Вінницький національний технічний університет

Факультет інформаційних технологій та комп`ютерної інженерії

Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: Розробка методу навчання в ігрових ситуаціях і програмного засобу для тренування кіберспортсменів

Виконав: студент IV курсу, групи 3ПІ-20б

спеціальності

121 – Інженерія програмного забезпечення
(шифр і назва напряму підготовки, спеціальності)

Двойнос Ілля Іванович
(прізвище та ініціали)

Керівник: к.т.н., доцент каф. ПЗ Ракитянська Г.Б.
(прізвище та ініціали)

Рецензент: к.т.н., доцент каф. ЗІ Баришев Ю.В.
(прізвище та ініціали)

Допущено до захисту

Зав. кафедри Романюк О. Н.

«10» червня 2024 р.

ВНТУ – 2024

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший (бакалаврський)
Галузь знань 12 – «Інформаційні технології»
Спеціальність 121 – «Інженерія програмного забезпечення»
Освітньо-професійна програма – «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор
О. Н. Романюк _____
«13» березня 2024 р.

З А В Д А Н Н Я

НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Двойносу Іллі Івановичу

1. Тема роботи – розробка методу навчання в ігрових ситуаціях і програмного засобу для тренування кіберспортсменів.

Керівник роботи: Ракитянська Ганна Борисівна, к.т.н., доцент, затверджені наказом вищого навчального закладу від 11 березня 2024 р. № 80.

2. Строк подання студентом роботи 3 червня 2024.

3. Вихідні дані до роботи: список дисциплін, список керівництв та завдань, сторінки керівництв і завдань, з можливістю їх додавання до списку прочитаних або виконаних.

4. Зміст текстової частини: вступ; аналіз стану програмних застосунків для тренування кіберспортсменів; аналіз аналогів; аналіз методів розв'язання задачі; постановка задач; розробка методу навчання в ігрових ситуаціях; аналіз вхідних даних; розробка інтерфейсу користувача; розробка моделі роботи програмного застосунку; розробка алгоритму роботи програмного застосунку; розробка модулів керівництв та завдань; розробка модулів авторизації та особистого кабінету; тестування розробленого програмного застосунку; створення інструкції користувача.

5. Перелік ілюстративного матеріалу: аналоги програмного застосунку; інтерфейс програмного забезпечення; модель роботи застосунку; алгоритм роботи застосунку; програмні модулі; результати тестування.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата			
		Завдання видав		Завдання прийняв	
1-4	Ракитянська Г.Б., к.т.н., доцент каф. ПЗ	13.03.24		03.06.24	

7. Дата видачі завдання 13 березня 2024

КАЛЕНДАРНИЙ ПЛАН

№ з\п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз підходів до методів навчання в ігрових ситуаціях та програмних застосунків для тренування кіберспортсменів	13.03.24 – 22.03.24	виконано
2	Розробка методу навчання в ігрових ситуаціях	13.03.24 – 22.03.24	виконано
3	Розробка алгоритмів роботи програмного забезпечення	25.03.24 – 19.04.24	виконано
4	Розробка програмного забезпечення	22.04.24 – 10.05.24	виконано
5	Тестування програмного забезпечення	13.05.24 – 24.05.24	виконано
6	Оформлення матеріалів до захисту БДР	27.05.24 – 03.06.24	виконано

Студент _____ **І. І. Двойнос**
(підпис) (ініціали та прізвище)

Керівник бакалаврської дипломної роботи _____ **Г. Б. Ракитянська**
(підпис) (ініціали та прізвище)

АНОТАЦІЯ

Бакалаврська дипломна робота складається зі 58 сторінок формату А4, на яких є 34 рисунки, 3 таблиці, список використаних джерел містить 18 найменувань.

У даній бакалаврській дипломній роботі розроблено метод для навчання в ігрових дисциплінах та програмний застосунок для навчання кіберспортсменів. Програмний застосунок призначений для платформ, які містять веб-браузери, та буде здатний надавати користувачам можливість набути нових навичок та знань, та поліпшити існуючі за рахунок навчальних завдань. Програмний застосунок розроблений для освітніх цілей.

Програмний застосунок розроблено з використанням мов HTML, JavaScript, PHP, CSS та середовища VisualStudio. Під час процесу розробки були використані бібліотеки BootsTrap та jQuery для динамічного відображення елементів DOM.

ABSTRACT

The bachelor thesis consists of 58 pages of A4 format, on which there are 34 figures, 3 tables, the list of used sources contains 18 items.

In this bachelor’s thesis was developed a method for education in game disciplines and program application for training cyber-athletes. The program application is designed for platforms that include web browsers and will be able to provide users with the opportunity to acquire new skills and knowledge, and improve existing ones through educational tasks. The program application is designed for educational purposes.

The program application is developed using HTML, JavaScript, PHP, CSS and VisualStudio environment. Libraries were used during the development process. During development process BootsTrap and jQuery libraries were used to dynamically display elements of DOM.

ЗМІСТ

ВСТУП 3
1 АНАЛІЗ РОЗВИТКУ МЕТОДІВ НАВЧАННЯ В ІГРОВИХ СИТУАЦІЯХ І ПРОГРАМНИХ ЗАСТОСУНКІВ ДЛЯ ТРЕНУВАННЯ КІБЕРСПОРТСМЕНІВ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ 7
1.1 Аналіз стану програмних застосунків для тренування кіберспортсменів 7
1.2 Порівняльний аналіз аналогів 9
1.3 Аналіз методів розв’язання задачі 14
1.4 Постановка задач бакалаврської дипломної роботи 15
1.5 Висновки 16
2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ .. 17
2.1 Розробка методу навчання в ігрових ситуаціях 17
2.2 Розробка моделі навчання в ігрових ситуаціях. 18
2.3 Аналіз вхідних даних системи 20
2.4 Розробка інтерфейсу веб-застосунку 21
2.5 Розробка моделі веб-застосунку 28
2.6 Розробка алгоритмів роботи веб-застосунку 29
2.7 Висновки 32
3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ ТРЕНУВАННЯ КІБЕРСПОРТСМЕНІВ 33
3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення 33
3.2 Розробка модулів відображення списку керівництв та завдань та створення їхніх сторінок 34
3.3 Розробка модулів авторизації та відображення особистого кабінету 36
3.4 Висновки 41
4 ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКУ 42
4.1 Аналіз методів тестування програмних застосунків 42
4.2 Тестування веб-застосунку 43
4.3 Розробка інструкції користувача 51
4.4 Висновки 53
ВИСНОВКИ 55
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 57
ДОДАТКИ 59
Додаток А – Технічне завдання 60
Додаток Б – Протокол перевірки на плагіат 64
Додаток В – Лістинг програми 65
Додаток Г – Ілюстративна частина 81

ВСТУП

Обґрунтування вибору теми дослідження. Ігрові дисципліни (або кібердисципліни) – це відеоігри, за якими проходять масштабні турніри з великим призовим фондом [1]. З кожним роком вони набувають все більшого визнання, як спортивні дисципліни. Вони приваблюють людей, які шукають унікальний та захоплюючий досвід, який дозволить їм позмагатися проти інших користувачів. Однак, часто освоєння найпопулярніших дисциплін є достатньо складним завданням, адже найкращі гравці постійно тренуються та покращують навички, тому для звичайних користувачів досягнення високого рівня є складним завданням. Однак існують способи, як пришвидшити цей процес, а саме детальні керівництва, які пояснюють особливості та механіки ігрових дисциплін, та завдання, які покликані допомогти набути нові навички, або покращити існуючі.

Для пошуку цих шляхів користувачі відправляються в мережу інтернет за допомогою веб-браузера, й заходять до різних веб-ресурсів. Дані веб-ресурси є різноманітними веб-застосунками, які й можуть містити необхідну інформацію [2]. Їхня розробка є складним завданням, оскільки необхідно враховувати велику кількість фаткорів, таких як наповнення сайту, технічна частина, простота пошуку необхідної інформації, захист особистих даних та інші. Достатньо часто розробники забувають про ці базові принципи, що призводить до незадоволення відвідувачів.

Також, веб-застосунки часто містять нав’язливий платний функціонал, або наповнені рекламою, яка відволікає відвідувачів від процесу навчання, або концентруються не на процесі навчання та покращення навичок, а на здобутку локальної валюти, яку можна потім обмінювати на ігрові предмети, змінюючи пріорітети гравців. Крім того, дані ресурси не містять категоризовану інформацію про найпопулярніші ігрові дисципліни, а лише про одну конкретну, а також не містять навчальні завдання, що змушує користувача виконувати зайві дії у вигляді проведення додаткового пошуку в мережі інтернет. Саме тому актуальною є

розробка методу навчання в ігрових ситуаціях та програмного застосунку для тренування кіберспортсменів.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета та завдання дослідження. Мета дипломної роботи полягає у підвищенні якості навчання кіберспортсменів за рахунок створення методу і програмного засобу навчання в ігрових ситуаціях, що базуються на керівництвах з виконання завдань, які дозволять швидше адаптовуватися залежно від ігрових ситуацій і будуть враховувати найактуальнішу інформацію з ігрових дисциплін для спортсменів різних рівнів підготовки.

Робота передбачає розробку наступних модулів: модуль відображення керівництв та завдань, модуль авторизації та відображення особистого кабінету.

Задачі дослідження:

- провести аналіз стану методів навчання в ігрових ситуаціях і програмних застосунків для навчання кіберспортсменів;
- запропонувати новий метод навчання в ігрових ситуаціях, програмний застосунок-реалізацію;
- розробити модель програмного застосунку;
- розробка наступних модулів: модуль відображення керівництв та завдань, модуль авторизації і відображення особистого кабінету;
- провести тестування розроблених алгоритмів та програмного застосунку.

Об'єкт дослідження – процес розробки програмного застосунку для тренування кіберспортсменів.

Предмет дослідження – метод навчання в ігрових ситуаціях та програмний застосунок для тренування кіберспортсменів.

Методи дослідження. У процесі досліджень використовувались:

- методи та засоби реалізації веб-застосунків мовами HTML, CSS і PHP з використанням BootStrap для спрощення побудови і стилізації DOM[3];
- методи динамічного оновлення елементів за рахунок використання бібліотек jQuery мови JavaScript;
- методи тестування реалізованої системи та алгоритмів.

Наукова новизна отриманих результатів:

1. Подальшого розвитку набув метод навчання в ігрових ситуаціях, який на відміну від існуючих реалізацій базуватиметься на математичній моделі, що враховує стратегії гри у вигляді керівництв до виконання завдань. Це сприяє швидкому засвоєнню нових ігрових стратегій та тактик, а також оптимізує загальний прогрес кіберспортсменів у змаганнях.
2. Подальшого розвитку набула програмна модель навчання в ігрових ситуаціях для тренування кіберспортсменів. Дана модель дозволяє створити унікальне програмне забезпечення, яке забезпечить персоналізоване навчання та тренування кіберспортсменів на основі індивідуальних навичок і потреб. Адаптивна модель навчання дозволяє точно вимірювати та аналізувати результати кожного тренувального сеансу, що допомагає виявляти слабкі місця гравців та робити висновки щодо їх покращення.

Практична цінність отриманих результатів. Практична цінність отриманих результатів полягає в розробці вебзастосунку, який здатний надавати користувачам можливість ефективно тренуватися безпосередньо в ігрових ситуаціях, читаючи керівництва та виконуючи завдання, використовуючи браузер. Програмні модулі, розроблені в роботі, можуть бути використані розробниками для створення нових подібних систем, для розширення функціоналу вебзастосунку або алгоритму тренування в цілому та користувачами для тренування та навчання в ігрових ситуаціях.

Особистий внесок здобувача. Усі наукові результати, що викладені у бакалаврській дипломній роботі, отримані особисто автором.

Публікації. За тематикою дослідження надіслано на опублікування 1 наукову працю у збірнику матеріалів науково-технічної конференції “Молодь в науці: дослідження, проблеми, перспективи (МН-2024)”.

Структура та обсяг роботи. Бакалаврська дипломна робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел, що містить 18 найменувань, 4 додатки. Робота містить 34 ілюстрації та 3 таблиці.

1 АНАЛІЗ РОЗВИТКУ МЕТОДІВ НАВЧАННЯ В ІГРОВИХ СИТУАЦІЯХ І ПРОГРАМНИХ ЗАСТОСУНКІВ ДЛЯ ТРЕНУВАННЯ КІБЕРСПОРТСМЕНІВ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану програмних застосунків для тренування кіберспортсменів

Індустрія відеоігор або індустрія інтерактивних розваг – економічний сектор, пов'язаний з розробкою, просуванням та продажем відеоігор. У неї входить велика кількість спеціальностей, за якими працюють тисячі людей по всьому світу [4].

Зі стрімким розвитком ігрової індустрії з’являється усе більше ігрових дисциплін, які здобувають велику популярність. Водночас, гравцям стає усе складніше не просто покращувати свій рівень знань та навичок у них, але й просто підтримувати його на хорошому рівні. Через обмеженість часу вони починають шукати способи, які оптимізують та полегшать цей процес, у різних веб-застосунках.

Останніми роками спостерігається значний ріст попиту на навчальні платформи та тренувальні програми для кіберспортсменів. Такі платформи дозволяють гравцям отримувати доступ до структурованих курсів, аналітики їхньої гри, а також персоналізованих рекомендацій від професійних тренерів. Залучення до процесу навчання елементів штучного інтелекту та машинного навчання також сприяє більш точному аналізу ігрових даних і адаптації навчальних програм під індивідуальні потреби кожного гравця.

Особливу увагу слід приділити використанню аналітичних інструментів, які дозволяють кіберспортсменам відслідковувати свої прогреси у реальному часі. Це включає детальний розбір ігрових сесій, виявлення слабких місць у стратегіях та індивідуальних навичках, а також рекомендації щодо покращення. Наприклад,

платформи можуть надавати аналіз рухів миші, швидкості реакції, точності пострілів та інших важливих аспектів гри, що значно полегшує процес удосконалення навичок.

Спеціалізовані програмні застосунки, які знаходяться у мережі інтернет, мають безліч переваг, на відміну від їхніх фізичних аналогів у вигляді ігрових журналів, таких як легкість пошуку та доступу до інформації, можливість переглянути думку інших користувачів, щодо конкретної теми та багато інших. Однак, навіть враховуючи перераховані переваги, пошук повного об’єму необхідних керівництв та навчальних методів для підвищення рівня навичок є непростим завданням, оскільки їхній автор повинен мати достатньо компетенції у обраній ігровій дисципліні, повинен вміти коротко та чітко подати необхідну інформацію у повному обсязі. Також сам веб-застосунок повинен мати катигоризацію даної інформації. Дані нюанси часто можуть змушусити користувачів одночасно збирати інформацію на багатьох ресурсах, що нівелює суть оптимізації процесу, оскільки час витрачається на додатковий пошук інформації та її складанні до купи. Саме тому, при розробці навчального веб-застосунку, необхідно враховувати дані аспекти [5].

Існує не так багато веб-застосунків, які здатні дотримуватись перелічених принципів, а тих, які спеціалізуються не на одній ігровій дисципліні, а на багатьох найпопулярніших, ще менше.

Отже, розробка веб-застосунку для навчання в ігрових дисциплінах є критично важливим завдання для розвитку кіберспортивної сфери. Використовуючи досвід існуючих веб-ресурсів, які спеціалізуються на керівництвах та завданнях в ігрових дисциплінах, їхні розробники підвищують рівень зацікавленості ігровою сферою та дозволяють гравцям полегшити процес здобуття нових та покращення існуючих навичок та знань. Зважаючи на постійний розвиток ігрової та кіберспортивної сфери, важливо підтримувати стабільний потік нових талановитих гравців, щоб залишатися конкурентноспроможними у них.

1.2 Порівняльний аналіз аналогів

Використання ресурсів з керівництвами та методами тренувань для ігор зростає разом з ростом популярності ігрової індустрії та кіберспортивної сфери. Розробка веб-застосунку для навчання в ігрових дисциплінах може допомогти гравцям отримати нові знання з дисциплін, покращити вміння та зекономити час. До найбільш відомих рішень відносять:

- IGN;
- gamepressure;
- thegamer;
- GamerGains;
- Gamer Sensei.

Одним із прикладів веб-застосунків з керівництвами є IGN (рис. 1.1). Це сайт з новинами ігрової індустрії, який містить також і детальні керівництва новачків, як до однокористувацьких, так і до багатокористувацьких ігрових дисциплін. Однак, ці керівництва складаються ігровими журналістами, які не завжди мають навички найвищого рівня в іграх. Даний ресурс є перевіреним часом та досі користується популярністю гравців з усіх куточків світу.

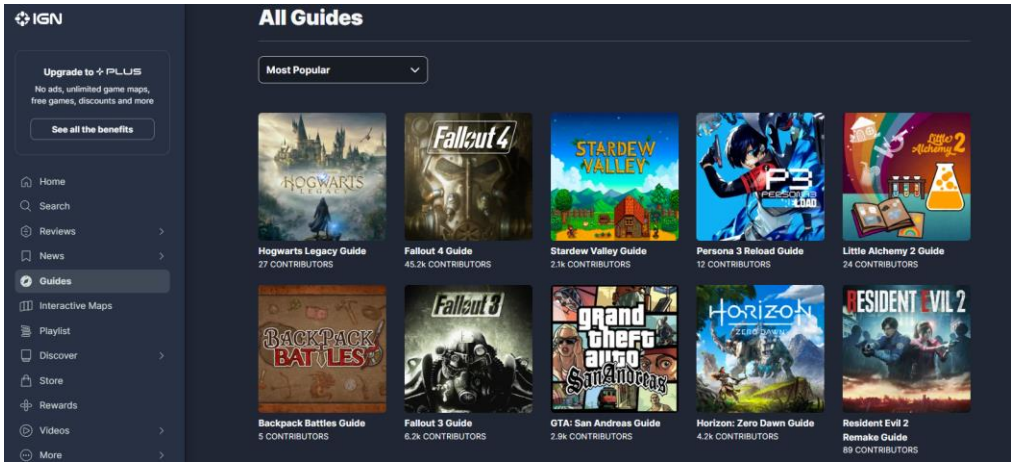


Рисунок 1.1 – Приклад роботи веб-застосунку IGN

Інший приклад – веб-застосунок gamepressure (рис. 1.2). Даний ресурс, окрім керівництв, також містить новини, енциклопедію, відео та інше, як і попередній. Але акцент сайту та його головна мета – це керівництва в ігрових дисциплінах [6]. Крім того, щоб упевнитись, що користувачі обрали керівництво до саме тієї гри, яка їм потрібна, у ньому вказано назву гри, дату її виходу, розробника, та додано медіафайли. Також, кожне керівництво та гра мають свою тему на їхньому форумі, що дозволяє здійснювати обмін думками.

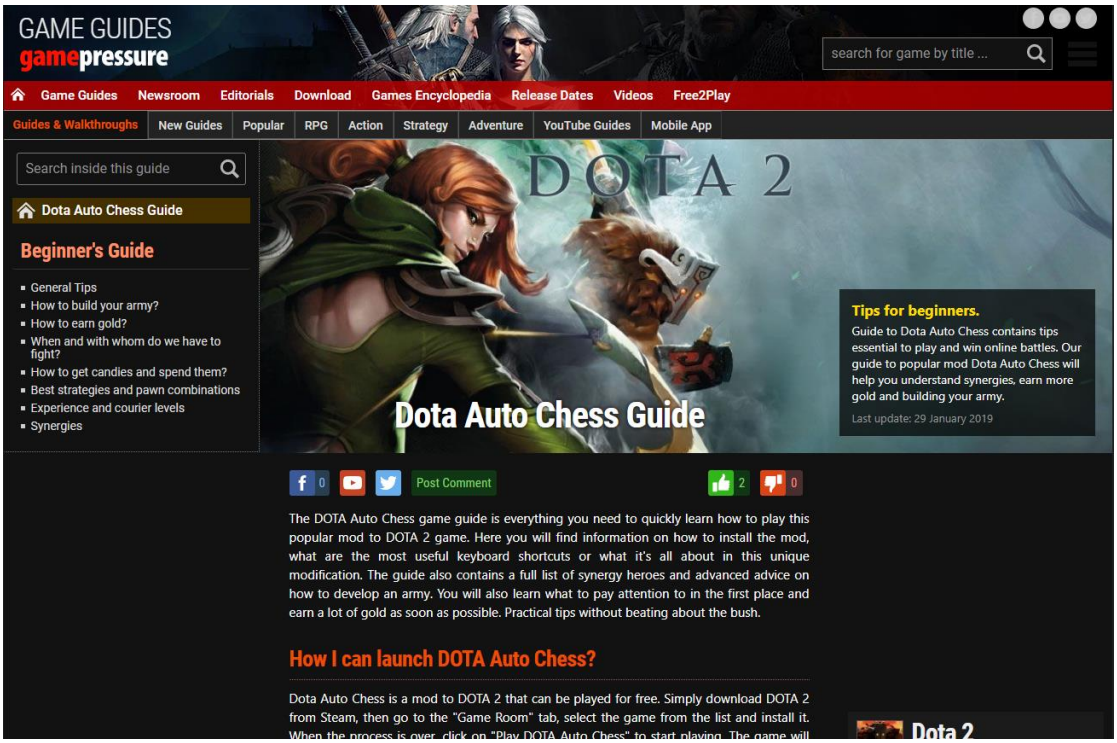


Рисунок 1.2 – Приклад роботи веб-застосунку gamepressure

Наступний приклад – веб-застосунок thegamer (рис. 1.3). Він відрізняється від двох попередніх тим, що у ньому зібрані лише керівництва до ігрових дисциплін, які складає кваліфікована команда з високим рівнем знань та навичок у цій сфері[7].

