

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій

«До захисту допущено»

В.о. завідувача кафедри

_____ Світлана ВАЩЕНКО

_____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня магістр

зі спеціальності 122 «Комп'ютерні науки»,
освітньо-професійної програми «Інформаційні технології проектування»
на тему: «Web-орієнтована система підтримки QA-відділу для ІТ-компаній»

Здобувача групи ІТ.м-23 Грищенко Владислав Віталійович
(шифр групи) (прізвище, ім'я, по батькові)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ (підпис)

Владислав ГРИЩЕНКО

(Ім'я та ПРИЗВИЩЕ здобувача)

Керівник _____ к.т.н., доц. Яна ЧИБІРЯК _____ (підпис)
(посада, науковий ступінь, вчене звання, Ім'я та ПРИЗВИЩЕ)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра інформаційних технологій
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Інформаційні технології проектування»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри ІТ

Світлана ВАЩЕНКО
«_____» _____ 2023 р.

ЗАВДАННЯ

на кваліфікаційну роботу магістра студентіві

Грищенко Владислав Віталійович
(прізвище, ім'я, по батькові)

1 Тема кваліфікаційної роботи _____

затверджена наказом по університету від «08» листопада 2023 р. № 1249-VI

2 Термін здачі студентом кваліфікаційної роботи «___» _____ грудня _____ 2023 р.

3 Вхідні дані до кваліфікаційної роботи _____

4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити) _____

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових слайдів презентації) _____

6. Консультанти випускної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Дата видачі завдання	_____		
Керівник	_____	(підпис)	
Завдання прийняв до виконання	_____	(підпис)	

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
--------------	--	---------------------------------------	-----------------

Магістрант _____ Владислав ГРИЩЕНКО

Керівник роботи _____ к.т.н., доц. Яна ЧИБІРЯК

АНОТАЦІЯ

Тема кваліфікаційної роботи магістра «Web-орієнтована система підтримки QA-відділу для IT-компаній».

Пояснювальна записка складається зі вступу, 4 розділів, висновків, списку використаних джерел із 37 найменувань, 3 додатків. Загальний обсяг роботи – 99 сторінок, у тому числі 36 сторінок основного тексту, 3 сторінки списку використаних джерел, 61 сторінка додатків.

Магістерська робота присвячена розробці та впровадженню web-системи підтримки для відділів забезпечення якості (QA) в інформаційно-технологічних компаніях. Робота включає аналіз потреб та викликів, з якими стикаються відділи контролю якості, визначення ключових завдань та вибір необхідних технологій для успішної реалізації системи підтримки. Зокрема, досліджується використання різних інструментів для оптимізації роботи QA-спеціалістів.

Описано структуру та функціонал системи, що забезпечує автоматизацію рутинних завдань та додаткові можливості для підвищення ефективності внутрішньої взаємодії QA-відділу. Розроблені модулі системи сприяють швидкому виконанню завдань та спрощують взаємодію з технічним обладнанням.

Практичне значення роботи полягає в удосконаленні робочих процесів та оптимізації ресурсів QA-відділів в IT-компаніях. Запропонована система не тільки підвищує якість виконання завдань, але й покращує ефективність роботи команди, забезпечуючи високий рівень взаємодії та автоматизації процесів. Поставлена задача полягає у впорядкуванні повсякденної діяльності QA-спеціалістів в індустрії інформаційних технологій.

Ключові слова: QA-відділ, контроль якості, тестова документація, тестові сценарії, тестувальник, QA-спеціаліст.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	6
ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Огляд останніх досліджень та публікацій	9
1.2 Аналіз інформаційних систем	10
2 ПОСТАВНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ	16
2.1 Мета та задачі дослідження	16
2.2 Методи дослідження	17
3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	19
3.1 Діаграми нотації IDEF0	19
3.2 Use Case діаграми	22
4 ПРАКТИЧНА РОЗРОБКА РІШЕННЯ	26
4.1 Архітектура web-системи	26
4.2 Програмна реалізація web-додатку	27
4.3 Розробка дизайну	27
4.4 Демонстрація роботи web-системи	33
4.5 Тестування web-системи	40
ВИСНОВОК	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
ДОДАТОК А	46
ДОДАТОК Б	59
ДОДАТОК В	91

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

QA – Контроль якості

IT – Інформаційні технології

HLD – Дизайн високого рівня

HTML – Мова гіпертекстової розмітки

CSS – Каскадна таблиця стилів

JS – (JavaScript) Мультипарадигмова мова програмування

PHP – Універсальна скриптова мова загального призначення

jQuery – Бібліотека JavaScript

SQL – Мова структурованих запитів

ВСТУП

Актуальність. У сучасному інформаційному віці, де технологічний прогрес стрімко розвивається, роль QA-інженерів в ІТ компаніях стає ключовою для забезпечення високої якості програмного забезпечення. Швидкі та постійні зміни у вимогах ринку, а також потреба швидкої адаптації до нових технологій ставлять перед розробниками та тестувальниками виклик, що вимагає наявності ефективних інструментів для підтримки їхньої професійної діяльності. В умовах високої швидкості випуску програмних продуктів та вимог до їхньої бездоганної якості, наявність ефективної системи підтримки для робітників QA-відділу стає критично важливою для успішного функціонування індустрії програмного забезпечення.

Робота QA-спеціаліста включає в себе велику кількість завдань, починаючи від створення тестових сценаріїв і закінчуючи відстеженням та аналізом результатів тестування. З урахуванням зростаючої складності програм, потрібно розробити ефективну web-систему, яка полегшить процеси контролю якості та підвищить продуктивність команди QA-інженерів [1].

Тема дослідження. Web-орієнтована система підтримки роботи QA-відділу для ІТ-компаній.

Мета. Метою даної роботи є розробка web-орієнтованої системи підтримки QA-інженерів для оптимізації їхньої роботи в ІТ-компаніях. Для досягнення цієї мети необхідно виконати такі завдання:

- провести аналіз предметної сфери;
- провести аналіз аналогів web-додатків;
- розробити архітектуру-web-системи
- обрати програмні засоби реалізації;
- реалізувати web-додаток та протестувати його роботу.

Об'єкт дослідження. Інформаційна підтримка діяльності QA-інженерів в ІТ-компаніях.

Предмет дослідження. Інформаційні технології процесу підтримки діяльності QA-інженерів для ІТ-компаній.

Практична новизна. Розробка та впровадження web-орієнтованої системи підтримки для QA-інженерів сприятиме підвищенню ефективності тестування, зменшенню часових витрат на створення тестової документації та поліпшить процес контролю якості програмного забезпечення.

Гіпотеза дослідження. Розробка web-системи підтримки для QA-інженерів дозволить зменшити час, витрачений на підготовку тестових сценаріїв, підвищить точність виявлення дефектів програмного продукту та покращить комунікацію в команді розробників. Це, в свою чергу, призведе до підвищення якості продукту та задоволеності замовників.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд останніх досліджень та публікацій

Останніми роками розробка та впровадження web-орієнтованих систем підтримки для відділів контролю якості в ІТ-секторі набули значного поширення. Динамічний розвиток цифрового ландшафту та зростаюча складність програмного забезпечення вимагають інноваційних рішень для оптимізації робочих процесів відділів контролю якості.

У низці досліджень підкреслюється, що ефективність процесів контролю якості тісно пов'язана з наявністю інтегрованих web-систем, що забезпечують швидкий і зручний доступ до інформації, автоматизацію завдань і поліпшення комунікації всередині відділу. Дослідники підкреслюють важливість забезпечення відкритості та гнучкості системи, що дозволяє їй адаптуватися до динаміки галузі та специфічних потреб окремих компаній.

Особлива увага приділяється впливу таких систем на підвищення продуктивності та ефективності роботи відділів контролю якості. Дослідження вказують на потенціал скорочення часу, що витрачається на тестування, підвищення точності виявлення помилок і поліпшення командної співпраці завдяки централізованому управлінню завданнями і ресурсами.

У відповідь на сучасні виклики та швидкозмінні вимоги ринку багато досліджень також підкреслюють важливість інтеграції з сучасними технологіями. Ця інтеграція має на меті забезпечити можливості автоматизації та аналізу даних, тим самим покращуючи стратегії тестування та полегшуючи прийняття управлінських рішень [2].

Таким чином, останні дослідження вказують на необхідність та перспективність використання web-орієнтованих систем підтримки для відділів контролю якості в ІТ-компаніях, що сприятиме покращенню процесів розробки програмного забезпечення з точки зору якості та ефективності.

1.2 Аналіз інформаційних систем

Застосування теми проєкту, в контексті аналізу інформаційних систем підкреслює критичну потребу в складних онлайн-інструментах. У робочих процесах контролю якості ефективно планування, координація та управління відіграють ключову роль, а інтеграція передових інформаційних рішень стає вирішальною. Аналіз має на меті визначити ключові функціональні можливості, спрямовані на забезпечення безперебійної роботи запропонованої web-системи [3].

Зокрема, основна увага приділяється наданню оптимальної підтримки командам контролю якості в їхній повсякденній діяльності. Очікується, що реалізація таких функцій покращить процеси планування, координації завдань та моніторингу робочого процесу відділу контролю якості.

Невід'ємним компонентом аналізу є визначення того, як система інтегрується в повсякденну практику QA-інженерів, надаючи їм ефективні інструменти для спрощення рутинних завдань, полегшення швидкого доступу до інформації та підвищення загальної продуктивності. Мета цього аналізу - визначити оптимальні рішення для покращення робочого процесу відділу контролю якості в контексті сучасного IT-середовища [4].

З метою порівняння були вибрані три інформаційні системи для планування робіт: «Jira Software», «TestRail» та «Xray».

«**Jira Software**» – флагманський продукт Atlassian, слугує комплексним рішенням для управління проєктами та відстеження проблем. Його інтуїтивно зрозумілий інтерфейс у поєднанні з широким набором функцій полегшує планування, відстеження та спільну роботу над проєктами [5]. Jira відома своєю адаптивністю до різних методологій, включаючи Agile та Scrum, що робить її ідеальним вибором для різних команд розробників. Завдяки потужній екосистемі розширень та плагінів, Jira пропонує масштабованість для задоволення потреб як невеликих команд, так і великих підприємств.

Основні функції додатку, які корисні для тестувальників:

- інтегрування тестування в цикли гнучкої розробки;
- налаштування робочих задач відповідно до процесу контролю якості, забезпечуючи ефективне відстеження завдань;
- створення детальних звітів для аналізу прогресу тестування та виявлення вузьких місць розробки;
- зручний контроль створених задач та їх статусів розробки;
- інтеграція з популярними інструментами тестування, покращуючи співпрацю.

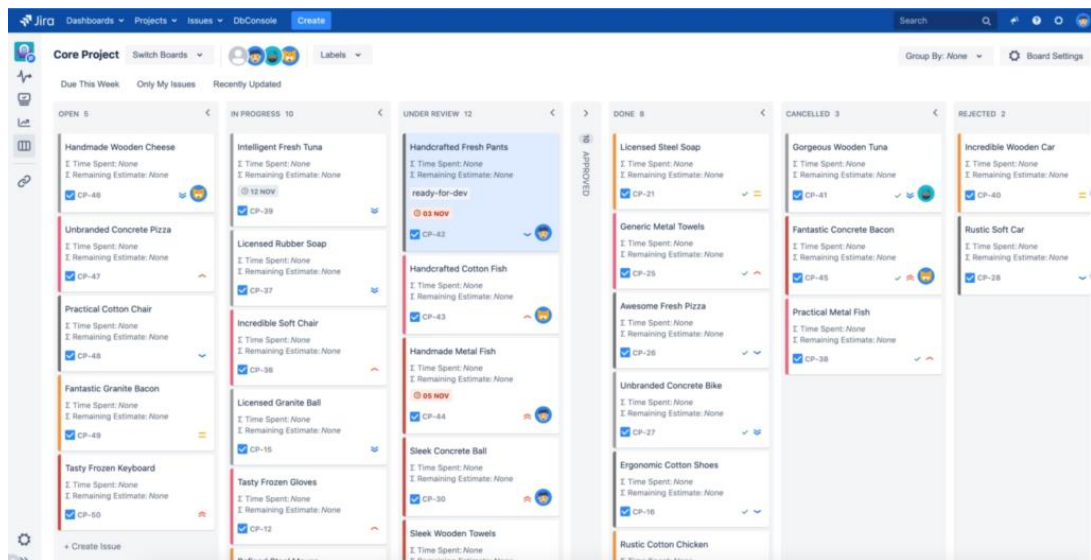


Рисунок 1.1 – Головна сторінка проекту у вигляді Kanban дошки «Jira».

Джерело: [12]

«Jira» пропонує кілька продуктів і варіантів розгортання, які спеціально розроблені для команд розробників програмного забезпечення, QA-відділу, бізнесу, операційних команд тощо. З іншого боку, налаштування конкретних та неочікуваних на початку інструментів для потреб команди завжди забирає багато часу, що вказує на її недосконалу гнучкість. Також, перенасиченість функціоналу – робить інтерфейс «Jira» досить складним у розумінні на початку роботи, особливо для новачків.

У другому випадку, розглянемо інший інструмент для відстеження та організації тестування програмного забезпечення – «TestRail». Продукт, розроблений Gurock Software GmbH, виділяється як доступний та ефективний інструмент управління тестуванням [6]. Завдяки простоті, він дозволяє користувачам легко організовувати тестові кейси та створювати змістовні звіти. Зручний інтерфейс «TestRail» дозволяє командам легко освоїти його, хоча він в першу чергу задовольняє потреби в управлінні тестуванням, а не в комплексному управлінні проєктами.

Основні можливості, які роблять дану web-систему досить корисною:

- інтуїтивна організація тестових кейсів для ефективного управління;
- формування вичерпних звіти допомагають виявити тенденції та потенційні проблеми;
- дозволяє інтегруватися з інструментами автоматизації для оптимізації процесів тестування;
- налаштування поля для збору конкретних даних, пов'язаних з контролем якості.

На рисунку 1.2 відображена сторінка проєкту реалізована в «TestRail».

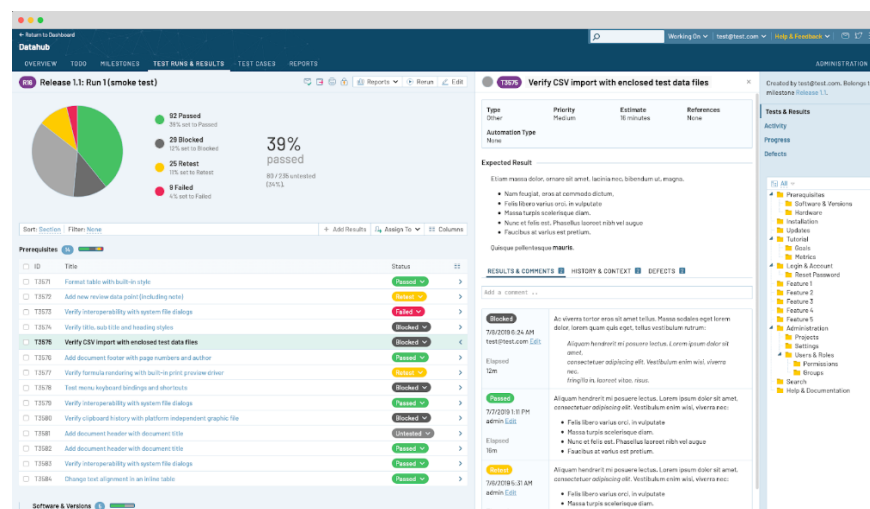


Рисунок 1.2 – Сторінка проєкту в «TestRail». Джерело: [14]

Хоча й web-додаток допомагає вам керувати важливими даними та структурами програмного забезпечення, такими як етапи проєкту, а також спрощує інтеграцію з інструментами для відстеження помилок, серед недоліків часто згадують обмежені можливості управління проєктами, потенційно високі витрати для невеликих компаній та відсутність вбудованих інструментів автоматизації.

«Xray» - це інструмент, який дозволяє оптимізувати процес управління та автоматизації тестування. Він допомагає організувати та керувати тестами, автоматизувати процес виконання тестів та створювати тестові звіти [7].

«Xray» покращує видимість результатів тестування, добре інтегрується з різними інструментами та відповідає методологіям Agile. Xpand IT, розробник «Xray», - португальська компанія, відома своїми передовими рішеннями в галузі IT-консалтингу та розробки програмного забезпечення.

Серед основних плюсів у використанні даного програмного продукту є:

- оптимізація виконання тестів, забезпечуючи узгодженість із завданнями розробки;
- забезпечення відстеження виконання тестів, допомагаючи в дотриманні вимог;
- безперешкодно підключається до різних інструментів тестування для створення комплексного середовища контролю якості;
- сприяє гнучкому тестуванню, дозволяючи QA адаптуватися до мінливих вимог проєкту.

Головна панель приладів з розширеними звітами та гаджетами для прийняття розумних рішень «Xray» відображені на рисунку 1.3.

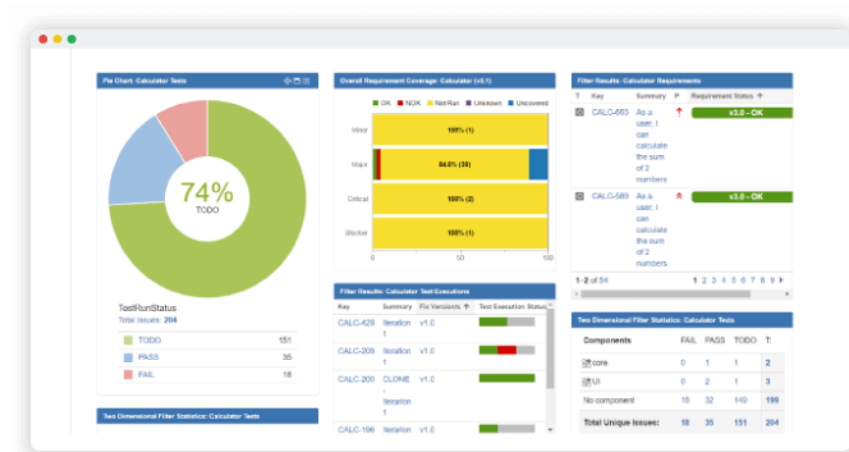


Рисунок 1.3 – Головна панель приладів «Xray». Джерело: [17]

Хоча можливості вдосконалення процесу забезпечення якості web-системи вражають, залежність від «Jira», обмежені можливості автоматизації та потенційні витрати для великих баз користувачів іноді роблять цей продукт менш придатним для використання та співпраці.

Було створено порівняльну таблицю вище наведених програмних продуктів (таблиця 1.1).

Таблиця 1.1 – Порівняльна таблиця аналогів web-додатків з «Saber»

Характеристика	Jira Software	TestRail	Xray	Saber
Створення карточок проєкту	+	+	-	+
Формування QA-команди	-	+	+	+
Управління розподілом завдань	+	-	+	+
Проміжні результати тестування	+	+	-	+

Продовження таблиці 1.1

Створення тестової документації	-	-	-	+
Створення тестових сценаріїв	-	+	+	+
Технічна підтримка	+	+	-	+

Аналізуючи інформацію з таблиці порівняння, можна визначити ключові функціональності для ідеальної web-системи для роботи QA-відділу. З цього випливає, що сучасний web-додаток для контролю якості програмного продукту має забезпечувати тестувальників можливостями формування тестових сценаріїв, управління розподілом завдань, зміни статусів тестових сценаріїв, а також отримання змістовного й актуального звіту по результатам тестування.

Підсумовуючи можна визначити, що розроблений web-додаток повинен відповідати наступним функціональним вимогам:

- створення та редагування карточок проєктів;
- створення, редагування та видалення тестових сценаріїв;
- можливість розподіляти роботу над проєктами між тестувальниками;
- формування звіту по статусу проведення тестування та актуального стану проєкту.

2 ПОСТАВНОВКА ЗАДАЧІ ТА МЕТОДИ ДОСЛІДЖЕННЯ

2.1 Мета та задачі дослідження

Метою даної роботи є розробка та впровадження web-орієнтованої системи підтримки QA-інженерів для автоматизації їхньої роботи в IT-компаніях. Використання сучасної web-орієнтованої системи підтримки QA-інженерів в IT-компаніях може значно підвищити ефективність і якість процесів розробки та тестування [8]. Така система не лише надає інтуїтивно зрозумілий та сучасний інтерфейс для спрощеної взаємодії з користувачем, але й дозволяє ефективно управляти проєктами та планувати завдання, скорочуючи час, необхідний для цих процесів.

Завдяки вбудованим можливостям автоматизації та інтеграції з іншими інструментами розробки, система сприяє підвищенню точності та швидкості тестування, а також зменшенню ручної праці. Покращена звітність та аналітика дають командам QA чітку картину стану проєкту, допомагаючи приймати обґрунтовані рішення. Загалом, використання такої системи дозволяє автоматизувати робочі процеси, скоротити витрати та підвищити ефективність роботи QA-інженерів - критично важливі фактори для успішної інтеграції та просування продуктів в інформаційно-технологічній сфері.

Розроблений web-додаток повинен відповідати наступним вимогам:

- створення та редагування карточок проєктів;
- створення, редагування та видалення тестових сценаріїв;
- можливість розподіляти роботу над проєктами між тестувальниками;
- формування звіту по статусу проведення тестування та актуального

стану проєкту.

Розроблена web-система також повинна мати зручний, зрозумілий й адаптивний інтерфейс з оптимальним використанням анімацій та кольорових гам.

Для досягнення мети цього проєкту необхідно виконати наступні завдання:

- провести аналіз існуючих аналогів web-системи;
- пояснити вибір технологій реалізації;
- розробити архітектуру web-системи та виконати моделювання її роботи;
- розробити структуру бази даних web-системи;
- реалізувати web-додаток та протестувати його роботу.

Під час визначення мети проєкту було проведено планування проєктної діяльності, деталі якого наведені в Додатку А.

2.2 Методи дослідження

Вибір методології дослідження є важливим кроком у забезпеченні точності та об'єктивності отриманих результатів. Під час переддипломної практики було застосовано комплексну методологію, що поєднує елементи аналізу бізнес-процесів, проєктування користувацького інтерфейсу та тестування програмного забезпечення. Такий методологічний підхід сприяв поглибленому дослідженню функціональних вимог та взаємодії різних компонентів всередині системи [9].

Для проведення ефективного аналізу та вимірювань використовувалися спеціалізовані інструменти для збору, аналізу та візуалізації даних. При дослідженні системи підтримки контролю якості були використані інструменти для відстеження помилок, визначення покриття коду, аналізу використання ресурсів та моніторингу продуктивності [10].

Однією з основних цілей дослідження було визначення ефективності web-орієнтованої системи підтримки контролю якості. В процесі оцінки були визначені та враховані такі критерії, як швидкість реагування системи, стабільність, зручність для користувача та відповідність функціональним вимогам.

Після завершення дослідження було проведено аналіз отриманих результатів. Було досліджено сильні та слабкі сторони системи, визначено можливості для подальшого вдосконалення та надано рекомендації щодо оптимізації функціоналу. Проведений аналіз дозволив сформулювати висновки та рекомендації, корисні для подальшого розвитку та вдосконалення системи підтримки контролю якості в ІТ-компаніях [11].

