

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ ЕНЕРГЕТИКИ
кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

“До захисту допущено”
Завідувач кафедри ЦТЕ
_____ Наталія АУШЕВА

“ ” _____ 2024 р.

Дипломна робота
на здобуття ступеня бакалавра
за освітньо-професійною програмою
“Цифрові технології в енергетиці”
зі спеціальності 122 “Комп’ютерні науки”

на тему: **Веб-тренажер для багатокористувацької онлайн-гри**
Mobile Legends

Виконав (-ла):
студент (-ка) IV курсу, групи ТР-03

_____ Кордяк Ростислав Михайлович
(прізвище, ім’я, по батькові)

_____ (підпис)

Керівник: *доцент каф. цифрових технологій в енергетиці*
доцент, к.т.н., Кублій Лариса Іванівна
(посада, науковий ступінь, вчене звання, прізвище, ім’я, по батькові)

_____ (підпис)

Рецензент: _____

_____ (посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім’я, по батькові)

_____ (підпис)

Нормоконтролер: _____

_____ (підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших авторів
без відповідних посилань.

Студент (-ка) _____
(підпис)

Київ – 2024

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ АТОМНОЇ ТА ТЕПЛОВОЇ
ЕНЕРГЕТИКИ**

Кафедра ЦИФРОВИХ ТЕХНОЛОГІЙ В ЕНЕРГЕТИЦІ

Рівень вищої освіти – перший (бакалаврський)

спеціальність 122 “Комп’ютерні науки”

Освітньо-професійна “Цифрові технології в енергетиці”
програма _____

ЗАТВЕРДЖУЮ

Завідувач кафедри ЦТЕ

_____ Наталія АУШЕВА

«__»_____2024 р.

**ЗАВДАННЯ
на дипломну роботу студенту**

КОРДЯКУ Ростиславу Михайловичу

(прізвище, ім’я, по батькові)

1. Тема роботи **Веб-тренажер для багатокористувацької
онлайн-гри Mobile Legends**

керівник роботи **Кублій Лариса Іванівна, к.т.н., доцент**

(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «__»_____ 202__ р.
№_____

2. Термін подання роботи студентом **07.06.2024 року**

3. Вихідні дані до роботи: мова розмітки HTML, мова таблиць стилів CSS, мова
програмування JavaScript, бібліотека React

4. Перелік питань, які потрібно розробити: _____

1) провести аналіз наявних аналогів

2) провести аналіз веб-систем з посібниками для відеоігор

3) обґрунтувати вибір інструментів, технологій та алгоритмів для реалізації програмного забезпечення

4) розробити архітектуру програмного забезпечення, баз даних і сховища файлів

5) створити прикладний програмний веб-інтерфейс

6) подати процес застосування веб-інтерфейсу

5. Орієнтований перелік ілюстративного матеріалу: діаграми, які демонструють роботу алгоритмів веб-інтерфейсу, архітектуру системи, взаємодію користувачів із системою, класи програмного забезпечення; приклади роботи з програмним продуктом.

6. Дата видачі завдання «19» вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1.	Затвердження теми роботи		
2.	Вивчення та аналіз задачі	17-20.04.24	
3.	Розробка архітектури і загальної структури системи	21-25.04.24	
4.	Розробка частин окремих підсистем	25.04-02.05.24	
5.	Програмна реалізація системи	03-07.05.24	
6.	Оформлення пояснювальної записки	08-12.05.24	
7.	Захист програмного продукту	13-17.05.23	
8.	Передзахист	03-06.06.23	
9.	Подання готової роботи на кафедру	07.06.2024	
10.	Захист	17-21.06.2024	

Студент

(підпис)

Ростислав КОРДЯК

(ім'я, ПРИЗВИЩЕ)

Керівник роботи

(підпис)

Лариса КУБЛІЙ

(ім'я, ПРИЗВИЩЕ)

АНОТАЦІЯ

Дипломна робота виконана на 54 сторінках, містить 26 ілюстрацій, 26 джерел в переліку посилань, 3 додатки.

Мета роботи – створення веб-застосунку з посібниками для гри Mobile Legends: Bang Bang.

Методи та засоби: мова розмітки HTML, мова таблиць стилів CSS, мова програмування JavaScript, бібліотека React для JavaScript, модуль React-i18next для забезпечення локалізацій, засіб Vite для спрощення розробки веб-застосунку.

Результат – розроблений веб-застосунок, який містить посібники для багатокористувацької онлайн-гри Mobile Legends.

ANNOTATION

The work consists of 54 pages, includes 26 illustrations, references to 26 sources, 3 appendices.

The purpose of the work is to create a web application with guides for the Mobile Legends: Bang Bang game.

Methods and tools: HTML markup language, CSS stylesheet language, JavaScript programming language, React library for JavaScript, React-i18next module for localization, Vite tool for simplifying web application development.

Result: developed a web application that contains guides for the multiplayer online game Mobile Legends.

ЗМІСТ

ВСТУП	8
1 ОГЛЯД СУЧАСНИХ ВЕБ-ТРЕНАЖЕРІВ	10
1.1 Аналіз веб-тренажерів для Mobile Legends	10
1.1.1 Офіційний веб-сайт Mobile Legends	10
1.1.2 Веб-архів Mobile Legends: Bang Bang Wiki	13
1.1.3 Веб-сайт ML Build	14
1.1.4 Веб-тренажер від Alexandre Games	15
1.2 Ресурси для інших відеоігор	16
1.2.1 Веб-застосунок OP.GG	17
1.2.2 Веб-сайт KeqingMains	17
1.3 Задачі, які розв’язуються програмним забезпеченням	21
2 ЗАСОБИ РОЗРОБКИ	22
2.1 Основні засоби розробки	22
2.1.1 Мова розмітки HTML	22
2.1.2 Мова стилів CSS	23
2.1.3 Мова програмування JavaScript	23
2.2 Допоміжні засоби розробки	24
2.2.1 Бібліотека React	24
2.2.2 Бібліотека React Router	25
2.2.3 Система Vite	26
2.2.4 Бібліотека React-i18next	26
2.2.5 Фреймворк Angular	27
2.2.6 Фреймворк Vue	29
2.2.7 Фреймворк Bootstrap	30
2.2.8 Фреймворк Tailwind	30
2.3 Засоби проектування і дизайну	31
2.3.1 Інструмент Figma	31
2.3.2 Редактор коду Sublime Text	32

	7
2.3.3 Редактор коду Microsoft Visual Studio Code	32
2.3.4 Система контролю версій Git	33
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ТРЕНАЖЕРА	34
3.1 Структура веб-застосунку	34
3.1.1 Основні файли	34
3.1.2 Компоненти	34
3.2 Функції програмної системи	38
3.3 Масив даних	41
3.4 Взаємодія програмних модулів	43
4 РОБОТА КОРИСТУВАЧА З ВЕБ-ТРЕНАЖЕРОМ	45
4.1 Зовнішній вигляд системи	45
4.2 Подальше вдосконалення на розширення	51
4.3 Діаграма прецедентів	52
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	55
ДОДАТОК А	58
ДОДАТОК Б	71
ДОДАТОК В	75

ВСТУП

У сучасному світі відеоігор інформація є ключовим елементом для успіху гравців. Популярна гра в жанрі MOBA – Mobile Legends: Bang Bang – не виняток. Розуміння особливостей кожного героя гри, його навичок, слабких і сильних сторін стає вирішальним фактором у досягненні перемоги. Однак часто інформація про героїв розпорошена по різних джерелах, що створює певні труднощі для гравців, які прагнуть швидко й ефективно підготуватися до гри

Саме для розв'язання цієї проблеми пропонується створення спеціалізованого веб-застосунку, який стане ресурсом для збирання та систематизації корисної інформації про кожного героя гри. Такий сайт дасть можливість гравцям легко ознайомлюватися з усіма необхідними нюансами ігрового процесу, що сприятиме підвищенню їхнього рівня гри і забезпеченню максимальної задоволеності від ігрового досвіду.

Актуальність теми полягає в тому, що наразі не існує ресурсу, який би відповідав сучасним стандартам веб-ресурсів з посібниками.

Мета роботи – розробка веб-застосунку для багатокористувацької гри Mobile Legends.

Завдання:

- аналіз методів і засобів створення веб-застосунків;
- аналіз існуючих програмних засобів для реалізації системи “Веб-тренажер для багатокористувацької онлайн-гри Mobile Legends”;
- розробка програмного забезпечення Веб-тренажер для багатокористувацької онлайн-гри Mobile Legends;
- тестування розробленого програмного забезпечення.

Результати дипломної роботи апробовано на XXI Міжнародній науково-практичній конференції молодих вчених і студентів “Сучасні проблеми наукового забезпечення енергетики”, м. Київ, 23-26 квітня 2024 року [1] (додаток Б).

Дипломна робота складається із 54 сторінок (без додатків і списку джерел), 26 рисунків, 3 додатків. Повний обсяг дипломної записки – 80 сторінок.

У першому розділі пояснювальної записки визначено задачу, яку виконує програмне забезпечення, розглянуто існуючі системи й проаналізовано їх.

У другому розділі визначено мову програмування, бібліотеки й середовище розробки для веб-тренажера. Обґрунтовано використання обраних засобів.

У третьому розділі описано програмну реалізацію проєкту.

У четвертому розділі продемонстровано приклад взаємодії користувача з системою.

Додаток А містить 13 сторінок з кодом основних модулів системи.

Додаток Б складається з 4 сторінок і містить скан-копії тез доповіді, представленої на конференції “Сучасні проблеми наукового забезпечення енергетики”.

Додаток В складається з 6 сторінок і містить фрагменти коду компонентів і функцій.

1 ОГЛЯД СУЧАСНИХ ВЕБ-ТРЕНАЖЕРІВ

Наявні аналоги, які надають інформацію про героїв гри Mobile Legends, не відповідають повною мірою потребам гравців. Вони часто є фрагментарними, незручними у використанні або не забезпечують достатньої глибини інформації. Це створює необхідність у розробці нового рішення, яке буде більш ефективним та корисним для користувачів.

1.1 Аналіз веб-тренажерів для Mobile Legends

Існує значна кількість веб-тренажерів для Mobile Legends, однак більшість з них мають низьку якість. Це створює суттєві проблеми для користувачів, які прагнуть покращити свої навички або досягти високих результатів у грі. Недосконалість таких тренажерів може включати поганий користувацький інтерфейс, недостатню кількість функцій або неточність у моделюванні ігрових ситуацій.

1.1.1 Офіційний веб-сайт Mobile Legends

Найпопулярніший ресурс з рекомендаціями щодо ігрового процесу – m.mobilelegends.com [2], містить вичерпну інформацію про гру. Тут можна знайти інформацію про останні оновлення гри, зміни у балансі, нових героїв та предмети.

Також є змога стежити за актуальними новинами про гру, анонсами подій та турнірів, а також публікуються інші важливі повідомлення від розробників (рисунок 1.1)

Однак, попри свою популярність, цей сайт має багато недоліків, що обмежують його корисність для гравців. Хоч описи героїв та їхніх здібностей присутні на сайті, описів функцій ролей та оптимальних стратегій використання героїв не вистачає. Така ж ситуація з ігровими предметами – є тільки поверхневі

описи предметів та їхні характеристики, але бракує рекомендацій щодо використання цих предметів у різних ситуаціях.

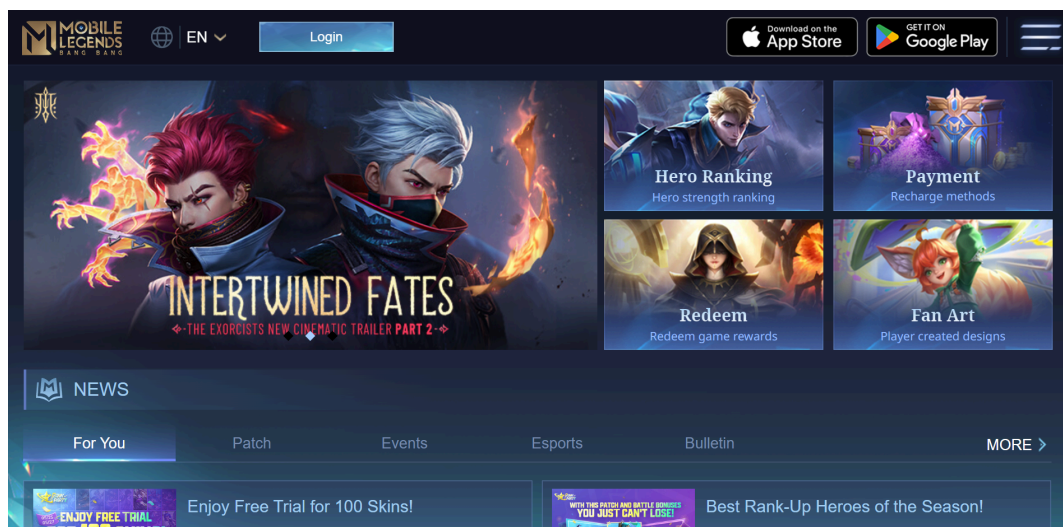


Рисунок 1.1 – Головна сторінка m.mobilelegends.com

Оскільки розглядається тільки частина з інструкціями для героїв, варто розглянути її детальніше. На головній сторінці веб-сайту пропонується переглянути здібності нещодавно випущених героїв, а також короткий посібник для цих героїв. Хоч такий дизайн здається зручним і зрозумілим, та для гри, яка налічує понад 100 унікальних персонажів, це меню не дає звичайному користувачу повноту вибору (рисунок 1.2).

Сторінка для певного героя представлена у вигляді інтерактивного профілю з пунктами навичок, підготовки, героями, що протидіють даному і переліком відео-посібників. Для звичайного гравця, який не знає всіх нюансів гри, такий простий дизайн може бути привабливим, але відсутність пояснювальної інформації завадить користувачу зрозуміти те, що йому пропонують.

Крім того, варто зазначити, що інтерактивний профіль героя містить лише основну інформацію про навички та стратегії. Для новачків, які тільки починають ознайомлюватися з грою, такий обмежений обсяг даних може бути недостатнім для повного розуміння потенціалу героя та його ролі у команді.

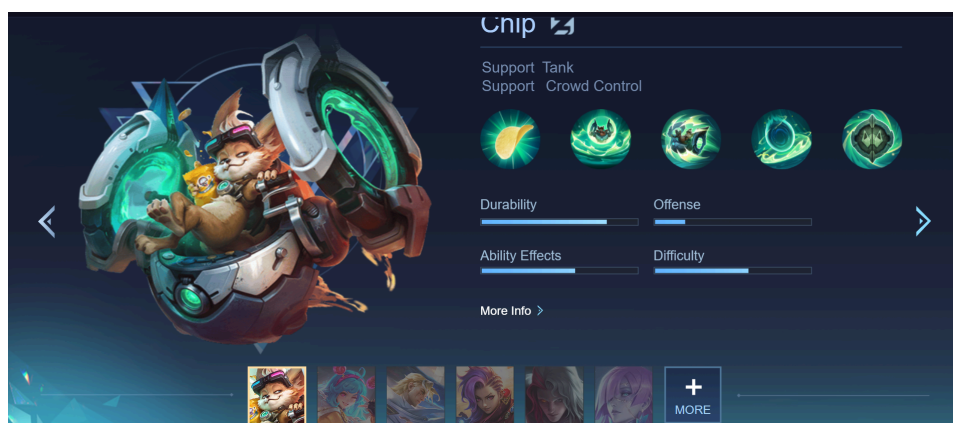


Рисунок 1.2 – Меню нових героїв

На сторінці підготовки розміщено одну заготовку набору предметів, емблем та допоміжної здібності. Ці заготовки взято у професіональних гравців, які мають набагато вищий рівень розуміння гри, ніж середньостатистичний гравець. Використання запропонованих заготовок може стати проблемою для менш досвідчених гравців. Не враховуючи їхній власний стиль гри та індивідуальні потреби команди, вони можуть не досягти бажаних результатів (рисунок 1.3).