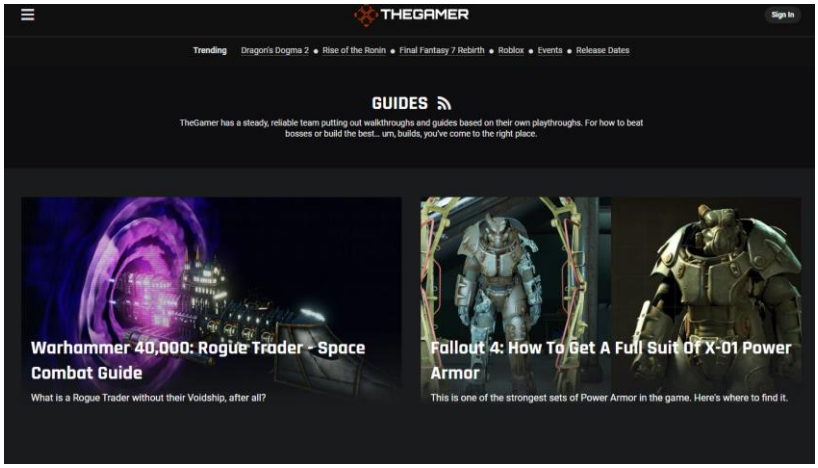


Рисунок 1.3 – Приклад роботи веб-застосунку thegamer

GamerGains (рис. 1.4) – веб-застосунок, який дозволяє гравцям виконувати прості завдання в популярних ігрових дисциплінах, винагороджуючи їх власною валютою, яку можна обмінювати на ігрові предмети.

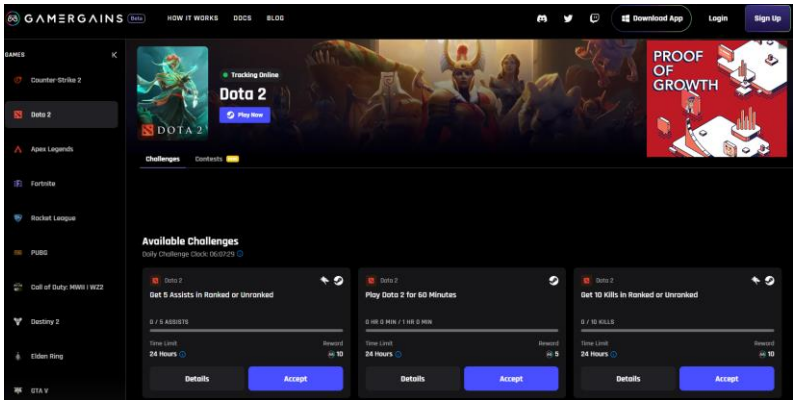


Рисунок 1.4 – Приклад роботи веб-застосунку GamerGains

Gamer Sensei (рис. 1.5) – веб-застосунок, який дозволяє користувачам обирати собі тренера в певній дисципліні, який буде давати настанови та пояснювати базові або просунуті елементи механік та взаємодій безпосередньо під час ігрового процесу, тобто наживо, що є значною перевагою [8]. Однак, даний ресурс працює лише за оплату, що є великим недоліком для багатьох гравців.

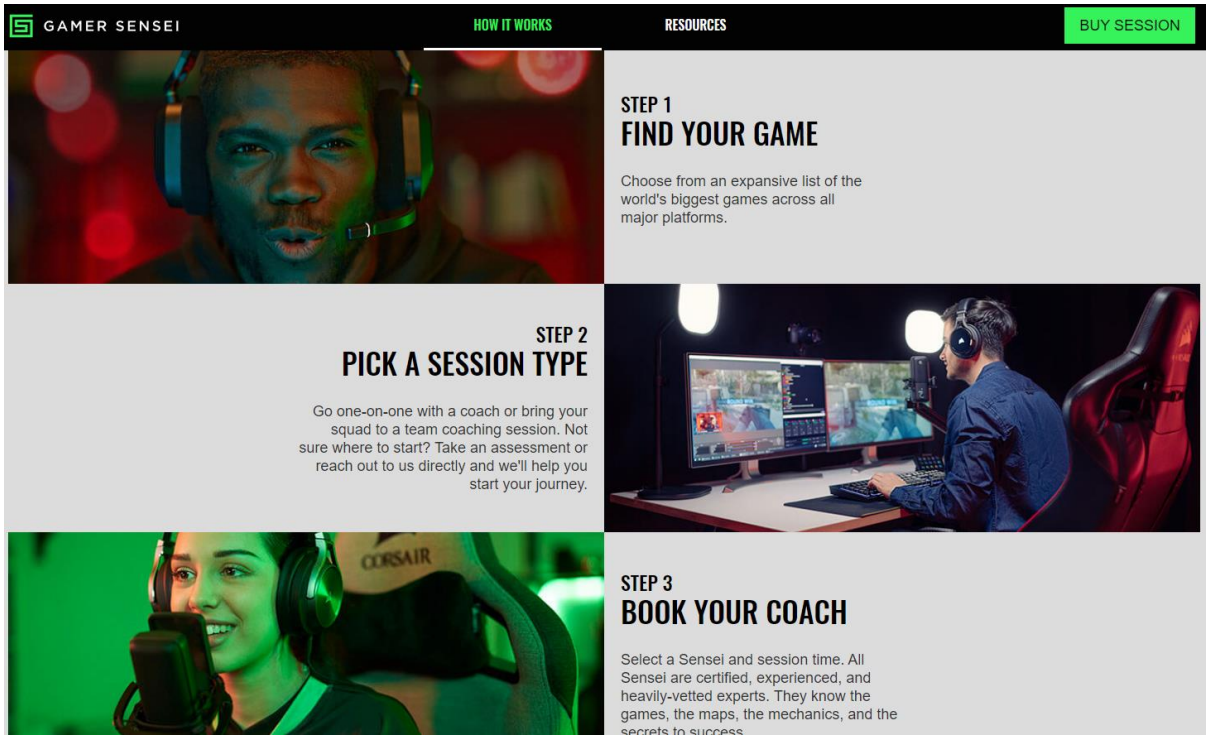


Рисунок 1.5 – Приклад роботи веб-застосунку Gamer Sensei

Розробка веб-застосунку для навчання в ігрових дисциплінах вимагає величезного досвіду з розробки і дизайну веб-застосунків, а також високого рівня знань в ігровій індустрії та сфері ігрових дисциплін. Це передбачає створення онлайн-системи, яка допоможе гравцям переглянути найпопулярніші ігрові дисципліни, визначити поточний рівень навичок, дозволить набути нових знань та закріпити їх практично, за допомогою практичних завдань.

Наведені аналоги мають власні недоліки у вигляді відсутності можливості визначення поточного рівня вмінь, відсутності різноманітного вибору дисциплін, відсутності ігрових керівництв, відсутності навчальних завдань, або наявності платного функціоналу. Тому прийнято рішення про розробку власного програмного застосунку, який не міститиме дані недоліки.

Результати порівняння аналогів зведено в табл. 1.1.

Таблиця 1.1 – Порівняльні характеристика онлайн платформ

Критерії	IGN	game pressure	the gamer	Gamer Gains	Gamer Sensei	Власний веб-затосунок
Визначення поточного рівня вмінь	-	-	-	+	+	+
Різноманітний вибір дисциплін	+	+	-	+	-	+
Ігрові керівництва	+	+	+	-	-	+
Навчальні завдання	-	-	-	+	+	+
Безкоштовно	-	+	+	+	-	+
Загальна оцінка	40%	60%	40%	80%	40%	100%

Відповідно до таблиці порівняльних характеристик розробка власного програмного застосунку для навчання кіберспортсменів є доцільною. Отриманий продукт зможе покрити недоліки існуючих рішень та встановити новий стандарт серед навчальних ресурсів для ігрових дисциплін.

Отже, розробка програмного застосунку для тренування кіберспортсменів дозволить гравцям відкрити для себе новий найбільш оптимальний спосіб отримати нові знання в ігрових дисциплінах і здобути нові та відточити набуті навички, що підвищить їхній загальний рівень вмінь та встановить новий стандарт для ігрової індустрії.

1.3 Аналіз методів розв’язання задачі

Розробка методу навчання в ігрових ситуаціях та програмного застосунку для тренування кіберспортсменів вимагає ретельного розгляду найкращих підходів до вирішення проблеми. Існує 3 основних методи, кожен з яких має свої переваги та недоліки. У цьому розділі проаналізовано найбільш ефективні методи розробки таких веб-застосунків.

Перший – це програмний застосунок, який збирає ігрову статистику за допомогою використання ігрових API (наприклад Steam API), під час виконання завдань, задля справедливого видання справжніх винагород (наприклад локальна валюта сайту). Даний тип застосунків дозволяє гравцям бачити свою ігрову статистику, що дозволяє їм порівнювати свої середні показники з іншими гравцями, що є плюсом, але одночасно це і є недоліком. Такий підхід змушує гравців концентруватися на цифрах, а не на процесі навчання, оскільки гравці будуть розчаровуватися, якщо їхні показники будуть нижчі, ніж у інших, що спонукатиме їх грати на статистику. Окрім того, великими недоліками є те, що не усі розробники ігрових дисциплін готові надавати API для збирання статистики, а локальні валюти часто використовують блок-чейн технології, які мають не зовсім хороші асоціації в користувачів. Також, як API, так і блок-чейн, потребують величезних навичок у розробці, що несе великі технічні та юридичні ризики [9].

Другий підхід – це сервіс з підключенням людей у вигляді тренерів, які будуть наживо контролювати процес. Даний підхід довзоляє гравцям набиратись знань завдяки професіоналам, які мають тисячі годин у дисциплінах. Недоліками цього підходу є те, що необхідно утримувати велику кількість співробітників, які будуть взаємодіяти з користувачами. Оскільки процес відбувається наживо, то необхідно мати надзвичайно великі технічні навички, щоб забезпечити безперервність та стабільність процесу навчання, щоб зв’язок був стабільним та не переривався, а також, канали зв’язку мають бути захищені, щоб ніякі дані користувачів та

співробітників не були викрадені. Тобто даний підхід несе великі ризики розробникам у разі технічних несправностей.

Третій підхід – це програмний застосунок у вигляді веб-застосунку без збору статистики, без підключення тренерів та з платонічними винагородами (наприклад у вигляді персональних досягнень). Даний підхід не надає користувачам більше інформації про ігровий процес та живого спілкування зі спеціалістом, проте дозволяє сфокусувати увагу користувача саме на тому, що є важливим для нього, та є легшим у реалізації, оскільки не потребує додакових знань у сторонніх технологіях, та несе менше загроз з боку захисту особистої інформації. Він дозволить сконцетрувати потужності технічної частини веб-застосунку саме на процес навчання.

Розглянувши переваги та недоліки кожного методу, було вирішено використати третій підхід. Він дозволить отримати повноцінний веб-застосунок без необхідності імплементації сторонніх технологій та полегшить процес захисту інформації користувачів. Більше того, такий метод розробки дозволить створити унікальну платформу, яка сфокусує гравців саме на процесі навчання в ігрових дисциплінах.

1.4 Постановка задач бакалаврської дипломної роботи

Проаналізувавши переваги та недоліки існуючих методів навчання в ігрових дисциплінах та програмних-застосунків для тренування кіберспортсменів, було визначено наступні завдання, які необхідно виконати для розробки методу навчання в ігрових ситуаціях та програмного застосунку для тренування кіберспортсменів:

- визначити найбільш ефективний підхід для визначення рівня навичок користувача у обраній дисципліні;
- розробити метод навчання в ігрових ситуаціях;
- розробити зручний та адаптивний інтерфейс веб-застосунку;

- розробити веб-застосунок для навчання в ігрових дисциплінах;
- провести тестування веб-застосунку згідно поставлених задач.

1.5 Висновки

У першому розділі було розглянути стан програмних застосунків для навчання в ігрових дисциплінах. Було проведено порівняння відомих платформ для навчання, таких як: IGN, gamepressure, thegamer, GamerGains, Gamer Sensei.

У результаті порівняння було виявлено, що досліджувані аналоги набули значної популярності серед гравців завдяки своїй зручності та доступності. Проаналізувавши переваги та недоліки їхніх методів навчання, було виявлено, що вони не об’єднують керівництва та завдання в іграх, містять платний функціонал, не надають можливість визначити поточний рівень навичок гравця та не містять усі найпопулярніші ігрові дисципліни. Таким чином було доведено доцільність розробки власного методу навчання в ігрових дисциплінах і програмного застосунку для тренування кіберспортсменів. На основі отриманої інформації було сформовано перелік завдань, які необхідно виконати для розробки власного методу та програмного-застосунку.

2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Розробка методу навчання в ігрових ситуаціях

Перед наведенням власного методу необхідно навести інші методи навчання в ігрових ситуаціях. Існує два основних методи.

Перший метод – метод навчання керівництвами. Його суть полягає у прочитанні інформації, яку наведено в керівництві. Даний метод дозволяє навести велику кількість інформації, це одночасно є і позитивним, і негативним фактором, оскільки засвоєння великої кількості інформації в текстовому вигляді є достатньо складним для пересічного гравця.

Другий метод – метод навчання ігровими завданнями. Його суть полягає у набутті практичних навичок безпосередньо в процесі гри. Даний метод дозволяє навчатись практикуючись в конкретних ігрових ситуаціях. Недоліком цього методу є те, що гравцю необхідно мати певний рівень знань, який дозволить йому приступити до виконання завдання, що й викликає труднощі в початківців.

В зв’язку з наявністю серйозних недоліків існуючих методів прийнято рішення створити власний. Даний метод комбінуватиме позитивні сторони попередніх двох методів, нівелюючи їхні недоліки. Таким чином користувачі прочитають керівництво, яке підготує їх до виконання завдань, а далі зможуть приступити до безпосереднього виконання.

Формальний опис методу виглядає наступним чином: користувачі застосунку обирають дисципліну, в якій бажають покращити навички, проходять опитування для визначення їхнього поточного рівня вмінь, обирають керівництва відповідного рівня для прочитання та завдання які необхідно виконувати у обраній ігровій дисципліні. Винагорода за прогрес нестиме платонічний характер (задоволення від покращення рівня навичок).

2.2 Розробка моделі навчання в ігрових ситуаціях.

Після наведення формального опису, враховуючи складність завдання, було вирішено створити математичну модель для чіткого розуміння процесу.

Математичні моделі особливо корисні при розробці програмних застосунків, оскільки забезпечують структурований та науково обгрунтований підхід до вирішення складних проблем, розуміння та оптимізації процесів. Використання математичних моделей дозволяє розробникам покращити функціональність, продуктивність та надійність своїх програмних продуктів.

Використання математичних моделей у розробці веб-застосунків має кілька важливих переваг. По-перше, це дозволяє розробникам більш точно моделювати та прогнозувати поведінку системи, що сприяє підвищенню надійності та стабільності. По-друге, математичні моделі допомагають оптимізувати ресурси та знижувати витрати, що є критично важливим для комерційних веб-застосунків. По-третє, використання моделей машинного навчання та аналізу даних дозволяє створювати інтелектуальні системи, які автоматично адаптуються до змін та вдосконалюються з часом.

Таким чином, математичні моделі є невід'ємною частиною процесу розробки веб-застосунків, надаючи розробникам потужні інструменти для аналізу, оптимізації та прогнозування. Використання цих моделей сприяє створенню високоякісних, надійних та ефективних програмних продуктів, які відповідають потребам сучасних користувачів.

Для розуміння методу навчання в ігрових дисциплінах, який буде імплементовано у програмний-застосунок для тренування кіберспортсменів, було розроблено математичну модель методу, який враховуватиме 3 ключових аспекти:

- рівень навичок користувача;
- вибір дисципліни та керівництв або завдань;
- відстеження прогресу користувача;

Рівень навичок (Skill Level, SL) визначається через початкове опитування і може змінюватись з часом залежно від кількості прочитаних керівництв та виконаних завдань. Позначимо рівень навичок як SL_i, де i – це індекс користувача. Рівні навичок можуть бути від 1 до 3.

Кожна ігрова дисципліна має набір керівництв, які позначаються відповідно до рівня складності:

- $G_{d,k}$ – набір керівництв для дисципліни d і рівня k;
- $T_{d,k}$ – набір завдань для дисципліни d і рівня k.

Користувач обирає дисципліну d і отримує відповідний набір керівництв та завдань для свого рівня SL_i.

Далі позначається:

- $C_{i,d,k}$ – кількість прочитаних керівництв користувем I в дисципліні d на рівні k;
- $E_{i,d,k}$ – кількість виконаних завдань користувачем i в дисципліні d на рівні.

Прогрес користувача може бути виражений, як відсоток виконання кожного типу активності:

- прогрес читання читання керівництв $P_{G,i,d,k} = (C_{i,d,k} / G_{d,k})* 100\%$;
- прогрес виконання завдань $P_{T,i,d,k} = (E_{i,d,k} / T_{d,k})* 100\%$.

В прогресі користувача $|G_{d,k}|$ та $|T_{d,k}|$ - це загальна кількість керівництв та завдань для дисципліни d і рівня k.

Далі йде формалізація моделі методу. На рисунку 2.1 зображено систему рівнянь для кожного користувача.

$$SL_i(t+1) = \begin{cases} SL_i(t)+1 & \text{якщо } P_{G,i,d,k}(t) \geq 80\% \text{ і } P_{T,i,d,k}(t) \geq 80\% \\ SL_i(t)-1 & \text{якщо } P_{G,i,d,k}(t) \leq 20\% \text{ і } P_{T,i,d,k}(t) \leq 20\% \\ SL_i(t) & \text{в іншому випадку} \end{cases}$$

Рисунок 2.1 – Система рівнянь користувача

У наведеній системі t – це поточний час, а t+1 – це наступний часовий інтервал.

Персональний кабінет користувача відображає його прогрес у вигляді списку прочитаних керівництв($C_{i,d,k}$) і виконаних завдань($E_{i,d,k}$).

Дана математична модель дозволяє структурувати навчальний прогрес користувачів та тренування кіберспортсменів, дозволяє оцінювати їхній прогрес, тобто є ефективною для імплементації у веб-застосунок для тренувань.

2.3 Аналіз вхідних даних системи

Етап аналізу вхідних даних системи забезпечує розуміння того, як дані будуть надходити, оброблятися та використовуватися в системі. Цей процес починається зі збору всіх можливих джерел даних, що можуть включати дані користувачів, інтеграції з іншими системами, а також внутрішні дані організації. Важливо визначити формати цих даних, частоту їхнього оновлення та обсяг, щоб мати чітке уявлення про те, з чим доведеться працювати.

Після збору даних проводиться їхнє попереднє оброблення, що включає очищення, нормалізацію та трансформацію. На цьому етапі виявляються та виправляються помилки, усуваються дублікати, а також приводяться дані до єдиного формату. Це дозволяє забезпечити їхню коректність та відповідність вимогам системи. Крім того, аналізується структура даних і взаємозв'язки між різними наборами даних, що допомагає краще зрозуміти, як їх оптимально зберігати та обробляти.

Далі здійснюється аналіз вимог до даних з боку системи. Це включає визначення необхідних полів, типів даних, обсягів зберігання та вимог до продуктивності. На основі цих вимог розробляється модель даних, яка описує, як дані будуть організовані в системі. Ця модель має бути гнучкою і масштабованою, щоб відповідати можливим змінам у майбутньому.

Відштовхуючись від потреб методу можна зробити висновок, що у програмному застосунку будуть два основних модулі, які відповідатимуть за керівництва та завдання і їхні сторінки, а також модуль користувача. Дані модулі потребуватимуть власні вхідні дані.

Таким чином для керівництв та завдань визначено наступні вхідні дані:

- вибір дисципліни: серед списку дисциплін обиратиметься необхідна ігрова дисципліна (список посилань);
- вибір керівництва або завдання: серед списку посилань керівництв та завдань обиратиметься необхідне керівництво або завдання (список посилань).

Для користувача визначено наступні вхідні дані:

- логін користувача (набір символів);
- пароль користувача (набір символів);
- дані опитувальника: вибір дисципліни (список); запитання про рівень вмінь певного аспекту дисципліни 1 (цифра від 1 до 3) запитання про рівень вмінь певного аспекту дисципліни 2 (цифра від 1 до 3); запитання про рівень вмінь певного аспекту дисципліни 3 (цифра від 1 до 3).

Отже, проаналізувавши вхідні дані системи, можна зробити висновок про потреби для реалізації системи. При розробці системи необхідно буде впевнитись, що вхідні дані, які приймаються будуть саме зазначених типів, для цього в свою чергу необхідно буде додати код для перевірки коректності введених даних.

2.4 Розробка інтерфейсу веб-застосунку

Процес розробки інтерфейсу користувача для веб-застосунку зазвичай проходить кілька основних етапів, від дослідження до впровадження і тестування. Спершу здійснюється вивчення потреб користувачів і аналіз ринку. На цьому етапі збираються вимоги, проводяться аналіз конкурентів і опитування користувачів, що допомагає визначити основні функції та дизайн майбутнього застосунку.

