищує встановлений ліміт.
Перегляд даних турніру	Від сторінки перегляду турніру користувач повинен мати доступ до списку учасників, парінгів та рейтингу учасників.
Керування турніром	Авторизований користувач, який є організатором турніру, повинен мати можливість створювати турніри, керувати їх параметрами, формувати парінги та вносити результати ігор в створені їм турніри.
Адміністративна панель	Адміністратор повинен мати можливість створювати нові юніти, редагувати існуючі юніти, змінювати їх назву, категорію, вартість в очках та доступність для фракцій. Також адміністратор повинен мати можливість видаляти турніри.

Нефункціональні вимоги

Нефункціональні вимоги визначають якість роботи вебзастосунку за показниками зручності використання, безпеки, надійності та коректності збереження даних. Інтерфейс системи повинен бути зрозумілим для користувача та дозволяти швидко виконувати основні дії без додаткових інструкцій.

Доступ до функцій створення ростерів, запису на турніри та керування даними повинен надаватися лише авторизованим користувачам. Для

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		28

розмежування можливостей у системі використовуються ролі: користувач та адміністратор. Адміністративні функції повинні бути доступні лише адміністратору.

Дані про користувачів, ростери, юнітів, фракції, турніри та учасників турнірів повинні зберігатися у базі даних MySQL. Це забезпечує структуроване збереження інформації, зв'язок між основними об'єктами системи та можливість подальшого розширення функціоналу вебзастосунку.

Вимоги до апаратного забезпечення

Нижче в таблицях (табл. 2.2. і табл. 2.3.) наведено мінімальні та рекомендовані вимоги до апаратного забезпечення серверної частини та клієнтського обладнання, необхідного для коректної роботи вебзастосунку UA Hammer.

Таблиця 2.2

Серверні вимоги

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Процесор	2 ядра, 2.0 GHz	4 ядра, 2.5 GHz +
ОЗП	4 GB	8 GB +
Сховище	10 GB SSD	30 GB SSD +
Вебсервер	Apache у складі XAMPP	Apache 2.4.58 у складі XAMPP
Мова програмування	PHP 8.0	PHP 8.0.30 +
База даних	MySQL/MariaDB	MySQL/MariaDB 10.4.32 +
Мережа	50 Mbps	100 Mbps +

Таблиця 2.3

Клієнтські вимоги

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Пристрій	ПК або ноутбук	ПК або ноутбук
Процесор	Intel Core i3 / AMD Ryzen 3 або аналогічний	Intel Core i5 / AMD Ryzen 5 або вище
ОЗП	4 GB	8 GB +
Операційна система	Windows 10 / Linux / macOS	Windows 10/11 / Linux / macOS

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		29

Параметр	Мінімальні вимоги	Рекомендовані вимоги
Браузер	Google Chrome, Microsoft Edge, Opera	Google Chrome, Microsoft Edge, Opera

2.2 Розробка бази даних системи

Для збереження даних вебзастосунку UA Hammer використовується реляційна система керування базами даних MySQL/MariaDB. Такий підхід дозволяє зберігати інформацію у вигляді пов'язаних таблиць, забезпечувати цілісність даних та виконувати запити для отримання інформації про користувачів, армійські ростери, юнітів, зброю, фракційний доступ, турніри, раунди, парінги та результати ігор.

Таблиця 2.4

Короткий опис сутностей бази даних

Таблиця	Короткий опис
users	Облікові записи користувачів: id, email, хеш пароля, ім'я користувача, роль, дата створення акаунта.
password_resets	Запити на відновлення пароля: id, посилання на користувача, хеш токена, час завершення дії, дата використання, дата створення.
rosters	Армійські ростери користувачів: id, унікальний код ростера, посилання на користувача, ігрова система, root фракції, підфракція, назва списку, ліміт очок, JSON-дані, дати створення та оновлення.
roster_units	Юніти, додані до ростера: id, код ростера, посилання на юніт, кількість обраних юнітів.
units	Бойові загони : id, назва, категорія, вартість в очках, активність, дальність руху, стійкість, захист броні, кількість життів, мораль, можливість контролю цілей, непробивний захист броні.
Weapons	Зброя юнітів: id, назва, тип зброї, дальність, кількість атак, навичка стрільби або ближнього бою, сила, пробиття броні, шкода.
weapon_abilities	Здібності зброї: id, назва здібності, опис правила.
weapon_ability_links	Зв'язок зброї зі здібностями: посилання на зброю, посилання на здібність, значення правила, цільове ключеве слово.

Таблиця	Короткий опис
tournaments	Турніри системи: id, назва, дата проведення, місце, вартість участі, опис, категорія, максимальна кількість учасників, посилання на організатора, дати створення та оновлення.
tournament_participants	Учасники турніру: id, посилання на турнір, посилання на користувачів, дата запису, турнірні очки, сумарний рахунок, перемоги, нічий, поразки, кількість ігор проти проксібота.
tournament_rounds	Раунди турніру: id, посилання на турнір, номер раунду, статус раунду, дата створення.
tournament_matches	Матчі турніру: id, посилання на турнір, посилання на раунд, номер столу, перший гравець, другий гравець, рахунок першого та другого гравця, переможець, статус матчу, дата створення й оновлення, турнірні очки першого та другого гравця, ознака матчу проти проксібота.

Для реалізації правильних зв'язків між сутностями бази даних використовувались основні підходи побудови взаємозв'язків у реляційних базах даних: один-до-багатьох та багато-до-багатьох. Зв'язки один-до-багатьох застосовуються між користувачем і створеними ним ростерами, між користувачем і організованими турнірами, між турніром і його раундами, а також між турніром і матчами.

Для реалізації зв'язків багато-до-багатьох були використані додаткові проміжні таблиці:

1. roster_units – між армійськими ростерами та юнітами;
2. unit_weapons – між юнітами та зброєю;
3. weapon_ability_links – між зброєю та її здібностями;
4. tournament_participants – між користувачами та турнірами.

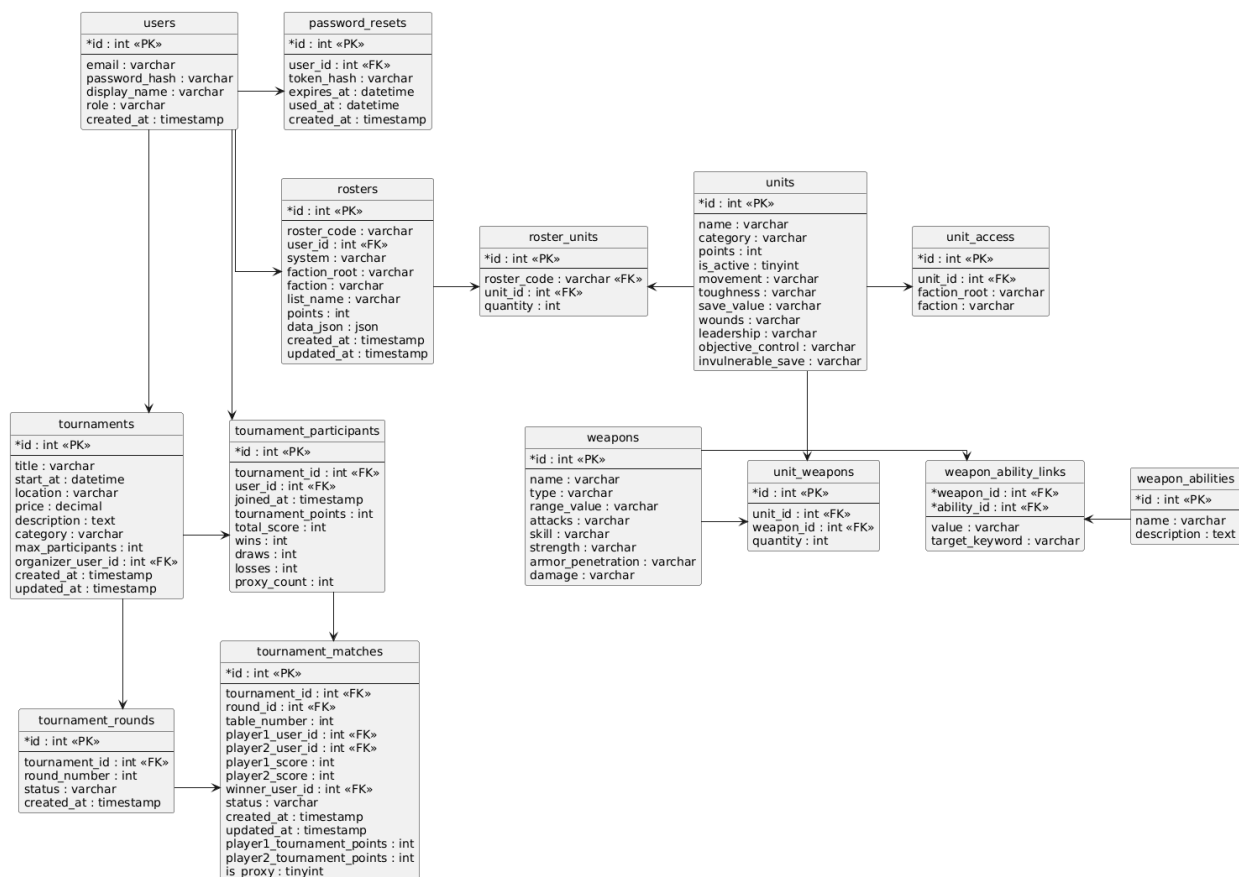


Рис. 2.2. Діаграма бази даних

Нижче наведено детальну інформацію про кожну із створених таблиць, їх поля, типи та призначення.

Таблиця users зберігає базову інформацію про кожного зареєстрованого користувача системи. Вона використовується для авторизації, ідентифікації користувача та розмежування прав доступу. Для цього в таблиці передбачено поле role, яке визначає тип користувача: звичайний користувач або адміністратор.

Таблиця 2.5.

Сутність users

Поле	Тип	ПК	ФК	Призначення
id	int	+	—	Унікальний ідентифікатор користувача
Email	varchar	—	—	Електронна адреса користувача для входу в систему
password_hash	varchar	—	—	Хеш пароля користувача

Продовження таблиці 2.5

Поле	Тип	ПК	ФК	Призначення
display_name	varchar	—	—	Ім'я користувача, яке відображається в системі
role	enum / varchar	—	—	Роль користувача: user або admin
created_at	timestamp	—	—	Дата та час створення облікового запису

Таблиця password_resets використовується для реалізації механізму відновлення пароля. Вона зберігає хеш токена скидання пароля, час завершення його дії та інформацію про те, чи був токен уже використаний.

Таблиця 2.6.

Сутність password_resets

Поле	Тип	ПК	ФК	Призначення
id	int	+	—	Унікальний ідентифікатор запису
user_id	int	—	users.id	Посилання на користувача
token_hash	varchar	—	—	Хеш токена для відновлення пароля
expires_at	datetime	—	—	Дата та час завершення дії токена
used_at	datetime	—	—	Дата та час використання токена
created_at	timestamp	—	—	Дата створення запиту на відновлення

Таблиця rosters зберігає армійські списки, створені користувачами. Кожен ростер має власний унікальний код, посилання на користувача, назву списку, фракцію, підфракцію та ліміт очок. Крім цього, таблиця використовується для зв'язку ростера з доданими до нього бойовими одиницями, що забезпечує можливість їх подальшого редагування та перегляду. Наявність окремої таблиці для зберігання ростерів дозволяє користувачу створювати декілька армійських списків, зберігати їх у базі даних та повертатися до них для внесення змін у будь-який момент після авторизації.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Таблиця 2.7.

Сутність rosters

Поле	Тип	ПК	ФК	Призначення
id	int	+	—	Унікальний ідентифікатор ростера
roster_code	varchar	—	—	Унікальний код ростера
user_id	int	—	users.id	Посилання на користувача, який створив ростер
system	varchar	—	—	Ігрова система
faction_root	varchar	—	—	Основна фракція
faction	varchar	—	—	Підфракція
list_name	varchar	—	—	Назва армійського списку
points	int	—	—	Ліміт очок ростера
data_json	json	—	—	Додаткові дані ростера у форматі JSON
created_at	timestamp	—	—	Дата створення
updated_at	timestamp	—	—	Дата оновлення

Таблиця roster_units є проміжною таблицею між ростерами та юнітами. Вона дозволяє додавати до одного ростера багато юнітів, а один і той самий юніт може використовуватися в різних ростерах.

Таблиця 2.8.

Сутність roster_units

Поле	Тип	ПК	ФК	Призначення
id	int	+	—	Унікальний ідентифікатор запису
roster_code	varchar	—	rosters.roster_code	Посилання на ростер
unit_id	int	—	units.id	Посилання на юніт
quantity	int	—	—	Кількість таких юнітів у ростері

Таблиця units містить список бойових юнітів, доступних у системі. У ній зберігаються назва, категорія, вартість в очках, активність юніта та основні ігрові характеристики.

Таблиця 2.9.

Сутність units

Поле	Тип	ПК	ФК	Призначення
id	int	+	—	Унікальний ідентифікатор юніта
name	varchar	—	—	Назва юніта
category	varchar	—	—	Категорія юніта
points	int	—	—	Вартість юніта в очках
is active	tinyint	—	—	Позначка активності юніта
movement	varchar	—	—	Показник руху
toughness	varchar	—	—	Показник витривалості
save value	varchar	—	—	Значення броні
wounds	varchar	—	—	Кількість життів
leadership	varchar	—	—	Показник лідерства
objective_control	varchar	—	—	Показник контролю цілі
invulnerable save	varchar	—	—	Значення непробиваємої броні

Таблиця unit_access визначає доступність юнітів для конкретних фракцій і підфракцій. Завдяки цій таблиці в армі білдері користувачу показуються лише ті юніти, які відповідають обраній ним фракції.

Таблиця 2.10.

Сутність unit_access

Поле	Тип	ПК	ФК	Призначення
id	int	+	—	Унікальний ідентифікатор запису
unit_id	int	—	units.id	Посилання на юніт
faction root	varchar	—	—	Основна фракція
faction	varchar	—	—	Підфракція

Таблиця weapons зберігає список зброї, яка може бути пов'язана з юнітами. У ній містяться основні характеристики зброї: тип, дальність, кількість атак, навичка, сила, пробиття броні та шкода. Також таблиця забезпечує централізоване зберігання інформації про озброєння, що дозволяє використовувати одну й ту саму зброю для декількох бойових одиниць без

дублювання даних. Завдяки цьому адміністратор може редагувати характеристики зброї лише в одному місці, після чого зміни автоматично застосовуються до всіх юнітів, яким вона призначена. Така структура бази даних спрощує супровід системи, підвищує цілісність даних та полегшує подальше розширення переліку озброєння.

Таблиця 2.11.

Сутність weapons

Поле	Тип	ПК	ФК	Призначення
id	int	+	–	Унікальний ідентифікатор зброї
name	varchar	–	–	Назва зброї
type	enum / varchar	–	–	Тип зброї: Ranged або Melee
range_value	varchar	–	–	Дальність зброї
attacks	varchar	–	–	Кількість атак
skill	varchar	–	–	Навичка стрільби або ближнього бою
strength	varchar	–	–	Сила зброї
armor_penetration	varchar	–	–	Пробиття броні
damage	varchar	–	–	Шкода зброї

Таблиця unit_weapons є проміжною таблицею між юнітами та зброєю. Вона дозволяє одному юніту мати декілька видів зброї, а одну зброю використовувати для різних юнітів.

Таблиця 2.12.

Сутність unit_weapons

Поле	Тип	ПК	ФК	Призначення
id	int	+	–	Унікальний ідентифікатор запису
unit_id	int	–	units.id	Посилання на юніт
weapon_id	int	–	weapons.id	Посилання на зброю
quantity	int	–	–	Кількість одиниць зброї

Таблиця weapon_abilities містить список спеціальних властивостей зброї. До таких властивостей можуть належати Pistol, Assault, Heavy, Rapid Fire,

Sustained Hits, Twin-linked, Anti та інші правила. Крім назви здібності, таблиця зберігає її текстовий опис, який використовується для відображення детальної інформації користувачу. Використання окремої таблиці дозволяє централізовано зберігати спеціальні правила, уникнути дублювання однакових записів для різних видів зброї та спростити їх оновлення або додавання нових здібностей у майбутньому. Такий підхід забезпечує гнучкість структури бази даних і полегшує розширення функціональних можливостей вебзастосунку.

Таблиця 2.13.

Сутність weapon_abilities

Поле	Тип	ПК	ФК	Призначення
id	int	+	–	Унікальний ідентифікатор здібності
Name	varchar	–	–	Назва здібності
description	text	–	–	Опис правила або властивості

Таблиця weapon_ability_links є проміжною таблицею між зброєю та її здібностями. Вона дозволяє одній зброї мати декілька здібностей, а одну й ту саму здібність використовувати для різної зброї. Додаткові поля value і target_keyword використовуються для властивостей, які мають числове значення або цільовий keyword, наприклад Anti-Vehicle 2+.

Таблиця 2.14.

Сутність weapon_ability_links

Поле	Тип	ПК	ФК	Призначення
weapon_id	int	+	weapons.id	Посилання на зброю
ability_id	int	+	weapon_abilities.id	Посилання на здібність
value	varchar	–	–	Значення здібності
target_keyword	varchar	–	–	Цільовий keyword для здібності

Таблиця tournaments зберігає турніри, створені користувачами. Вона містить назву турніру, дату проведення, місце, вартість участі, опис, категорію, максимальну кількість учасників та посилання на організатора.

Таблиця 2.15.

Сутність tournaments

Поле	Тип	ПК	ФК	Призначення
id	int	+	—	Унікальний ідентифікатор турніру
updated_at	timestamp	—	—	Дата оновлення
max_participants	int	—	—	Максимальна кількість учасників
Title	varchar	—	—	Назва турніру
start_at	datetime	—	—	Дата та час проведення
location	varchar	—	—	Місце проведення
price	decimal	—	—	Вартість участі
description	text	—	—	Опис турніру
category	varchar	—	—	Категорія турніру
organizer_user_id	int	—	users.id	Посилання на організатора
created_at	timestamp	—	—	Дата створення

Таблиця tournament_participants зберігає учасників турніру. Вона реалізує зв'язок між користувачами та турнірами, а також містить статистику гравця в межах конкретного турніру.

Таблиця 2.16.

Сутність tournament_participants

Поле	Тип	ПК	ФК	Призначення
id	int	+	—	Унікальний ідентифікатор запису
tournament_id	int	—	tournaments.id	Посилання на турнір
user_id	int	—	users.id	Посилання на користувача
joined_at	timestamp	—	—	Дата запису на турнір
tournament_points	int	—	—	Турнірні очки учасника
total_score	int	—	—	Загальна сума набраних очок
wins	int	—	—	Кількість перемог
draws	int	—	—	Кількість нічиїх
losses	int	—	—	Кількість поразок
proxy_count	int	—	—	Кількість матчів проти проксібота

Таблиця tournament_rounds використовується для зберігання раундів турніру. Кожен турнір може містити декілька раундів, у межах яких формуються пари гравців.

Таблиця 2.17.