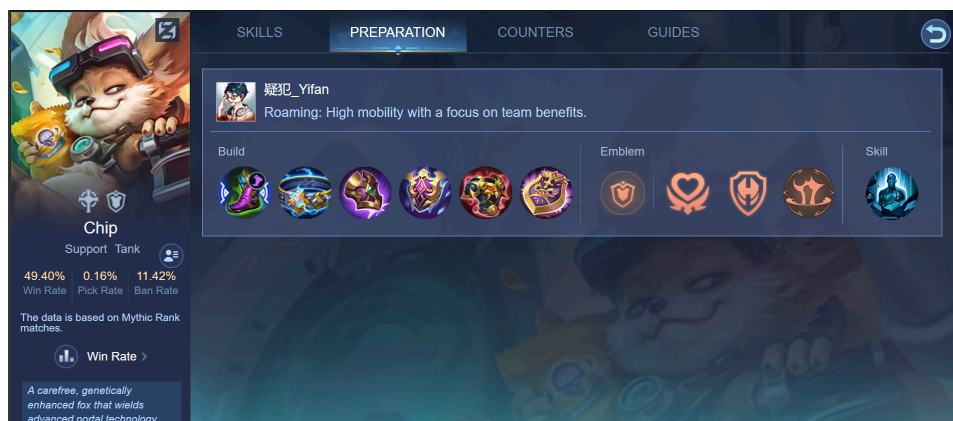


Рисунок 1.3 – Сторінка підготовки героя

Такі заготовки можуть вводити в оману, адже вони не враховують різноманітні фактори, що виникають під час реальних ігор, такі як адаптація до противника та синергія з товаришами по команді.

1.1.2 Веб-архів Mobile Legends: Bang Bang Wiki

Сайт Mobile Legends: Bang Bang Wiki – це веб-сайт, створений спільнотою Mobile Legends: Bang Bang на хостингу Fandom [3]. На цьому сайті є інформація практично про все, що стосується гри, наприклад: костюми, події, типи шкоди. Незважаючи на великий обсяг інформації різного роду, даний веб-ресурс також містить посібники з використання героїв.

Хоч на перший погляд може здаватися, що цей веб-сайт містить унікальну інформацію про використання героїв, вся ця інформація вже є в відповідному меню всередині самої гри. Сайт Mobile Legends: Bang Bang Wiki – це веб-архів, що збирає інформацію про гру з різних джерел, і в якому не публікуються унікальні посібники для героїв (рисунок 1.4).

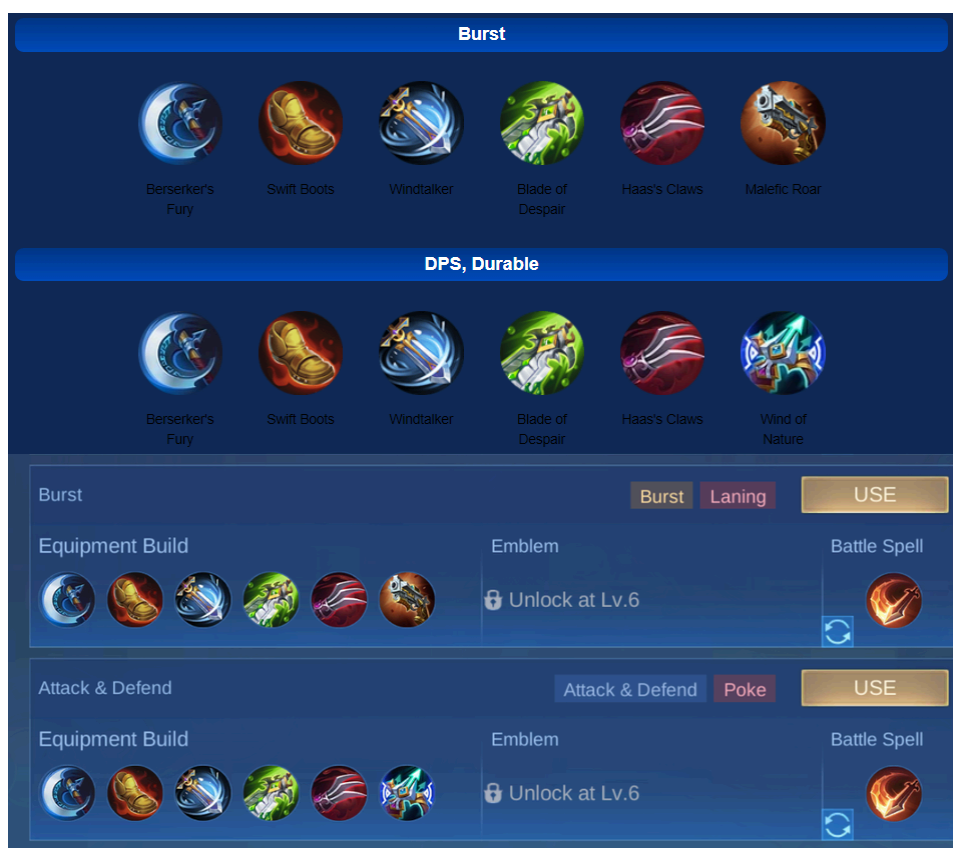


Рисунок 1.4 – Порівняння інформації MLBB Wiki з інформацією у грі

Видно, що набір предметів на верхньому зображенні такий самий, як знизу.

1.1.3 Веб-сайт ML Build

Найбільший недолік цього ресурсу [4] – обраний стиль списку. Герої подані у вигляді вертикального списку, де кожен герой займає окремий рядок. Враховуючи відсутність фільтрів та функції пошуку, знайти потрібного героя – проблематично (рисунок 2.5).

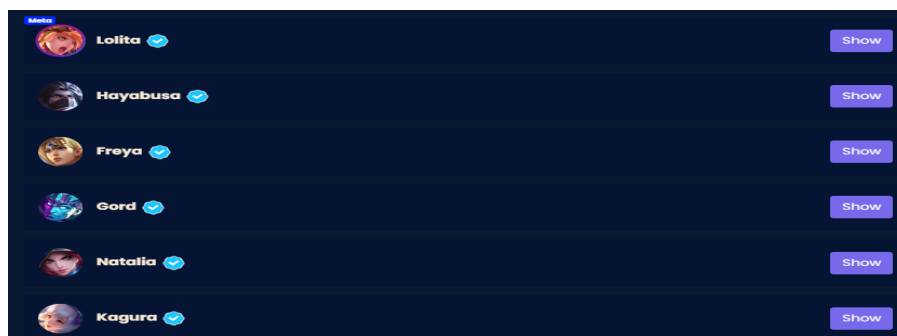


Рисунок 1.5 – Список героїв ML Builds

Самі посібники виконані в мінімалістичному стилі. Вони не дають повної інформації про предмети, а тільки перелічують популярні набори. Також варто врахувати помилки з відображенням піктограм (рисунок 2.5)

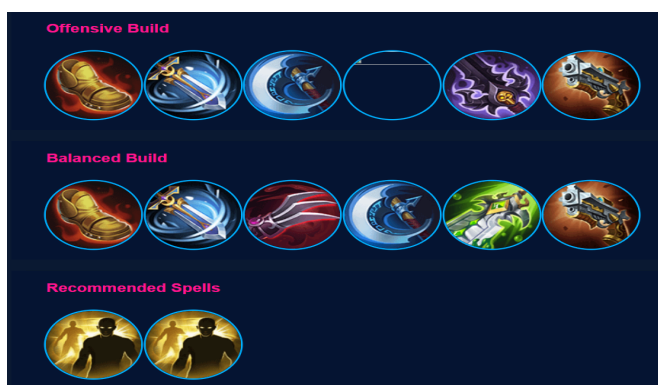


Рисунок 1.6 – Один із посібників ML Builds

Одне із зображень не завантажилось, а серед рекомендованих навичок одна і та сама піктограма повторюється двічі.

1.1.4 Веб-тренажер від Alexandre Games

Веб-сайт alexandregames.com [5] пропонує декілька розгорнутих посібників для Mobile Legends.

Цей блог – це місце, де любителі мобільних ігор можуть знайти цікаву інформацію про нові та популярні ігри для Android, iOS та ПК. Автор, Александр Домінгос, разом зі своєю дружиною, присвячують свій час та енергію, щоб надавати гравцям свіжі новини, поради та огляди гри, щоб зробити їхні ігрові досвіди ще цікавішими та насиченішими.

Повнота посібників насправді хороша. Видно, що автор докладает чимало зусиль для того, щоб створити їх. Він розбирається у грі й надає дійсно хороші поради щодо ігрового процесу.

Але дизайн цього блогу не відповідає сучасним стандартам веб-сайтів. В меню героїв є тільки декілька героїв, відсутні будь-які фільтри й пошук, а також немає короткого опису героїв (рисунок 1.7)

MOBILE LEGENDS BANG BANG HEROES GUIDE

Explore our complete Mobile Legends Bang Bang hero guide and dominate the battlefield with strategies and tips for each character!

Choose Your Mobile Legends Characters:



Рисунок 1.7 – Меню героїв на Alexandre Games

Також самі посібники складаються із суцільного тексту, який не дуже добре читається. Такий дизайн важко сприймається користувачем (рисунок 1.8).

Tactics and Strategy

Early Game

During the early game, Aulus is a versatile and potent hero capable of holding his own against most opponents. Focus on the following:

- **Lane Control:** Utilize your skills to dominate your lane, ensuring you get as many last hits as possible to gain gold and experience.
- **Harassment:** Use "Aulus, Charge!" to close the gap and harass enemy heroes, forcing them to play defensively.
- **Survivability:** Take advantage of the damage reduction from "Aulus, Charge!" to withstand enemy attacks and stay in the lane longer.

Mid Game

In the mid game, Aulus's role shifts slightly as you start to focus more on objectives and team fights:

- **Pushing Turrets:** Your enhanced Basic Attacks from "The Power of Axe" make you a formidable pusher. Focus on taking down enemy turrets to apply pressure on the enemy team.
- **Positioning:** Good positioning is key. Stay at the frontlines during team fights to absorb damage and disrupt the enemy team.
- **Turning Team Fights:** With proper timing and positioning, you can turn the tide of team fights. Use "Undying Fury" to deal significant damage and disrupt enemy formations.

Late Game

Рисунок 1.8 – Посібник від Alexandre Games

Загалом, якби автор зайнявся дизайном цього веб-сайту, то він би став конкурентом навіть офіційного веб-сайту Mobile Legends.

1.2 Ресурси для інших відеоігор

Існує ряд веб-тренажерів для інших ігор, які виділяються високою якістю та корисністю для гравців. Однією з ключових переваг таких тренажерів є добре продуманий користувацький інтерфейс, який дає можливість зручно та ефективно використовувати всі їхні можливості. Крім того, вони можуть містити широкий спектр функцій, які дають можливість гравцям сформувати унікальний стиль гри. Додатково, такі тренажери можуть відзначатися високою точністю у моделюванні

ігрових ситуацій, що дає можливість гравцям отримувати реалістичний досвід та ефективно розвивати свої навички.

1.2.1 Веб-застосунок OP.GG

Веб-сайт OP.GG [6] збирає інформацію про найстабільніші і найпопулярніші способи гри в League of Legends. Ця платформа має сучасний вигляд і зручний зрозумілий інтерфейс, а також дає користувачам максимально корисну інформацію. Багато гравців з усього світу діляться своїми даними, щоб сформувати якомога більшу вибірку, яка в подальшому обробляється програмою для визначення найефективнішого стилю гри за того чи іншого героя.

Однією з основних переваг цього веб-сайту є зручне меню, яке містить інформацію про всіх доступних героїв. Це меню дає можливість користувачам швидко і легко знаходити потрібного героя без зайвих зусиль. Завдяки чіткій структурі та інтуїтивному дизайну, гравці можуть одразу бачити повний перелік героїв і переходити до детальнішої інформації. (рисунки 1.9).

Ще однією суттєвою перевагою є різноманітність порад щодо вибору предметів. Замість того, щоб пропонувати один універсальний набір предметів для всіх випадків, веб-сайт op.gg надає кілька варіантів комбінацій. Такий підхід дає можливість гравцям адаптувати свої стратегії відповідно до конкретних умов гри та стилю гри кожного користувача. Завдяки цьому гравці можуть краще налаштувати ігровий процес під свої індивідуальні потреби та переваги, що сприяє більшій гнучкості і підвищенню ефективності в грі. Це також допомагає гравцям експериментувати з різними стратегіями та знаходити оптимальні рішення для різних ситуацій, що робить ігровий процес більш захоплюючим і динамічним (рисунки 1.10)

1.2.2 Веб-сайт KeqingMains

Ще один хороший ресурс уже для іншої гри – keqingmains.com [7] для Genshin Impact. Веб-застосунок KeqingMains має на меті публікувати детальні дослідження, пов'язані з персонажами відеогри Genshin Impact.

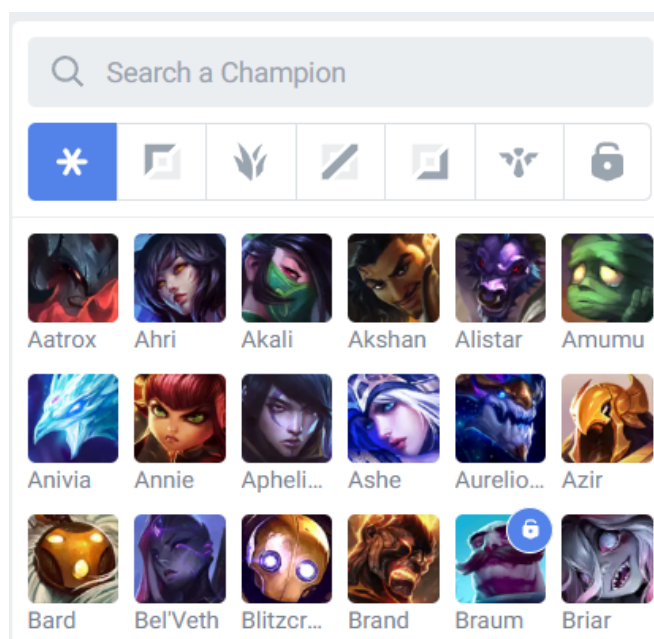


Рисунок 1.9 – Меню героїв на сайті OP.GG

Bel'Veth Builds						
	>		>		21.77%	61.66%
					2,799 Games	
	>		>		11.28%	60.37%
					1,451 Games	
	>		>		8.63%	61.98%
					1,110 Games	
	>		>		5.83%	64.13%
					750 Games	
	>		>		4.53%	61.41%
					583 Games	

Рисунок 1.10 – Рекомендації щодо наборів предметів

На сайті є окремий блок де публікуються новини про додавання або оновлення посібників для персонажів гри. Такий підхід дає можливість відвідувачам сайту одразу бачити оновлення, а також сприяє збереженню цікавості до ресурсу та підтримці активності відвідувачів (рисунок 1.11).

Вибір потрібного персонажа здійснюється за допомогою розширеного меню з фільтрами зброї, якою володіє персонаж та його елементу. Фільтри допомагають гравцям швидко знаходити потрібного персонажа (рисунок 1.12).

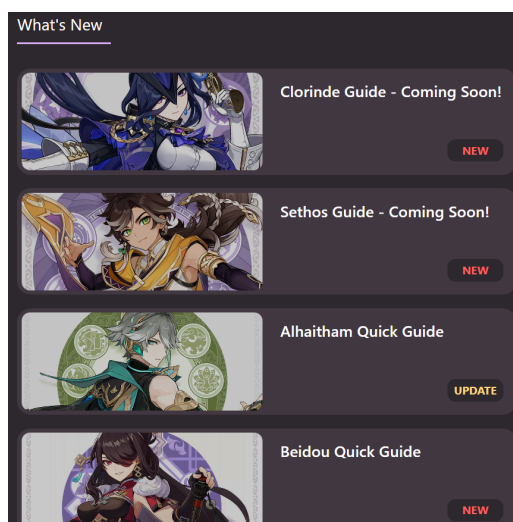


Рисунок 1.11 – Блок новин KeqingMains

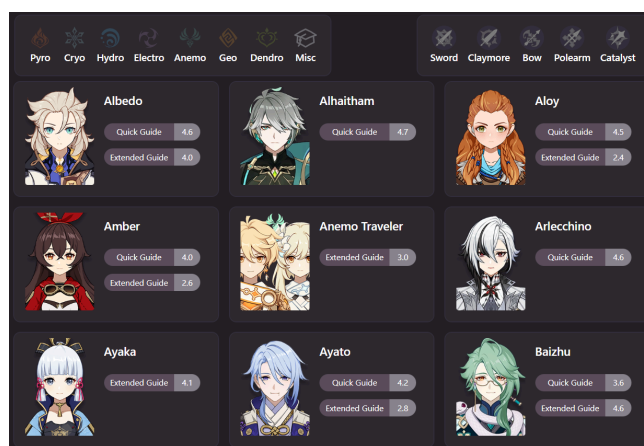


Рисунок 1.12 – Меню персонажів KeqingMains

Розширене меню вибору персонажа з фільтрами є важливим інструментом для гравців, особливо коли у грі доступна велика кількість персонажів.

