

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

**РОЗРОБКА ІНТЕРАКТИВНОЇ НАВЧАЛЬНОЇ ВЕБПЛАТФОРМИ З
ВИКОРИСТАННЯМ ГЕЙМІФІКАЦІЇ**

**DEVELOPMENT OF AN INTERACTIVE EDUCATIONAL WEB
PLATFORM USING GAMIFICATION**

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-42
Примич Марія Олександрівна

Керівник:
Ліщина Наталія Миколаївна

Кваліфікаційну роботу
допущено до захисту
«10» 06 2025 р.

Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна



Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М.П.
«20» 12 2024р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Примич Марії Олександрівні

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: «Розробка інтерактивної навчальної вебплатформи з використанням гейміфікації».

Керівник роботи: к.т.н., доцент, завідувач кафедри Ліщина Н.М.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: Visual Studio Code, HTML, CSS, JavaScript, методичні вказівки до виконання кваліфікаційної роботи бакалавра.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити): проаналізувати та дослідити гейміфікацію в навчанні та дослідити вдалі застосунки з цим методом, спроектувати UpTree з UML-діаграмами, створити дизайн у Figma, підібрати технологій та виконання функціоналу, розробити вебзастосунок та працювати з базами даних і тестування застосунку.

5. Перелік графічного матеріалу: 20 рисунків, 1 додаток.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Ліщина Н. М.		
Специфікація вимог до розробленої системи	Ліщина Н. М.		
Розробка об'єкта проектування	Ліщина Н. М.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту	1,11 %		
Академічна доброчесність	Ліщина Н. М.		

7. Дата видачі завдання «20» грудня 2024 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	<i>вик.</i>
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	<i>вик.</i>
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	<i>вик.</i>
4	Розробка функціонально-структурної схеми роботи об'єкта проектування та проектування бази даних	до 29.03.2025 р.	<i>вик.</i>
5	Практична реалізація об'єкта проектування та розробка бази даних	до 26.04.2025 р.	<i>вик.</i>
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	<i>вик.</i>
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедрі	до 10.06.2025 р.	<i>вик.</i>

Здобувач вищої освіти



Марія ПРИМИЧ

Керівник кваліфікаційної роботи



Наталія ЛІЩИНА

АНОТАЦІЯ

Примич М. О. Розробка інтерактивної навчальної вебплатформи з використанням гейміфікації. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Кваліфікаційна робота бакалавра складається з вступу, трьох розділів, висновків, списку використання джерел та додатків.

Перший розділ зосереджений на комплексному вивченні проблем зосередженості під час навчання, а також пошуку нових підходів для їх вирішення через призму гейміфікації. Центальною темою є гейміфікація в освітньому процесі: аналізуються її основні принципи, ефективні механізми та практичні приклади успішного впровадження на таких популярних платформах.

Другий розділ розпочинається з використання UML-діаграм, що допомогло точно визначити функціональні та нефункціональні потреби системи. Розглядатимуться всі деталі дизайнерських рішень за допомогою Figma. Також у цій роботі виконано детальний розбір найважливіших веб-технологій. Окрім теоретичної бази, робота включає в себе вагому практичну складову.

Докладно буде розглянуто створення структури веб-сайту за допомогою React, а також реалізація всієї функціональності за допомогою JavaScript. Особлива увага приділятиметься роботі з базами даних, буде досліджено надійність бази даних MongoDB та механізми, які забезпечують захист даних користувачів. Завершенням роботи стане опис проведеного тестування розробленого застосунку.

Ключові слова: мотивація, гейміфікація, механіки, освітній процес, функціональні вимоги, нефункціональні вимоги, вебзастосунок, React.js, бази даних.

ABSTRACT

Prymych M. Development of an Interactive Educational Web Platform Using Gamification. Manuscript. Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2025.

The bachelor's qualification thesis consists of an introduction, three chapters, conclusions, a list of references, and appendices.

The first section focuses on a comprehensive study of the problems of concentration during learning, as well as the search for new approaches to solving them through the prism of gamification. The central topic is gamification in the educational process: its basic principles, effective mechanisms and practical examples of successful implementation on such popular platforms are analyzed.

The second section begins with the use of UML diagrams, which helped to accurately determine the functional and non-functional needs of the system. All the details of design solutions using Figma will be considered. This work also includes a detailed analysis of the most important web technologies. In addition to the theoretical basis, the work includes a significant practical component.

The creation of a website structure using React will be considered in detail, as well as the implementation of all functionality using JavaScript. Special attention will be paid to working with databases, the reliability of the MongoDB database and mechanisms that ensure the protection of user data will be examined. The work will be completed with a description of the testing of the developed application

Keywords: motivation, gamification, mechanics, educational process, functional requirements, non-functional requirements, web application, React.js, databases.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	15
Висновки до розділу 1	15
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ	17
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	17
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання	23
Висновки до розділу 2	33
РОЗДІЛ 3 РОЗРОБКА ІНТЕРАКТИВНОЇ НАВЧАЛЬНОЇ ВЕБПЛАТФОРМИ З ВИКОРИСТАННЯМ ГЕЙМІФІКАЦІЇ	34
3.1 Практична реалізація об'єкта проектування	34
3.2 Тестування та налагодження інформаційно-комп'ютерної системи	38
3.3 Розробка бази даних	42
3.4 Забезпечення інструктивної інформації користувача	47
3.5 Захист HTTPS з'єднання	48
Висновки до розділу 3	50
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТКИ	55

ВСТУП

Актуальність теми. У добу цифрової революції, коли інформація просто виливається з екранів, а час летить стрімко, складнощі з зосередженням та боротьба зі стресом стають справжньою епідемією. Освіта, відчуваючи цей тиск, повинна адаптуватися, саме тому гейміфікація, що впроваджує ігрові принципи у навчання, виходить на перший план, як один з ключових інструментів для підтримки внутрішньої мотивації та всебічного розвитку особистості.

Ця кваліфікаційна робота має на меті розробку інтуїтивно легкої, безпечної та функціональної вебплатформи з застосуванням методу гейміфікації для полегшення роботи з концентрацією учнів, студентів та просто користувачів данної платформи.

Об'єктом кваліфікаційної роботи є практичне втілення вебплатформи UpTree з елементами гейміфікації.

Предметом кваліфікаційної роботи є реалізація інтерактивної навчальної вебплатформи з використанням гейміфікації.

Для досягнення визначеної мети були визначені такі завдання:

- проаналізувати концепції гейміфікації та дослідити популярні навчальні вебплатформи та їхні гейміфіковані елементи;
- сформулювати функціональні та нефункціональні вимоги до вебзастосунку UpTree;
- обрати оптимальний технологічний стек для розробки;
- створити інтуїтивно зрозумілий та естетично привабливий користувацький інтерфейс;
- провести тестування ключових компонентів фронтенду та бекенду для перевірки коректності їх функціонування;
- спроектувати та реалізувати оптимальну архітектуру бази даних, що ефективно зберігатиме та управлятиме всією інформацією про користувачів;
- розробити та інтегрувати вичерпний розділ «Довідка», який забезпечить швидку адаптацію нових користувачів;

– впровадити протокол безпечного з'єднання HTTPS для шифрування трафіку та забезпечення конфіденційності та цілісності даних.

В якості апробації матеріалів роботи написано тезу, яка була опублікована в збірнику X Міжнародної науково-практичної конференції з проблем вищої освіти і науки «Інформаційні технології в освіті, науці і виробництві (ІТОНВ-2025)» м. Луцьк, 23-24 травня 2025 р. (додаток А).

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

Освітній процес є важливим і невід’ємною частиною для забезпечення знань молоді. Так як знання прямо зараз є великим ядром для відкриття можливостей у реалізації будь яких теорій та ідей. Різноманітність та важкість інформації може дещо лякати і навіть таким чином відштовхувати від себе. Їх не організованість, по причинах: повітряні тривоги, брак відповідного облаштування в укриттях та інші організаційні негаразди, що виникли через воєнний стан, ускладнюють процес здобуття знань.

Також через нелегкі часи в країні, кожен житель України відчуває величезний стрес. Це суттєво погіршує когнітивну здатність учнів, зокрема, їхню концентрацію та пам’ять, які вкрай необхідні для результативного навчання. Безперервне відчуття тривоги, страх і можливі травматичні події емоційно виснажують дітей, провокуючи апатію та зниження інтересу до навчання. Фізичні прояви стресу, на кшталт втоми та проблем зі сном, додатково ускладнюють освітній процес. В поведінці може виникати роздратування або ухилення від навчання, що ще більше зменшує академічну успішність.

Крім того через перенавантаженість навчальних програм, мотивація опускається до рівня нуля. І студенти просто перестають працювати на заняттях, лишаючись без підтримки. У наш час є достатньо нелегкою задачею для здобувачів знань самотійно впоратись з організацією власного часу та виставлення пріоритетності. Надмірно велика кількість інформації, поганий тайм-менеджмент, пасивні та застарілі методи навчання без усякого заохочення можуть призводити до вигорання.

Для того аби вирішити вище зазначені складнощі існують різноманітні методи аби виправити уже втрачений інтерес.

У сучасному світі активно розвивається адаптація навчальних програм, що робить навчання більш цікавим та актуальним, пов'язуючи шкільні знання з реальним життям і роблячи їх менш обтяжливими. Створення позитивної атмосфери на заняттях нагадує формування дружньої спільноти, де панує взаємоповага та підтримка, а досягнення кожного цінуються. Коли навчання стає особисто значущим та відбувається у сприятливому оточенні, зацікавленість учнів природно збільшується. Підвищення мотивації та самостійності передбачає надання учням можливості вибору в навчанні, що пробуджує їхній інтерес та відповідальність за особистий прогрес.

Один з найефективніших методів являється застосування інтерактивних та ігрових методів у навчальному процесі. Цей метод є важливим педагогічним підходом, спрямованим на активізацію навчального процесу та збільшення зацікавленості учнів. В основі цього лежить діяльнісне навчання, де знання здобуваються через пряму взаємодію з матеріалом та співпрацю. Серед цих методів особливе місце посідає гейміфікація, або використання ігрових механік у навчанні.

Гейміфікація перетворює навчання на захопливий процес, використовуючи елементи, знайомі з ігор. Одним з ключових елементів є система балів та нагород, де учні отримують віртуальні або реальні відзнаки за свої досягнення, активність та виконання завдань. Це стимулює їх старанність та бажання прогресувати.

Іншим важливим аспектом є система рівнів та відображення прогресу, яка візуально показує учням їхній шлях у навчанні, мотивуючи до підкорення нових «вершин» знань. Елемент змагання через лідерські таблиці додає азарту та спонукає учнів докладати більше зусиль, щоб досягти кращих результатів порівняно з однокласниками. Занурення навчального матеріалу в цікавий сюжет або наратив робить навчання емоційно привабливим та допомагає краще засвоювати інформацію через залучення уяви. Постановка викликів та надання можливості відчувати радість від досягнення успіху також є потужним мотиватором. Нарешті, важливу роль відіграє швидкий зворотний зв'язок, який

допомагає учням розуміти свої помилки та вчасно їх виправляти, сприяючи постійному вдосконаленню.

Щодо представників то великих досягнень у напрямку гейміфікації досягли такі платформи як Duolingo, Habitica а також Kahoot!. Ці додатки є достатньо популярними або ж мають свою віддану кількість прихильників, у сферах навчання та розвитку чи створені звичок. Для наглядності, графік на рисунку 1.1 зображено те скільки користувачів завантажили додаток за рік.

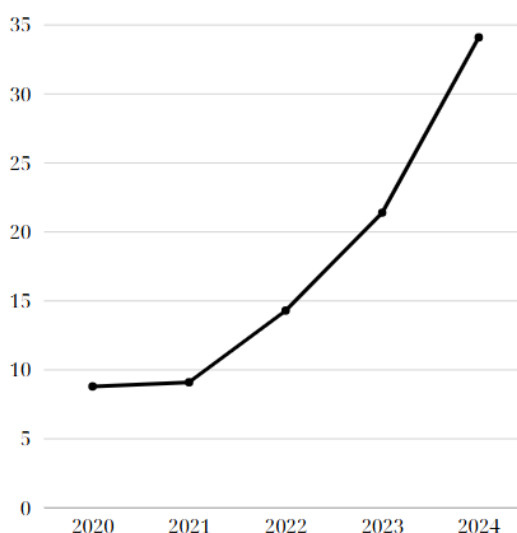


Рисунок 1.1 – Діаграма щоденної активності користувачів Duolingo [1]

На зображенні ми можемо спостерігати, що упродовж років, кількість користувачів, які вираховуються у мільйонах, помітно зростає. У 2024 році кількість щоденних активних користувачів збільшилась на 59 % у порівнянні з попереднім роком. Усе завдяки кросплатформеності, саме тому було проаналізовано ці платформи.

Першим є Duolingo, застосунок має велику популярність завдяки великому вибору у вивченні різних мов. Цей додаток можна впізнати за виглядом сови, який зображений на аватарці та є талісманом застосунку, що зображено нижче на рисунку 1.2.



Рисунок 1.2 – Офіційний талісман Duolingo, сова з іменем Duo [2]

Важливо зазначити що цей додаток чудово використовує ігрові механіки. Отож, варто проаналізувати які ж вони.

Найпоширенішою механікою у методіці гейміфікації – система балів. Накопичення певних значків, коїнів, очків що демонструє користувачеві його ефективність, результат праці достатньо сильно мотивує та заохочує продовжувати збільшувати свої досягнення. Крім того у застосунку присутня стратегія конкуренції, тобто змагання між користувачами за певний рейтинг також є мотивуючим поштовхом для роботи над собою. Ліги, таблиці лідерів, а також сезонні таблиці з реальними людьми, дають змогу користувачеві відчувати себе не самотнім і причетним до спільного вивчення. Серії днів, або так звані «Streaks» [2]. Візуальне відзначення власного прогресу кожного дня, заохочує користувача продовжувати робити це щодня. І чим більше днів підряд ви заходите і виконуєте завдання тим сильніше ваше бажання буде зайти завтра. Для кращого розуміння усі вище зазначені пункти зображенні нижче на рисунку 1.3.