Далі створюються вайрфрейми та прототипи. Вайрфрейми являють собою схематичні моделі, що демонструють розташування елементів на сторінках без деталізації графічного дизайну. Прототипи є більш інтерактивними і дозволяють користувачам взаємодіяти з інтерфейсом, перевіряючи основні сценарії використання. Для цього часто використовуються інструменти, такі як Figma, Sketch або Adobe XD.

Після затвердження прототипів відбувається візуальне оформлення дизайну. Дизайнер деталізує елементи інтерфейсу, включаючи кольорові схеми, шрифти, іконки та інші графічні компоненти, забезпечуючи естетичну привабливість і зручність використання. Важливо враховувати принципи адаптивного дизайну, щоб інтерфейс правильно відображався на різних пристроях.

Завершальний етап включає розробку і тестування. Розробники перетворюють дизайн у функціональний веб-застосунок, використовуючи HTML, CSS, JavaScript та інші технології. Після цього проводиться тестування для виявлення і виправлення помилок, забезпечення зручності користування та відповідності вимогам. Тестування може включати автоматизовані тести, юзабіліті-тестування та A/B тестування. Після успішного завершення тестування веб-застосунок готовий до запуску і подальшої підтримки.

Інтерфейс веб-застосунку було вирішено зробити максимально простим та зручним для використання [10].

Основним кольором було обрано білий, через його нейтральність. Допоміжними кольорами було обрано сірий, зелений та синій, оскільки вони будуть концентрувати увагу на важливих елементах веб-застосунку.

При відкритті веб-застосунку, користувачі потрапляють на головну сторінку, яка містить панель навігації, яка містить логотип та посилання на інші сторінки, та блок з інформацією про сайт (рис. 2.2).

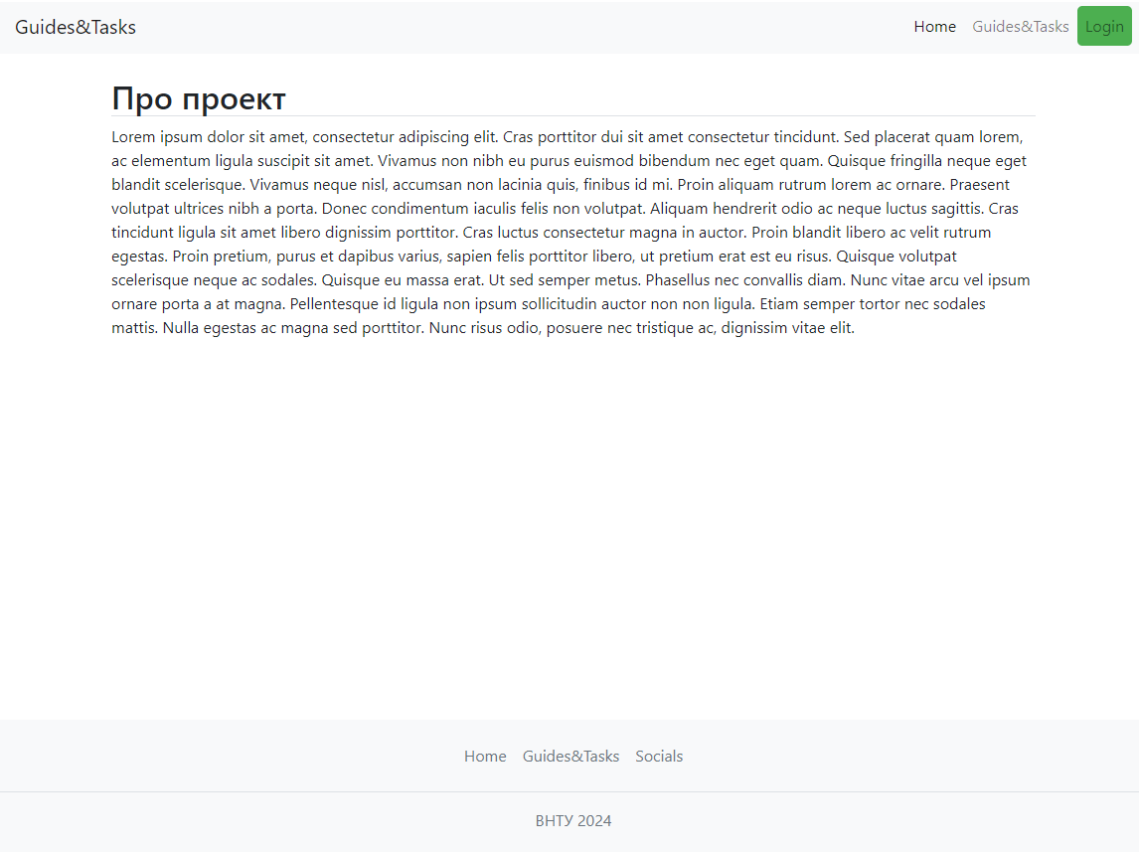


Рисунок 2.2 – Початкова сторінка

Оскільки веб-застосунок передбачає реєстрацію окремих користувачів у системі, спроектовано та створено панель реєстрації користувача (рис. 2.3). Користувач повинен ввести дані для реєстрації у вигляді логіну та пароля.

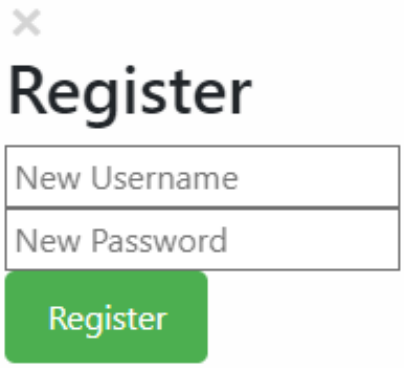


Рисунок 2.3 – Панель реєстрації користувача

Далі користувач повинен авторизуватися, за допомогою введення даних, які він використав для реєстрації (рис. 2.4). Далі йде процес авторизації користувача.

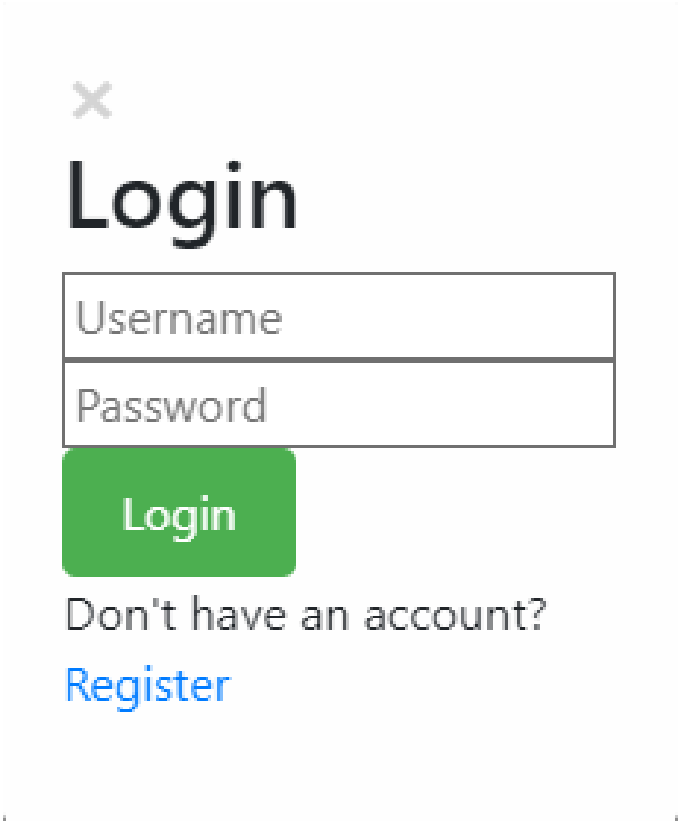


Рисунок 2.4 – Панель авторизації користувача

Якщо були введені правильні дані, користувач отримує доступ до повного функціоналу сайту. Далі, обравши ігрову дисципліну у навігаційній панелі, користувач повинен пройти опитування для визначення поточного рівня знань та навичок у ній (рис. 2.5).

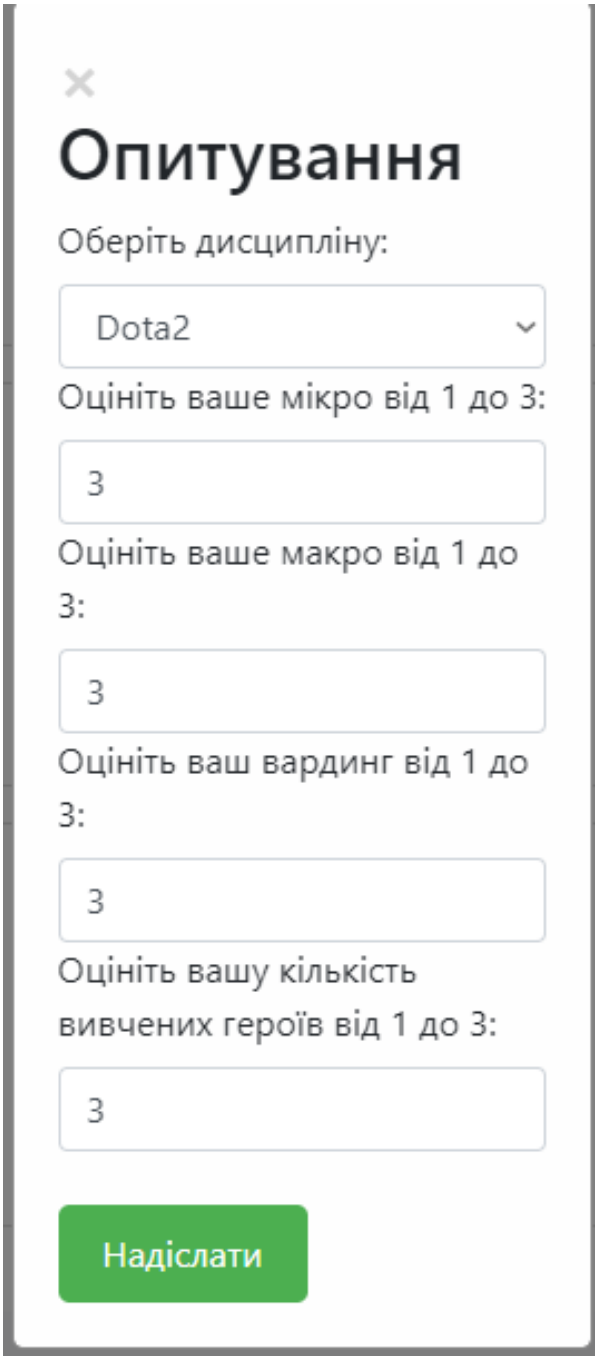


Рисунок 2.5 – Опитування

Після проходження опитування користувач отримує список керівництв та завдань до обраної ігрової дисципліни, які відповідають його поточному рівню. В списку керівництв та завдань є короткий опис до кожного та рівень складності. Керівництва містять інформацію про певні механіки ігрової дисципліни у вигляді

тексту та кнопку «Переглянуто», а також можуть містити відео, знімки екрану під час ігрового процесу (Рис. 2.6).

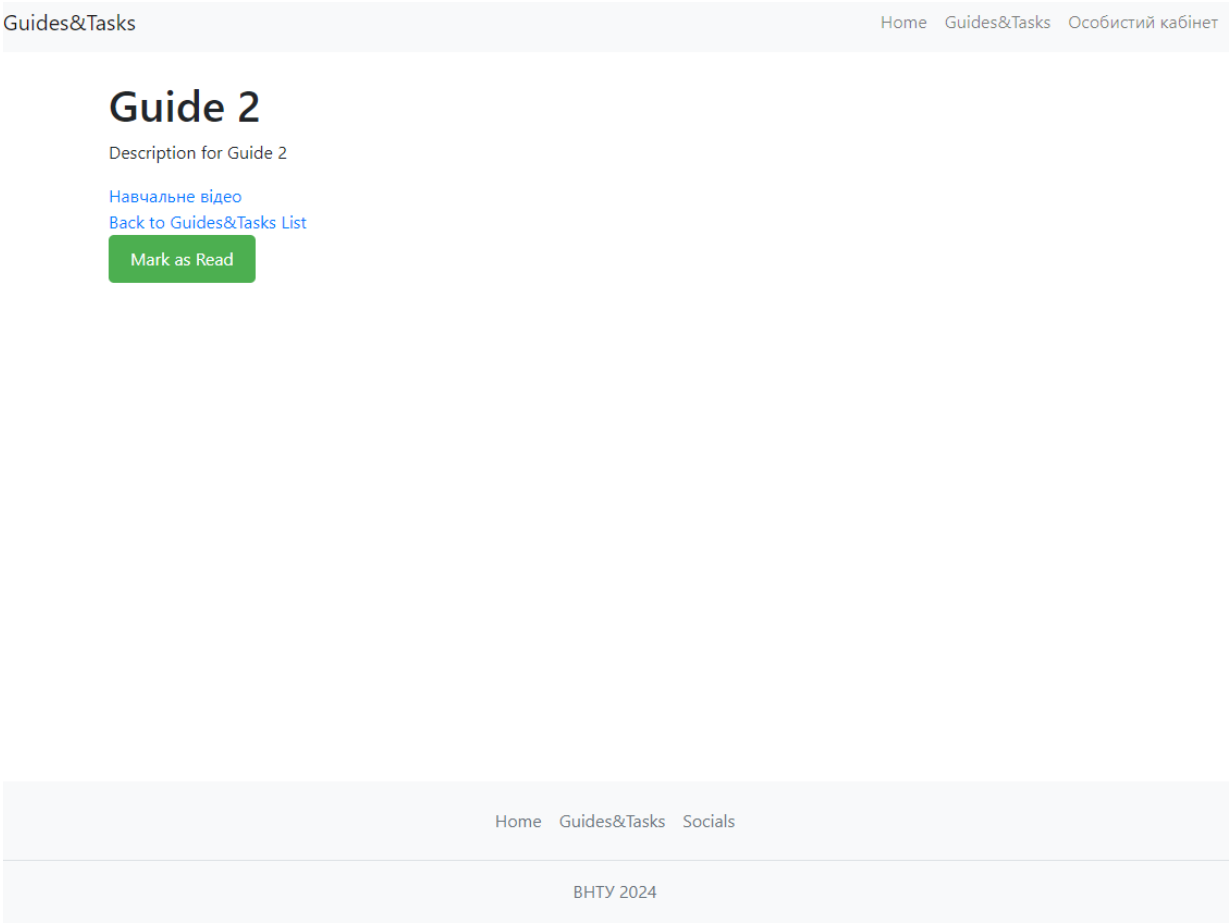


Рисунок 2.6 – Сторінка керівництва

Після перегляду керівництва, користувач може обрати завдання для виконання. Воно містить інформацію про зміст завдання та рекомендації у вигляді тексту та кнопку «Виконано». Також, може містити відео та знімки екрану під час ігрового процесу (рис. 2.7).

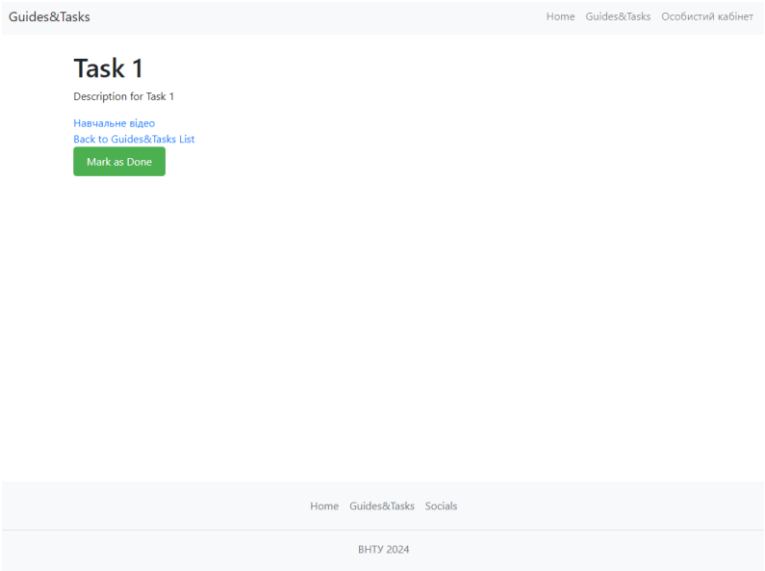


Рисунок 2.7 – Сторінка завдання

Кнопки «Переглянуто» та «Виконано» дозволяють користувачу позначити переглянуті керівництва та виконані завдання. Користувач може переглянути їх у своєму особистому кабінеті, натиснувши на іконку свого профілю та обравши відповідну дисципліну.

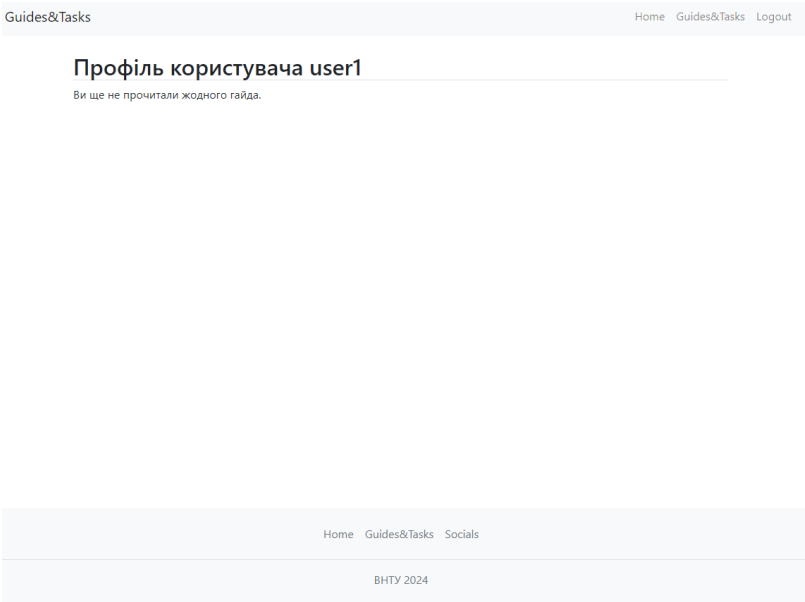


Рисунок 2.8 – Особистий кабінет користувача

Розробка графічного інтерфейсу веб-застосунку була проведена у середовищі розробки Visual Studio з використанням бібліотек BootsTrap і jQuery.

2.5 Розробка моделі веб-застосунку

Діаграми діяльності є важливим інструментом для розробки моделей веб-застосунків. Вони дозволяють розробникам візуалізувати процеси та потоки дій, що відбуваються у застосунку, що сприяє глибшому розумінню логіки роботи системи. Використання діаграм діяльності має багато переваг, які допомагають оптимізувати процес розробки та забезпечити високу якість кінцевого продукту. Вони є потужним інструментом для моделювання, аналізу і оптимізації процесів у веб-застосунках, оскільки сприяють покращенню комунікації, забезпечують чітке розуміння логіки роботи системи і допомагають створити високоякісний продукт, що відповідає вимогам користувачів і замовників. Використання ддіаграм діяльності під час розробки веб-застосунків дозволяє досягти високого рівня організованості та ефективності процесу розробки.

Для розуміння роботи програмних модулів застосунку для тренування кіберспортсменів було розроблено модель роботи у вигляді діаграми діяльності (рис. 2.9). Згідно вказаної діаграми користувачу спочатку необхідно обрати ігрову дисципліну в навігаційній панелі. Далі пропонується авторизуватись та здійснити проходження опитування для визначення поточного рівня знань та вмінь, якщо воно ще не було пройдене. Якщо користувач вирішує авторизуватись, але процес проходить не вдало, то система запропонує йому авторизуватися знову. Після авторизації пропонуються доступні керівництва та завдання, які користувач може переглядати та виконувати.

Рисунок 2.9 – Діаграма діяльності веб-застосунку

2.6 Розробка алгоритмів роботи веб-застосунку

Розробка алгоритмів є однією з ключових складових процесу створення веб-застосунків. Алгоритми визначають логіку та послідовність дій, які виконує

програма для досягнення поставлених завдань. Вони відіграють критичну роль у забезпеченні функціональності, продуктивності та надійності веб-застосунку, оскільки визначають логіку роботи системи, впливають на її продуктивність, надійність, безпеку та загальну якість. Використання ефективних та оптимізованих алгоритмів дозволяє створювати високоякісні веб-застосунки, які відповідають потребам користувачів та забезпечують конкурентоспроможність на ринку.

Для розробки веб-застосунку для тренування кіберспортсменів необхідно розробити загальний алгоритм, згідно якого він буде працювати (рис. 2.10). Згідно зазначеного алгоритму, користувач спершу обирає ігрову дисципліну. Якщо користувач не авторизований, то йому буде запропоновано авторизуватися. Якщо користувач вирішить не авторизовуватись, то йому видасть список усіх наявних завдань та керівництв з коротким описом та рівнем складності, але він не зможе їх обрати доти, доки не здійснить авторизацію. Якщо користувач авторизовується, та вперше обирає дану ігрову дисципліну, то йому пропонується пройти опитування для визначення рівня поточних вмінь та знань. Після проходження опитування користувач отримує список з керівництвами та завданнями, які відповідають його поточному рівню. Далі користувач може обирати, що саме він бажає переглянути або виконати. Після перегляду керівництва, або виконання завданя він може позначити їх, як прочитане або виконане відповідно.