Після проведення аналізу актуальності завдання та визначення типу програмного забезпечення, наступним кроком є прийняття рішення про технологію реалізації web-системи.

Для програмної реалізації інформаційної технології були обрані наступні технології: HTML – для створення структури web-сторінок [12], CSS – для стилізації та дизайну, забезпечення візуально привабливого користувацького інтерфейсу [13], JavaScript (JS) – для створення інтерактивних функцій на web-сторінках [14], бібліотека скриптів – jQuery [15], PHP – для написання сценаріїв на стороні сервера, що забезпечує динамічну генерацію контенту [16], MySQL – як система управління реляційними базами даних для ефективної організації та зберігання даних [17], phpMyAdmin – для адміністрування баз даних, що спрощує управління базами даних MySQL [18], Laragon – як фреймворк для web-додатків на PHP [19], що забезпечує надійне та ефективне середовище розробки, Visual Studio Code (VS Code) – як інтегроване середовище розробки (IDE) для редагування коду та управління проєктами [20].

Цей вибір технологій має на меті створити комплексне та ефективне програмне рішення для передбачуваної інформаційної технології, поєднуючи сильні сторони у створенні функціонального та зручного для користувача web-додатку.

3 ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Після детального вивчення предметної області, з'ясування актуальності та цілей розробки, а також формулювання плану дій для реалізації проєкту, наступним кроком стала розробка зазначеної web-системи. У цьому процесі були розроблені діаграми в нотації IDEF0, щоб всебічно проілюструвати послідовність процесів, пов'язаних з автоматизацією розподілу завдань між членами команди.

3.1 Діаграми нотації IDEF0

Для розробки функціональної моделі web-системи використовуються різні методи, включаючи структурний та об'єктно-орієнтований аналіз. Одним з найбільш ефективних методів є IDEF0, який дозволяє візуалізувати робочі процеси та системи та чітко показати взаємодію функцій між собою та із зовнішнім середовищем [21].

Ключові елементи діаграми IDEF0 включають в себе:

- механізми, які представляють ресурси, необхідні для виконання функцій;
- вхідні дані, які використовуються для виконання функцій;
- вихідні дані, які є результатом виконання функцій;
- управління, яке визначає умови виконання функцій.

Діаграми IDEF0 часто використовуються для аналізу існуючих систем та процесів, а також для розробки нових систем та процесів. Вони допомагають визначити ключові функції та взаємозв'язки між ними[22].

Процес «Підтримка роботи QA-відділу» має наступні дані:

- вхідні дані: інформація про проєкт, інформація про тестові сценарії, інформація про назначені проєкти;
- вихідні дані: тестова документація проєкту, статус тестування проєкту;

- управління: розподіл задач, вимоги ТЗ проєкту, техніки тест-дизайну;
- механізми: web-додаток, технічне забезпечення, база даних.

На рисунку 3.1 зображено контекстну діаграму web-системи IDEF0, що ілюструє підтримку роботи QA-відділу в ІТ-компаніях.

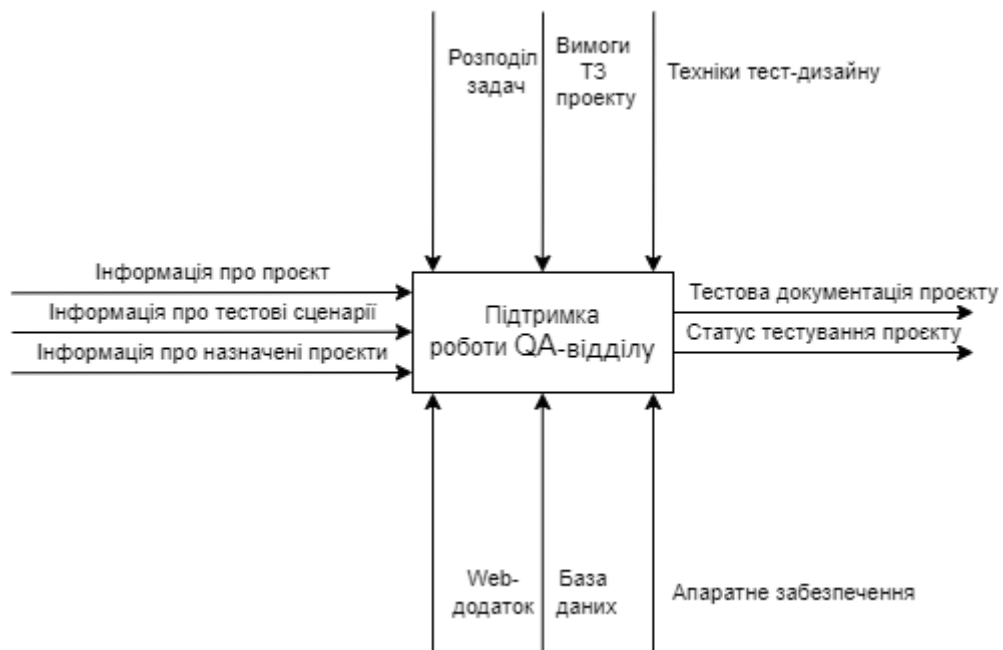


Рисунок 3.1 – Контекстна діаграма IDEF0. Джерело: побудовано автором

Для деталізації процесу підтримки роботи QA-спеціалістів було проведено декомпозицію першого рівня (див. рис. 3.2).

Дані для діаграми наступні:

- вхідні дані: інформація про проєкти, інформація про тестові сценарії, інформація про назначені проєкти;
- вихідні дані: тестова документація проєкту, статус тестування проєкту;
- управління: розподіл задач, вимоги ТЗ проєкту, тест-дизайн;
- механізми: web-додаток, технічне забезпечення, база даних.

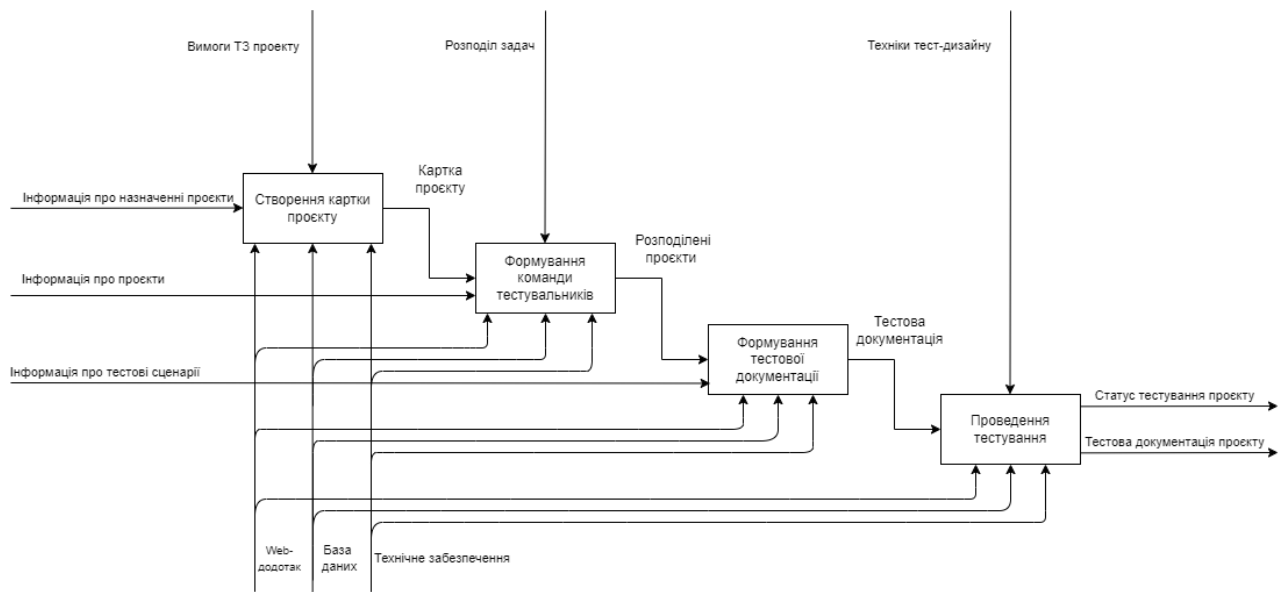


Рисунок 3.2 – Діаграма декомпозиції першого рівня. Джерело: побудовано автором

Також було зроблено декомпозицію процесу призначення QA-співробітників на проекти (див. рис. 3.3).

Дані для діаграми наступні:

- вхідні дані: інформація про проекти, картка проекту.
- вихідні дані: розподілені проекти.
- управління: розподіл задач;
- механізми: web-додаток, технічне забезпечення, база даних.

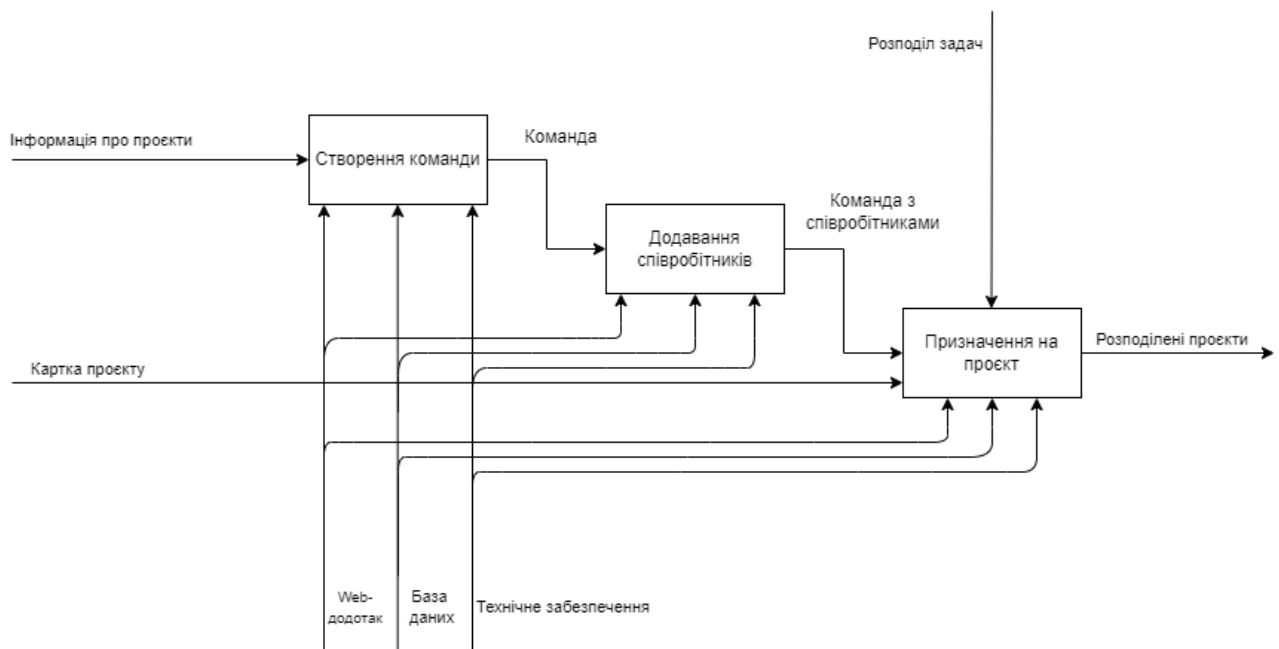


Рисунок 3.3 – Діаграма декомпозиції процесу «Призначення тестувальників на проєкти». Джерело: побудовано автором

3.2 Use Case діаграми

Після завершення етапу моделювання дуже важливо розробити діаграму варіантів використання. Діаграма варіантів використання є життєво важливим інструментом у сфері розробки програмного забезпечення, особливо в контексті об'єктно-орієнтованого аналізу та проєктування. Вона допомагає візуалізувати функціональні вимоги до системи та взаємозв'язки між різними користувачами (акторами) і системою [23]. Ключовими компонентами діаграми варіантів використання є актори, взаємозв'язки, межі системи та самі варіанти використання. Діаграми варіантів використання покращують розуміння функціональності системи, допомагають у визначенні вимог користувачів і полегшують комунікацію між розробниками та зацікавленими сторонами.

Для web-орієнтованої системи підтримки роботи QA-відділу, визначені наступні варіанти використання:

- перегляд створення, архівування картки проєкту;
- перегляд статусу, створення, редагування, видалення документації;

- створення команди;
- додавання та призначення співробітників.

Діаграма варіантів використання для інформаційної технології, що підтримує операційне управління проектом для менеджера проекту, зображена на рисунку 3.3

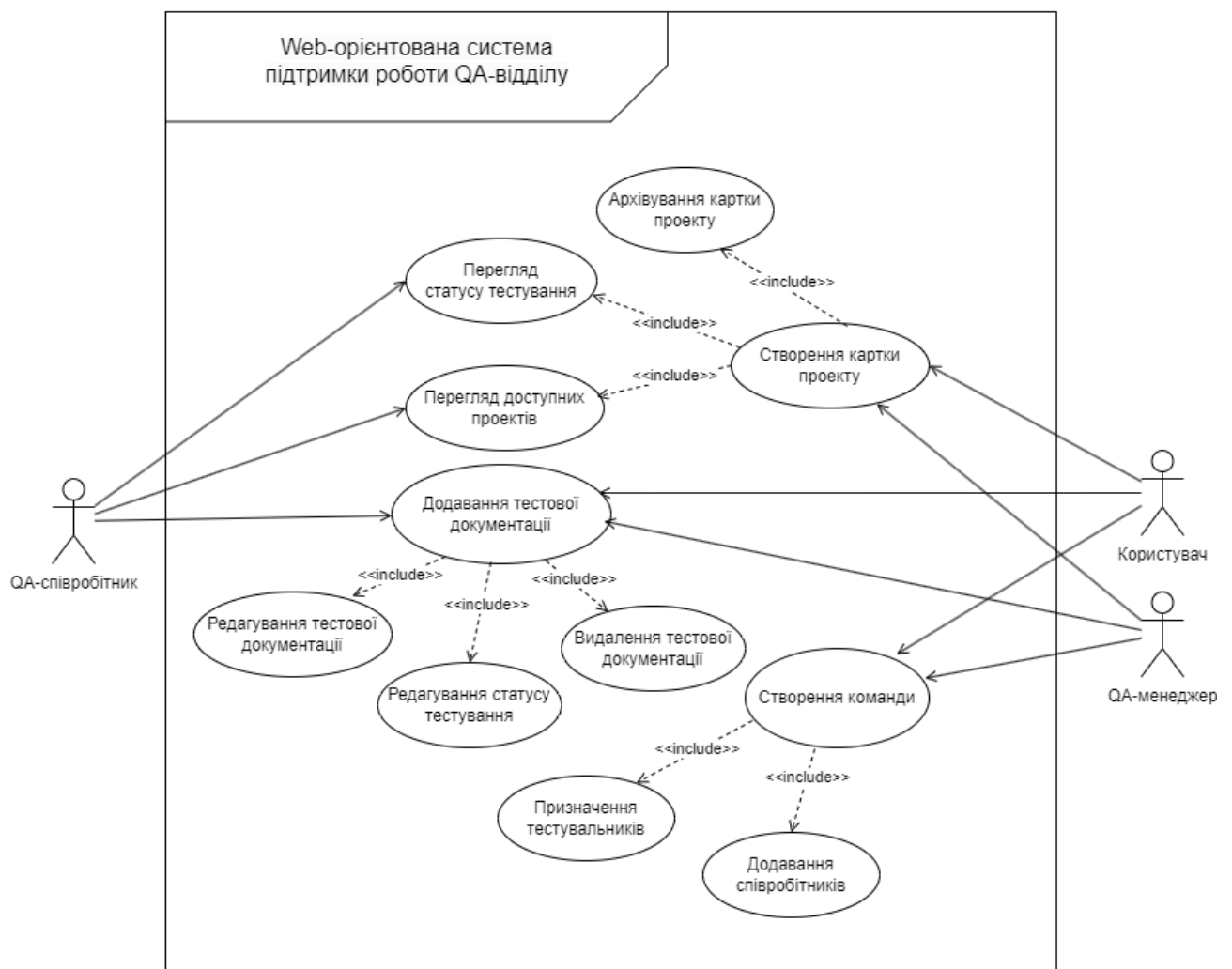


Рисунок 3.3 – Use Case діаграма. Джерело: побудовано автором

3.3 Проектування моделі бази даних

Моделювання баз даних - важливий етап у створенні системи зберігання та обробки інформації. Цей процес передбачає визначення та організацію даних, а також встановлення взаємозв'язків між ними. Ретельне виконання цього етапу

розробки забезпечує високу ефективність і широку функціональність інформаційних технологій. Усі логічні моделі баз даних складаються з трьох основних компонентів: сутностей, зв'язків та атрибутів.

Сутність - це абстрактний об'єкт, що представляє групу речей або понять, пов'язаних між собою. Зв'язок між сутностями встановлюється на основі того, що дані однієї сутності залежать від даних іншої. Атрибути представляють властивості або характеристики, які доповнюють опис сутності.

Після створення каркасу web-системи необхідно заповнити його даними. Для цього використовується система управління базами даних (СУБД), яка дозволяє легко додавати та редагувати інформацію на веб-ресурсах. База даних - це зручне сховище для однотипної інформації.

База даних «saber» була розроблена за допомогою СУБД «phpMyAdmin». На рисунку 3.4 зображено реалізацію моделі бази даних web-орієнтованої системи підтримки QA-відділу в ІТ-компаніях.

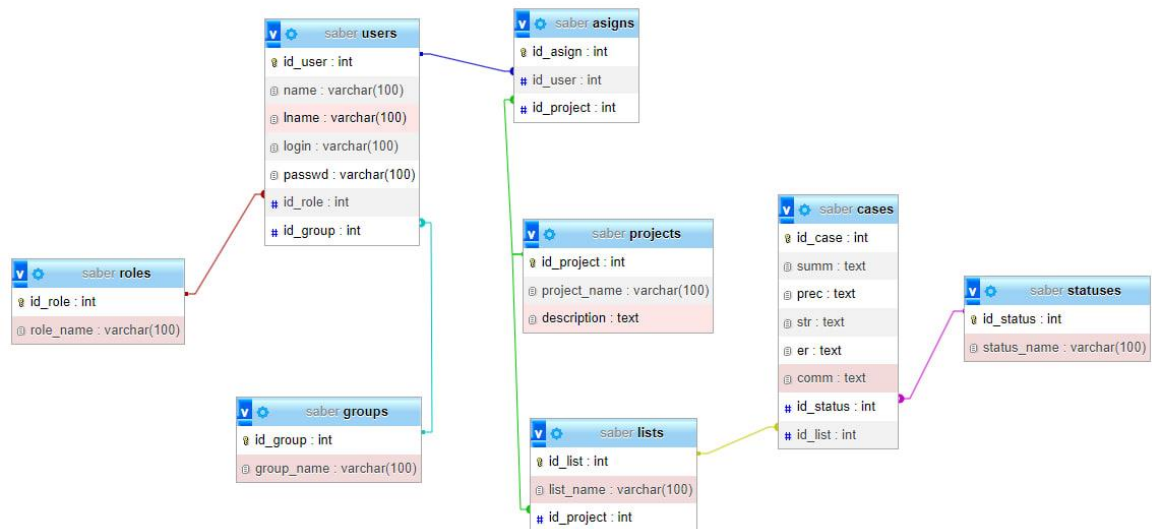


Рисунок 3.4 – Реалізована модель бази даних. Джерело: побудовано автором

Під час створення функціональних можливостей QA-менеджера web-додатку було створено та використано наступні сутності:

- Користувачі (Users) – містить інформацію про користувачів;

- Ролі або посади (Roles) – містить інформацію про ролі або посади користувачів;
- Групи користувачів (Groups) – містить інформацію про групи користувачів;
- Призначення на проєкти (Asigns) – містить інформацію про призначення на проєкти;
- Проєкти (Projects) – містить інформацію про проєкти.

Таблиця Списків перевірок (Lists) має наступні атрибути:

- id_list (integer) – первинний ключ таблиці Lists;
- list_name (character varying (100)) – назва списку;
- id_project (integer) – номер проєкту списку;

Таблиця з Тестовими сценаріями (Cases) має наступні атрибути:

- id_case (integer) – первинний ключ таблиці Cases;
- summ (text) – назва тестового сценарію;
- prec (text) – передумови;
- str (text) – кроки відтворення;
- er (text) – очікуваний результат;
- comm (text) – коментарі;
- id_status (integer) – статуси тестових сценаріїв;
- id_list (integer) – в якому списку.

Таблиця з Статусами (Statuses):

- id_status (integer);
- status_name (character varying (100)).

4 ПРАКТИЧНА РОЗРОБКА РІШЕННЯ

4.1 Архітектура web-системи

Кожна розробка web-систем повинна починатися з побудови архітектури компонентів, які формуватимуть функціональність і дизайн продукту. Високорівневе проектування (HLD) - це концептуальний план роботи над програмою, який може бути адаптований до конкретних вимог та обставин [11].

Діаграма HLD для web-системи, спрямованого на підтримку QA-відділу в ІТ-компаніях, складається з наступних компонентів:

- контролер, що взаємодіє з моделлю для отримання та обробки даних.
- модель, яка складається з двох складових: перша взаємодіє з об'єктно-реляційним відображенням у web-системі, а друга – з сервіс-контейнером.
- база даних, яка забезпечує web-систему інформацією та контентом.
- відображення, яке візуалізує представлення даних.

Діаграма високого рівня представлена на рисунку 4.1.

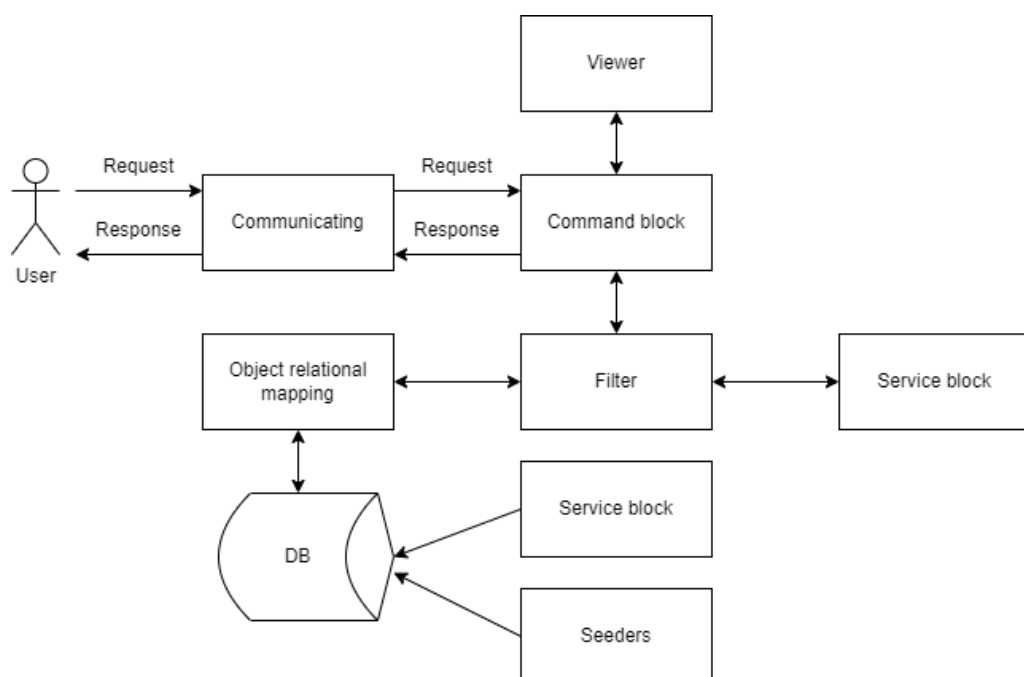


Рисунок 4.1 – Діаграма високого рівня. Джерело: побудовано автором

4.2 Розробка дизайну web-додатку

Під час розробки веб-додатку було прийнято рішення створити макет продукту. Це необхідно для того, щоб чітко визначити положення, розмір та колір елементів у рішенні. Дотримуючись вимог, на рисунку 4.2 зображено макет головної робочої сторінки розробленої web-системи.