Статистика

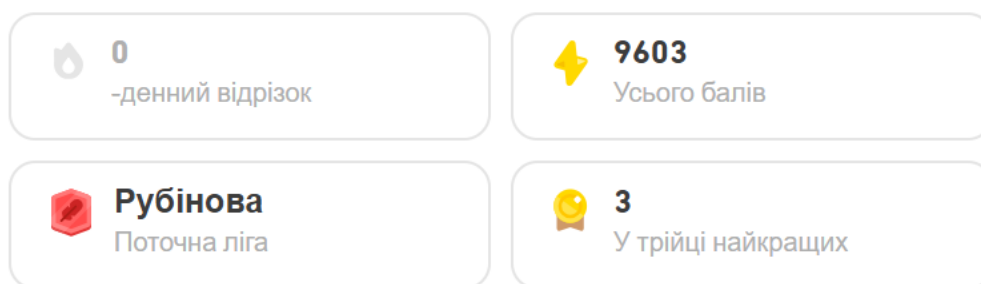


Рисунок 1.3 – Механіки які використовуються у застосунку Duolingo [2]

Отож, заглиблюючись у застосунок я дізналась що у розробці такого популярного кросплатформера було використано декілька мов програмування. Ці знання в майбутньому допоможуть мені у виборі технологій для власного проекту. Для розробки вебверсії, були використані HTML, CSS а також JavaScript зі своїми фреймворками або з відомими бібліотеками такими як React, Angular або Vue.js.

Другий додаток це Habitica, аватарка зображена на рисунку 1.4 аватарку додатку, яка має власне коло прихильників.



Рисунок 1.4 – Аватарка додатку Habitica [3]

Особливість цієї платформи полягає у тому що ви створюєте себе як персонажа і за рахунок щоденного виконання завдань які є вашою рутиною, хоббі або певною тимчасовою справою. Усе це дає вашому персонажі досвід і може впливати на здоров'я вашого персонажа, як продемонстровано на рисунку 1.5. Ця механіка створює відчуття зв'язку з персонажем, тобто створюється відчуття зв'язку і користувач робить це з думкою «Я = персонаж».

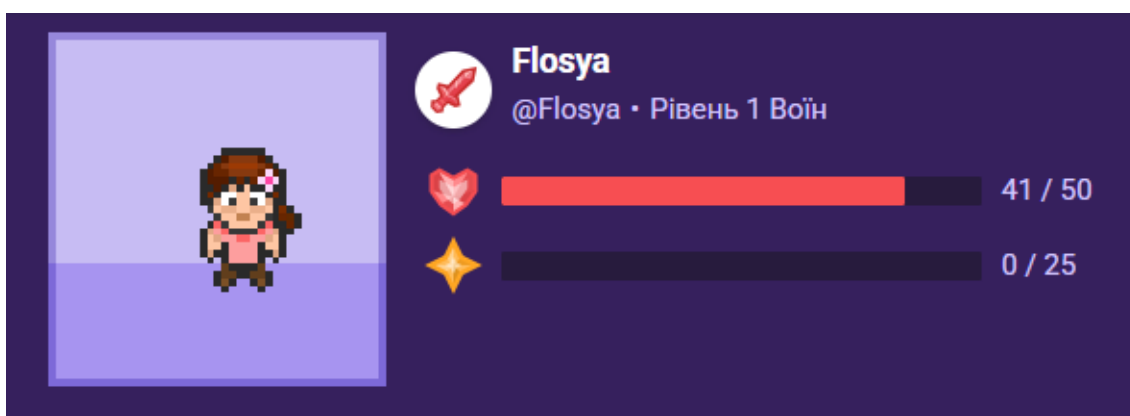


Рисунок 1.5 – Частина інтерфейсу вебзастосунку Habitica [3]

Крім того, за виконання важких або періодичних завдань ви можете отримати винагороду у вигляді золота або предмет, який можна купити за ігрову валюту(золото), що є мотивацією від винагородження та заохоченням продовжувати працювати над собою та своїм персонажем. Також у цьому застосунку існує можливість працювати в компанії, ви можете створити власну або ж доєднатись до когось, та знайти собі друзів. Завдяки цьому можна побороти більш складних монстрів. Та отримати особливі предмети для вашого персонажа [3]. Завдяки таким особливим механікам, абсолютно увесь вебзастосунок мотивує користувача усіма своїм інтерактивами працювати над собою. А щодо технологій які використовувались при створенні та візуалізації. Було використано HTML, SCSS, JavaScript з використанням фреймворку Vue.js.

І третім у списку популярних застосунків що використовуються гейміфікацію є «Kahoot!». Це веб-сервіс та мобільна програма для створення та проведення ігрових вікторин, опитувань і дискусій в реальному часі.

Основна концепція Kahoot! полягає у перетворенні навчання на захоплюючу гру, яка спонукає до змагання та динаміки. Вчитель або модератор розробляє серію питань з множинним вибором, а учні або учасники підключаються до гри через свої пристрої, використовуючи унікальний ігровий код [4]. Головними механіками цього застосунку є нарахування балів за правильні й швидкі відповіді адже усі ці ігри проводяться у реальному часі, засвоєні знання та швидке мислення винагороджується. Крім того завдяки різним створеним компаніям існують рейтинги та таблиці лідерів, які стимулюють користувача до відчуття змагань. І також механіка «streak», тобто серії ігор, які мотивують до регулярної участі. Kahoot! здобув надзвичайну популярність як серед навчальних закладів, так і серед репетиторів завдяки своєму ігровому формату, який перетворює навчання на захоплюючий та інтерактивний процес.

Для створення використовувались HTML, CSS, JavaScript та його можливі бібліотеки або фреймворт, точно які саме не відомо.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

На дану кваліфікаційну я виділила певну кількість завдань що допоможуть мені реалізувати мій проєкт. Вони виглядають таким чином:

- проаналізувати концепції гейміфікації та дослідити популярні навчальні вебплатформи та їхні гейміфіковані елементи;
- сформулювати функціональні та нефункціональні вимоги до вебзастосунку UpTree;
- обрати оптимальний технологічний стек для розробки;
- створити інтуїтивно зрозумілий та естетично привабливий користувацький інтерфейс;
- провести тестування ключових компонентів фронтенду та бекенду для перевірки коректності їх функціонування;
- спроектувати та реалізувати оптимальну архітектуру бази даних, що ефективно зберігатиме та управлятиме всією інформацією про користувачів;
- розробити та інтегрувати вичерпний розділ «Довідка», який забезпечить швидку адаптацію нових користувачів;
- впровадити протокол безпечного з'єднання HTTPS для шифрування трафіку та забезпечення конфіденційності та цілісності даних.

Висновки до розділу 1

У першому розділі я зосередилася на дослідженні складнощів, що супроводжують зосередженість та емоційне напруження, а також ознайомилися з рядом актуальних способів їх подолання. Основний задум цього розділу полягав у дослідженні застосування гейміфікації в навчальному процесі. Було пояснено визначили поняття «гейміфікація» та розглянуто її ключові складові. Наведено приклади успішних програм, де застосовується гейміфікація, наприклад, Duolingo, Habitica та Kahoot!. У цьому розділі також було детально проаналізовано, як саме механізми гейміфікації використовуються для

стимулювання користувачів до самовдосконалення та сприяння регулярному навчанню. Окрім цього, я дослідила технології та інструменти, що використовувалися для розробки цих додатків.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Для того аби розпочати розробку, я опиралась на деякі уже відомі методи які показали свою ефективність на прикладі інших успішних аналогів. У процесі моделювання майбутнього проекту, я використовувала різноманітні UML-діаграми. Так як це надзвичайно корисний інструмент при моделюванні проекту. Вони дозволяють глибше зрозуміти систему, спроектувати її більш ефективно, полегшують розробку та сприяють якісній документації, що в підсумку позитивно впливає на кінцевий продукт.

Головною задачею було виділити позитивний вплив на користувачів за допомогою гейміфікації виконання завдань. Щодо хороших наслідків, то варто зазначити, що виділення пріоритетності завдань деколи дуже полегшує організацію власних справ та швидке підлаштування щодо потрібних до вказаних дедлайнів. Проблеми з організацією власного часу та розподіленням пріоритетності обумовлено тим, що тривале перебування у стресі має негативний вплив на когнітивні здібності. Воно погіршує здатність до зосередження, знижує ефективність пам'яті та ускладнює інтелектуальну діяльність загалом. На фоні війни останніх років, велика кількість українців стикнулись з проблемою стресу та тривожності, що вплинуло на їх концентрацію. Як показує статистика 31 % українців зазначили про труднощі зі зосередженням уваги в умовах війни [5].

Крім того частина населення страждає синдромом дефіциту уваги та гіперактивності, що виявляється через низку симптомів, серед яких складнощі з концентрацією, надмірна активність та проблеми з контролем імпульсів. 3-4 % дорослих українців відчувають симптоми СДУГ, що не зникають, при цьому у 60-80 % випадків вони тривають з дитинства, хоча з віком можуть слабшати таку статистику наведено у роботі [6].

Саме тому для цих двох типів людей гейміфікація є чудовим методом полегшити або покращити свою продуктивність та прокачати свою концентрацію адже гейміфіковані застосунки здатні посприяти людям у контролюванні тривоги та великої емоційної напруги. Позитивний вплив гейміфікації можна помітити за статистикою на рисунку 2.1.



Рисунок 2.1 – Діаграма що демонструє вплив гейміфікації на працівників за статистикою вказаною у роботі [8]

Це досягається через інтерактивні завдання, що заохочують взаємодію, віртуальні простори для роботи, а також моніторинг прогресу у подоланні дратівливих факторів. А також миттєвий позитивний фідбек, як-от нараховані бали, віртуальні трофеї чи доступ до нових етапів гри, здатен активізувати нейронні ланцюги, відповідальні за відчуття винагороди, безпосередньо у мозку. Цей аспект є надзвичайно важливим, особливо для тих, хто живе з СДУГ, адже їм зазвичай необхідні частіші та більш помітні підтвердження. Це все можна пов'язати через створення відчуття результату та руху вперед.

Тому першою кроком була діаграма класів, аби візуалізувати структури, це дозволяє наочно представити усі аспекти системи, взаємодії користувачів,

обробку даних та послідовність операцій. Вигляд моєї діаграми класів знаходиться на рисунку 2.2.

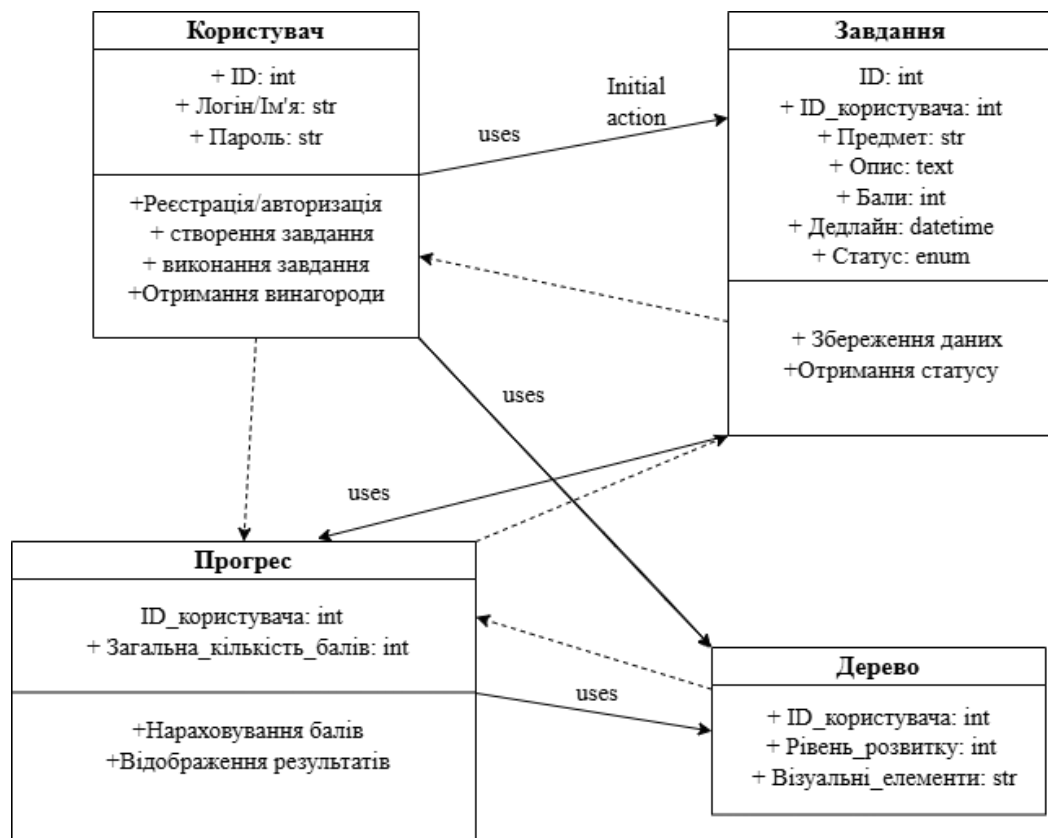


Рисунок 2.2 – Діаграма класів проекту «UpTree»

Як вказано на рисунку, головним у цій діаграмі є користувач, адже система розпочинає діяти як тільки користувач має змогу взаємодіяти із завданням. Тобто можливість створити, виконати та отримати статус. Це призводить до того, що користувач отримує прогрес і є залежним від нього. Адже прогрес це є винагородою. Відображення результатів, рівень якого досягнув користувач, кількість не виконаних завдань, а також кількість очків які уже були здобуті, і відображення нараховування балів це є частиною класу «Прогрес». «Дерево» це клас, що відповідає за відображення візуального інтерактиву є залежним від прогресу напряму.