Сутність tournament_rounds

Поле	Тип	ПК	ФК	Призначення
id	int	+	–	Унікальний ідентифікатор раунду
tournament_id	int	–	tournaments.id	Посилання на турнір
round_number	int	–	–	Номер раунду
Status	enum / varchar	–	–	Статус раунду
created at	timestamp	–	–	Дата створення раунду

Таблиця tournament_matches зберігає матчі турніру. Вона містить інформацію про раунд, номер столу, учасників матчу, результати гравців, переможця, статус матчу, турнірні очки та ознаку гри проти проксібота.

Таблиця 2.18.

Сутність tournament_matches

Поле	Тип	ПК	ФК	Призначення
id	int	+	–	Унікальний ідентифікатор матчу
tournament_id	int	–	tournaments.id	Посилання на турнір
round_id	int	–	tournament_rounds.id	Посилання на раунд
table_number	int	–	–	Номер ігрового столу
player1_user_id	int	–	users.id	Посилання на першого гравця
player2_user_id	int	–	users.id	Посилання на другого гравця
player1_score	int	–	–	Рахунок першого гравця
player2_score	int	–	–	Рахунок другого гравця
winner_user_id	int	–	users.id	Посилання на переможця
status	enum / varchar	–	–	Статус матчу
created_at	timestamp	–	–	Дата створення

Поле	Тип	ПК	ФК	Призначення
updated_at	timestamp	—	—	Дата оновлення
player1_tournament_points	int	—	—	Турнірні очки першого гравця
player2_tournament_points	int	—	—	Турнірні очки другого гравця
is_proxy	tinyint	—	—	Ознака матчу проти проксібота

Для реалізації складних зв'язків багато-до-багатьох у базі даних використовуються проміжні таблиці `roster_units`, `unit_weapons`, `weapon_ability_links` та `tournament_participants`. Таблиця `roster_units` пов'язує ростери з юнітами, `unit_weapons` – юнітів зі зброєю, `weapon_ability_links` – зброю з її здібностями, а `tournament_participants` – користувачів із турнірами. Такий підхід дозволяє уникнути дублювання даних і забезпечує гнучке керування складом ростерів, зброєю юнітів та участю користувачів у турнірах.

Для роботи з базою даних у вебзастосунку UA Hammer використовується MySQL у середовищі XAMPP. Доступ до бази даних реалізовано за допомогою PDO у PHP. Підключення до бази винесено в окремий файл `conector.php`, який створює об'єкт підключення до БД і використовується на сторінках системи. Запити до бази виконуються через підготовлені SQL-запити.

У системі використовується процедурна структура PHP-файлів, де кожен файл відповідає за окрему частину функціоналу: авторизацію, реєстрацію, роботу з акаунтом, створення ростерів, редагування юнітів, створення турнірів, запис учасників, формування парінгів і збереження результатів матчів. Такий підхід є достатнім для реалізації навчального вебзастосунку та дозволяє підтримувати логіку взаємодії з базою даних у зрозумілій структурі.

2.3 Проєктування та реалізація алгоритмів роботи системи

Вебзастосунок UA Hammer є простим у використанні та містить основні функції, необхідні для роботи з армійськими ростерами, юнітами, зброєю та турнірами. Алгоритм роботи із системою побудований таким чином, щоб

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		40

користувач міг швидко створити власний армійський список, переглянути доступні юніти, взяти участь у турнірі або створити власний турнір. Загальний алгоритм роботи системи зображено на рисунку 2.3, а нижче наведено детальний опис основних сценаріїв.

Взаємодія користувача із вебзастосунком реалізована за клієнт-серверною архітектурою. Усі дії користувача виконуються через веббраузер, який надсилає HTTP-запити до серверної частини системи. Після отримання запиту сервер виконує перевірку введених даних, обробляє запит відповідно до логіки вебзастосунку та звертається до бази даних для отримання або збереження інформації. Після завершення обробки сформована відповідь повертається користувачу, а інтерфейс відображає актуальний стан даних.

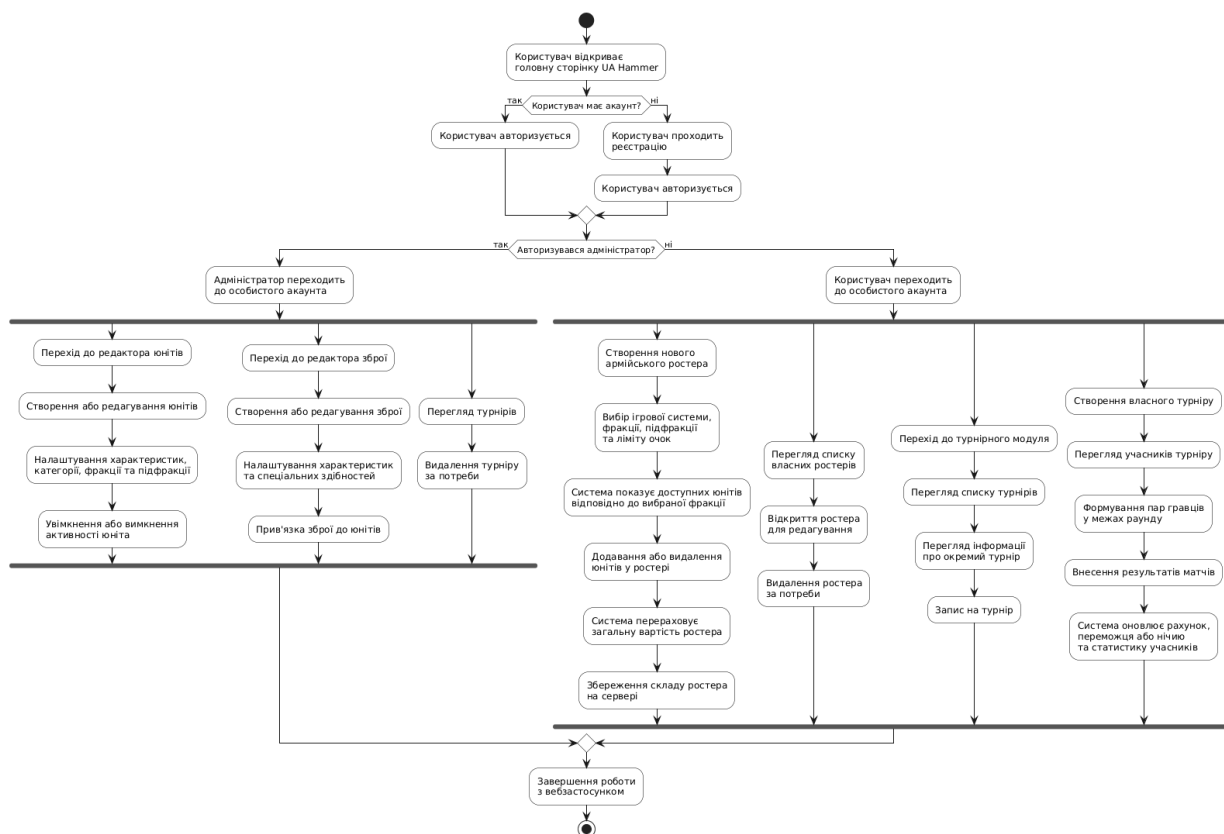


Рис. 2.3. Діаграма активності

Як показано на рисунку, робота системи починається з авторизації користувача. Після успішного входу користувач отримує доступ до функцій вебзастосунку та може перейти до створення або редагування ростера, перегляду турнірів чи виконання інших доступних операцій відповідно до своїх привілегій.

Кожна дія супроводжується перевіркою введених даних та взаємодією із серверною частиною системи, після чого результат відображається користувачу.

Після авторизації користувач отримує доступ до особистого акаунта, де може переглянути власні дані та перейти до основних функцій вебзастосунку. Однією з головних можливостей системи є створення армійського ростера. Для цього користувач вказує назву списку, ігрову систему, основну фракцію, підфракцію та ліміт очок. Після створення ростера система відкриває сторінку його редагування, на якій користувач може працювати зі складом армійського списку.

Під час редагування ростера система відображає лише ті юніти, які відповідають вибраній фракції та підфракції. Користувач може переглядати доступні бойові одиниці, додавати їх до ростера або видаляти зі списку. У процесі редагування система враховує вартість юнітів в очках і дозволяє контролювати загальну вартість армійського списку. Це дає змогу користувачу поступово формувати ростер і стежити за тим, щоб він відповідав установленому ліміту очок. Після додавання або видалення бойових одиниць система автоматично виконує перерахунок загальної вартості ростера та оновлює інформацію, що відображається користувачу. Усі внесені зміни зберігаються у базі даних, що дозволяє продовжити роботу з армійським списком після повторної авторизації.

Окрім створення нового ростера, користувач може переглядати список раніше створених армійських списків. Кожен ростер можна відкрити для подальшого редагування або видалити, якщо він більше не потрібний. Завдяки цьому користувач має можливість зберігати декілька варіантів армійських списків і повертатися до них у будь-який момент після авторизації.

Окремим напрямом роботи системи є турнірний модуль. Користувач може переглядати список створених турнірів і відкривати сторінку окремої події, де відображається її опис, дата проведення, місце, категорія, вартість участі та кількість учасників. Якщо кількість учасників не перевищує встановлений ліміт, користувач може записатися на турнір. Перед записом система перевіряє, чи не

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		42

був користувач уже зареєстрований на цю подію, що дозволяє уникнути повторної реєстрації.

Користувач також може створити власний турнір, вказавши його назву, дату проведення, місце, категорію, ціну та максимальну кількість учасників. Після створення турніру організатор отримує можливість переглядати список учасників, формувати пари гравців у межах раунду та вносити результати матчів після проведення туру. Система зберігає рахунок гравців, визначає переможця або нічию та оновлює турнірну статистику учасників. Після завершення кожного раунду результати матчів автоматично зберігаються у базі даних і використовуються для оновлення турнірної інформації. Це дозволяє організатору і учасникам переглядати поточний стан турніру, формувати наступні парінги та забезпечує цілісність збережених даних.

Якщо в системі авторизується адміністратор, він отримує доступ до розширених можливостей, які недоступні звичайному користувачу. Адміністратор може переглядати власний акаунт, працювати з редактором юнітів, створювати нові бойові одиниці, редагувати їхні характеристики, вартість в очках, категорію, фракцію та підфракцію. Також адміністратор може змінювати активність юніта, щоб керувати його доступністю в армійському редакторі. Сторінка перевіряє чи є у користувача відповідна роль для роботи адміністратором, такий механізм забезпечує безпеку додатку від неавторизованого втручання з сторони.

Крім роботи з юнітами, адміністратор має можливість прив'язувати до них створену зброю та працювати з редактором зброї. У цьому редакторі можна створювати і редагувати зброю, її характеристики та спеціальні здібності. Додатково адміністратор може видаляти турніри, якщо це потрібно для підтримки порядку в системі або усунення помилково створених подій.

Таким чином, алгоритм роботи вебзастосунку UA Hammer охоплює основні сценарії взаємодії звичайного користувача, організатора турніру та адміністратора. Система дозволяє створювати армійські ростери, працювати з

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		43

юнітами, організовувати турніри, вносити результати матчів і керувати довідковими даними, що робить її зручною платформою для гравців.

2.4 Реалізація функціоналу системи

Розглянемо основні етапи реалізації функціоналу вебзастосунку UA Hammer. Система призначена для створення армійських ростерів, керування юнітами та зброєю, а також для організації турнірів з Warhammer 40,000. Основна взаємодія користувача із системою відбувається через вебінтерфейс, а обробка даних виконується на серверній частині за допомогою PHP та бази даних MySQL/MariaDB.

Першим етапом роботи користувача із системою є реєстрація або авторизація. Під час реєстрації користувач вводить email, пароль, підтвердження пароля та особисті дані. Після надсилання форми система перевіряє правильність введених даних, хешує пароль і створює новий запис у таблиці users. Для збереження пароля використовується не відкритий текст, а його хеш, що підвищує безпеку зберігання облікових даних.

Функція авторизації перевіряє, чи існує користувач із введеним email у базі даних. Якщо запис знайдено, система порівнює введений пароль із хешем пароля. У разі успішної авторизації створюється сесія користувача, у якій зберігаються його ідентифікатор, email, ім'я та роль.

```
$st = $pdo->prepare("SELECT id, email, password_hash, display_name,  
role FROM users WHERE email = ? LIMIT 1");
```

```
$st->execute([$email]);
```

```
$user = $st->fetch();
```

```
if ($user && password_verify($password, $user["password_hash"])) {
```

```
    $_SESSION["user"] = [
```

```
        "id" => (int)$user["id"],
```

```
        "email" => $user["email"],
```

```
        "display_name" => $user["display_name"] ?: $user["email"],
```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		44

```

        "role" => $user["role"] ?: "user"
    ];
}

```

Ліст. 2.1. Реалізація авторизації користувача

Одним з основних функціональних модулів системи є армі білдер. Для створення нового армійського ростера користувач заповнює форму, у якій вказує ігрову систему, основну фракцію, підфракцію, назву списку та ліміт очок. Після цього система генерує унікальний код ростера та створює запис у таблиці rosters.

```

$rosterCode = bin2hex(random_bytes(12));

$st = $pdo->prepare("INSERT INTO rosters (roster_code, user_id, system,
faction_root, faction, list_name, points) VALUES (?, ?, ?, ?, ?, ?, ?)");

$st->execute([
    $rosterCode,
    $userId,
    $system,
    $factionRoot,
    $faction,
    $listName,
    $points
]);

```

Ліст. 2.2. Створення нового армійського ростера

Після створення ростера користувач переходить на сторінку його редагування. На цій сторінці система отримує з бази даних список доступних юнітів. Вибірка виконується з урахуванням таблиці unit_access, яка визначає, до яких фракцій і підфракцій належить кожен юніт.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		45

```

$st = $pdo->prepare("SELECT u.* FROM units u INNER JOIN
unit_access ua ON ua.unit_id = u.id WHERE u.is_active = 1 AND
ua.faction_root = ? AND (ua.faction = ? OR ua.faction IS NULL OR ua.faction =
") ORDER BY u.name ASC");

$st->execute([$factionRoot, $faction]);

$units = $st->fetchAll();

```

Ліст. 2.3. Отримання доступних юнітів для ростера

Додавання юніта до ростера реалізовано через проміжну таблицю roster_units. Коли користувач обирає юніт і додає його до списку, система створює новий запис із кодом ростера та ідентифікатором юніта. Якщо користувач видаляє юніт, відповідний запис видаляється з таблиці roster_units.

```

$st = $pdo->prepare("INSERT INTO roster_units (roster_code, unit_id,
quantity) VALUES (?, ?, 1)");

$st->execute([$rosterCode, $unitId]);

```

Ліст. 2.4. Додавання юніта до ростера

```

$st = $pdo->prepare("DELETE FROM roster_units WHERE id = ? AND
roster_code = ?");

$st->execute([$rosterUnitId, $rosterCode]);

```

Ліст. 2.5. Видалення юніта з ростера

Для адміністратора реалізовано окремий функціонал керування юнітами, який дозволяє створювати нові юніти, редагувати їх характеристики, задавати категорію, вартість в очках, фракційний доступ і стан активності. Поле is_active дозволяє приховувати юніт з армі білдера без повного видалення його з бази даних.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		46

```
$st = $pdo->prepare("INSERT INTO units (name, category, points,
is_active, movement, toughness, save_value, wounds, leadership,
objective_control, invulnerable_save) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
```

```
$st->execute([
    $name,
    $category,
    $points,
    $isActive,
    $movement,
    $toughness,
    $saveValue,
    $wounds,
    $leadership,
    $objectiveControl,
    $invulnerableSave
]);
```

Ліст. 2.6. Створення нового юніта адміністратором

Окремо реалізовано редактор зброї. У системі зброя зберігається в таблиці weapons, а її спеціальні здібності - у таблиці weapon_abilities. Зв'язок між зброєю та здібностями реалізовано через таблицю weapon_ability_links. Такий підхід дозволяє одній зброї мати декілька здібностей, а одну й ту саму здібність використовувати для різних видів зброї.

```
$st = $pdo->prepare("INSERT INTO weapons (name, type, range_value,
attacks, skill, strength, armor_penetration, damage) VALUES (?, ?, ?, ?, ?, ?, ?, ?,
?)");
```

```
$st->execute([
    $name,
```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		47

```

$type,
$rangeValue,
$attacks,
$skill,
$strength,
$armorPenetration,
$damage
]);
$weaponId = $pdo->lastInsertId();

```

Ліст. 2.7. Створення зброї

Для прив'язки здібності до зброї використовується таблиця `weapon_ability_links`. Додаткові поля `value` і `target_keyword` дозволяють зберігати значення правила або цільове ключове слово, наприклад `Anti-Vehicle 2+`.

```

$st = $pdo->prepare("INSERT INTO weapon_ability_links (weapon_id,
ability_id, value, target_keyword) VALUES (?, ?, ?, ?)");

```

```

$st->execute([
    $weaponId,
    $abilityId,
    $value,
    $targetKeyword
]);

```

Ліст. 2.8. Прив'язка здібності до зброї

Наступним важливим модулем системи є турнірний модуль. Користувач може створити новий турнір, вказавши назву, дату проведення, місце, вартість участі, категорію, опис і максимальну кількість учасників. Після створення

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		48

турніру id користувача записується як `organizer_user_id`, що дозволяє визначити організатора турніру.

```
$st = $pdo->prepare("INSERT INTO tournaments (title, start_at, location, price, description, category, max_participants, organizer_user_id) VALUES (?, ?, ?, ?, ?, ?, ?, ?)");
```

```
$st->execute([
    $title,
    $startAt,
    $location,
    $price,
    $description,
    $category,
    $maxParticipants,
    $userId
]);
```

Ліст. 2.9. Створення нового турніру

Запис користувача на турнір реалізовано через таблицю `tournament_participants`. Перед додаванням нового учасника система перевіряє, чи не перевищено максимальну кількість учасників, а також чи не був користувач уже записаний на цей турнір.

```
$st = $pdo->prepare("INSERT INTO tournament_participants (tournament_id, user_id) VALUES (?, ?)");
$st->execute([$tournamentId, $userId]);
```

Ліст. 2.10. Запис користувача на турнір

Для проведення турніру організатор створює раунди та формує пари гравців. Дані про раунд зберігаються в таблиці `tournament_rounds`, а дані про матчі - у таблиці `tournament_matches`. Кожен матч містить номер столу, першого та другого гравця, рахунок, переможця, статус і турнірні очки.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		49

```

$st = $pdo->prepare("INSERT INTO tournament_rounds (tournament_id,
round_number, status) VALUES (?, ?, 'active')");
$st->execute([$tournamentId, $roundNumber]);
$roundId = $pdo->lastInsertId();

```

Ліст. 2.11. Створення раунду турніру

```

$st = $pdo->prepare("INSERT INTO tournament_matches (tournament_id,
round_id, table_number, player1_user_id, player2_user_id, status, is_proxy)
VALUES (?, ?, ?, ?, ?, 'planned', ?)");

```

```

$st->execute([
    $tournamentId,
    $roundId,
    $tableNumber,
    $player1Id,
    $player2Id,
    $isProxy
]);