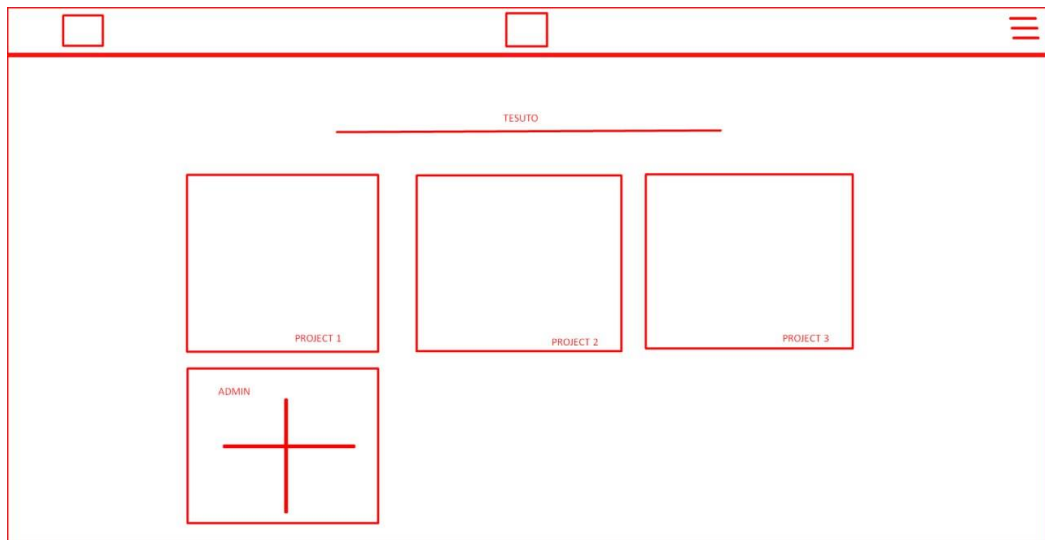


Рисунок 4.2 - Макет головної сторінки веб-додатку. Джерело: побудовано автором

Розробка дизайну передбачає створення логотипу та зовнішнього вигляду веб-сторінок, а також підбір контенту. На основі назви продукту було створено два варіанти логотипу веб-додатку за допомогою Adobe Photoshop [12]. Вони проілюстровані на рисунку 4.3.



Рисунок 4.3 – Логотипи веб-додатку. Джерело: побудовано автором

Для приємного вигляду та тематичного забарвлення було обрано 4 основні кольори: чорний, червоний, синій та блакитний. Також, підбрано шрифт тексту Century Gothic для зручного сприймати інформації розміщеної на ресурсі.

Задля вирішення проблем перехресного перегляду та їх зв'язків між собою використовувалася бібліотека JQuery [14] (рис.4.4).

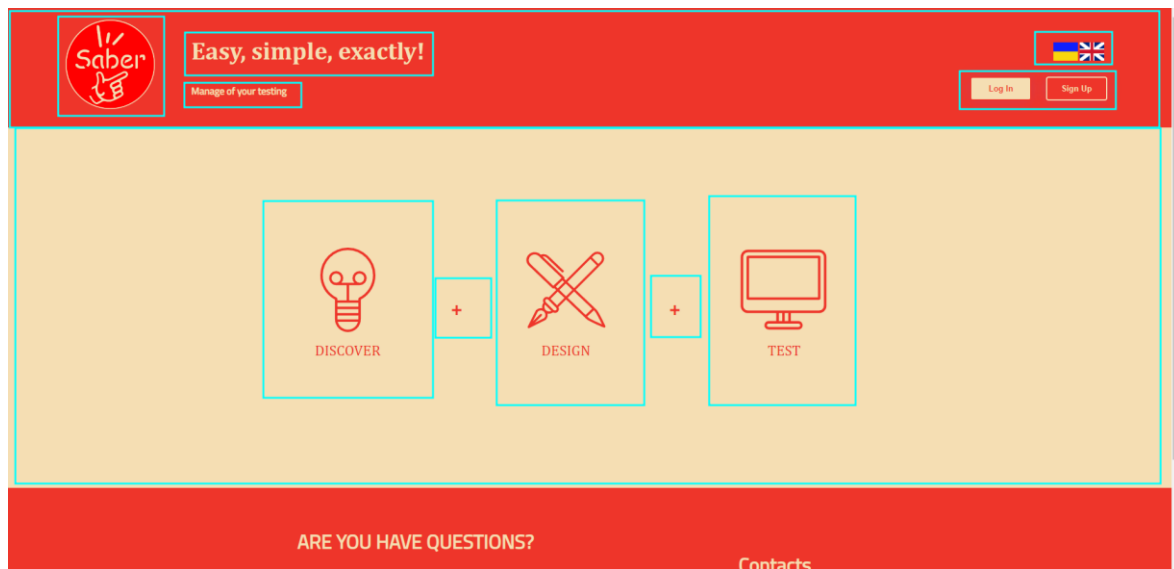


Рисунок 4.4 – Розташування та зв'язок між блоками JQuery. Джерело: побудовано автором

4.3 Програмна реалізація веб-додатку

Для початку роботи над веб-додатком необхідно визначитись із середовищем розробки та підтримки. В якості локального веб-сервера було обрано Laragon [15]. Цей пакет включає в себе ретельно підібраний набір серверного програмного забезпечення та зручну утиліту для управління та налаштування всіх доступних компонентів. Для запуску цього рішення достатньо створити папку для веб-додатку за адресою D:\Laragon\www\Saber та натиснути кнопку "Start All" (див. рис. 4.5).

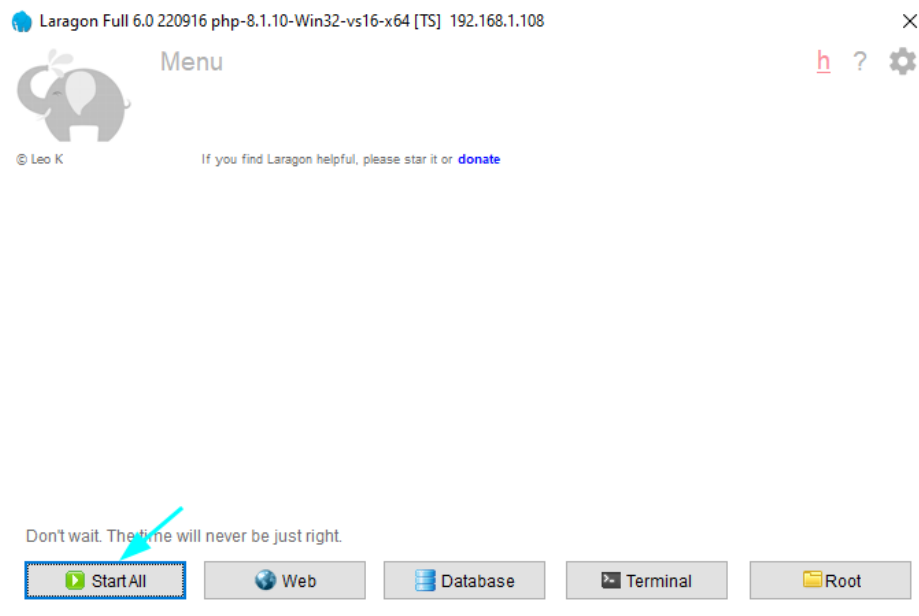


Рисунок 4.5 – Інтерфейс Laragon. Джерело: [21]

Система управління базами даних (СУБД) - ще один важливий компонент у створенні програмного продукту з великим обсягом інформації. Вона слугує основним інструментом для роботи з динамічними веб-додатками. Тому в якості СУБД для програмного продукту було обрано phpMyAdmin. Судячи з її переваг та недоліків, вона є оптимальним рішенням для поставленої задачі.

За допомогою цієї СУБД було створено базу даних під назвою "saber" та розроблено 7 таблиць з даними відповідно до вимог проекту (рис. 4.6).

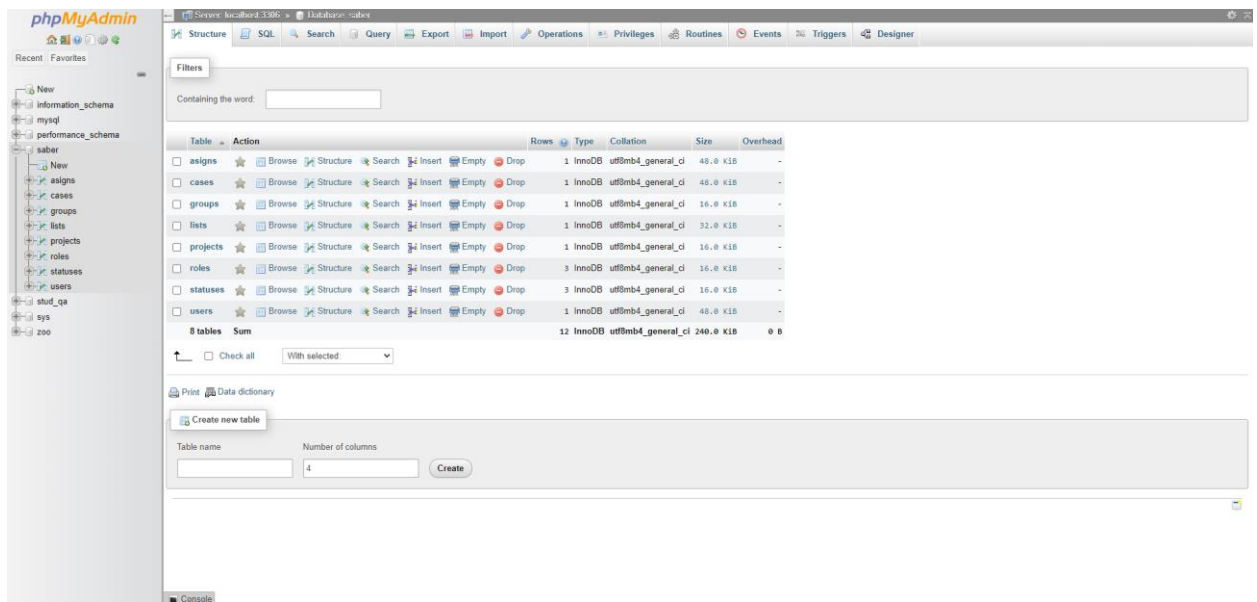


Рисунок 4.6 – Інтерфейс phpMyAdmin та БД «saber». Джерело: [30]

Для внесення інформації про нових користувачів до бази даних було вирішено створити файл `reg.php`, який містить набори перевірок на достовірність та унікальність даних, що додаються. Після успішної реєстрації він надсилає запит на збереження інформації в базі даних (рис. 4.7).

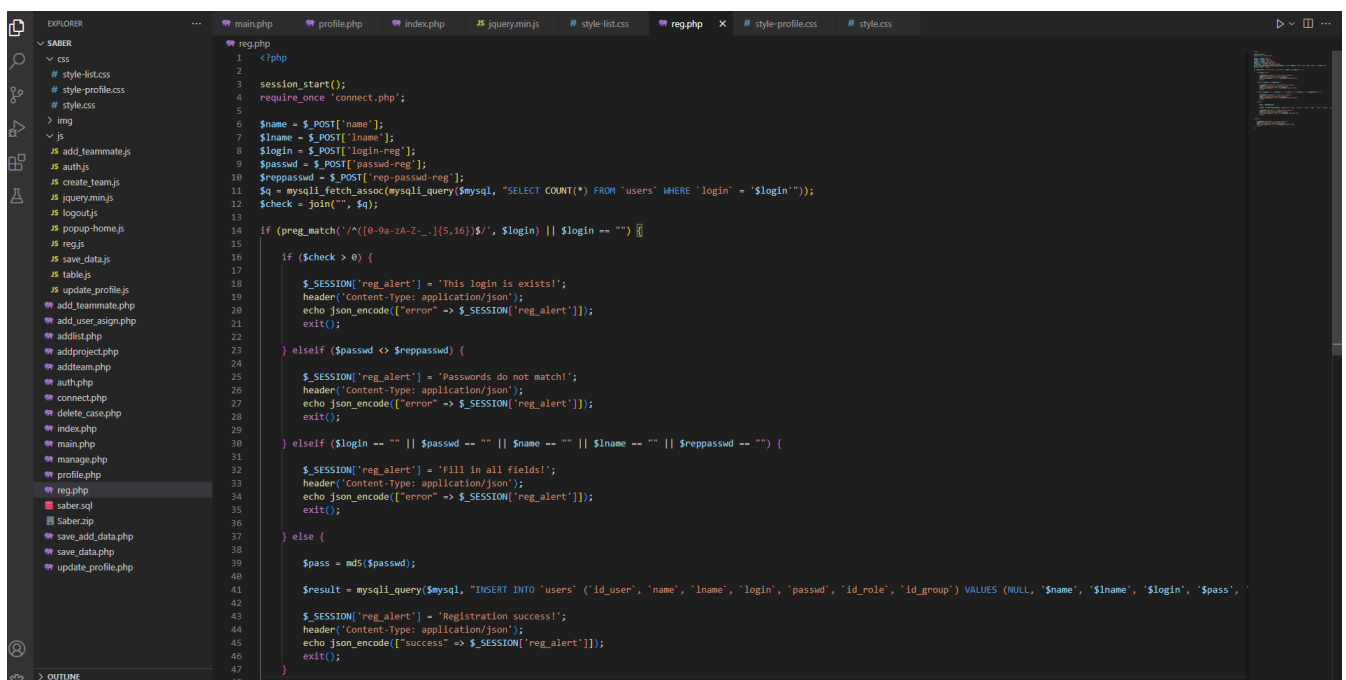
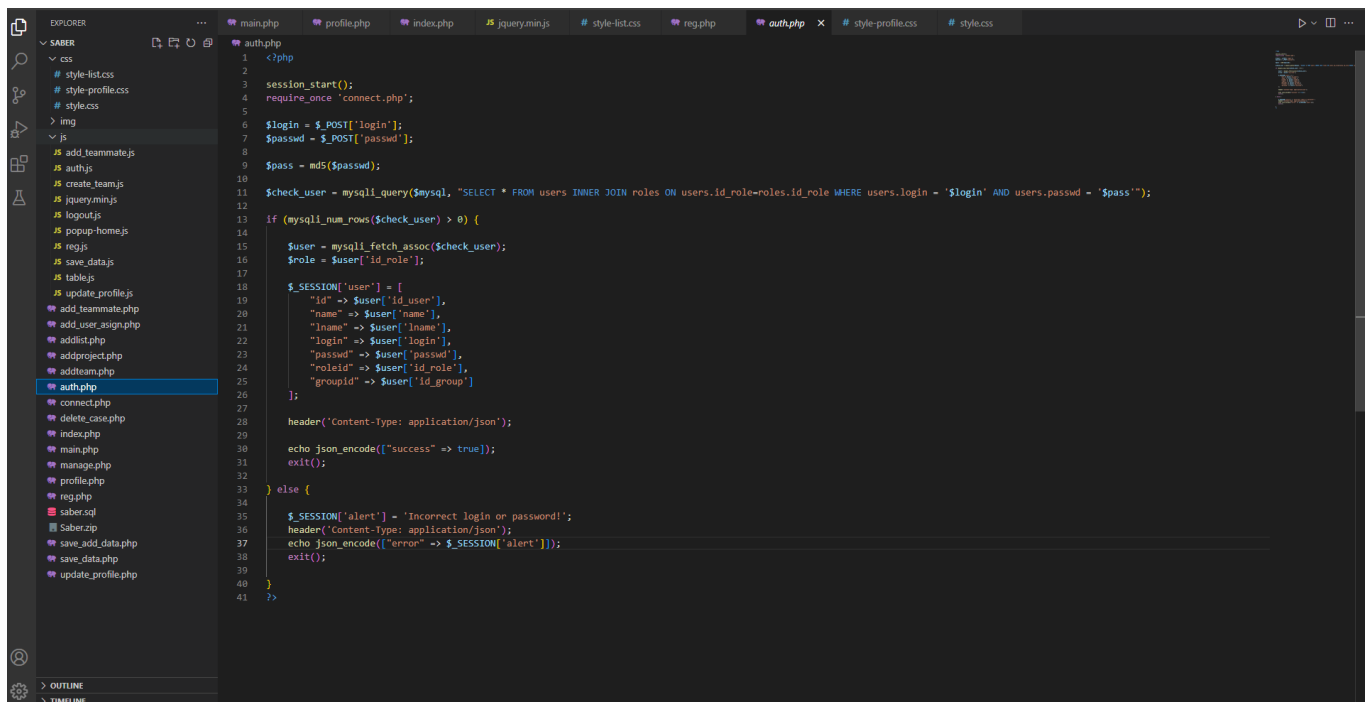


Рисунок 4.7 – Лістинг коду файлу `reg.php`. Джерело: побудовано автором

Після додавання нового користувача та збереження його інформації, він повинен пройти аутентифікацію, тобто перевірку його даних на відповідність інформації, що зберігається в базі даних. Якщо веб-додаток знайде збіг, відвідувач увійде до свого профілю і зможе продовжити користуватися його функціями. За цей процес відповідає файл автентифікації - auth.php (рис. 4.8).



```

1 <?php
2
3 session_start();
4 require_once 'connect.php';
5
6 $login = $_POST['login'];
7 $passwd = $_POST['passwd'];
8
9 $pass = md5($passwd);
10
11 $check_user = mysql_query($mysql, "SELECT * FROM users INNER JOIN roles ON users.id_role=roles.id_role WHERE users.login = '$login' AND users.passwd = '$pass'");
12
13 if (mysql_num_rows($check_user) > 0) {
14
15     $user = mysql_fetch_assoc($check_user);
16     $role = $user['id_role'];
17
18     $_SESSION['user'] = [
19         "id" => $user['id_user'],
20         "name" => $user['name'],
21         "lname" => $user['lname'],
22         "login" => $user['login'],
23         "passwd" => $user['passwd'],
24         "roleid" => $user['id_role'],
25         "groupid" => $user['id_group']
26     ];
27
28     header('Content-type: application/json');
29
30     echo json_encode(["success" => true]);
31     exit();
32 }
33
34 else {
35     $_SESSION['alert'] = 'Incorrect login or password!';
36     header('Content-type: application/json');
37     echo json_encode(["error" => $_SESSION['alert']]);
38     exit();
39 }
40
41 >>

```

Рисунок 4.8 – Лістинг коду файлу auth.php. Джерело: побудовано автором

По всій web-системі для приємного враження від користування та використання були реалізовані поп-ап елементи, які при взаємодії з різними елементами, виникають на екрані або змінюють своє місце положення. Приклад таких функцій переміщення наведено у файлі popup-home.js (рис. 4.9).

```

22 const Title3 = document.getElementById('title3');
23 const Title4 = document.getElementById('title4');
24 const Title5 = document.getElementById('title5');
25 const Title6 = document.getElementById('title6');
26 const Title7 = document.getElementById('title7');
27
28 openPopUpLog.addEventListener('click', function (e) {
29   PopUpLog.classList.add('active');
30 })
31
32 openPopUpReg.addEventListener('click', function (e) {
33   PopUpReg.classList.add('active');
34 })
35
36 closePopUpLog.addEventListener('click', function (e) {
37   PopUpLog.classList.remove('active');
38 })
39
40 closePopUpReg.addEventListener('click', function (e) {
41   PopUpReg.classList.remove('active');
42 })
43
44 Username.addEventListener('focus', function (e) {
45   Title1.classList.add('move1');
46 })
47
48 Password.addEventListener('focus', function (e) {
49   Title2.classList.add('move2');
50 })
51
52 Title1.addEventListener('click', function (e) {
53   Title1.classList.add('move1');
54   Username.focus();
55 })
56
57 Title2.addEventListener('click', function (e) {
58   Title2.classList.add('move2');
59   Password.focus();
60 })
61
62 Name.addEventListener('focus', function (e) {
63   Title3.classList.add('move3');
64 })
65
66 LastName.addEventListener('focus', function (e) {
67   Title4.classList.add('move4');
68 })
69
70 Login.addEventListener('focus', function (e) {

```

Рисунок 4.9 – Лістинг коду файлу popup-home.js. Джерело: побудовано автором

Для реалізації механізму створення адаптивної та інтерактивної таблиці списку тестових сценаріїв був написаний скрипт table.js, який замість комірок при подвійному кліку вставляв елемент textarea, для редагування, додавання або видалення даних. Лістинг коду table.js зображено на рисунку 4.10.

```

1 document.addEventListener('input', function (e) {
2   if (e.target.tagName.toLowerCase() === 'textarea') {
3     autoExpand(e.target);
4   }
5 });
6
7 document.addEventListener('dblclick', function (e) {
8   if (e.target.classList.contains('editable')) {
9
10    var textarea = document.createElement('textarea');
11    textarea.value = e.target.innerHTML;
12
13    e.target.innerHTML = '';
14    e.target.appendChild(textarea);
15
16    textarea.focus();
17
18    textarea.addEventListener('blur', function () {
19
20      e.target.innerHTML = formatCellContent(textarea.value);
21
22      activateLinksInCell(e.target);
23    });
24  }
25 });
26
27 var cells = document.querySelectorAll('.list-cases td.editable');
28 cells.forEach(function (cell) {
29   activateLinksInCell(cell);
30 });
31
32 function formatCellContent(content) {
33   content = content.replace(/</n>/g, '
');
34
35   content = content.replace(/(https?:\/\/[^\s]+)/g, function (match) {
36     return '<a href="' + match + '" target="_blank"> + match + '</a>';
37   });
38
39   return content;
40 }
41
42 function autoExpand(textarea) {
43   textarea.style.height = 'auto';
44   textarea.style.height = (textarea.scrollHeight) + 'px';
45 }
46
47
48
49

```

Рисунок 4.10 – Лістинг коду файлу table.js. Джерело: побудовано автором

Після редагування такої таблиці, потрібно було зберегти всі зміни та оновити їх у базі даних. При цьому відслідковуючи зачеплені поля й ідентифікувати кожний з них. Це реалізовано у файлі `save_data.js`, лістинг коду якого зображено на рисунку 4.11.

```

1 save_data.js > document.addEventListener("DOMContentLoaded", callback) > collectData
2 document.addEventListener("DOMContentLoaded", function () {
3   const saveButton = document.getElementById('save');
4   saveButton.addEventListener('click', function () {
5     saveData();
6   });
7
8   function saveData() {
9     const data = collectData();
10    fetch('save_data.php', {
11      method: 'POST',
12      headers: {
13        'Content-Type': 'application/json',
14      },
15      body: JSON.stringify(data),
16    })
17    .then(response => response.json())
18    .then(responseData => {
19      if (responseData.success) {
20        location.reload();
21      } else {
22        console.error('Error saving data:', responseData.error);
23      }
24    })
25    .catch(error => {
26      console.error('Error saving data:', error);
27    });
28  }
29
30  function collectData() {
31    const tableRows = document.querySelectorAll('#list-cases tr');
32    const data = [];
33
34    tableRows.forEach(row => {
35      const caseId = row.getAttribute('data-case-id');
36      const summary = row.querySelector('summary').innerText;
37      const prec = row.querySelector('#prec').innerText;
38      const str = row.querySelector('#str').innerText;
39      const com = row.querySelector('#com').innerText;
40      const statusDropdown = row.querySelector('select.statusDropdown');
41      const statusId = statusDropdown.options[statusDropdown.selectedIndex].value;
42
43      data.push({
44        caseId: caseId,

```

Рисунок 4.11 – Лістинг коду файлу `save_data.js`. Джерело: побудовано автором

4.4 Демонстрація роботи web-системи

При першому переході за посиланням на web-систему, відображається домашня сторінка екрану на якій представлено сайт та наведено коротку інструкцію використання. Також в хедері присутні кнопки зміни локалізації, входу в акаунт та реєстрації акаунту (рис. 4.12).