Наступною була діаграма діяльності для того аби зрозуміти як краще реалізувати роботу веб-застосунку в цілому вигляді. Вигляд діаграми зображено на рисунку 2.3.

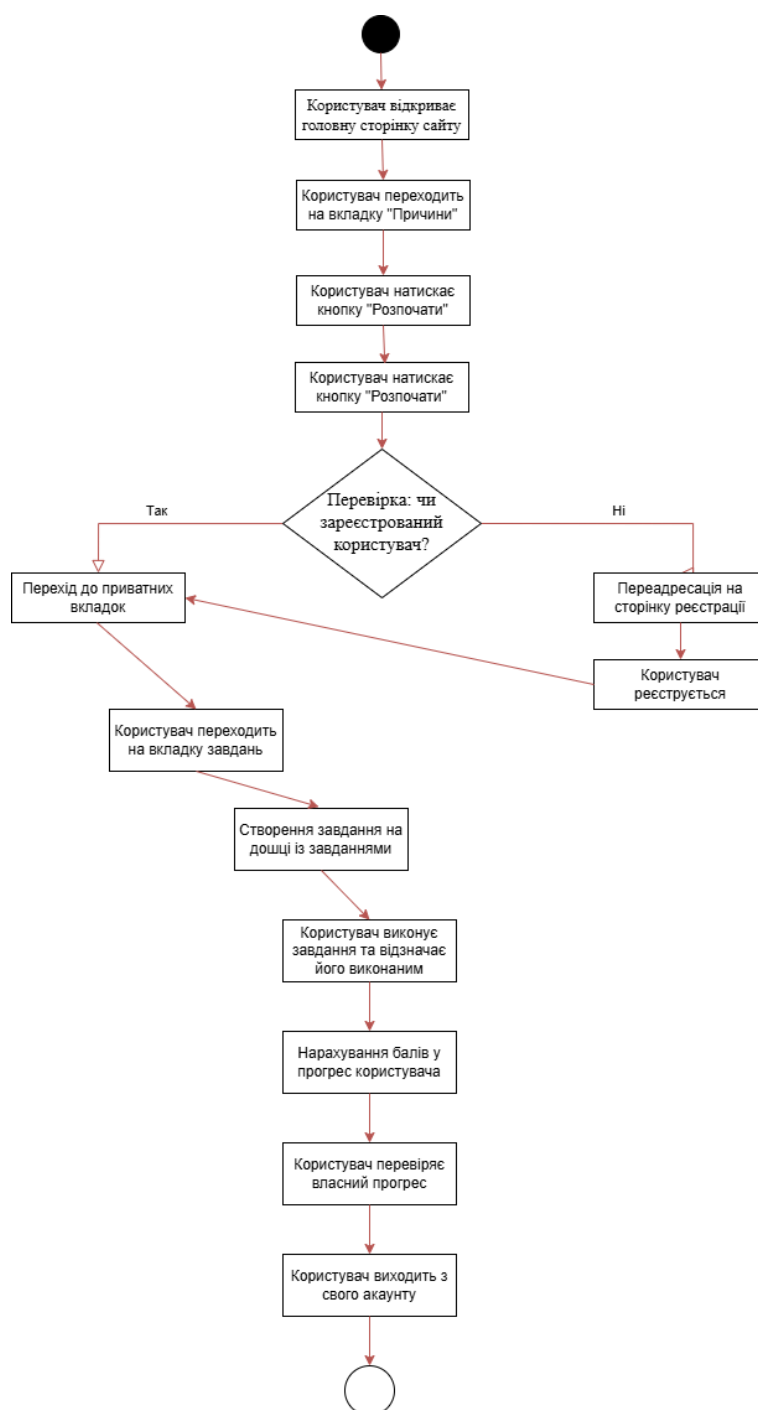


Рисунок 2.3 –Діаграма діяльності вебзастосунку «UrTree»

Діаграма діяльності яку я зобразила сконцентрована на користувацькому досвіді та на взаємодії між користувачем (або гостем) і моїм вебзастосунком UrTree. Сценарій який продемонстровано в діаграмі показує ключову «стежину» користувача в системі. А тобто який порядок дій виконуватиме користувач завітавши на сторінку «UrTree». Ідея вебзастосунку полягала у тому аби

використування було прямим і максимально зрозумілим. Саме тому усеє проце виглядає таким послідовним.

Отож після того як головні задуми були реалізовані у діаграмах, мені потрібно з'ясувати вимоги до мого проєкту. І під час аналізу можливих сценаріїв, я прийшла до висновку що користувач насамперед стикатиметься з головною сторінкою, яка має містити привабливий контент для новачків. Це спонукатиме гостей залишитись, а постійних відвідувачів нагадуватиме про їхню мету. Крім того, проєкт має передбачати перехід на сторінку з описом переваг використання застосунку, де повинна бути кнопка «Розпочати», що перенаправляє на реєстрацію/авторизацію.

Отже, гість мусить мати змогу зареєструватися на сайті, вказавши необхідні дані, а саме ім'я користувача та пароль, інакше доступ до всього інтерактиву сайту буде обмежений. Натомість, користувач, який вже створив обліковий запис, повинен мати змогу авторизуватися на сайті, використовуючи свій логін та пароль. Система повинна забезпечувати надійне зберігання паролів користувачів та надавати можливість відновлення паролю за рахунок збереженої інформації в базі даних.

У випадку спроби переходу на вкладки «Прогрес» або «Завдання» неавторизований користувач має бути перенаправлений на сторінку реєстрації/авторизації. Зареєстрований користувач має змогу переглядати свій поточний прогрес у вигляді накопичених балів та візуального зображення свого «дерева», що ілюструє його прогрес. Зареєстрований користувач повинен мати змогу переглядати дошку зі своїми створеними завданнями та створити нове завдання, натиснувши на кнопку «Створити». Система має показувати модальне вікно, де відбуватиметься створення самого завдання. Користувач повинен ввести назву предмету чи завдання, мати можливість описати суть завдання, визначити та вказати кількість балів за виконання, а також встановити дати дедлайну.

Після заповнення всіх необхідних полів користувач повинен мати змогу завершити створення завдання, яке має бути відображене на дошці завдань

користувача. Система має сортувати завдання, починаючи з найбільш термінових і до найменш термінових.

Після виконання завдання користувач має змогу відмітити створене ним завдання як виконане, після чого система автоматично нарахує користувачу відповідну кількість балів, які повинні відображатися на вкладці «Прогрес». Саме такими я бачу функціональні вимоги свого застосунку.

А щодо нефункціональних вимог то було встановлено такі очікування до проекту. Першим у списку є розділ ефективності та швидкодії. Саме тому першим очікуванням є те що, веб-додаток має вражати швидкістю тобто завантаження сторінки та ключові операції таких як-от створення чи позначення завдань. Ресурси сервера потрібно оптимізувати, аби він стабільно функціонував навіть під навантаженням великої кількості користувачів.

Другим і не менш важливим є вимоги щодо комфорту користувачів. Тому веб-додаток має бути інтуїтивно зрозумілим та викликати приємні відчуття, бути дружнім до користувачів з різним рівнем обізнаності в техніці, забезпечуючи простоту опанування та результативність виконання задач. Навігація повинна бути логічною, елементи інтерфейсу – зрозумілими, а взаємодія – безпроблемною.

І так як кожен користувач матиме власну створену інформацію, третьою важливою вимогою до свого проекту було поставлено безпека та цілісність даних користувачів. Отже безпека інформації тих, хто користується сервісом - понад усе. Приватна інформація має бути добре захищена від будь-якого нелегального проникнення. Цілісність та конфіденційність даних тих, хто користується сервісом, мусять бути забезпечені на найвищому рівні. Система зобов'язана гарантувати надійний захист від зовнішніх атак та випадкового зникнення інформації.

Зокрема варто думати про майбутнє проекту. Тому четвертою вимогою є акцент на підтримці та розвитку. Код вебзастосунку мусить бути організованим та добре задокументованим, що спростить його подальшу підтримку, поновлення та розширення новими функціями. Система має бути розроблена

таким чином, щоб її було легко підтримувати та розвивати в майбутньому, мінімізуючи витрати часу та ресурсів на впровадження змін.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Головною метою при створенні дизайну мого проєкту є візуальна ідентичність, який миттєво пояснює мету вебзастосунку. Тому для розробки дизайну, я обрала інструмент для дизайну інтерфейсів та взаємодії з користувачами, що повністю працює у веб-браузері, а саме Figma. Ця платформа була випущена у загальний доступ ще у 2016 році, але останні 3 роки набула великої популярності та сильно закріпилась у світі UI та UX дизайну [8]. Оскільки Figma працює в браузері, немає потреби у встановленні додаткового програмного забезпечення, а доступ до проєктів та їхній спільний перегляд та редагування стають надзвичайно зручними з будь-якого пристрою, що має підключення до мережі Інтернет. Головною фішкою є функція спільної праці у реальному часі, що робить його чудовим варіантом для команд, які займаються розробкою веб-сайтів, мобільних додатків та інших цифрових продуктів.

Отож під час роздумів створення проєкту, я задумалась щодо кольорової палітри. Впродовж деяких досліджень було доведено що сині та зелені кольори мають позитивний вплив на наше здоров'я та емоційний стан. До прикладу при тривалій взаємодії з синім кольором у людини фіксується зменшення тиску крові, емоційний стан стабілізується. Використання синього кольору може допомогти позбутися стресу, сприяти поліпшенню когнітивних функцій мозку та покращити пам'ять така інформація вказується у роботі [9].

У своєму проєкті я використала асоціацію знань користувача з деревом, головним кольором сайту я обрала колір що більше асоціюється з природньою тематикою. Тому під час пошуку я отримала чимало статистичної інформації щодо позитивного впливу зеленого кольору. Я опиралась на інформацію, що коли очі дивляться на зелений колір знижується тиск в судинах очей і

відновлюється зір. А також багато студентів та працівників зазначали що у 50-70 % випадків зелений колір у їх просторі знижує рівень стресу, а також створює відчуття безпеки та концентрації. Опираючись на цю статистику я вирішила, що дизайн моєї кваліфікаційної роботи складатиметься з різних відтінків зеленого. А застосування стилізованого зображення дерева не тільки приємне для ока, а й містить глибоку символіку росту, розвитку та досягнення поставлених цілей. У контексті концепції перетворення буденних справ на «епічні квести», образ дерева метафорично зображає поступове зростання та здобуття нових «вершин» через виконання завдань. Цей центральний візуальний елемент запам'ятовується та формує унікальний характер сайту. Інтуїтивно зрозуміла навігація – ще одна важлива перевага дизайну. Розміщення головного меню у верхній частині екрана відповідає усталеним веб-стандартам, що робить навігацію сайтом легкою та звичною для користувачів. Крім того меню завжди закріплене. Чіткі та лаконічні назви розділів допомагають швидко орієнтуватися та знаходити необхідну інформацію. Вигляд цього меню можна побачити на рисунку 2.4.

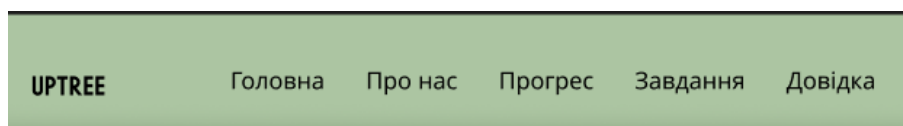


Рисунок 2.4 – Вигляд меню веб застосунку UpTree

Головна сторінка ефективно використовує привабливий заклик до дії. Фраза «Перетворіть свої рутинні справи на епічні квести!» є яскравою, оригінальною та викликає зацікавленість. Додатковий, більш детальний текст, розміщений поруч, розкриває цю ідею, пояснює концепцію перетворення повсякденності на захопливі пригоди та заохочує користувачів почати свою «епічну подорож». Таке поєднання ефектного заголовка та інформативного підзаголовка є потужним інструментом для залучення нових користувачів. Приклад дизайну зображеного на рисунку 2.5.



Рисунок 2.5 – Дизайн головної сторінки UpTree

Другий екран демонструє чітко структуровані блоки, що одразу сповіщає про переваги платформи. Використання мінімалістичного дизайну в цьому розділі допомагає користувачам швидко ознайомитися з основною інформацією, не відволікаючись на зайві елементи. Виразна кнопка «Розпочати» під цими блоками, є прямим закликком до подальшої взаємодії та логічно завершує цей інформаційний блок, роблячи сайт інтуїтивно зрозумілим.

Наступні дві вкладки є приватними і доступними лише після реєстрації або авторизації користувача. Тому що вони представляють собою екран персонального профілю та інтерактивну дошку з завданнями. Отож вкладка «Статистика» демонструє увагу до персоналізації та відстеження прогресу користувача. Наявність імені користувача, його «рангу досягнень» та візуалізації прогресу у вигляді шкали створює відчуття залученості та мотивації. Інформація про день народження та кількість виконаних «квестів» додає персонального контексту та сприяє формуванню зв'язку користувача з платформою. Такий підхід до відображення особистих досягнень є важливим елементом для

підтримки активності користувачів та їх залученості до власного прогресу. Приклад такої персоналізації надано на рисунку 2.6.