На цьому ресурсі розміщені як стислі, так і розширені інструкції щодо максимізації корисності того чи іншого персонажа. Ці посібники дуже добре написані, містять багато нюансів і деталізованих порад, що допомагають гравцям оптимізувати ігровий процес. Кожен посібник повинен бути створений за загальноприйнятим шаблоном. Це забезпечує однорідність та зручність для користувачів, які шукають інформацію. Стандартизований шаблон дає можливість

гравцям легко знаходити необхідну інформацію та швидко орієнтуватися в посібниках для різних персонажів гри. Крім того, це сприяє якості контенту, оскільки автори можуть краще структурувати та організувати свої матеріали відповідно до шаблону, що спрощує сприйняття інформації гравцями.

Також для таких посібників передбачено зміст, який дає можливість швидко орієнтуватися в структурі посібника. Кожен пункт змісту може бути гіперпосиланням, що переносить користувача на відповідне місце на веб-сторінці. Зазвичай зміст містить основні розділи або теми, які охоплюються в посібнику, і може бути розділений на підрозділи (рисунки 1.13).

Table of Contents 1. New Content 2. Infographic 3. Character Overview 3.1. Playstyles 3.2. Talents 3.3. Constellations 3.4. Mirror Mechanics 3.5. Combos 4. Artifacts 4.1. ER Requirements 4.2. Artifact Stats 4.3. Artifact Sets 5. Weapons 6. Teams 6.1. Notable Teammates 6.2. Quicken 6.3. Hyperbloom 6.4. Burgeon 6.5. Nilou Bloom 7. Credits	Shorter Continuous (With Burst) <table> <tr> <td>Purpose</td><td>Maximizes 3-Mirror uptime to fit a shorter rotation: obtain 3 Mirrors sooner, C6 combo</td></tr> <tr> <td>Combo Duration</td><td>11.5–12s</td></tr> <tr> <td>Rotation Duration</td><td>18–20s</td></tr> <tr> <td>Quickhand Notation</td><td>Q + E + N3D N3D N3CD + N3CD N3CD</td></tr> <tr> <td>Expanded Notation</td><td> 1. Burst 2. Skill 3. Attack for ~5s 4. Charged Attack 5. Attack for ~4s </td></tr> <tr> <td>Notes</td><td>This combo fits into the shortest possible rotations where Alhaitham uses his Burst every rotation.</td></tr> </table> Continuous (No Burst)	Purpose	Maximizes 3-Mirror uptime to fit a shorter rotation: obtain 3 Mirrors sooner, C6 combo	Combo Duration	11.5–12s	Rotation Duration	18–20s	Quickhand Notation	Q + E + N3D N3D N3CD + N3CD N3CD	Expanded Notation	1. Burst 2. Skill 3. Attack for ~5s 4. Charged Attack 5. Attack for ~4s	Notes	This combo fits into the shortest possible rotations where Alhaitham uses his Burst every rotation.
Purpose	Maximizes 3-Mirror uptime to fit a shorter rotation: obtain 3 Mirrors sooner, C6 combo												
Combo Duration	11.5–12s												
Rotation Duration	18–20s												
Quickhand Notation	Q + E + N3D N3D N3CD + N3CD N3CD												
Expanded Notation	1. Burst 2. Skill 3. Attack for ~5s 4. Charged Attack 5. Attack for ~4s												
Notes	This combo fits into the shortest possible rotations where Alhaitham uses his Burst every rotation.												

Рисунок 1.13 – Фрагмент посібника KeqingMains

На KeqingMains постійно публікуються оновлення, що враховують зміни в грі. Оскільки Genshin Impact регулярно отримує новий контент та оновлення, сайт адаптує свої матеріали, щоб залишатися актуальним і корисним ресурсом для гравців.

Розглянуті ресурси допомагають сформувати загальне уявлення про те, як має виглядати відповідне програмне забезпечення для Mobile Legends. Вже існуючі аналоги для цієї гри не можуть зрівнятися з наведеними веб-сайтами того ж типу для інших ігор, а отже потрібно розробити веб-тренажер, який буде на рівні з цими веб-сайтами.

1.3 Задачі, які розв’язуються програмним забезпеченням

Створення такого ресурсу вимагає врахування багатьох аспектів, включаючи зручність навігації, детальність інструкцій, актуальність рекомендацій та інтерактивність вмісту.

Для розробки даного веб-застосунку поставлено такі задачі:

- розробити дизайн веб-сторінки;
- розробити меню героїв;
- розробити фільтри для пошуку героїв за роллю та лініями;
- розробити показові інструкції для кількох героїв, які включатимуть огляд їхніх навичок, сильних і слабких сторін;
- розробити стрічку новин;
- зв’язати інтерфейс застосунку з базою даних;
- оптимізувати інтерфейс під різні пристрої;
- працювати над оновленнями та над виявленням помилок.

Проведений аналіз аналогів дав змогу поставити чіткі задачі, які потрібно розробити. Враховано недоліки існуючих ресурсів для Mobile Legends, а також переваги ресурсів для інших ігор, таких як League of Legends і Genshin Impact.

Однією з головних задач є покращення інтерфейсу користувача. Багато існуючих ресурсів для Mobile Legends мають застарілий або незручний дизайн, що ускладнює навігацію та пошук необхідної інформації. Інтерфейс повинен бути інтуїтивно зрозумілим, з чітко структурованими розділами та легким доступом до пошукових і фільтрувальних функцій.

Функції, запропоновані ресурсами OP.GG і KeqingMains, стали хорошим орієнтиром для розробки власного веб-застосунку.

2 ЗАСОБИ РОЗРОБКИ

Для розробки будь-якого веб-сайту чи веб-застосунку використовується мова розмітки HTML (HyperText Markup Language) та мова таблиць стилів CSS (Cascading Style Sheets) [8], а також більшість веб-сайтів застосовують мову фронтенд-розробки JavaScript [9]. Проте для більшості проектів навіть такого набору інструментів мало. Використано бібліотеки та сторонні засоби для полегшення процесу розробки

2.1 Основні засоби розробки

Використано основні засоби розробки веб-сайтів – HTML, CSS і JavaScript. Це основа для всіх застосунків та сторінок на веб основі.

2.1.1 Мова розмітки HTML

Стандартна мова розмітки для створення й відображення веб-сторінок у браузерах Інтернету – HTML (Hypertext Markup Language). Вона використовується для створення структури та організації контенту на веб-сторінках, включаючи тексти, зображення, відео, гіперпосилання й інші елементи.

Мова HTML складається з набору тегів, кожен з яких має власну функцію у визначенні типу та структури контенту. Теги можуть бути вкладеними один в одного для створення складних елементів. Наприклад, тег `<html>` визначає початок і кінець документа HTML, тег `<head>` визначає мета-інформацію документа, таку як заголовок і метатеги, а тег `<body>` визначає основний вміст документа.

Крім основної структури документа, HTML також дає можливість вбудовувати різноманітні мультимедійні елементи, такі як зображення, відео та

аудіо, за допомогою відповідних тегів, таких як `<img>`, `<video>` та `<audio>`. Також можна створювати посилання на інші веб-сторінки за допомогою тега `<a>`.

Мова HTML дає можливість використовувати CSS (Cascading Style Sheets) для стилізації веб-сторінок, що дає можливість змінювати вигляд і розміщення елементів на сторінці. Також він може використовувати JavaScript для додавання динамічного змісту та взаємодії з користувачем.

2.1.2 Мова стилів CSS

Мова CSS (Cascading Style Sheets) – це мова стилів, яка використовується для оформлення веб-сторінок, створення їх зовнішнього вигляду та визначення стилів для різних елементів HTML. Мова CSS дає можливість відокремлювати вміст веб-сторінки від її представлення, що полегшує зміну дизайну без впливу на сам контент.

У CSS визначаються різні правила стилів, які вказують браузеру, як формувати конкретні елементи. Кожне правило стилю складається з селектора та блоку властивостей. Селектор визначає, які елементи HTML будуть застосовані до цього правила, а блок властивостей містить список стилів, таких як кольори, розміри, шрифти тощо.

Крім традиційних статичних стилів, CSS також підтримує анімацію і переходи, що дає можливість створювати рухомі ефекти та інтерактивність на сторінках.

2.1.3 Мова програмування JavaScript

Мова JavaScript – це високорівнева, об'єктно-орієнтована мова програмування, яка використовується для створення динамічного змісту і взаємодії на веб-сторінках. Вона є однією з основних технологій для розробки клієнтської частини веб-додатків.

Мова JavaScript використовується для реагування на події, які виникають на веб-сторінці, такі як клік мишею, натискання клавіші, завантаження сторінки та

інші. Він дає можливість виконувати різні дії відповідно до цих подій, такі як зміна змісту сторінки, валідація введених даних, анімація та багато іншого.

Дана мова програмування також дає можливість робити запити до сервера без перезавантаження сторінки за допомогою технології AJAX (Asynchronous JavaScript and XML), що дає можливість динамічно оновлювати вміст сторінки без зміни її загальної структури.

Мова програмування JavaScript також має багато вбудованих об'єктів та функцій для роботи з різними аспектами веб-розробки, такими як робота з DOM (Document Object Model), обробка подій, робота з формами, маніпуляції з даними та багато іншого. Він також підтримує розширення за допомогою сторонніх бібліотек та фреймворків, таких як jQuery, React, Angular та інші.

2.2 Допоміжні засоби розробки

Для сучасних веб-сайтів часто мало використовувати тільки стандартні HTML, CSS і JavaScript. Хоча ці технології є основними для створення веб-сторінок, їх поєднання з іншими інструментами і технологіями може значно покращити функціональність, продуктивність та дизайн веб-сайту. Не всі розглянуті засоби були застосовані для розробки програмної системи, але варто розуміти всі слабкі та сильні сторони кожного засобу.

2.2.1 Бібліотека React

Наперекір популярній думці, React [10] – це фреймворк, а відкрита бібліотека мови JavaScript для створення користувацьких інтерфейсів. Вона розроблена компанією Facebook і широко використовується для розробки сучасних веб-додатків та інтерфейсів користувача. Однією з ключових особливостей React є використання компонентної архітектури.

У React користувацький інтерфейс поділяється на невеликі, самостійні компоненти, які можуть бути вкладені один у одного. Кожен компонент відповідає

за відображення певної частини інтерфейсу та має власний стан і методи життєвого циклу. Це дає можливість покращити читабельність коду, підтримку та перевикористання компонентів.

Основна концепція React – це віртуальний DOM (Document Object Model). React використовує віртуальний DOM для ефективного відображення змін в інтерфейсі. Замість безпосередньої маніпуляції реальним DOM, React використовує віртуальний DOM для обчислення різниці між поточним та попереднім станом інтерфейсу, і тільки оновлює ті елементи, які дійсно змінилися. Це дає можливість покращити продуктивність додатків та забезпечити плавне відображення змін.

Ця бібліотека також підтримує використання JSX (JavaScript XML), який дає можливість включати HTML-подібний синтаксис безпосередньо в JavaScript коді компонентів. Це полегшує створення та розуміння структури інтерфейсу.

Більше того, React має велику екосистему [11], яка складається з багатьох корисних бібліотек та інструментів для розробки, таких як React Router для навігації, Redux для управління станом додатків, Material-UI для реалізації Material Design, і багато інших.

2.2.2 Бібліотека React Router

Допоміжна бібліотека React Router [12] – це бібліотека для маршрутизації веб-додатків на основі React. Вона дає можливість реалізувати навігацію між різними сторінками або виглядами в React-додатках без перезавантаження сторінки.

Основними компонентами React Router є `<BrowserRouter>`, `<Route>`, `<Link>` та `<Switch>`. `<BrowserRouter>` визначає контекст маршрутизації для додатка, `<Route>` визначає, які компоненти відображати на основі маршруту URL, `<Link>` використовується для створення гіперпосилань на різні маршрути, а `<Switch>` визначає групу маршрутів і відображає тільки перший, що відповідає поточному URL.

Ця бібліотека дає можливість використовувати динамічні маршрути, параметри й запити для створення більш гнучкої системи маршрутизації. Він також підтримує вкладені маршрути, аутентифікацію та авторизацію, захист маршрутів й інші просунуті функції.

2.2.3 Система Vite

Інструмент Vite [13] – це нова модерна система для розробки веб-додатків, яка спрощує процес створення та розгортання веб-додатків за допомогою сучасних інструментів і технологій. Його основна мета – забезпечити швидку розробку та переваги, такі як миттєва ребілд і гаряче перезавантаження.

Основна ідея Vite полягає в тому, що він використовує ES модулі в браузері під час розробки. Це дає можливість уникнути процесу попередньої компіляції, що забезпечує миттєве відображення змін в коді без затримок. В результаті отримується надзвичайно швидкий час розробки.

Система Vite також підтримує різні типи файлів, такі як JavaScript, TypeScript, CSS, JSON і SVG, і автоматично обробляє їх у мережі, щоб забезпечити найкращу продуктивність під час розробки. Він також підтримує розширення за допомогою плагінів, що дає можливість розширити його функціональність за потреби.

Однією з ключових особливостей Vite – можливість використовувати різні інструменти для розробки, такі як React, Vue, Svelte. Він підтримує різні попередні налаштування, а також ті налаштування, які дають можливість швидко розпочати роботу з будь-яким інструментом розробки.

2.2.4 Бібліотека React-i18next

Можливість зміни мови реалізовано за допомогою React-i18next [14]. Це бібліотека для міжнародної локалізації (i18n) React-додатків. Вона надає прості та потужні засоби для локалізації текстів, форматування дат і чисел, а також інших мовних ресурсів в React-додатках.

Основні концепції, на яких базується react-i18next, включають використання файлів перекладу, ключової-значення (key-value) пар, що дають можливість реюзати переклади, а також підтримку різних мов та локалів.

Бібліотека react-i18next надає компоненти вищого рівня, такі як `<Trans>` та `useTranslation()`, які полегшують використання перекладів у React-додатках. Крім того, вона має розширені можливості для реактивного оновлення текстів під час зміни мови або локалі.

Ця бібліотека також підтримує різні стратегії завантаження перекладів, такі як завантаження локальних файлів, завантаження динамічних перекладів під час виконання, а також підтримку зовнішніх сервісів перекладу.

2.2.5 Фреймворк Angular

Фреймворк Angular [15] – це платформа для розробки односторінкових веб-застосунків, створений компанією Google. Його основною метою є спрощення розробки динамічних веб-сторінок, що дає можливість розробникам створювати складні та інтерактивні інтерфейси користувача. Основна архітектура Angular базується на компонентному підході, де кожен компонент є ізольованим блоком з власною логікою, HTML-шаблоном та стилями.

Один з ключових елементів Angular – це двостороннє зв'язування даних, яке забезпечує автоматичну синхронізацію між моделлю даних та відображенням у реальному часі. Це робить розробку більш інтуїтивною та зручною, оскільки зміни в інтерфейсі одразу відображаються в даних і навпаки. Angular також підтримує модульність, що дає можливість розбивати код на логічні блоки, які можуть бути незалежно розроблені та протестовані.

Ще одним важливим аспектом Angular є система залежностей, яка спрощує управління залежностями між компонентами. Ін'єкція залежностей дає можливість легко замінювати чи оновлювати окремі частини застосунку без зміни основного коду. Це значно підвищує гнучкість та масштабованість застосунків.

Фреймворк Angular використовує TypeScript як основну мову програмування, що додає типізацію та поліпшує інструментальні можливості для

розробників, включаючи автодоповнення коду, статичну перевірку та рефакторинг. Це сприяє написанню більш надійного та підтримуваного коду.

Рендеринг у Angular здійснюється за допомогою декларативних шаблонів, що дає можливість легко визначати, як дані повинні відображатися в інтерфейсі користувача. Це робиться за допомогою спеціальних директив, які можуть змінювати DOM або додавати динамічну поведінку до елементів HTML.

Фреймворк Angular також має потужну екосистему інструментів та бібліотек, таких як Angular CLI для створення та управління проєктами, а також вбудовані інструменти для тестування, оптимізації продуктивності та забезпечення безпеки. Це робить його популярним вибором для розробки масштабованих та продуктивних веб-застосунків.