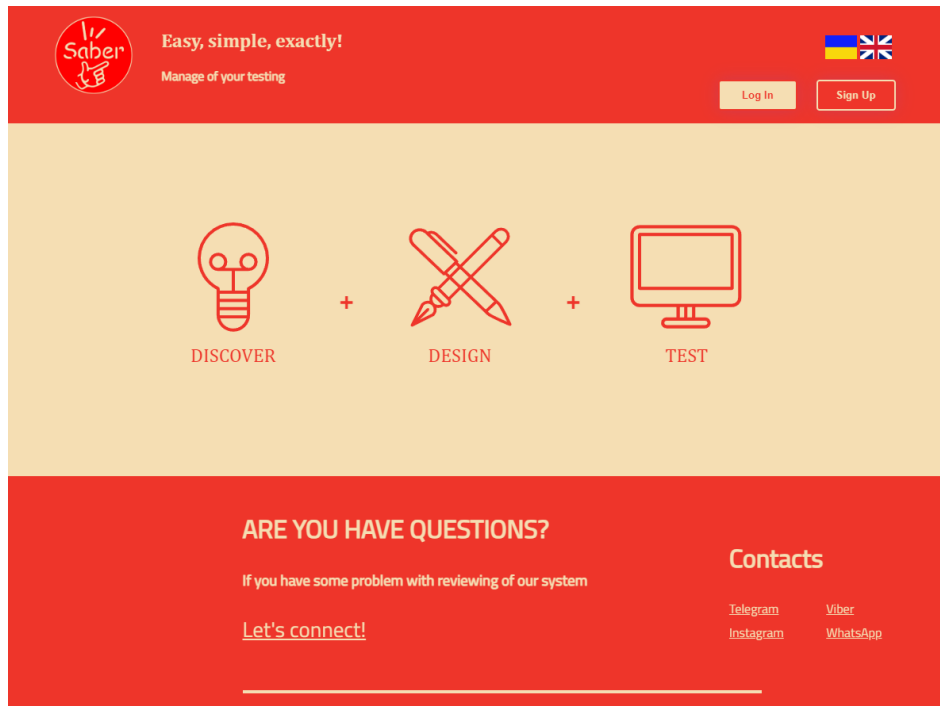


Рисунок 4.12 – Домашня сторінку web-системи. Джерело: побудовано автором

Якщо користувач не зареєстрований, він може відкрити поп-ап форму реєстрації та зареєструватися. Усі текстові поля та кнопки мають дизайнерську CSS-стилістику та приємну анімацію при взаємодії з ними (рис. 4.13).

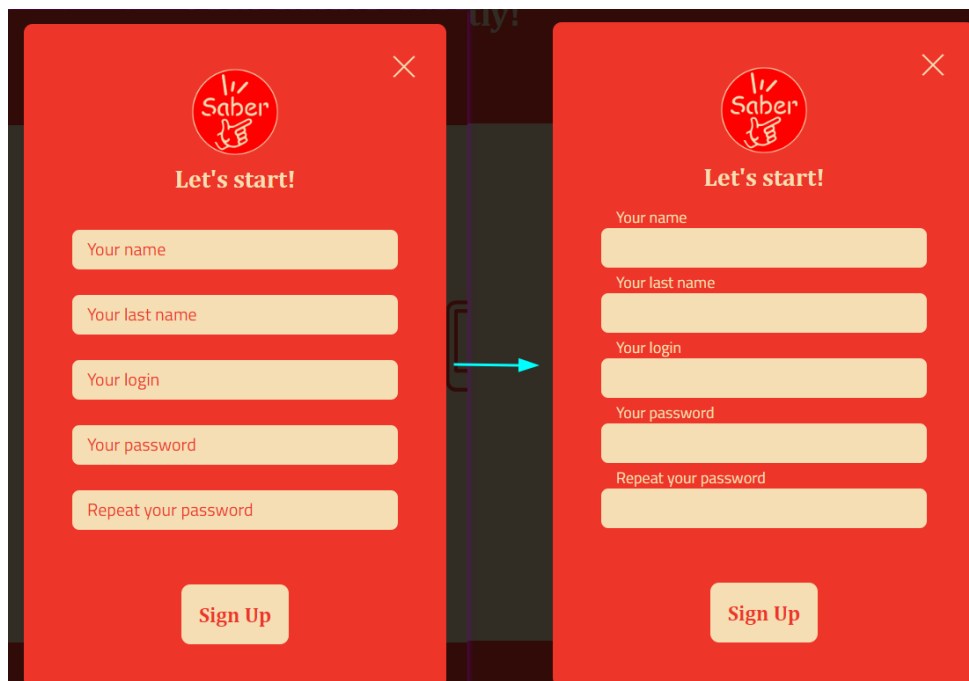


Рисунок 4.13 – Форма реєстрації web-додатку. Джерело: побудовано автором.

Після успішної реєстрації користувач може відкрити іншу форму – авторизації та увійти в новостворений акаунт (рис. 4.14).

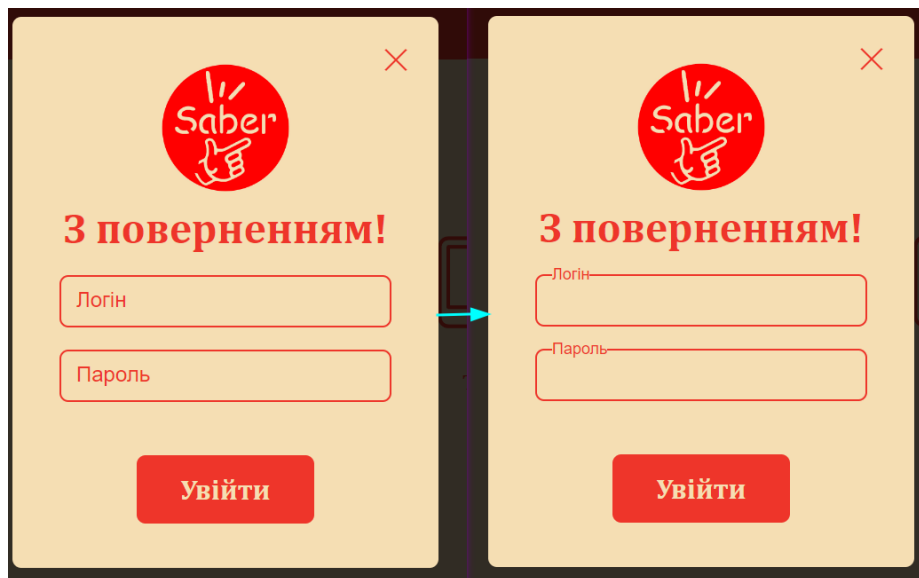


Рисунок 4.14 – Форма авторизації web-додатку. Джерело: побудовано автором.

Після першого входу в акаунт, користувач потрапляє на головну робочу сторінку де вже має змогу створити свій перший проект. Спочатку звісно, в нього не буде проектів і в такому випадку буде відображатися порожній стан – зображено на рисунку 4.15.

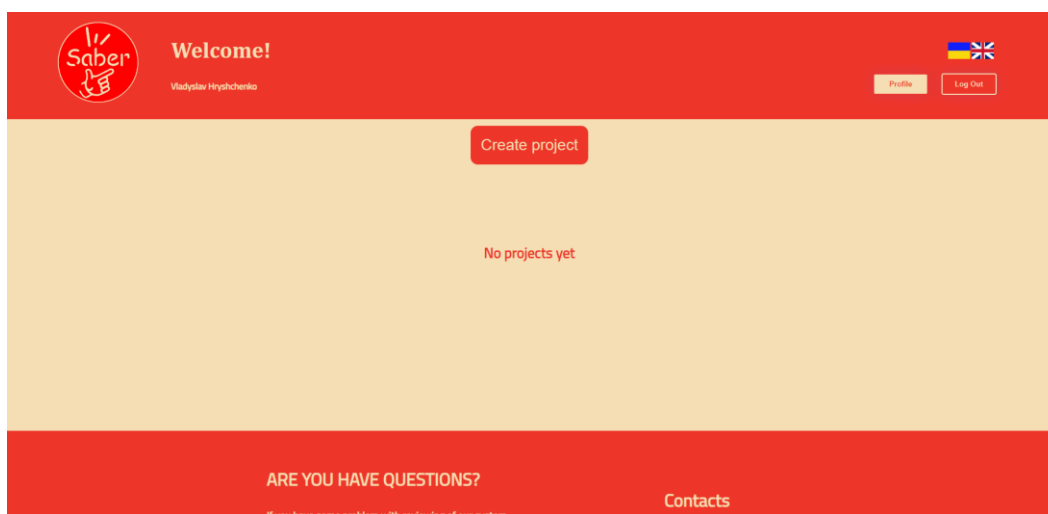


Рисунок 4.15 – Головна сторінка з порожнім станом. Джерело: побудовано автором.

Для створення свого першого проекту користувач може натиснути на кнопку «Створити проект» та заповнивши дані у поп-ап формі вже спостерігати його на своєму екрані (рис. 4.16).

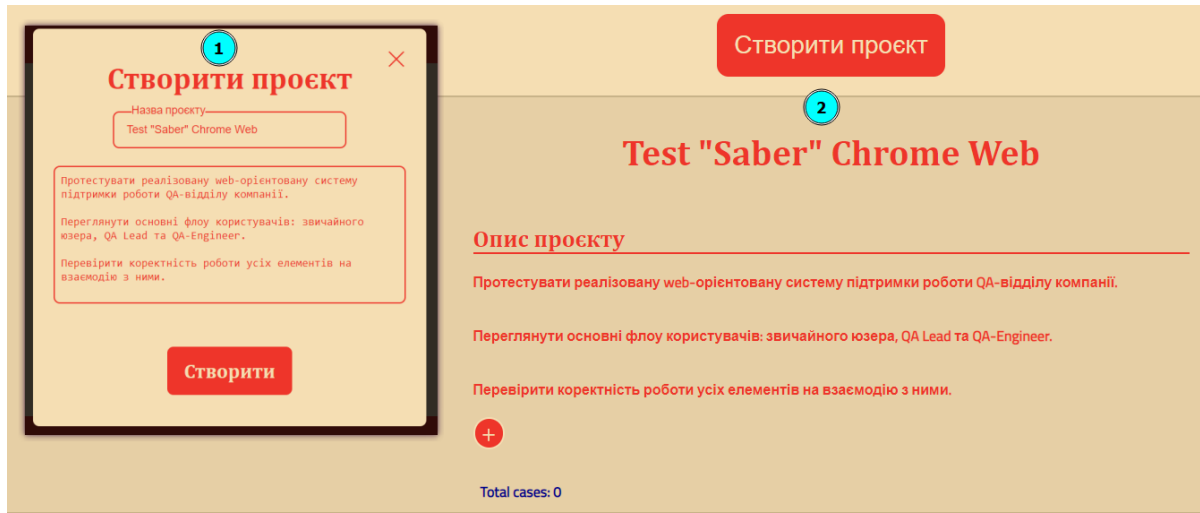


Рисунок 4.16 – Створення проекту. Джерело: побудовано автором.

Після створення проекту – користувач може додати до проекту списки перевірок використовуючи кнопку додавання. Приклад заповнення списками проекту наведено на рисунку 4.17.

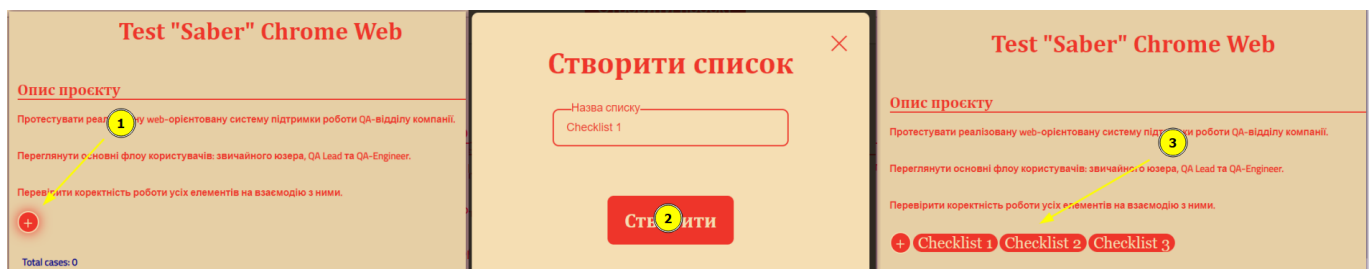


Рисунок 4.17 – Додавання списків перевірок до проекту. Джерело: побудовано автором.

При натисканні на створені списки відкривається таблиця для формування тестових сценаріїв (рис. 4.18).

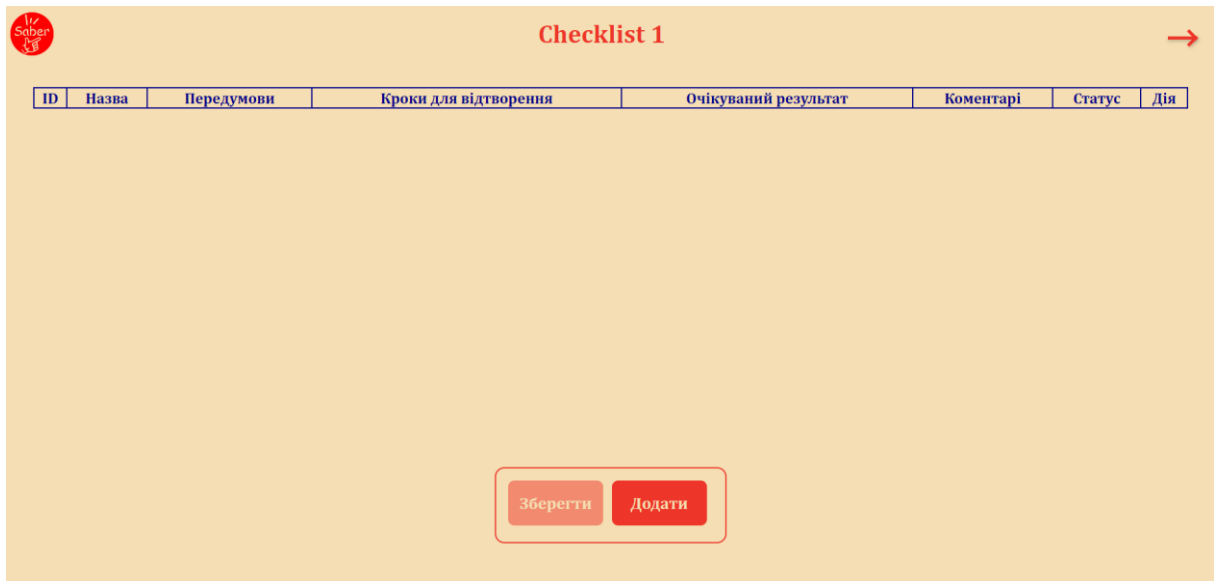


Рисунок 4.18 – Таблиця для тестових сценаріїв. Джерело: побудовано автором.

Після цього користувач може заповнювати таблицю своїми сценаріями тестування та взаємодіяти з їх статусами (рис. 4.19).

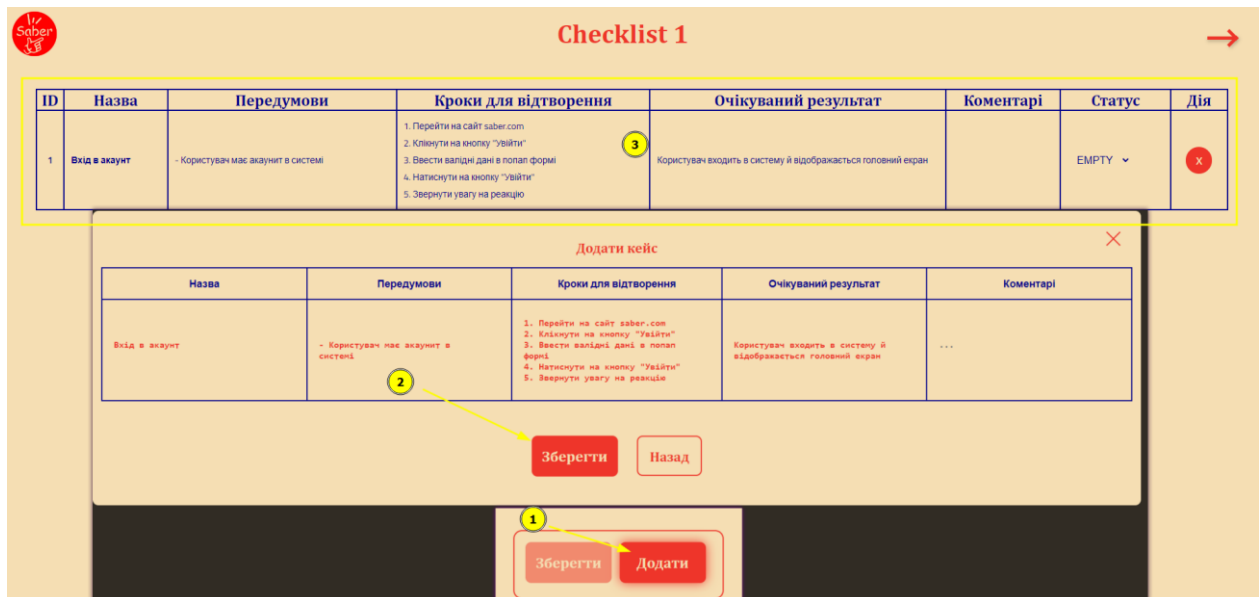


Рисунок 4.19 – Заповнення таблиці. Джерело: побудовано автором.

Також при необхідності редагування, реалізовано зручну функцію зміни даних в таблиці. Все що для цього потрібно – зробили дабл-клік по потрібній комірці. Функціонал редагування даних зображено на рисунку 4.20.

Checklist 1 →

ID	Назва	Передумови	Кроки для відтворення	Очікуваний результат	Коментарі	Статус	Дія
1	Вхід в акаунт	- Користувач має акаунт в системі	1. Перейти на сайт saber.com 2. Клікнути на кнопку "увійти" 3. Ввести валідні дані в пошуку форми 4. Натиснути на кнопку "увійти" 5. Звернути увагу на реакцію	Користувач входить в систему й відображається головний екран		EMPTY ▾	✘

Рисунок 4.20 – Заповнення таблиці. Джерело: побудовано автором.

Під час процесу контролю якості, користувач проходить створені тестові сценарії на виставляє статус результату по яким на картці проекту формується звіт (рис. 4.21).

ID	Назва	Передумови	Кроки для відтворення	Очікуваний результат	Коментарі	Статус	Дія
1	Вхід в акаунт	- Користувач має акаунт в системі	1. Перейти на сайт saber.com 2. Клікнути на кнопку "увійти" 3. Ввести валідні дані в пошуку форми 4. Натиснути на кнопку "увійти" 5. Звернути увагу на реакцію	Користувач входить в систему й відображається головний екран		PASS ▾	✘

Test "Saber" Chrome Web

Опис проекту

Протестувати реалізовану web-орієнтовану систему підтримки роботи QA-відділу компанії.

Переглянути основні флоу користувачів: звичайного юзера, QA Lead та QA-Engineer.

Перевірити коректність роботи усіх елементів на взаємодію з ними.

Total cases: 1

PASS: 1 (100%) FAIL: 0 (0%) EMPTY: 0 (0%)

Рисунок 4.21 – Формування звіту по статусу проходження тестування.

Джерело: побудовано автором.

Таким, чином користувач може працювати самостійно та створювати більше своїх карток проєктів та списків тестування. Якщо він перейде до сторінки профілю, то може редагувати свої дані акаунту за допомогою форми та створити

свою команду тестувальників і стати QA-менеджером (рис. 4.22). При цьому стані, йому доступні функції додавання співробітників та призначення їх на проекти (рис. 4.23).

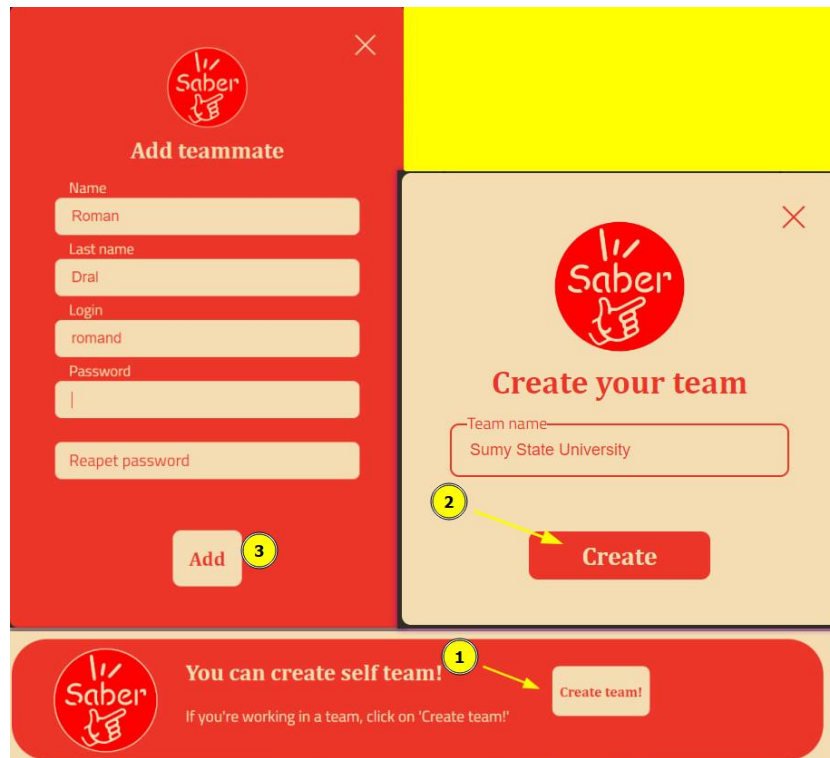


Рисунок 4.22 – Створення команди та додавання співробітників. Джерело: побудовано автором.