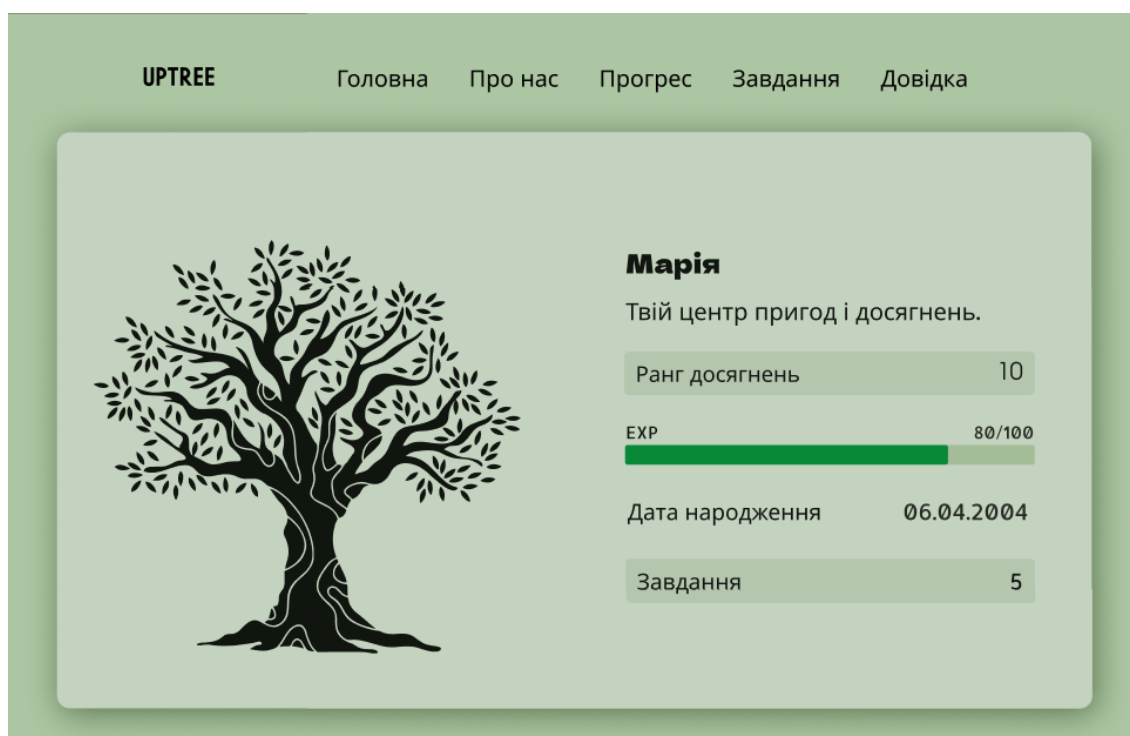


Рисунок 2.6 – Вигляд приватної вкладки «Прогрес»

Організація навчального контенту на четвертому екрані також заслуговує на увагу. Чітко розмежовані блоки з назвою завдання, ключовою інформацією та датою дедлайну створюють зручну структуру для перегляду доступних завдань, а також відстеження особистого прогресу. Візуальне позначення досвіду який користувач отримує за виконання власного чітко поставленого завдання забезпечує миттєвий зворотний зв'язок про завершення певного етапу. На цьому ж екрані розміщені зручні та зрозумілі кнопки для навігації та виконання дій. Їх розташування та чіткі назви спрямовують користувача до перегляду всього доступного списку завдань або створення нового, що покращує загальну зручність використання сайту. Вигляд четвертої вкладки вебзастосунку UpTree зображено на рисунку 2.7.

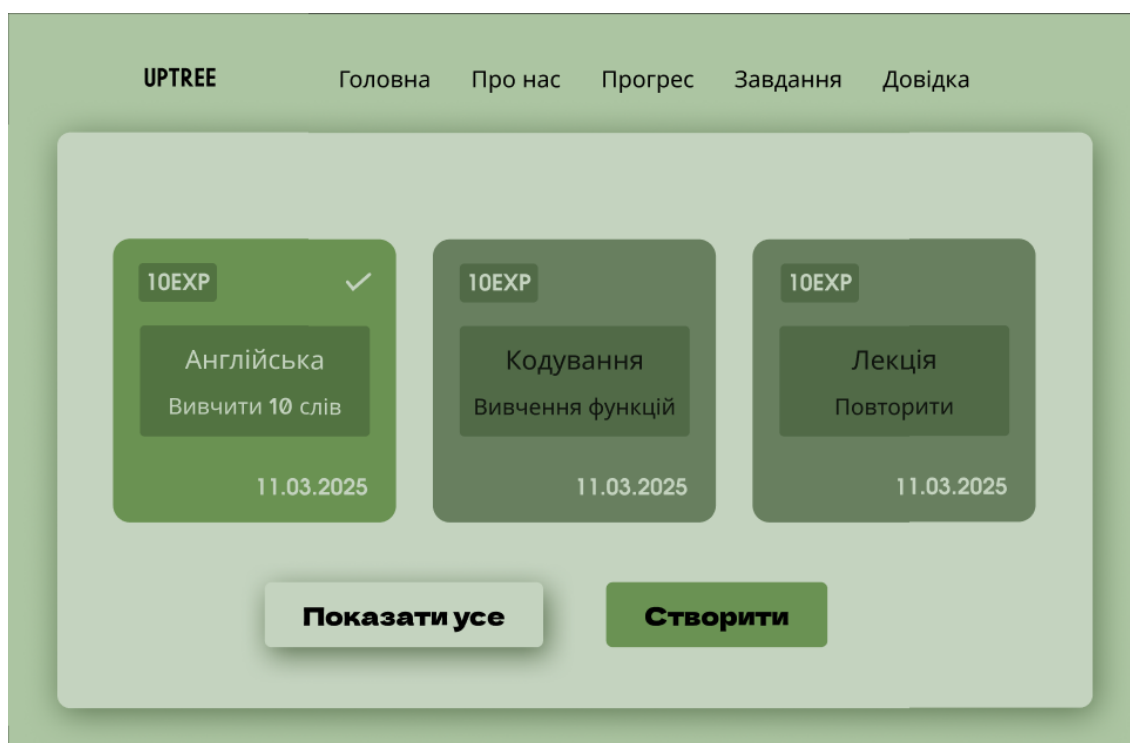


Рисунок 2.7 – Вигляд вкладки «Завдання»

Наскрізною перевагою всього дизайну є єдина та гармонійна стилістика, якої я старалась притримуватись протягом усієї роботи. Після того як дизайн був готовим, я розпочала пошук та підбір технологій та інструментів що допоможуть мені під час розробки мого проєкту. Я розгляда середовище для розробки, серед цікавих мені варіантів були Visual Studio Code(VS Code), IntelliJ IDEA і PyCharm. Адже усі ці середовища чудово підходять для розробки мого проєкту. Кожен з них підтримує нижче зазначені інструменти та має свої переваги:

– IntelliJ IDEA, це відмінне інтегроване середовище розробки, що славиться підтримкою мов Java та Kotlin, таке визначення було зазначено у роботі [10]. А також дозволяє розробляти і на інших мовах таких як Python, PHP, JavaScript та інші. Також IntelliJ IDEA підтримує розробку застосунків для мобільної платформи Android. Щодо переваг цього середовища, всередині проєкту існує величезна підтримка backend-розробки та frontend-технологій. Що дає змогу використовувати усі можливі знання для комфортного створення власного продукту. Крім того у середовищі є інтеграція з базами даних та

схожими інструментами що полегшує та збільшує різноманіття роботи з технологіями. А також під час написання вашого коду, в IntelliJ IDEA існує вбудований редактор, який буде виправляти ваші можливі помилки, доповнювати вас, оптимізуючи вашу продуктивність у роботі. Щодо недомог, варто зазначити що IntelliJ IDEA може бути ресурсомістким у порівнянні з іншими середовищами для розробки, а також має обмеження для безкоштовної версії IntelliJ IDEA Community Edition і для повного розкриття цього потенціалу потрібно купити ліцензійовану версію IntelliJ IDEA Ultimate Edition;

– наступним варіантом є WebStorm це одне з інтегрованих середовищ для розробки яке спеціалізується на розробці вебзастосунків. І це уже велика перевага у тому що цей проєкт забезпечує величезну підтримку HTML, CSS, JavaScript, TypeScript а також основні frontend-фреймворки такі як React, Angular, Vue.js а також backend-платформи Node.js. Адже абсолютно усе середовище зосереджене на комфортному створенні вебзастосунків. Всередині вбудовані термінал та Git-інтеграція. А також підключені інструменти веб-розробки і бази даних. А щодо недомог, то головним недомогом є те що це середовище є платним повністю. А тобто для його використання потрібно бути впевненим у тому що саме це середовище яке потрібно для використання. А також є проблема величини ресурсів, тобто споживання великої кількості системних ресурсів;

– останнім середовищем є Visual Studio Code, це один з найпопулярніших кросплатформних редакторів коду. Цей редактор має велику кількість прихильників. Який славетний завдяки своїй гнучкості, легкості, зрозумілості, великій кількості розширень які можна встановити через розширення. Щодо переваг, то VS Code наповнений різноманітними підтримками та відкладчиками, які не надуть користувачеві загубитись у власній справі та не помилитись у написанні коду. Крім того всередині середовище вбудований термінал та є Git-інтеграція. Крім того на відмінну від двох попередніх середовищ, VS Code не

створює проблеми з ресурсомісткістю. І після аналізу цих середовищ, я прийшла до висновку що мені буде зручно використати саме Visual Studio Code.

Отож, після вибору середовища, я розпочала пошук підходящої бази даних яка зберігатиме потрібну інформацію про користувача. Я мала змогу обрати між PostgreSQL, MongoDB та Firebase (Firestore). Адже усі вони мали свої переваги для реалізації моєї кваліфікаційної роботи:

- першим можливим моїм вибором є PostgreSQL. Це реляційна база даних, що славиться надійністю та стабільністю. Завдяки довгому часу розробки та покращенню з допомогою різноманітних компаній. Актуальність системи керування базами даних полягає в тому, що це сховище для інформації має чудову підтримку JSON. А також чудова величезна та ефективна підтримка складних запитів. Виконання різноманітних дії з великою кількістю інформації. І різноманіття розширень для створення специфічних функціоналів. Крім того у PostgreSQL є велика спільнота людей, які створюють чудову документацію аби допомагати колегам і бажаючим знати більше;

- другим варіантом став Firebase (Firestore) – орієнтована на документи база даних яка надає послуги хостингу, зберігання інформації у реальному часі та синхронізації між клієнтами [11]. До причин популярності цієї системи керування варто віднести масштабованості та безпеки адже Firebase – хмарний сервіс, що забезпечує регулярні резервні копії. Що зменшує у свою чергу втрат, пошкоджень інформації. Також перевагою є вбудована аутентифікація, як полегшує роботу та керування користувачами. Крім того великою перевагою цієї бази даних є можливість працювати у реальному часі. Тобто оновлення даних таблиці лідерів, сповіщення користувачів та інші функції. Є достатньо простою та зрозумілою для новачків і пропонує безкоштовний план що для невеликих проектів може бути достатньо;

- MongoDB – також як і попередня база даних орієнтована на документи. Але ця система чудово підходить для збереження ігрової інформації(рівень, досвід, кількість завдань тощо) у вигляді документів. Зручна та швидка у розробці та зберіганні JSON-подібних даних. Завдяки власним гнучким схемам,

ця база даних надає легку адаптацію у неочікуваних або раптових вимогах до даних [12].

Після аналізу баз даних, мій вибір впав на MongoDB, адже ця база даних чудово підходить мені у реалізації ігрових механік мого проекту. Завдяки таким перевагам як гнучкість схем а також можливість збереження ігрових даних.

Коли середовище та база даних були обрані, я зрозуміла що для реалізації власної кваліфікаційної роботи, мені потрібно створити скелет майбутнього вебзастосунку. Для цієї реалізації мій вибір був очевидним – HyperText Markup Language(HTML). Головне завдання якого полягає у визначенні логічної організації вебсторінки через використання тегів або елементів, так зазначено у роботі [13]. Ці теги дають вказівки браузеру щодо значення певного текстового фрагмента. Це дає змогу браузерам коректно розуміти й показувати інформацію, а також сприяє її доступності для пошукових систем та допоміжних технологій. HTML надзвичайно легкий в опануванні та надзвичайно практичний, адже його підтримують усі веб-переглядачі, гарантуючи правильну візуалізацію вмісту.

Завдяки HTML можна організувати дані з надзвичайною гнучкістю і він слугує базою для CSS та JavaScript. Як загальнодоступний стандарт, HTML постійно вдосконалюється, незмінно залишаючись незамінним інструментом у світі веб-розробки.

Після утвореного скелету застосунку, я використовуватиму CSS, або Cascading Style Sheets (Каскадні таблиці стилів) – це мова, яка доповнює HTML, керуючи візуальним оформленням та виглядом вебсторінок. До появи CSS стилі нерідко вбудовувались прямо в HTML-теги, що робило код складним, незручним для читання та внесення змін. CSS розв'язав цю проблему, дозволяючи розробникам прописувати правила стилізації в окремих файлах або блоках, саме так написано у роботі [14].

Перевагами цієї мови є великий спектр можливостей, які надає сучасний CSS, значно розширюючи функціонал та межі веб-дизайну. Завдяки йому, дизайнери мають абсолютний контроль над кожним аспектом вигляду елементів – від їхніх кольорів до розміщення. З впровадженням таких потужних

інструментів, як Flexbox та Grid, CSS відкриває шлях до створення адаптивних макетів, які бездоганно виглядають на будь-яких пристроях. Це також відчутно скорочує робочий час розробників, дозволяючи централізовано коригувати стилі в масштабах всього сайту.

Окрім цього, розподіл стилів позитивно впливає на швидкість завантаження веб-сторінок. CSS також дозволяє створювати інтерактивні компоненти та анімації, додаючи динаміки веб-сторінкам. Не менш важливим є те, що кросбраузерна сумісність сучасного CSS зводить до мінімуму проблеми з відображенням у різних браузерах.

І наступним кроком у моїй розробці було використання JavaScript та його бібліотеки React. А все тому що коли HTML формує каркас та CSS визначає зовнішній вигляд, JavaScript вдихає в веб-сторінки життя, додаючи їм інтерактивність та рух. Завдяки JavaScript веб-сторінки реагують на дії відвідувачів, змінюють вміст без потреби повного перезавантаження, контролюють медіафайли, створюють анімації та пропонують багато іншого. Саме JavaScript є «душею» сучасних вебдодатків, перетворюючи їх на динамічні та захоплюючі інтерфейси.