```

Ліст. 2.12. Створення матчу турніру

Після завершення гри організатор вносить результат матчу. Система визначає переможця або нічию. Якщо різниця між рахунками гравців становить 10 очок або менше, результат вважається нічиєю. Якщо різниця становить 11 очок або більше, переможцем стає гравець із більшим рахунком. Після цього система оновлює таблицю tournament_matches і статистику учасників у таблиці tournament_participants.

```

$scoreDiff = abs($score1 - $score2);
$drawMargin = 10;

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		50

```

if ($scoreDiff <= $drawMargin) {
    $player1TournamentPoints = 1;
    $player2TournamentPoints = 1;
    $winnerUserId = null;
} elseif ($score1 > $score2) {
    $player1TournamentPoints = 2;
    $player2TournamentPoints = 0;
    $winnerUserId = $player1Id;
} else {
    $player1TournamentPoints = 0;
    $player2TournamentPoints = 2;
    $winnerUserId = $player2Id;
}

```

Ліст. 2.13. Визначення результату матчу

```

$st = $pdo->prepare("UPDATE tournament_matches SET player1_score =
?, player2_score = ?, winner_user_id = ?, player1_tournament_points = ?,
player2_tournament_points = ?, status = 'finished' WHERE id = ?");
$st->execute([
    $score1,
    $score2,
    $winnerUserId,
    $player1TournamentPoints,
    $player2TournamentPoints,
    $matchId]);

```

Ліст. 2.14. Збереження результату матчу

Підсумовуючи, було створено діаграму компонентів вебзастосунку UA Hammer, на якій визначено основні частини системи та взаємодію між ними. Діаграма компонентів доречно розміщується саме наприкінці розділу 2.4, після

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		51

опису реалізованого функціоналу, оскільки вона узагальнює всі реалізовані модулі системи.

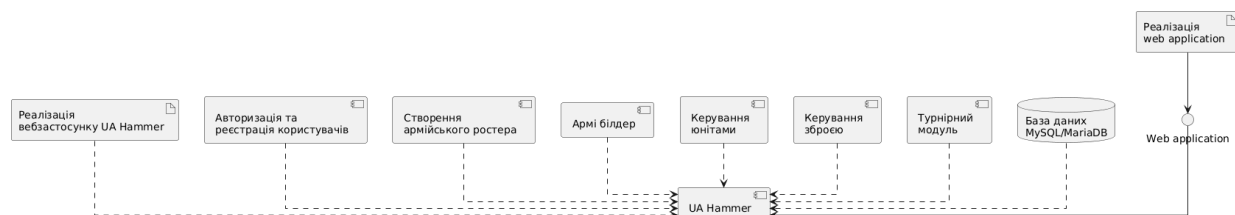


Рис. 2.4. Діаграма компонентів вебзастосунку UA Hammer

На діаграмі відображено основні компоненти вебзастосунку та їх взаємодію. Модуль авторизації та реєстрації забезпечує створення облікових записів користувачів, виконання входу до системи та керування користувацькою сесією. Модуль армійського білдера призначений для створення й редагування армійських ростерів, вибору фракції та керування складом бойових одиниць. Для адміністрування довідкової інформації передбачено модулі керування юнітами та зброєю, які дозволяють створювати, редагувати та змінювати характеристики відповідних об'єктів, а також визначати їх доступність у системі. Турнірний модуль забезпечує створення турнірів, реєстрацію учасників, формування парінгів, проведення раундів і збереження результатів матчів. Зберігання всієї інформації здійснюється в базі даних MySQL/MariaDB, яка містить відомості про користувачів, армійські ростери, юнітів, зброю, турніри, раунди та результати проведених матчів.

Таким чином, у вебзастосунку UA Hammer реалізовано основні функціональні модулі: реєстрацію та авторизацію користувачів, створення і редагування армійських ростерів, керування юнітами та зброєю, створення турнірів, запис учасників, формування раундів, створення парінгів і збереження результатів матчів. Усі дані зберігаються в реляційній базі даних MySQL/MariaDB, а взаємодія з нею виконується через PDO та підготовлені SQL-запити.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		52

Висновки до другого розділу.

У другому розділі було розглянуто основні етапи проектування та реалізації вебзастосунку UA Hammer. Спочатку були визначені ключові сценарії використання системи для звичайного користувача, організатора турніру та адміністратора. На основі цих сценаріїв було побудовано UML-діаграми, які відображають загальну логіку роботи вебзастосунку, процес створення армійського ростера, роботу турнірного модуля та взаємодію основних компонентів системи.

Окрему увагу було приділено проектуванню бази даних. Для збереження даних у системі використовується база даних MySQL/MariaDB. Вона містить таблиці для користувачів, армійських ростерів, юнітів, зброї, здібностей зброї, турнірів, учасників, раундів і матчів. Зв'язки між сутностями реалізовано за допомогою зовнішніх ключів та проміжних таблиць, що дозволяє коректно зберігати склад ростерів, фракційний доступ юнітів, прив'язку зброї та участь користувачів у турнірах.

Серверна частина вебзастосунку реалізована мовою PHP. Взаємодія з базою даних виконується через PDO та підготовлені SQL-запити, що забезпечує безпечну передачу даних. Функціонал системи розділено між окремими PHP-файлами, кожен з яких відповідає за конкретну дію: авторизацію, реєстрацію, створення ростера, додавання юнітів, керування зброєю, створення турніру, формування парінгів або збереження результатів матчів.

Також було описано реалізацію основних функціональних модулів системи. Армі білдер дозволяє користувачу створювати ростери, обирати фракцію, переглядати доступних юнітів і формувати склад армії з урахуванням вартості в очках. Адміністративна частина дає можливість керувати юнітами, їх характеристиками, фракційним доступом, зброєю та спеціальними здібностями. Турнірний модуль забезпечує створення турнірів, запис учасників, формування раундів, створення пар гравців і внесення результатів матчів.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		53

У результаті було сформовано логічну структуру вебзастосунку, яка відповідає поставленим вимогам і дозволяє підтримувати основні сценарії роботи UA Hammer.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		54

РОЗДІЛ 3. ІНТЕРФЕЙС, ПОРЯДОК РОБОТИ ТА ТЕСТУВАННЯ СИСТЕМИ «UA HAMMER».

3.1 Порядок встановлення та налаштування параметрів системи

Перед запуском вебзастосунку UA Hammer необхідно підготувати локальне серверне середовище. Для роботи системи використовується ХАМРР, оскільки він містить основні компоненти, необхідні для запуску PHP-застосунку: вебсервер Apache, інтерпретатор PHP та систему керування базами даних MySQL/MariaDB.

Першим кроком потрібно встановити ХАМРР на комп'ютер і запустити його панель керування. У вікні ХАМРР Control Panel необхідно активувати модулі Apache та MySQL. Apache відповідає за запуск вебсервера, а MySQL використовується для роботи з базою даних системи.

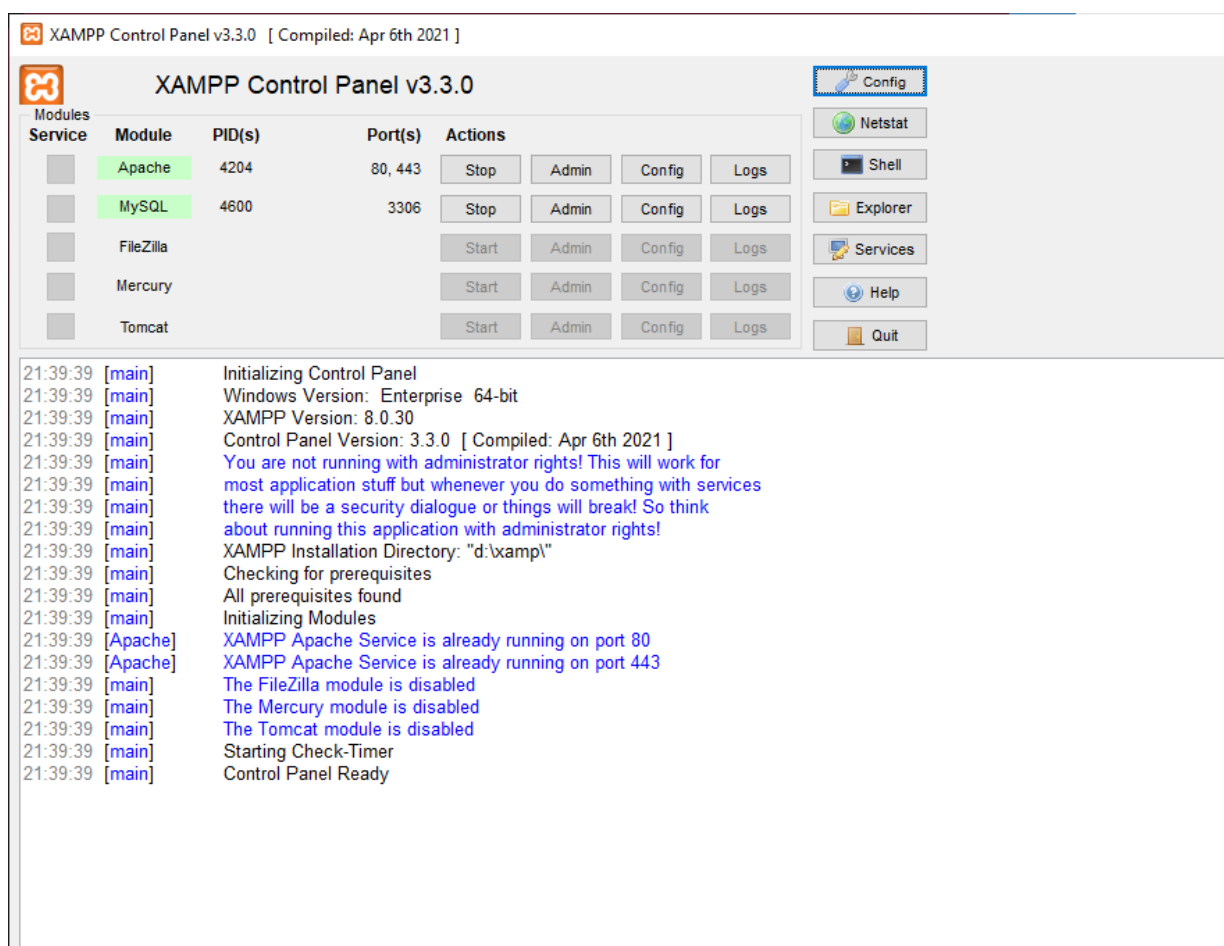


Рис. 3.1. Панель керування ХАМРР із запущеними модулями Apache та MySQL

Після запуску MySQL потрібно відкрити phpMyAdmin у браузері. Для цього можна перейти за адресою: <http://localhost/phpmyadmin>

У phpMyAdmin необхідно завантажити базу даних для вебзастосунку. У межах проєкту використовується база даних `uahammer_db`, у якій зберігається інформація про користувачів, армійські ростери, юнітів, зброю, здібності зброї, турніри, учасників, раунди та матчі. Після створення бази потрібно імпортувати SQL-файл зі структурою таблиць або вручну виконати SQL-запити для створення потрібних таблиць.

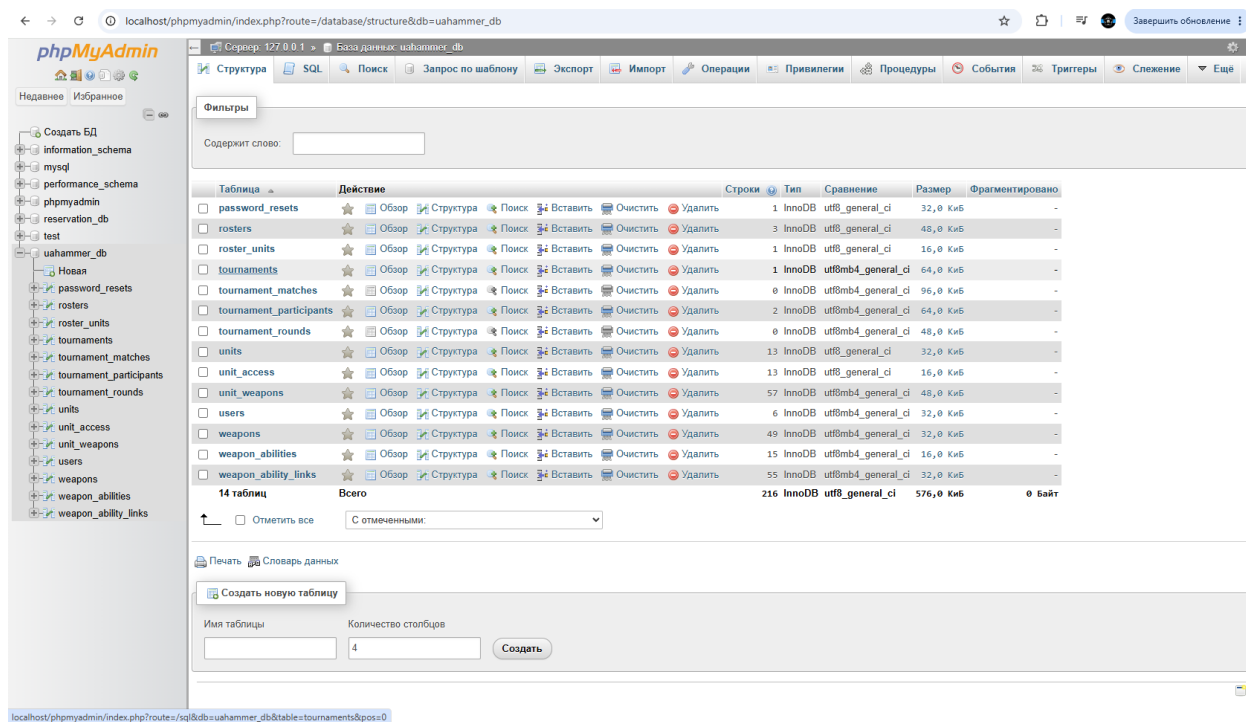


Рис. 3.2. phpMyAdmin з базою `uahammer_db`

Наступним кроком потрібно розмістити файли проєкту в каталозі `htdocs`, який знаходиться в папці встановленого XAMPP. У моєму випадку файли вебзастосунку розміщено за шляхом: `C:\xampp\htdocs\uahammerdiplom`

Після цього вебзастосунок можна відкрити в браузері за адресою:

<http://localhost/ua%20hammer%20diplom/home.php>

У цій адресі `localhost` означає локальний сервер, запущений через Apache у XAMPP.

Після розміщення файлів проєкту потрібно перевірити налаштування підключення до бази даних. Для цього використовується файл `conector.php`, у якому створюється об'єкт PDO для взаємодії з MySQL/MariaDB. У локальному середовищі XAMPP використовується база даних `uahammer_db`, сервер `127.0.0.1`, користувач `root` і порожній пароль. Також у параметрах PDO задається режим обробки помилок через винятки та режим отримання результатів у вигляді асоціативного масиву.