Рисунок 4.22 – Призначення співробітників на проекти. Джерело: побудовано автором.

4.5 Тестування web-системи

Для тестування web-системи було створено список перевірок з тестовими сценаріями використовуючи при цьому розроблений продукт. Результати створеної тестової документації та статуси тестування зображено на рисунку 4.23.

ID	Назва	Передумови	Кроки для відтворення	Очікуваний результат	Коментарі	Статус	Дія
1	Створення акаунту		1. Відкрити сайт 2. Натиснути на кнопку "Створити" 3. Ввести валідні дані 4. Натиснути на кнопку "створити" 5. Звернути увагу на реакцію	Акаунт користувача створено успішно		PASS	✗
2	Вхід в акаунт	- Створений акаунт	1. Відкрити сайт 2. Натиснути на кнопку "увійти" 3. Ввести валідні дані 4. Натиснути на кнопку "увійти" 5. Звернути увагу на реакцію	Головна робоча сторінка сайту відкривається		PASS	✗
3	Створення картки проекту	- Користувач увійшов в акаунт - Користувач не має команди або є QA Lead	1. Натиснути на кнопку "Створити проєкт" 2. Ввести назву та опис проєкту 3. Натиснути на кнопку "Створити" 4. Звернути увагу на реакцію	Створений проєкт відображається на сторінці		PASS	✗
4	Створення команди	- Користувач увійшов в акаунт - Користувач не має команди	1. Натиснути на кнопку "Профіль" у хедері 2. Прокрутити сторінку вниз до блоку створення команди 3. Натиснути на кнопку "Створити команду" 4. Ввести валідну назву команди	Назва створеної команди та статус користувача "QA Lead" відображається в хедері		PASS	✗
5	Призначення співробітника на проєкт	- Користувач увійшов в акаунт - Користувач є QA Lead - Користувач має проєкт - Користувач має співробітників які не призначені на проєкт	1. Натиснути на кнопку "Додати" 2. Натиснути на співробітника якого потрібно призначити на проєкт	Призначений співробітник відображається в картці проєкту		PASS	✗

Рисунок 4.23 – Список перевірок та тестові сценарії до web-системи. Джерело: побудовано автором.

ВИСНОВОК

У ході виконання дипломної роботи розроблено web-орієнтовану систему для підтримки роботи QA-відділу в IT-компаніях. Назвою було вирішено обрати «Saber».

Це включало в себе аналіз існуючих веб-систем та проведення дослідження виявлених функціональних вимог:

- створення, редагування, видалення тестових сценаріїв;
- створення окремих команд для розподілу об'єму робіт;
- можливість редагування статусів тестових сценаріїв;
- формування звіту по статусу проходження тестування.

Було проведено аналіз існуючих web-систем та дослідження виявлених функціональних вимог.

Вивчення предметної області виявило необхідність розробки web-орієнтованої системи для підтримки роботи QA-відділу в IT-компаніях, оскільки на даний момент не існує розроблених аналогів такої системи. Під час аналізу існуючих аналогів програмних продуктів були визначені ключові переваги, на які слід орієнтуватися на етапі проектування. Крім того, були визнані недоліки, які потребують подальшого доопрацювання під час реалізації проекту.

На наступному етапі розробки дипломного проекту були встановлені цілі, визначені функціональні та нефункціональні вимоги до web-системи. Були сформульовані завдання для його реалізації та сплановані робочі процеси, що допомогло визначити часові рамки проекту.

Далі в процесі розробки було проведено детальне проектування операційних процесів та сценаріїв використання майбутнього продукту. Це було досягнуто шляхом створення діаграм IDEF0, які допомагають розкласти загальні процеси на більш детальні. Крім того, для ілюстрації різних сценаріїв використання системи, була побудована діаграма варіантів використання (Use

Case). Такий підхід дозволив чітко визначити функціонал та розбити його на менші компоненти для зручності впровадження.

Таким чином, створена веб-орієнтована система підтримки для відділу контролю якості в ІТ-компаніях дозволить оптимізувати процес планування та проведення тестування, підвищить продуктивність роботи QA-спеціалістів, позитивно вплине на моральний дух QA-відділу й полегшить роботу QA-менеджеру, завдяки можливості розподілу завдань для роботи в команді.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Brown A. Applications of Deep Learning in Neural Networks // *Advanced Machine Learning Journal*. Academic Press, 2016. Vol. 72. P. 102–130.
2. Rodriguez-Recio A., Montero J.A.J., R. Alenda. Enhancing Genome-Assisted Evaluation in Large Datasets through the Boosting Algorithm // *Journal of Agricultural Informatics*. Springer, 2014. Vol. 103, № 2. P. 781–792.
3. Belgiu M., Drăgu L. Applications and Future Trends of Random Forest in Remote Sensing // *Remote Sensing Applications Journal*. Springer, 2018. Vol. 126. P. 52–69.
4. Sharma R. et al. Recent Advances and Emerging Challenges in Natural Language Processing // *Journal of Artificial Intelligence Research*. Elsevier, 2024. Vol. 95, № 1. P. 112–145.
5. Jira Software [Електронний ресурс] – Режим доступу до ресурсу: <https://kahoot.com/schools-u/>.
6. TestRail [Електронний ресурс] – Режим доступу до ресурсу: <https://classroom.google.com/u/0/h/>.
7. Xray [Електронний ресурс] – Режим доступу до ресурсу: <https://datorium.eu/>.
8. Alaidaros H., Omar M., Romli R. The state of the art of agile kanban method: challenges and opportunities // *Independent Journal of Management & Production*. Independent Journal of Management and Production, 2021. Vol. 12, № 8. P. 2535–2550.
9. Rodriguez M.Z. et al. Clustering algorithms: A comparative approach // *PLoS One*. Public Library of Science, 2019. Vol. 14, № 1. P. e0210236.
10. UML Use Case Diagram Tutorial | Lucidchart [Electronic resource]. URL: <https://www.lucidchart.com/pages/uml-use-case-diagram> (accessed: 22.11.2023).
11. Hinderks A. et al. A Methodology for Integrating User Experience Methods and Techniques into Agile Software Development

12. HTML [Електронний ресурс] – Режим доступу до ресурсу: <https://www.html.com/>.
13. Що таке CSS? [Електронний ресурс] – Режим доступу до ресурсу: https://css.in.ua/article/shcho-take-html_10/.
14. Мова JavaScript та її можливості [Електронний ресурс] – Режим доступу до ресурсу: <https://sites.google.com/mova-javascript-ta-ieie-mozlivosti/>.
15. Уроки з jQuery [Електронний ресурс] – Режим доступу до ресурсу: <https://astwellsoft.com/uk/blog/tehnology/jquery.html/>.
16. PHP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/>.
17. Bringing MySQL to the web [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/>.
18. Розпочніть разом з phpMyAdmin. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.phpMyAdmin.html/>
19. Налаштування Laragon [Електронний ресурс] – Режим доступу до ресурсу: <https://armedsoft.com/ua/blog/nalashtuvannya-laragon/>.
20. Laragon Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://laragon.org/docs/>.
21. Visual Studio Code IDE [Електронний ресурс] – Режим доступу до ресурсу: https://vs.code/article/top_10_ide_10/.
22. Microsoft Visual Studio Code Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://code.visualstudio.com/docs>.
23. Методологія IDEF0 [Електронний ресурс] – Режим доступу до ресурсу: https://stud.com.ua/87184/ekonomika/metodologiya_idef0/.
24. Дизайн високого рівня [Електронний ресурс] – <https://www.wiki.uk-ua.nina/az8F.html/>.
25. Розробка uml діаграми варіантів використання [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/5200239/page:6/>.

26. JavaScript Language and Its Capabilities [Электронный ресурс] – Режим доступа до ресурсу: <https://sites.google.com/mova-javascript-ta-ieie-mozlivosti/>.
27. jQuery Lessons [Электронный ресурс] – Режим доступа до ресурсу: <https://astwellsoft.com/uk/blog/tehnology/jquery.html/>.
28. PHP та SQL – запити [Электронный ресурс] – Режим доступа до ресурсу: <https://www.php.net/queries/sql>.
29. Bringing MySQL to the Web [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mysql.com/>.
30. Getting Started with phpMyAdmin [Электронный ресурс] – Режим доступа до ресурсу: <https://www.phpmyadmin.net/>.
31. Laragon Configuration [Электронный ресурс] – Режим доступа до ресурсу: <https://armedsoft.com/ua/blog/nalashtuvannya-laragon/>.
32. VS Code [Электронный ресурс] – Режим доступа до ресурсу: <https://code.visualstudio.com/info/>.
33. IDEF0 Creation [Электронный ресурс] – Режим доступа до ресурсу: <https://idef/diagrams.com.ua/>.
34. High-Level Design [Электронный ресурс] – Режим доступа до ресурсу: <https://www.wiki.uk-ua.nina./az8F.html/>.
35. Development of UML Use Case Diagrams [Электронный ресурс] – Режим доступа до ресурсу: <https://studfile.net/preview/5200239/page:6/>.
36. McConnell, S. (2004). Code Complete: A Practical Handbook of Software Construction. Microsoft Press.
37. Fowler, M. (2002). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley.

ДОДАТОК А

А1. Планування робіт

На сучасному етапі розвитку ІТ-індустрії web-орієнтовані системи стали невід'ємною частиною управління контролю якості та планування тестування в компаніях. Це особливо важливо для QA-менеджерів, які прагнуть оптимізувати робочі процеси та підвищити ефективність роботи команди. Автоматизація рутинних завдань, особливо тих, що потребують значних витрат часу та ручної праці, відкриває нові можливості для зосередження уваги на більш складних та стратегічних аспектах.

Web-додатки слугують фундаментальним інструментом підтримки QA-спеціалістів, відіграючи ключову роль в успішній реалізації проєктів.

А2. Деталізація мети методом SMART.

Чітке визначення цілей на етапі ескізного проєктування є ключовим фактором для забезпечення ефективного та якісного виконання проєкту. Використання методу SMART для деталізації цілей проєкту дозволяє структурувати та чітко визначити необхідні параметри, що сприяє кращому плануванню та виконанню проєкту. Результати цього процесу відображені в Таблиці А.1.

Таблиця А.1 – Деталізація мети проєкту методом SMART

Specific(Конкретна)	Розробити та імплементувати web-орієнтовану систему підтримки QA-інженерів для оптимізації їхньої роботи в IT-компаніях.
Measurable(Вимірювана)	Результатом створення дипломного проєкту є web-орієнтована система підтримки QA-відділу в IT-компаніях.
Achievable(Досяжна)	Для досягнення мети проєкту необхідні знання мов програмування JavaScript, PHP, HTML, CSS, бібліотеки jQuery, бази даних MySQL, та навичок створення документації.
Relevant(Реалістична)	Розроблена інформаційна технологія дозволить оптимізувати процес контролю якості розроблюваних продуктів, формувати звітність зі статусами тестування проєктів та розподілити обов'язки QA-відділу для роботи в команді.
Time-framed(Обмежена у часі)	Термін досягнення мети проєкту визначено з замовником і дорівнює 3 місяці.

А3. Планування змісту робіт.

Структура розбиття робіт (WBS) - це інструмент управління проєктами, який допомагає розбити проєкт на керовані компоненти, допомагаючи керівникам проєктів і командам у визначенні та структуруванні обсягу робіт. Метою розробки WBS є створення чіткої, детальної та ієрархічної структури. WBS організовує роботу в ієрархію, де верхній рівень представляє кінцевий результат, а кожен наступний, нижчий рівень забезпечує більш детальне визначення завдань. На рисунку А.1 представлено WBS для проєкту розробки web-орієнтованої системи підтримки QA-відділу в IT-компаніях.

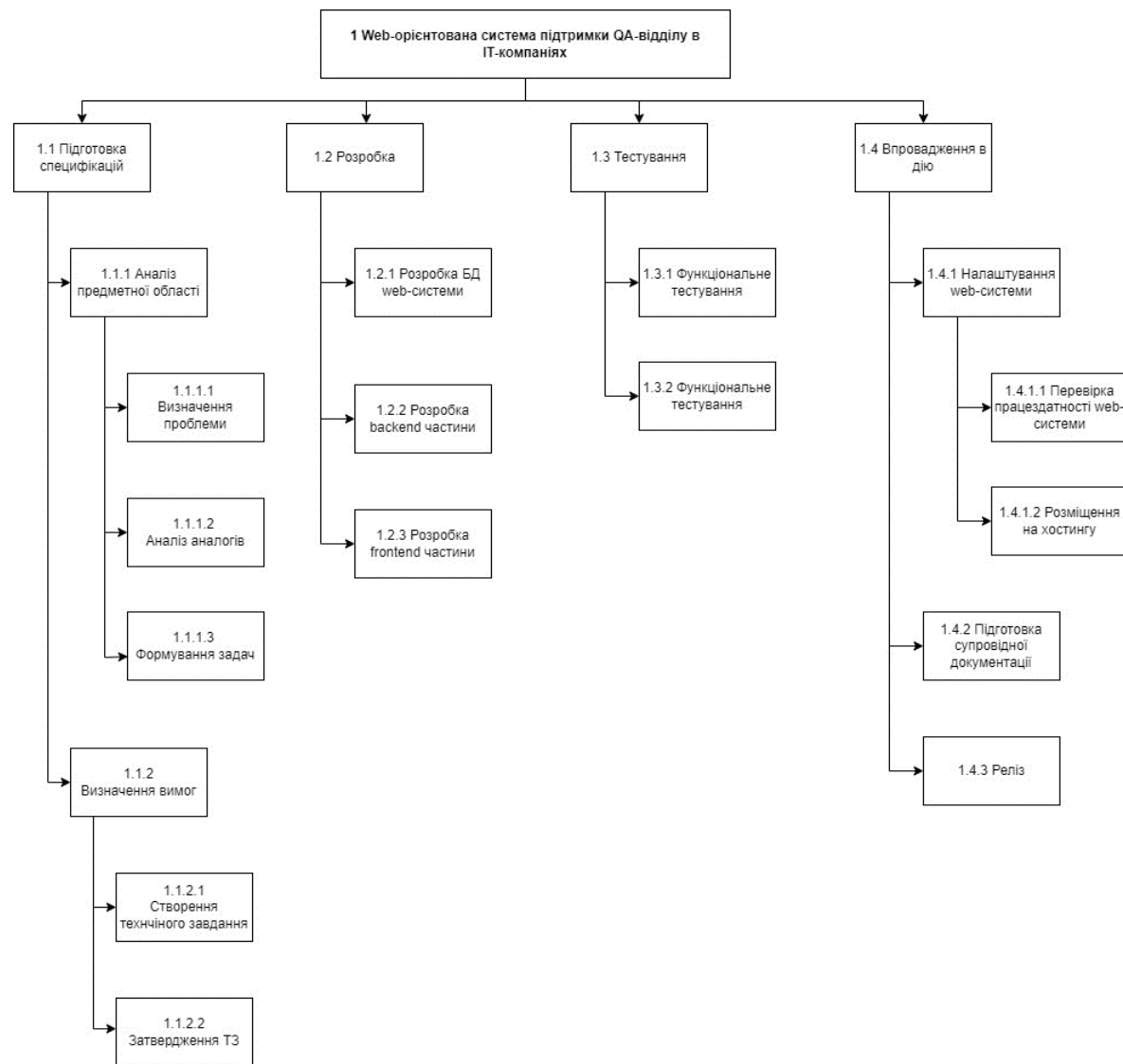


Рисунок А.1 – WBS-структура робіт проєкту

А4. Планування структури виконавців.

Розробка організаційної структури проєкту (OBS) є наступним кроком після декомпозиції процесів за допомогою WBS. OBS, або організаційна структура - це структура, яка використовується для ілюстрації ієрархії команди, що виконує проєкт, пояснюючи, як організована команда і як розподілені ресурси відповідно до цілей виконання проєкту. На рисунку А.2 зображено організаційну структуру для планування роботи над проєктом. Інформація про учасників проєкту наведена в таблиці А.2.

Таблиця А.2 – Виконавці проєкту

Роль	Ім'я	Проектна роль
Розробник	Грищенко В.В.	Розробка frontend-у та backend-у web-системи.
Тестувальник	Грищенко В.В.	Проведення тестування програмного продукту.
Проектувальник	Грищенко В.В.	Проектування структури програмного продукту та бази даних.
Керівник проєкту	Чибіряк Я.І.	Створення завдання на розробку проєкту, рецензія імплементованого програмного продукту.
Менеджер	Грищенко В.В.	Контроль над дотриманням дедлайну, розподіл ресурсів на розробку проєкту. Проведення аналізу та збору інформації.

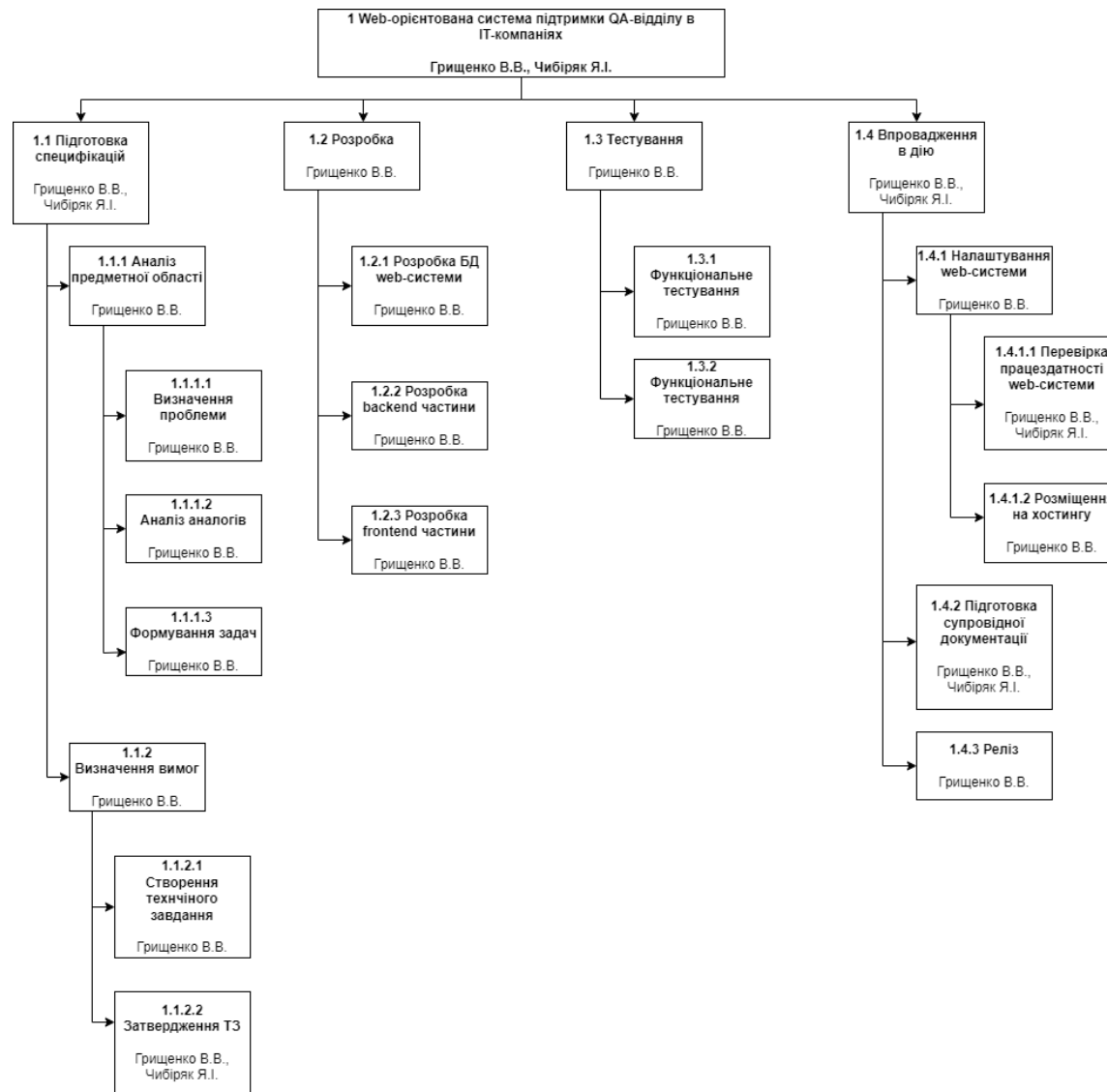


Рисунок А.2 – OBS-структура робіт проєкту

A5. Діаграма Ганта.

Розробка календарного плану проєкту є ключовим компонентом управління проєктом. Цей графік являє собою діаграму з розподілом часових рамок для кожного завдання, що полегшує точне визначення загальної тривалості проєкту з урахуванням наявних ресурсів. Календарний план проєкту проілюстровано на рисунку А.3.

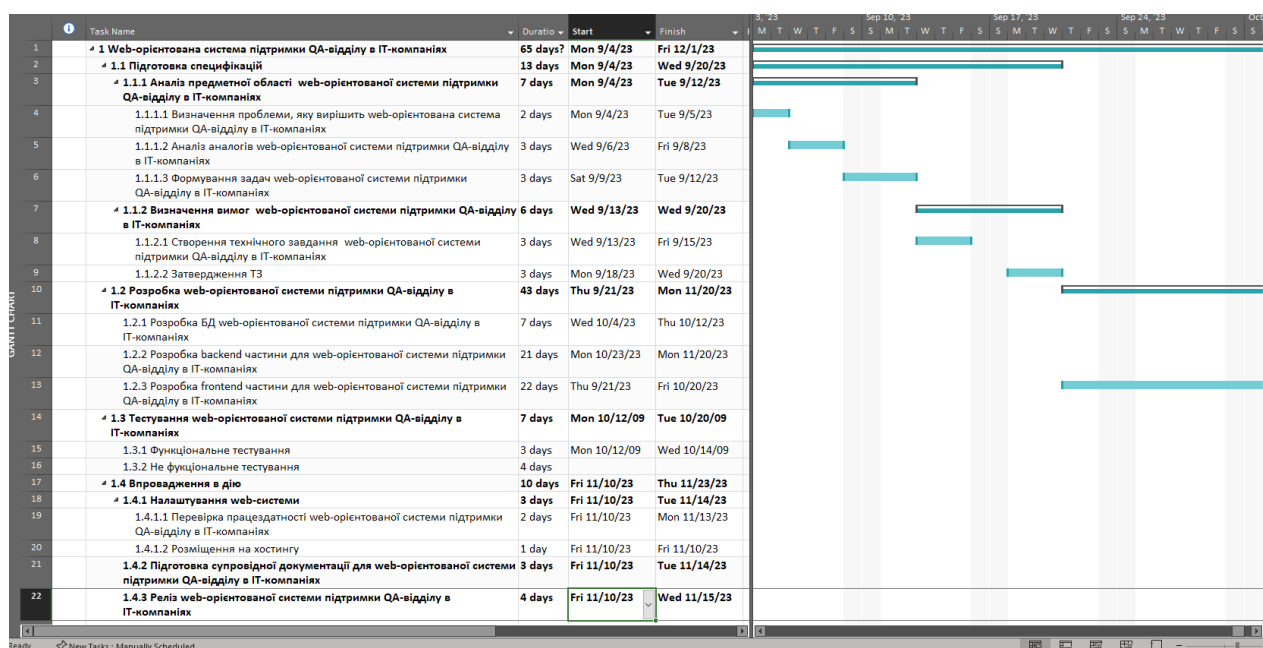


Рисунок А.3 – Календарний графік проєкту

A6. Управління ризиками проєкту.