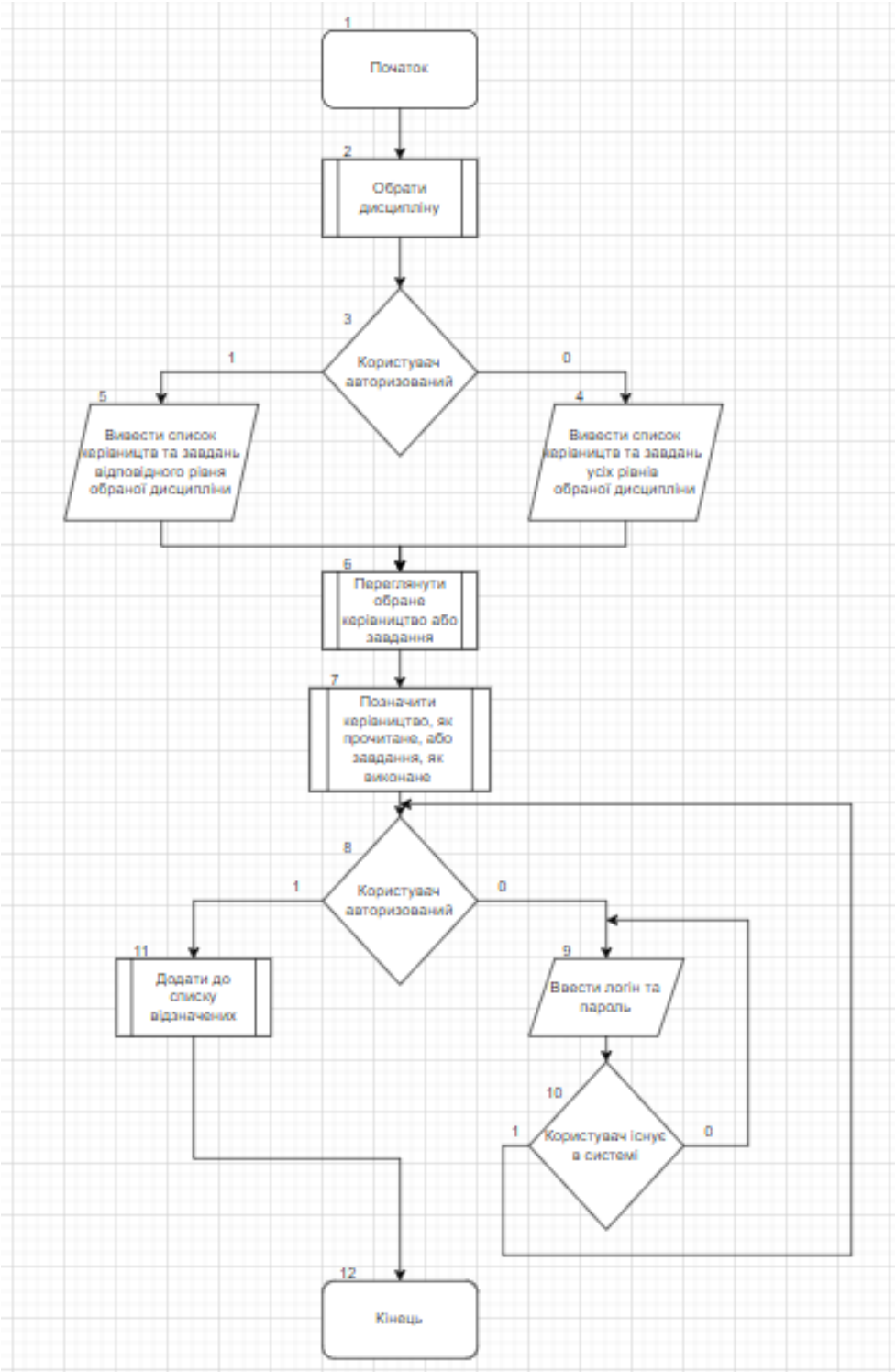


Рисунок 2.10 – Загальний алгоритм роботи

Розглянувши даний алгоритм, можна робити висновок, що розроблений веб-застосунок надає можливість користувачам ефективно переглядати керівництва та

виконувати завдання для навчання в обраних ним ігрових дисциплінах. Даний алгоритм дозволяє оптимізувати та пришвидшити процес навчання, що робить розроблений веб-застосунок найкращим для гравців, фанатів та ентузіастів ігор, які бажають підвищити свій рівень у них.

2.7 Висновки

У другому розділі було проведено аналіз вхідних та вихідних даних веб-застосунку. Також було розроблено дизайн інтерфейсу веб-застосунку, математичну модель методу навчання в ігрових дисциплінах для програмного застосунку, модель системи у вигляді діаграми та алгоритм роботи системи.

3. РОЗРОБКА ПРОГРАМНИХ МОДУЛІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО ЗАСТОСУНКУ ДЛЯ ТРЕНУВАННЯ КІБЕРСПОРТСМЕНІВ

3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного забезпечення

При розробці програмних модулів для програмного застосунку у вигляді веб-застосунку для навчання в ігрових дисциплінах і тренування кіберспортсменів необхідно здійснити ретельний вибір технологій для реалізації. HTML та CSS – це мови програмування, які є основою веб-застосунків, оскільки надають можливість створити розмітку та стилізувати її, відповідно до тематики та функціоналу сайту [11].

З метою оптимізації процесу розробки було вирішено використати фреймворк BootsTrap. Даний фреймворк дозволяє використати готові стилізовані макети частин сайту, які легко редагувати. Таким чином немає необхідності у ручному написанні окремих частин сайту з нуля. Процес підключення даного фреймворку також є достатньо простим, необхідно лише вставити посилання для його підключення поруч з мета-тегами в коді сторінок сайту [12].

Функціонал роботи сайту реалізований за допомогою мови програмування JavaScript, оскільки дана об'єктно-орієнтована мова програмування дозволяє запускати функції під час роботи сайту та оновлювати окремі частини сайту без необхідності у його перезавантаженні [13]. Для маніпуляції частинами сайту було використано бібліотеки jQuery.

jQuery – це бібліотека, що сформована для мови програмування JavaScript [14]. Вона сприймається розробниками, як фреймворк, й дозволяє маніпулювати стилями елементів за допомогою CSS, здійснювати обробку подій в веб-документі, використовувати анімації та ефекти, відображати елементи у DOM-структурі сторінки та використовувати технології AJAX [15].

Для серверної частини сайту було використано PHP [16]. PHP – це мова програмування для генерації HTML-сторінок на стороні веб-сервера. PHP інтерпретується вебсервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта [17].

За допомогою поєднання даних технологій стає можлива розробка веб-застосунку для навчання у ігрових дисциплінах. HTML, CSS та BootsTrap виступатимуть у якості основи, доки JavaScript, jQuery та PHP забезпечуватимуть користувацьку та серверну частину веб-застосунку.

Отже, вибір технологій для розробки програмних модулів веб-застосунку для навчання в ігрових дисциплінах, що використовує HTML, CSS, BootsTrap, JavaScript, jQuery та PHP дозволяє отримати надійний інструментарій для розробки надійного та зручного веб-застосунку.

3.2 Розробка модулів відображення списку керівництв та завдань та створення їхніх сторінок

Головним завданням даних модулів є надання можливості користувачу обрати та переглянути керівництва і виконати завдання. Для цього було розроблено окремі функції, кожна з яких відповідає за забезпечення роботи певної частини веб-застосунку.

Усе починається з вибору ігрової дисципліни, який відбувається за допомогою елементів вибору у вигляді кнопок після натиснення на яких буде відображено відповідний їм список керівництв і завдань (рис. 3.1).

```
<div class="container mt-4">
    <h2>Дисципліни</h2>
    <div class="list-group list-group-horizontal">
        <a href="index.php" class="list-group-item list-group-item-action active" aria-current="true">Dota 2</a>
        <a href="CS2.php" class="list-group-item list-group-item-action">Counter Strike 2</a>
        <a href="StarCraft2.php" class="list-group-item list-group-item-action">Star craft 2</a>
    </div>
</div>
```

Рисунок 3.1 – Код кнопок для вибору ігрової дисципліни

Для відображення списку керівництв і завдань був написаний код мовою програмування PHP. Керівництва подаються масивом array, після чого проходить перевірка чи був заданий критерій сортування, а далі йде створення елементів DOM і відображення вмісту кожного з них, за допомогою echo, який знаходиться в колі foreach (рис. 3.2).

```
<?php
// Simulated data for guides (you'll replace this with your actual data source)
$guides = array(
    array('id' => 1, 'title' => 'Guide 1', 'description' => 'Description for Guide 1', 'video_link' => 'youtube.com', 'difficulty' => 2, 'date_added' => '2024-04-20'),
    array('id' => 2, 'title' => 'Guide 2', 'description' => 'Description for Guide 2', 'video_link' => 'youtube.com', 'difficulty' => 1, 'date_added' => '2024-04-19'),
    array('id' => 3, 'title' => 'Guide 3', 'description' => 'Description for Guide 3', 'video_link' => 'youtube.com', 'difficulty' => 3, 'date_added' => '2024-04-21'),
);

// Check if sorting parameter is provided
if (isset($_GET['sort'])) {
    $sortBy = $_GET['sort'];
    switch ($sortBy) {
        case 'difficulty':
            usort($guides, function ($a, $b) {
                return $a['difficulty'] - $b['difficulty'];
            });
            break;
        case 'alphabetical':
            usort($guides, function ($a, $b) {
                return strcmp($a['title'], $b['title']);
            });
            break;
        case 'date':
            usort($guides, function ($a, $b) {
                return strtotime($b['date_added']) - strtotime($a['date_added']);
            });
            break;
    }
}

// Display guides
foreach ($guides as $guide) {
    echo '<div class="card mb-3">';
    echo '<div class="card-body">';
    echo '<h5 class="card-title"><a href="guide.php?id=' . $guide['id'] . '">' . $guide['title'] . '</a></h5>';
    echo '<p class="card-text">' . $guide['description'] . '</p>';
    echo '<p class="card-text">Difficulty: ' . $guide['difficulty'] . '</p>';
    echo '<p class="card-text">Date added: ' . $guide['date_added'] . '</p>';
    echo '</div>';
    echo '</div>';
}
?>
```

Рисунок 3.2 – Код для генерації DOM для керівництв

Список керівництв та завдань буде розміщено у спеціально відведеному для нього елементі, який знаходиться у коді сторінки після контейнеру з кнопками для вибору дисципліни (рис. 3.3).

```
<div id="guideList">
    <!-- Guides will be dynamically inserted here -->
</div>
```

Рисунок 3.3 – Код елементу для списку керівництв та завдань

У функції продемонстровано, що для кожного керівництва буде створено окреме посилання на його сторінку, за допомогою вибору його id. Наприклад для керівництва 1, буде створено посилання «guide.php?id=1». Натиснення на дане посилання відкриває його сторінку, яка шукає керівництво за критерієм «id» за допомогою «array_filter», повертаючи його «id», здійснює перевірку чи дане керівництво існує завдяки «if(!empty($selectedGuide))» і створює елементи DOM для його вмісту за допомогою «echo». Якщо керівництво не було знайдене, то буде виведено відповідний текст (рис. 3.4).

```
if (isset($_GET['id'])) {
    $guideId = $_GET['id'];
    // Find the guide with the matching ID
    $selectedGuide = array_filter($guides, function ($guide) use ($guideId) {
        return $guide['id'] == $guideId;
    });
    // If guide found, display its details
    if (!empty($selectedGuide)) {
        $selectedGuide = array_values($selectedGuide)[0]; // Extracting the first (and only) element
        echo '<h1>' . $selectedGuide['title'] . '</h1>';
        echo '<p>' . $selectedGuide['description'] . '</p>';
        echo '<a href="' . $selectedGuide['video_link'] . '" target="_blank">Навчальне відео</a>';
        ?>
    <br>
    <a href="index.php">Back to Guides&Tasks List</a>
    <br>
    <button id="markAsRead">Mark as Read</button>
<?php
    } else {
        echo 'Guide not found.';
    }
} else {
    echo 'Guide ID not provided.';
}
?>
```

Рисунок 3.4 – Код для генерації сторінки керівництва

3.3 Розробка модулів авторизації та відображення особистого кабінету

Основним завданням даних модулів є забезпечення авторизації користувача та надання йому можливості увійти в особистий кабінет, де буде відображено позначені керівництва та завдання. Для цього на кожній активній створінці

містяться елементи DOM та код PHP, які дозволять забезпечити перевірку, чи користувач авторизований, увійти за допомогою окремого вікна авторизації, відобразити посилання на особистий кабінет, якщо користувач був авторизований (рис 3.5).

```
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item">
                    <a class="nav-link" href="home.php">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="get_guides.php">Guides&Tasks</a>
                </li>
                <li class="nav-item">
    <?php
    if (isset($_SESSION['username'])) {
        echo '<a class="nav-link" href="user_profile.php">Особистий кабінет</a>';
    } else {
        echo '<button id="loginButton" class="nav-link">Login</button>';
    }
    ?>
                </li>
            </ul>
        </div>
    </nav>
</header>

<!-- Модальне вікно для входу -->
<div id="loginModal" class="modal">
    <div class="modal-content">
        <span class="close">&times;</span>
        <h2>Login</h2>
        <form id="loginForm">
            <input type="text" id="username" placeholder="Username">
            <input type="password" id="password" placeholder="Password">
            <button type="submit" id="loginSubmit">Login</button>
        </form>
        <p>Don't have an account? <a href="#" id="registerLink">Register</a></p>
    </div>
</div>
```

Рисунок 3.5 – Елементи DOM та коду для авторизації

При натисненні на кнопку «Login» відображається модальне вікно для авторизації. За замовчуванням значення його display дорівнює none, тому воно відображається лише після натиснення кнопки. Воно складається з контейнеру, який містить форму з полями для вводу ім’я користувача та паролю, а також кнопкою, яка відстежує підтвердження відправлення даних користувача для авторизації (рис. 3.6).

```
<!-- Модальне вікно для входу -->
<div id="loginModal" class="modal">
    <div class="modal-content">
        <span class="close">&times;</span>
        <h2>Login</h2>
        <form id="loginForm">
            <input type="text" id="username" placeholder="Username" class="form-control">
            <input type="password" id="password" placeholder="Password" class="form-control">
            <br>
            <button type="submit" id="loginSubmit">Login</button>
        </form>
        <p>Don't have an account? <a href="#" id="registerLink">Register</a></p>
    </div>
</div>
```

Рисунок 3.6 – Код форми авторизації користувача

Після введення даних для авторизації вони відправляються методом POST, на обробку до модуля входу. Це здійснюється за допомогою написаної за допомогою JavaScript функції (рис. 3.7.).

```
// Обробник події для подання форми входу
$("#loginForm").submit(function(event) {
    event.preventDefault(); // Запобігаємо стандартній поведінці форми

    // Отримуємо дані з форми
    var username = $("#username").val();
    var password = $("#password").val();

    // Викликаємо AJAX-запит для перевірки логіну та паролю
    $.post("login.php", { username: username, password: password }, function(response) {
        handleLoginResponse(response); // Обробка відповіді від сервера
    });
});
```

Рисунок 3.7 – Код обробника відправки даних

Обробка даних здійснюється у окремому файлі, який здійснює обробку цього запиту «Login.php». У ньому міститься код отримання запиту, його перевірки, зберігання користувача в сесії та відповіді на запит. Залежно від успішності

перевірки відправляється відповідне повідомлення. Після отримання даних, він здійснює їхню перевірку та відповідає на запит (рис. 3.8).

```
// Отримуємо дані з POST-запиту
$inputUsername = $_POST['username'];
$inputPassword = $_POST['password'];

// Перевірка логіну та паролю
$isLoggedIn = false;
foreach ($registeredUsers as $user) {
    if ($user['username'] === $inputUsername && $user['password'] === $inputPassword) {
        $isLoggedIn = true;
        $_SESSION['username'] = $inputUsername; // Зберігаємо ім'я користувача в сесії
        break;
    }
}

// Відповідь на запит
if ($isLoggedIn) {
    echo "success";
} else {
    echo "error";
}
?>
```

Рисунок 3.8 – Код обробки авторизації користувача

Після успішної авторизації, користувачу відображається посилання на особистий кабінет в навігаційній панелі на будь якій сторінці застосунку завдяки збереженні сесії користувача. Натиснувши на нього користувач потрапляє на його сторінку. При потраплянні на сторінку здійснюється перевірка, чи користувач авторизований. Якщо авторизація була здійснена успішно, запускається функція displayUserProfile, яка генерує особистий кабінет користувача у вигляді елементів навігації та контейнеру для списку прочитаних керівництв та виконаних завдань за допомогою echo (рис. 3.9).

```
// Загальна функція для відображення особистого кабінету користувача
1 reference
function displayUserProfile($username, $readGuides)
{
    echo "<header>";
    echo "<nav class='navbar navbar-expand-lg navbar-light bg-light'>";
    echo "<a class='navbar-brand' href='#'>Dota 2 Guides</a>";
    echo "<button class='navbar-toggler' type='button' data-toggle='collapse' data-target='#navbarNav'
            aria-controls='navbarNav' aria-expanded='false' aria-label='Toggle navigation'>";
    echo "<span class='navbar-toggler-icon'></span>";
    echo "</button>";
    echo "<div class='collapse navbar-collapse' id='navbarNav'>";
    echo "<ul class='navbar-nav ml-auto'>";
    echo "<li class='nav-item active'>";
    echo "<a class='nav-link' href='#'>Home <span class='sr-only'>(current)</span></a>";
    echo "</li>";
    echo "<li class='nav-item'>";
    echo "<a class='nav-link' href='#'>Guides</a>";
    echo "</li>";
    echo "<li class='nav-item'>";
    echo "<a class='nav-link' href='logout.php'>Logout</a>"; // Додано посилання для виходу
    echo "</li>";
    echo "</ul>";
    echo "</div>";
    echo "</nav>";
    echo "</header>";

    echo "<div class='container'>";
    echo "<div class='mt-4'>";
    echo "<h2>Профіль користувача $username</h2>";
    if (empty($readGuides)) {
        echo "<p>Ви ще не прочитали жодного гайда.</p>";
    } else {
        echo "<p>Ви прочитали наступні гайди:</p>";
        echo "<ul>";
        foreach ($readGuides as $guide) {
            echo "<li>$guide</li>";
        }
        echo "</ul>";
    }
    echo "</div>";
    echo "</div>";
}

// Перевірка, чи існує збережена сесія користувача
if (isset($_SESSION['username'])) {
    $loggedInUser = $_SESSION['username'];
    // Пошук користувача за логіном у зареєстрованих користувачах
    foreach ($registeredUsers as $user) {
        if ($user['username'] === $loggedInUser) {
            displayUserProfile($user['username'], $user['read_guides']);
            break;
        }
    }
} else {
    echo "Користувач не авторизований";
}
?>
```

Рисунок 3.9 – Код для обробки особистого кабінету

Серед усіх елементів особистого кабінету генерується і посилання на вихід з користувача, оскільки повинна бути можливість змінити користувача. Натиснувши на спеціальне посилання у вигляді кнопки «Logout» в панелі навігації, запускається фрагмент модуля під назвою logout.php, який закриває сесію за допомогою функцій

session_unset і session_destroy та повертає користувача на головну сторінку, яка містить стандартні елементи навігації, модальні форми для авторизації та реєстрації, за допомогою перенаправлення header(Location: index.php) (рис. 3.10).

```php
<?php
// Розпочинаємо сесію
session_start();

// Закриваємо сесію (виходимо зі свого облікового запису)
session_unset();
session_destroy();

// Перенаправляємо користувача на сторінку входу або на головну сторінку
header("Location: index.php");
exit;
?>
```

Рисунок 3.10 – Код logout.php

При поверненні на головну сторінку користувачу відображатиме такі ж елементи, як перед тим, як він авторизувався

3.4 Висновки

У третьому розділі було обґрунтовано вибір мови програмування та технологій для розробки системи. Було здійснено аналіз потреб програмного продукту та вибір мов програмування, а саме HTML, CSS, JavaScript та PHP. Для зручності розробки веб-застосунку було використано BootsTrap та jQuery, оскільки вони спрощують процес реалізації функціуналу. Також було розроблено модулі веб-застосунку.

4 ТЕСТУВАННЯ ВЕБ-ЗАСТОСУНКУ

4.1 Аналіз методів тестування програмних застосунків

Тестування програмних застосунків є невід'ємною частиною процесу їх розробки та забезпечення якості. Ефективні методи тестування дозволяють виявити помилки, недоліки та покращити функціональність, що сприяє задоволенню користувачів. У цьому розділі ми розглянемо деякі з основних методів тестування веб-застосунків, зокрема в контексті платформ для навчання та тренування кіберспортсменів.

Функціональне тестування: цей метод перевіряє, чи працюють усі функції веб-застосунку відповідно до визначених вимог. Тестувальники імітують різні сценарії використання, вводять дані, натискають кнопки та посилання, щоб переконатися, що система працює коректно. Метою є виявлення будь-яких помилок або невідповідностей у роботі функцій, що особливо важливо для навчальних платформ, де кожен елемент має критичне значення для користувачів.