```
<?php
declare(strict_types=1);
$pdo = new PDO(
    "mysql:host=127.0.0.1;dbname=uahammer_db;charset=utf8mb4",
    "root",
    "",
    [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ]
);
```

Рис. 3.3. Налаштування підключення до бази даних

Після налаштування підключення потрібно перевірити відкриття головної сторінки системи. Якщо сторінка `home.php` відкривається без помилок, це означає, що вебсервер Apache працює коректно, а файли проєкту розміщені у правильній директорії.

Після виконання цих дій вебзастосунок UA Hammer буде готовий до роботи в локальному середовищі. Користувачі зможуть реєструватися, авторизуватися, створювати армійські ростери, працювати з армі білдером, та з турнірним модулем. Адміністратор, у свою чергу, зможе керувати юнітами, зброєю та модерувати турніри в системі.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		57

3.2 Структура вебінтерфейсу системи UA Hammer

Вебінтерфейс системи UA Hammer реалізовано у вигляді набору PHP-сторінок, кожна з яких відповідає за окрему частину функціоналу. Користувач взаємодіє із системою через браузер, переходячи між сторінками головного меню, формами авторизації, особистим кабінетом, армії білдером, турнірним модулем та адміністративними редакторами. Інтерфейс побудований таким чином, щоб користувач міг швидко перейти до потрібного розділу та виконати основні дії без складного налаштування.

Після відкриття вебзастосунку користувач потрапляє на головну сторінку системи. На ній розміщено основну навігацію, через яку можна перейти до авторизації, перегляду інформаційних сторінок, армії білдера або турнірного модуля. Також у шапці сторінки передбачено перемикач теми інтерфейсу, що дозволяє використовувати світле або темне оформлення.



Рис. 3.4. Головна сторінка вебзастосунку UA Hammer

Для отримання доступу до основного функціоналу користувач повинен авторизуватися в системі. Сторінка авторизації містить форму з полями для введення email та пароля. Після введення даних система перевіряє їх у базі даних і, у разі успішного входу, створює сесію користувача. Якщо користувач ще не має акаунта, він може перейти на сторінку реєстрації.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		58

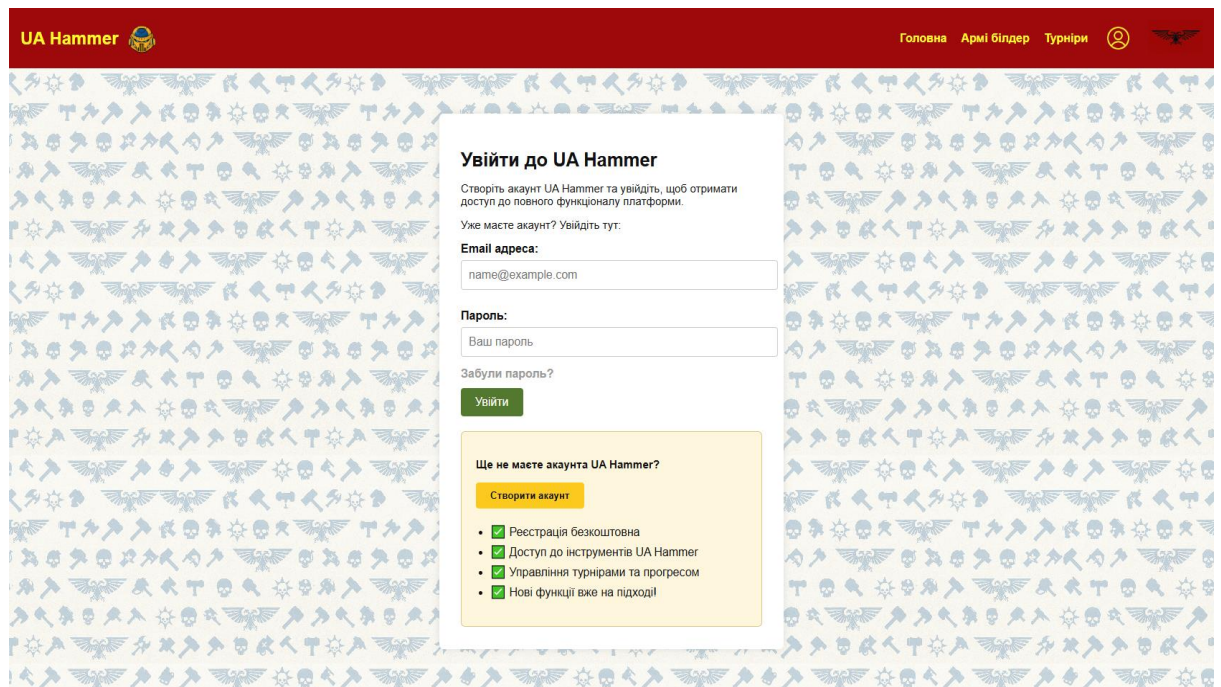


Рис. 3.5. Сторінка авторизації користувача

Сторінка реєстрації дозволяє створити новий обліковий запис. Користувач вводить email, пароль, підтвердження пароля та додаткові особисті дані. Після успішної перевірки введеної інформації пароль хешується, а новий користувач додається до таблиці users. Надалі цей обліковий запис можна використовувати для входу в систему.

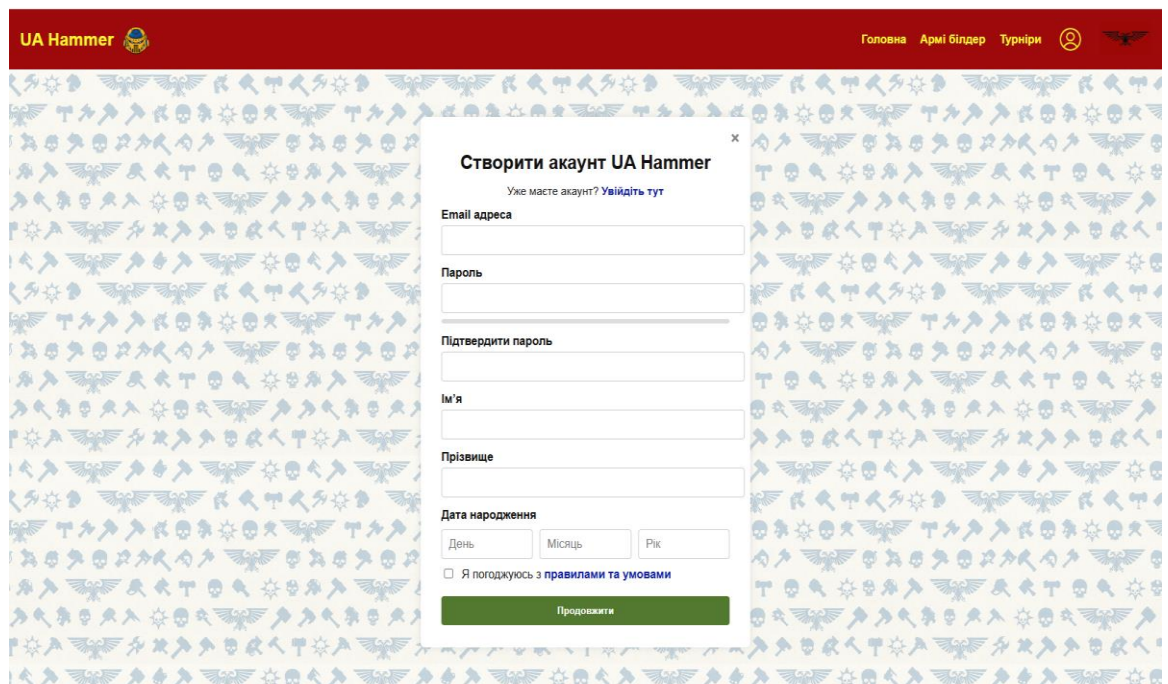


Рис. 3.6. Сторінка реєстрації нового користувача

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		59

У системі також передбачено механізм відновлення пароля. Якщо користувач забув пароль, він може перейти на сторінку відновлення та вказати email, який прив'язаний до облікового запису. Після надсилання запиту система формує запис на відновлення пароля та надсилає користувачу лист із посиланням для зміни пароля.

Рис. 3.7. Сторінка запиту на відновлення пароля

Після обробки запиту користувач отримує електронний лист, у якому міститься службове посилання для переходу до форми зміни пароля. Такий підхід дозволяє підтвердити, що запит на відновлення був створений саме власником відповідної електронної адреси.

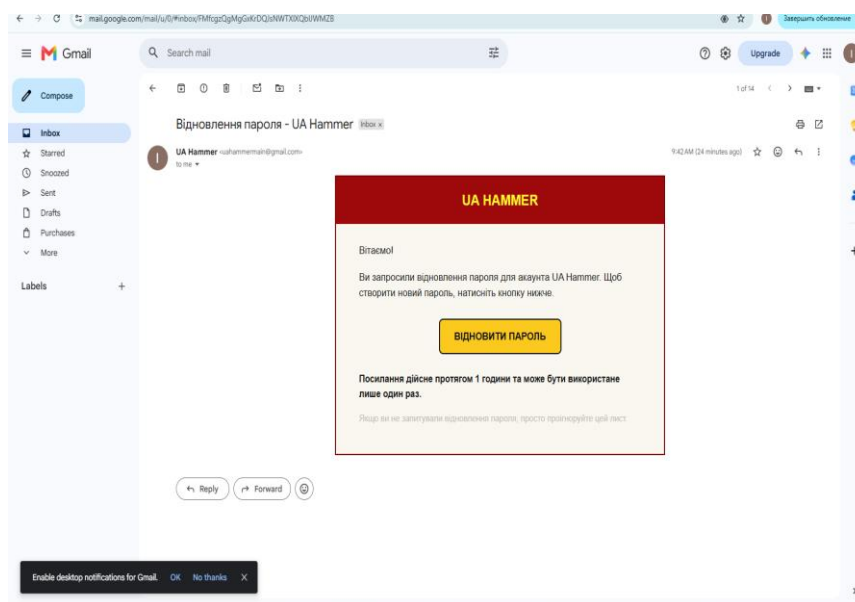


Рис. 3.8. Лист для відновлення пароля, надісланий користувачу

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				60
Змн.	Арк.	№ докум.	Підпис	Дата		

Після переходу за отриманим посиланням відкривається сторінка введення нового пароля. На цій сторінці користувач вказує новий пароль і його підтвердження. Після успішної перевірки новий пароль зберігається в системі, а користувач надалі може виконати авторизацію з новими обліковими даними.

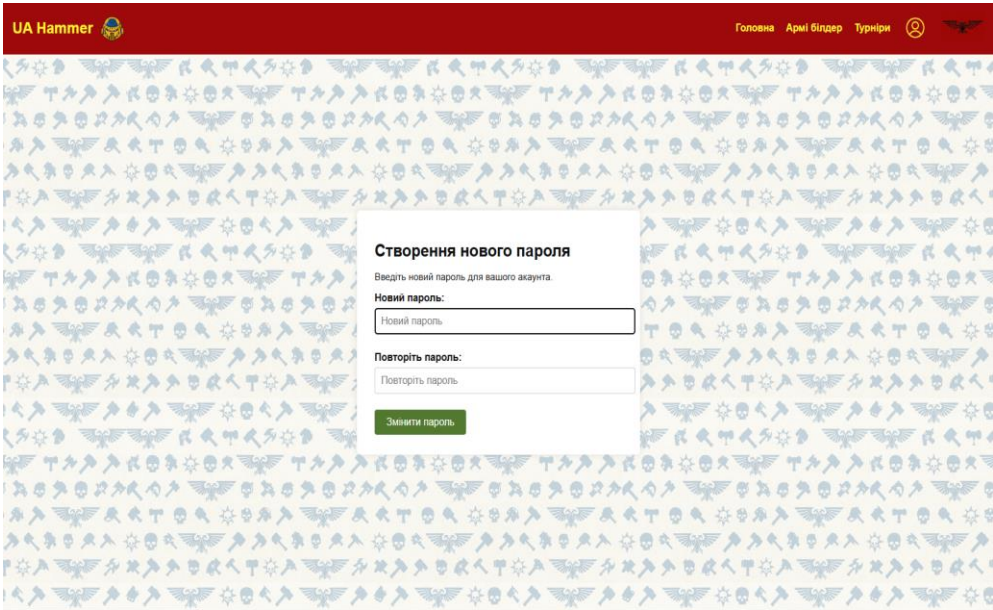


Рис. 3.9. Сторінка введення нового пароля

Після авторизації користувач переходить до особистого кабінету. У ньому відображаються основні дані користувача: email, ім'я, роль та дата створення акаунта. З особистого кабінету користувач може перейти до власних ростерів або вийти із системи. Якщо авторизований користувач має роль адміністратора, у кабінеті додатково відображається кнопка переходу до редактора юнітів.

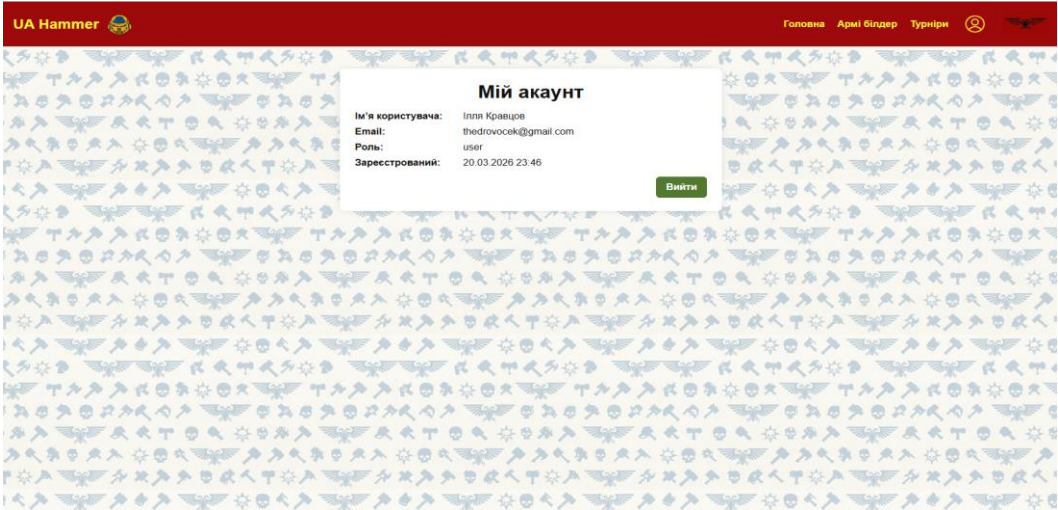


Рис. 3.10. Особистий кабінет користувача

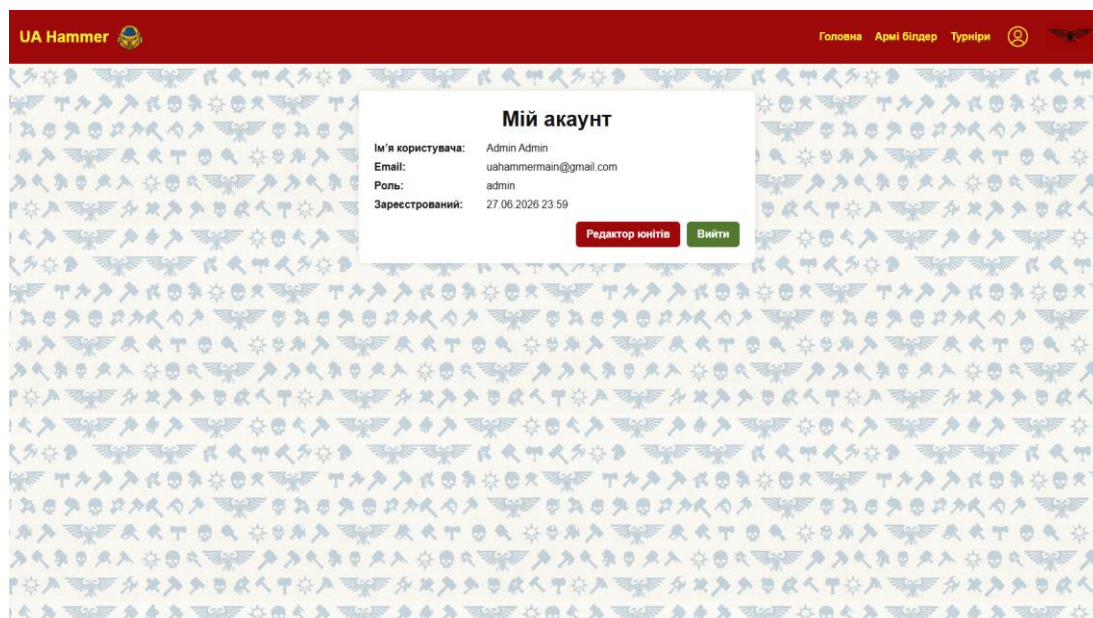


Рис. 3.11. Особистий кабінет адміністратора

Одним з основних розділів системи є армі білдер. На сторінці списку ростерів користувач бачить усі створені ним армійські списки. Кожен ростер відображається у вигляді окремої картки з назвою, фракцією, підфракцією, лімітом очок і датою створення. З цієї сторінки користувач може відкрити ростер для редагування або видалити його.

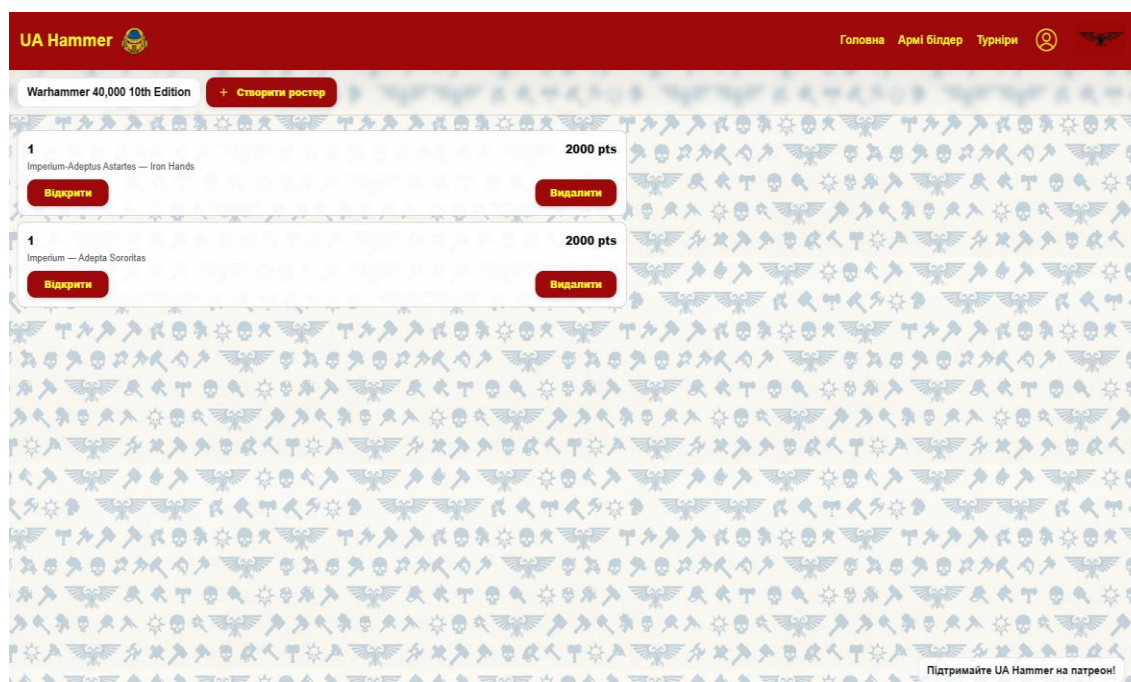


Рис. 3.12. Список армійських ростерів користувача

Створення нового ростера виконується через окрему форму. У ній користувач вказує назву армійського списку, ігрову систему, основну фракцію,

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		62

підфракцію та ліміт очок. Після надсилання форми система створює новий запис у базі даних і переводить користувача на сторінку редагування ростера.

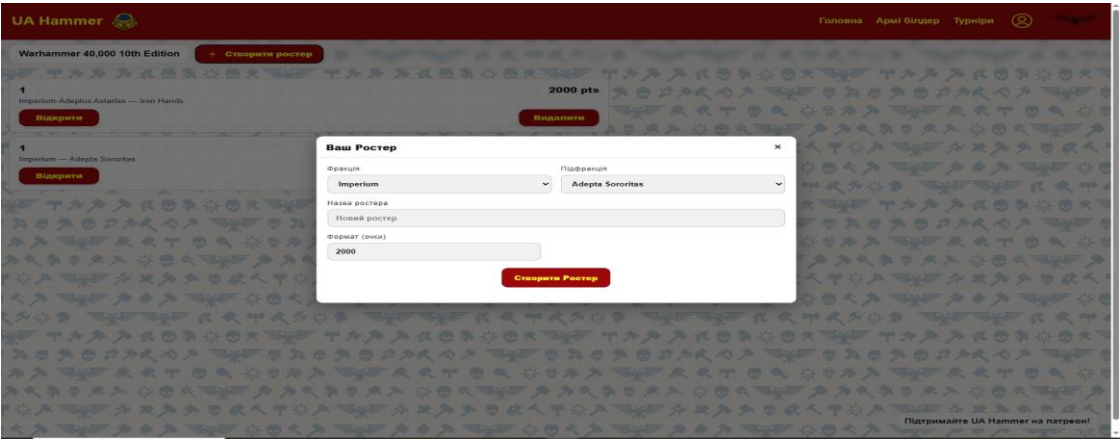


Рис. 3.13. Форма створення нового армійського ростера

На сторінці редагування ростера користувач бачить два основні блоки: список доступних юнітів і список уже доданих юнітів. Система показує лише ті юніти, які відповідають вибраній фракції та підфракції. Користувач може додавати юнітів до ростера, видаляти їх, переглядати загальну вартість армійського списку та контролювати залишок очок.

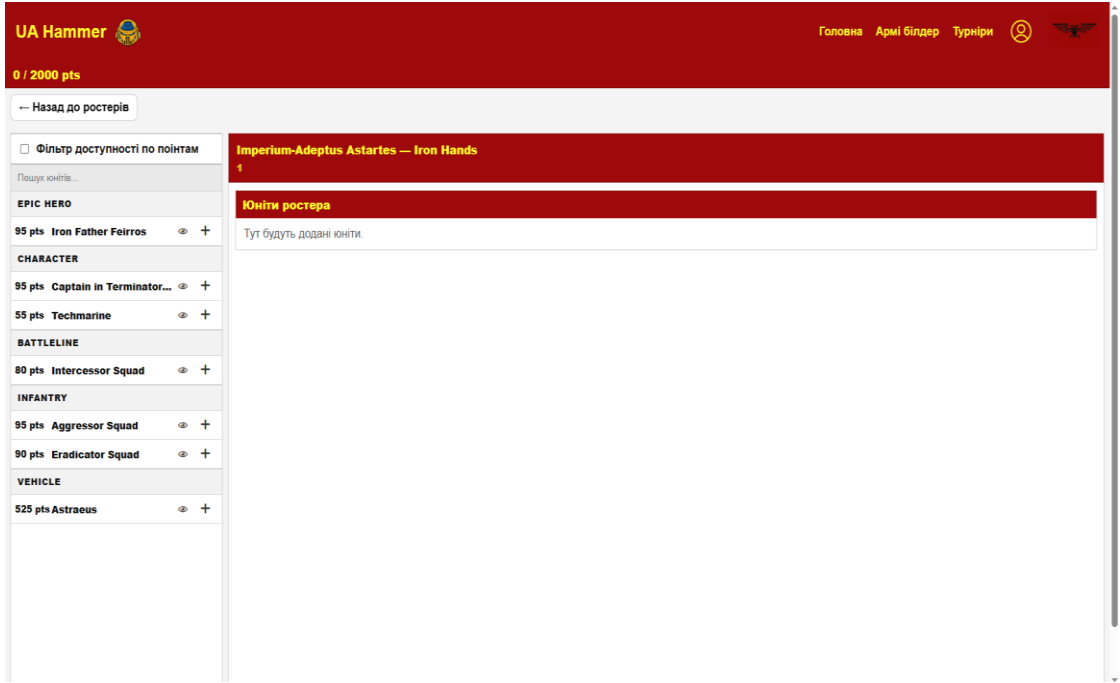


Рис. 3.14. Сторінка редагування армійського ростера

На сторінці редагування ростера також передбачено перегляд детальної інформації про юніта. Користувач може ознайомитися з характеристиками бойової одиниці, де відображаються основні атрибути. Це дозволяє користувачу

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		63

оцінити можливості юніта перед додаванням його до армійського списку і перевірити їх під час гри. Окремо відображається зброя, прив’язана до юніта. Для кожної одиниці зброї показуються її характеристики, якщо вона має спеціальні правила, вони також виводяться поруч із її характеристиками. Завдяки цьому користувач може не лише додати юніта до ростера, а й переглянути його бойові можливості без переходу до сторонніх джерел.

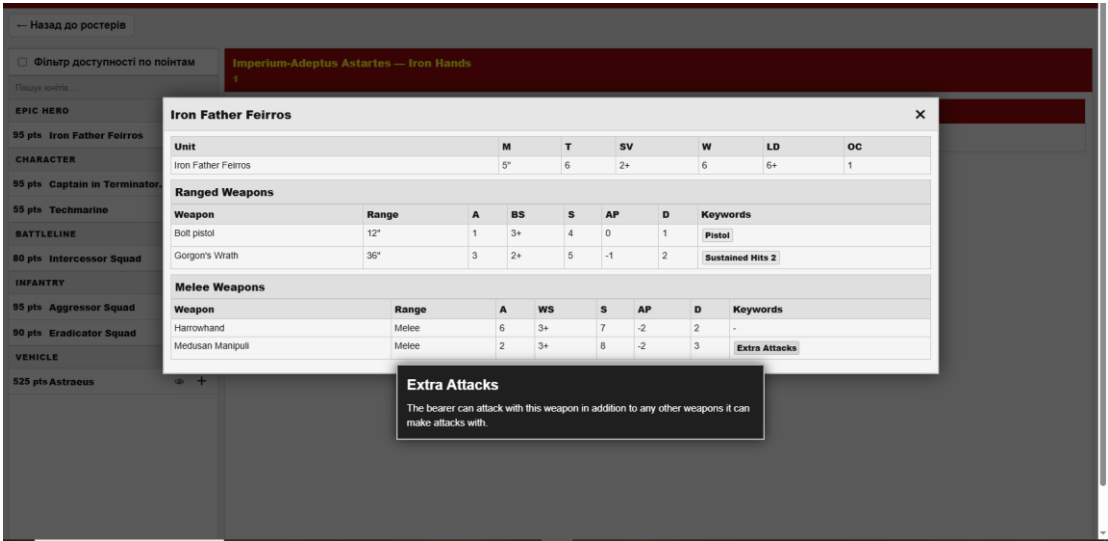


Рис. 3.15. Перегляд характеристик та зброї юніта

Турнірний модуль містить сторінку зі списком доступних турнірів. На ній користувач може переглянути назву турніру, дату проведення, місце, категорію, вартість участі та кількість учасників. Якщо турнір ще має вільні місця, користувач може записатися на нього, або покинути його.

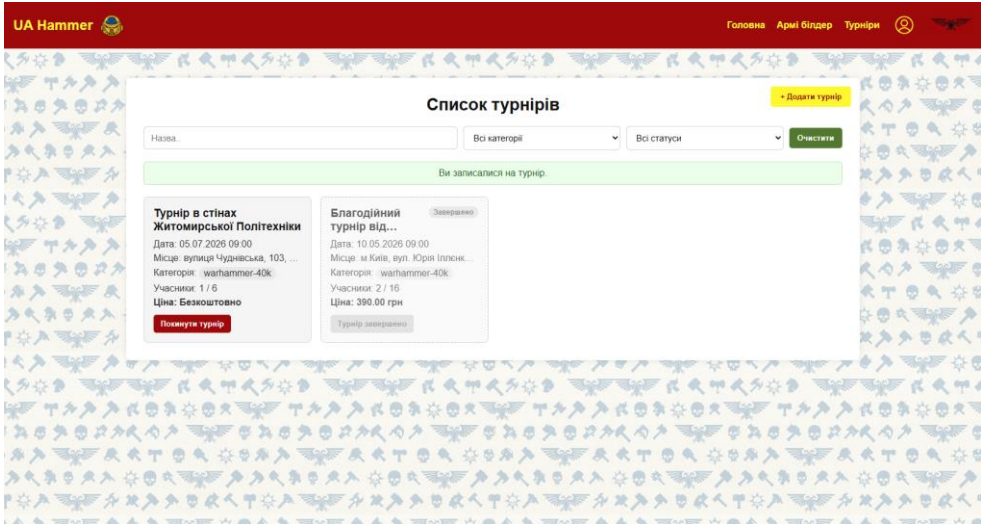


Рис. 3.16. Список турнірів системи

При натисканні на картку турніру відкривається модальне вікно з короткою інформацією про подію. У цьому вікні користувач може переглянути основні дані турніру: назву, категорію, дату та час проведення, місце, вартість участі, організатора, короткий опис і список учасників. Якщо користувачу потрібна повна інформація або додаткові можливості керування турніром, він може натиснути кнопку “Детальніше”.