Вибір між React і Angular часто зводиться до конкретних потреб проєкту та особистих уподобань розробників. React може бути кращим вибором через декілька ключових причин. По-перше, React забезпечує більшу гнучкість та легкість інтеграції з іншими бібліотеками або фреймворками, оскільки він фокусується лише на представленні даних (view layer) і не нав'язує жорсткої архітектури. Це дає можливість розробникам обирати інструменти та бібліотеки, які найкраще підходять для конкретних задач.

По-друге, React має нижчий поріг входження. Оскільки він є бібліотекою, а не повноцінним фреймворком, розробникам простіше почати з React і поступово нарощувати функціонал за потреби. Це особливо корисно для новачків або для команд, які хочуть швидко вивчити новий інструмент.

По-третє, React використовує JSX, який дає можливість писати компоненти з використанням знайомого синтаксису JavaScript, що спрощує процес розробки та робить код більш зрозумілим і легким для підтримки. Крім того, великі компанії, такі як Facebook, активно підтримують і розвивають React, що забезпечує стабільність та регулярні оновлення.

Нарешті, екосистема навколо React є надзвичайно розвиненою з великою кількістю сторонніх бібліотек та інструментів, що дає можливість швидко знаходити розв'язок для широкого спектра завдань. Це робить React привабливим

вибором для розробників, які шукають максимальну свободу та гнучкість у своїх проєктах.

2.2.5 Фреймворк Vue

Фреймворк Vue [16] – це прогресивний фреймворк для створення інтерфейсів користувача, відомий своєю простотою, гнучкістю та потужністю. Він пропонує легке впровадження у вже існуючі проєкти, що робить його привабливим для розробників, які хочуть поступово модернізувати свої застосунки. Vue використовує шаблони, що нагадують HTML, і підтримує двостороннє зв'язування даних, подібно до Angular, що спрощує роботу з даними і створення інтерактивних компонентів.

Однак, навіть при врахуванні всіх переваг Vue, React може бути кращим вибором через кілька важливих причин. По-перше, React має ширшу підтримку та більшу спільноту розробників, що означає більше ресурсів для навчання, більше доступних плагінів та інструментів, а також швидший пошук розв'язання проблем. Це забезпечує стабільність і підтримку для великих проєктів.

По-друге, React є продуктом компанії Facebook, яка активно підтримує та розвиває його. Це забезпечує постійні оновлення та впровадження нових функцій, а також гарантує довгострокову підтримку. На відміну від Vue, який розроблений і підтримується спільнотою та невеликою групою основних розробників, React має більш надійну корпоративну підтримку.

По-третє, React забезпечує високу продуктивність завдяки використанню віртуального DOM, що мінімізує операції з реальним DOM і покращує швидкість роботи застосунків. Це особливо важливо для великих і складних проєктів, де продуктивність має вирішальне значення.

Крім того, React використовує JSX, що дає можливість комбінувати JavaScript та HTML в одному файлі, роблячи код більш зрозумілим і простим у написанні та підтримці. Ця особливість робить React привабливим для розробників, які хочуть мати більше контролю над своїм кодом та його структурою.

Загалом, хоча Vue є відмінним фреймворком з багатьма перевагами, React може бути кращим вибором через його більшу підтримку спільноти, корпоративну підтримку від Facebook, високу продуктивність та гнучкість.

2.2.6 Фреймворк Bootstrap

Фреймворк Bootstrap [17] — це комплексний фреймворк для фронтенд-розробки, який надає набір готових компонентів і утиліт для створення адаптивних і привабливих веб-сайтів. Він включає стандартні компоненти, такі як кнопки, навігаційні панелі, модальні вікна та багато інших, що дає можливість швидко створювати прототипи та готові проекти. Bootstrap використовує систему сіток, яка допомагає легко організовувати макети, і має вбудовану підтримку адаптивного дизайну, що робить його ідеальним для створення сайтів, які добре виглядають на різних пристроях.

2.2.8 Фреймворк Tailwind

Фреймворк Tailwind CSS [18], на відміну від Bootstrap, є утилітарним CSS-фреймворком, який надає набір низькорівневих класів для створення користувацьких інтерфейсів. Замість використання готових компонентів, розробники можуть використовувати ці класи для створення унікальних дизайнів, які точно відповідають їхнім вимогам.

Фреймворк Tailwind дає можливість працювати безпосередньо з CSS у HTML, що значно спрощує та прискорює процес розробки, знижуючи потребу в написанні власного CSS. Він також підтримує конфігурацію та розширення, що дає змогу налаштовувати його під конкретні потреби проекту.

Розмір веб-застосунку може вплинути на вибір технологій і інструментів для розробки. Якщо застосунок є невеликим і має обмежений функціонал, то використання CSS-фреймворків, таких як Tailwind CSS або Bootstrap, може виявитися зайвим.

Це може бути пов'язано з тим, що використання цих фреймворків може вимагати додаткового часу на ознайомлення з їхніми можливостями та впровадження в проєкт.

У випадку невеликого застосунку, розробка може бути ефективною без використання готових CSS-фреймворків. Розробники можуть написати власний CSS або використовувати менші бібліотеки чи інструменти для стилізації, які відповідають конкретним потребам проєкту. Це може дати можливість зберегти час і ресурси, які були б витрачені на інтеграцію та налаштування готових фреймворків.

2.3 Засоби проєктування і дизайну

Для реалізації проєкту потрібно визначити, яким редактором коду користуватися, і в якій програмі розробити макет веб-сторінок. Також треба створити дрібну графіку, таку як піктограми, і контролювати версії розробленого застосунку.

2.3.1 Інструмент Figma

Для створення концептуальної моделі зовнішнього вигляду веб-застосунку було використано Figma [19]. Це інструмент для дизайну та прототипування інтерфейсів користувача, який працює в браузері. Він дає можливість дизайнерам створювати та редагувати інтерфейси в режимі реального часу, забезпечуючи можливість спільної роботи над проєктами.

Однією з основних переваг Figma є її хмарна платформа, що дає можливість зберігати всі проєкти онлайн, надаючи доступ до них з будь-якого місця та пристрою. Це забезпечує плавну співпрацю між дизайнерами, розробниками та іншими учасниками команди.

Інструмент Figma пропонує широкий спектр інструментів для створення графічних елементів, включаючи векторні інструменти, текстові блоки, форми та

стилі. Інтерфейс інструменту інтуїтивно зрозумілий та схожий на традиційні програми для графічного дизайну. Дизайнери можуть створювати інтерактивні прототипи, що дає можливість тестувати інтерфейси без необхідності написання коду. Можна створювати переходи між різними екранами, додаючи інтерактивність і реалістичність до прототипів.

2.3.2 Редактор коду Sublime Text

Редактор Sublime Text [20] – це текстовий редактор, який широко використовується серед програмістів та розробників для написання коду. Він відомий своєю швидкістю, легкістю використання та потужними можливостями. Основні особливості Sublime Text включають розширений пошук та заміну, автоматичне доповнення коду, можливість розділення екрану на декілька панелей для одночасного перегляду різних частин файлу, а також можливість редагувати та переглядати файли у режимі реального часу.

Інтерфейс Sublime Text досить простий та інтуїтивно зрозумілий, але при цьому він наділений потужними функціями, які дають можливість розробникам працювати ефективно. Редактор підтримує ряд мов програмування, а також включає можливості розширення функціональності за допомогою різних плагінів та розширень, що дає можливість адаптувати його до конкретних потреб користувача.

Одним з ключових переваг Sublime Text є його швидкість роботи. Він має мінімальний час завантаження та швидку реакцію на дії користувача, що робить його ідеальним для роботи з великими проєктами та великими обсягами коду. Крім того, Sublime Text має широку спільноту користувачів, яка підтримує різні розширення та надає корисні поради та рішення для роботи з редактором.

2.3.3 Редактор коду Microsoft Visual Studio Code

Використано Microsoft Visual Studio Code, або VS Code [21], – безкоштовний редактор коду з відкритим вихідним кодом, розроблений компанією Microsoft. Він підтримує різні операційні системи, такі як Windows, macOS та Linux. VS Code

відзначається своєю високою продуктивністю, швидкістю та широкою функціональністю, що робить його популярним серед розробників.

Основними характеристиками Visual Studio Code є підтримка великої кількості мов програмування, таких як JavaScript, Python, Java, C++, TypeScript та багатьох інших. Це досягається завдяки розширенням, які можна легко встановити з вбудованого ринку розширень. Розширення дають можливість додати підтримку нових мов, інструментів для дебагінгу, систем контролю версій та інших функцій, що робить VS Code дуже гнучким та адаптивним до потреб користувача.

Редактор VS Code має інтуїтивно зрозумілий інтерфейс з можливістю налаштування, включаючи теми, гарячі клавіші та налаштування робочого простору. Редактор підтримує потужні функції, такі як IntelliSense, який надає інтелектуальні підказки під час написання коду, автозаповнення та інформацію про параметри функцій. Інтегрований дебагер дає можливість проводити відлагодження коду безпосередньо з редактора, що значно спрощує процес розробки та тестування.

Однією з важливих особливостей VS Code є вбудована підтримка Git та інших систем контролю версій, що дає можливість легко відслідковувати зміни в коді, робити коміти та керувати гілками без виходу з редактора.

2.3.4 Система контролю версій Git

Система Git [22] – це розподілена система керування версіями, яка використовується для відстеження змін у вихідному коді під час розробки програмного забезпечення. Вона дає можливість кільком розробникам працювати над одним проектом одночасно, зберігаючи всі зміни, що дає можливість легко відновлювати попередні версії коду та працювати з різними гілками проекту.

Однією з основних переваг Git є його розподілений характер. Кожен розробник має повну копію репозиторію, включаючи всю історію змін, що забезпечує високу надійність та можливість роботи в автономному режимі. Це також дає можливість легко об'єднувати зміни з різних копій репозиторію. Основні операції в Git включають коміти, які фіксують зміни у локальному

репозиторії, та пуші, які відправляють зміни на віддалений сервер. Інші важливі операції включають мерджі для об'єднання змін з різних гілок, і пуллі, які отримують зміни з віддаленого репозиторію.

Система Git підтримує роботу з гілками, що дає можливість розробникам працювати над різними функціями чи виправленням багів у відокремлених середовищах. Це забезпечує можливість паралельної розробки та полегшує інтеграцію змін у основну гілку проекту. Крім того, Git має вбудовані механізми для розв'язання конфліктів, які можуть виникати під час об'єднання змін з різних гілок. Він також підтримує різні системи хостингу, такі як GitHub, GitLab та Bitbucket, які надають додаткові інструменти для управління проектами, спільної роботи та інтеграції з іншими сервісами.

На основі аналізу програмних засобів обґрунтовано використання основних технологій, що використовуються в сучасній веб-розробці, такі як HTML, CSS та JavaScript. Мова HTML [23] використовується для створення структури веб-сторінок, CSS – для їх стилізації, а JavaScript – для додавання динамічності та взаємодії.

Далі були розглянуті допоміжні засоби розробки, такі як React, React Router, Vite та React-18next. Проведено аналіз відмінностей Angular, React і Vue, а в кінцевому результаті вибрано React для розробки односторінкового веб-застосунку. Бібліотека React є потужною бібліотекою для створення користувацьких інтерфейсів, а React Router дає можливість легко управляти навігацією в React-додатках. Система Vite, з іншого боку, надає швидку та продуктивну розробку веб-додатків, а React-18next – засіб для міжнародної локалізації React-додатків.

Нарешті, представлені засоби проектування і дизайну, такі як Figma для створення макетів інтерфейсів і Microsoft Visual Studio Code для розробки коду. Крім того, розглянуто Git як систему керування версіями, яка дає можливість зберігати та відслідковувати зміни у вихідному коді.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-ТРЕНАЖЕРА

Під час реалізації веб-тренажера було використано різні методи розробки на мові JavaScript. Стали перевагою особливості бібліотеки React, які спростили процес розробки застосунку.

3.1 Структура веб-застосунку

Створено ряд окремих компонентів [24-26], які взаємодіють між собою і утворюють цілісну програму. Кожен з цих компонентів виконує певну функцію або набір функцій, і вони спільно працюють для забезпечення всіх функціональних можливостей програми.

3.1.1 Основні файли

Файл `main.jsx` – це головний файл у структурі. В ньому реалізується завантаження роутера для переключення сторінок всередині веб-застосунку. Крім того, саме в цьому файлі створюється віртуальну об'єктну модель документа (рисунок 3.1).

Файл `App.jsx` – другий за важливістю файл. У ньому записані всі функції, які виконуються в застосунку, а також у нього завантажуються компоненти головної сторінки, такі як: пошук, меню героїв, пагінація (рисунок 3.2). Він виступає як центральний вузол програми, що керує рендерингом та функціональністю інших компонентів.

3.1.2 Компоненти

Компонент `Navigation` містить у собі реалізацію навігаційного блоку у верхній частині сторінки. У ньому також розміщено компоненти `ThemeSwitch` і `LangSwitch` (рисунок B1).

```

import React from 'react'
import App from './App'
import Nana from './guides/Nana/Nana';
import ReactDOM from "react-dom/client";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Navigation from './components/nav/Navigation';

import './index.css';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <BrowserRouter>
      <Navigation/>
      <Routes>
        <Route path="/" index element={<App />}/>
        <Route path="nana" element={<Nana />}/>
      </Routes>
    </BrowserRouter>
  </React.StrictMode>,
);

```

Рисунок 3.1 – Вміст файлу main.jsx

```

return (
  <div className="App">
    <section className="hero-select">
      <SearchBar searchName={searchName} handleSearch={handleSearch} />
      <RoleSelect handleChangeRole={handleChangeRole} />
      <HeroResults currentHeroes={currentHeroes} searchName={searchName} highlightText={highlightText} />
      <Pagination
        heroesPerPage={heroesPerPage}
        totalHeroes={filteredHeroes.length}
        paginate={paginate}
        currentPage={currentPage}
      />
      <LaneSelect handleChangeLane={handleChangeLane} />
    </section>

    <section className="news">
      <News />
    </section>
  </div>
);

```

Рисунок 3.2 – Фрагмент App.jsx

У компоненті News створюється блок новин, який оновлюється адміністратором веб-застосунку. Основна мета компонента – відображення актуальної інформації про стан посібників (рисунк B2)

У компоненті HeroResults відбувається відображення списку героїв. Кожен герой представлений у вигляді картки з фоновим зображенням, ім'ям та інформацією про його роль і лінію (рисунк B3).

Для кожного героя зі списку currentHeroes створюється відповідна картка. Ця картка містить ім'я героя, яке може бути підсвічене, якщо відбувається пошук за ним. Крім того, вона містить піктограми для ролі та лінії героя.

Якщо для героя існує відповідний гайд, картка перетворюється на посилання, яке веде на сторінку з детальною інформацією про героя. В іншому випадку, картка просто відображається без можливості переходу.

У компоненті RoleSelect створюється форма для вибору ролей героїв. Кожна роль представлена у вигляді чекбокса, який дає можливість користувачеві вибрати одну або декілька ролей для фільтрації героїв за цим параметром. Компонент містить форму з класом RoleSelect, в якій для кожної ролі зі списку (Tank, Fighter, Assassin, Marksman, Mage, Support) генерується чекбокс. Для кожної ролі створюється `<label>`, в якому вміщується `<input type="checkbox">`. Класи `label-${role.toLowerCase()}` встановлюються для міток з метою стилізації кожної ролі окремо за допомогою CSS. Коли користувач змінює стан прапорця, викликається функція `handleChangeRole` (рисунк B4).

У компоненті LaneSelect так само як і в попередньому RoleSelect, створюється форма для вибору ліній героїв. Кожна лінія представлена у вигляді прапорця, що дає можливість користувачеві вибрати одну або декілька ліній для фільтрації героїв за цим параметром (рисунк B5).

У компоненті Pagination відображається навігація між сторінками з героями. Цей компонент приймає кількість героїв на сторінці (`heroesPerPage`), загальну кількість героїв (`totalHeroes`), функцію для зміни сторінки (`paginate`) та поточну сторінку (`currentPage`). Використовується умовний оператор, який перевіряє, чи загальна кількість героїв перевищує кількість героїв на сторінці. Якщо це так, то

генерується список номерів сторінок у вигляді неупорядкованого списку (ul). Кожен номер сторінки представлений як li елемент списку з класом pageNumber. Якщо поточна сторінка співпадає з номером сторінки, то на елементі додається клас activePage, щоб виділити її візуально (рисунк B6).