На етапі планування проєкту особлива увага приділяється аналізу та управлінню ризиками. Цей процес включає кілька етапів: ідентифікацію ризиків, оцінку їх впливу та ймовірності, розробку стратегій для їх пом'якшення або усунення, а також моніторинг і контроль за застосуванням цих стратегій. Ризики цього проєкту перераховані в таблиці А.3. Оцінки ризиків наведені в таблиці А.4. У таблиці А.5 наведено шкалу ризиків за типом, величиною впливу та ймовірністю.

Таблиця А.3 – Ризики проекту.

№ ризику	Назва (опис) ризику
1	Відсутність кваліфікованих знань
2	Технічні збої у обладнанні
3	Проблеми з інтернетом
4	Проблема з електропостачанням
5	Відсутність достатнього фінансування проекту
6	Захворювання одного з членів команди
7	Недостатня якість вихідного коду
8	Поява альтернативного продукту
9	Недостатній контроль якості
10	Зміна вимог замовника

Таблиця А.4 – Результати визначення ймовірності, впливу та рангу ризику проекту.

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
1	Відсутність кваліфікованих знань	0,1	0,2	0,02
2	Технічні збої у обладнанні	0,3	0,2	0,06
3	Проблеми з інтернетом	0,5	0,05	0,025
4	Проблема з електропостачанням	0,3	0,2	0,06

Продовження таблиці А.4

№ ризику	Назва (опис) ризику	Ймовірність (0,1-0,9)	Вплив (0,05-0,8)	Ранг
5	Відсутність достатнього фінансування проєкту	0,5	0,2	0,1
6	Захворювання одного з членів команди	0,1	0,4	0,04
7	Недостатня якість вихідного коду	0,7	0,2	0,14
8	Поява альтернативного продукту	0,5	0,2	0,1
9	Недостатній контроль якості	0,3	0,2	0,06
10	Зміна вимог замовника	0,1	0,8	0,08

Таблиця А.5 – Шкала оцінювання ризику типом, ймовірністю та величиною впливу.

Оцінка	Ймовірність виникнення	Вплив ризику	Тип ризику
1	Низька	Низький	Прийнятні
2	Середня	Середній	Виправдані
3	Висока	Високий	Недопустимі

Для мінімізації негативного впливу потенційних загроз на проєкт критично важливо розробити детальний план реагування на ризики. Цей план повинен включати аналіз та оцінку ефективності запропонованих стратегій зниження

ризиків з урахуванням потенційних наслідків для проекту. Така оцінка базується на критеріях, визначених у таблиці А.5. В результаті створення плану реагування була сформована матриця, що відображає ймовірність і вплив різних ризиків, як описано в таблиці А.6. Ця матриця використовує кольори для класифікації ризиків: зелений - прийнятні, жовтий - виправдані та червоний - неприйнятні ризики.

Таблиця А.6 – Матриця ймовірності та впливу

Ймовірність виникнення ризику	Вплив ризику				
	0,05	0,1	0,2	0,4	0,8
0,9	0,045	0,09	0,18	0,3	0,72
0,7	0,035	0,07	0,14	0,28	0,56
0,5	0,025	0,05	0,1 R3, R4	0,2	0,4
0,3	0,015	0,03 R1	0,06	0,12	0,24
0,1	0,005	0,01	0,02	0,04	0,08 R2

У таблиці А.7 наведено класифікацію ризиків за рівнями відповідно до значень їх індексів. Детальний опис ризиків та стратегій управління ними наведено в таблиці А.8.

Таблиця А.7 – Шкала оцінювання за рівнем ризику.

№	Назва	Межі	Ризики, які входять(номера)
1	Прийнятні	$0,005 \leq R \leq 0,05$	1, 3, 6, 9
2	Виправдані	$0,05 \leq R \leq 0,14$	2, 4, 5, 7, 8, 10
3	Недопустимі	$0,14 \leq R \leq 0,72$	-

Таблиця А.8 – Ризики та стратегії реагування

ID	Статус ризику	Опис ризику	Ймовірність виникнення	Вплив ризику	Ранг ризику	План А	Тип стратегії реагування	План Б
RS-1	Відкритий	Відсутність кваліфікованих знань	Низька	Середній	0,02	<ol style="list-style-type: none"> 1. Ознайомитися з потрібними технологіями (пришвидшені курси і тд.). 2. Обрати більш доступну технологію, якою можна замінити потреби. 	Зменшення	Найняти розробника, який буде мати досвід та знання з відповідного стеку технологій
RS-2	Відкритий	Технічні збої у обладнанні	Низька	Середній	0,06	<ol style="list-style-type: none"> 1. Регулярне обслуговування обладнання. 2. Мати резервні копії інформації та розроблюваної системи. 	Зменшення	Мати запасне обладнання.

Продовження таблиці А.8

RS-3	Відкритий	Проблеми з інтернетом	Середня	Низький	0,025	Мати декілька джерел інтернету	Прийняття	Знайти найближче легкодоступне місце з інтернетом (кафе, коворкінг)
RS-4	Відкритий	Проблема з електропостачанням	Низька	Середній	0,06	Мати альтернативне джерело електроенергії.	Прийняття	Знайти найближче місце з можливістю підключитися до електроенергії.
RS-5	Відкритий	Відсутність достатнього фінансування проекту	Середня	Середній	0,1	Підписання контракту з замовником щодо своєчасного та повноцінного фінансування послуг	Зменшення	Мати резервні кошти для перекриття недостатнього фінансування

Продовження таблиці А.8

RS-6	Відкритий	Захворюванн я одного з членів команди	Низька	Високий	0,04	Замінити розробника іншою людиною на короткий термін	Прийняття	Пошук підрядників або аутсорсинг певних задач
RS-7	Відкритий	Недостатня якість вихідного коду	Висока	Середній	0,14	Забезпечення увімкнення автоматичного збереження даних, створення локальних копій.	Зменшення	Створення локальних та хмарних копій.
RS-8	Новий	Поява альтернативн ого продукту	Середня	Середній	0,1	Уникання витоку інформації з кола розробників задля збереження унікальності продукту.	Прийняття	

Продовження таблиці А.8

RS-9	Відкритий	Недостатній контроль якості	Низька	Середній	0,06	Впровадження системи управління якістю та регулярні перевірки	Зменшення	Залучення зовнішніх аудиторів для незалежної оцінки якості
RS-10	Відкритий	Зміна вимог замовника	Низька	Високий	0,08	Регулярне узгодження вимог з замовником	Прийняття	Гнучке управління проектом, що дозволяє адаптуватися до змін

ДОДАТОК Б

Б.1. Лістинг коду файлу index.php

```

<!DOCTYPE html>
<html lang="en">

<head>
  <link rel="icon" href="img/logo1.ico" type="image/x-icon">
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1"
/>

  <title>Saber - Home</title>

  <!-- Main style -->
  <link type="text/css" rel="stylesheet" href="css/style.css" />

  <!-- Google fonts -->
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Cairo" />
</head>

<body>
  <header>
    
    <div>
      <h1 class="lang" key="parag1">Easy, simple, exactly!</h1>
      <h2 class="lang" key="parag2">Manage of your testing</h2>
    </div>
    <div class="buttons">
      <div class="language">
        <button id="ua" class="translate" name="ua">
          
        </button>
        <button id="en" class="translate" name="en">
          
        </button>
      </div>
      <div class="logreg">
        <button class="log lang" key="log-in" id="open-popup-log"
name="log-in">
          Log In
        </button>
        <button class="reg lang" key="signup" id="open-popup-reg">
          Sign Up
      </div>
    </div>
  </header>

```

```

        </button>
    </div>
</div>
</header>
<div class="images">
    <div class="icontext">
        
        <h2 class="lang" key="discover">DISCOVER</h2>
    </div>
    <h1>+</h1>
    <div class="icontext">
        
        <h2 class="lang" key="design">DESIGN</h2>
    </div>
    <h1>+</h1>
    <div class="icontext">
        
        <h2 class="lang" key="test">TEST</h2>
    </div>
</div>

<div class="popup-log" id="popup-log">
    <div class="popup-log-container">
        <div class="popup-log-body">
            
            <p class="lang" key="welcome">Welcome!</p>
            <div id="error-msg" class="alert">
                <div class="error-text" id="error-message"></div>
            </div>

            <form id="loginForm" action="auth.php" method="post">
                <input type="text" class="lang" id="login" name="login" />
                <div class="title1 lang" key="login" id="title1">Login</div>

                <input type="password" class="lang" id="passwd"
name="passwd" />
                <div class="title2 lang" key="passwd"
id="title2">Password</div>

                <button class="lang" key="log-in" type="button"
onclick="submitForm()">Log In</button>
            </form>

            <div class="popup-close-log" id="close-popup-log"></div>
        </div>
    </div>
</div>

<div class="popup-reg" id="popup-reg">
    <div class="popup-reg-container">

```

```

<div class="popup-reg-body">
  
  <p class="lang" key="start">Let's start!</p>
  <div id="error-msg-reg" class="alert">
    <div class="error-text" id="error-message-reg"></div>
  </div>

  <form id="regForm" action="reg.php" method="post">
    <input type="text" class="lang" id="name" name="name" />
    <div class="title3 lang" key="name" id="title3">Your
name</div>

    <input type="text" class="lang" id="lname" name="lname" />
    <div class="title4 lang" key="lname" id="title4">Your last
name</div>

    <input type="text" class="lang" id="login-reg" name="login-
reg" />
    <div class="title5 lang" key="login-reg" id="title5">Your
login</div>

    <input type="password" class="lang" id="passwd-reg"
name="passwd-reg" />
    <div class="title6 lang" key="passwd-reg" id="title6">Your
password</div>

    <input type="password" class="lang" id="rep-passwd-reg"
name="rep-passwd-reg" />
    <div class="title7 lang" key="rep-passwd-reg"
id="title7">Repeat your password</div>

    <button class="lang" key="signup" type="button"
onclick="submitRegistrationForm()">Sign Up</button>
  </form>

  <div class="popup-close-reg" id="close-popup-reg"></div>
</div>
</div>
</div>
<footer>
  <div class="workus">
    <h1 class="lang" key="want">ARE YOU HAVE QUESTIONS?</h1>
    <h3 class="lang" key="got">If you have some problem with
reviewing of our system</h3>
    <a href="mailto:vladicainferno@gmail.com" class="lang"
key="connect">Let's connect!</a>
  </div>
  <div class="contacts">
    <h1 class="lang" key="contacts">Contacts</h1>
    <div class="cont1">

```

```

        <a href="https://t.me/VladicaInferno">Telegram</a>
        <a href="https://instagram.com/vladicaInferno">Instagram</a>
    </div>
    <div class="cont2">
        <a href="https://viber://add?chat=380991118491">Viber</a>
        <a href="https://wa.me:/380991118491">WhatsApp</a>
    </div>
</div>
<hr />
</footer>
<script src="js/jquery.min.js"></script>
<script src="js/reg.js"></script>
<script src="js/auth.js"></script>
<script src="js/popup-home.js"></script>
</body>

</html>

```

Б.2. Лістинг коду файла main.php

```

<?php

session_start();
require_once 'connect.php';

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <link rel="icon" href="img/logol1.ico" type="image/x-icon">
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1"
/>

    <title>Saber - Main</title>

    <!-- Main style -->
    <link type="text/css" rel="stylesheet" href="css/style.css" />
    <link type="text/css" rel="stylesheet" href="css/style-list.css" />

    <!-- Google fonts -->
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Cairo" />
</head>

<body>

```

```

<header>
  
  <div>
    <h1 class="lang" <?php

      if ($_SESSION['user']['groupid'] == NULL) {
        ?>key="welcome">

          <?php
            echo "Welcome!";

          } else {
            ?>>
            <?php
              $groupid = $_SESSION['user']['groupid'];
              $group = mysqli_query($mysql, "SELECT * FROM `groups` WHERE
id_group = '$groupid'");
              $groupname = mysqli_fetch_assoc($group);

              echo $groupname['group_name'];
            }
            ?>
          </h1>
          <h2 class="lang">
            <?php

              if ($_SESSION['user']['roleid'] == 3) {

                echo "QA Lead: ";

              } else if ($_SESSION['user']['roleid'] == 2) {

                echo "QA Engineer: ";

              } else {

                echo "";

              }

              print_r($_SESSION['user']['name']);
              echo " ";
              print_r($_SESSION['user']['lname']);

            ?>
          </h2>
        </div>
        <div class="buttons">
          <div class="language">
            <button id="ua" class="translate">

```

```

        
    </button>
    <button id="en" class="translate">
        
    </button>
</div>
<div class="logreg">

    <button class="log lang" key="profile" id="open-profile">
        Profile
    </button>

    <button class="reg lang" key="logout" id="open-popup-out">
        Log Out
    </button>
</div>
</div>
</header>

<div class="popup-out" id="popup-out">
    <div class="popup-out-container">
        <div class="popup-out-body">
            <p class="lang" key="ausure">Are you sure?</p>
            <button class="out lang" key="logout" id="logoutButton">Log
Out</button>
            <button class="cancel lang" key="cancel" id="cancel-
out">Cancel</button>
            <div class="popup-close-out" id="close-popup-out"></div>
        </div>
    </div>
</div>

<?php
    $userid = $_SESSION['user']['id'];
    $result1 = mysqli_query($mysql, "SELECT * FROM `users` INNER JOIN
`asigns` ON users.id_user=asigns.id_user INNER JOIN `projects` ON
asigns.id_project=projects.id_project WHERE users.id_user = '$userid'
ORDER BY projects.id_project DESC;");
    if ($_SESSION['user']['roleid'] == 3 || $_SESSION['user']['roleid']
== 1):
        ?>

        <div class="manage-buttons">
            <button id="open-popup-add-project" class="open-popup-add-
project lang" key="createproject">Create project</button>
        </div>

<?php endif;

if (mysqli_num_rows($result1) == 0) {

```



```

?>
<div class="empty-projects">
  <p class="lang" key="noprojects">No projects yet</p>
</div>
<?php }
$projectId = null;
?>

<div class="add-project-popup" id="add-project-popup">
  <div class="add-project-popup-container">
    <div class="add-project-popup-body">
      <p class="lang" key="createproject">Create project</p>

      <form id="projectForm" action="addproject.php" method="post">
        <input type="text" class="lang" id="project-name"
name="project-name" required />
        <div class="title-project lang" key="projectname" id="title-
project">Project name</div>

        <div class="title-desc lang" key="projectdesc" id="title-
desc">Project description</div>
        <textarea id="project-desc" data-field-name="project"
name="project-desc"></textarea>

        <button class="lang" key="create"
type="submit">Create</button>
      </form>

      <div class="close-add-project" id="close-add-project"></div>
    </div>
  </div>
</div>

<?php

$listCount = 0;

while ($select_while1 = mysqli_fetch_assoc($result1)) {
  $projectId = $select_while1['id_project'];

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

    $assignedUsersQuery = mysqli_query($mysql, "SELECT users.name,
users.lname FROM `users` INNER JOIN `asigns` ON
users.id_user=asigns.id_user WHERE asigns.id_project = '$projectId'
AND users.id_role = 2");
    $assignedUsers = array();

    $NotassignedUsersQuery = mysqli_query($mysql, "SELECT * FROM
`users` WHERE users.id_group = " . $_SESSION['user']['groupid'] . "

```

```

AND users.id_role = 2 AND NOT EXISTS (SELECT 1 FROM assigns WHERE
assigns.id_user = users.id_user AND assigns.id_project =
'$projectId');
    $NotassignedUsers = array();

    } else {
        $assignedUsersQuery = mysqli_query($mysql, "SELECT users.name,
users.lname FROM `users` INNER JOIN `assigns` ON
users.id_user=assigns.id_user WHERE assigns.id_project = '$projectId'
AND users.id_role = 2");
        $assignedUsers = array();

        $NotassignedUsersQuery = mysqli_query($mysql, "SELECT * FROM
`users` WHERE users.id_role = 2 AND NOT EXISTS (SELECT 1 FROM assigns
WHERE assigns.id_user = users.id_user AND assigns.id_project =
'$projectId')");
        $NotassignedUsers = array();
    }
?>

<div class="project-card" id="project-card">
    <h1>
        <?php echo $select_while1['project_name']; ?>
    </h1>

    <div class="card">
        <h3 class="lang" key="projectdesc">Description</h3>
        <p class="desclink">
            <?php echo nl2brToBr($select_while1['description']); ?>
        </p>

        <?php
            $projectId = $select_while1['id_project'];
            $result2 = mysqli_query($mysql, "SELECT * FROM `lists` WHERE
id_project = '$projectId'");

            ?>

            <button id="open-popup-add-list-<?php echo $listCount; ?>"
class="open-popup-add-list lang">+</button>

            <div class="add-list-popup" id="add-list-popup-<?php echo
$listCount; ?>">
                <div class="add-list-popup-container">
                    <div class="add-list-popup-body">
                        <p class="lang" key="createlist">Create list</p>

                        <form id="listForm" action="addlist.php" method="post">
                            <input type="text" class="lang" id="list-name-<?php
echo $listCount; ?>" name="listname" required />

```

```

                <div class="title-list lang" key="listname" id="title-
list-<?php echo $listCount; ?>">
                    List name
                </div>
                <input class="not-display" name="project-id"
value="<?php echo $projectId; ?>" />

                <button class="lang" key="create"
type="submit">Create</button>
            </form>

            <div class="close-add-list" id="close-add-list-<?php
echo $listCount; ?>"></div>
        </div>
    </div>
</div>

<?php
    while ($select_while2 = mysqli_fetch_assoc($result2)) {
        $listId = $select_while2['id_list'];
        $casesResult = mysqli_query($mysql, "SELECT * FROM `cases`
WHERE id_list = '$listId'");
        ?>
        <button class="open-popup-list" id="open-popup-list-<?php
echo $select_while2['id_list']; ?>">
            <?php echo $select_while2['list_name']; ?>
        </button>

        <?php
    }

    $pass = mysqli_query($mysql, "SELECT
COUNT(*) AS total_cases,
SUM(CASE WHEN c.id_status = 1 THEN 1 ELSE 0 END) AS
status_1_cases,
SUM(CASE WHEN c.id_status = 2 THEN 1 ELSE 0 END) AS
status_2_cases,
SUM(CASE WHEN c.id_status = 3 THEN 1 ELSE 0 END) AS
status_3_cases,
FLOOR((SUM(CASE WHEN c.id_status = 1 THEN 1 ELSE 0 END)
/ COUNT(*)) * 100) AS percentage_status_1,
FLOOR((SUM(CASE WHEN c.id_status = 2 THEN 1 ELSE 0 END)
/ COUNT(*)) * 100) AS percentage_status_2,
FLOOR((SUM(CASE WHEN c.id_status = 3 THEN 1 ELSE 0 END)
/ COUNT(*)) * 100) AS percentage_status_3
FROM
cases c
JOIN lists l ON c.id_list = l.id_list
JOIN projects p ON l.id_project = p.id_project

```

```

        WHERE
        p.id_project = $projectId;");

    $passcases = mysqli_fetch_assoc($pass);

    echo "<p class='total-report'>Total cases: " .
    $passcases['total_cases'] . "</p>";

    if ($passcases['total_cases'] == 0) {

        echo " ";

    } else {

        echo "<div class='status-report'><p class='pass'>PASS: " .
        $passcases['status_1_cases'] . " (" .
        $passcases['percentage_status_1'] . "%)</p>";
        echo "<p class='fail'>FAIL: " . $passcases['status_2_cases']
        . " (" . $passcases['percentage_status_2'] . "%)</p>";
        echo "<p class='empty'>EMPTY: " .
        $passcases['status_3_cases'] . " (" .
        $passcases['percentage_status_3'] . "%)</p></div>";

    }

    $listCount++;

    while ($user = mysqli_fetch_assoc($assignedUsersQuery)) {
        $assignedUsers[] = $user['name'] . ' ' . $user['lname'];
    }

    $assignedUsersString = implode(', ', $assignedUsers);

    if ($assignedUsersString != NULL) {

        echo '<h3 class="assign-to lang" key="assigned">Assigned
to:</h3><p class="assigned-users">' . $assignedUsersString . '</p>';

    }

    ?>

    <?php if ($_SESSION['user']['roleid'] == 3): ?>

        <button id="open-add-users-modal-<?php echo $listCount; ?>"
class="open-add-users-modal lang"
        key="addtoproject">Assign</button>

    <?php endif; ?>

```

```

        <div class="add-users-modal" id="add-users-modal-<?php echo
$listCount; ?>">
            <div class="add-users-modal-container">
                <div class="add-users-modal-body">
                    <p class="lang" key="addtoproject">Assign</p>

                    <form id="AddToProjectForm" action="add_user_asisn.php"
method="post">
                        <?php
                            while ($user =
mysqli_fetch_assoc($NotassignedUsersQuery)) {
                                $NotassignedUsers[] = $user;
                            }

                            if ($NotassignedUsers == NULL): ?>
                                <h4 class="no-teammates lang" key="noteammates">No
teammates remaining</h4>
                                <?php endif;

                                foreach ($NotassignedUsers as $NotAsign) {
                                    echo '<button type="submit" class="assign-users"
name="user_id" value="" . $NotAsign['id_user'] . ''>' .
                                        $NotAsign['name'] . ' ' . $NotAsign['lname'] .
'</button>';
                                }
                                $projectId = $select_while1['id_project'];
                                ?>
                                <input class="not-display" name="project-id"
value="<?php echo $projectId ?>" />
                                </form>

                                <div class="close-add-users-modal" id="close-add-users-
modal-<?php echo $listCount; ?>"></div>
                                </div>
                            </div>
                        </div>

                    </div>
                </div>

                <?php
            }
            ?>

<div class="popup-list" id="popup-list">
    <div class="head">
        

        <h2 id="popup-list-title"></h2>