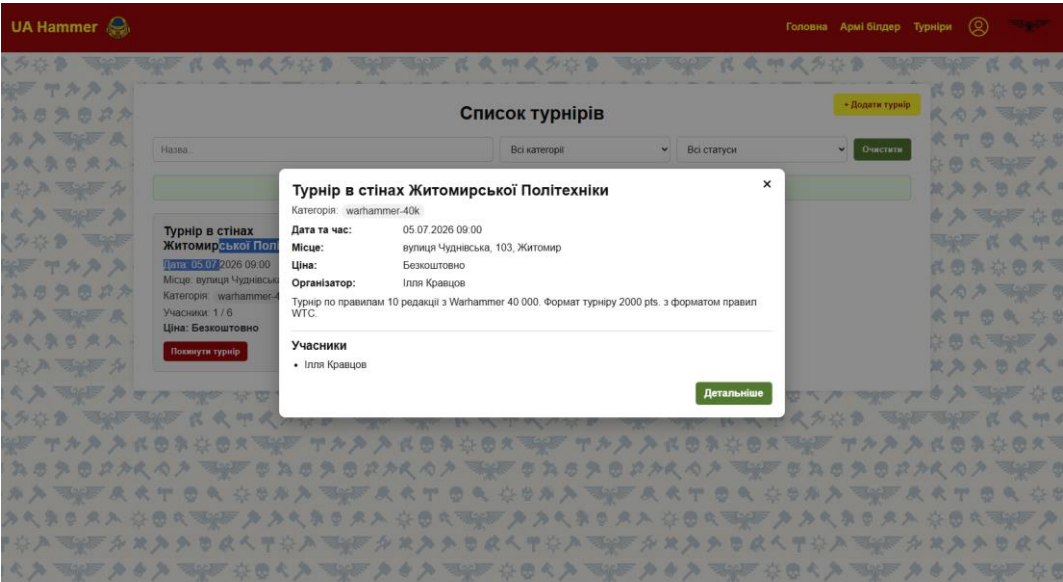


Рис. 3.17. Модальне вікно з короткою інформацією про турнір

Сторінка окремого турніру містить детальну інформацію про подію. На ній відображається опис турніру, дата проведення, місце, максимальна кількість учасників та список зареєстрованих гравців. Організатор турніру має додаткові можливості: редагування турніру, видалити турнір, формування раундів, створення пар гравців і внесення результатів матчів.

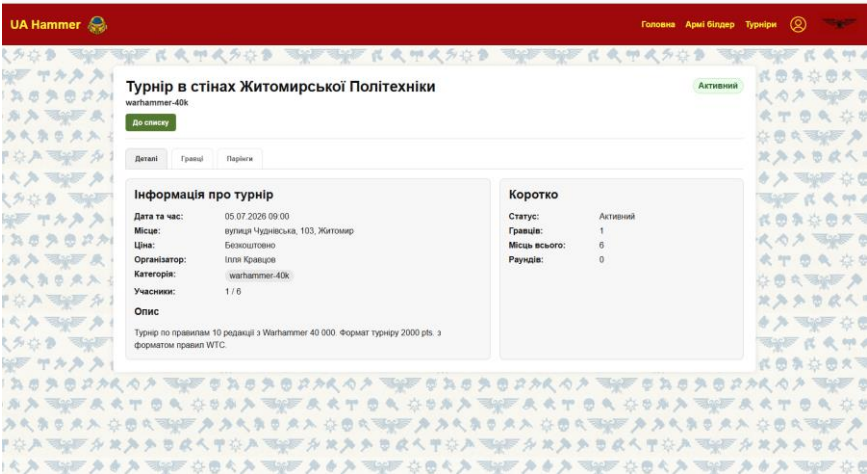


Рис. 3.18. Сторінка окремого турніру для користувача

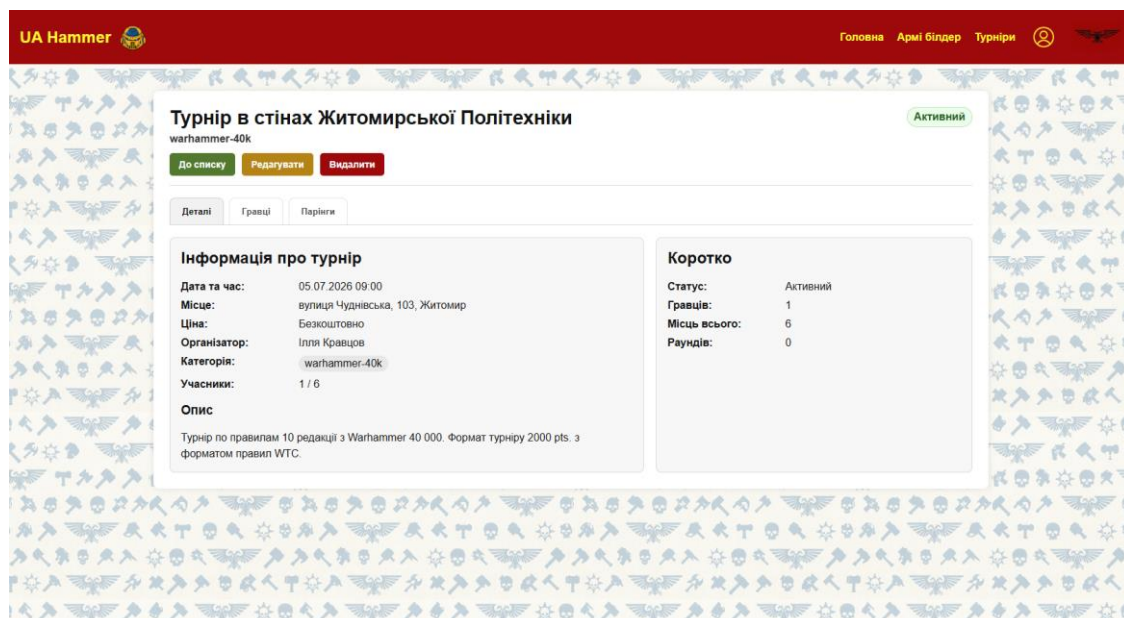


Рис. 3.19. Сторінка окремого турніру для організатора

Формування парінгів використовується для організації матчів у межах конкретного раунду. Організатор створює пари гравців вручну або автоматично, після чого система зберігає інформацію про раунд і матчі в базі даних. Для кожного матчу вказується номер столу, перший і другий гравець, статус матчу та ознака гри проти проксібота, якщо кількість учасників непарна.

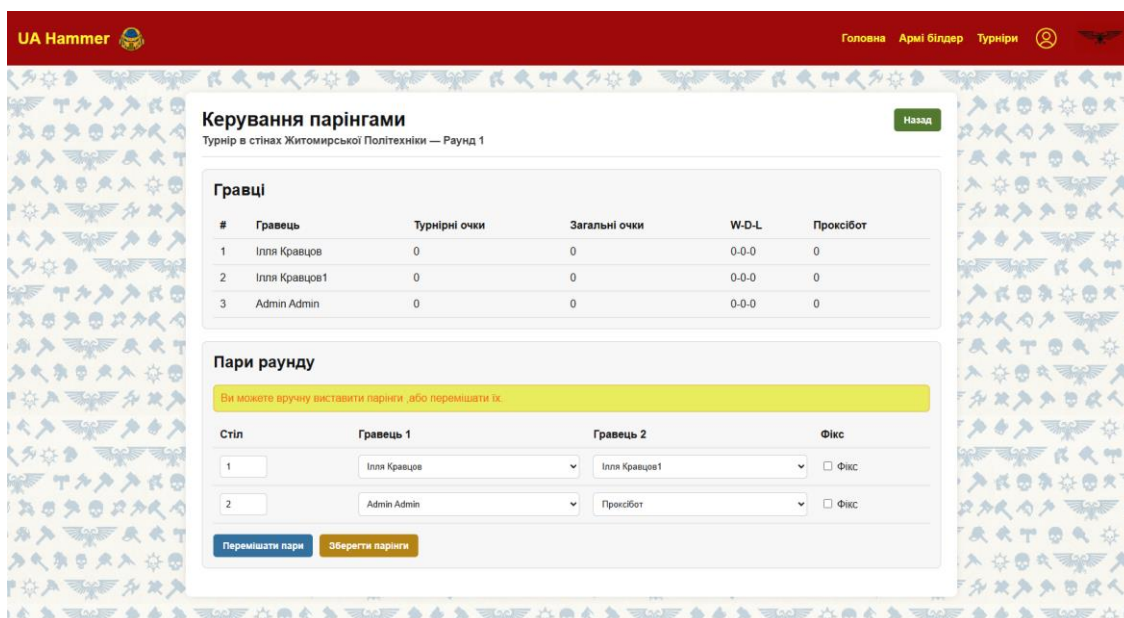


Рис. 3.20. Формування парінгів турніру

Після завершення раунду організатор вносить результати гри. На сторінці результату вказується рахунок першого та другого гравця. Після збереження

система визначає переможця або нічию, нараховує турнірні очки та оновлює статистику учасників: кількість перемог, нічий, поразок і загальний рахунок. Інші користувачі можуть проглянути результати, але не змінювати їх.

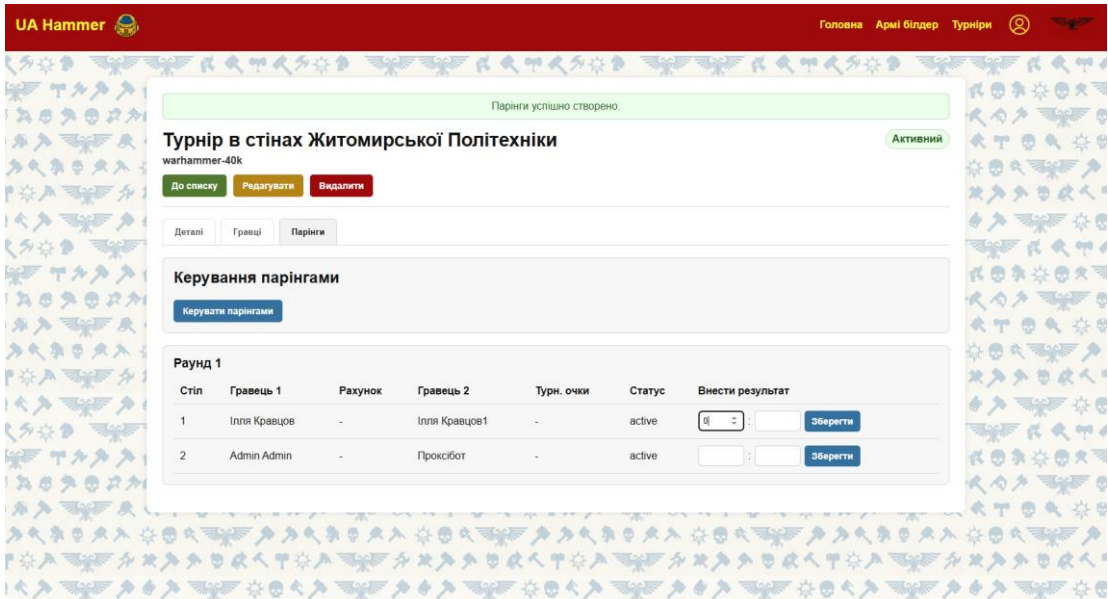


Рис. 3.21. Внесення результатів раунду

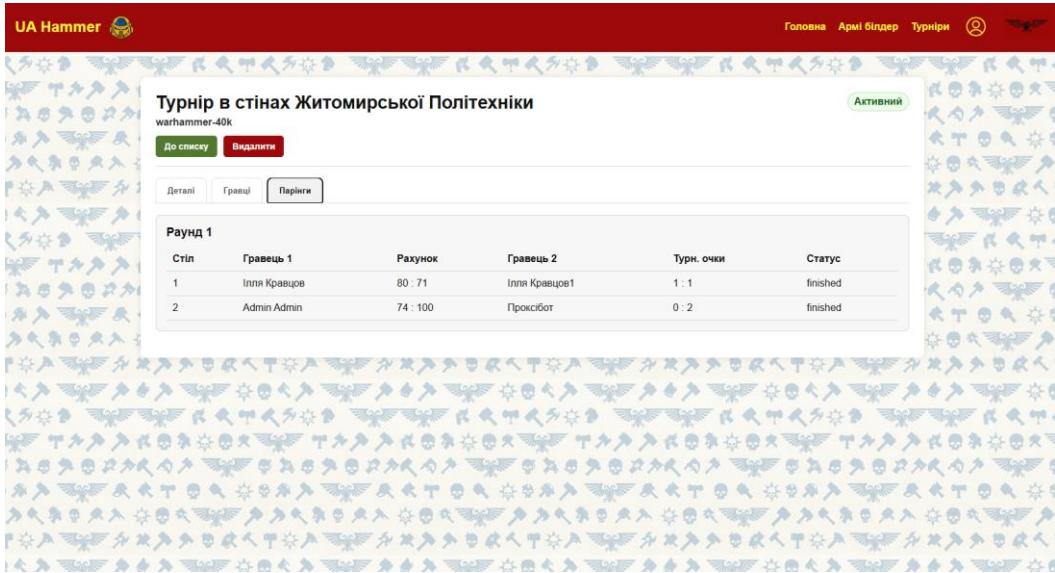


Рис. 3.22. Результатів раунду для користувачів

Після внесення результатів система також оновлює загальну турнірну таблицю. У ній відображаються всі учасники турніру та їхні показники: кількість турнірних очок, загальний набраний рахунок, перемоги, нічий, поразки та кількість матчів проти проксібота. Загальний рахунок формується на основі результатів зіграних матчів і дозволяє порівнювати гравців не лише за перемогами, а й за сумою набраних очок у партіях. Завдяки цьому організатор і

учасники можуть бачити поточне положення гравців у турнірі після кожного раунду.

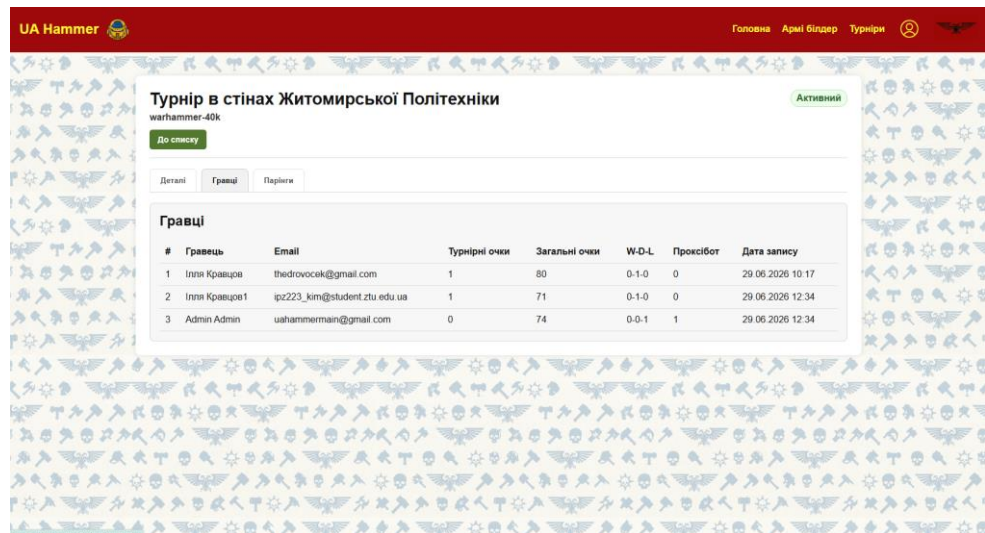


Рис. 3.23. Поточний результат матчу

Адміністративна частина системи доступна лише користувачам із роллю admin. У конструкторі юнітів він може створювати нові бойові одиниці, редагувати існуючі юніти їх характеристики, прикріплювати зброю, задавати категорію, вартість в очках, фракцію і доступність юніта в списку. Завдяки цьому відбувається керування юнітами, які будуть доступні користувачам в армії білдері.

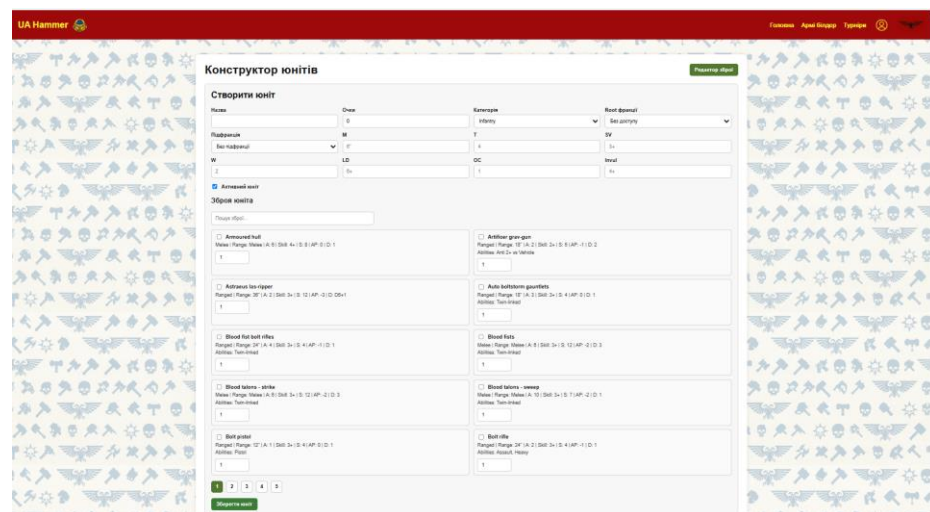


Рис. 3.24. Форма створення та редагування юніта

Список юнитів						
ID	Назва	Очки	Категорія	Характеристики	Деталь	Зброя
4	Iron Father Fetters	95	Elite Hero	M.P. T:6, S:2+, W:6, LD B: GC1	Impetum-Adeplus Adaptives — Iron Fetters	Bolt pistol, Grogg's Vitah, Harmsand, Medusan Matpail
6	Captain in Armour	95	Character	M.P. T:8, S:2+, W:8, LD B: GC1, Incl 4+	Impetum-Adeplus Adaptives	Combi-weapon, Relic fist, Relic weapon, Storm boiler
10	Catapult	70	Character	M.P. T:4, S:2+, W:4, LD B: GC1	Impetum-Adeplus Adaptives — Black Temple	Catapult Adaptives chainmower, Combi-weapon, Heavy bolt pistol, Master-crafted power weapon
14	Insigher	65	Character	M.P. T:8, S:2+, W:8, LD B: GC1, Incl 4+	Impetum — Adeplus Sorales	Bolt pistol, Boltgun, Sorales close combat weapon
3	Technician	55	Character	M.P. T:4, S:2+, W:4, LD B: GC1	Impetum-Adeplus Adaptives	Forge buffer, Gray-pail, Omnisson power arm, Servo-arm
13	Tor Guard	90	Character	M.P. T:8, S:2+, W:8, LD B: GC1, Incl 4+	Impetum-Adeplus Adaptives — Imperial Fists	Artificial grav-gun, Hand of Defiance
2	Intensator Squad	80	Buffette	M.P. T:4, S:2+, W:4, LD B: GC2	Impetum-Adeplus Adaptives	Bolt pistol x5, Bolt rifle x5, Close combat weapon x5
1	Aggressor Squad	95	Infantry	M.P. T:8, S:2+, W:8, LD B: GC1	Impetum-Adeplus Adaptives	Auto ballstom gauntlets, Planeform gauntlets, Fragitorm grenade launcher, Twin power fists
8	Eradicator Squad	90	Infantry	M.P. T:8, S:2+, W:8, LD B: GC1	Impetum-Adeplus Adaptives	Bolt pistol, Close combat weapon, Meta rifle, Multi-melta
12	Inner Circle Companions	90	Infantry	M.P. T:4, S:2+, W:4, LD B: GC2	Impetum-Adeplus Adaptives — Dark Angels	Calabrate greatsword - strike, Calabrate greatsword - snmp, Heavy bolt pistol
5	Asiatum	525	Vehicle	M.P. T:12, S:2+, W:24, LD B: GC1, Incl 4+	Impetum-Adeplus Adaptives	Armoured hull, Asiatum las eyes, Ionball heavy shatles, Plasma eradicator - standard, Plasma eradicator - supercharge, Storm boiler, Twin heavy boiler, Twin laserion, Twin macro-accelerator canon
11	Death Company Deathought	180	Vehicle	M.P. T:10, S:2+, W:12, LD B: GC4	Impetum-Adeplus Adaptives — Blood Angels	Blood fist bolt rifle, Blood fists, Blood talons - strike, Blood talons - snmp, Twin heavy boiler, Twin lance mortar heavy shatles, Twin multi-melta

Рис. 3.25. Список юнітів для редагування

Окремо в системі реалізовано редактор зброї. На цій сторінці адміністратор може створювати нову зброю, редагувати існуючу її тип, дальність, кількість атак, навичку, силу, пробиття броні та шкоду. Також до зброї можна прив'язувати спеціальні здібності.

[illegible]

Рис. 3.26. Форма створення та редагування зброї

Список зброї									
Пошук зброї:									
ID	Назва	Тип	Range	A	DSWS	S	AP	D	Abilities
16	Jamocured hat	Melee	Melee	6	4+	0	-	1	-
43	Afficion grey-gun	Ranged	18"	2	2+	5	-	1	Anti 2+ Vehicle
6	Ardisma tas-sipper	Ranged	36"	2	3+	12	-	3	DD-1
21	Azula ballroom panderite	Ranged	18"	3	3+	4	-	1	Two-Instad
35	Blood hat hot rifles	Ranged	24"	4	3+	4	-	1	Two-Instad
30	Blood hats	Melee	Melee	6	3+	12	-	2	Two-Instad
39	Blood latons - Mike	Melee	Melee	6	3+	12	-	2	Two-Instad
40	Blood latons - swamp	Melee	Melee	10	3+	7	-	2	Two-Instad
2	Ball point	Ranged	12"	1	3+	4	-	1	Platf
1	Ball rifle	Ranged	24"	2	3+	4	-	1	Assault, Heavy

Рис. 3.27. Список зброї для редагування

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				69
Змн.	Арк.	№ докум.	Підпис	Дата		

Таким чином, структура вебінтерфейсу UA Hammer складається з кількох логічно пов'язаних частин: головної сторінки, авторизації та реєстрації, особистого кабінету, армі білдера, турнірного модуля та адміністративних редакторів. Такий поділ дозволяє користувачу швидко знаходити потрібний функціонал, а адміністратору - підтримувати актуальність даних про юнітів, зброю та турніри.

3.3 Тестування роботи системи

Для перевірки працездатності та продуктивності вебзастосунку UA Hammer було проведено тестування основних сторінок системи. Оскільки проєкт розгорнуто локально за допомогою XAMPP та відкривається через адресу localhost, для перевірки через зовнішні онлайн-сервіси було використано ngrok. Цей інструмент дозволяє створити тимчасове публічне посилання на локальний сервер Apache. Його використання дало можливість протестувати вебзастосунок в умовах, наближених до роботи на віддаленому сервері. Це дозволило використати для перевірки сторонні сервіси аналізу продуктивності та перевірити коректність роботи сторінок через інтернет без розгортання системи на хостингу.

Перед початком тестування було запущено модулі Apache та MySQL у XAMPP. Після цього в командному рядку було виконано запуск тунелю до локального сервера на порту 80. У результаті ngrok сформував публічне посилання, через яке вебзастосунок «UA Hammer» став доступним не лише локально, а й через інтернет.