Ключем до успіху та відповідності сучасності JavaScript є його розгалужена екосистема фреймворків та бібліотек. Ці готові набори коду й інструментів значно пришвидшують процес розробки, пропонуючи розробникам перевірені рішення для вирішення типових задач та створення складних архітектур.

За статистикою роботи [15] з-поміж фронтенд-бібліотек та фреймворків, React стабільно утримує лідерство, надаючи компонентний підхід до розробки користувацьких інтерфейсів. За ним ідуть Vue.js, Angular а також Svelte. Та все ж першість впевнено тримає React і це чудово можна побачити на графіку що зображений на рисунку 2.8.

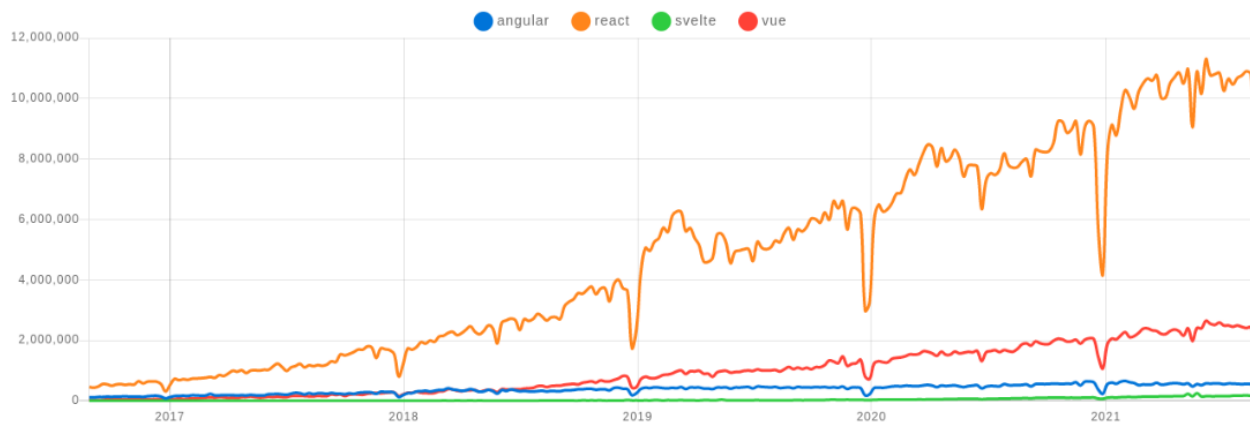


Рисунок 2.8 – Графік популярності бібліотек JavaScript [12]

Слідкуючи за графіком, можна помітити не аби який відрив популярності React. Адже саме тоді, коли розробник візуалізує інтерфейс у визначеному стані. React ефективно оновлює лише необхідні частини реального DOM, значно підвищуючи продуктивність додатків. Односпрямований потік даних сприяє передбачуваності керування станом, а можливість повторного використання компонентів відчутно прискорює розробку та забезпечує цілісність дизайну. React має підтримку від великої і активної спільноти, яка надає доступ до різноманітних ресурсів та інструментів, а використання JSX (JavaScript XML) робить код інтерфейсу зрозумілішим.

React знаходить застосування у передових світових корпораціях для розробки односторінкових додатків (SPA) та комплексних користувацьких інтерфейсів, де ключовим фактором є висока продуктивність. Його функціонал також розширюється на мобільну сферу завдяки React Native, який дозволяє розробляти нативні мобільні додатки, використовуючи базу React. Через свою результативність, універсальність та розвинену екосистему, React зберігає позиції лідера у сучасній веб-розробці, надаючи потужний метод для створення інтерактивних та високопродуктивних веб-застосунків.

Застосування цих фреймворків та бібліотек не лише пришвидшує процес створення програмного забезпечення, а й веде до розробки стійкіших,

масштабованіших та простіших у підтримці застосунків. Вони пропонують уніфіковані підходи до розробки, полегшуючи ефективну взаємодію команд.

Висновки до розділу 2

У другому розділі проведено детальний аналіз наявних UML-діаграм класів та діяльності, що дозволило окреслити передбачувані функції та сформулювати коректні функціональні й нефункціональні вимоги до майбутнього застосунку.

Згодом було зосереджено увагу на дизайні вебзастосунку, де детально висвітлено особливості та тонкощі, застосовані під час його розробки. Розглянуто специфіку роботи в середовищах розробки Visual Studio Code, IntelliJ IDEA та PyCharm, оцінено їхні позитивні та негативні сторони, що сприяло вибору оптимального інструменту.

Окрему увагу приділено дослідженню баз даних PostgreSQL, MongoDB та Firebase (Firestore), де проведено їхній порівняльний аналіз з акцентом на перевагах та недоліках кожної. На завершення цього розділу розглянуто ключові технології веб-розробки: HTML, CSS та JavaScript, а також проаналізовано бібліотеку React, її популярність та актуальність в контексті запланованої розробки.

РОЗДІЛ 3

РОЗРОБКА ІНТЕРАКТИВНОЇ НАВЧАЛЬНОЇ ВЕБПЛАТФОРМИ З ВИКОРИСТАННЯМ ГЕЙМІФІКАЦІЇ

3.1 Практична реалізація об'єкта проєктування

Під час написання вебзастосунку, я розпочала свою роботу зі структурування вкладок. Адже впорядкований код полегшує роботу в майбутньому. Тому я розділила кожну можливе вікно і при написанні скелету, дизайну я опиралась на створені папки для роботи в якій знаходяться файли з написаним скелетом та стилями. Тож вигляд мого структурування можна побачити на рисунку 3.1.

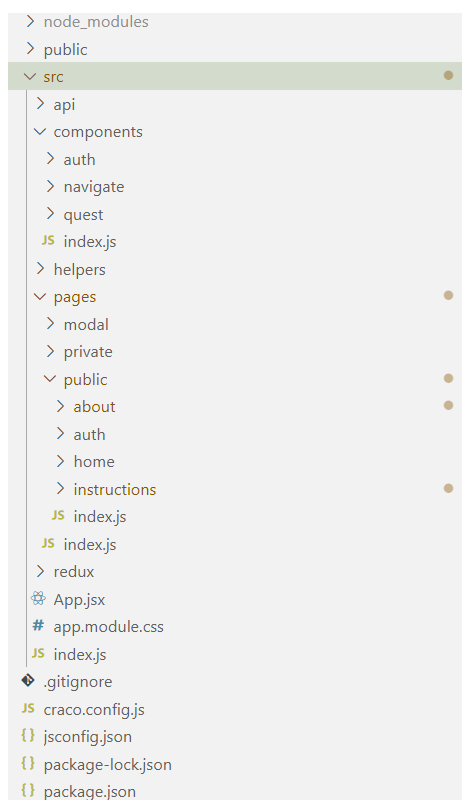


Рисунок 3.1 – Вигляд структурованих частин коду кваліфікаційної роботи

Отож, опираючись на дизайн Figma. Можна побачити що усі вкладки та можливі додаткові модульні вікна, я розділила на секції та описувала усіх їх окремо. Під час написання у мене не було складнощів щодо написання стилів.

Але особливість мого коду полягає в тому що весь головний контент зберігається всередині світлого контейнера, що слугує дошкою для наповнення. Це дозволяє мені створити відчуття ігрового екрану. Тому я старалась зберегти цю тенденцію у всьому моєму застосунку. Для того аби реалізувати навігацію я скористалась функцією NavLink. Це вузькоспеціалізований компонент, що призначений для побудови посилань. Він автоматично застосовує клас активності до певного елемента, коли його пропса «to», тобто шлях збігається з поточною адресою веб-сторінки в URL. Приклад частини таких посилань зображено на лістингу 3.1.

Лістинг 3.1– Частина навігаційного коду з використанням NavLink

```
<NavLink className={style.link} to="/character">
    Прогрес
</NavLink>
```

кінець лістингу 3.1

Після чого я продовжила напиння коду через елементарну систему маршрутизації для веб-застосунку React, використовуючи react-router-dom v5. Він забезпечує можливість навігації між різними розділами «Головна», «Про нас» та «Допомога», а також містить логіку захисту маршрутів з огляду на стан автентифікації користувача. Для того аби спростити процес завантаження, я використала таку функцію як Switch. Вона дозволяє відображати лише перший дочірній Route, тобто шлях, що відповідає поточному URL. Також розглядаючи мій код можна помітити вжитий компонент BrowserRouter. Це базовий елемент для навігації у веб-застосунках, котрий задіює HTML5 History API з метою підтримки відповідності інтерфейсу користувача та URL-адреси. Вигляд коду зображено на лістингу 3.2.

Лістинг 3.2 – Реалізація маршрутизатора для веб-застосунку

```
<BrowserRouter>
  <Navigate />
  {this.props.modal && <Modal />}
  <div className={style.app}>
    <Switch>
      <Route path="/" exact component={Home} />
      <Route path="/about" component={About} />
```

```

    <Route path="/instructions" component={Instructions} />
    {isAuth && <Route path="/character" component={Character} />}
    {isAuth && <Route path="/quests" component={Quests} />}
    {!isAuth && <Route path="/auth" component={Auth} />}
    <Redirect to={isAuth ? "/character" : "/auth/login"} />
  </Switch>
</div>
</BrowserRouter>

```

кінець лістингу 3.2

Після того як усі основа була створена. Наступним кроком була робота над інтерактивною частиною застосунку. А саме нарахування користувачу очків за виконання завдання. Отож першим і головним завданням було правильно скласти формулу для розрахунку кількості очків, які потрібно набрати для переходу у наступний рівень. Ця формула слугує для розрахунку експоненційного зростання досвіду, необхідного для досягнення кожного наступного рівня у системі гейміфікації UpTree. Я описала цю формулу з розрахунком що число 1,25 (або $1+0,25$) вказує на збільшення на 25 %. Отже, для кожного наступного рівня, початкову кількість балів необхідно помножити на 1,25. яка буде збільшувати рівень нарахування очків для переходу на наступний рівень. Цей відсоток є одним з найкращих рішень, адже забезпечує рівномірний зріст. На лістингу 3.3 продемонстровано яку формулу було створено.

Лістинг 3.3 – Вигляд формули для вирішення складності рівня

```

exports.toNextLvl = (lvl) => {
  const res = Math.round(100 * Math.pow(1.25, (lvl-1)))+20;
  return res; };

```

кінець лістингу 3.3

Після цього я розпочала працювати над візуальною частиною, а саме над індикатором прогресу. Важливо було аби нараховані бали відповідали шкалі прогресу. Отже, розглянемо реалізацію прогрес-бару в React.js для відображення досвіду у системі. Подано приклад коду, що демонструє спосіб візуалізації поточного рівня досвіду користувача, а також загальну кількість досвіду, необхідну для переходу на наступний рівень. Використовуючи ці показники, обчислюється відсоток заповнення. Цей відсоток, у свою чергу, застосовується

для динамічної зміни ширини внутрішнього елемента div. Фінальний результат – візуальний індикатор прогресу, довжина якого змінюється пропорційно накопиченому досвіду (EXP). Це надає користувачеві зрозумілу картину прогресу та наближення до нового рівня. Те як я реалізувала цю ідею продемонстровано на лістингу 3.4.

Лістинг 3.4 – Код для відображення прогресу користувача

```
<div className={style.progressBar}>
  <div className={style.progressHead}>
    <span>Exp</span>
    <span>
      {xp}\{toNextlvl}
    </span>
  </div>
  <div style={{ width: percentage + "%" }}
    className={style.progressProgrs}></div>
  </div>
</div>
```

кінець лістингу 3.4

Коли формула для вирахування очків рівня та індикатор прогресу були створені. Я розпочала працювати над ще однією візуальною частиною застосунку. Візуальний символ веб застосунку – дерево, яке користувач бачить на вкладці «Прогрес». Для того аби користувач відчував зацікавленість та вмотивованість, я добавила візуально винагороду. А саме зміну вигляду дерева. Для того аби користувач відчував що разом з його знаннями та рівнем росте і його дерево. Для цього я використала декілька svg-картинок, які використовуватиме система.

Отже, алгоритм для розробки цього уривку коду виглядає так. Спочатку, система аналізує дані про користувача, щоб визначити наявний досвід та кількість балів, необхідних для переходу на новий рівень. Далі, система звертається до власного списку, щоб отримати перелік стадій росту дерева. Після цього відбувається обчислення. Система враховує ваш поточний досвід та різницю між поточним і необхідним рівнем, і перетворює отримане значення у масштаб «від 0 до кількості доступних видів дерев». В кінці, система оцінює

розраховану «стадію росту» та обирає відповідне зображення дерева з каталогу. Приклад коду який я використала у своїй кваліфікаційній роботі зображено на лістингу 3.5.

Лістинг 3.5 – Код для висвітлення різних зображень дерева

```
exports.tree = async (req, res) => {
  const { xp, toNextlvl } = req.user;
  const tree_list = await getAllTree();
  const tree = Math.floor((xp * tree_list.length) / toNextlvl);
  return tree_list[(tree >= tree_list.length ? tree_list.length - 1 : tree)];
};
```

кінець лістингу 3.5

Завдяки цьому участку коду, користувачі матимуть змогу отримати візуальну нагороду у вигляді зростаючого дерева, що позначатиме їх власне зростання.

3.2 Тестування та налагодження інформаційно-комп'ютерної системи

Тестування проєкту «UpTree» – це не просто примха, а життєво важлива стадія розробки, яка забезпечує його успіх, стійкість та позитивне враження від користування. Без адекватного тестування існує ризик випуску продукту з дефектами, що може спричинити втрату довіри споживачів і навіть фінансові втрати.