У компоненті ThemeSwitch реалізовано перемикач для зміни між світлим та темним режимами веб-застосунку. При натисненні на компонент, перевіряється поточний стан darkMode. Якщо він дорівнює false, тоді встановлюється значення true, що вмикає темний режим. (рисунк B7).

У компоненті LangSwitch реалізовано перемикач для зміни мови веб-сайту між англійською та українською. При натисненні на кнопку зображення прапорця, викликається функція toggleLanguage (рисунк B8).

Використовуються хуки useState для збереження поточної мови та useTranslation для отримання перекладів. Кнопка для зміни мови має клас change-lang, а зображення прапорця має клас flag-icon. Залежно від поточної мови, використовується відповідне зображення прапорця для англійської та української мови.

Компонент Nana слугує прикладом ігрового посібника для героя. Спершу відображається зміст з посиланнями на розділи, такі як "Gameplay", "Items" та "Synergy". Потім йде опис героя, включаючи його портрет, ім'я та іншу інформацію. Поряд з описом героя наведено ігрову інформацію, таку як роль та локація. Після цього йдуть розділи "Gameplay", "Items" та "Synergy", де відображається відповідна інформація про героя та його геймплей. Кожен розділ містить заголовок і якрне посилання, що дає можливість швидко перейти до відповідного розділу (рисунк B9).

3.2 Функції програмної системи

Для забезпечення роботи функціональних можливостей системи були розроблені функції за допомогою хуку React useState та useEffect. Такі функції

працюють краще, ніж функції звичайного JavaScript, оскільки виконуються асинхронно за замовчуванням, що дає можливість розміщувати контент динамічно на сторінці

Функція `handleChangeRole` викликається при зміні стану прапорця. Вона отримує об'єкт події `event` і витягує з нього значення `id` і `checked` прапорця. Потім знаходить мітку та контейнер прапорця за його `id`. Якщо прапорець відзначений, вона додає `id` до масиву `searchRole`, а також додає клас `"checked"` до мітки і контейнера. Якщо прапорець не відзначений, вона видаляє `id` з масиву `searchRole` і видаляє клас `"checked"` з мітки і контейнера (рисунок B10).

Функція `handleChangeLane` є аналогічною до попередньої функції `handleChangeRole`. Вона використовується для обробки змін стану прапорців, що відповідають за вибір ліній героя. Як і попередня функція, `handleChangeLane` керує станом і стилізацією прапорців залежно від вибору користувача (рисунок B11).

Функція `handleSearch` викликається при зміні значення в полі пошуку. Параметр `event` представляє об'єкт події, який містить інформацію про зміни в полі вводу. Функція отримує значення поля пошуку за допомогою `event.target.value` та встановлює його в стан компоненту, використовуючи сеттер `setSearchName`. Ця функція відповідає за оновлення стану пошукового запиту на основі введеного користувачем тексту (рисунок B12).

Функція `highlightText` використовується для виділення тексту, який відповідає запиту пошуку, у вигляді підсвічених частин. Вона отримує два параметри: `text`, який представляє текст, який потрібно відобразити, та `searchName`, який представляє рядок запиту пошуку. Якщо рядок пошуку порожній, функція просто повертає вихідний текст без будь-яких змін. Якщо рядок пошуку не порожній, функція розбиває текст на частини за допомогою регулярного виразу, де розділювачем є сам рядок пошуку.

Після розділення тексту функція перебирає кожен частину. Якщо частина тексту відповідає запиту пошуку (за врахуванням регістру), то вона обгортається

тегом `<span>` з класом "highlight", що надає їй стилізацію. В іншому випадку частина тексту повертається без змін.

На виході функція повертає масив частин тексту, де частини, які відповідають запиту пошуку, підсвічені (рисунки B13).

Функція `paginate` викликається при переході на іншу сторінку в пагінації. Вона отримує номер сторінки, на яку потрібно перейти, і встановлює його в стані компоненту за допомогою `setCurrentPage`. Це дає можливість змінювати поточну сторінку пагінації при кліку на номер сторінки (рисунки B14).

У наступному фрагменті коду спочатку обчислюються індекси для визначення першого та останнього героїв на поточній сторінці пагінації. Далі створюється масив `filteredHeroes`, що містить героїв, які відповідають умовам пошуку та фільтрування. Герої фільтруються за наступними критеріями: ім'я героя, роль героя (відповідає будь-якій вибраній ролі в фільтрі), лінія героя (відповідає будь-якій вибраній лінії в фільтрі).

Після фільтрації героїв за алфавітом, визначається масив `currentHeroes`, який містить героїв, котрих потрібно відображати на поточній сторінці. Також використовується `useEffect`, щоб встановити першу сторінку пагінації, коли змінюється довжина масиву `filteredHeroes` (рисунки B15).

Функції `handleClickPassive`, `handleClickAbility1`, `handleClickAbility2` та `handleClickAbility3` відповідають за зміну видимості опису різних навичок героя в компоненті. Кожна з цих функцій встановлює значення `visibleDescription` відповідно до обраної навички. Якщо `visibleDescription` вже містить ідентифікатор навички, то воно змінюється на `null`, що призводить до приховання опису. В іншому випадку, `visibleDescription` встановлюється на ідентифікатор клікнутої навички, що призводить до відображення відповідного опису (рисунки B16).

Наведені функції забезпечують правильну роботу веб-застосунка, впроваджують нові можливості. Серед можливостей, що забезпечуються функціями є: фільтрація списку за ролями героїв, фільтрація списку за лініями героїв, пошук у списку за ім'ям героя, виділення шуканого слова в

результатах, переключення сторінок списку героїв, динамічне розгорнення опису навичок.

3.3 Масив даних

Як тимчасовий варіант способу збереження даних про героїв було обрано занести дані в JS-файл і створити масив об'єктів heroData. Проте після ретельного обмірковування було вирішено залишити раніше обране рішення як постійне. Оскільки зв'язків між даними немає, немає і сенсу створювати базу даних і більше навантажувати систему. Таке рішення не тільки економить ресурси серверу, але і пришвидшує роботу веб-застосунку на всіх платформах (рисунок 3.3).

```
export const heroData = [  
  {  
    name: "Aamon",  
    role: ["Assassin", ""],  
    lane: ["Jungle", ""],  
    image: "src/img/aamon.png",  
  },  
  {  
    name: "Akai",  
    role: ["Tank", ""],  
    lane: ["Roam", ""],  
    image: "src/img/akai.png",  
  },  
  {  
    name: "Aldous",  
    role: ["Fighter", ""],  
    lane: ["Experience", ""],  
    image: "src/img/aldous.png",  
  },  
  {  
    name: "Alice",  
    role: ["Mage", "Tank"],  
    lane: ["Experience", ""],  
    image: "src/img/alice.png",  
  },  
]
```

Рисунок 3.3 – Масив heroData

Даний масив складається з об'єктів, кожен з яких має ім'я (name), роль (role), лінію (lane) і посилання на зображення (image). Також у випадку якщо на сайті існує посібник на героя, то буде також властивість guide – посилання на посібник.

Використання JavaScript для зберігання даних має кілька суттєвих переваг, які роблять цей підхід привабливим для невеликих проектів або прототипів. Одна з головних переваг полягає в його простоті. Дані можуть бути збережені безпосередньо у файлі JavaScript у вигляді масиву об'єктів, що дає можливість швидко і легко додавати нові записи або змінювати існуючі без необхідності налаштовувати і обслуговувати базу даних. Це значно спрощує розробку та обслуговування проекту, особливо для розробників, які не мають досвіду роботи з базами даних.

Ще однією важливою перевагою є швидкість доступу до даних. Оскільки дані зберігаються локально у файлі JavaScript, немає потреби виконувати запити до бази даних, що може значно знизити затримки і підвищити продуктивність веб-застосунку. Це особливо важливо для веб-застосунків, де швидкість реакції на дії користувача є критичною.

Крім того, зберігання даних у файлі JavaScript робить їх легко доступними для обробки на клієнтській стороні. Оскільки JavaScript є основною мовою програмування для клієнтської частини веб-застосунків, робота з даними безпосередньо у JavaScript спрощує їхню обробку і відображення. Це також сприяє зменшенню кількості коду, що потребує виконання, оскільки немає необхідності у додаткових інструментах чи бібліотеках для доступу до даних.

Легкість в обслуговуванні є ще однією перевагою цього підходу. Додавання нових героїв або зміна існуючих даних може бути здійснена просто шляхом редагування файлу JavaScript, що зручно і швидко. Це також означає, що для управління даними не потрібні додаткові інструменти або інтерфейси, що може знизити витрати на розробку та підтримку.

У розділі показано, як окремі компоненти, виконуючи свої специфічні функції, взаємодіють між собою для забезпечення коректної роботи програми.

Кожен компонент розроблений таким чином, щоб бути максимально автономним і гнучким.

Функції, реалізовані в рамках цих компонентів, забезпечують ключову логіку програми і користувацький досвід. Вони відповідають за обробку даних, взаємодію з користувачем і інтеграцію різних частин системи.

Загалом, програмна реалізація демонструє інтеграцію різних технологій і підходів для створення надійного, масштабованого та зручного для користувача веб-застосунку. Така структура дає можливість забезпечити високу якість продукту і гнучкість у його подальшому розвитку.

3.4 Взаємодія програмних модулів

Щоб показати зв'язок між компонентами системи, використано діаграму компонентів. Діаграма компонентів є одним із видів UML діаграм, які використовуються для моделювання фізичної структури системи. Вона допомагає зрозуміти, як різні частини системи взаємодіють між собою та як вони організовані (рисунки 3.4).

Файл `main.jsx` завантажує `App.jsx`. У файлі `App.jsx` завантажується кілька компонентів: `Navigation`, `News`, `HeroResults`, `RoleSelect`, `LaneSelect`, `Pagination`. Компонент `Navigation` включає компоненти `ThemeSwitch` та `LangSwitch`. Компонент `HeroResults` включає `Nana` для відображення посібників конкретних героїв. Кожен компонент розроблений таким чином, щоб бути максимально автономним і гнучким.

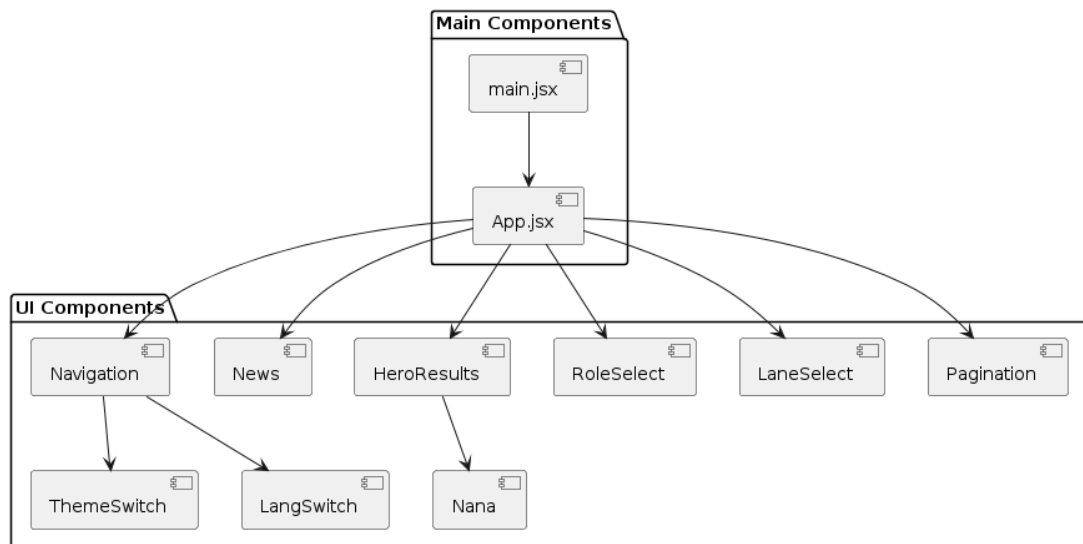


Рисунок 3.4 – Діаграма компонентів

У цьому розділі показано, як окремі компоненти, виконуючи свої специфічні функції, взаємодіють між собою для забезпечення коректної роботи програми.

Функції, що реалізовані в рамках цих компонентів, забезпечують ключову логіку програми і користувацький досвід. Вони відповідають за обробку даних, взаємодію з користувачем і інтеграцію різних частин системи.

Загалом, програмна реалізація демонструє інтеграцію різних технологій і підходів для створення надійного, масштабованого та зручного для користувача веб-застосунку. Така структура дає можливість забезпечити високу якість продукту і гнучкість у його подальшому розвитку.

4 РОБОТА КОРИСТУВАЧА З ВЕБ-ТРЕНАЖЕРОМ

Перевага веб-застосунків у тому, що вони доступні з будь-якого пристрою з підключенням до Інтернету без необхідності встановлення. Це забезпечує користувачам зручність і гнучкість у використанні програмного забезпечення, оскільки вони можуть працювати з веб-застосунками з будь-якого місця і в будь-який час.

Крім того, веб-застосунки автоматично оновлюються на сервері. Використання такої можливості гарантує, що всі користувачі завжди мають доступ до останніх версій і функцій без необхідності вручну завантажувати і встановлювати оновлення.

4.1 Зовнішній вигляд системи

Насамперед користувач може змінювати тему веб-застосунку за допомогою перемикача (рисунок 4.1).



Рисунок 4.1 – Перемикач теми

Далі з правого боку розміщено перемикач мови. Мови за замовчуванням дві – англійська та українська (рисунки 4.2, 4.3).

Коли користувач натискає на кнопку зміни мови, мова перемикається для всіх сторінок, а не тільки для тієї, на якій натиснено кнопку.

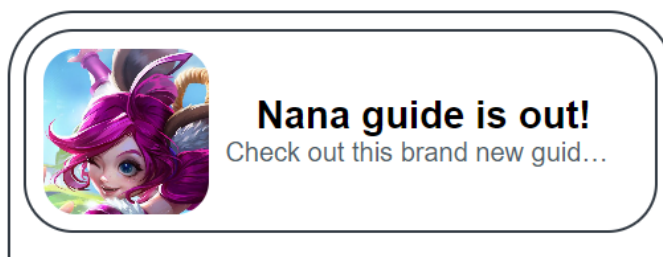


Рисунок 4.2 – Елемент інтерфейсу англійською мовою

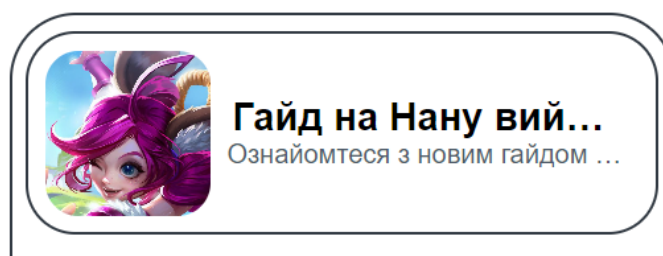


Рисунок 4.3 – Елемент інтерфейсу українською мовою

Нижче на сторінці розташований основний інтерфейс веб-застосунку – меню героїв. Це меню надає користувачам зручні інструменти для пошуку та фільтрації героїв відповідно до їхніх потреб. У верхній частині меню розташований пошуковий рядок, який дає можливість користувачам швидко знаходити героїв за ім'ям. Це особливо корисно, коли користувач вже знає, якого героя він шукає.

Під пошуковим рядком знаходиться фільтр ролей. Цей фільтр дає можливість відсортувати героїв за їхніми ігровими ролями, такими як танк, підтримка, атакувальний герой тощо. Така фільтрація допомагає користувачам легко знайти героїв, які відповідають їхнім вимогам у грі, залежно від тактичних потреб команди.

Справа від списку героїв розташований фільтр ліній. Він дає можливість користувачам відсортувати героїв за їхньою позицією на полі бою, наприклад, верхня, середня або нижня лінія. Це допомагає швидко знайти героїв, які найкраще підходять для гри на конкретній лінії.

Користувач має можливість комбінувати різні фільтри. Якщо вибрати кілька ліній або кілька ролей, то відображатимуться ті герої, які задовольняють хоча б один з обраних критеріїв. Наприклад, якщо обрано роль танка та підтримку, то відобразяться всі герої, які належать до будь-якої з цих ролей. Якщо ж вибрати хоча б одну роль і одну лінію, то відображатимуться лише ті герої, які відповідають як обраній ролі, так і обраній лінії. Це забезпечує точність пошуку та допомагає користувачам швидко знайти ідеального героя для конкретної ситуації (рисунок 4.4).