```

ngrok
One gateway for every AI model. Available in early access *now*: https://ngrok.com/r/ai

Session Status      online
Account             thedrovocek@gmail.com (Plan: Free)
Version             3.39.8-msix-stable
Region              Europe (eu)
Latency              27ms
Web Interface       http://127.0.0.1:4040
Forwarding            https://egotistic-creamed-unicycle.ngrok-free.dev -> http://localhost:80

Connections
  ttl   opn   rt1   rt5   p50   p90
    8    0    0.00  0.01  3.19  15.53

HTTP Requests
-----
  
```

Рис. 3.28. Запуск вебзастосунку UA Hammer через ngrok

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				70
Змн.	Арк.	№ докум.	Підпис	Дата		

Створення тунелю дало вебзастосунку тимчасову публічну адресу, яка перенаправляє всі запити до локального вебзастосунку.

Після отримання публічного посилання було перевірено відкриття головної сторінки вебзастосунку через адресу ngrok. Це дозволило переконатися, що локальний вебсервер Apache коректно обробляє запити, а сторінки системи можуть бути відкриті через зовнішнє посилання.

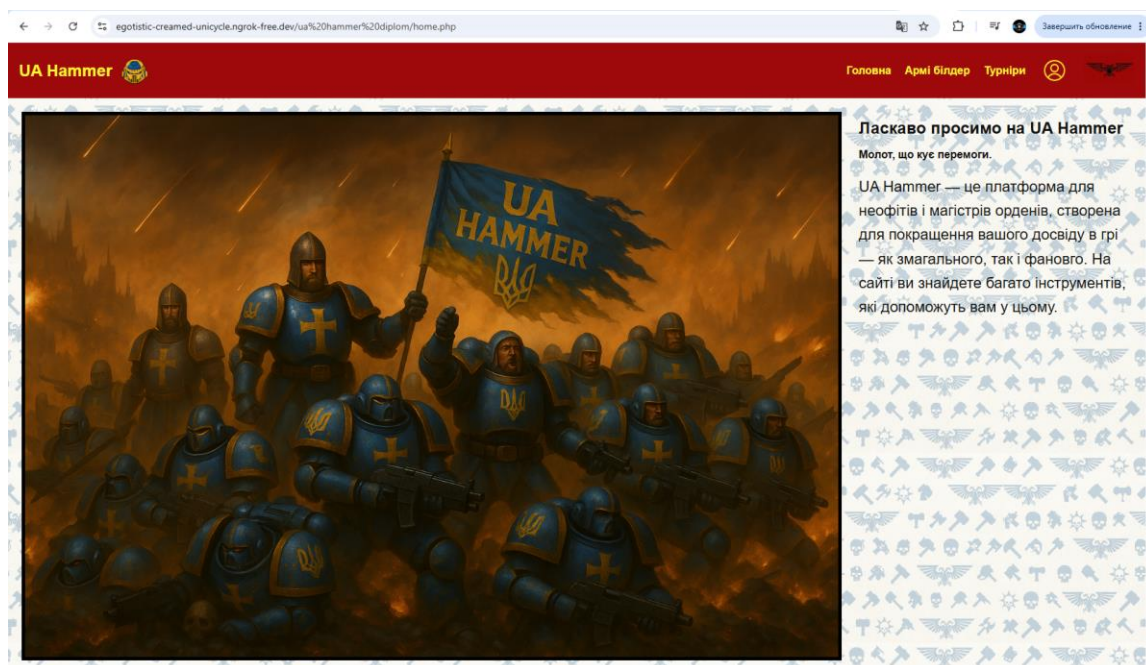


Рис. 3.29. Відкриття вебзастосунку UA Hammer через публічне посилання ngrok

Під час перевірки було підтверджено коректне завантаження головної сторінки, відображення графічних елементів інтерфейсу та роботу навігації між основними сторінками вебзастосунку. Також було перевірено доступність системи без використання локальної адреси localhost.

Для тестування продуктивності було використано сервіс PageSpeed Insights. Цей інструмент дозволяє перевірити швидкість завантаження вебсторінки, доступність інтерфейсу, відповідність кращим практикам та рівень оптимізації сторінки.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		71

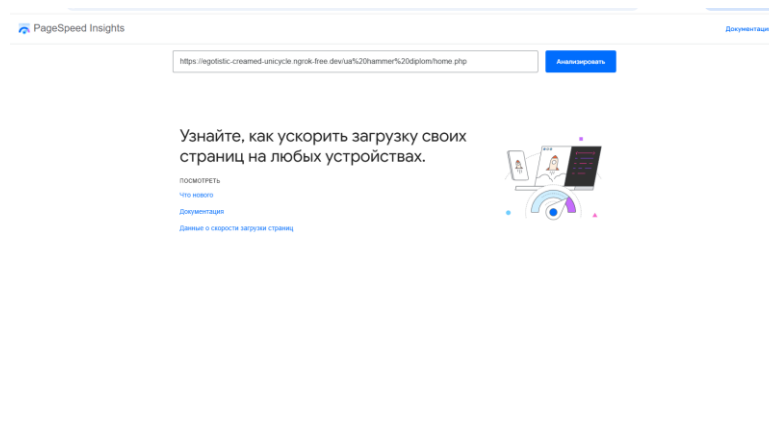


Рис. 3.30. Введення публічного посилання UA Hammer у PageSpeed Insights

Після введення адреси сервіс автоматично виконує аналіз сторінки та формує звіт, який містить оцінку швидкодії й рекомендації щодо можливого покращення продуктивності вебзастосунку. Це дозволяє визначити сильні та слабкі сторони реалізації клієнтської частини системи.

Після завершення аналізу було отримано звіт із результатами тестування. У звіті відображаються основні показники продуктивності сторінки, зокрема Performance, Accessibility, Best Practices та SEO. Ці показники дозволяють оцінити, наскільки швидко завантажується сторінка, чи зручною вона є для користувачів, чи відповідає базовим рекомендаціям якості веброзробки та чи має коректну структуру для пошукових систем.

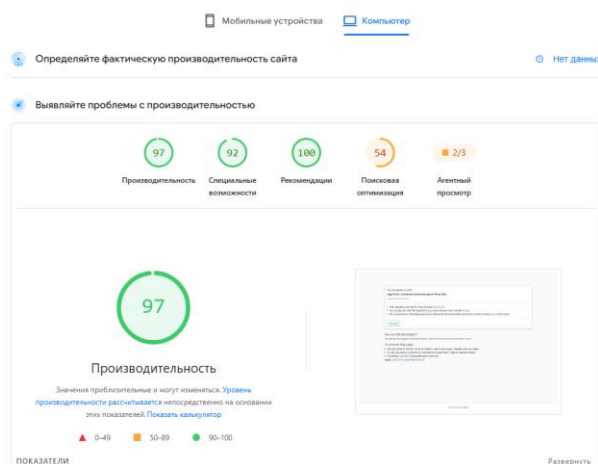


Рис. 3.31. Результати тестування головної сторінки UA Hammer у PageSpeed Insights

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				72
Змн.	Арк.	№ докум.	Підпис	Дата		

Отримані результати дозволяють оцінити якість реалізації вебінтерфейсу та визначити можливі напрями його оптимізації. Аналіз показників продуктивності є важливим етапом перевірки вебзастосунку, оскільки безпосередньо впливає на швидкість роботи та комфорт використання системи.

Окрему увагу було приділено показникам швидкості завантаження сторінки. PageSpeed Insights аналізує такі метрики, як First Contentful Paint, Largest Contentful Paint, Total Blocking Time та Cumulative Layout Shift. Ці показники дозволяють визначити, наскільки швидко користувач бачить перший вміст сторінки, коли завантажується найбільший елемент сторінки, чи є затримки під час взаємодії та чи відбуваються небажані зміщення елементів інтерфейсу.

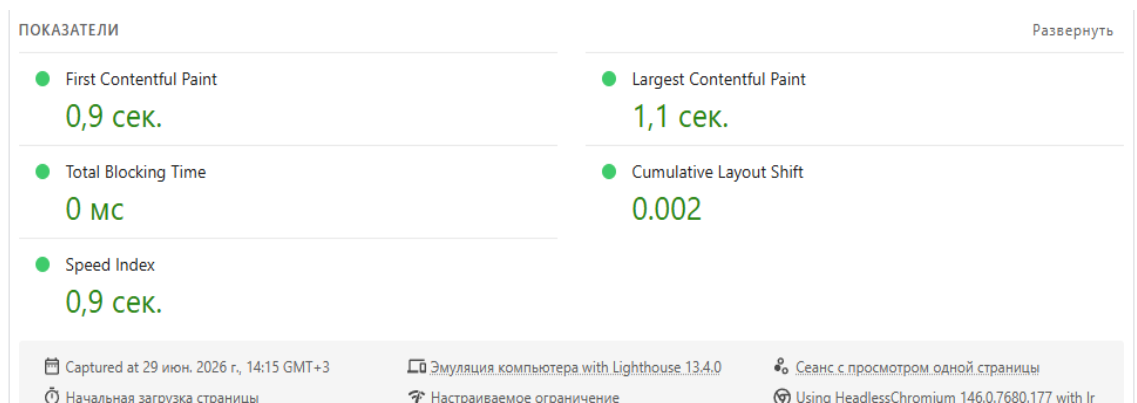


Рис. 3.32. Перегляд основних метрик продуктивності сторінки

За результатами тестування було встановлено, що вебзастосунок UA Hammer коректно працює в локальному середовищі XAMPP та може бути тимчасово відкритий через публічне посилання ngrok. Використання PageSpeed Insights дозволило оцінити основні показники продуктивності сторінки, зокрема швидкість завантаження, час відображення основного вмісту та стабільність елементів інтерфейсу. Проведена перевірка підтвердила працездатність вебзастосунку, коректну роботу локального сервера Apache та можливість демонстрації системи через зовнішнє посилання.

Висновки до третього розділу.

У третьому розділі було розглянуто порядок встановлення, налаштування та перевірки роботи вебзастосунку UA Hammer. Було описано підготовку

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		73

локального середовища за допомогою XAMPP, запуск модулів Apache та MySQL, створення бази даних MySQL/MariaDB і розміщення файлів проєкту в каталозі htdocs. Також було наведено налаштування підключення до бази даних через файл connector.php, у якому використовується PDO для безпечної взаємодії PHP-коду з базою даних.

Далі було описано структуру вебінтерфейсу системи. Було розглянуто головну сторінку, сторінки авторизації та реєстрації, механізм відновлення пароля, особистий кабінет користувача, армі білдер, турнірний модуль та адміністративні редактори юнітів та зброї. Інтерфейс системи побудований таким чином, щоб користувач міг створювати армійські ростери, переглядати доступних юнітів, брати участь у турнірах, а адміністратор - керувати основними довідковими даними системи.

Окрему увагу було приділено тестуванню роботи вебзастосунку. Для перевірки доступності системи через зовнішнє посилання було використано ngrok, який дозволив тимчасово відкрити локальний вебзастосунок через інтернет. Для оцінювання продуктивності сторінки було застосовано PageSpeed Insights, за допомогою якого перевірено основні показники швидкості завантаження.

За результатами виконаної роботи було встановлено, що вебзастосунок UA Hammer коректно працює в локальному середовищі XAMPP, успішно взаємодіє з базою даних MySQL/MariaDB та може бути продемонстрований через публічне посилання ngrok. Проведена перевірка підтвердила працездатність основних сторінок системи, коректність її запуску та можливість подальшого розширення функціоналу.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		74

ВИСНОВКИ

У процесі виконання випускної кваліфікаційної роботи було спроектовано та реалізовано вебзастосунок UA Hammer, призначений для створення армійських ростерів, перегляду інформації про юнітів, роботи з турнірами та адміністрування даних системи. Основна увага була зосереджена на створенні зручного інструмента для гравців Warhammer 40,000, який дозволяє формувати армійські списки, переглядати доступні бойові одиниці та брати участь у турнірних подіях.

У першому розділі було проведено аналіз предметної області та розглянуто існуючі рішення, які мають схожий функціонал. Було визначено їхні переваги та недоліки, а також сформульовано основні вимоги до майбутнього вебзастосунку. На основі цього було обґрунтовано необхідність створення системи UA Hammer, яка поєднує функції армі білдера, інформаційного ресурсу та турнірного модуля.

У другому розділі було виконано проектування вебзастосунку. Було визначено основні сценарії роботи користувача, адміністратора та організатора турніру. Також було спроектовано структуру бази даних MySQL/MariaDB, яка містить таблиці для збереження користувачів, армійських ростерів, юнітів, зброї, спеціальних правил, турнірів, учасників, раундів і матчів. Окремо було описано основні алгоритми роботи системи, зокрема створення ростера, додавання юнітів, створення турніру, формування парінгів і внесення результатів матчів.

У третьому розділі було розглянуто порядок встановлення та налаштування системи. Було описано запуск вебзастосунку в локальному середовищі XAMPP, налаштування Apache та MySQL, створення бази даних і підключення до неї через файл connector.php. Також було розглянуто структуру вебінтерфейсу системи: головну сторінку, авторизацію, реєстрацію, відновлення пароля, особистий кабінет, армі білдер, турнірний модуль та адміністративні редактори юнітів і зброї. Окрему увагу було приділено тестуванню роботи системи. Для перевірки доступності вебзастосунку через зовнішнє посилання було використано ngrok, а для оцінювання продуктивності сторінки - PageSpeed Insights. У результаті було перевірено основні показники швидкості

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		75

завантаження, зокрема First Contentful Paint, Largest Contentful Paint, Total Blocking Time, Speed Index та Cumulative Layout Shift. Проведена перевірка підтвердила, що вебзастосунок коректно працює в локальному середовищі та може бути продемонстрований через публічне посилання.

Підсумовуючи, у межах роботи було створено працездатний вебзастосунок UA Hammer, який реалізує основні функції для роботи з армійськими ростерами, юнітами, зброєю та турнірами. Система має зрозумілу структуру, зберігає дані в базі MySQL/MariaDB і може бути розширена новими можливостями в майбутньому. До таких можливостей можна віднести детальні сторінки юнітів, розширену статистику турнірів, покращену систему фільтрації та модуль наративних кампаній.

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		76

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. New Recruit – Army Builder and List Management Tool. URL: <https://www.newrecruit.eu/> (дата звернення: 10.03.2026).
2. Tabletop Admiral – Warhammer 40,000 Army Builder. URL: <https://www.tabletopadmiral.com/> (дата звернення: 10.03.2026).
3. Warlord Games – Official Website. URL: <https://warlordgames.com/> (дата звернення: 10.03.2026).
4. Games Workshop. Warhammer 40,000. URL: <https://www.warhammer.com/> (дата звернення: 11.04.2026).
5. PHP Documentation. URL: <https://www.php.net/docs.php> (дата звернення: 15.04.2026).
6. PHP Manual. Sessions. URL: <https://www.php.net/manual/en/book.session.php> (дата звернення: 16.04.2026).
7. PHP Manual. Password Hashing Functions. URL: <https://www.php.net/manual/en/ref.password.php> (дата звернення: 18.04.2026).
8. MySQL Documentation. URL: <https://dev.mysql.com/doc/> (дата звернення: 10.05.2026).
9. MDN Web Docs. HTML: HyperText Markup Language. URL: <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення: 11.05.2026).
10. MDN Web Docs. Cascading Style Sheets (CSS). URL: <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення: 22.05.2026).
11. MDN Web Docs. JavaScript. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення: 22.05.2026).
12. PHP Manual. PDO. URL: <https://www.php.net/manual/en/book.pdo.php> (дата звернення: 23.05.2026).

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				77
Змн.	Арк.	№ докум.	Підпис	Дата		

13. PHP Manual. PDO MySQL Driver. URL: <https://www.php.net/manual/en/ref.pdo-mysql.php> (дата звернення: 23.05.2026).
14. Apache Friends. XAMPP. URL: <https://www.apachefriends.org/community.html> (дата звернення: 24.05.2026).
15. phpMyAdmin Documentation. URL: <https://docs.phpmyadmin.net/en/latest/> (дата звернення: 24.05.2026).
16. Google PageSpeed Insights. URL: <https://pagespeed.web.dev/> (дата звернення: 25.05.2026).
17. ngrok Documentation. URL: <https://ngrok.com/docs/start> (дата звернення: 25.05.2026).
18. Bootstrap Documentation. URL: <https://getbootstrap.com/docs/5.3/getting-started/introduction/> (дата звернення: 26.05.2026).
19. W3C. Web Content Accessibility Guidelines (WCAG). URL: <https://www.w3.org/WAI/standards-guidelines/wcag/> (дата звернення: 26.05.2026).
20. SQL Tutorial. W3Schools. URL: <https://www.w3schools.com/sql/> (дата звернення: 27.05.2026).
21. MDN Web Docs. HTTP Overview. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview> (дата звернення: 27.05.2026).
22. MDN Web Docs. Fetch API. URL: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API (дата звернення: 27.05.2026).
23. MDN Web Docs. JSON. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON (дата звернення: 27.05.2026).

24. PHP Manual. filter_input. URL:
<https://www.php.net/manual/en/function.filter-input.php> (дата звернення:
28.05.2026).

25. PHP Manual. PDO::prepare. URL:
<https://www.php.net/manual/en/pdo.prepare.php> (дата звернення: 28.05.2026).

26. PHP Manual. password_verify. URL:
<https://www.php.net/manual/en/function.password-verify.php> (дата звернення:
28.05.2026).

27. MySQL Reference Manual. CREATE TABLE Statement. URL:
<https://dev.mysql.com/doc/refman/8.0/en/create-table.html> (дата звернення:
29.05.2026).

28. OWASP Foundation. SQL Injection Prevention Cheat Sheet. URL:
https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html (дата звернення: 29.05.2026).

29. OWASP Foundation. Authentication Cheat Sheet. URL:
https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html (дата звернення: 29.05.2026).

30. Visual Studio Code Documentation. URL:
<https://code.visualstudio.com/docs> (дата звернення: 30.05.2026).

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		79

Додатки

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		80

conector.php