Головними умовами які я хочу наділити свій вебзастосунок це надійність. «UpTree» – це застосунок, що працює з даними користувачів (реєстрація, прогрес, завдання), і будь-які помилки тут можуть бути фатальними. Тестування дозволяє виявити та виправити баги, збої та непередбачувану поведінку до того, як застосунок потрапить до реальних користувачів. Перевірка злагодженої роботи з інформацією гарантує, що всі функції працюють так, як очікується: реєстрація проходить успішно, завдання створюються і зберігаються коректно, бали нараховуються правильно, а дерево росте відповідно до прогресу. Отож, я

приступила до тестуванні свого веб-застосунку я розпочала оглядати свій сервер з точки зору користувача і попросила своїх знайомих дещо мені допомогти.

Першим зауваженням, яке ми помітили було невелика кількість символів при введенні пароля. Перші варіанти зареєстрованих тестувальників, вводили 4 символи для паролю і реєстрація проходила успішно. Тому у частину з паролем було додано атрибут `minLength`. Цей атрибут використовується безпосередньо до поля введення пароля. Він вказує браузеру, що поле вводу має містити мінімум встановлену кількість знаків. У випадку спроби відправки форми, де символів менше, браузер заблокує надсилання і покаже стандартний повідомлення про валідацію.

Після чого я розпочала тестувати формулу щодо розрахунку кількості балів. Перша ідея формули була максимально простою та дещо примітивною. Вигляд цієї формули наведено у лістингу 3.6.

Лістинг 3.6 – Перша розрахункова формула рівню

```
exports.toNextLvl = (lvl) => {  
  const res = 100 + 30 * lvl;  
  return res; };
```

кінець лістингу 3.6

Тому після аналізу деякий ігрових моментів. Пошук більш логічної формули. Сенса формули полягав у відсотковому зростанні. Це зумовлено рядом позитивних аргументів. Адже з кожним новим рівнем гравець змушений буде докладати дедалі більше старань задля досягнення поставленої мети, що породжує відчуття прогресуючого виклику. Це дозволяє уникнути одноманітності на пізніх стадіях використання.

Завдяки чому користувач має змогу отримати відчуття перемоги – здолаття перешкод, які значно більші, що дарує відчутно більше насолоди. І з точки зору розробки формула повинна бути гнучка та легка для розуміння та реалізації.

Під час пошуку даних про підвищення рівня, стало відомо, що формули, які оперують так званим середнім відсотком, від 15 % до 25 %, вважаються оптимальним варіантом, забезпечуючи помітне зростання складності, водночас не сповільнюючи прогрес на початкових етапах. Тому спочатку у формулі було використано відсоткове збільшення у 15 %, а кількість основних балів для переходу з першого на другий рівень була встановлена на рівні 100. Обчислення показали незначне збільшення рівнів, що змусило підвищити відсоток до 20 %. Однак, аналіз зростання кількості балів не відповідав потребам, тому кінцева формула отримала фіксоване відсоткове збільшення у 25 %.

І коли формула була сталою, я приступила до створення умов при яких відбуватиметься зміна вигляду дерева. Першою ідеєю було те аби з розвитком рівнів змінювався вигляд дерева. Але це сповільнює відчуття прогресу і кінцевого рівня в моєму застосунку немає. Тому я вирішила, що цю задумку варто втілити у сам розвиток абсолютно кожного рівня.

Для цього мені потрібно було динамічно обчислювати відсоток заповнення індикатора прогресу, спираючись на поточний досвід користувача та досвід, необхідний для досягнення наступного рівня. Отримане значення відсотка застосовується як інлайновий стиль «width» до внутрішнього div-елемента, щоб візуалізувати прогрес за допомогою зміни ширини графічного елемента.

Тому початкова формула виглядала так як зображено на лістингу 3.7.

Лістинг 3.7 – Вигляд першого коду для візуалізації прогресу

```
exports.tree = async (req, res) => {
  const { xp, toNextlvl } = req.user;
  const tree_list = await getAllTree();
  const tree = Math.floor((xp * tree_list.length) / toNextlvl);
  return tree_list[tree];};
```

кінець лістингу 3.7

Але при тестуванні цієї механіки, було виявлено що при наборі кількості балів рівно максимуму кількості балів з можливих, картинка дерева зникла зовсім.

Тому код довелося змінювати, і у рядок з поверненням додати зміни. І таким чином рядок коду видозмінився як зображено у лістингу 3.8.

Лістинг 3.8 – Видозмінна коду, яка виправляє помилку зникнення дерева

```
return tree_list[  
tree>=tree_list.length?tree_list.length-1:tree];
```

кінець лістингу 3.8

У цьому рядку я використала тернарний оператор або його також називають умовним, вирізняється в багатьох мовах програмування, через те, що єдиний оператор, який використовує три операнди. Він дає змогу стисло записувати умовні вирази, що за інших умов потребували б використання розширених конструкцій if...else.

Отже, застосувавши тернарний оператор, система перевіряє, чи розрахований індекс tree не виходить за межі розміру масиву tree_list. Якщо значення tree є більшим або рівним довжині tree_list, тоді повертається останній елемент масиву а саме індекс tree_list.length - 1. Це потрібно, щоб уникнути помилок через вихід за межі масиву. В іншому разі, якщо індекс tree в допустимому діапазоні, повертаємо елемент з масиву за вказаним індексом tree. Цей рядок коду гарантує, що функція завжди повертає коректний елемент з tree_list.

Для прикладу я створила умовну ситуацію, під час якої могла виникнути проблема уже після виправленого коду. На рисунку 3.2 показано зображено створену можливу ситуацію, під час якої раніше відбувалась поломка.



Рисунок 3.2 – Виправлена конфліктна ситуація

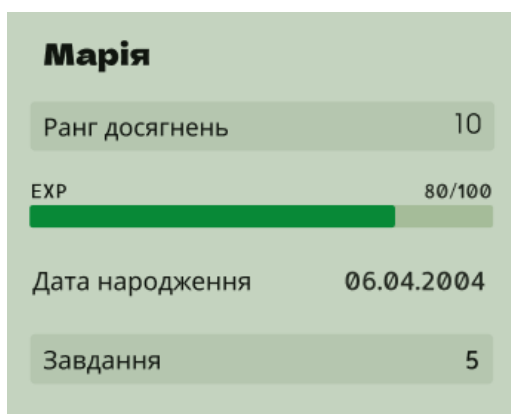
Як ми бачимо, красиве, доглянуте дерево нікуди не зникло. Код працює і успішно проходить тестування на різних рівнях.

3.3 Розробка бази даних

Моя кваліфікаційна робота є залежною від збереження інформації про користувачів та їх створених завданнях. Отож потреба у використанні середовища для роботи з великою кількістю інформації є актуальною. І після проведення аналізу підходящих систем для роботи з даними. Мій вибір зупинився на MongoDB та його можливостях. Як система, управління базами даних з орієнтацією на документи, типу NoSQL, MongoDB пропонує чимало плюсів у порівнянні з іншими видами баз даних. Можливість зберігати дані в JSON форматі є великою перевагою, адже цей формат це форма обміну даними у текстовому вигляді, що має легкість для сприйняття людиною та простоту розбору для машин. Цей формат ґрунтується на невеличкій частині JavaScript, проте функціонує цілком самостійно від цієї мови.

Отож для початку я проаналізувала яку інформацію потрібно зберегти і яким чином її можна пов'язати. Головною інформацією яку вноситиме користувач для майбутнього користування це власне своє ім'я та пароль. Це потрібно для того або аутентифікуватись на сайті та мати змогу користуватись вкладками «Прогрес» та «Завдання».

Наступна частина інформації це ранг якого досягнув користувач, рівень який він набрав у шкалі прогресу, загальна кількість завдань що знаходиться на дошці завдань. А також збереження кількості досвіду яку потрібно набрати для того аби отримати новий ранг і перейти на наступний рівень. Уся ця інформація пов'язана з цією частиною застосунку яку зображено на рисунку 3.3.

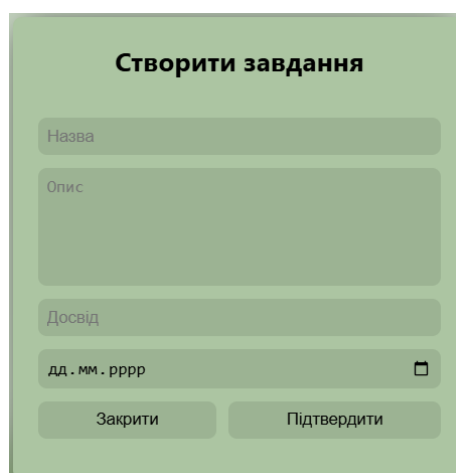


Марія

Ранг досягнень	10
EXP	80/100
Дата народження	06.04.2004
Завдання	5

Рисунок 3.3 – Приклад вигляду інформації яку потрібно зберегти

Наступною окремою частиною для збереження, є вкладка «Завдання» на якій користувач створюватиме та виконуватиме завдання. Для того аби зберегти усе що стосується карточки з завданням. То при створені завдання приклад початкової пустої форми зображено на рисунку 3.4.



Створити завдання

Назва	
Опис	
Досвід	
дд . мм . рrrr	☐
Закрити	Підтвердити

Рисунок 3.4 – Пуста форма для заповнення інформації завдання

Потрібно заповнити такі пункти як назву та опис завдання, кількість досвіду за завдання, час до виконання а також статус завдання. Головним завданням було створити правильний зв'язок для того аби при створенні завдання, воно зберігалось у певної особи. Для цього при створенні бази даних, я використала унікальний ідентифікатор кожного користувача тобто ID користувача який надається при реєстрації кожного користувача. ID є первинним ключем, що визначає кожен запис у колекції Користувач. Тому вигляд зв'язку таблиць, я продемонструвала на рисунку 3.5.

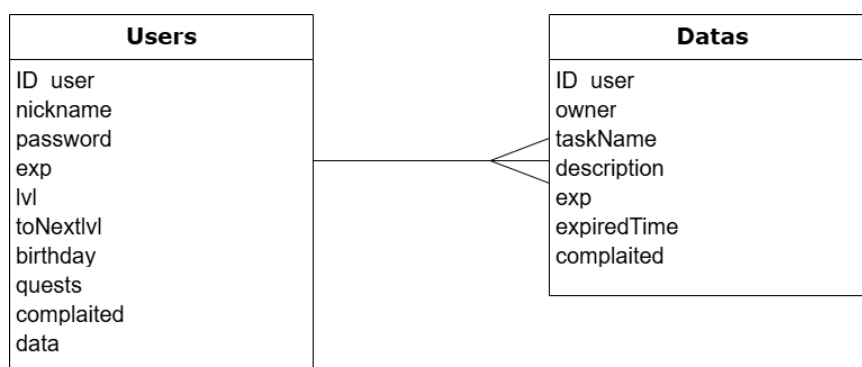


Рисунок 3.5 – Діаграма відношення

Якщо розглянути діаграму, то можна побачити між сутностями Користувач та Дані існує взаємозв'язок, позначений лінією з розгалуженням з боку Даних та однією лінією з боку Користувача це зображує зв'язок «Один-до-багатьох». Цей вид взаємодії вказує на те, що один користувач може мати безліч записів даних або завдань. І після того як логіку бази даних було створено. Я перейшла до реалізації у вигляді коду.

Для початку я впевнилась у тому що усі компоненти які потрібні мені для розробки уже приєднані Node.js, Express, Mongoose до моєї розробки. Після чого розпочалась розробка серверної частини застосунку що тісно пов'язана з базами даних. Отож, створивши базову структуру для того аби все злагодженого працювало. Я розпочала писати схему для колекції Users, лістинг коду якого продемонстровано у лістингу 3.9.

Лістинг 3.9 – Код реалізації таблиці User

```

<div className={style.info}>
  <div>
    <h1>{name}</h1>
  </div>
  <div>
    <div className={style.highlight}>
      <span>Ранг</span>
      <span>{lvl}</span>
    </div>
    <div className={style.progressBar}>
      <div className={style.progressHead}>
        <span>Exp</span>
        <span>
          {xp}\{toNextlvl}
        </span>
      </div>
      <div className={style.progressBar}>
        <div style={{ width: percentage + "%" }}
className={style.progressProgrs}></div>
      </div>
    </div>
    <div className={style.highlight}>
      <span>Дата народження</span>
      <span>{birthday}</span>
    </div>
    <div className={style.highlight}>
      <span>Завдання</span>
      <span>
        {complaited}\{quests}
      </span>
    </div>
  </div>
</div>

```

кінець лістингу 3.9

У цьому додатку можна побачити те, як я створила критерії для заповнення інформації щодо користувача. При збереженні інформації, пункт data наповнюється масивом в якому збережені унікальні коди що відповідають пункту owner. Приклад збереженої інформації наведено нижче на рисунку 3.6.

```

_id: ObjectId('6834d0d730330e2c89b34b70')
name: "Pio"
password: "$2b$10$JGvv082LBbuHh3dC3KyaKehKwn56P2df8FccYp2Li16k1vPlzmcPK"
xp: 20
lvl: 1
toNextlvl: 120
birthday: "2025-05-26"
quests: 2
complaited: 1
data: Array (2)
  0: "6834d0f030330e2c89b34b7b"
  1: "6834d16f30330e2c89b34bd6"

```

Рисунок 3.6 – Приклад збереження інформації

На рисунку можна помітити що інформація ID користувача та пароль зображена у вигляді певного набору букв, цифр та символів. Це зроблено з метою забезпечення безпеки особистих даних користувача. Крім того можна помітити що інформація про завдання зберігається як масив з певним унікальним кодом кожного завдання. Реалізацію коду для збереження інформації щодо рисунку вище у лістингу 3.10.