Тестування користувацького інтерфейсу: це тестування оцінює зручність, доступність та естетичність інтерфейсу платформи. Тестувальники аналізують розташування елементів, навігацію, стиль та інші аспекти, щоб забезпечити інтуїтивно зрозумілий та приємний для користувачів інтерфейс. Для платформ, призначених для навчання кіберспортсменів, важливо, щоб користувачі легко знаходили потрібні матеріали та функції, не витрачаючи зайвий час на пошук.

Тестування сумісності: цей метод перевіряє, як платформа працює на різних пристроях, операційних системах та браузерах. Тестувальники перевіряють, чи коректно відображається контент, чи працюють усі функції на різних конфігураціях. Для навчальних веб-застосунків критично важливо забезпечити однакову функціональність та доступність на різних платформах, оскільки користувачі можуть використовувати різні пристрої для доступу до матеріалів.

Тестування продуктивності: цей метод оцінює швидкість та ефективність платформи під різними навантаженнями. Тестувальники виконують різні завдання, спостерігаючи за часом відгуку, часом завантаження сторінок та іншими показниками продуктивності. Для платформ, де користувачі взаємодіють з інтерактивними матеріалами або відео, важливо забезпечити швидку і стабільну роботу навіть під великим навантаженням.

Тестування безпеки: це тестування перевіряє рівень безпеки веб-застосунку, виявляє потенційні загрози та захищає конфіденційні дані користувачів. Тестувальники проводять різні сценарії атак, щоб знайти вразливості та забезпечити захист від них. Для платформ, які збирають особисту інформацію або мають фінансові транзакції, питання безпеки є особливо важливим.

Тестування навантаження: цей метод оцінює, як платформа працює під великим навантаженням, коли багато користувачів одночасно взаємодіють з нею. Тестувальники моделюють ситуації високого навантаження, щоб виявити можливі проблеми з продуктивністю або стабільністю. Для навчальних платформ важливо забезпечити стабільну роботу під час масових заходів або пікових періодів користування.

Застосування цих методів тестування сприяє забезпеченню високої якості та надійності веб-застосунків. Важливо адаптувати методи тестування до конкретних потреб платформи, враховуючи її специфіку та вимоги користувачів. Комплексний підхід до тестування дозволяє виявити широкий спектр проблем та забезпечити високу якість продукту, що є ключовим фактором успіху для будь-якої онлайн платформи.

4.2 Тестування веб-застосунку

Для початку тестування необхідно запустити серверну платформу, яка дозволить протестувати веб-застосунок за допомогою локального хоста або веб-хостингу. Далі необхідно провести функціональне тестування веб-застосунку. Воно

включає в себе перевірку введеня та виведення даних, реакції на некоректні вхідні дані, відповідності бізнес логіки [18].

Далі проводиться тестування продуктивності роботи веб-застосунку, а саме оцінка швидкості встановлення з’єднання між веб-застосунком та сервером, завантаження елементів, роботи основних функцій, доступу до інших сторінок. Це дозволить оцінити використання ресурсів веб-браузера.

Після проведення тестування можна провести загальну оцінку роботи веб-застосунку. Усі результати повинні бути записані у відповідну документацію, яка використовуватиметься для вдосконалення веб-застосунку, його оптимізації та забезпечення безперебійної роботи.

Результат тестування продуктивності зображено на рисунку 4.1.

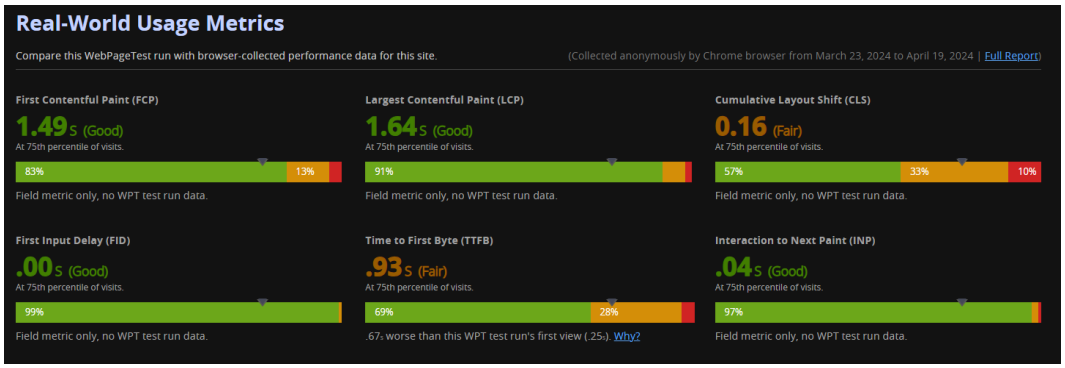


Рисунок 4.1 – Тестування швидкодії веб-застосунку

Наступним кроком буде перевірка вікна входу користувача. Якщо користувач увів коректні дані та авторизація пройшла успішно, вікно входу закриється, а на панелі навігації замість кнопки «Login» буде «Особистий кабінет». Але якщо буде введено неправильні дані для авторизації, то користувачу висвітиться повідомлення про помилку. Результат тестування введення неправильних даних для авторизації зображено на рисунку 4.2

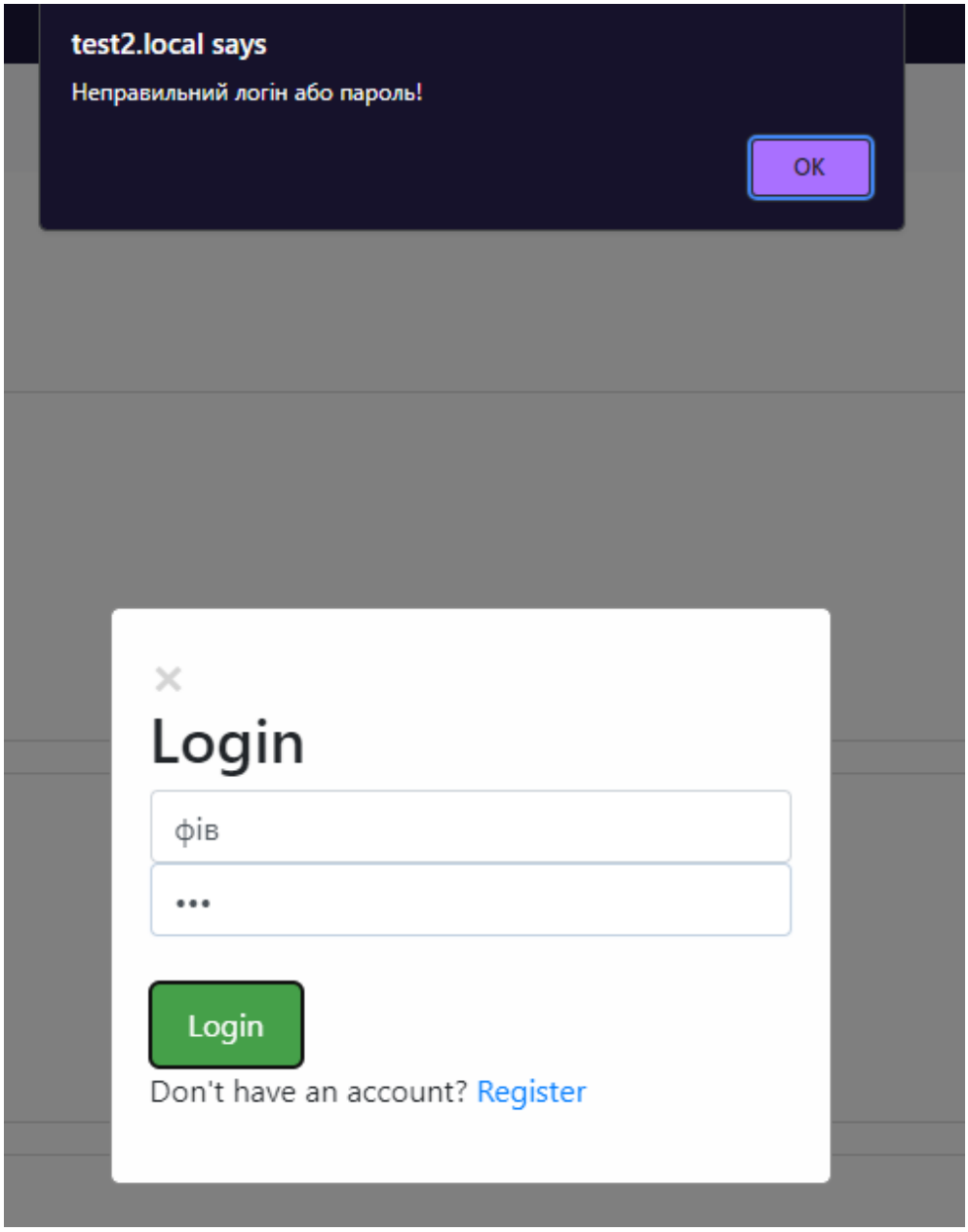


Рисунок 4.2 – Авторизація користувача

Далі йде вікно реєстрації. Якщо користувач вводить коректні дані, то система реєструє його, дозволяючи далі авторизуватися. Якщо користувач спробує ввести некоректні дані, або залишити поля пустими, то йому висвітиться повідомлення про помилку. Результат введення неправильних даних для реєстрації зображено на рисунку 4.3.

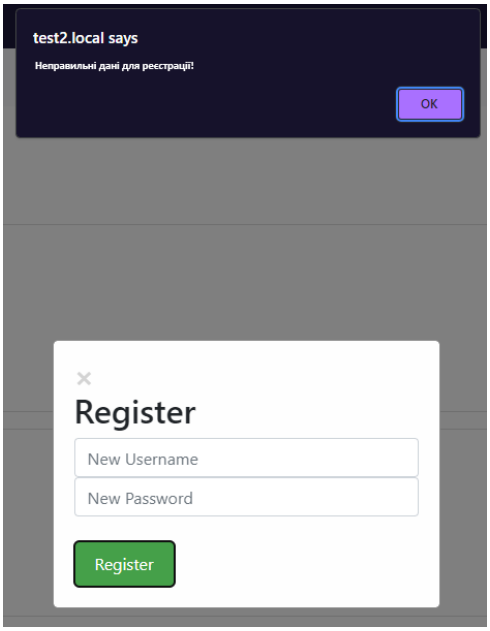


Рисунок 4.3 – Реєстрація користувача

Після перевірки реєстрації та авторизації йде перевірка особистого кабінету. До нього можна перейти авторизувавшись та натиснувши на посилання на особистий кабінет у панелі навігації. У ньому відображається «Профіль користувача (+логін)» користувача та список прочитаних керівництв і виконаних завдань. Якщо користувач ще не позначив жодне керівництво, як прочитане, та жодне завдання, як виконане, то на місці списку буде повідомлення «Ви ще не прочитали жодного керівництва та не виконали жодного завдання». Також є можливість розлогінитися, натиснувши кнопку «Logout». Результат тестування зображено на рисунку 4.4.

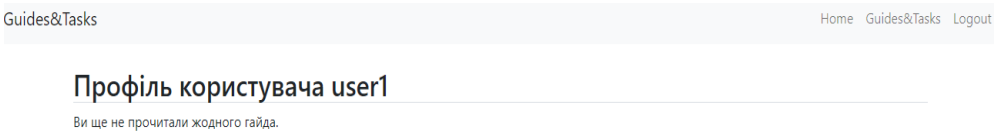


Рисунок 4.4 – Особистий кабінет користувача

Далі йде перевірка отримання доступу до особистого кабінету користувача користувачем, який не авторизований. Для цього необхідно розлогінитися з аккаунту користувача і ввести в адресній стрічці браузера посилання на особистий кабінет. В такому випадку адекватною відповіддю системи вважатиметься відображення сторінки, яка не міститиме інформацію користувача та список позначених керівництв і виконаних завдань, а натомість демонструватиме повідомлення, яка повідомлятиме, що користувач не авторизований, що й зображено на рисунку 4.5

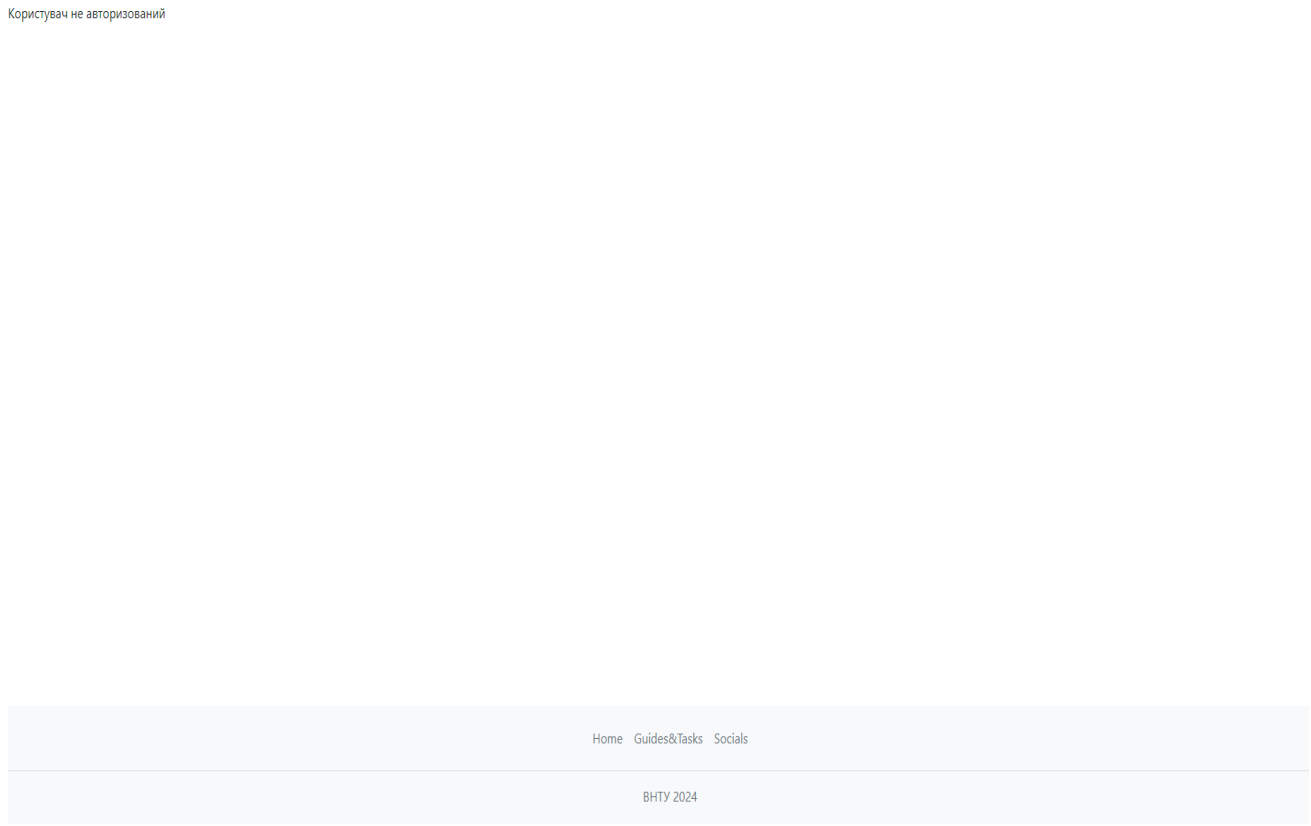


Рисунок 4.5 – Реакція на спробу переходу до особистого кабінету неавторизованим користувачем

Далі йде перевірка проходження опитування. Воно буде запропоноване користувачу при обранні дисципліни вище списку керівництв та завдань

дисципліни, якщо він усе ще не проходив його. Для його проходження користувачу необхідно буде обирати пункти, а потім надіслати відповіді для їхнього аналізу. Після закінчення проходження опитування йому буде висвітлене повідомлення про успішне проходження та відображено його поточний рівень навичок. Результат тестування проходження опитування зображено на рисунку 4.6.

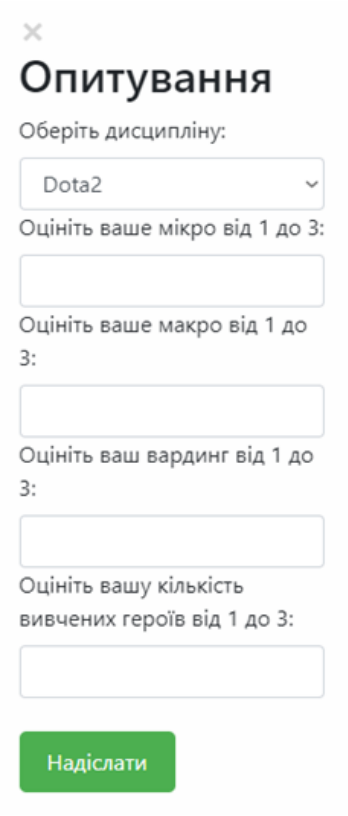


Рисунок 4.6 – Проходження опитування

Далі йде перевірка коректності відображення списку керівництв та завдань. Для цього користувачу необхідно обрати ігрову дисципліну на панелі навігації. Якщо користувач авторизований, то список буде складатися з керівництв та завдань, які рівні або менші, ніж його рівень вмінь, який він отримав після проходження опитування. Якщо користувач не авторизований, або не пройшов опитування, то йому буде згенеровано список керівництв та завдань усіх рівнів. Також, список містить панель сортування, яка дозволяє відсортувати керівництва

та завдання за складністю, датою додавання, або в алфавітному порядку. Результат відображення списку керівництв та завдань для неавторизованого користувача зображено на рисунку 4.7.

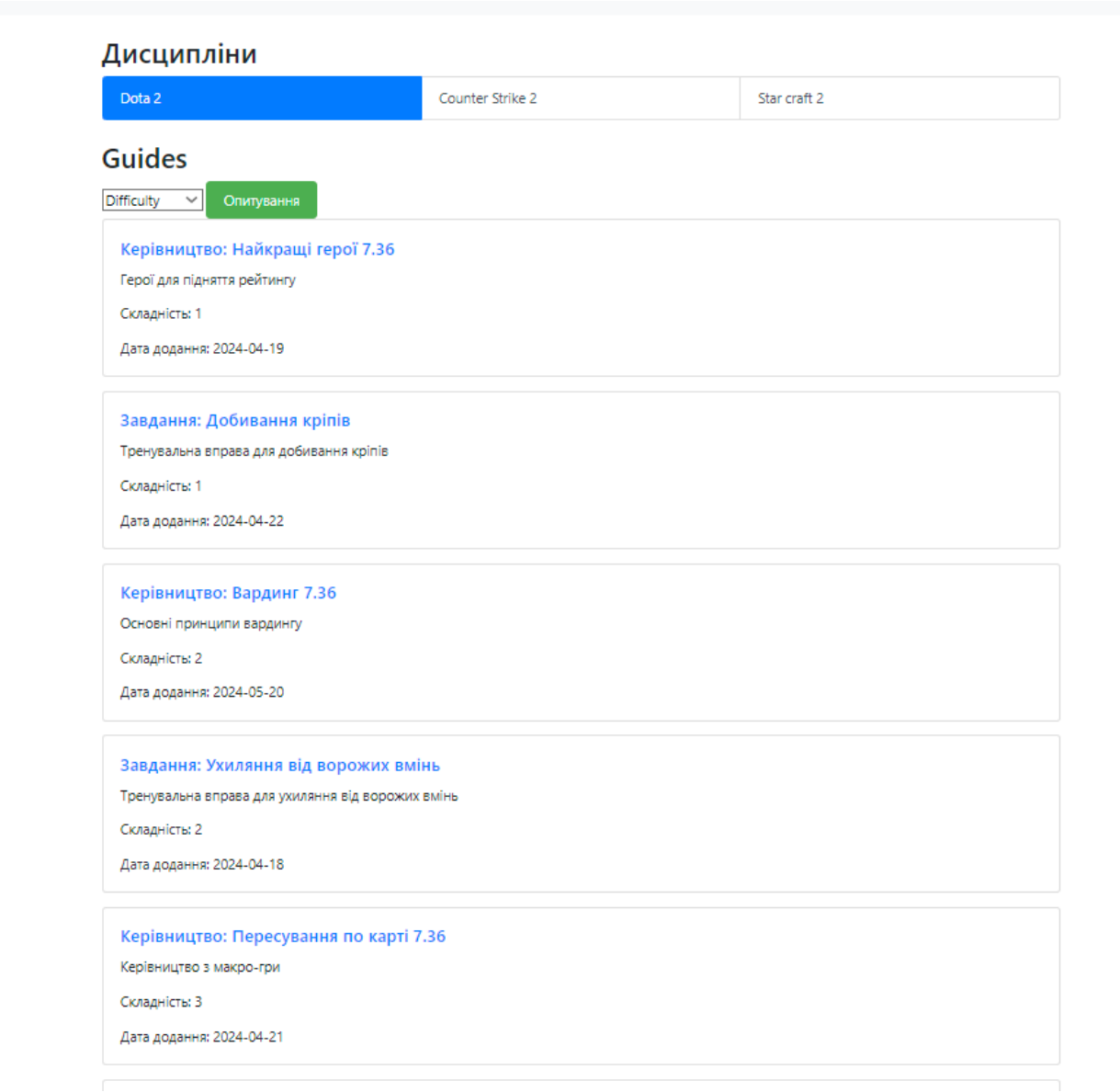


Рисунок 4.7 – Список керівництв неавторизованого користувача

Далі йде перевірка роботи сторінки керівництва та завдання. Натиснувши на назву керівництва або завдання, користувач потрапляє на відповідну сторінку обраного керівництва або завдання. У ній відображається назва керівництва або

завдання, опис до нього, посилання на навчальне відео на youtube, посилання, яке повертає користувача на сторінку зі списком завдань та керівництв, та кнопка «Mark as Read» (якщо це керівництво) або «Mark as done» (якщо це завдання). При натисканні «Mark as Read» на сторінці керівництва авторизованому користувачу повинно бути висвітлено повідомлення, що він відзначив керівництво, як прочитане, та після того його повинно додати до особистого кабінету у відповідний список. Результат відображення сторінки керівництва та позначення його, як прочитане, зображено на рисунку 4.8.