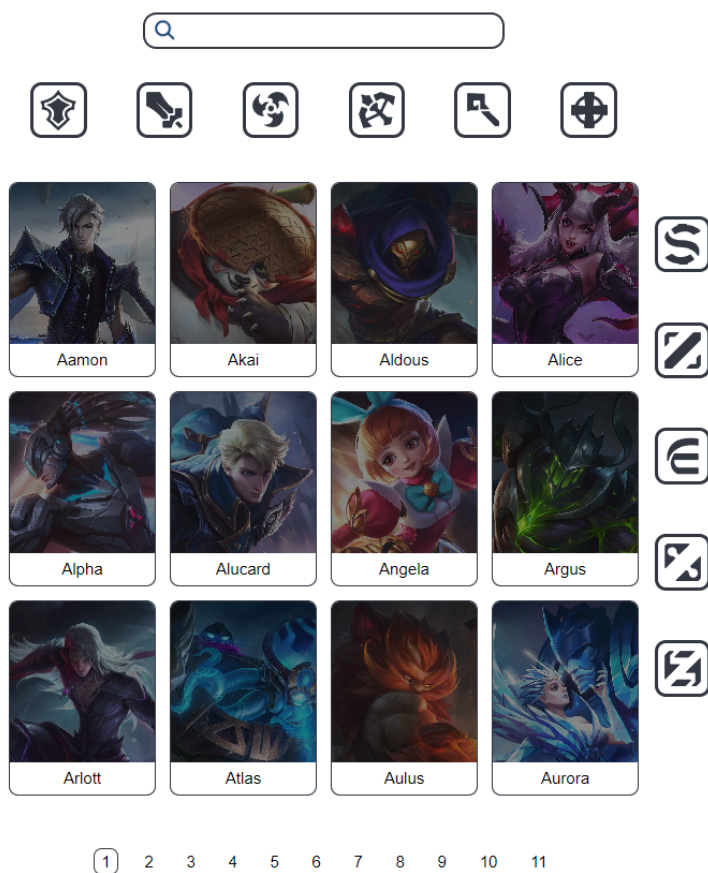


Рисунок 4.4 – Меню вибору героїв

Крім оновлень, у блоці новин можуть з'являтися оголошення про додання нових посібників.

Ці посібники можуть містити корисні поради щодо стратегії гри, детальну інформацію про героїв, рекомендації щодо використання їхніх здібностей та інші матеріали, які допоможуть гравцям покращити свої навички та досягти кращих результатів у грі (рисунки 4.5).

Що нового:

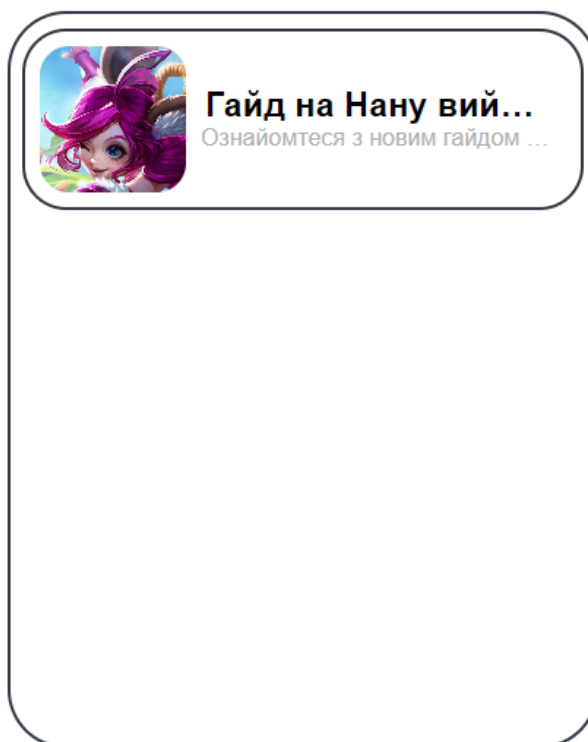


Рисунок 4.5 – Блок новин

Якщо натиснути на один із записів, відкриється відповідна сторінка. Наприклад зараз, натиснувши на новину про гайд на Нану користувач потрапить на сторінку посібника про Нану.

На початку сторінки можна бачити зображення героя, а також його короткий опис, який було взято із самої гри, а трохи правіше – зміст, який рухається з прокручуванням сторінки (рисунки 4.6).



Рисунок 4.6 – Профіль героя і зміст

Далі йде пункт “Основи геймплею”, в якому зібрана інформація про навички героя. У цьому розділі користувачі можуть детально ознайомитися з усіма здібностями героя, дізнатися про їхні особливості та ефекти. Кожна навичка описана з точки зору її застосування в грі, включаючи характеристики, тривалість дії, шкоду або лікування, а також інші важливі параметри.

Розділ також може містити поради щодо того, як ефективно використовувати навички героя в різних ситуаціях (рисунок 4.7).

Основи геймплею

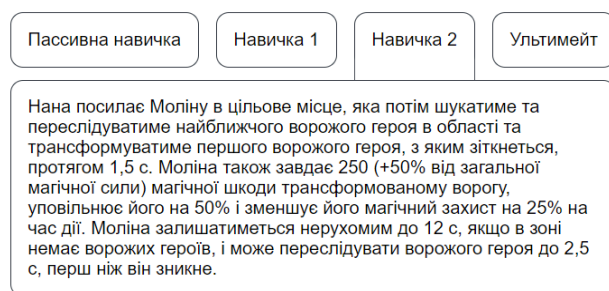


Рисунок 4.7 – Розділ “Основи геймплею”

Наприклад, показано як правильно поєднувати здібності для максимального ефекту, які комбінації навичок є найбільш потужними або які ситуації вимагають особливого підходу до використання здібностей.

Наступний розділ – “Предмети”, в якому зібрані рекомендації щодо того, які предмети купувати для даного героя, а також вказівки щодо того, для кого підходять ці предмети. У цьому розділі користувачі можуть знайти детальну інформацію про те, які предмети найкраще підходять для героя в різних ігрових ситуаціях.

Кожен предмет має опис його властивостей, ефектів та впливу на гру. Наприклад, тут можна дізнатися про предмети, що підвищують атаку, захист, швидкість або надають інші корисні бонуси. Рекомендації базуються на найкращих практиках і досвіді інших гравців, що допомагає користувачам ефективно використовувати ресурси та досягати максимальних результатів.

Окрім загальних рекомендацій, розділ також містить вказівки щодо того, для яких типів героїв підходять ті чи інші предмети (рисунок 4.8).

Предмети

 Clock of Destiny	Дає 25 НР і 4 магічної сили кожні 20с. До 12-ти разів. Коли Час досягне максимуму, герой отримає додаткові 5% магічної сили та 300 мани. Найкраще підходить для магів, які йдуть на жертви на початку гри, щоб збільшити шкоду та міцність наприкінці гри.
 Enchanted Talisman	Загалом кожні 10 секунд відновлює 15% максимальної кількості мани. Скорочення максимального часу відновлення збільшено на 5%. Найкраще підходить для магів або підтримки, які особливо покладаються на скорочення часу відновлення або відновлення мани.

Рисунок 4.8 – Розділ “Предмети”

Такий підхід дає можливість гравцям не тільки обрати правильні предмети для конкретного героя, але й зрозуміти, як ці предмети впливають на загальну стратегію гри. Наприклад, предмети, що підвищують магічну силу, можуть бути

рекомендовані для магів, тоді як предмети для збільшення броні та здоров'я більше підходять танкам.

У самому кінці сторінки розташований розділ “Синергія”, у якому пропонуються герої, які добре працюють разом із даним (рисунок 4.9).

Синергія

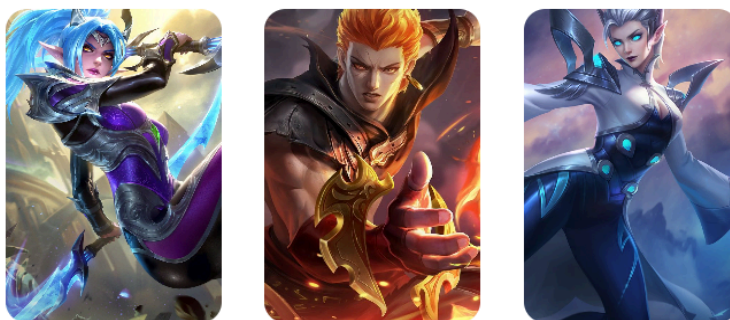


Рисунок 4.9 – Розділ “Синергія”

Там може розміщуватися така інформація, як: “Герой 1 чудово доповнює головного героя завдяки своїм унікальним здібностям, що підсилюють бойові навички команди. Герой 2 забезпечує важливу підтримку, надаючи необхідні ресурси та захист під час складних ситуацій. Герой 3 володіє здібностями, які синергічно взаємодіють з головним героєм, даючи можливість ефективніше реалізовувати спільні стратегії та досягати перемоги”.

4.2 Подальше вдосконалення на розширення

У подальшому планується розробити мобільну версію застосунку за допомогою React Native. React Native – це фреймворк для розробки мобільних додатків, що базується на JavaScript та React. Він дає можливість розробникам створювати мобільні застосунки для платформ iOS та Android, використовуючи ті

ж принципи, які застосовуються при розробці веб-додатків з React. Однією з особливостей React Native є можливість використання одного коду для обох платформ, що полегшує розробку та підтримку додатків для різних пристроїв.

Також в стані обмірковування перебуває ідея розширити повноту посібників, залучивши більш досвідчених гравців для їх створення. Залучення людей з багатим досвідом може значно покращити якість та корисність матеріалів, а також розширити їхній охоплення. Досвідчені гравці можуть надати цінні поради, кращі практики та глибше розуміння тематики, що допоможе новачкам краще зрозуміти та оволодіти предметом. Крім того, співпраця з досвідченими фахівцями може стимулювати обмін знаннями та ідеями, сприяючи загальному розвитку та підвищенню рівня всієї спільноти Mobile Legends.

Наостанок, розглядається варіант додавання більшої кількості мов для локалізацій веб-застосунку. Якщо включити більше мов у нову версію веб-застосунку, це приверне увагу тих користувачів, котрі не володіють ні англійською, ні українською мовами. Поширені мови у світі – іспанська, німецька, китайська.

4.3 Діаграма прецедентів

Щоб схематично показати взаємодію користувача з системою, створено діаграму прецедентів. Діаграма прецедентів (Use Case Diagram) є одним із видів UML-діаграм, яка використовується для моделювання функціональних вимог системи. Вона допомагає визначити, які дії можуть виконувати користувачі у системі (рисунок 4.10).

Опис прецедентів:

- перегляд посібників. Користувач або гість може переглядати посібники;
- пошук героя. Користувач або гість може шукати героїв за допомогою пошукового поля;

- фільтрація героя. Користувач або гість може фільтрувати героїв за роллю та лінією;
- перегляд новин. Користувач або гість може переглядати новини;
- зміна мови. Користувач може змінювати мову сайту між англійською та українською;
- зміна теми. Користувач може змінювати тему сайту між світлою та темною.

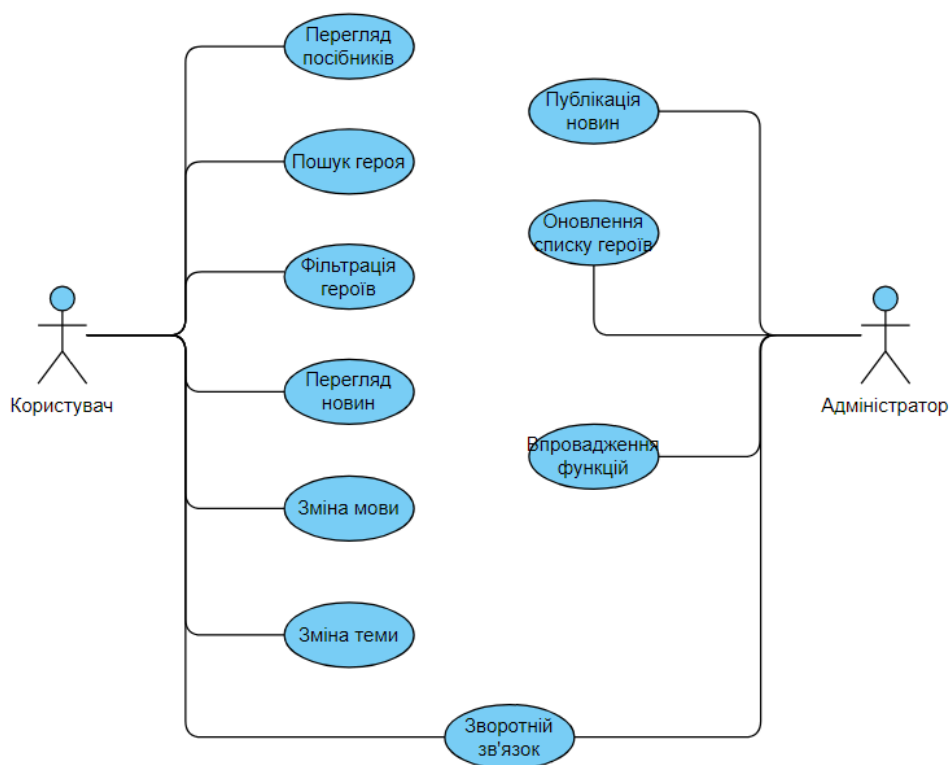


Рисунок 4.10 – Діаграма прецедентів

Показано принцип взаємодії користувача із системою. Користувач може вибрати потрібного героя за допомогою функції пошуку, функцій фільтрів та пагінації. Ці функції дають можливість користувачам ефективно взаємодіяти з системою, швидко знаходячи потрібну інформацію про героїв та обираючи тих, за яких вони хочуть навчитися грати або дізнатися більше про них. Також розглянуто різні можливості покращення системи у майбутньому.

ВИСНОВКИ

На основі аналізу методів, підходів та алгоритмів розв’язання поставлених задач обґрунтовано використання методів фронтенд-розробки, методів розробки веб-застосунків.

На основі аналізу засобів розробки програмного забезпечення обґрунтовано використання мови розмітки HTML, мови стилів CSS, мови програмування JavaScript, бібліотек React, React-i18next, системи Vite.

Розроблено програмну систему “Веб-тренажер для багатокористувацької онлайн-гри Mobile Legends”.

Показано модель взаємодії користувача з системою і на її основі доведено роботоспроможність системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кордяк Р.М., Кублій Л.І. Веб-тренажер для багатокористувацької онлайн-гри Mobile Legends. Сучасні проблеми наукового забезпечення енергетики. У 2-х т. : Матеріали XXI Міжнародної науково-практичної конференції молодих вчених і студентів, м. Київ, 23-26 квітня 2024 року. Київ: КПІ ім. Ігоря Сікорського, Вид-во “Політехніка”, 2024. Т. 2. С. 282-283.
2. Mobile Legends: Bang Bang. Mobile Legends: Bang Bang. URL: <https://m.mobilelegends.com/en> (дата звернення: 16.04.2024).
3. Mobile Legends: Bang Bang Wiki. Mobile Legends: Bang Bang Wiki | Fandom. URL: https://mobile-legends.fandom.com/wiki/MLBB_Wiki (дата звернення: 26.04.2024).
4. Mobile Legends Build is a Guidance For Mobile Legends Game. Mobile Legends Build is a Guidance For Mobile Legends Game. URL: <https://mobilelegendsbuild.com/> (дата звернення: 26.04.2024).
5. Alexandre Games. Alexandre Games Blog. URL: <https://alexandregames.com/> (дата звернення: 27.04.2024).
6. OP.GG - The Best LoL Builds and Tier List. Search Riot ID and Tagline for Stats. OP.GG. URL: <http://www.op.gg> (дата звернення: 20.04.2024).
7. Genshin Impact Guides and Theorcrafting | KQM. KQM. URL: <https://keqingmains.com/> (дата звернення: 24.04.2024).
8. MDN Web Docs. MDN Web Docs. URL: <https://developer.mozilla.org/> (дата звернення: 16.04.2024).
9. ECMA-262 – Ecma International. Ecma International. URL: <https://ecma-international.org/publications-and-standards/standards/ecma-262/> (дата звернення: 15.04.2024).
10. Quick Start – React. React. URL: <https://react.dev/learn> (дата звернення: 15.04.2024).