```
<?php
declare(strict_types=1);

$pdo = new PDO(
    "mysql:host=127.0.0.1;dbname=uahammer_db;charset=utf8mb4",
    "root",
    "",
    [
        PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
        PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    ]
);
```

composer.json

```
{
    "require": {
        "phpmailer/phpmailer": "^7.1"
    }
}
```

log.php

```
<?php
session_start();
require __DIR__ . '/conector.php';

function safe_next($url) {
    if (!$url) return 'account.php';
    if (preg_match('#^https?://#i', $url)) return 'account.php';
    return ltrim($url, '/');
}

$next = $_GET['next'] ?? $_POST['next'] ?? 'account.php';

if (isset($_SESSION["user"]) && is_array($_SESSION["user"])) {
    header("Location: " . safe_next($next));
    exit;
}

$errorMessage = "";

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

    if ($email === "" || $password === "") {
        $errorMessage = "Вкажіть email і пароль!";
    } else {
        $st = $pdo->prepare("
            SELECT id,email,password_hash,display_name,role
            FROM users
            WHERE email=?
            LIMIT 1
```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		81

```

    );
    $st->execute([$email]);
    $user = $st->fetch();

    if (!$user || !password_verify($password, $user["password_hash"])) {
        $errorMessage = "Невірний email або пароль!";
    } else {
        session_regenerate_id(true);

        $_SESSION["user"] = [
            "id" => (int)$user["id"],
            "email" => $user["email"],
            "display_name" => $user["display_name"] ?: $user["email"],
            "name" => $user["display_name"] ?: $user["email"],
            "role" => $user["role"] ?: "user",
        ];

        header("Location: " . safe_next($next));
        exit;
    }
}
}

```

recover_request.php

```

<?php
require_once __DIR__ . '/conector.php';
require_once __DIR__ . '/vendor/autoload.php';

use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

$message = '';

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $email = trim($_POST["email"] ?? "");

    if ($email === "") {
        $message = "Вкажіть email!";
    } else {
        $st = $pdo->prepare("
            SELECT id,email
            FROM users
            WHERE email=?
            LIMIT 1
        ");
        $st->execute([$email]);
        $user = $st->fetch();

        if ($user) {
            $pdo->exec("
                DELETE FROM password_resets
                WHERE expires_at <= NOW()
                OR used_at IS NOT NULL
            ");

            $st = $pdo->prepare("
                DELETE FROM password_resets
                WHERE user_id = ?
            ");
            $st->execute([(int)$user["id"]]);

            $token = bin2hex(random_bytes(32));

```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				82
Змн.	Арк.	№ докум.	Підпис	Дата		

```

$tokenHash = password_hash($token, PASSWORD_DEFAULT);

$stmt = $pdo->prepare("
    INSERT INTO password_resets
        (user_id, token_hash, expires_at)
    VALUES
        (?, ?, DATE_ADD(NOW(), INTERVAL 1 HOUR))
");
$stmt->execute([(int)$user["id"], $tokenHash]);

$resetLink =
    "http://localhost/ua%20hammer%20diplom/recover_reset.php?token="
    . urlencode($token);

try {
    $mail = new PHPMailer(true);

    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'uahammermain@gmail.com';
    $mail->Password = '*****';
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
    $mail->Port = 587;
    $mail->CharSet = 'UTF-8';

    $mail->setFrom('uahammermain@gmail.com', 'UA Hammer');
    $mail->addAddress($email);
    $mail->isHTML(true);
    $mail->Subject = 'Відновлення пароля - UA Hammer';

    $mail->Body = '
        <p>Ви запросили відновлення пароля для акаунта UA Hammer.</p>
        <p><a href="' . $resetLink . '">Відновити пароль</a></p>
        <p>Посилання дійсне протягом 1 години.</p>
    ';

    $mail->send();
    $message = "Лист для відновлення пароля надіслано.";
} catch (Exception $e) {
    $message = "Помилка надсилання листа.";
}
} else {
    $message = "Лист для відновлення пароля надіслано.";
}
}
}

```

recover_reset.php

```

<?php
require_once __DIR__ . '/conector.php';

$message = '';
$errorMessage = '';
$success = false;

$token = $_GET['token'] ?? $_POST['token'] ?? '';
$resetRow = null;

if ($token === '') {
    $errorMessage = 'Посилання для відновлення некоректне.';
} else {

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		83

```

$st = $pdo->prepare("
    SELECT id, user_id, token_hash, expires_at, used_at
    FROM password_resets
    WHERE used_at IS NULL
        AND expires_at > NOW()
    ORDER BY id DESC
");
$st->execute();
$rows = $st->fetchAll();

foreach ($rows as $row) {
    if (password_verify($token, $row['token_hash'])) {
        $resetRow = $row;
        break;
    }
}

if (!$resetRow) {
    $errorMessage = 'Посилання недійсне або термін його дії минув.';
}

if ($_SERVER["REQUEST_METHOD"] === "POST" && $resetRow) {
    $password = $_POST['password'] ?? '';
    $passwordConfirm = $_POST['password_confirm'] ?? '';

    if ($password === '' || $passwordConfirm === '') {
        $errorMessage = 'Заповніть обидва поля пароля.';
    } elseif ($password !== $passwordConfirm) {
        $errorMessage = 'Паролі не співпадають.';
    } elseif (mb_strlen($password) < 6) {
        $errorMessage = 'Пароль має бути не коротший за 6 символів.';
    } else {
        $passwordHash = password_hash($password, PASSWORD_DEFAULT);

        $st = $pdo->prepare("
            UPDATE users
            SET password_hash = ?
            WHERE id = ?
        ");
        $st->execute([$passwordHash, (int)$resetRow['user_id']]);

        $st = $pdo->prepare("
            UPDATE password_resets
            SET used_at = NOW()
            WHERE id = ?
        ");
        $st->execute([(int)$resetRow['id']]);

        $success = true;
        $message = 'Пароль успішно змінено. Тепер можете увійти.';
    }
}
}

```

Додаток С

roster_create.php

```

<?php
session_start();

if (!isset($_SESSION['user']) || !is_array($_SESSION['user'])) {

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		84

```

        header('Location: log.php');
        exit;
    }

    require __DIR__ . '/conector.php';

    $uid = (int)($_SESSION['user']['id'] ?? 0);

    $system      = $_POST['system'] ?? '';
    $factionRoot = $_POST['faction_root'] ?? '';
    $faction      = $_POST['faction'] ?? null;
    $listName     = trim($_POST['list_name'] ?? '');
    $points       = (int)($_POST['points'] ?? 2000);

    if ($system === '' || $factionRoot === '' || $faction === '' || $listName === '') {
        die('Помилка: не всі поля заповнені');
    }

    $rosterCode = bin2hex(random_bytes(12));

    $st = $pdo->prepare("
        INSERT INTO rosters
            (roster_code, user_id, system, faction_root, faction,
             list_name, points, data_json)
        VALUES
            (?, ?, ?, ?, ?, ?, ?, ?)
    ");

    $st->execute([
        $rosterCode,
        $uid,
        $system,
        $factionRoot,
        $faction,
        $listName,
        $points,
        json_encode([], JSON_UNESCAPED_UNICODE)
    ]);

    header('Location: roster.php?id=' . $rosterCode);
    exit;

```

roster_add_unit.php

```

<?php
session_start();
require __DIR__ . '/conector.php';

header('Content-Type: application/json; charset=utf-8');

if (!isset($_SESSION['user']) || !is_array($_SESSION['user'])) {
    echo json_encode([
        'success' => false,
        'message' => 'Не авторизовано'
    ]);
    exit;
}

$uid = (int)($_SESSION['user']['id'] ?? 0);
$rosterCode = $_POST['roster_code'] ?? '';
$unitId = (int)($_POST['unit_id'] ?? 0);

```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		85

```

$st = $pdo->prepare("
    SELECT *
    FROM rosters
    WHERE roster_code = ? AND user_id = ?
    LIMIT 1
");
$st->execute([$rosterCode, $uid]);
$roster = $st->fetch();

if (!$roster) {
    echo json_encode([
        'success' => false,
        'message' => 'Ростер не знайдено або у вас немає доступу'
    ]);
    exit;
}

$st = $pdo->prepare("
    SELECT u.id
    FROM units u
    JOIN unit_access ua ON ua.unit_id = u.id
    WHERE u.id = ?
        AND ua.faction_root = ?
        AND (ua.faction IS NULL OR ua.faction = ?)
        AND u.is_active = 1
    LIMIT 1
");
$st->execute([
    $unitId,
    $roster['faction_root'],
    $roster['faction']
]);

$unit = $st->fetch();

if (!$unit) {
    echo json_encode([
        'success' => false,
        'message' => 'Цей юніт недоступний для цього ростера'
    ]);
    exit;
}

$st = $pdo->prepare("
    INSERT INTO roster_units
        (roster_code, unit_id, quantity)
    VALUES
        (?, ?, 1)
");

$st->execute([
    $rosterCode,
    $unitId
]);

echo json_encode([
    'success' => true
]);
exit;

```

roster_data.php

```
<?php
session_start();
require __DIR__ . '/conector.php';

header('Content-Type: application/json; charset=utf-8');

if (!isset($_SESSION['user']) || !is_array($_SESSION['user'])) {
    echo json_encode([
        'success' => false,
        'message' => 'Не авторизовано'
    ]);
    exit;
}

$uid = (int)($_SESSION['user']['id'] ?? 0);
$rosterCode = $_GET['id'] ?? '';

$stmt = $pdo->prepare("
    SELECT *
    FROM rosters
    WHERE roster_code = ? AND user_id = ?
    LIMIT 1
");
$stmt->execute([$rosterCode, $uid]);
$roster = $stmt->fetch();

if (!$roster) {
    echo json_encode([
        'success' => false,
        'message' => 'Ростер не знайдено'
    ]);
    exit;
}

$stmtRosterUnits = $pdo->prepare("
    SELECT ru.id, ru.quantity, u.id AS unit_id,
           u.name, u.points, u.category
    FROM roster_units ru
    JOIN units u ON u.id = ru.unit_id
    WHERE ru.roster_code = ?
    ORDER BY
        CASE u.category
            WHEN 'Epic Hero' THEN 1
            WHEN 'Character' THEN 2
            WHEN 'Battleline' THEN 3
            WHEN 'Infantry' THEN 4
            WHEN 'Vehicle' THEN 5
            ELSE 99
        END,
        u.name ASC
");
$stmtRosterUnits->execute([$rosterCode]);
$rosterUnits = $stmtRosterUnits->fetchAll();

$usedPoints = 0;
$unitCounts = [];

foreach ($rosterUnits as $ru) {
    $usedPoints += $ru['points'] * $ru['quantity'];

    if (!isset($unitCounts[$ru['name']])) {
```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		87