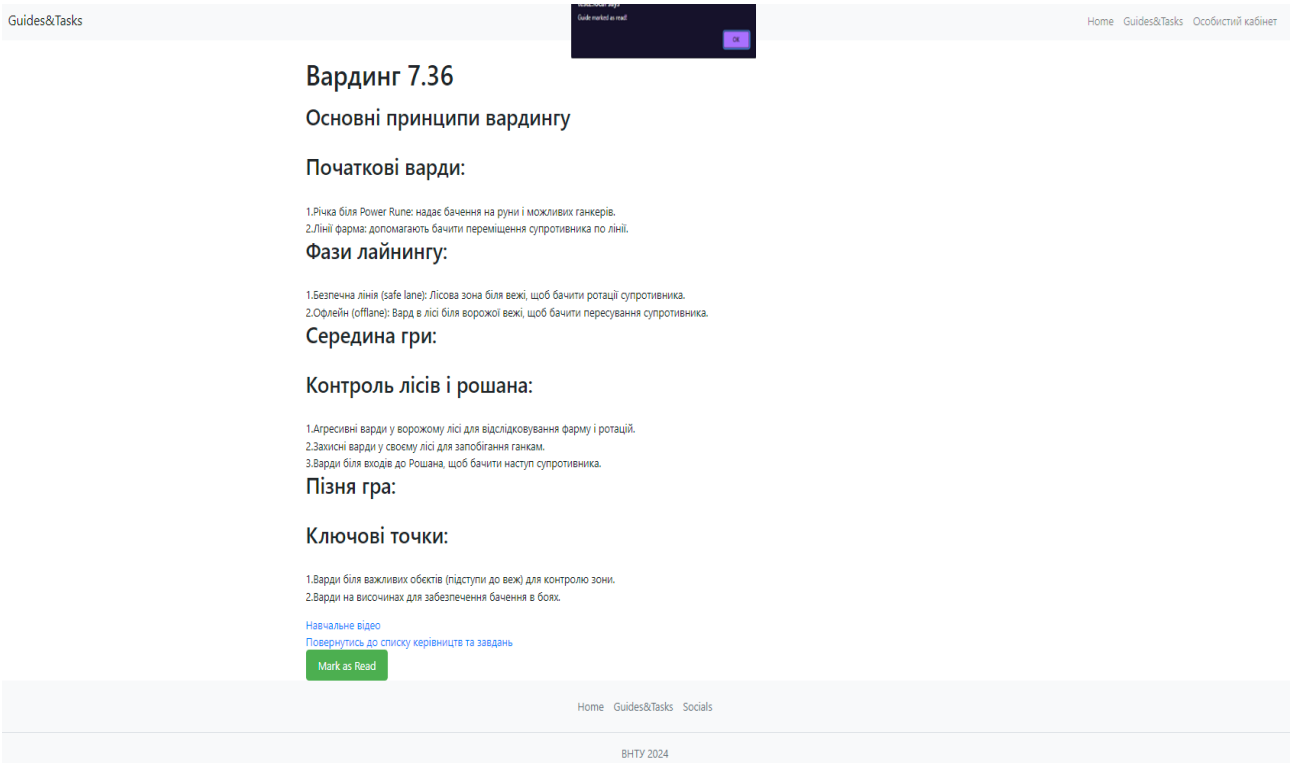


Рисунок 4.8 – Сторінка керівництва та повідомлення про позначення

Наступним кроком буде перевірка реакції системи на спробу зайти на сторінку керівництва або завдання, якого не існує. Для її перевірки в адресній строці браузера потрібно замінити id (унікальний код) керівництва або завдання на неіснуючий. Адекватною відповіддю системи буде вважатися відображення

сторінки з повідомленням про відсутність даного керівництва, що й зображено на рисунку 4.9.

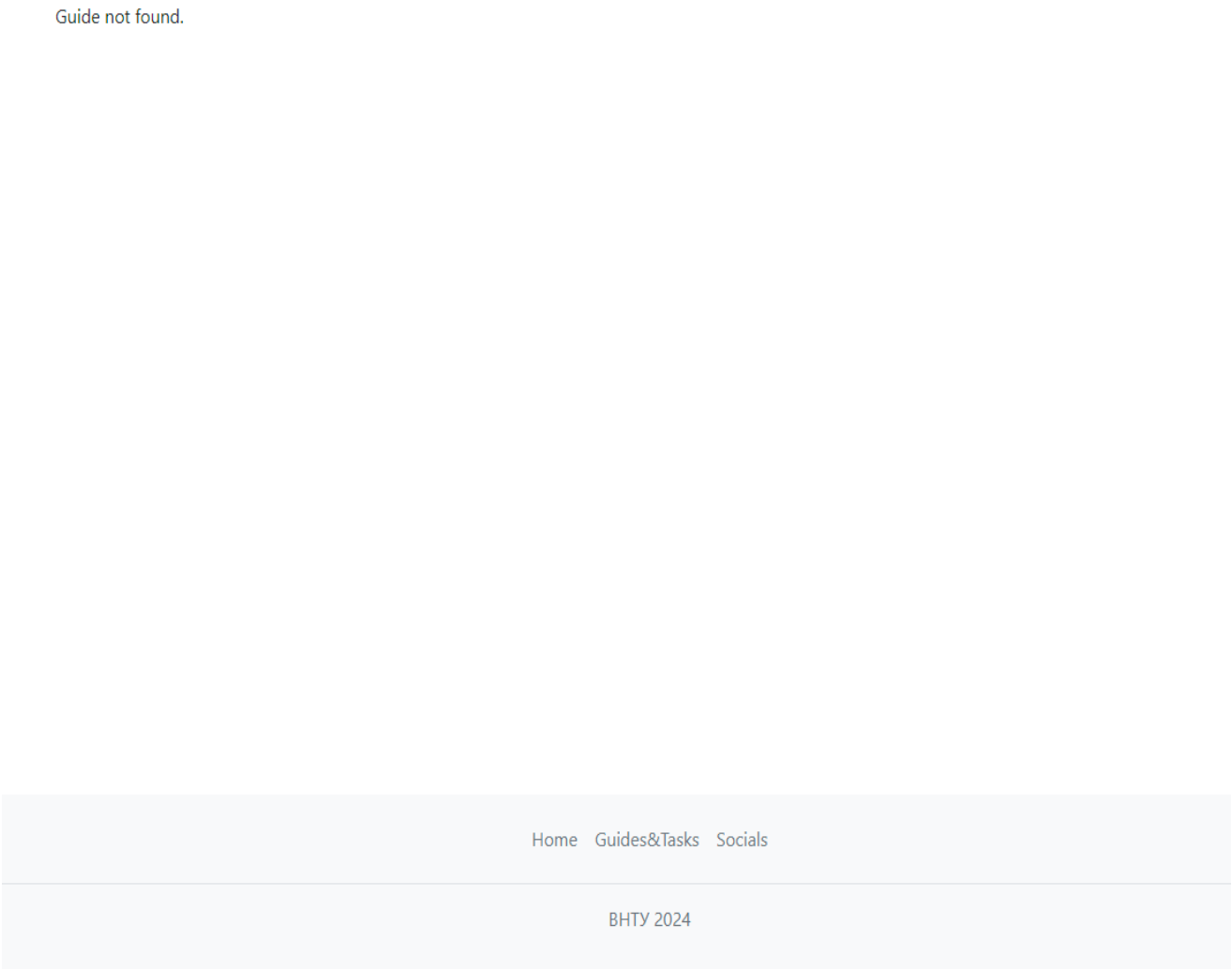


Рисунок 4.9 – Реакція на спробу переходу до неіснуючого керівництва

Отже, було проведено тестування роботи веб-застосунку. Помилок виявлено не було. Тестування продуктивності продемонструвало хороші показники.

4.3 Розробка інструкції користувача

При розробці інструкції користувача необхідно дотримуватись чітких принципів. Кожен розділ інструкції повинен містити чіткі та послідовні кроки, та

бажано, щоб вони були проілюстровані скріншотами або іншими візуальними матеріалами. Це допомагає користувачам краще зрозуміти процеси та швидше засвоїти інформацію. Крім того, важливо включити розділи з поширеними питаннями та вирішенням можливих проблем, що можуть виникнути під час роботи з програмою.

На етапі написання тексту інструкції слід використовувати просту та зрозумілу мову, уникаючи складної термінології та технічного жаргону. Важливо забезпечити логічну послідовність викладу та надавати конкретні приклади, що ілюструють кожен крок. Використання активних дієслів та позитивного тону також сприяє покращенню зрозумілості та залученості користувачів.

Також необхідно вказати мінімальну та рекомендовану конфігурацію комп’ютеру, оскільки користувачу необхідно знати, чи його система відповідає вимогам для запуску застосунку.

Дана інструкція користувача передбачає визначення технічних вимог для запуску програмного продукту. Деталі мінімальної та рекомендованої конфігурації комп’ютера наведені в таблицях 4.10 та 4.11.

Таблиця 4.10 – Мінімальна конфігурація

ОС	Windows 10/11 (64-bit), macOS 10.14 (Mojave) або Ubuntu 18.04 LTS (64-bit)
Тип процесора	Intel Core I3-2100 3.1 GHz або AMD fx 4300K 3.8 GHz
Об’єм оперативної пам’яті	4 GB
Пропускна здатність мережі	10 Mbps

Таблиця 4.11 – Рекомендована конфігурація

ОС	Windows 10/11 (64-bit), macOS 10.14 (Mojave) або Ubuntu 18.04 LTS (64-bit)
Тип процесора	Intel Pentium G6400 4.0 GHz або AMD Athlon 3000G 3.5 GHz;
Об’єм оперативної пам’яті	8 GB
Пропускна здатність мережі	100 Mbps

Після запуску веб-застосунку користувач бачить головну сторінку сайту, де надано інформацію про сайт. Далі він повинен обрати дисципліну, за допомогою панелі навігації. Обравши дисципліну йому пропонується авторизуватися або зареєструватися. Після авторизації користувач може пройти опитування для визначення поточного рівня знань та навичок в обраній ігровій дисципліні. Після проходження опитування, він отримує список доступних керівництв та завдань, які відповідають його поточному визначеному рівню. Обравши керівництво або завдання, користувач повинен прочитати зміст керівництва або завдання та переглянути навчальні відео та знімки екрана, якщо вони є. Якщо це керівництво, то користувач, засвоївши матеріал, повинен відзначити його, як переглянуте. Якщо це завдання, то, після безпосереднього виконання завдання в ігровій дисципліні, користувач повинен відзначити його, як виконане. Відзначені керівництва та завдання відображаються у його особистому кабінеті, тому він може зайти туди та переглянути їх.

4.4 Висновки

У четвертому розділі було проведено тестування програмних модулів програмного застосунку для тренування кіберспортсменів. Було перевірено коректність роботи веб-застосунку. Також, було наведено мінімальну та

рекомендовану конфігурації системи та розроблено інструкцію користувача з використання веб-застосунку.

ВИСНОВКИ

У бакалаврській дипломній роботі було розроблено метод для навчання в ігрових дисциплінах та програмний застосунок для тренування кіберспортсменів.

Для розробки програмного застосунку у вигляді веб-застосунку для тренування кіберспортсменів було використано середовище програмної розробки VisualStudio.

Під час виконання бакалаврської дипломної роботи було проаналізовано стан проблеми на сьогоднішній день. Було розглянуто основні аналоги програмного продукту. Було визначено їхні недоліки та порівняно з власним програмним продуктом. Базуючись на порівнянні було встановлено основні задачі бакалаврської дипломної роботи.

Під час аналізу технологій розробки було обґрунтовано вибір мов програмування HTML, CSS, JavaScript та PHP і бібліотек BootsTrap та jQuery.

У бакалаврській дипломній роботі було зроблено наступне:

- визначено загальний алгоритм системи;
- розроблено метод навчання в ігрових ситуаціях;
- розроблено зручний та адаптивний графічний інтерфейс програмного застосунку;
- розроблено алгоритми роботи застосунку;
- розроблено програмні модулі роботи застосунку;
- розроблено програмний застосунок для тренування кіберспортсменів;
- проведено тестування застосунку згідно поставлених задач.

Було розроблено метод навчання в ігрових ситуаціях та схему загального алгоритму роботи програмного застосунку для тренування кіберспортсменів. Також, було розроблено модель роботи застосунку у вигляді діаграми діяльності.

Тестування програми довело повну працездатність даного застосунку та відповідність поставленому технічному завданню. Також було розроблено інструкцію користувача.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кіберспорт. URL: https://uk.wikipedia.org/wiki/Кіберспорт (дата звернення: 10.04.2024).

2. Створення та редагування веб-ресурсів для розширення застосунку. URL: https://learn.microsoft.com/uk-ua/dynamics365/customerengagement/on-premises/customize/create-edit-web-resources?view=op-9-1 (дата звернення: 10.04.2024).

3. Що таке CSS. URL: https://css.in.ua/article/shcho-take-html_10 (дата звернення: 10.04.2024).

4. Індустрія відеоігор. URL: https://uk.wikipedia.org/wiki/Індустрія_відеоігор (дата звернення: 15.04.2024).

5. Етапи створення веб сайтів. URL: https://webtune.com.ua/statti/web-rozrobka/etapy-stvorennya-veb-sajtiv/ (дата звернення: 15.04.2024).

6. Гайди в іграх. URL: https://gamekombo.com/guides/ (дата звернення: 20.04.2024).

7. The Gamer. URL: https://www.thegamer.com (дата звернення: 22.04.2024).

8. Gamer Sensei. URL: https://www.gamersensei.com (дата звернення: 26.04.2024).

9. Що таке API. URL: https://hostiq.ua/blog/ukr/what-is-api/ (дата звернення: 01.05.2024).

10. Юзабіліті сайту. URL: https://webtune.com.ua/statti/web-rozrobka/yuzabiliti-sajtu/ (дата звернення: 10.05.2024).

11. HTML і CSS: Що це, кому та для чого воно потрібно. URL: https://goit.global/ua/articles/html-i-css-shcho-tse-komu-ta-dlia-choho-potribno/ (дата звернення: 04.05.2024).

12. Що таке BootsTrap? URL: https://itmaster.biz.ua/programming/web-prohramuvannia/bootstrap.html (дата звернення: 10.05.2024).

13. HTML JavaScript. URL: https://w3schoolsua.github.io/html/html_scripts.html#gsc.tab=0 (дата звернення: 07.05.2024).

14. Що таке jQuery та де вона використовується. URL: https://dzudzylo.com/javascript/shho-take-jquery-ta-de-vona-vykorystovuyetsya.html (дата звернення: 13.05.2024).

15. Що таке AXAJ? URL: https://avada-media.ua/ua/ajax/ (дата звернення: 15.05.2024).

16. Що таке PHP? URL: https://hyperhost.ua/uk/wiki/chto-takoe-php (дата звернення: 09.05.2024).

17. PHP проти JavaScript – різниця між ними. URL: https://www.guru99.com/uk/php-vs-javascript.html (дата звернення: 19.05.2024).

18. Localhost: що це таке, і як він використовується у розробці. URL: https://spacelab.ua/articles/localhost-sho-ce-take-i-yak-vin-vikoristovuyetsya-u-rozrobci/ (дата звернення: 20.05.2024).

ДОДАТКИ

60

Додаток А – Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри ПЗ
д.т.н., професор
_____ О. Н. Романюк
«12» березня 2024 р.

Технічне завдання
на бакалаврську дипломну роботу «Розробка методу навчання в ігрових ситуаціях і програмного засобу для тренування кіберспортсменів» за спеціальністю 121 – Інженерія програмного забезпечення

Керівник бакалаврської дипломної роботи:
_____ к.т.н., доцент каф. ПЗ Г. Б. Ракитянська
«12» березня 2024 р.
Виконав:
_____ студент гр. 3ПІ-20б І. І. Двойнос
«12» березня 2024 р.

Вінниця – 2024 року

1. Найменування та галузь застосування

Бакалаврська дипломна робота: «Розробка методу навчання в ігрових ситуаціях і програмного засобу для тренування кіберспортсменів».

Галузь застосування – кіберспорт.

2. Підстава для розробки

Підставою для виконання бакалаврської дипломної роботи (БДР) є індивідуальне завдання на БДР та наказ №80 від «11» березня 2024 р. ректора ВНТУ про закріплення тем БДР.

3. Мета та призначення розробки

Метою дослідження є підвищення якості тренування кіберспортсменів за рахунок створення методу навчання в ігрових ситуаціях за рахунок використання власного навчального методу, який базується на математичній моделі, що дозволяє структуризувати процес тренування.

4. Вихідні дані для проведення НДР

Перелік основних літературних джерел, на основі яких буде виконуватись БДР.

1. Джеймс Пол Гі. Навчання через дію: Реальний зв’язок між інноваціями, заробітками і багатством. 2003.-С.188-199.
2. Джейн Макґоніґал. Реальність зламана: Чому ігри роблять нас кращими і як вони можуть змінити світ. 2011.-С.220-250.

5. Технічні вимоги

Вихідні дані до роботи: список дисциплін, список керівництв та завдань, сторінки керівництв і завдань, з можливістю їх додавання до списку прочитаних або виконаних.

6. Конструктивні вимоги

Графічна та текстова документація повинна відповідати діючим стандартам України.

7. Перелік технічної документації, що пред'являється по закінченню робіт:

1. Пояснювальна записка до БДР.
2. Технічне завдання.
3. Лістинги програми

8. Вимоги до рівня уніфікації та стандартизації

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ

9. Стадії та етапи розробки:

№ з\п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи
1	Аналіз підходів до методів навчання в ігрових ситуаціях та програмних застосунків для тренування кіберспортсменів	13.03.24 –22.03.24
2	Розробка математичної моделі методу навчання в ігрових ситуаціях та інтерфейсу програмного застосунку	13.03.24 –22.03.24
3	Розробка моделі та алгоритму роботи програмного застосунку	25.03.24 –19.04.24
4	Розробка програмного застосунку	22.04.24 –10.05.24
5	Тестування програмного забезпечення	13.05.24 – 24.05.24
6	Оформлення матеріалів до захисту БДР	27.05.24 – 03.06.24

10. Порядок контролю та прийняття

Виконання етапів бакалаврської дипломної роботи контролюється керівником згідно з графіком виконання роботи. Прийняття бакалаврської дипломної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком.

64

Додаток Б – Протокол перевірки на плагіат

ПРОТОКОЛ
ПЕРЕВІРКИ БАКАЛАВРСЬКОЇ ДИПЛОМНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка методу навчання в ігрових ситуаціях і програмного засобу для тренування кіберспортсменів

Тип роботи: БДР

Підрозділ: кафедра програмного забезпечення, ФІТКІ

Науковий керівник: Ракитянська Г.Б.

Оригінальність	94,2%
Схожість	5,8%

Аналіз звіту подібності

▪ **Запозичення, виявлені у роботі, оформленні коректно і не містять ознак плагіату.**

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цілісності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховання недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Черноволик Г.О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck

Автор роботи _____ Двойнос І.І.

Керівник роботи _____ Ракитянська Г.Б.