```

```

    <div class=" arrow-hover" id="close-popup-list">
      <div></div>
    </div>
  </div>

  <table class="main-table" id="editableTable">
    <tr>
      <th>ID</th>
      <th class="lang" key="summ">Summary</th>
      <th class="lang" key="prec">Preconditions</th>
      <th class="lang" key="str">Steps to reproduce</th>
      <th class="lang" key="er">Expected result</th>
      <th class="lang" key="comm">Comments</th>
      <th class="lang" key="status">Status</th>
      <th class="lang" key="action">Action</th>
    </tr>
    <tbody id="list-cases" class="list-cases"></tbody>
  </table>
</div>

<div class="list-buttons" id="list-buttons">
  <button class="save lang" key="save" id="save">Save</button>
  <button class="add lang" key="add" id="open-popup-
add">Add</button>
</div>

<div class="popup-delete" id="popup-delete">
  <div class="popup-delete-container">
    <div class="popup-delete-body">
      <p class="lang" key="ausure">Are you sure?</p>
      <button class="delete-butt lang" key="delete"
onclick="deleteRow()" id="deleteButton">Delete</button>
      <button class="cancel lang" key="cancel" id="cancel-delete"
onclick="closeDeletePopup()">Cancel</button>
      <div class="popup-close-delete" id="close-popup-delete"
onclick="closeDeletePopup()"></div>
    </div>
  </div>
</div>

<div class="popup-add" id="popup-add">
  <div class="popup-add-container">
    <div class="popup-add-body">
      <p class="lang" key="addcase">Add case</p>
      <table id="addTable">
        <tr>
          <th class="lang" key="summ">Summary</th>
          <th class="lang" key="prec">Preconditions</th>
          <th class="lang" key="str">Steps to reproduce</th>

```

```

        <th class="lang" key="er">Expected result</th>
        <th class="lang" key="comm">Comments</th>
    </tr>
    <tr>
        <td>
            <textarea class="addable" id="area1" data-field-
name="summ" placeholder="..."></textarea>
        </td>
        <td>
            <textarea class="addable" id="area2" data-field-
name="prec" placeholder="..."></textarea>
        </td>
        <td>
            <textarea class="addable" id="area3" data-field-
name="str" placeholder="..."></textarea>
        </td>
        <td>
            <textarea class="addable" id="area4" data-field-
name="er" placeholder="..."></textarea>
        </td>
        <td>
            <textarea class="addable" id="area5" data-field-
name="comm" placeholder="..."></textarea>
        </td>
    </tr>
</table>
    <button class="save-add lang" key="save" id="save-add"
onclick="saveAddData()">Save</button>
    <button class="cancel lang" key="cancel" id="cancel-add">
    Cancel
    </button>
    <div class="popup-close-add" id="close-popup-add"></div>
</div>
</div>
<div class="workus">
    <h1 class="lang" key="want">ARE YOU HAVE QUESTIONS?</h1>
    <h3 class="lang" key="got">
    If you have some problem with reviewing of our system
    </h3>
    <a href="mailto:vladicainferno@gmail.com" class="lang"
key="connect">Let's connect!</a>
</div>
<div class="contacts">
    <h1 class="lang" key="contacts">Contacts</h1>
    <div class="cont1">
        <a href="https://t.me/VladicaInferno">Telegram</a>
        <a href="https://instagram.com/vladicaInferno">Instagram</a>
    </div>

```

```

    <div class="cont2">
      <a href="https://viber://add?chat=380991118491">Viber</a>
      <a href="https://wa.me:/380991118491">WhatsApp</a>
    </div>
  </div>
  <hr />
</footer>

</body>

<script src="js/save_data.js"></script>

<script>

  document.getElementById('open-profile').addEventListener('click',
function () {
  window.location.href = 'profile.php';
});

const deleteButton = document.getElementById('deleteButton');
const deletePopup = document.getElementById('popup-delete');

function openDeletePopup(caseId) {
  deleteButton.setAttribute('data-case-id', caseId);
  deletePopup.classList.add('active');
}

function closeDeletePopup() {
  deletePopup.classList.remove('active');
}

function deleteRow() {

  const caseId = deleteButton.getAttribute('data-case-id');

  fetch('delete_case.php', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({ caseId: caseId }),
  })
  .then(response => response.json())
  .then(data => {
    if (data.success) {

      const deletedRow = document.querySelector(`tr[data-case-
id="${caseId}"]`);

      if (deletedRow) {

```



```

        deletedRow.remove();

        const rows = document.querySelectorAll('#list-cases tr');
        rows.forEach((row, index) => {
            const rowCaseId = row.getAttribute('data-case-id');
            if (rowCaseId) {
                row.querySelector('.center').innerText = index + 1;
            }
        });
    }
    closeDeletePopup();
} else {
    console.error('Error deleting case:', data.error);
}
})
.catch(error => {
    console.error('Error deleting case:', error);
});
}

function saveAddData() {

    const area1 = document.getElementById('area1').value;
    const area2 = document.getElementById('area2').value;
    const area3 = document.getElementById('area3').value;
    const area4 = document.getElementById('area4').value;
    const area5 = document.getElementById('area5').value;

    const listId = document.getElementById('not-
display').getAttribute('data-list-id');

    fetch('save_add_data.php', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json',
        },
        body: JSON.stringify({
            area1: area1,
            area2: area2,
            area3: area3,
            area4: area4,
            area5: area5,
            listId: listId,
        }),
    })
    .then(response => response.json())
    .then(data => {
        if (data.success) {
            console.log('Data saved successfully!');
        }
    });
}

```

```

        location.reload();
    } else {
        console.error('Error saving data:', data.error);
    }
})
.catch(error => {
    console.error('Error saving data:', error);
});
});
}

const saveButton = document.getElementById('save');
const listCases = document.getElementById('list-cases');
saveButton.setAttribute('disabled', '');

listCases.addEventListener('input', function () {
    const hasChanges = checkForChanges();
    saveButton.disabled = !hasChanges;
    saveButton.classList.toggle('active', hasChanges);
});

function checkForChanges() {
    const editableCells = listCases.querySelectorAll('.editable');
    let hasChanges = false;

    editableCells.forEach(function (editableCell) {
        const originalContent = editableCell.getAttribute('data-original-content');
        const currentContent = editableCell.innerHTML.trim();

        if (originalContent !== currentContent) {
            hasChanges = true;
        }
    });

    return hasChanges;
}

<?php
$userid = $_SESSION['user']['id'];
$result1 = mysqli_query($mysql, "SELECT * FROM `users` INNER JOIN `asigns` ON users.id_user=asigns.id_user INNER JOIN `projects` ON asigns.id_project=projects.id_project WHERE users.id_user = '$userid'");

$projectId = null;

while ($select_while1 = mysqli_fetch_assoc($result1)) {
    $projectId = $select_while1['id_project'];
}

```

```

$result3 = mysqli_query($mysql, "SELECT * FROM `lists` WHERE
id_project = '$projectId'");

while ($select_while2 = mysqli_fetch_assoc($result3)) {
    ?>
    const openPopUpList<?php echo $select_while2['id_list']; ?> =
document.getElementById('open-popup-list-<?php echo
$select_while2['id_list']; ?>');

    openPopUpList<?php echo $select_while2['id_list'];
?>.addEventListener('click', function (e) {
    document.getElementById('popup-list-title').innerText = "<?php
echo $select_while2['list_name']; ?>";

    const ListButtons = document.getElementById('list-buttons');
    ListButtons.classList.add('active');

    var casesHTML = '';

    <?php
    $listId = $select_while2['id_list'];
    $cases = mysqli_query($mysql, "SELECT * FROM `cases` WHERE
id_list = '$listId'");

    echo 'casesHTML += \<div class="not-display" id="not-display"
data-list-id="" . $listId . "">\';';

    $num = 1;

    while ($case = mysqli_fetch_assoc($cases)) {

        echo 'casesHTML += \<tr data-case-id="" . $case['id_case']
. "">\';';
        echo 'casesHTML += \<td class="center" >' . $num .
'\</td>\';';
        echo 'casesHTML += \<td class="summary editable" id="summ"
data-case-id="" . $case['id_case'] . "">' . nl2brToBr($case['summ']) .
'\</td>\';';
        echo 'casesHTML += \<td class="editable" id="prec" data-
case-id="" . $case['id_case'] . "">' . nl2brToBr($case['prec']) .
'\</td>\';';
        echo 'casesHTML += \<td class="editable" id="str" data-
case-id="" . $case['id_case'] . "">' . nl2brToBr($case['str']) .
'\</td>\';';
        echo 'casesHTML += \<td class="editable" id="er" data-case-
id="" . $case['id_case'] . "">' . nl2brToBr($case['er']) . '\</td>\';';
        echo 'casesHTML += \<td class="editable" id="comm" data-
case-id="" . $case['id_case'] . "">' . nl2brToBr($case['comm']) .
'\</td>\';';
        echo 'casesHTML += \<td id="status">\';';

```

```

        echo 'casesHTML += \'<select class="statusDropdown"
onchange="changeColor(this)">\';';

        $statusOptionsResult = mysqli_query($mysql, "SELECT * FROM
`statuses`");
        while ($statusOption =
mysqli_fetch_assoc($statusOptionsResult)) {
            $selected = ($statusOption['id_status'] ==
$case['id_status']) ? 'selected' : '';
            echo 'casesHTML += \'<option class="opts" .
$statusOption['id_status'] . " value=" . $statusOption['id_status']
. " ' . $selected . '>' . $statusOption['status_name'] .
'</option>\';';
        }
        echo 'casesHTML += \'</select></td>\';';
        echo 'casesHTML += \'<td id="delete"><button class="delete"
onclick="openDeletePopup(' . $case['id_case'] . ') " data-case-id="' .
$case['id_case'] . '">X</button></td>\';';
        echo 'casesHTML += \'</tr>\';';
        $num++;
    }
?>;

document.getElementById('list-cases').innerHTML = casesHTML;

document.getElementById('popup-list').classList.add('active');

const listCases = document.getElementById('list-cases');
listCases.querySelectorAll('.editable').forEach(function
(editableCell) {
    const cellContent = editableCell.innerHTML.trim();

    const isLink = /(http(s)?:\//\//[\^\s]+)/g.test(cellContent);

    if (isLink) {
        const replacedContent =
cellContent.replace(/(http(s)?:\//\//[\^\s]+)/g, '<a href="$1"
target="_blank">$1</a>');
        const tempDiv = document.createElement('div');

        tempDiv.innerHTML = replacedContent;
        editableCell.innerHTML = tempDiv.innerHTML;
    }
});
});
<?php
}
}
?>

```

```

const closePopUpList = document.getElementById('close-popup-list');
const ListButtons = document.getElementById('list-buttons');
closePopUpList.addEventListener('click', function (e) {
  document.getElementById('popup-list').classList.remove('active');
  ListButtons.classList.remove('active');
});

document.addEventListener("DOMContentLoaded", function () {
  const saveAddButton = document.getElementById('save-add');
  const addTable = document.getElementById('addTable');
  saveAddButton.setAttribute('disabled', '');

  addTable.addEventListener('input', function () {
    const hasChanges = checkForChangesInAddTable();
    saveAddButton.disabled = !hasChanges;
    saveAddButton.classList.toggle('active', hasChanges);
  });

  function checkForChangesInAddTable() {
    const editableCells = addTable.querySelectorAll('.addable');
    let hasChanges = false;

    editableCells.forEach(function (editableCell) {
      const originalContent = editableCell.getAttribute('data-original-content');
      const currentContent = editableCell.value.trim();

      if (originalContent !== currentContent) {
        hasChanges = true;
      }
    });
  }

  return hasChanges;
});
});
</script>

<script>
  // Попап створення ліста

  let i = 0;

  while (true) {
    const openPopUpAddList = document.getElementById(`open-popup-add-list-${i}`);
    const closePopUpAddList = document.getElementById(`close-add-list-${i}`);
  }

```

```

    const PopUpAddList = document.getElementById(`add-list-popup-
    ${i}`);

    const openPopUpAddUsersToProject = document.getElementById(`open-
    add-users-modal-${i}`);
    const closePopUpAddUsersToProject =
    document.getElementById(`close-add-users-modal-${i}`);
    const PopUpAddUsersToProject = document.getElementById(`add-users-
    modal-${i}`);

    if (!openPopUpAddList && !openPopUpAddUsersToProject) {
        break;
    }

    if (openPopUpAddList) {
        openPopUpAddList.addEventListener('click', function (e) {
            PopUpAddList.classList.add('active');
        });

        closePopUpAddList.addEventListener('click', function (e) {
            PopUpAddList.classList.remove('active');
        });

        const TitleList = document.getElementById(`title-list-${i}`);
        const ListName = document.getElementById(`list-name-${i}`);

        ListName.addEventListener('focus', function (e) {
            TitleList.classList.add('move1');
        });

        TitleList.addEventListener('click', function (e) {
            TitleList.classList.add('move1');
            ListName.focus();
        });
    }

    if (openPopUpAddUsersToProject) {
        openPopUpAddUsersToProject.addEventListener('click', function
    (e) {
            PopUpAddUsersToProject.classList.add('active');
        });

        closePopUpAddUsersToProject.addEventListener('click', function
    (e) {
            PopUpAddUsersToProject.classList.remove('active');
        });

        const TitleList = document.getElementById(`title-list-${i}`);
        const ListName = document.getElementById(`list-name-${i}`);

```

```

    ListName.addEventListener('focus', function (e) {
        TitleList.classList.add('move1');
    });

    TitleList.addEventListener('click', function (e) {
        TitleList.classList.add('move1');
        ListName.focus();
    });

    const TitleDesc = document.getElementById('title-desc');
    const Desc = document.getElementById('project-desc');

    Desc.addEventListener('focus', function (e) {
        TitleDesc.classList.add('move2');
    });

    TitleDesc.addEventListener('click', function (e) {
        TitleDesc.classList.add('move2');
        Desc.focus();
    });
}

i++;
}

</script>

<script>
    // Попап створення проекту

    const openPopUpAddProject = document.getElementById('open-popup-add-
project');
    const closePopUpAddProject = document.getElementById('close-add-
project');
    const PopUpAddProject = document.getElementById('add-project-
popup');

    openPopUpAddProject.addEventListener('click', function (e) {
        PopUpAddProject.classList.add('active');
        console.log('Button clicked!');
    })

    closePopUpAddProject.addEventListener('click', function (e) {
        PopUpAddProject.classList.remove('active');
    })

    const TitleProject = document.getElementById('title-project');
    const ProjectName = document.getElementById('project-name');

    ProjectName.addEventListener('focus', function (e) {

```

```

    TitleProject.classList.add('move1');
  })

TitleProject.addEventListener('click', function (e) {
  TitleProject.classList.add('move1');
  ProjectName.focus();
})

const TitleDesc = document.getElementById('title-desc');
const Desc = document.getElementById('project-desc');

Desc.addEventListener('focus', function (e) {
  TitleDesc.classList.add('move2');
})

TitleDesc.addEventListener('click', function (e) {
  TitleDesc.classList.add('move2');
  Desc.focus();
})
</script>

<script>
  // Попап виходу

const openPopUpOut = document.getElementById('open-popup-out');
const closePopUpOut = document.getElementById('close-popup-out');
const PopUpOut = document.getElementById('popup-out');

openPopUpOut.addEventListener('click', function (e) {
  PopUpOut.classList.add('active');
})

closePopUpOut.addEventListener('click', function (e) {
  PopUpOut.classList.remove('active');
})

const CancelOut = document.getElementById('cancel-out');

CancelOut.addEventListener('click', function (e) {
  PopUpOut.classList.remove('active');
})
</script>

<script>
  // Попап створення кейсу

const openPopUpAdd = document.getElementById('open-popup-add');
const closePopUpAdd = document.getElementById('close-popup-add');
const PopUpAdd = document.getElementById('popup-add');

```



```

openPopUpAdd.addEventListener('click', function (e) {
    PopUpAdd.classList.add('active');
})

closePopUpAdd.addEventListener('click', function (e) {
    PopUpAdd.classList.remove('active');
})

const CancelAdd = document.getElementById('cancel-add');

CancelAdd.addEventListener('click', function (e) {
    PopUpAdd.classList.remove('active');
})
</script>

<script src="js/auth.js"></script>
<script src="js/jquery.min.js"></script>
<script src="js/table.js"></script>
<script src="js/logout.js"></script>

</html>

<?php
function nl2brToBr($string)
{
    return str_replace(["\r\n", "\r", "\n"], '<br>', $string);
}
?>

```

Б.3. Лістинг коду файлу profile.php

```

<?php

session_start();
require_once 'connect.php';

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <link rel="icon" href="img/logo1.ico" type="image/x-icon">
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1" />

    <title>Profile</title>

```

```

    <!-- Main style -->
    <link type="text/css" rel="stylesheet" href="css/style.css" />
    <link type="text/css" rel="stylesheet" href="css/style-list.css"
/>
    <link type="text/css" rel="stylesheet" href="css/style-
profile.css" />

    <!-- Google fonts -->
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Cairo" />
</head>

<body>
    <header>
        
        <div>
            <h1 class="lang" <?php

                if ($_SESSION['user']['groupid'] == NULL) {
                    ?>key="profile">

                        <?php
                            echo "Profile";

                } else {
                    ?>>
                        <?php
                            $groupid = $_SESSION['user']['groupid'];
                            $group = mysqli_query($mysql, "SELECT * FROM
`groups` WHERE id_group = '$groupid'");
                            $groupname = mysqli_fetch_assoc($group);

                            echo $groupname['group_name'];
                        }
                    ?>
                </h1>
                <h2 class="lang">
                    <?php

                        if ($_SESSION['user']['roleid'] == 3) {

                            echo "QA Lead: ";

                        } else if ($_SESSION['user']['roleid'] == 2) {

                            echo "QA Engineer: ";

                        } else {

```

```

        echo "";

    }

    print_r($_SESSION['user']['name']);
    echo " ";
    print_r($_SESSION['user']['lname']);

    ?>
</h2>
</div>
<div class="buttons">
    <div class="language">
        <button id="ua" class="translate">
            
        </button>
        <button id="en" class="translate">
            
        </button>
    </div>
    <div class="logreg">

        <button class="log lang" key="back" id="open-main">
            Back
        </button>

        <button class="reg lang" key="logout" id="open-popup-
out">
            Log Out
        </button>
    </div>
</div>
</header>

<div class="popup-out" id="popup-out">
    <div class="popup-out-container">
        <div class="popup-out-body">
            <p class="lang" key="ausure">Are you sure?</p>
            <button class="out lang" key="logout"
id="logoutButton">Log Out</button>
            <button class="cancel lang" key="cancel" id="cancel-
out">Cancel</button>
            <div class="popup-close-out" id="close-popup-
out"></div>
        </div>
    </div>
</div>

<div class="center">

```

```

<?php if ($_SESSION['user']['roleid'] == 3): ?>

    <button class="add-teammate lang" id="add-teammate"
key="addteammate">Add teammate</button>

<?php endif; ?>

</div>

<div class="add-teammate-popup" id="add-teammate-popup">
    <div class="add-teammate-popup-container">
        <div class="add-teammate-popup-body">
            
            <p class="lang" key="addteammate">Add teammate</p>

            <form action="add_teammate.php" method="post"
id="addTeammateForm">

                <input type="text" id="teammate-name"
name="teammate-name" />
                <div class="titlename lang" key="nameteammate"
id="title1">
                    Name
                </div>

                <input type="text" id="teammate-lname"
name="teammate-lname" />
                <div class="titlelname lang" key="lnameteammate"
id="title2">
                    Last name
                </div>

                <input type="text" id="teammate-login"
name="teammate-login" />
                <div class="titlelogin lang" key="loginteammate"
id="title3">
                    Login
                </div>

                <input type="password" id="teammate-passwd"
name="teammate-passwd" />
                <div class="titlepasswd lang" key="passwdteammate"
id="title4">
                    Password
                </div>

                <input type="password" id="teammate-rep-passwd"
name="teammate-rep-passwd" />
                <div class="titlerepasswd lang"
key="repasswdteammate" id="title5">

```

```

        Repeat password
    </div>

    <button class="lang" key="add" type="button"
onclick="addTeammate()">Create</button>

</form>

<div class="close-add-teammate" id="close-add-
teammate"></div>
</div>
</div>
</div>

<div class="ed-profile" id="ed-profile">
    <div class="ed-profile-container">
        <div class="ed-profile-body">
            <h1 class="lang" key="profileadd">Profile editing</h1>

            <div id="error-msg" class="alert">
                <div class="error-text" id="error-message"></div>
            </div>

            <form method="post" action="update_profile.php"
id="updateForm">

                <div class="title3-ed lang" key="ed-name"
id="title3-ed">
                    Change name:
                </div>
                <input type="text" class="lang" name="name"
value="<?php print_r($_SESSION['user']['name']); ?>" />

                <div class="title4-ed lang" key="ed-lname"
id="title4-ed">
                    Change last name:
                </div>
                <input type="text" class="lang" name="lname"
value="<?php
print_r($_SESSION['user']['lname']); ?>" />

                <div class="title5-ed lang" key="ed-cur-passwd"
id="title5-ed">
                    Current password:
                </div>
                <input type="password" class="lang"
name="current_password" />

                <div class="title6-ed lang" key="ed-passwd"
id="title6-ed">

```

```

                New password:
            </div>
            <input type="password" class="lang"
name="new_password" />

            <div class="title7-ed lang" key="ed-rep-passwd"
id="title7-ed">
                Repeat new password:
            </div>
            <input type="password" class="lang"
name="confirm_password" />

            <button class="save-butt lang" key="save"
type="button" onclick="updateForm()">Save</button>

        </form>
    </div>
</div>
</div>

<div class="company-container">

    <?php if ($_SESSION['user']['groupid'] == NULL): ?>

        <div class="company-text">
            <h1 class="lang" key="youcancreate">You can create
self team!</h1>
            <p class="lang" key="ifyou">If you're working in a
team, click on 'Create team!'</p>
        </div>

            <button class="create-team lang" id="open-popup-team"
key="createteam">Create team!</button>

        <?php endif; ?>

    </div>

    <div class="add-team-popup" id="add-team-popup">
        <div class="add-team-popup-container">
            <div class="add-team-popup-body">
                
                <p class="lang" key="createyourteam">Create self
team</p>

                <form action="addteam.php" method="post"
id="createTeamForm">

```

```

        <input type="text" class="lang" id="team-name"
name="team_name" required />
        <div class="title1-create lang" key="teamname"
id="title1-create">Team name</div>

        <button class="lang" key="create" type="button"
onclick="createTeam()">Create</button>

    </form>

    <div class="close-add-team" id="close-add-team"></div>
</div>
</div>
</div>

<footer>
    <div class="workus">
        <h1 class="lang" key="want">ARE YOU HAVE QUESTIONS?</h1>
        <h3 class="lang" key="got">
            If you have some problem with reviewing of our system
        </h3>
        <a href="mailto:vladicainferno@gmail.com" class="lang"
key="connect">Let's connect!</a>
    </div>
    <div class="contacts">
        <h1 class="lang" key="contacts">Contacts</h1>
        <div class="cont1">
            <a href="https://t.me/VladicaInferno">Telegram</a>
            <a
href="https://instagram.com/vladicaInferno">Instagram</a>
        </div>
        <div class="cont2">
            <a
href="https://viber://add?chat=380991118491">Viber</a>
            <a href="https://wa.me:/380991118491">WhatsApp</a>
        </div>
    </div>
    <hr />
</footer>

<script>

    document.addEventListener('DOMContentLoaded', function () {

        var logoutButton =
document.getElementById('logoutButton');

        logoutButton.addEventListener('click', function () {

            window.location.href = 'index.php';