```

        $unitCounts[$ru['name']] = 0;
    }

    $unitCounts[$ru['name']]++;
}

echo json_encode([
    'success' => true,
    'usedPoints' => $usedPoints,
    'maxPoints' => (int)$roster['points'],
    'remainingPoints' => (int)$roster['points'] - $usedPoints
]);

```

datasheet.php

```

<?php
session_start();
require __DIR__ . '/conector.php';

header('Content-Type: application/json; charset=utf-8');

if (!isset($_SESSION['user']) || !is_array($_SESSION['user'])) {
    echo json_encode(['success' => false, 'message' => 'Не авторизовано']);
    exit;
}

$unitId = (int)($_GET['unit_id'] ?? 0);

$stmt = $pdo->prepare("
    SELECT *
    FROM units
    WHERE id = ?
    LIMIT 1
");
$stmt->execute([$unitId]);
$unit = $stmt->fetch();

if (!$unit) {
    echo json_encode(['success' => false, 'message' => 'Юніт не знайдено']);
    exit;
}

$stmtWeapons = $pdo->prepare("
    SELECT
        w.id,
        w.name,
        w.type,
        w.range_value,
        w.attacks,
        w.skill,
        w.strength,
        w.armor_penetration,
        w.damage
    FROM unit_weapons uw
    JOIN weapons w ON w.id = uw.weapon_id
    WHERE uw.unit_id = ?
    ORDER BY w.type DESC, w.name ASC
");
$stmtWeapons->execute([$unitId]);
$weapons = $stmtWeapons->fetchAll();

$stmtAbilities = $pdo->prepare("
    SELECT

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		88

```

        wal.weapon_id,
        wa.name,
        wa.description,
        wal.value,
        wal.target_keyword
    FROM weapon_ability_links wal
    JOIN weapon_abilities wa ON wa.id = wal.ability_id
    WHERE wal.weapon_id IN (
        SELECT weapon_id
        FROM unit_weapons
        WHERE unit_id = ?
    )
    ORDER BY wa.name ASC
");
$stAbilities->execute([$unitId]);
$abilityRows = $stAbilities->fetchAll();

$weaponAbilities = [];

foreach ($abilityRows as $row) {
    $weaponId = (int)$row['weapon_id'];

    if (!isset($weaponAbilities[$weaponId])) {
        $weaponAbilities[$weaponId] = [];
    }

    $label = $row['name'];

    if ($row['name'] === 'Anti' && !empty($row['target_keyword'])) {
        $label .= '-' . $row['target_keyword'];
    }

    if ($row['value'] !== null && $row['value'] !== '') {
        $label .= ' ' . $row['value'];
    }

    $weaponAbilities[$weaponId][] = [
        'label' => $label,
        'name' => $row['name'],
        'description' => $row['description'] ?? ''
    ];
}

echo json_encode([
    'success' => true,
    'unit' => $unit,
    'weapons' => $weapons,
    'weaponAbilities' => $weaponAbilities
]);

```

Додаток D

TourCraetor.php

```

<?php
session_start();

if (!isset($_SESSION['user'])) {

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		89

```

        header('Location: log.php?next=TourCraetor.php');
        exit;
    }

    require __DIR__ . '/conector.php';

    if ($_SERVER["REQUEST_METHOD"] === "POST") {
        $uid = (int)($_SESSION['user']['id'] ?? 0);
        $title = trim($_POST["title"] ?? "");
        $datetime = trim($_POST["datetime"] ?? "");
        $location = trim($_POST["location"] ?? "");
        $price = trim($_POST["price"] ?? "");
        $maxParticipants = (int)($_POST["max_participants"] ?? 16);
        $description = trim($_POST["description"] ?? "");
        $category = trim($_POST["category"] ?? "warhammer-40k");

        $startAt = null;

        if ($datetime !== "") {
            $ts = strtotime($datetime);
            if ($ts !== false) {
                $startAt = date('Y-m-d H:i:s', $ts);
            }
        }

        $pdo->beginTransaction();

        $stmt = $pdo->prepare("
            INSERT INTO tournaments
            (organizer_user_id, title, start_at, location, price,
            max_participants, description, category)
            VALUES
            (:uid, :title, :start_at, :location, :price,
            :max_participants, :description, :category)
        ");

        $stmt->execute([
            ':uid' => $uid,
            ':title' => $title,
            ':start_at' => $startAt,
            ':location' => $location,
            ':price' => number_format((float)$price, 2, '.', ''),
            ':max_participants' => $maxParticipants,
            ':description' => $description !== '' ? $description : null,
            ':category' => $category,
        ]);

        $pdo->commit();

        header("Location: tourlist.php?created=1");
        exit;
    }
}

```

tourPairingsSave.php

```

<?php
session_start();

if (!isset($_SESSION['user'])) {
    header('Location: log.php?next=tourlist.php');
    exit;
}

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докum.	Підпис	Дата		90

```

require __DIR__ . '/conector.php';

$userId = (int)($_SESSION['user']['id'] ?? 0);
$tournamentId = (int)($_POST['tournament_id'] ?? 0);

$stmt = $pdo->prepare("
    SELECT id, organizer_user_id
    FROM tournaments
    WHERE id = ?
    LIMIT 1
");
$stmt->execute([$tournamentId]);
$tournament = $stmt->fetch();

if (!$tournament || (int)$tournament['organizer_user_id'] !== $userId) {
    header('Location: index.php');
    exit;
}

$stmt = $pdo->prepare("
    SELECT user_id
    FROM tournament_participants
    WHERE tournament_id = ?
    ORDER BY joined_at ASC
");
$stmt->execute([$tournamentId]);
$participantIds = array_map('intval', $stmt->fetchAll(PDO::FETCH_COLUMN));

$tableNumbers = $_POST['table_number'] ?? [];
$player1Ids = $_POST['player1_user_id'] ?? [];
$player2Ids = $_POST['player2_user_id'] ?? [];

$participantMap = [];
foreach ($participantIds as $pid) {
    $participantMap[$pid] = true;
}

$usedPlayers = [];
$matches = [];
$proxyCount = 0;
$maxProxyCount = count($participantIds) % 2 === 1 ? 1 : 0;

for ($i = 0; $i < count($player1Ids); $i++) {
    $tableNumber = (int)$tableNumbers[$i];
    $player1 = (int)$player1Ids[$i];
    $player2 = (int)$player2Ids[$i];

    if ($tableNumber <= 0 || $player1 <= 0) {
        header('Location: tourDetails.php?id=' . $tournamentId .
            '&pairing_error=1');
        exit;
    }

    if (!isset($participantMap[$player1]) || isset($usedPlayers[$player1])) {
        header('Location: tourDetails.php?id=' . $tournamentId .
            '&pairing_error=1');
        exit;
    }

    $usedPlayers[$player1] = true;

    $isProxy = 0;
    $player2Value = null;

```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		91

```

if ($player2 === 0) {
    $isProxy = 1;
    $proxyCount++;
} else {
    if (!isset($participantMap[$player2])
        || $player1 === $player2
        || isset($usedPlayers[$player2])) {
        header('Location: tourDetails.php?id=' . $tournamentId .
'&pairing_error=1');
        exit;
    }

    $usedPlayers[$player2] = true;
    $player2Value = $player2;
}

$matches[] = [
    'table_number' => $tableNumber,
    'player1_user_id' => $player1,
    'player2_user_id' => $player2Value,
    'is_proxy' => $isProxy
];
}

if ($proxyCount > $maxProxyCount) {
    header('Location: tourDetails.php?id=' . $tournamentId . '&pairing_error=1');
    exit;
}

try {
    $pdo->beginTransaction();

    $stmt = $pdo->prepare("
        SELECT COALESCE(MAX(round_number), 0) + 1
        FROM tournament_rounds
        WHERE tournament_id = ?
        FOR UPDATE
    ");
    $stmt->execute([$tournamentId]);
    $roundNumber = (int)$stmt->fetchColumn();

    $stmt = $pdo->prepare("
        INSERT INTO tournament_rounds
        (tournament_id, round_number, status)
        VALUES
        (?, ?, 'active')
    ");
    $stmt->execute([$tournamentId, $roundNumber]);

    $roundId = (int)$pdo->lastInsertId();

    $stmt = $pdo->prepare("
        INSERT INTO tournament_matches
        (tournament_id, round_id, table_number,
        player1_user_id, player2_user_id, is_proxy, status)
        VALUES
        (:tournament_id, :round_id, :table_number,
        :player1_user_id, :player2_user_id, :is_proxy, 'active')
    ");

    foreach ($matches as $m) {
        $stmt->execute([

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		92

```

        ':tournament_id' => $tournamentId,
        ':round_id' => $roundId,
        ':table_number' => $m['table_number'],
        ':player1_user_id' => $m['player1_user_id'],
        ':player2_user_id' => $m['player2_user_id'],
        ':is_proxy' => $m['is_proxy']
    ));
}

$pdo->commit();

header('Location: tourDetails.php?id=' . $tournamentId .
'&pairings_created=1');
exit;
} catch (Throwable $e) {
    if ($pdo->inTransaction()) {
        $pdo->rollBack();
    }

    header('Location: tourDetails.php?id=' . $tournamentId . '&pairing_error=1');
    exit;
}

```

tourMatchResult.php

```

<?php
session_start();

if (!isset($_SESSION['user'])) {
    header('Location: log.php?next=tourlist.php');
    exit;
}

require __DIR__ . '/conector.php';

$userId = (int)($_SESSION['user']['id'] ?? 0);
$matchId = (int)($_POST['match_id'] ?? 0);

function result_parts($points) {
    $points = (int)$points;

    if ($points === 2) {
        return [1, 0, 0];
    }

    if ($points === 1) {
        return [0, 1, 0];
    }

    return [0, 0, 1];
}

$stmt = $pdo->prepare("
    SELECT
        m.id,
        m.tournament_id,
        m.round_id,
        m.player1_user_id,
        m.player2_user_id,
        m.is_proxy,
        t.organizer_user_id
    FROM tournament_matches m
    JOIN tournaments t ON t.id = m.tournament_id

```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		93

```

        WHERE m.id = ?
        LIMIT 1
    ");
    $stmt->execute([$matchId]);
    $match = $stmt->fetch();

    if (!$match) {
        header('Location: tourlist.php');
        exit;
    }

    $tournamentId = (int)$match['tournament_id'];

    if ((int)$match['organizer_user_id'] !== $userId) {
        header('Location: index.php');
        exit;
    }

    $score1 = (int)($_POST['player1_score'] ?? 0);
    $score2 = (int)($_POST['player2_score'] ?? 0);

    $player1Id = (int)$match['player1_user_id'];
    $player2Id = $match['player2_user_id'] !== null
        ? (int)$match['player2_user_id']
        : null;

    $isProxy = (int)$match['is_proxy'] === 1;

    $player1TournamentPoints = 0;
    $player2TournamentPoints = 0;
    $winnerUserId = null;

    $scoreDiff = abs($score1 - $score2);
    $drawMargin = 10;

    if ($scoreDiff <= $drawMargin) {
        $player1TournamentPoints = 1;
        $player2TournamentPoints = 1;
        $winnerUserId = null;
    } elseif ($score1 > $score2) {
        $player1TournamentPoints = 2;
        $player2TournamentPoints = 0;
        $winnerUserId = $player1Id;
    } else {
        $player1TournamentPoints = 0;
        $player2TournamentPoints = 2;
        $winnerUserId = $isProxy ? null : $player2Id;
    }

    $stmt = $pdo->prepare("
        UPDATE tournament_matches
        SET
            player1_score = :player1_score,
            player2_score = :player2_score,
            winner_user_id = :winner_user_id,
            player1_tournament_points = :player1_tournament_points,
            player2_tournament_points = :player2_tournament_points,
            status = 'finished'
        WHERE id = :id
    ");

    $stmt->execute([
        ':player1_score' => $score1,

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		94

```

        ':player2_score' => $score2,
        ':winner_user_id' => $winnerUserId,
        ':player1_tournament_points' => $player1TournamentPoints,
        ':player2_tournament_points' => $player2TournamentPoints,
        ':id' => $matchId
    ]);

```

Додаток Е

adminUnits.php

```

<?php
session_start();

if (!isset($_SESSION['user']) || !is_array($_SESSION['user'])) {
    header('Location: log.php?next=adminUnits.php');
    exit;
}

require __DIR__ . '/conector.php';

$userId = (int)($_SESSION['user']['id'] ?? 0);
$userRole = $_SESSION['user']['role'] ?? 'user';

if ($userId <= 0 || $userRole !== 'admin') {
    header('Location: home.php');
    exit;
}

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $action = $_POST['action'] ?? '';

    if ($action === 'save') {
        $id = (int)($_POST['id'] ?? 0);
        $name = trim($_POST['name'] ?? '');
        $points = (int)($_POST['points'] ?? 0);
        $category = trim($_POST['category_mode'] ?? '');

        $movement = trim($_POST['movement'] ?? '');
        $toughness = trim($_POST['toughness'] ?? '');
        $saveValue = trim($_POST['save_value'] ?? '');
        $wounds = trim($_POST['wounds'] ?? '');
        $leadership = trim($_POST['leadership'] ?? '');
        $objectiveControl = trim($_POST['objective_control'] ?? '');
        $invulnerableSave = trim($_POST['invulnerable_save'] ?? '');

        if ($id > 0) {
            $stmt = $pdo->prepare("
                UPDATE units
                SET
                    name = :name,
                    points = :points,
                    category = :category,
                    movement = :movement,
                    toughness = :toughness,
                    save_value = :save_value,
                    wounds = :wounds,
                    leadership = :leadership,
                    objective_control = :objective_control,
                    invulnerable_save = :invulnerable_save
                WHERE id = :id
            ");

```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		95

```

$stmt->execute([
    ':name' => $name,
    ':points' => $points,
    ':category' => $category,
    ':movement' => $movement,
    ':toughness' => $toughness,
    ':save_value' => $saveValue,
    ':wounds' => $wounds,
    ':leadership' => $leadership,
    ':objective_control' => $objectiveControl,
    ':invulnerable_save' => $invulnerableSave,
    ':id' => $id
]);
} else {
$stmt = $pdo->prepare("
    INSERT INTO units
        (name, points, category, is_active, movement, toughness,
         save_value, wounds, leadership, objective_control, invulnerable_save)
    VALUES
        (:name, :points, :category, 1, :movement, :toughness,
         :save_value, :wounds, :leadership, :objective_control,
:invulnerable_save)
    ");

$stmt->execute([
    ':name' => $name,
    ':points' => $points,
    ':category' => $category,
    ':movement' => $movement,
    ':toughness' => $toughness,
    ':save_value' => $saveValue,
    ':wounds' => $wounds,
    ':leadership' => $leadership,
    ':objective_control' => $objectiveControl,
    ':invulnerable_save' => $invulnerableSave
]);
}
}
}
}

```

adminWeapons.php

```

<?php
session_start();

if (!isset($_SESSION['user']) || !is_array($_SESSION['user'])) {
    header('Location: log.php?next=adminWeapons.php');
    exit;
}

require __DIR__ . '/conector.php';

$userId = (int)($_SESSION['user']['id'] ?? 0);
$userRole = $_SESSION['user']['role'] ?? 'user';

if ($userId <= 0 || $userRole !== 'admin') {
    header('Location: home.php');
    exit;
}

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $action = $_POST['action'] ?? '';

```

		Кравцов І.М			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		96

```

if ($action === 'save_weapon') {
    $id = (int)($_POST['id'] ?? 0);
    $weaponName = trim($_POST['weapon_name'] ?? '');
    $weaponType = trim($_POST['weapon_type'] ?? 'Ranged');
    $rangeValue = trim($_POST['range_value'] ?? '');
    $attacks = trim($_POST['attacks'] ?? '');
    $skill = trim($_POST['skill'] ?? '');
    $strength = trim($_POST['strength'] ?? '');
    $armorPenetration = trim($_POST['armor_penetration'] ?? '');
    $damage = trim($_POST['damage'] ?? '');

    $selectedAbilityIds = $_POST['ability_ids'] ?? [];
    $abilityValues = $_POST['ability_values'] ?? [];
    $abilityTargets = $_POST['ability_targets'] ?? [];

    $cleanAbilities = [];

    if (is_array($selectedAbilityIds)) {
        foreach ($selectedAbilityIds as $abilityIdRaw) {
            $abilityId = (int)$abilityIdRaw;

            if ($abilityId <= 0) {
                continue;
            }

            $value = isset($abilityValues[$abilityId])
                ? trim($abilityValues[$abilityId])
                : '';

            $target = isset($abilityTargets[$abilityId])
                ? trim($abilityTargets[$abilityId])
                : '';

            $cleanAbilities[$abilityId] = [
                'value' => $value !== '' ? $value : null,
                'target_keyword' => $target !== '' ? $target : null
            ];
        }
    }

    $pdo->beginTransaction();

    if ($id > 0) {
        $stmt = $pdo->prepare("
            UPDATE weapons
            SET
                name = :name,
                type = :type,
                range_value = :range_value,
                attacks = :attacks,
                skill = :skill,
                strength = :strength,
                armor_penetration = :armor_penetration,
                damage = :damage
            WHERE id = :id
        ");

        $stmt->execute([
            ':name' => $weaponName,
            ':type' => $weaponType,
            ':range_value' => $rangeValue !== '' ? $rangeValue : null,
            ':attacks' => $attacks !== '' ? $attacks : null,

```

		Кравцов І.М.			ІПЗ.КР.Б – 121 – 26 – ПЗ	Арк.
		Вакалюк Т.А..				
Змн.	Арк.	№ докум.	Підпис	Дата		97

```

        ':skill' => $skill !== '' ? $skill : null,
        ':strength' => $strength !== '' ? $strength : null,
        ':armor_penetration' => $armorPenetration !== '' ? $armorPenetration :
null,
        ':damage' => $damage !== '' ? $damage : null,
        ':id' => $id
    ]);

    $weaponId = $id;

    $stmt = $pdo->prepare("
        DELETE FROM weapon_ability_links
        WHERE weapon_id = ?
    ");
    $stmt->execute([$weaponId]);
} else {
    $stmt = $pdo->prepare("
        INSERT INTO weapons
        (name, type, range_value, attacks, skill,
        strength, armor_penetration, damage)
        VALUES
        (:name, :type, :range_value, :attacks, :skill,
        :strength, :armor_penetration, :damage)
    ");

    $stmt->execute([
        ':name' => $weaponName,
        ':type' => $weaponType,
        ':range_value' => $rangeValue !== '' ? $rangeValue : null,
        ':attacks' => $attacks !== '' ? $attacks : null,
        ':skill' => $skill !== '' ? $skill : null,
        ':strength' => $strength !== '' ? $strength : null,
        ':armor_penetration' => $armorPenetration !== '' ? $armorPenetration :
null,
        ':damage' => $damage !== '' ? $damage : null
    ]);

    $weaponId = (int)$pdo->lastInsertId();
}

$stmt = $pdo->prepare("
    INSERT INTO weapon_ability_links
    (weapon_id, ability_id, value, target_keyword)
    VALUES
    (?, ?, ?, ?)
");

foreach ($cleanAbilities as $abilityId => $data) {
    $stmt->execute([
        $weaponId,
        $abilityId,
        $data['value'],
        $data['target_keyword']
    ]);
}

$pdo->commit();
}
}

```