Додаток В – Лістинг програми

Index.php:

```
<?php session_start(); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Guides&Tasks</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet" href="styles.css">
</head>
<body class="d-flex flex-column min-vh-100">

    <header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="home.php">Guides&Tasks</a>
            <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav"
                aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarNav">
                <ul class="navbar-nav ml-auto">
                    <li class="nav-item">
```

```
                <a class="nav-link" href="home.php">Home</a>
            </li>
            <li class="nav-item">
                <a class="nav-link active" href="index.php">Guides&Tasks</a>
            </li>
            <li class="nav-item">
        <?php
        if (isset($_SESSION['username'])) {
            echo '<a class="nav-link" href="user_profile.php">Особистий
кабінет</a>';
        } else {
            echo '<button id="loginButton" class="nav-link">Login</button>';
        }
        ?>
            </li>
          </ul>
        </div>
      </nav>
    </header>

    <!-- Модальне вікно для входу -->
    <div id="loginModal" class="modal">
      <div class="modal-content">
        <span class="close">&times;</span>
        <h2>Login</h2>
        <form id="loginForm">
          <input type="text" id="username" placeholder="Username"
class="form-control">
```

```
                <input type="password" id="password" placeholder="Password"
class="form-control">
                <br>
                <button type="submit" id="loginSubmit">Login</button>
            </form>
            <p>Don't have an account? <a href="#"
id="registerLink">Register</a></p>
        </div>
    </div>

    <!-- Модальне вікно для реєстрації -->
    <div id="registerModal" class="modal">
        <div class="modal-content">
            <span class="close">&times;</span>
            <h2>Register</h2>
            <form id="registerForm">
                <input type="text" id="newUsername" placeholder="New Username"
class="form-control">
                <input type="password" id="newPassword" placeholder="New
Password" class="form-control">
                <br>
                <button type="submit" id="registerSubmit">Register</button>
            </form>
        </div>
    </div>

    <!-- Модальне вікно опитування -->
    <div id="surveyModal" class="modal">
```

```
            <div class="modal-content">
                <span id="closeSurv" class="close">&times;</span>
                <h2>Опитування</h2>
                <form id="surveyForm">
                    <label for="DisciplineSelect">Оберіть дисципліну:</label>
                    <select id="DisciplineSelect" class="form-control">
                        <option value="Dota2">Dota2</option>
                        <option value="CS 2">CS 2</option>
                        <option value="Valorant">Valorant</option>
                    </select>
                    <label for="formMicro">Оцініть ваше мікро від 1 до 3:</label>
                    <input type="number" id="formMicro" name="formMicro"
class="form-control" required>
                    <label for="formMacro">Оцініть ваше макро від 1 до 3:</label>
                    <input type="number" id="formMacro" name="formMacro"
class="form-control" required>
                    <label for="formWarding">Оцініть ваш вардинг від 1 до 3:</label>
                    <input type="number" id="formWarding" name="formWarding"
class="form-control" required>
                    <label for="heroPool">Оцініть вашу кількість вивчених героїв від 1
до 3:</label>
                    <input type="number" id="heroPool" name="heroPool" class="form-
control" required>
                    <br>
                    <button type="submit">Надіслати</button>
                </form>
            </div>
        </div>
```

```
<div class="container mt-4">
   <h2>Guides</h2>
<select id="sortBy">
      <option value="difficulty">Difficulty</option>
      <option value="alphabetical">Alphabetical</option>
      <option value="date">Date Added</option>
   </select>
   <button id="openSurveyButton">Опитування</button>
   <div id="guideList">
      <!-- Guides will be dynamically inserted here -->
   </div>
</div>

<footer id="footer" class="py-3 mt-auto bg-light">
<ul class="nav justify-content-center border-bottom pb-3 mb-3">
  <li class="nav-item"><a href="home.php" class="nav-link px-2 text-
muted">Home</a></li>
  <li class="nav-item"><a href="index.php" class="nav-link px-2 text-
muted">Guides&Tasks</a></li>
  <li class="nav-item"><a href="#" class="nav-link px-2 text-
muted">Socials</a></li>
</ul>
<p class="text-center text-muted">ВНТУ 2024</p>
</footer>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></scri
pt>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<script src="scripts.js"></script>
<script>
// Функція для обробки відповіді після входу користувача
function handleLoginResponse(response) {
    if (response === "success") {
        isLoggedIn = true; // Оновлення статусу авторизації
        location.reload();

    } else {
        alert("Неправильний логін або пароль!");
    }
}

// Обробник події для подання форми входу
$("#loginForm").submit(function(event) {
    event.preventDefault(); // Запобігаємо стандартній поведінці форми

    // Отримуємо дані з форми
    var username = $("#username").val();
    var password = $("#password").val();

    // Викликаємо AJAX-запит для перевірки логіну та паролю
```

```
        $.post("login.php", { username: username, password: password },
function(response) {
            handleLoginResponse(response); // Обробка відповіді від сервера
        });
    });
    var surveyModal = document.getElementById('surveyModal');

    // Отримуємо посилання на кнопку, яка відкриває модальне вікно
    var openSurveyButton = document.getElementById("openSurveyButton");

    // Отримуємо посилання на елемент закриття модального вікна
    var closeSurv = document.getElementById("closeSurv");

    // Функція для відкриття модального вікна
    function openSurveyModal() {
        surveyModal.style.display = "block";
    }

    // Функція для закриття модального вікна
    function closeSurveyModal() {
        surveyModal.style.display = "none";
    }

    // Відкриття модального вікна при натисканні на кнопку
    openSurveyButton.onclick = function () {
        openSurveyModal();
    }
```

```
closeSurv.onclick = function () {
    closeSurveyModal();
}

// Закриття модального вікна при натисканні на хрестик або поза ним
/*for (var i = 0; i < closeSurv.length; i++) {
    closeSurv[i].onclick = function () {
        closeSurveyModal();
    }
}*/

/*window.onclick = function (event) {
    if (event.target == surveyModal) {
        closeSurveyModal();
    }
}*/

// Надсилання форми опитування
$('#surveyForm').submit(function (event) {
    event.preventDefault();

    // Отримування даних форми
    var formData = {
        disciplineSelect: $('#disciplineSelect').val(),
        formMicro: $('#formMicro').val(),
        formMacro: $('#formMacro').val(),
        formWarding: $('#formWarding').val(),
        heroPool: $('#feedback').val()
```

```
            };

            // Відправка даних на сервер для обробки опитування
            // Треба допилити код

            // Закриття модального вікна
            alert("Опитування пройдено успішно!");
            closeSurveyModal();

        });

    </script>

    </body>
    </html>
```

guide.php:

```
<?php session_start();?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Guide</title>
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <link rel="stylesheet" href="styles.css">
</head>
<body class="d-flex flex-column min-vh-100">
    <header>
        <nav class="navbar navbar-expand-lg navbar-light bg-light">
            <a class="navbar-brand" href="index.php">Guides&Tasks</a>
```

```
                <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarNav"
                    aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
                  <span class="navbar-toggler-icon"></span>
                </button>
                <div class="collapse navbar-collapse" id="navbarNav">
                  <ul class="navbar-nav ml-auto">
                    <li class="nav-item">
                      <a class="nav-link" href="home.php">Home</a>
                    </li>
                    <li class="nav-item">
                      <a class="nav-link" href="index.php">Guides&Tasks</a>
                    </li>
                    <li class="nav-item">
                      <?php
                      /*// Початок сесії
                      //session_start();
                      $isLoggedIn = isset($_GET['loggedIn']) && $_GET['loggedIn']
=== 'true';
                      // Перевірка авторизації
                      if (isset($_SESSION['user_id'])) {
                        // Якщо користувач авторизований, відображаємо посилання
на особистий кабінет
                        echo '<a href="user_profile.php" class="nav-link">Особистий
кабінет</a>';
                        // Відображення посилання на особистий кабінет в інших
місцях, якщо потрібно
                      } else {
                        // Якщо користувач не авторизований, відображаємо кнопку
для входу
                        echo '<button id="loginButton" class="nav-
link">Login</button>';
                      }*/
              if (isset($_SESSION['username'])) {
                echo '<a class="nav-link" href="user_profile.php">Особистий
кабінет</a>';
              } else {
                echo '<button id="loginButton" class="nav-link">Login</button>';
              }

                      ?>
```

```
                    </li>
                </ul>
            </div>
        </nav>
    </header>

    <!-- Модальне вікно для входу -->
    <div id="loginModal" class="modal">
        <div class="modal-content">
            <span class="close">&times;</span>
            <h2>Login</h2>
            <form id="loginForm">
                <input type="text" id="username" placeholder="Username"
class="form-control">
                <input type="password" id="password" placeholder="Password"
class="form-control">
                <br>
                <button type="submit" id="loginSubmit">Login</button>
            </form>
            <p>Don't have an account? <a href="#"
id="registerLink">Register</a></p>
        </div>
    </div>

    <!-- Модальне вікно для реєстрації -->
    <div id="registerModal" class="modal">
        <div class="modal-content">
            <span class="close">&times;</span>
            <h2>Register</h2>
            <form id="registerForm">
                <input type="text" id="newUsername" placeholder="New Username"
class="form-control">
                <input type="password" id="newPassword" placeholder="New
Password" class="form-control">
                <br>
                <button type="submit" id="registerSubmit">Register</button>
            </form>
        </div>
    </div>
  <!--    <a href="index.php">Back to Guides List</a>
    <h1>Title of the Guide</h1>
    <button id="markAsRead">Mark as Read</button>
```

```
<p>Description of the Guide</p>
<a href="video_link">Watch the video</a>
<br>-->
<div id="guidePageGuide">
    <div class="container mt-4">
<?php
$guides = array(
    array('id' => 1, 'title' => 'Guide 1', 'description' => 'Description for Guide 1',
'video_link' => 'https://www.youtube.com', 'difficulty' => 2, 'date_added' => '2024-04-
20'),
    array('id' => 2, 'title' => 'Guide 2', 'description' => 'Description for Guide 2',
'video_link' => 'https://www.youtube.com', 'difficulty' => 1, 'date_added' => '2024-04-
19'),
    array('id' => 3, 'title' => 'Guide 3', 'description' => 'Description for Guide 3',
'video_link' => 'https://www.youtube.com', 'difficulty' => 3, 'date_added' => '2024-04-
21'),
);

// Assuming you're receiving the guide ID in the URL parameter
if (isset($_GET['id])) {
    $guideId = $_GET['id'];
    // Find the guide with the matching ID
    $selectedGuide = array_filter($guides, function ($guide) use ($guideId) {
        return $guide['id'] == $guideId;
    });
    // If guide found, display its details
    if (!empty($selectedGuide)) {
        $selectedGuide = array_values($selectedGuide)[0]; // Extracting the first
(and only) element
        echo '<h1>' . $selectedGuide['title'] . '</h1>';
        echo '<p>' . $selectedGuide['description'] . '</p>';
        echo '<a href="' . $selectedGuide['video_link'] . '"
target="_blank">Навчальне відео</a>';
        ?>
    <br>
    <a href="index.php">Back to Guides&Tasks List</a>
    <br>
    <button id="markAsRead">Mark as Read</button>
<?php
    } else {
        echo 'Guide not found.';
    }
```

```
            } else {
                echo 'Guide ID not provided.';
            }
            ?>
                </div>
            </div>

            <footer id="footer" class="py-3 mt-auto bg-light">
            <ul class="nav justify-content-center border-bottom pb-3 mb-3">
              <li class="nav-item"><a href="home.php" class="nav-link px-2 text-
muted">Home</a></li>
              <li class="nav-item"><a href="index.php" class="nav-link px-2 text-
muted">Guides&Tasks</a></li>
              <li class="nav-item"><a href="#" class="nav-link px-2 text-
muted">Socials</a></li>
            </ul>
            <p class="text-center text-muted">ВНТУ 2024</p>
            </footer>

            <script>
            // Code to mark the guide as read
            document.getElementById("markAsRead").onclick = function() {
              // Your code here to mark the guide as read
              // For example, you can use local storage or send an AJAX request to the
server
              alert("Guide marked as read!");
            };
            </script>
            <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
            <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.4/dist/umd/popper.min.js"></scri
pt>
            <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
            <script src="scripts.js"></script>
        </body>
        </html>
```

login.php:

```
        <?php
        session_start();// Розпочинаємо сесію
```

```
// Зареєстровані користувачі
$registeredUsers = array(
    array(
        'username' => 'user1',
        'password' => 'password1',
        'read_guides' => array()
    ),
    array(
        'username' => 'user2',
        'password' => 'password2',
        'read_guides' => array()
    ),
    array(
        'username' => 'user3',
        'password' => 'password3',
        'read_guides' => array()
    )
);

// Перевіряємо логін та пароль
$isLoggedIn = false;
foreach ($registeredUsers as $user) {
    if ($user['username'] === $inputUsername && $user['password'] ===
$inputPassword) {
        $isLoggedIn = true;
        $_SESSION['username'] = $inputUsername; // Зберігаємо ім'я користувача
в сесії
        break;
    }
}

// Отримуємо дані з POST-запиту
$inputUsername = $_POST['username'];
$inputPassword = $_POST['password'];

// Перевірка логіну та паролю
$isLoggedIn = false;
foreach ($registeredUsers as $user) {
    if ($user['username'] === $inputUsername && $user['password'] ===
$inputPassword) {
        $isLoggedIn = true;
```

```
            $_SESSION['username'] = $inputUsername; // Зберігаємо ім'я користувача
в сесії
            break;
        }
    }

    // Відповідь на запит
    if ($isLoggedIn) {
        echo "success";
    } else {
        echo "error";
    }
    ?>
```

user_profile.php:

```
<!DOCTYPE html>
    <html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>User Profile</title>
        <!-- Посилання на Bootstrap CSS -->
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
        <!-- Посилання на власні стилі -->
        <link rel="stylesheet" href="styles.css">
    </head>
    <body class="d-flex flex-column min-vh-100">
    <?php
    session_start();

    // Зареєстровані користувачі
    $registeredUsers = array(
        array(
            'username' => 'user1',
            'password' => 'password1',
            'read_guides' => array()
        ),
        array(
            'username' => 'user2',
            'password' => 'password2',
            'read_guides' => array()
        ),
```

```
array(
    'username' => 'user3',
    'password' => 'password3',
    'read_guides' => array()
));>
```

Додаток Г – Іллюстративна частина

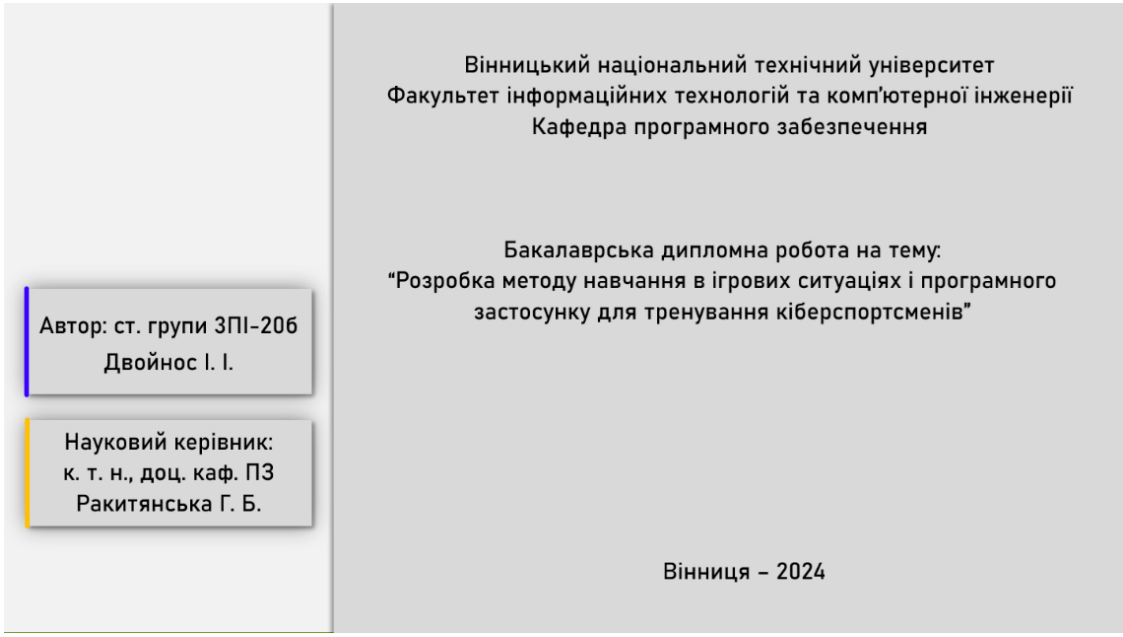


Рисунок Г.1 – Титульний слайд

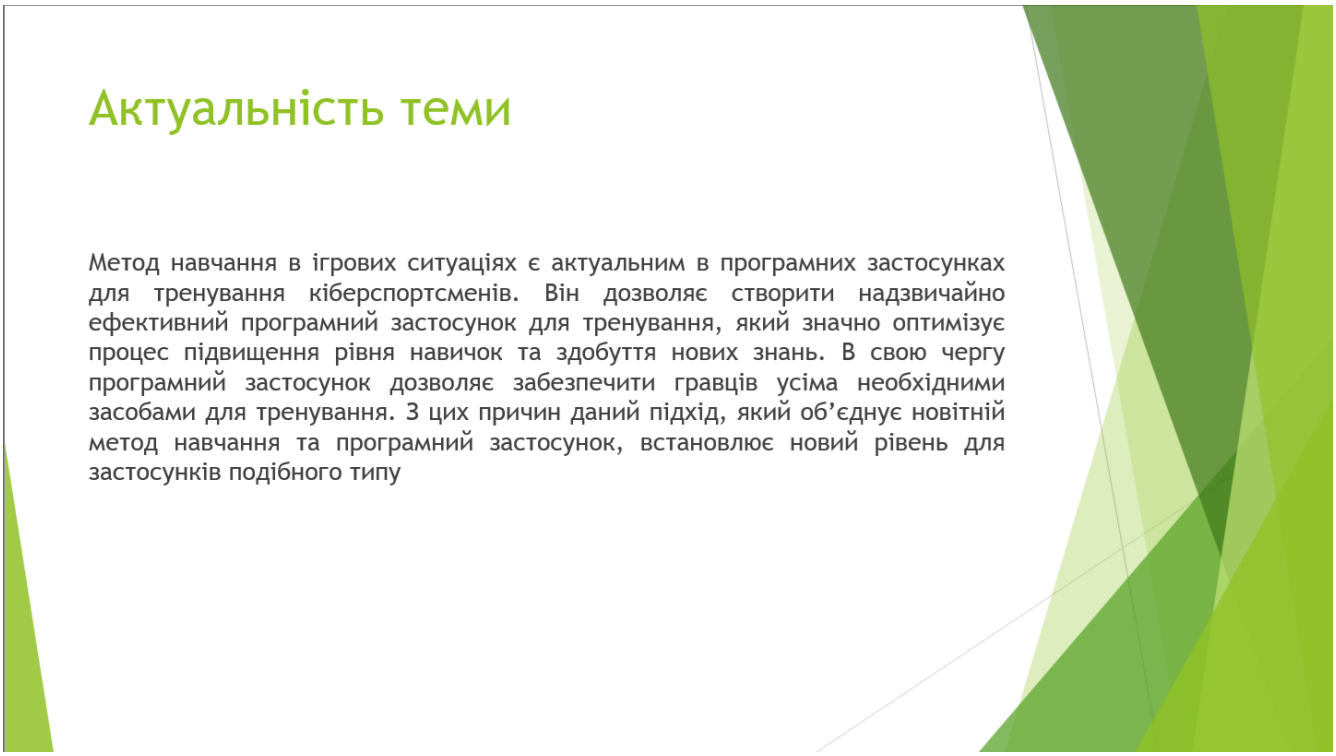


Рисунок Г.2 – Актуальність теми

Мета, об’єкт та предмет дослідження

Метою роботи є підвищення якості навчання кіберспортсменів за рахунок створення методу і програмного засобу навчання в ігрових ситуаціях, що базуються на керівництвах з виконання завдань, які дозволять швидше адаптовуватися залежно від ігрових ситуацій і будуть враховувати найактуальнішу інформацію з ігрових дисциплін для спортсменів різних рівнів підготовки.

Об’єктом дослідження є процес розробки програмного застосунку для тренування кіберспортсменів.

Предметом дослідження є метод навчання в ігрових ситуаціях та програмний застосунок для тренування кіберспортсменів.

Рисунок Г.3 – Мета, об’єкт та предмет дослідження

Завдання

- провести аналіз стану методів навчання в ігрових ситуаціях і програмних застосунків для навчання кіберспортсменів;
- запропонувати новий метод навчання в ігрових ситуаціях, програмний застосунок-реалізацію;
- розробити модель програмного застосунку;
- розробка наступних модулів: модуль відображення керівництв та завдань, модуль авторизації і відображення особистого кабінету;
- провести тестування розроблених алгоритмів та програмного застосунку.

Рисунок Г.4 – Завдання

Наукова новизна

1. Подальшого розвитку набув метод навчання в ігрових ситуаціях, який на відміну від існуючих реалізацій базуватиметься на математичній моделі, що враховує стратегії гри у вигляді керівництв до виконання завдань. Це сприяє швидкому засвоєнню нових ігрових стратегій та тактик, а також оптимізує загальний прогрес кіберспортсменів у змаганнях.
2. Подальшого розвитку набула програмна модель навчання в ігрових ситуаціях для тренування кіберспортсменів. Дана модель дозволяє створити унікальне програмне забезпечення, яке забезпечить персоналізоване навчання та тренування кіберспортсменів на основі індивідуальних навичок і потреб. Адаптивна модель навчання дозволяє точно вимірювати та аналізувати результати кожного тренувального сеансу, що допомагає виявляти слабкі місця гравців та робити висновки щодо їх покращення.

Рисунок Г.5 – Наукова новизна

Практична цінність отриманих результатів

Практична цінність отриманих результатів полягає в розробці вебзастосунку, який здатний надавати користувачам можливість ефективно тренуватися безпосередньо в ігрових ситуаціях, читаючи керівництва та виконуючи завдання, використовуючи браузер. Програмні модулі, розроблені в роботі, можуть бути використані розробниками для створення нових подібних систем, для розширення функціоналу вебзастосунку або алгоритму тренування в цілому та користувачами для тренування та навчання в ігрових ситуаціях.

Рисунок Г.6 – Практична цінність отриманих результатів

Аналоги. Gamepressure

Gamepressure є першим аналогом. Даний ресурс, окрім керівництв, також містить новини, енциклопедію, відео та інше, як і попередній. Але акцент сайту та його головна мета - це керівництва в ігрових дисциплінах. Крім того, щоб упевнитись, що користувачі обрали керівництво до саме тієї гри, яка їм потрібна, у ньому вказано назву гри, дату її виходу, розробника, та додано медіафайли. Також, кожне керівництво та гра мають свою тему на їхньому форумі, що дозволяє здійснювати обмін думками. Тим не менш, у ньому відсутній спосіб визначити поточний рівень вмінь та тренувальні завдання. Тобто він орієнтується на навчання без практичного тренування.