Лістинг 3.10 – Вигляд коду для заповнення інформації щодо другої таблиці

```
const dataSchema = new Schema( {
  owner: { type: String, required: true },
  name: { type: String, required: true },
  task: { type: String, required: true },
  xp: { type: Number, default: 0 },
  expiredTime: { type: Date },
  complaited: { type: Boolean, default: false },
}, { versionKey: false,
  } );
module.exports = model("data", dataSchema);
```

кінець лістингу 3.10

Уся ця інформація відправляється на збереження у база даних MongoDB у JSON форматі. До лістингу вище, де зберігається інформація про завдання. Приклад збереженої інформації в базі даних MongoDB зображено на рисунку 3.7.

```
_id: ObjectId('6834d0f030330e2c89b34b7b')
owner : "6834d0d730330e2c89b34b70"
name : "Завдання"
task : "Виконання 1"
xp : 100
expiredTime : 2025-05-26T00:00:00.000+00:00
complaited : false
```

```
_id: ObjectId('6834d16f30330e2c89b34bd6')
owner : "6834d0d730330e2c89b34b70"
name : "Завдання"
task : "Виконано 2"
xp : 20
expiredTime : 2025-05-26T00:00:00.000+00:00
complaited : true
```

Рисунок 3.7 – Приклад інформації збереженої у JSON форматі

Таким чином я реалізувала збереження користувацьких даних за допомогою MongoDB та додаткових інструментів.

3.4 Забезпечення інструктивної інформації користувача

Новачки часто губляться, опинившись у незнайомому застосунку, особливо якщо він містить елементи гри, ранги та особливі системи. Доступні інструкції швидко ознайомлять їх з ключовими можливостями: як створити обліковий запис, як знайти перше завдання, як функціонує система досвіду/рівнів. Це знижує «бар'єр входження» та уникнути відтоку на початковому етапі взаємодії.

Наявність розділу з документацією, інструкціями та корисними порадами має вирішальне значення для будь-якої програми, особливо для платформ з елементами гейміфікації, як UpTree. Це не просто додатковий бонус, а фундаментальний аспект, що забезпечує успішність продукту, поліпшує взаємодію з користувачем та сприяє утриманню аудиторії. Цей розділ є важливим для користувача, оскільки полегшує адаптацію та знижує бар'єр входу.

Для нових користувачів це фактично посібник у невідомому середовищі, адже навіть найінтуїтивніша програма має свої нюанси. Він дозволяє користувачеві отримати чіткі вказівки, замість того, щоб відчувати розгубленість або гадати, як щось функціонує. Отже, зменшується вірогідність того, що користувач покине програму, так і не зрозумівши її суті, та сприяє його швидшому залученню в екосистему UpTree. Цей розділ також відіграє ключову роль у розумінні механізмів гейміфікації та стимулюванні прогресу. У гейміфікованих системах, як-от UpTree, критично важливо, щоб користувачі розуміли «правила гри». Цей розділ пояснює, як нараховуються бали досвіду (EXP). Тому у створенні тексту цієї вкладки я серйозно підійшла до дружельюбного тексту.

Одним з моїх хвилювань було те що користувач зловживатиме балами та виставлятиме їх не рівноцінно до складності власного завдання. Тому у вкладці «Довідка», яку користувач вільно може відвідати знаходиться частина порад щодо цього.

Інструкції та поради реалізовані у такій же стилістиці як і весь інший сайт. Поради щодо правильної оцінки завдань допомагають користувачам уникнути «знецінення» прогресу та підтримувати реалістичний рівень виклику. Коли користувач розуміє, як працює система і як він може максимізувати свій прогрес, це суттєво підвищує його мотивацію та залученість, допомагаючи йому ставити реалістичні цілі та відчувати справжнє досягнення. Підсумовуючи, наявність цього розділу сприяє підвищенню довіри та прозорості. Коли програма надає чіткі та доступні інструкції, а також роз'яснює свої механізми, це формує відчуття відкритості та надійності. Користувачі бачать, що розробники піклуються про їхній досвід і готові надати всю необхідну інформацію. Це зміцнює довіру до платформи та заохочує довгострокову взаємодію, оскільки користувач почувається впевненіше та захищеніше.

3.5 Захист HTTPS з'єднання

Дуже важливою складовою на сайті є забезпечення користувачів конфіденційністю. Якщо паролі або приватна інформація буде надсилатись без шифрування, існує високий ризик їхнього перехоплення зловмисниками. Це може спричинити злам облікових записів, розголошення особистих даних, що суттєво підірве довіру користувачів.

Тому забезпечення безпеки даних гарантує MobgoDB. Захист даних у MongoDB – це комплексний підхід, який охоплює різні аспекти, щоб спільно забезпечити міцний захист інформації. Першочергово важливим є процес автентифікації, що верифікує особу користувача, який прагне отримати доступ до бази даних. Це ключовий крок для запобігання несанкціонованому доступу,

адже без нього будь-хто, хто має доступ до мережі серверу MongoDB, міг би вільно взаємодіяти з даними.

Ця база даних підтримує різноманітні механізми автентифікації, серед яких найбільш розповсюдженим та рекомендованим є SCRAM, який використовує логін/пароль з хешуванням та сіллю для захисту. Додаткові опції включають x.509 Client Certificates для більш потужної автентифікації на базі цифрових сертифікатів, а також інтеграцію з корпоративними системами управління ідентифікацією.

Після успішної автентифікації, вступає в дію авторизація, яка визначає, які саме операції дозволено виконувати аутентифікованому користувачу на конкретних ресурсах, як от колекції чи бази даних. Це втілює принцип мінімальних привілеїв, що обмежує потенційні втрати у випадку компрометації облікового запису, надаючи користувачам доступ лише до тих даних та функцій, які їм справді необхідні. MongoDB застосовує Role-Based Access Control (RBAC), де користувачам призначаються ролі з конкретними дозволами. Це можуть бути як вбудовані ролі, що надають базові дозволи на читання або запис, так і кастомні ролі, створені для точного врахування потреб застосунку.

Ключовим елементом захисту є шифрування, яке перетворює дані на нечитабельний формат. Це гарантує, що навіть у випадку несанкціонованого доступу до файлів бази даних або перехоплення мережевого трафіку, без ключа шифрування дані залишаться нерозшифрованими. Приклад такого набору символів продемонстровано на рисунку 3.8.

```
_id: ObjectId('6834d05ed6142944aea38369')
name : "Марія"
password : "$2b$10$FFw.5QMcoPFM02U/uEq7x.brFwmOgtQm028boQhftHEluRrqpJUni"
```

Рисунок 3.8 – Демонстрація зашифрованої інформації

Шифрування застосовується як для даних «в стані спокою», так і для даних «в русі». Для шифрування «в русі» критично важливо використовувати TLS/SSL, забезпечуючи безпеку з'єднання між вашим Node.js бекендом та MongoDB.

MongoDB Enterprise Edition також пропонує шифрування на рівні полів, що дозволяє шифрувати окремі поля безпосередньо на клієнтській стороні, роблячи їх недоступними навіть для адміністраторів бази даних. Мережева безпека сфокусована на обмеженні доступу до сервера MongoDB на рівні мережі. Це мінімізує «поверхню атаки», дозволяючи підключення до бази даних тільки з певних, довірених IP-адрес або мереж, наприклад, IP-адрес вашого бекенд-сервера, використовуючи фаєрволи.

Для застосунку UpTree, який працює з чутливими даними користувачів, впровадження цих аспектів є критично важливим. Завжди підключайтесь до MongoDB з обліковими даними користувача, котрий має відповідні ролі, налаштуйте ці ролі для вашого бекенду, щоб забезпечити принцип найменших привілеїв до колекцій Users та Datas. Забезпечте TLS/SSL шифрування для усіх з'єднань між вашим Node.js застосунком і MongoDB, а також налаштуйте мережеву безпеку, дозволяючи доступ до порту MongoDB лише з довірених IP-адрес.

Висновки до розділу 3

У цьому розділі я продемонструвала свої знання та навички, працюючи над розробкою власного вебдодатку UpTree. Зокрема, я збудувала каркас сайту, використовуючи бібліотеку React, надала йому стилю за допомогою CSS та інтегрувала всю логіку застосунку за допомогою JavaScript.

А також, у цьому розділі я продемонструвала роботу з базами даних. Я створила діаграму взаємодій, аби візуалізувати зв'язки між двома таблицями. Окрім цього, я обґрунтувала необхідність створення та додавання інформаційної сторінки до мого проєкту. Окремо було вивчено та описано надійність бази даних MongoDB та засоби захисту даних користувачів. Також було проведено тестування мого додатку та поділено досвід виправлення помилок, підбору формул та знаходження рішень проблем, що виникли в процесі розробки UpTree.

ВИСНОВКИ

У кінцевому результаті реалізації проекту було успішно розроблено веб-додаток з інтеграцією гейміфікації, що демонструє повну відповідність технічним та функціональним специфікаціям.

Було проаналізовано гейміфікацію, що дало змогу встановити основні механізми та засади гейміфікації, котрі результативно спонукають користувачів у навчанні. Вивчення відомих онлайн-ресурсів посприяло виявленню вдалих прикладів впровадження гейміфікованих складових.

Після цього було визначено функціональні вимоги (відстеження прогресу користувача, система завдань та винагород, механізми взаємодії між користувачами, а також можливість доступу до довідкових матеріалів) й нефункціональні вимоги, а тобто критерії якості, безпеки, продуктивності, здатність до розширення, можливість системи працювати зі збільшенням кількості користувачів.

Проведено ретельний аналіз та відбір технологічного стеку. Після проведеного аналізу середовищ розробки (Visual Studio Code, IntelliJ IDEA і PyCham), крім того проведено вибір баз даних серед PostgreSQL, MongoDB та Firebase/Firestore. Також було досліджено та включено основні технології вебдодатків – HTML, CSS та JavaScript. І обґрунтовано причини вибору React.js. Рішення було прийняте з врахуванням сучасних трендів у галузі, активності спільноти розробників, а також стабільної підтримки обраних технологій.

Розробка інтерфейсу користувача, котрий був би водночас інтуїтивним та візуально приємним. Це перетворилося на основну ціль для зацікавлення та утримання користувачів.

Крім того було проведена ретельна перевірка критичних складових фронтенду та бекенду, адже це є надзвичайно важливою для забезпечення правильної роботи та безперебійної стабільності веб-додатку.

Також була розроблена ефективна архітектура бази даних. Надійне зберігання та управління всією інформацією про користувачів, їхній поступ і

навчальні матеріали є важливою складовою вебплатформи було реалізовано за допомогою MongoDB. Розробка та втілення оптимальної структури гарантують високу швидкість доступу до даних, цілісність інформації та легкість її масштабування в міру збільшення кількості користувачів та обсягів інформації в UpTree.

Створено розділ «Довідка», який є значущим кроком у полегшенні адаптації нових користувачів та забезпеченні їх повною інформацією про можливості UpTree.

Було впроваджено протокол HTTPS задля безпеки та приватності даних користувача. Адже зашифрований трафік стає щитом від крадіжки даних та стороннього втручання, забезпечуючи цілісність інформації та укріплюючи довіру користувачів до UpTree. А також досліджено та описано надійність бази даних MongoDB та методи захисту інформації користувачів за допомогою HTTPS.

Ця робота відображає мій комплексний підхід до розробки, від розуміння теоретичних основ до успішної практичної реалізації та налагодження складних технічних систем. Розроблена вебплатформа, що містить елементи гейміфікації, цілковито відповідає заявленим потребам, дозволяючи користувачам взаємодіяти зі своїми цілями. Платформа демонструє готовність до подальшого удосконалення та розширення, гарантуючи відмінну якість реалізованого продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Статистика Duolingo. URL: <https://investors.duolingo.com/news-releases/news-release-details/duolingo-hits-100m-maus-reports-59-dau-growth-and-41-revenue> (дата звернення: 12.03.2025).
2. Duolingo. URL: <https://uk.duolingo.com/> (дата звернення: 06.03.2025).
3. Habitica. URL: <https://habitica.com/static/home> (дата звернення: 06.03.2025).
4. Kahoot! URL: <https://kahoot.com> (дата звернення: 06.03.2025).
5. Понад 70 % українців відчувають стрес або знервованість. Гвара Медіа. URL: <https://gwaramedia.com/ponad-70-ukraincziv-vidchuvayut-stres-abo-znervovanist> (дата звернення: 18.04.2025).
6. Протокол лікування РДУГ в Україні. Дорослі з РДУГ. Дорослі з РДУГ. Мета ГО «Дорослі з РДУГ» – забезпечення визнання та ефективного лікування розладу дефіциту уваги та гіперактивності (РДУГ) у дорослих в Україні. URL: <https://rdug.info/knowledge-base/protokol-rdug-vid-moz> (дата звернення: 22.04.2025).
7. Guide. Gamification at work URL: <https://www.iseazy.com/guides/gamification-at-work/> (дата звернення: 18.04.2025).
8. Free Online UX Design Tool & UX Software. Figma. URL: <https://www.figma.com/ux-design-tool/> (дата звернення: 21.04.2025).
9. Цикало С. С., Фефелова З. Є. вПсихологічний вплив синього та зеленого кольорів на людину. URL: <https://openarchive.nure.ua/server/api/core/bitstreams/c4d35e0e-97fc-4002-a2bd-0ea2f669e681/content> (дата звернення: 23.04.2025).
10. JetBrains. IntelliJ IDEA – the IDE for Pro Java and Kotlin Development. URL: <https://www.jetbrains.com/idea/> (дата звернення: 23.04.2025).
11. Firebase як бекенд для будь-яких застосунків. URL: <https://dou.ua/forums/topic/44058/> (дата звернення: 26.04.2024).

12. MongoDB. URL: <https://www.mongodb.com/> (дата звернення: 28.04.2025).

13. Сучасний підручник та довідник з HTML. URL: <https://html.tutorial.in.ua/what-is-html> (дата звернення: 05.05.2025).

14. Що таке CSS для веб-розробника: вся базова інформація. ITSTEP Академія Київ. URL: <https://kiev.itstep.org/blog/what-is-css-and-why-does-a-web-developer-need-it> (дата звернення: 10.05.2025).

15. Огляд фреймворків JavaScript. Oleh Mell. URL: <https://dou.ua/forums/topic/34739/> (дата звернення: 30.05.2025).