11. React v16.8: The One With Hooks – React Blog. React – A JavaScript library for building user interfaces. URL: <https://legacy.reactjs.org/blog/2019/02/06/react-v16.8.0.html> (дата звернення: 21.04.2024).
12. Home v6.23.1 | React Router. Home v6.23.1 | React Router. URL: <https://reactrouter.com/en/main> (дата звернення: 30.04.2024).
13. Getting Started. vitejs. URL: <https://vitejs.dev/guide> (дата звернення: 15.04.2024).
14. Introduction | react-i18next documentation. Introduction | react-i18next documentation. URL: <https://react.i18next.com/> (дата звернення: 02.05.2024).
15. Angular. Home • Angular. URL: <https://angular.dev/> (дата звернення: 01.05.2024).
16. Vue.js. Vue.js - The Progressive JavaScript Framework | Vue.js. URL: <https://vuejs.org/> (дата звернення: 01.05.2024).
17. Bootstrap. Bootstrap · The most popular HTML, CSS, and JS library in the world. URL: <https://getbootstrap.com/> (дата звернення: 01.05.2024).
18. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. URL: <https://tailwindcss.com/> (дата звернення: 01.05.2024).
19. Figma: The Collaborative Interface Design Tool. Figma. URL: <https://www.figma.com/> (дата звернення: 15.04.2024).
20. Sublime Text - the sophisticated text editor for code, markup and prose. Sublime Text - Text Editing, Done Right. URL: <https://www.sublimetext.com/> (дата звернення: 15.04.2024).
21. Microsoft. Visual Studio Code – Code Editing. Redefined. Visual Studio Code – Code Editing. Redefined. URL: <https://code.visualstudio.com/> (дата звернення: 15.04.2024).
22. Git. Git. URL: <https://www.git-scm.com/> (дата звернення: 21.04.2024).

23. Бородкіна І. Л., Бородкін Г.О. Web-технології та Web-дизайн: застосування мови HTML для створення електронних ресурсів : навч. посіб. Київ, 2020. 212 с.
24. Мартін Р. С. Чистий код. Створення і рефакторинг за допомогою Agile / пер. з англ. Бондара-Терещенка І. Київ, 2019. 448
25. Мартін Р. С. Чиста архітектура / пер. з англ. Бондара-Терещенка І. Київ, 2019. 368
26. Simpson K. You Don't Know JS Yet: Get Started. Independent publisher, 2020. 143 p.

ДОДАТОК А

Веб-тренажер для багатокористувацької онлайн-гри Mobile Legends

Програмні засоби:

- мова програмування – JavaScript;
- мова стилів – CSS;
- мова розмітки – HTML;
- допоміжна бібліотека – React;
- обробка датасету – Pandas.

```
// Файл main.jsx
import React from 'react'
import App from './App'
import Nana from './guides/Nana/Nana';
import ReactDOM from "react-dom/client";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Navigation from './components/nav/Navigation';

import './index.css';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <BrowserRouter>
      <Navigation/>
      <Routes>
        <Route path="/" index element={<App />}/>
        <Route path="nana" element={<Nana />}/>
      </Routes>
    </BrowserRouter>
  </React.StrictMode>,
);
```

```
// Файл App.jsx
import React, { useState, useEffect } from 'react';
import SearchBar from './components/search-bar/SearchBar';
import RoleSelect from './components/role-select/RoleSelect';
import LaneSelect from './components/lane-select/LaneSelect';
import HeroResults from './components/hero-results/HeroResults';
import Pagination from './components/pagination/Pagination';
import News from './components/news/News';

import { heroData } from './data';

import './App.css';

const App = () => {
  const [heroes, setHeroes] = useState(heroData);
  const [searchName, setSearchName] = useState('');
  const [searchRole, setSearchRole] = useState([]);
  const [searchLane, setSearchLane] = useState([]);
  const [currentPage, setCurrentPage] = useState(1);
  const [heroesPerPage] = useState(12);

  const handleChangeRole = (event) => {
    const { id, checked } = event.target;
    const label = document.getElementById(id).parentElement;
    const container = label.parentElement;
    if (checked) {
      setSearchRole([...searchRole, id]);
      label.classList.add("checked");
      container.classList.add("checked");
    } else {
      setSearchRole(searchRole.filter(role => role !== id));
      label.classList.remove("checked");
      container.classList.remove("checked");
    }
  }
}
```

```
};
```

```
const handleChangeLane = (event) => {
  const { id, checked } = event.target;
  const label = document.getElementById(id).parentElement;
  const container = label.parentElement;
  if (checked) {
    setSearchLane([...searchLane, id]);
    label.classList.add("checked");
    container.classList.add("checked");
  } else {
    setSearchLane(searchLane.filter(lane => lane !== id));
    label.classList.remove("checked");
    container.classList.remove("checked");
  }
};
```

```
const handleSearch = (event) => {
  setSearchName(event.target.value);
};
```

```
const highlightText = (text, searchName) => {
  if (!searchName) return text;
  const parts = text.split(new RegExp(`(${searchName})`, 'gi'));
  return parts.map((part, index) =>
    part.toLowerCase() === searchName.toLowerCase() ? <span key={index}
className="highlight">{part}</span> : part
  );
};
```

```
const indexOfLastHero = currentPage * heroesPerPage;
const indexOfFirstHero = indexOfLastHero - heroesPerPage;
```

```
const paginate = (pageNumber) => setCurrentPage(pageNumber);
```



```

const filteredHeroes = heroes.filter(hero =>
  (searchName === '' || hero.name.toLowerCase().includes(searchName.toLowerCase())) &&
  (searchRole.length === 0 || searchRole.some(role => hero.role.includes(role))) &&
  (searchLane.length === 0 || searchLane.some(lane => hero.lane.includes(lane)))
).sort((a, b) => a.name.localeCompare(b.name));

const currentHeroes = filteredHeroes.slice(indexOfFirstHero, indexOfLastHero);

useEffect(() => {
  setCurrentPage(1);
}, [filteredHeroes.length]);

return (
  <div className="App">
    <section className="hero-select">
      <SearchBar searchName={searchName} handleSearch={handleSearch} />
      <RoleSelect handleChangeRole={handleChangeRole} />
      <HeroResults currentHeroes={currentHeroes} searchName={searchName}
highlightText={highlightText} />
      <Pagination
        heroesPerPage={heroesPerPage}
        totalHeroes={filteredHeroes.length}
        paginate={paginate}
        currentPage={currentPage}
      />
      <LaneSelect handleChangeLane={handleChangeLane} />
    </section>

    <section className="news">
      <News />
    </section>
  </div>
);
};

```

```
export default App;
```

```
// Файл App.css
```

```
:root {
  --white: #fff;
  --black: #16191c;
  --grey-1: #333b44;
}

* {
  -ms-overflow-style: none;
  scrollbar-width: none;
}

*::-webkit-scrollbar {
  display: none;
}

body {
  margin: 0;
  -ms-overflow-style: none;
  scrollbar-width: none;
  background-color: var(--white);
  transition: background-color 0.2s ease-in;
}

.App {
  color: white;
  font-family: sans-serif;
  display: grid;
  grid-template-columns: 3fr 2fr;
  grid-template-rows: 0.2fr 8fr;
  gap: 1vh;
}
```

```
header {  
  grid-row: 1;  
  grid-column: 1/3;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  background-color: var(--grey-1);  
}
```

```
.flex-centered {  
  display: flex;  
  align-items: center;  
  justify-content: center;  
}
```

```
input[type="checkbox"] {  
  display: none;  
}
```

```
.flex-container {  
  display: flex;  
  gap: 1em;  
  flex-wrap: wrap;  
  place-content: center;  
  place-items: center;  
  height: fit-content;  
}
```

```
.label-container {  
  width: 2.5em;  
  height: 2.5em;  
  padding: 0.3em;  
  border-radius: 0.7em;  
  background-color: var(--white);  
}
```

```
border: .2em solid var(--grey-1);
transition: border .2s ease-in, background-color .2s ease-in;
}
```

```
.label-container > label {
  background-color: var(--grey-1);
  transition: border .2s ease-in, background-color .2s ease-in;
}
```

```
.label-container > label.checked {
  background-color: var(--white);
}
```

```
.label-container.checked {
  background-color: var(--grey-1);
}
```

```
label {
  display: inline-block;
  width: inherit;
  height: inherit;

  background-color: var(--white);
  background-position: center;
  background-repeat: no-repeat;

  mask-repeat: no-repeat;
  mask-size: 90%;
  mask-position: center;

  transition: background-color 0.2s ease-in-out;
}
```

```
.highlight {
  background-color: #00B8FF;
```

```
}
```

```
label:hover {
  cursor: pointer;
}
```

```
.hero-select {
  grid-row: 2;
  grid-column: 1;
  margin-left: 10vw;
  display: grid;
  grid-template-columns: 6fr 0.5fr;
  grid-template-rows: 5vw 6vw 6fr 1fr;
  gap: 1em;
}
```

```
.dark-mode {
  color: var(--white);
  background-color: var(--black);
  #toggle {
    background-color: var(--white);
    .toggle-inner {
      background-color: var(--black);
    }
  }
  .label-container > label {
    background-color: var(--white);
  }
  .label-container {
    background-color: var(--grey-1);
    border: .2em solid transparent;
  }
  .label-container > label.checked {
    background-color: var(--grey-1);
  }
}
```

```
.label-container.checked {
  background-color: var(--white);
}
}
```

```
// Файл HeroResults.jsx
```

```
import React from 'react';
```

```
import { Outlet, Link } from 'react-router-dom';
```

```
import './HeroResults.css';
```

```
import './HeroCard.css';
```

```
const HeroResults = ({ currentHeroes, searchName, highlightText }) => (
  <div className="HeroResults">
    <div className="flex-container">
      {currentHeroes.map((hero, index) => {
        const HeroCardContent = (
          <div className="HeroCard" key={index} style={{ backgroundImage:
            `url(${hero.image})` }}>

            <div className="BlackLine">
              <p>{highlightText(hero.name, searchName)}</p>
              <img className="role-icon"
src={`src/components/role-select/icons/${hero.role[0].toLowerCase()}-icon.svg`} alt={hero.role} />
              {hero.role[1] !== "" && (
                <img className="role-icon"
src={`src/components/role-select/icons/${hero.role[1].toLowerCase()}-icon.svg`} alt={hero.role} />
              )}
              <img className="lane-icon"
src={`src/components/lane-select/icons/${hero.lane[0].toLowerCase()}-icon.svg`} alt={hero.lane} />
              {hero.lane[1] !== "" && (
                <img className="lane-icon"
src={`src/components/lane-select/icons/${hero.lane[1].toLowerCase()}-icon.svg`} alt={hero.lane} />
              )}
            </div>
```

```

    </div>
  );

  return hero.guide ? (
    <Link to={`/${hero.name.toLowerCase()}`} key={index}>
      {HeroCardContent}
    </Link>
  ) : (
    <div key={index}>{HeroCardContent}</div>
  );
})
}
</div>
</div>
);

```

```
export default HeroResults;
```

```
// Файл HeroResults.css
```

```

.HeroResults {
  grid-column: 1;
  grid-row: 3 / 4;
  display: flex;
  gap: 1em;
  flex-wrap: wrap;
  place-items: center;
  place-content: center;
  height: fit-content;
  flex-direction: column;
  max-height: 100vh;
}

```

```
// Файл HeroCard.css
```

```
.HeroCard {
  aspect-ratio: 9 / 12;
  height: 12em;
  display: flex;
  align-items: center;
  justify-content: flex-end;
  flex-direction: column;
  color: #fff;
  border-radius: .5em;
  background-repeat: no-repeat;
  background-size: cover;
  background-color: #272E36;
  transition: all .1s ease-in-out;
  box-sizing: border-box;
  box-shadow: inset 0 0 300em 200em rgba(21, 21, 24, 0.5);
  outline: .1em solid #333B44;
}
```

```
.HeroCard:hover {
  cursor: pointer;
  transform: scale(110%);
  box-shadow: none;
  outline: .1em solid #00B8FF;
}
```

```
.BlackLine {
  background-color: #fff;
  color: #000;
  height: 2em;
  width: 100%;
  border-radius: 0 0 .5em .5em;
  transition: all .1s ease-in-out;
  display: flex;
  align-items: center;
  justify-content: center;
```



```
}
```

```
.BlackLine > p {
  transition: all .1s ease-in-out;
  text-wrap: nowrap;
}
```

```
.role-icon {
  width: 0;
  opacity: 0;
  width: 0;
  background-size: contain;
  overflow: hidden;
  transition: all .1s ease-in-out;
  filter: brightness(0) saturate(100%) invert(66%) sepia(39%) saturate(235%)
hue-rotate(171deg) brightness(93%) contrast(101%);
}
```

```
.lane-icon {
  width: 0;
  opacity: 0;
  width: 0;
  background-size: contain;
  overflow: hidden;
  transition: all .1s ease-in-out;
  filter: brightness(0) saturate(100%) invert(49%) sepia(76%) saturate(468%)
hue-rotate(359deg) brightness(96%) contrast(89%);
}
```

```
.HeroCard:hover > .BlackLine > .role-icon, .HeroCard:hover > .BlackLine > .lane-icon {
  width: 1em;
  opacity: 1;
  transition: all .1s ease-in-out;
  margin-left: .2em;
}
```

```
}
```

```
.hidden {  
  opacity: 0;  
  transition: all .1s ease-in-out;  
  display: none;  
  width: 2em;  
  height: 2em;  
}
```

```
.dark-mode {  
  .HeroCard {  
    outline: 1.5px solid #333B44;  
  }  
  .HeroCard:hover {  
    outline: 1.5px solid #fff;  
  }  
  .BlackLine {  
    background-color: #333B44;  
    color: #fff;  
  }  
}
```

ДОДАТОК Б

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СУЧАСНІ ПРОБЛЕМИ НАУКОВОГО ЗАБЕЗПЕЧЕННЯ ЕНЕРГЕТИКИ

Матеріали XXI Міжнародної
науково-практичної конференції
молодих вчених і студентів
м. Київ, 23–26 квітня 2024 року

ТОМ 2



Київ- 2024

Сервіс пошуку закладів харчування за смаковими вподобаннями користувача.	274
<i>ГУРИН І.А., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Тихоход В.О.</i>	
Система обліку й аналізу ефективності роботи підприємства.	276
<i>ДЕМИДОВИЧ Д.П., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Кублій Л.І.</i>	
Обмін інформацією системи Odoo з месенджером Telegram.	278
<i>ЗЕЛІНСЬКИЙ В.Я., бакалаврант</i>	
<i>Керівник - проф., д.т.н. Шушура О.М.</i>	
Система генерації мультимедійного контенту з використанням штучного інтелекту.	280
<i>КОНДРАШІН О.М., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Тихоход В.О.</i>	
Веб-тренажер для багатокористувацької онлайн-гри Mobile Legends.	282
<i>КОРДЯК Р.М., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Кублій Л.І.</i>	
Розробка лабораторного стенду для дослідження застосування методів ідентифікації пристроїв інтернету речей на базі архітектури цифрових об'єктів.	284
<i>КОРЧАН В.М., аспірант</i>	
<i>Керівник - проф., к.т.н. Морозова І.В.</i>	
Застосування великих мовних моделей для забезпечення якості освітньої діяльності кафедри.	286
<i>СУГУЛОВ Є.С., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Коваль О.В.</i>	
Веб-застосунок для організації діяльності спортивної школи.	289
<i>ХОЛОДНИЦЬКА А.Ю., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Кублій Л.І.</i>	
Веб-додаток обліку товарів в складському приміщені.	291
<i>ХОРОШИХ О.Л., бакалаврант</i>	
<i>Керівник - доц., к.е.н. Сегеда І.В.</i>	
Розробка сервісу веб застосунку забезпечення якості освітньої діяльності кафедри з використанням мовних моделей на базі апаратних засобів кафедри.	293
<i>ЧУКІН С.В., бакалаврант</i>	
<i>Керівник - асист. Голець В.О.</i>	
Розв'язання гравітаційної задачі N тіл за допомогою швидкого методу мультиполів.	295
<i>ЮРИК Д.М., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Лабжинський В.А.</i>	
Засоби генерації текстових математичних задач шкільної програми з використанням Windows form.	298
<i>ЯКИМЕНКО Д.В., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Кублій Л.І.</i>	
Веб-система пошуку та рейтингування закладів навчальних закладів.	300
<i>ЯЦЕНКО М.І., бакалаврант</i>	
<i>Керівник - доц., к.т.н. Тихоход В.О.</i>	

УДК 004.42:379.8

¹ Бакалаврант 4 курсу Кордяк Р.М.¹ Доц., к.т.н. Кублій Л.І.<https://scholar.google.com.ua/citations?user=rXA1eMIAAAAJ&hl=uk>¹ КПП ім. Ігоря Сікорського

ВЕБ-ТРЕНАЖЕР ДЛЯ БАГАТОКОРИСТУВАЦЬКОЇ ОНЛАЙН-ГРИ MOBILE LEGENDS

Постановка проблеми та її актуальність. Веб-тренажер для багатокористувацької гри, в даному випадку Mobile Legends, — це веб-сайт, у якому збирається корисна інформація про кожного героя гри, що дає можливість гравцям з легкістю вивчати нюанси ігрового процесу. Саме для такої гри це дуже актуально, адже в неї щоденно заходять більше п'яти мільйонів людей, а рекомендації щодо того, як грати в цю гру, практично відсутні. Звичайно, існує кілька сервісів, які поверхнево характеризують найпопулярніших героїв, проте, в більшості випадків, як і дизайн, так і сама інформація — застарілі. Дане рішення пропонує як сучасний і зручний дизайн, так і постійні оновлення інформації відповідно до змін у грі.