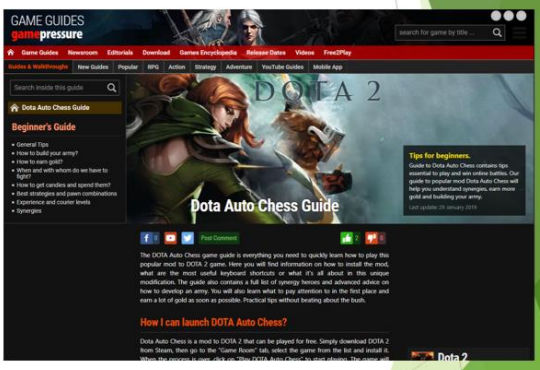


Рисунок Г.7 – Аналоги. Gamepressure

Аналоги. IGN

Другим аналогом є IGN. Це сайт з новинами ігрової індустрії, який містить також і детальні керівництва новачків, як до однокористувацьких, так і до багатокористувацьких ігрових дисциплін. Однак, ці керівництва складаються ігровими журналістами, які не завжди мають навички найвищого рівня в іграх. Даний ресурс є перевіреним часом та досі користується популярністю гравців з усіх куточків світу. Ресурс має недоліки, які схожі до попереднього аналогу. Крім того, як вже було зазначено вище, не усі керівництва складаються професійними гравцями, а частіше ігровими журналістами, яким нерідко бракує вмінь в обраній дисципліні для поглибленого навчання досвідчених гравців. Також містить нав'язливий платний функціонал.

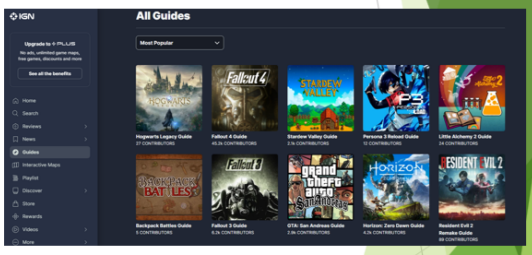


Рисунок Г.8 – Аналоги. IGN

Аналоги.Thegamer

Даний аналог відрізняється від двох попередніх тим, що у ньому зібрані лише керівництва до ігрових дисциплін, які складає кваліфікована команда з високим рівнем знань та навичок у цій сфері. Тим не менш, через це, швидкість викладання навчального матеріалу значно нижча, та кількість кіберспортивних дисциплін не є настільки високою, як у раніше перелічених аналогах.

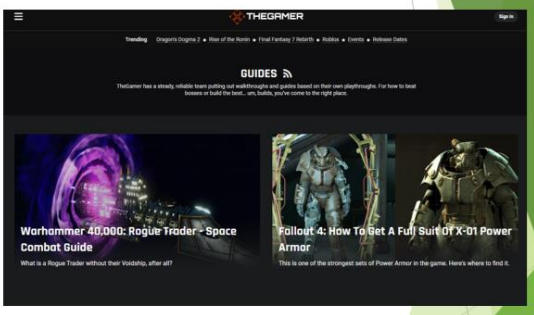


Рисунок Г.9 – Аналоги. Thegamer

Аналоги. Gamer Gains

Даний аналог дозволяє гравцям визначити їхній поточний рівень вмінь та виконувати прості завдання в популярних ігрових дисциплінах, винагороджуючи їх власною валютою, яку можна обмінювати на ігрові предмети. Великим недоліком є відсутність у ньому навчальних матеріалів, які дозволять гравцеві підготуватись до виконання завдань.

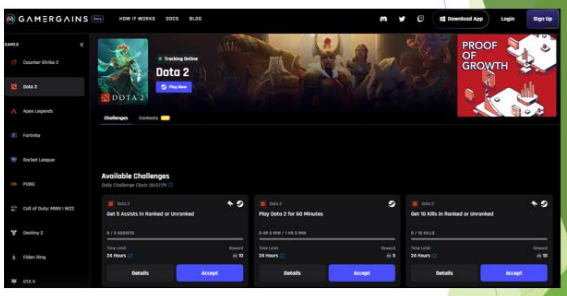


Рисунок Г.10 – Аналоги. Gamer Gains

Порівняльний аналіз аналогів

Критерії	game pressure	IGN	The gamer	Gamer Gains	Власний застосунок
Визначення поточного рівня вмінь	-	-	-	+	+
Різноманітний вибір дисциплін	+	+	-	+	+
Ігрові керівництва	+	+	+	-	+
Навчальні завдання	-	-	-	+	+
Наявність безкоштовного плану	+	-	+	+	+
Загальна оцінка	60%	40%	40%	80%	100%

Рисунок Г.11 – Порівняльний аналіз аналогів

Метод навчання в ігрових ситуаціях

Для розуміння методу навчання в ігрових дисциплінах, який було імплементовано у програмний-застосунок для тренування кіберспортсменів, було розроблено математичну модель методу, яка враховує 3 ключових аспекти:

- Рівень навичок користувача;
- вибір дисципліни та керівництв або завдань;
- відстеження прогресу користувача.

Рівень навичок (Skill Level, SL) визначається через початкове опитування і може змінюватись з часом залежно від кількості прочитаних керівництв та виконаних завдань. Позначимо рівень навичок як SLi, де i - це індекс користувача. Рівні навичок можуть бути від 1 до 3.

Кожна ігрова дисципліна має набір керівництв, які позначаються відповідно до рівня складності:

- Gd,k - набір керівництв для дисципліни d і рівня k;
- Td,k - набір завдань для дисципліни d і рівня k.

Користувач обирає дисципліну d і отримує відповідний набір керівництв та завдань для свого рівня SLi.

Далі позначається:

- Ci,d,k - кількість прочитаних керівництв користувем I в дисципліні d на рівні k;
- Ei,d,k - кількість виконаних завдань користувачем i в дисципліні d на рівні.

$$SL_i(t+1) = \begin{cases} SL_i(t)+1 & \text{якщо } P_{G,i,d,k}(t) \geq 80\% \text{ і } P_{T,i,d,k}(t) \geq 80\% \\ SL_i(t)-1 & \text{якщо } P_{G,i,d,k}(t) \leq 20\% \text{ і } P_{T,i,d,k}(t) \leq 20\% \\ SL_i(t) & \text{в іншому випадку} \end{cases}$$

Прогрес користувача може бути виражений, як відсоток виконання кожного типу активності:

- прогрес читання читання керівництв PG,i,d,k = (Ci,d,k / Gd,k)* 100%;
- прогрес виконання завдань PT,i,d,k = (Ei,d,k / Td,k)* 100%.

В прогресі користувача |Gd,k | та |Td,k | - це загальна кількість керівництв та завдань для дисципліни d і рівня k.

У наведеній системі рівняння користувача t - це поточний час, а t+1 - це наступний часовий інтервал.

Персональний кабінет користувача відображає його прогрес у вигляді списку прочитаних керівництв(Ci,d,k) і виконаних завдань(Ei,d,k).

Дана математична модель дозволяє структурувати навчальний прогрес користувачів та тренування кіберспортсменів, дозволяє оцінювати їхній прогрес, тобто є ефективною для імплементації у програмний застосунок.

Рисунок Г.12 – Метод навчання в ігрових ситуаціях

Діаграма діяльності застосунку

Для розуміння роботи програмних модулів застосунку для тренування кіберспортсменів було розроблено модель роботи у вигляді діаграми діяльності. Згідно наведеної діаграми користувачу спочатку необхідно обрати ігрову дисципліну в навігаційній панелі. Далі пропонується авторизуватись та здійснити проходження опитування для визначення поточного рівня знань та вмінь, якщо воно ще не було пройдене. Якщо користувач вирішує авторизуватись, але процес проходить не вдало, то система запропонує йому авторизуватися знову. Після авторизації пропонуються доступні керівництва та завдання, які користувач може переглядати та виконувати.

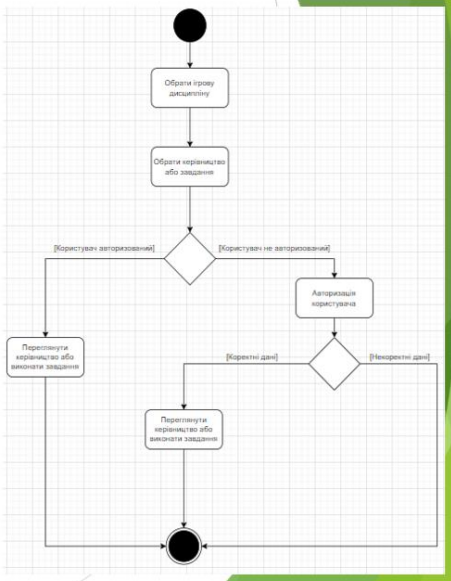


Рисунок Г.13 – Діаграма діяльності застосунку

Алгоритм роботи програми

Для розробки застосунку для тренування кіберспортсменів розроблено загальний алгоритм, згідно якого він буде працювати. Згідно зазначеного алгоритму, користувач спершу обирає ігрову дисципліну. Якщо користувач не авторизований, то йому буде запропоновано авторизуватися. Якщо користувач вирішить не авторизовуватись, то йому видасть список усіх наявних завдань та керівництв з коротким описом та рівнем складності. Якщо користувач авторизовується, та вперше обирає дану ігрову дисципліну, то йому пропонується пройти опитування для визначення рівня поточних вмінь та знань. Після проходження опитування користувач отримує список з керівництвами та завданнями, які відповідають його поточному рівню. Далі користувач може обирати, що саме він бажає переглянути або виконати. Після перегляду керівництва, або виконання завданя він може позначити їх, як прочитане або виконане відповідно.

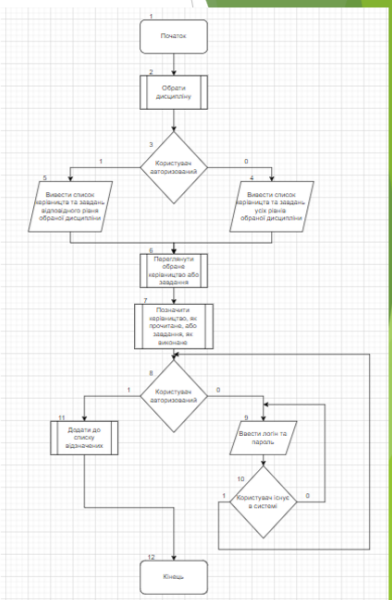


Рисунок Г.14 – Алгоритм роботи програми

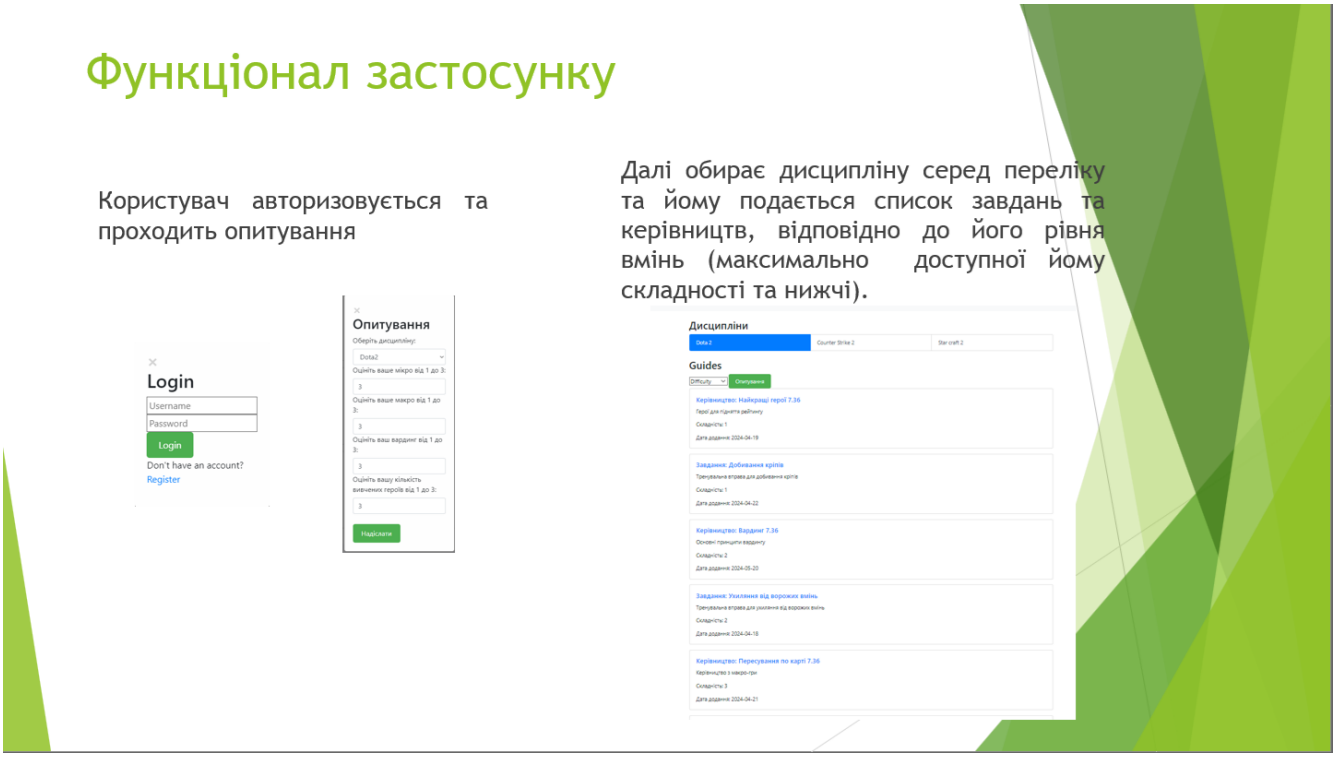


Рисунок Г.15 – Функціонал застосунку

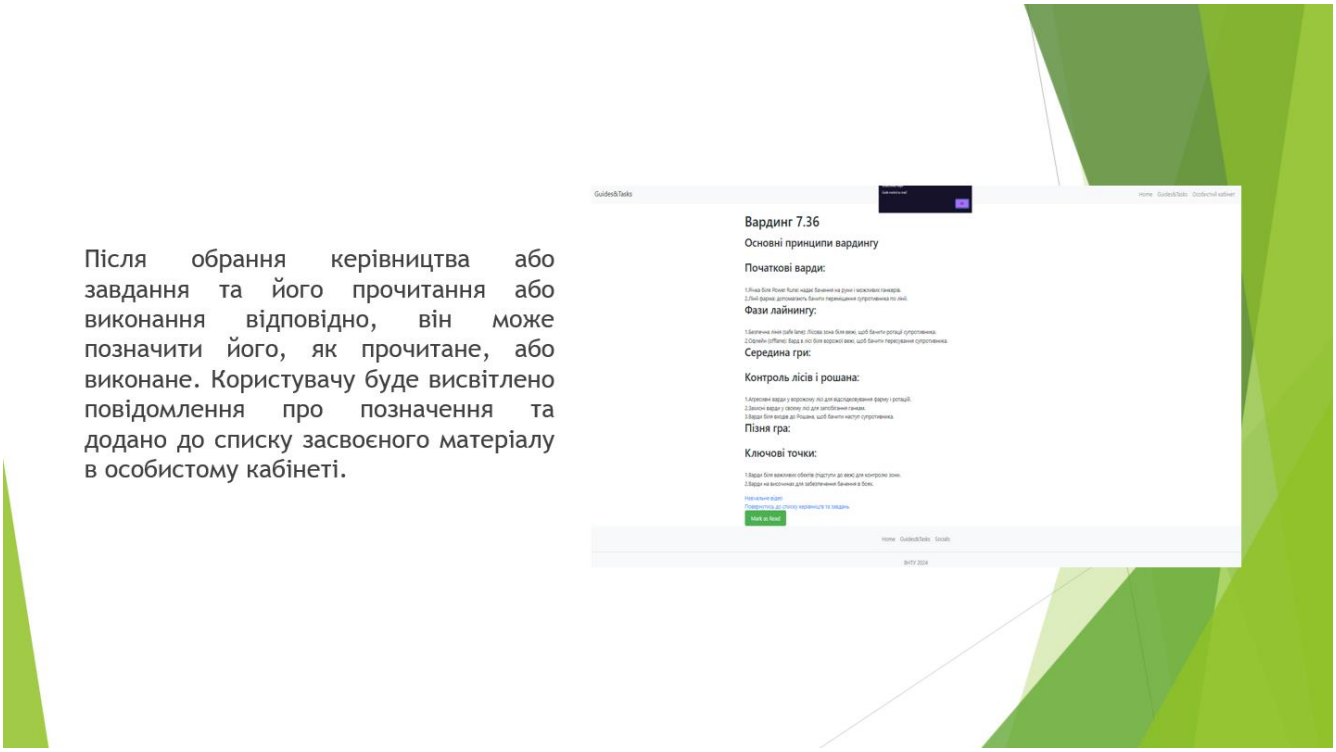
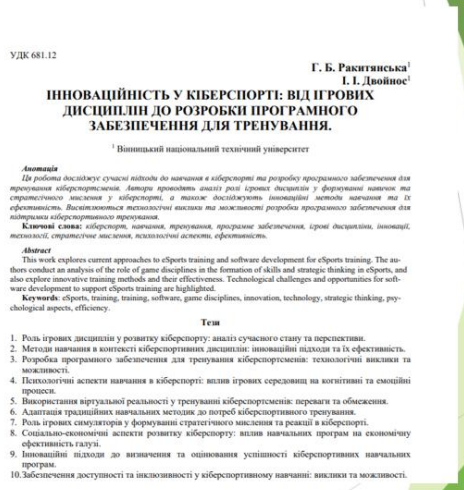


Рисунок Г.16 – Функціонал застосунку

Апробація та публікації

Результати роботи доповідались на конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2024)»



УДК 681.12

Г. Б. Ракитянська[1]
І. І. Двойнос[1]

ІННОВАЦІЙНІСТЬ У КІБЕРСПОРТІ: ВІД ІГРОВИХ ДИСЦИПЛІН ДО РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ТРЕНУВАННЯ.

[1] Вінницький національний технічний університет

Анотація

Ця робота досліджує сучасні підходи до навчання в кіберспорті та розробку програмного забезпечення для тренування кіберспортсменів. Автори проводять аналіз ролі ігрових дисциплін у формуванні навичок та стратегічного мислення у кіберспорті, а також досліджують інноваційні методи навчання та їх ефективність. Висвітлюються технологічні виклики та можливості розробки програмного забезпечення для підтримки кіберспортивного тренування.

Ключові слова: *кіберспорт, навчання, тренування, програмне забезпечення, ігрові дисципліни, інновації, технології, стратегічне мислення, психологічні аспекти, ефективність.*

Abstract

This work explores current approaches to eSports training and software development for eSports training. The authors conduct an analysis of the role of game disciplines in the formation of skills and strategic thinking in eSports, and also explore innovative training methods and their effectiveness. Technological challenges and opportunities for software development to support eSports training are highlighted.

Keywords: eSports, training, training, software, game disciplines, innovation, technology, strategic thinking, psychological aspects, efficiency.

Тези

1. Роль ігрових дисциплін у розвитку кіберспорту: аналіз сучасного стану та перспективи.
2. Методи навчання в контексті кіберспортивних дисциплін: інноваційні підходи та їх ефективність.
3. Розробка програмного забезпечення для тренування кіберспортсменів: технологічні виклики та можливості.
4. Психологічні аспекти навчання в кіберспорті: вплив ігрових середовищ на когнітивні та емоційні процеси.
5. Використання віртуальної реальності у тренуванні кіберспортсменів: переваги та обмеження.
6. Адаптація традиційних навчальних методик до потреб кіберспортивного тренування.
7. Роль ігрових симуляторів у формуванні стратегічного мислення та реакції в кіберспорті.
8. Соціально-економічні аспекти розвитку кіберспорту: вплив навчальних програм на економічну ефективність галузі.
9. Інноваційні підходи до визначення та оцінювання успішності кіберспортивних навчальних програм.
10. Забезпечення доступності та інклюзивності у кіберспортивному навчанні: виклики та можливості.

Рисунок Г.17 – Апробація та публікації

Висновки

У бакалаврській дипломній роботі було розроблено метод для навчання в ігрових дисциплінах та програмний застосунок для тренування кіберспортсменів.

Під час виконання бакалаврської дипломної роботи було проаналізовано стан проблеми на сьогоднішній день. Було розглянуто основні аналоги програмного продукту. Було визначено їхні недоліки та порівняно з власним програмним продуктом.

Під час аналізу технологій розробки було обґрунтовано вибір мов програмування HTML, CSS, JavaScript та PHP і бібліотек BootsTrap та jQuery.

Було розроблено метод навчання в ігрових дисциплінах та схему загального алгоритму роботи програмного застосунку для тренування кіберспортсменів. Також, було розроблено модель роботи застосунку у вигляді діаграми діяльності.

Тестування програми довело повну працездатність застосунку та відповідність поставленому технічному завданню. Також було розроблено інструкцію користувача.

Рисунок Г.18 – Висновки