ДОДАТКИ

Додаток А

Підтвердження апробації результатів кваліфікаційної роботи

Міністерство освіти і науки України
 Волинська обласна рада
 Луцька міська рада
 Луцький національний технічний університет
 Національний технічний університет України «Київський політехнічний
 інститут імені Ігоря Сікорського»
 Національний університет «Львівська політехніка»
 Військовий інститут телекомунікацій та інформатизації
 імені Героїв Крут (м. Київ)
 Тернопільський національний педагогічний університет імені В. Гнатюка
 Рівненський державний гуманітарний університет
 Навчально-науковий інститут «Українська інженерно-педагогічна академія»
 Харківського національного університету ім. В.Н. Каразіна
 Люблінська політехніка (Польща)
 Фрайберзька гірнича академія (Німеччина)
 Гронінгенський університет (Нідерланди)
 Кавказький університет (Грузія)
 Політехнічний університет Браганси (Португалія)



ТЕЗИ ДОПОВІДЕЙ
X Міжнародної науково-практичної конференції з
проблем вищої освіти і науки «Інформаційні технології
в освіті, науці і виробництві (ІТОНВ-2025)

23-24 травня 2025 р.

Луцьк 2025

Shumyliak Liliia	Implementation of web technologies in	271
Cibák Luboš	the management of medical department	
Tarnovetska Olha	workflows	

СЕКЦІЯ 5. ІНФОРМАЦІЙНІ ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ І ТЕХНОЛОГІЇ

Антонюк А.О.	Голосовий помічник на основі	275
Повстяна Ю.С.	штучного інтелекту	
Вовк П.Б.	Підготовка середовища та налаштування локального кластера Kubernetes: від теорії до практики	282
Вознюк А.В.	Розробка 2D ігор на unity з генерацією	286
Данилко К.С.	світу	
Вознюк М.В.	Аналіз впливу темної теми на зручність	289
Неділько О.В.	користування веб-додатками	
Вознюк А.В.	Автоматизація сервісів доставки їжі за	293
Хітько Ю.В.	допомогою Telegram-бота з AI та платіжним модулем	
Вознюк А.В.	Технічні рішення та освітні переваги	298
Чоботар Ю.С.	при створенні і впровадженні	
Примич М.О.	вебплатформи для онлайн-навчання	
Газдюк К.П.	Адаптивне планування логістичних	304
Дмитрашук К.М.	маршрутів в умовах воєнного стану	
Кривинчук В.О.		
Головня С.А.	Застосування сучасних бібліотек Python	308
	для задач з аналізу та прогнозування	
Демідов П.Г.	Порівняльна характеристика нейро-	312
Андрієнко В.А.	нечітких та нейронних технологій на	
Маргаза Д.Ю.	прикладі розв'язання задачі	
Муляр В.П.	прогнозування вартості валюти на фінансовому ринку	

УДК 004.85

ТЕХНІЧНІ РІШЕННЯ ТА ОСВІТНІ ПЕРЕВАГИ ПРИ СТВОРЕННІ І ВПРОВАДЖЕННІ ВЕБПЛАТФОРМИ ДЛЯ ОНЛАЙН-НАВЧАННЯ

Вознюк Анастасія Вадимівна

Луцький національний технічний університет, асистент кафедри інженерія програмного забезпечення, a.vozniuk@lutsk-ntu.com.ua

Чоботар Юлія Сергіївна

Луцький національний технічний університет, студентка 4 курсу кафедри інженерія програмного забезпечення, chobotar.ly@gmail.com

Примич Марія Олександрівна

Луцький національний технічний університет, студентка 4 курсу кафедри інженерія програмного забезпечення, mashameky@gmail.com

TECHNICAL SOLUTIONS AND EDUCATIONAL BENEFITS IN THE DEVELOPMENT AND IMPLEMENTATION OF AN ONLINE LEARNING PLATFORM

Vozniuk Anastasiia Vadymivna

Lutsk National Technical University, Assistant at the Department of Software Engineering, a.vozniuk@lutsk-ntu.com.ua

Chobotar Yulia Serhiivna

Lutsk National Technical University, Student at the Department of Software Engineering, chobotar.ly@gmail.com

Prymych Maria Alexandrovna

Lutsk National Technical University, Student at the Department of Software Engineering, mashameky@gmail.com

The development of an online educational platform is presented, focusing on modern trends in distance learning accelerated by the COVID-19 pandemic. It has been established that such platforms provide flexible, accessible, and cost-effective learning opportunities. The integration of video courses and gamification elements enhances user engagement and supports self-paced learning. An experiment by the Ukrainian company CleverStaff demonstrates the effectiveness of gamified approaches in improving motivation. The study also outlines the main challenges of online education and emphasizes the growing relevance of hybrid models, microlearning, and AI-powered personalization.

Keywords: online education, distance learning, gamification, video courses, COVID-19 pandemic, digital transformation, web platforms.

Пандемія COVID-19 суттєво вплинула на всі сфери життя, зокрема й на освіту. Із запровадженням карантинних обмежень

мільйони учнів, студентів і викладачів по всьому світу були змушені у стислі терміни перейти на дистанційне навчання. Це стало поштовхом до стрімкого розвитку онлайн-освіти, яка до того моменту залишалася переважно альтернативною або додатковою формою навчання. Онлайн-платформи почали масово впроваджуватися в шкільній, університетській та післядипломній освіті, а технології відеозв'язку, інтерактивного контенту, хмарних сервісів і мобільних застосунків стали звичним інструментом освітнього процесу.

Зростання популярності онлайн-навчання добре ілюструє графік [1] частотності пошукових запитів за ключовими словами “online school”, “online education”, “online teaching” і “online learning” у період із 2019 по 2023 рік.

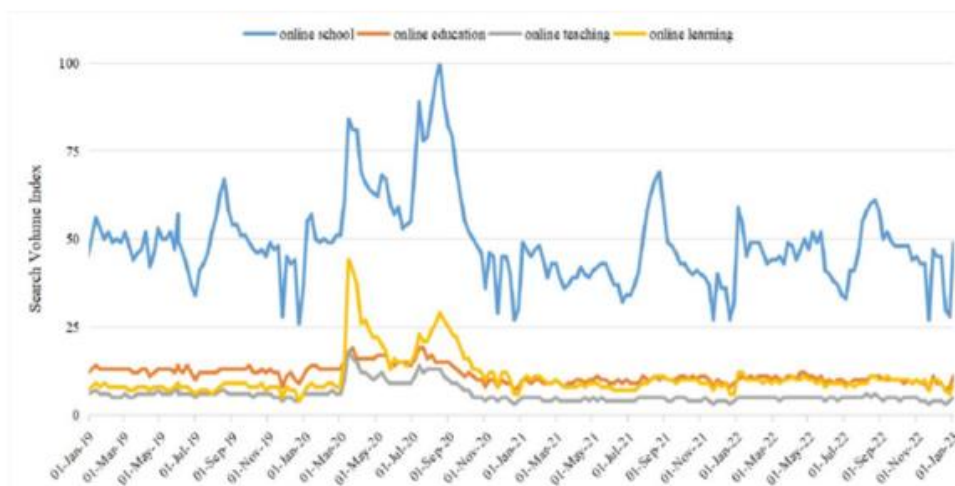


Рисунок 1 – Графік популярності пошукових запитів 2019-2023 років

На ньому чітко видно, що найвищий інтерес спостерігався на початку 2020 року, що прямо пов'язано з початком пандемії. Найпомітніше зростання демонструє запит “online school”, що цілком логічно, адже саме школи одними з перших були переведені в онлайн-формат. Інші запити також демонструють зростання, але менш виражене. Після піку у 2020 році рівень зацікавленості дещо знизився, однак залишився на стабільно

вищому рівні, ніж до пандемії, що свідчить про те, що онлайн-освіта закріпилася як невіддільна частина освітньої екосистеми.

Однією з головних причин популярності онлайн-формату стало те, що він забезпечує гнучкість і доступність: користувачі можуть навчатись у зручний для себе час і місці, маючи доступ до освітніх матеріалів з будь-якої точки світу. Такий формат дозволяє поєднувати навчання з іншими аспектами життя — роботою, хобі, сім'єю.

Також доступність онлайн освіти дає змогу людям з обмеженнями розвиватись прямо з дому і отримувати бажані знання уникаючи труднощів традиційного офлайн навчання.

Кожен студент чи учень має змогу підібрати для себе застосунки з різними методами освоєння інформації та можливістю обрати зручний час для навчання, повторення матеріалу. Крім того, цифрове навчання дозволяє значно зменшити витрати, пов'язані з поїздками, проживанням і друком матеріалів.

Під час пандемії активно почали розвиватись вебзастосунки та мобільні застосунки, які зосереджені на залученості та мотивації користувача.

Одним із ключових елементів сучасної онлайн-освіти стали відеокурси, які суттєво змінили підхід до подачі навчального матеріалу. Завдяки можливості запису лекцій у зручному форматі, викладачі можуть охопити значну аудиторію без необхідності постійної присутності. Відеоуроки дозволяють студентам переглядати матеріал у будь-який зручний для них час, ставити відео на паузу, повертатися до складних моментів і засвоювати інформацію у власному темпі.

Однак для ще більшого залучення та утримання уваги користувачів, освітні платформи дедалі частіше впроваджують додаткові механізми, зокрема гейміфікацію, яка робить навчання більш захопливим та цікавим, перетворюючи його на своєрідну гру, що сприяє активній участі та позитивним емоціям.

За рахунок аспектів внутрішньої, зовнішньої мотивації, автономії, відчуття компетентності та соціальних взаємодій у 2022 році українська компанія CleverStaff провели невеликий експеримент розробивши ексклюзивне оновлення – систему гейміфікації [5]. Її ціль — поліпшити взаємодію користувачів та рекрутерів через зацікавлення у тій же роботі. За кілька місяців після релізу команда розробників проаналізувала, що змінилось, коли рекрутери почали застосовувати гейміфікований софт. Результати опитування зображені на діаграмі [2].

Що вам більше за все сподобалося?



Рисунок 2 – Діаграма опитування серед працівників CleverStaff

Результати підтвердили, мотивація зростає через нагородження чи отримання призового місця серед інших користувачів що забезпечує ефективність у справах.

Водночас зросла можливість масштабування навчання, адже один онлайн-курс можуть проходити тисячі людей одночасно. А також онлайн-освіта здатна допомогти вирішити питання нестачі викладачів та навчальних приміщень.

Разом з тим, онлайн-освіта має і свої недоліки. Відсутність живої комунікації може впливати на якість взаємодії між студентами та викладачами, а також ускладнювати соціалізацію. Освіта у Інтернеті може призводити до відчуття самотності та зниження зацікавленості до навчання.

Для багатьох користувачів проблемою є низька самодисципліна або брак мотивації. Не всі мають рівний доступ до якісного інтернету чи необхідної техніки, крім того, не всі педагоги достатньо підготовлені до ефективного використання цифрових технологій, що може впливати на якість подачі матеріалу.

Сьогодні онлайн-освіта продовжує розвиватися, трансформуючись у гібридні моделі, які поєднують переваги очного та дистанційного навчання. Зростає популярність короткострокових сертифікаційних курсів, мікронавчання, адаптивних платформ з елементами штучного інтелекту та автоматизованими системами підтримки, що надає змогу аналізувати дані про прогрес студента, його сильні та слабкі моменти. Усе це свідчить про те, що онлайн-освіта перестала бути тимчасовим рішенням і набула стратегічного значення в контексті модернізації всієї освітньої системи.

Онлайн-освіта з тимчасового рішення у період пандемії перетворилася на повноцінний і перспективний формат навчання, що продовжує активно розвиватися. Її ключові переваги — гнучкість, доступність і масштабованість — роблять цифрове навчання актуальним інструментом для забезпечення якісної освіти в умовах діджиталізації. Проте для підвищення ефективності та мотивації користувачів важливо не просто переносити контент у цифрове середовище, а адаптувати його до нових форматів взаємодії.

Саме тому особливо доцільною є розробка онлайн-курсів, що поєднують відеоуроки та елементи гейміфікації. Відео дає змогу навчатись у власному темпі, повертатися до складного матеріалу та краще сприймати інформацію завдяки візуалізації. Гейміфікація ж стимулює інтерес і залученість, сприяє формуванню внутрішньої мотивації через систему досягнень, рейтинги, нагороди чи інші інтерактивні механізми. Такий підхід дозволяє зробити освітній процес не лише доступним, а й дійсно ефективним — особливо в умовах, коли учням і студентам часто бракує самодисципліни чи зовнішніх стимулів.

Інвестування в розробку подібних платформ є виправданим і стратегічно важливим кроком, адже воно сприяє як підвищенню якості освіти, так і розширенню можливостей для здобуття знань незалежно від місця проживання, соціального статусу чи фізичних можливостей. Онлайн-навчання з відеоуроками та гейміфікацією — це відповідь на сучасні освітні виклики та реальна перспектива для побудови ефективної, привабливої й доступної освіти майбутнього.

Список використаних джерел

1. Increasing Public Interest in Online Education during the COVID-19 Pandemic in the United States: An Analysis of Google Trends Data. URL: https://www.researchgate.net/publication/387507139_Increasing_Public_Interest_in_Online_Education_during_the_COVID-19_Pandemic_in_the_United_States_An_Analysis_of_Google_Trends_Data (дата звернення: 29.04.2025).
2. The COVID-19 pandemic has changed education forever. URL: https://www.weforum.org/stories/2020/04/coronavirus-education-global-covid19-online-digital-learning/?utm_source=chatgpt.com (дата звернення: 13.04.2025).
3. Офіційний сайт Міністерства освіти і науки України. URL: <https://mon.gov.ua/> (дата звернення: 13.04.2025)
4. Державна служба якості освіти України. URL: <https://sqe.gov.ua/> (дата звернення: 13.04.2025)
5. Як гейміфікація впливає на ефективність роботи. URL: <https://happymonday.ua/gejmifikatsiya-vplyvaye-na-robotu-rekrutera> (дата звернення: 27.04.2025)