Аналіз останніх досліджень. Під час розробки веб-тренажера для гри Mobile Legends було проаналізовано такі веб-сайти, як OP.GG [1] і KQM [2]. Веб-сайт OP.GG збирає інформацію про найстабільніші і найпопулярніші способи гри в League of Legends [3]. Ця платформа має сучасний вигляд і зручний зрозумілий інтерфейс, а також дає користувачам максимально корисну інформацію. Багато гравців з усього світу діляться своїми даними, щоб сформувати якомога більшу вибірку, яка в подальшому обробляється програмою для визначення найефективнішого стилю гри за того чи іншого героя. Тут була запозичена сама ідея пропонувати користувачеві інформацію, підкріплену математичними обчисленнями або статистичним аналізом. Мета веб-застосунку KQM (KeqingMains) — публікувати дослідження, пов'язані з персонажами відеогри Genshin Impact. На сайті є стрічка новин, ідею про яку було запозичено саме звідти. Також на KQM постійно публікуються оновлення, оскільки гра постійно змінюється, а отже, і посібники потрібно оновлювати також. Таким чином, серед запозичень є меню героїв з їхніми зображеннями, стандартизація вигляду посібників про героїв, блок новин.

Формулювання мети. Основна мета й особливість розробки веб-тренажера — можливість для досвідчених гравців ділитися власними порадами. Також веб-застосунок має відрізнятися унікальним дизайном і функціоналом.

Основна частина. Користувач створеного веб-тренажера може авторизуватися на сайті, після чого йому стане доступна опція “створити”, яка дозволить почати писати нову публікацію. Якщо адміністрація затверджує її, то на головній веб-сторінці з'являється повідомлення про те, що розділ певного героя було оновлено. Це заохочує відвідувачів проглядати нові публікації та дізнаватися щось нове для себе.

Відвідувач сайту може вільно переглядати будь-яку розміщену інформацію без реєстрації за допомогою зрозумілого інтерфейсу. Першим, що бачить користувач — блок новин, який інформує про оновлення старих посібників або публікацію нових. Кожна новина через деякий час змінюється іншою за допомогою анімацій. Після блоку новин розміщено основну частину веб-застосунку — вибір героїв. Тут користувач може побачити картки з іменами та зображеннями всіх доступних героїв, а також коротку інформацію про їхню роль. Список можна відсортувати за роллю героя і за найпопулярнішою позицією, на якій грає герой. Список оновлюється динамічно, без перезавантаження сторінки. Усі дані про героїв беруться з бази даних, яка поки що локально зберігається в JSON-файлі. На сторінці конкретного героя розміщена більш детальна інформація про нього, а також

вказівки щодо його ігрового процесу. Інформація щодо базових характеристик героя береться з таблиці, яка міститься в базі даних. Далі йдуть пояснення про вибір правильних предметів з підрахунком ефективності, інформація про сильні і слабкі сторони героя, поради щодо бою з найбільш популярними суперниками для цього героя. У кінці сторінки є висновки про героя і деякі нюанси, котрі не вдалося вмістити в основну частину.

Зареєстрованим користувачам надається можливість повідомляти про помилки різного роду: помилки, пов'язані з написанням тексту, помилки в обчисленнях, помилкові вхідні дані. Розроблений застосунок також є відкритим програмним забезпеченням (open-source software). Це означає, що програмний код є доступним для редагування, і всі охочі мають змогу допомогти з розробкою шляхом редагування коду.

Після дослідження сучасних тенденцій зовнішнього вигляду застосунків було прийнято рішення, що темна тема буде темою за замовчуванням. Проте є можливість зміни теми з темної на світлу.

Застосунок має дві робочі мови — українську та англійську. Хоч англomовних ресурсів у даній ніші вже є досить багато, проте, як було зазначено вище, вони аж ніяк не найкращі. Якщо ж мова йде про українськомовні ресурси, то аналогів даному продуктові просто не існує. Варто зазначити, що гра популярна в Україні до такої міри, що збірна України перемогла на Чемпіонаті Європи з Mobile Legends: Bang Bang у 2022 році [4]. Розроблений програмний продукт повинен стати як сильним конкурентом для подібних зарубіжних сервісів, так і цікавою та зручною новинкою на українському ринку. Пріоритизується направленість на зручність користування, щоб застосунок був зрозумілим навіть для найбільш необізнаного користувача.

Для розробки програмного застосунку використано мову розмітки HTML разом з мовою стилів CSS і мовою програмування JavaScript. Стандартної об'єктної моделі документа для веб-розробки вистачає для проєкту такого масштабу, і такі доповнення, як TypeScript або Tailwind CSS, будуть тільки перевантажувати процес розробки. Основною бібліотекою для розміщення елементів на сторінці є React — бібліотека JavaScript, яка дає можливість будувати інтерактивні інтерфейси. Головною перевагою обраної бібліотеки є швидкодія. Завдяки модульній системі побудови інтерфейсів веб-застосунків розміщується на одній сторінці, що дає можливість не завантажувати нові сторінки при переході між розділами. Для серверної частини застосовується інструмент Vite, за допомогою якого можна швидко налаштовувати сервер і бачити всі зміни в реальному часі. В якості системи керування базами даних використано MongoDB. Дані будуть зберігатися у вигляді таблиць і за потреби будуть використовуватися під час завантаження сторінок.

Висновки. Після ретельного аналізу потреб гравців і вимог до сучасних ігрових платформ, веб-тренажер для гри Mobile Legends став відповіддю на виклик забезпечення доступної та актуальної інформації про кожного героя гри. Інтерфейс, що поєднує зручність, естетику та інтуїтивність, разом із можливістю користувачів ділитися власними порадами, створює середовище для покращення навичок у грі та сприяє активному обміну досвідом серед геймерської спільноти. З урахуванням широкого загального інтересу до гри і різноманіття користувачів з різних країн, застосунок підтримує вибір мови інтерфейсу і надає користувачам максимум корисної та актуальної інформації у зручному форматі. Використання розробленого веб-тренажера покращить користувацький досвід у сегменті веб-платформ для гравців Mobile Legends, сприятиме вдосконаленню ігрового процесу і досягненню нових вершин в Mobile Legends.

Перелік посилань:

1. OP.GG. URL: <https://www.op.gg>.
2. Genshin Impact Guides and Theorycrafting -- KQM. URL: <https://keqingmains.com>.
3. League of Legends. URL: <https://www.leagueoflegends.com>.
4. Збірна України виграла Чемпіонат Європи з Mobile Legends. 04.10.2022. URL: <https://players.com.ua/news/zbirna-ukrayini-vigrala-chempionat-yevropi-z-mobile-legends>.

ДОДАТОК В

Фрагменти програмного коду

```
const Navigation = () => {
  return (
    <nav className="Navigation">
      <ThemeSwitch/>
      <a href="">Гайди</a>
      <a href="">Соціальні мережі</a>
      <a href="">Зворотній зв'язок</a>
      <LangSwitch/>
    </nav>
  )
}
```

Рисунок В1 – Реалізація компонента Navigation

```
return (
  <div className="News flex-centered">
    <h2>{t('whatsNew')}</h2>
    <div className="news-container">
      <Link to={` /nana`} >
        <div className="news-entry">
          <div className="cover"></div>
          <div className="article flex-centered">
            <h3 className="truncate">{t('nanaNewsHeader')}</h3>
            <p className="truncate">{t('nanaNewsDesc')}</p>
          </div>
        </div>
      </Link>
    </div>
  </div>
)
```

Рисунок В2 – Програмна реалізація компонента News

```

const HeroResults = ({ currentHeroes, searchName, highlightText }) => (
  <div className="HeroResults">
    <div className="flex-container">
      {currentHeroes.map((hero, index) => {
        const HeroCardContent = (
          <div className="HeroCard" key={index} style={{ backgroundImage: `url(${hero.image})` }}>
            <div className="BlackLine">
              <p>{highlightText(hero.name, searchName)}</p>
              <img className="role-icon" src={`src/components/role-select/icons/${hero.role[0].toLowerCase()}-icon.svg`} alt={hero.role} />
              {hero.role[1] !== "" && (
                <img className="role-icon" src={`src/components/role-select/icons/${hero.role[1].toLowerCase()}-icon.svg`} alt={hero.role} />
              )}
              <img className="lane-icon" src={`src/components/lane-select/icons/${hero.lane[0].toLowerCase()}-icon.svg`} alt={hero.lane} />
              {hero.lane[1] !== "" && (
                <img className="lane-icon" src={`src/components/lane-select/icons/${hero.lane[1].toLowerCase()}-icon.svg`} alt={hero.lane} />
              )}
            </div>
          </div>
        );
        return hero.guide ? (
          <Link to={`/${hero.name.toLowerCase()}`} key={index}>
            {HeroCardContent}
          </Link>
        ) : (
          <div key={index}>{HeroCardContent}</div>
        );
      })}
    </div>
  </div>
);

```

Рисунок В3 – Код компонента HeroResults

```

const RoleSelect = ({ handleChangeRole }) => (
  <form className="RoleSelect">
    {[ 'Tank', 'Fighter', 'Assassin', 'Marksman', 'Mage', 'Support' ].map(role => (
      <div className="label-container" key={role}>
        <label className={`label-${role.toLowerCase()}`}>
          <input onChange={handleChangeRole} type="checkbox" id={role} />
        </label>
      </div>
    ))}
  </form>
);

```

Рисунок В4 – Реалізація компонента RoleSelect


```

<form className="LaneSelect">
  {[ 'Gold', 'Middle', 'Experience', 'Jungle', 'Roam' ].map(lane => (
    <div className="label-container" key={lane}>
      <label className={`label-${lane.toLowerCase()}`>
        <input onChange={handleChangeLane} type="checkbox" id={lane} />
      </label>
    </div>
  )]}
</form>

```

Рисунок В5 – Реалізація компонента LaneSelect

```

const Pagination = ({ heroesPerPage, totalHeroes, paginate, currentPage }) => (
  <div className="pagination flex-centered">
    {totalHeroes > heroesPerPage && (
      <ul className="pageNumbers flex-centered">
        {Array.from({ length: Math.ceil(totalHeroes / heroesPerPage) }, (_, i) => (
          <li key={i} className={`pageNumber ${currentPage === i + 1 ? 'activePage' : ''}`} onClick={() => paginate(i + 1)}>
            {i + 1}
          </li>
        ))}
      </ul>
    )}
  </div>
);

```

Рисунок В6 – Реалізація компонента Pagination

```

return (
  <div
    id="toggle"
    onClick={() => darkMode === false ? setDarkMode(true) : setDarkMode(false)}
  >
    <div className="toggle-inner"/>
  </div>
)

```

Рисунок В7 – Реалізація компонента ThemeSwitch

```

const LangSwitch = () => {
  const { t, i18n } = useTranslation();
  const [currentLanguage, setCurrentLanguage] = useState(i18n.language);
  const toggleLanguage = () => {
    const newLanguage = currentLanguage === 'en' ? 'uk' : 'en';
    setCurrentLanguage(newLanguage);
    i18n.changeLanguage(newLanguage);
  };
  return (
    <button className="change-lang" onClick={toggleLanguage}>
      <img className="flag-icon" src={currentLanguage === 'en' ? '/src/icons/en_flag.png' : '/src/icons/uk_flag.png'} alt="Language Flag" />
    </button>
  )
}

```

Рисунок В8 – Реалізація компонента LangSwitch

```

return (
  <div className="Nana">
    { /* {profile} */ }

    <div className="table-of-contents">
      <h4>{t('tableOfContents')}</h4>
      <ul className='ul-toc'>
        <li>
          <a href="#gameplay">{t('gameplay')}</a>
        </li>
        <li>
          <a href="#items">{t('items')}</a>
        </li>
        <li>
          <a href="#synergy">{t('synergy')}</a>
        </li>
      </ul>
    </div>

    <section className="profile">
      <div className="pfp"></div>
      <div className="description">
        <h2>Nana</h2>
        <p>{t('nanaLore')}</p>
        <div className="gameplay-info">
          <div className="cont">
            <div id="primary-role"></div>
            <label htmlFor="primary-role">{t('primaryRole')}</label>
          </div>
          <div className="cont">
            <div id="secondary-role"></div>
            <label htmlFor="secondary-role">{t('secondaryRole')}</label>
          </div>
        </div>
      </div>
    </section>
  </div>
)

```

Рисунок В9 – Фрагмент компонента Nana

```

const handleChangeRole = (event) => {
  const { id, checked } = event.target;
  const label = document.getElementById(id).parentElement;
  const container = label.parentElement;
  if (checked) {
    setSearchRole([...searchRole, id]);
    label.classList.add("checked");
    container.classList.add("checked");
  } else {
    setSearchRole(searchRole.filter(role => role !== id));
    label.classList.remove("checked");
    container.classList.remove("checked");
  }
};

```

Рисунок В10 – Функція handleChangeRole

```
const handleChangeLane = (event) => {
  const { id, checked } = event.target;
  const label = document.getElementById(id).parentElement;
  const container = label.parentElement;
  if (checked) {
    setSearchLane([...searchLane, id]);
    label.classList.add("checked");
    container.classList.add("checked");
  } else {
    setSearchLane(searchLane.filter(lane => lane !== id));
    label.classList.remove("checked");
    container.classList.remove("checked");
  }
};
```

Рисунок В11 – Функція handleChangeLane

```
const handleSearch = (event) => {
  setSearchName(event.target.value);
};
```

Рисунок В12 – Функція handleSearch

```
const highlightText = (text, searchName) => {
  if (!searchName) return text;
  const parts = text.split(new RegExp(`(${searchName})`, 'gi'));
  return parts.map((part, index) =>
    part.toLowerCase() === searchName.toLowerCase() ? <span key={index} className="highlight">{part}</span> : part
  );
};
```

Рисунок В13 – Функція highlightText

```
const paginate = (pageNumber) => setCurrentPage(pageNumber);
```

Рисунок В14 – Функція paginate

```

const indexOfLastHero = currentPage * heroesPerPage;
const indexOfFirstHero = indexOfLastHero - heroesPerPage;

const filteredHeroes = heroes.filter(hero =>
  (searchName === '' || hero.name.toLowerCase().includes(searchName.toLowerCase())) &&
  (searchRole.length === 0 || searchRole.some(role => hero.role.includes(role))) &&
  (searchLane.length === 0 || searchLane.some(lane => hero.lane.includes(lane)))
).sort((a, b) => a.name.localeCompare(b.name));

const currentHeroes = filteredHeroes.slice(indexOfFirstHero, indexOfLastHero);

useEffect(() => {
  setCurrentPage(1);
}, [filteredHeroes.length]);

```

Рисунок В15 – Фрагмент коду з фільтрацією

```

const handleClickPassive = () => {
  setVisibleDescription(visibleDescription === 'passive' ? null : 'passive');
};

const handleClickAbility1 = () => {
  setVisibleDescription(visibleDescription === 'ability1' ? null : 'ability1');
};

const handleClickAbility2 = () => {
  setVisibleDescription(visibleDescription === 'ability2' ? null : 'ability2');
};

const handleClickAbility3 = () => {
  setVisibleDescription(visibleDescription === 'ability3' ? null : 'ability3');
};

```

Рисунок В16 – Фрагмент коду, який відповідає за відображення опису навичок