```

```

        });
    });
</script>

<script>

    document.getElementById('open-main').addEventListener('click',
function () {
    window.location.href = 'main.php';
    });

</script>

<script>

    const openPopUpAddTeammate = document.getElementById('add-
teammate');
    const closePopUpAddTeammate = document.getElementById('close-
add-teammate');

    const PopUpAddTeammate = document.getElementById('add-
teammate-popup');
    const TeammateName = document.getElementById('teammate-name');
    const TeammateLastName = document.getElementById('teammate-
lname');
    const TeammateLogin = document.getElementById('teammate-
login');
    const TeammatePassword = document.getElementById('teammate-
passwd');
    const TeammateRepPassword = document.getElementById('teammate-
rep-passwd');
    const Title1 = document.getElementById('title1');
    const Title2 = document.getElementById('title2');
    const Title3 = document.getElementById('title3');
    const Title4 = document.getElementById('title4');
    const Title5 = document.getElementById('title5');

    openPopUpAddTeammate.addEventListener('click', function (e) {
        console.log('Кнопка Add teammate нажата');
        PopUpAddTeammate.classList.add('active');
    })

    closePopUpAddTeammate.addEventListener('click', function (e) {
        PopUpAddTeammate.classList.remove('active');
    })

    TeammateName.addEventListener('focus', function (e) {
        Title1.classList.add('move1');
    })

```



```

TeammateLastName.addEventListener('focus', function (e) {
    Title2.classList.add('move2');
})

TeammateLogin.addEventListener('focus', function (e) {
    Title3.classList.add('move3');
})

TeammatePassword.addEventListener('focus', function (e) {
    Title4.classList.add('move4');
})

TeammateRepPassword.addEventListener('focus', function (e) {
    Title5.classList.add('move5');
})

Title1.addEventListener('click', function (e) {
    Title1.classList.add('move1');
    TeammateName.focus();
})

Title2.addEventListener('click', function (e) {
    Title2.classList.add('move2');
    TeammateLastName.focus();
})

Title3.addEventListener('click', function (e) {
    Title3.classList.add('move3');
    TeammateLogin.focus();
})

Title4.addEventListener('click', function (e) {
    Title4.classList.add('move4');
    TeammatePassword.focus();
})

Title5.addEventListener('click', function (e) {
    Title5.classList.add('move5');
    TeammateRepPassword.focus();
})
</script>

<script>
    // Попап "Створення команди"

    const openPopUpTeam = document.getElementById('open-popup-
team');
    const closePopUpTeam = document.getElementById('close-add-
team');
    const PopUpTeam = document.getElementById('add-team-popup');

```

```

openPopUpTeam.addEventListener('click', function (e) {
    PopUpTeam.classList.add('active');
})

closePopUpTeam.addEventListener('click', function (e) {
    PopUpTeam.classList.remove('active');
})

const TitleCreate = document.getElementById('title1-create');
const TeamName = document.getElementById('team-name');

TeamName.addEventListener('focus', function (e) {
    TitleCreate.classList.add('move1');
})

TitleCreate.addEventListener('click', function (e) {
    TitleCreate.classList.add('move1');
    TeamName.focus();
})
</script>
<script>
    // Попап виходу

const openPopUpOut = document.getElementById('open-popup-
out');
const closePopUpOut = document.getElementById('close-popup-
out');
const PopUpOut = document.getElementById('popup-out');

openPopUpOut.addEventListener('click', function (e) {
    PopUpOut.classList.add('active');
})

closePopUpOut.addEventListener('click', function (e) {
    PopUpOut.classList.remove('active');
})

const CancelOut = document.getElementById('cancel-out');

CancelOut.addEventListener('click', function (e) {
    PopUpOut.classList.remove('active');
})
</script>

<script src="js/jquery.min.js"></script>
<script src="js/update_profile.js"></script>
<script src="js/create_team.js"></script>
<script src="js/add_teammate.js"></script>
<script src="js/logout.js"></script>

```

```
</body>
```

```
</html>
```

Б.4. Лістинг коду файлу add_teammate.php

```
<?php
```

```
session_start();
require_once 'connect.php';

$name = $_POST['teammate-name'];
$lname = $_POST['teammate-lname'];
$login = $_POST['teammate-login'];
$password = $_POST['teammate-passwd'];
$reppassword = $_POST['teammate-rep-passwd'];
$groupid = $_SESSION['user']['groupid'];
$q = mysqli_fetch_assoc(mysqli_query($mysql, "SELECT COUNT(*) FROM
`users` WHERE `login` = '$login'"));
$check = join("", $q);

if (preg_match('/^([0-9a-zA-Z-_.]{5,16})$/ ', $login) || $login == "")
{

    if ($check > 0) {

        $_SESSION['alert'] = 'This login is exists!';
        header('Content-Type: application/json');
        echo json_encode(["error" => $_SESSION['alert']]);
        exit();

    } elseif ($password <> $reppassword) {

        $_SESSION['alert'] = 'Passwords do not match!';
        header('Content-Type: application/json');
        echo json_encode(["error" => $_SESSION['alert']]);
        exit();

    } elseif ($login == "" || $password == "" || $name == "" || $lname
== "" || $reppassword == "") {

        $_SESSION['alert'] = 'Fill in all fields!';
        header('Content-Type: application/json');
        echo json_encode(["error" => $_SESSION['alert']]);
        exit();

    } else {

        $pass = md5($password);
```

```

        $result = mysqli_query($mysql, "INSERT INTO `users`
(`id_user`, `name`, `lname`, `login`, `passwd`, `id_role`, `id_group`)
VALUES (NULL, '$name', '$lname', '$login', '$pass', '2',
'$groupid')");

        $_SESSION['alert'] = 'Registration success!';
        header('Content-Type: application/json');
        echo json_encode(["success" => $_SESSION['alert']]);
        exit();
    }
} else {

    $_SESSION['alert'] = 'Incorrect login!';
    header('Content-Type: application/json');
    echo json_encode(["error" => $_SESSION['alert']]);
    exit();
}
?>

```

Б.5. Лістинг коду файлу add_user_asign.php

```

<?php
session_start();
require_once 'connect.php';

if ($_SERVER["REQUEST_METHOD"] === "POST") {

    $user_id = $_POST['user_id'];
    $projectId = isset($_POST['project-id']) ? $_POST['project-id'] : null;

    echo $projectId;

    mysqli_query($mysql, "INSERT INTO `asigns` (`id_user`, `id_project`) VALUES ($user_id, $projectId)");

    header('Location: main.php');
    exit();

}
?>

```

Б.6. Лістинг коду файлу addlist.php

```

<?php
session_start();
require_once 'connect.php';

$listName = $_POST['listname'];

```

```

$projectId = isset($_POST['project-id']) ? $_POST['project-id'] :
null;

mysqli_query($mysql, "INSERT INTO `lists` (`list_name`, `id_project`)
VALUES ('$listName', '$projectId')");

header('Location: main.php');
exit();

?>

```

Б.7. Лістинг коду файлу addproject.php

```

<?php
session_start();
require_once 'connect.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {

    if (isset($_POST['project-name'], $_POST['project-desc'])) {
        $projectName = mysqli_real_escape_string($mysql,
$_POST['project-name']);
        $projectDesc = mysqli_real_escape_string($mysql,
$_POST['project-desc']);

        $userid = mysqli_real_escape_string($mysql,
$_SESSION['user']['id']);

        mysqli_query($mysql, "INSERT INTO `projects` (`id_project`,
`project_name`, `description`) VALUES (NULL, '$projectName',
'$projectDesc')");

        $projectID = mysqli_insert_id($mysql);

        if ($projectID) {

            mysqli_query($mysql, "INSERT INTO `assigns` (`id_assign`,
`id_user`, `id_project`) VALUES (NULL, '$userid', '$projectID')");

            header('Location: main.php');
            exit();

        } else {

            echo "Failed to insert the project.";

```

```

    }

    } else {
        echo "Missing project-name or project-desc in POST request.";
    }
} else {
    echo "Invalid request method.";
}
?>

```

Б.8. Лістинг коду файлу addteam.php

```

<?php
session_start();
require_once 'connect.php';

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $teamName = $_POST['team_name'];

    $checkQuery = "SELECT COUNT(*) as count FROM `groups` WHERE
`group_name` = '$teamName'";
    $checkResult = mysqli_query($mysql, $checkQuery);
    $count = mysqli_fetch_assoc($checkResult)['count'];

    if ($count > 0) {
        $_SESSION['alert'] = 'Team with this name already exists!';
        header('Content-Type: application/json');
        echo json_encode(["error" => $_SESSION['alert']]);
        exit();
    }

    $insertQuery = "INSERT INTO `groups` (`group_name`) VALUES
('$teamName')";
    $insertResult = mysqli_query($mysql, $insertQuery);

    if ($insertResult) {
        $teamId = mysqli_insert_id($mysql);

        $userId = $_SESSION['user']['id'];
        $updateUserRoleQuery = "UPDATE `users` SET `id_role` = 3,
`id_group` = $teamId WHERE `id_user` = $userId";
        $updateUserRoleResult = mysqli_query($mysql,
$updateUserRoleQuery);

        $_SESSION['user']['groupid'] = $teamId;
        $_SESSION['user']['roleid'] = 3;
    }
}

```

```

        header('Content-Type: application/json');
        echo json_encode(["success" => true]);
        exit();
    }
}
?>

```

Б.9. Лістинг коду файлу delete_case.php

```

<?php
require_once('connect.php');

$data = json_decode(file_get_contents('php://input'), true);

$caseId = $data['caseId'];

$query = "DELETE FROM `cases` WHERE id_case = $caseId";

$result = mysqli_query($mysql, $query);

if ($result) {
    echo json_encode(['success' => true]);
} else {
    echo json_encode(['success' => false, 'error' =>
mysqli_error($mysql)]);
}

mysqli_close($mysql);
?>

```

Б.10. Лістинг коду файлу reg.php

```

<?php

session_start();
require_once 'connect.php';

$name = $_POST['name'];
$lname = $_POST['lname'];
$login = $_POST['login-reg'];
$password = $_POST['passwd-reg'];
$reppasswd = $_POST['rep-passwd-reg'];
$q = mysqli_fetch_assoc(mysqli_query($mysql, "SELECT COUNT(*) FROM
`users` WHERE `login` = '$login'"));
$check = join("", $q);

```

```

if (preg_match('/^([0-9a-zA-Z-_.]{5,16})$/', $login) || $login == "")
{
    if ($check > 0) {

        $_SESSION['reg_alert'] = 'This login is exists!';
        header('Content-Type: application/json');
        echo json_encode(["error" => $_SESSION['reg_alert']]);
        exit();

    } elseif ($passwd <> $reppasswd) {

        $_SESSION['reg_alert'] = 'Passwords do not match!';
        header('Content-Type: application/json');
        echo json_encode(["error" => $_SESSION['reg_alert']]);
        exit();

    } elseif ($login == "" || $passwd == "" || $name == "" || $lname
== "" || $reppasswd == "") {

        $_SESSION['reg_alert'] = 'Fill in all fields!';
        header('Content-Type: application/json');
        echo json_encode(["error" => $_SESSION['reg_alert']]);
        exit();

    } else {

        $pass = md5($passwd);

        $result = mysqli_query($mysql, "INSERT INTO `users`
(`id_user`, `name`, `lname`, `login`, `passwd`, `id_role`, `id_group`)
VALUES (NULL, '$name', '$lname', '$login', '$pass', '1', NULL)");

        $_SESSION['reg_alert'] = 'Registration success!';
        header('Content-Type: application/json');
        echo json_encode(["success" => $_SESSION['reg_alert']]);
        exit();

    }

} else {

    $_SESSION['reg_alert'] = 'Incorrect login!';
    header('Content-Type: application/json');
    echo json_encode(["error" => $_SESSION['reg_alert']]);
    exit();

}
?>

```

Б.11. Лістинг коду файлу save_add_data.php


```

<?php

require_once 'connect.php';

$data = json_decode(file_get_contents('php://input'), true);

$area1 = mysqli_real_escape_string($mysql, $data['area1']);
$area2 = mysqli_real_escape_string($mysql, $data['area2']);
$area3 = mysqli_real_escape_string($mysql, $data['area3']);
$area4 = mysqli_real_escape_string($mysql, $data['area4']);
$area5 = mysqli_real_escape_string($mysql, $data['area5']);
$listId = mysqli_real_escape_string($mysql, $data['listId']);

$query = "INSERT INTO `cases` (`summ`, `prec`, `str`, `er`, `comm`,
`id_status`, `id_list`)
        VALUES ('$area1', '$area2', '$area3', '$area4', '$area5', 3,
'$listId')";

$result = mysqli_query($mysql, $query);

if ($result) {

    echo json_encode(['success' => true]);
} else {

    echo json_encode(['success' => false, 'error' =>
mysqli_error($mysql)]);
}

mysqli_close($mysql);
?>

```

Б.12. Лістинг коду файлу savedate.php

```

<?php

require_once 'connect.php';

$data = json_decode(file_get_contents('php://input'), true);

if ($data && is_array($data)) {

    foreach ($data as $caseData) {
        $caseId = $caseData['caseId'];
        $summary = mysqli_real_escape_string($mysql,
$caseData['summary']);
        $prec = mysqli_real_escape_string($mysql, $caseData['prec']);
        $str = mysqli_real_escape_string($mysql, $caseData['str']);
        $er = mysqli_real_escape_string($mysql, $caseData['er']);
        $comm = mysqli_real_escape_string($mysql, $caseData['comm']);

```

```

        $statusId = $caseData['statusId'];

        $sql = "INSERT INTO `cases` (`id_case`, `summ`, `prec`, `str`,
`er`, `comm`, `id_status`, `id_list`)
            VALUES ('$caseId', '$summary', '$prec', '$str', '$er',
'$comm', '$statusId', '1')
            ON DUPLICATE KEY UPDATE
                `summ` = VALUES(`summ`), `prec` = VALUES(`prec`),
`str` = VALUES(`str`), `er` = VALUES(`er`), `comm` = VALUES(`comm`),
`id_status` = VALUES(`id_status`)";

        $result = mysqli_query($mysql, $sql);

        if (!$result) {
            $response = array('success' => false, 'error' =>
mysqli_error($mysql));
            echo json_encode($response);
            exit;
        }

        $response = array('success' => true);
        echo json_encode($response);
    } else {
        $response = array('success' => false, 'error' => 'Invalid data
received');
        echo json_encode($response);
    }
?>

```

Б.13. Лістинг коду файлу update_profile.php

```

<?php
session_start();

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $user_id = $_SESSION['user']['id'];
    $name = $_POST['name'];
    $lname = $_POST['lname'];
    $current_password = $_POST['current_password'];
    $new_password = $_POST['new_password'];
    $confirm_password = $_POST['confirm_password'];

    $stored_password_md5 = $_SESSION['user']['passwd'];

    if (!empty($current_password) && md5($current_password) !==
$stored_password_md5) {
        $_SESSION['alert'] = 'Incorrect current password!';
    }
}

```

```

    header('Content-Type: application/json');
    echo json_encode(["error" => $_SESSION['alert']]);
    exit();
}

if (!empty($new_password) || !empty($confirm_password)) {
    if ($new_password !== $confirm_password) {
        $_SESSION['alert'] = 'Passwords do not match!';
        header('Content-Type: application/json');
        echo json_encode(["error" => $_SESSION['alert']]);
        exit();
    }
}

$update_query = "UPDATE `users` SET ";

if (!empty($name)) {
    $update_query .= "`name`='{$name}', ";
    $_SESSION['user']['name'] = $name;
}

// Проверка, введена ли новая фамилия
if (!empty($lname)) {
    $update_query .= "`lname`='{$lname}', ";
    $_SESSION['user']['lname'] = $lname;
}

if (!empty($new_password) && $new_password === $confirm_password)
{
    $new_password_md5 = md5($new_password);
    $update_query .= "`passwd`='{$new_password_md5}', ";
    $_SESSION['user']['passwd'] = $new_password_md5;
}

$update_query = rtrim($update_query, ', ');

$update_query .= " WHERE `id_user`='{$user_id}'";

mysqli_query($mysql, $update_query);

header('Content-Type: application/json');
echo json_encode(["success" => true]);
exit();
}
?>

```

Б.14. Лістинг коду файлу jquery.min.php

// Бібліотека перекладача

```

var arrLang = {
  'en': {
    'parag1': 'Easy, simple, exactly!',
    'parag2': 'Manage of your testing',
    'log-in': 'Log In',
    'signup': 'Sign Up',
    'discover': 'DISCOVER',
    'design': 'DESIGN',
    'test': 'TEST',
    'welcome': 'Welcome!',
    'login': 'Login',
    'passwd': 'Password',
    'start': "Let's start!",
    'name': 'Your name',
    'lname': 'Your last name',
    'login-reg': 'Your login',
    'passwd-reg': 'Your password',
    'rep-passwd-reg': 'Repeat your password',
    'want': 'ARE YOU HAVE QUESTIONS?',
    'got': "If you have some problem with reviewing of our system",
    'connect': "Let's connect!",
    'contacts': 'Contacts',
    'logout': 'Log Out',
    'error': 'Incorrect login or password!',
    'profile': 'Profile',
    'back': 'Back',
    'ausure': 'Are you sure?',
    'cancel': 'Cancel',
    'addteammate': 'Add teammate',
    'nameteammate': "Name",
    'lnameteammate': 'Last name',
    'loginteammate': 'Login',
    'passwdteammate': 'Password',
    'reppasswdteammate': 'Reapet password',
    'create': 'Create',
    'profileadd': 'Profile edditing',
    'ed-name': 'Change name:',
    'ed-lname': 'Change last name:',
    'ed-cur-passwd': 'Change password:',
    'ed-passwd': 'New password:',
    'ed-rep-passwd': 'Repeat new password:',
    'save': 'Save',
    'add': 'Add',
    'youcancreate': 'You can create self team!',
    'ifyou': "If you're working in a team, click on 'Create team!'",
    'createteam': 'Create team!',
    'createyourteam': 'Create your team',
    'teamname': 'Team name',
  }
}

```

```

'createproject': 'Create project',
'noprojects': 'No projects yet',
'projectname': 'Project name',
'projectdesc': 'Project description',
'addtoproject': 'Assign',
'assigned': 'Assign to',
'noteammates': 'No teammates remaining',
'createlist': 'Create list',
'listname': 'List name',
'summ': 'Summary',
'prec': 'Preconditions',
'str': 'Steps to reproduce',
'er': 'Expected result',
'comm': 'Comments',
'status': 'Status',
'action': 'Action',
'addcase': 'Add case',
'delete': 'Delete'
},
'ua': {
'parag1': 'Легко, просто, точно!',
'parag2': 'Керуй своїм тестуванням',
'log-in': 'Увійти',
'signup': 'Створити',
'discover': 'ДІЗНАВАЙСЯ',
'design': 'ПРОЕКТУЙ',
'test': 'ТЕСТУЙ',
'welcome': 'З поверненням!',
'login': "Логін",
'passwd': 'Пароль',
'start': 'Привіт!',
'name': "Ваше ім'я",
'lname': 'Ваше прізвище',
'login-reg': 'Ваш логін',
'passwd-reg': 'Ваш пароль',
'rep-passwd-reg': 'Повторіть ваш пароль',
'want': 'МАСТЕ ПИТАННЯ?',
'got': 'Якщо у вас виникли проблеми з оглядом нашої системи',
'connect': "На зв'язок!",
'contacts': 'Контакти',
'logout': 'Вийти',
'error': 'Не вірний логін або пароль!',
'profile': 'Профіль',
'back': 'Назад',
'ausure': 'Ви впевнені?',
'cancel': 'Назад',
'addteammate': 'Додати робітника',
'nameteammate': "Ім'я",
'lnameteammate': 'Прізвище',

```

```

'loginteammate': 'Логін',
'passwdteammate': 'Пароль',
'repasswdteammate': 'Повторіть пароль',
'create': 'Створити',
'profileadd': 'Редагування профілю',
'ed-name': 'Змінити ім'я:',
'ed-lname': 'Змінити прізвище:',
'ed-cur-passwd': 'Актуальний пароль:',
'ed-passwd': 'Новий пароль:',
'ed-rep-passwd': 'Повторіть новий пароль:',
'save': 'Зберегти',
'add': 'Додати',
'youscancreate': 'Ви можете створити власну команду!',
'ifyou': 'Якщо ви працюєте в команді, натисніть "Створити команду!"',
'createteam' : 'Створити команду!',
'createyourteam': 'Створити свою команду',
'teamname': 'Назва команди',
'createproject': 'Створити проект',
'noprojects': 'Немає проектів',
'projectname': 'Назва проекту',
'projectdesc': 'Опис проекту',
'addtoproject': 'Призначити',
'assigned': 'Призначені',
'noteammates': 'Робітників більше немає',
'createlist': 'Створити список',
'listname': 'Назва списку',
'summ': 'Назва',
'pres': 'Передумови',
'str': 'Кроки для відтворення',
'er': 'Очікуваний результат',
'comm': 'Коментарі',
'status': 'Статус',
'action': 'Дія',
'addcase': 'Додати кейс',
'delete': 'Видалити'
}
}

$(function () {

var savedLang = sessionStorage.getItem('selectedLang');
if (savedLang) {
    setLanguage(savedLang);
}

$('.translate').click(function () {
    var lang = $(this).attr('id');

    setLanguage(lang);

```

```

    sessionStorage.setItem('selectedLang', lang);
});

function setLanguage(lang) {
    $('.lang').each(function (index, item) {
        $(this).text(arrLang[lang][$(this).attr('key')]);
    });
}
});

var textarea = document.getElementById('area');

textarea.addEventListener('keyup', function(){
    if(this.scrollTop > 0){
        this.style.height = this.scrollHeight + "px";
    }

    if(this.scrollLeft > 0){
        this.style.width = this.scrollWidth + "px";
    }
});

```

Б.15. Лістинг коду файлу update_profile.php

```

function updateForm() {
    $.ajax({
        type: "POST",
        url: "update_profile.php",
        data: $("#updateForm").serialize(),
        dataType: "json",
        success: function (response) {
            if (response.success) {
                window.location.href = "profile.php";
            } else {
                showErrorMsg(response.error);
            }
        },
        error: function (xhr, status, error) {
            console.error("AJAX Error: " + status, error);
        }
    });
}

function showErrorMsg(message) {
    var errorMsg = $("#error-msg");
    var errorText = $("#error-message");

    errorText.text(message);
}

```

```
errorMsg.show();

setTimeout(function () {
  hideErrorMsg();
}, 5000);
}

function hideErrorMsg() {
  var errorMsg = $("#error-msg");
  errorMsg.hide();
}
```


ДОДАТОК В

В.1 Посилання на web-систему

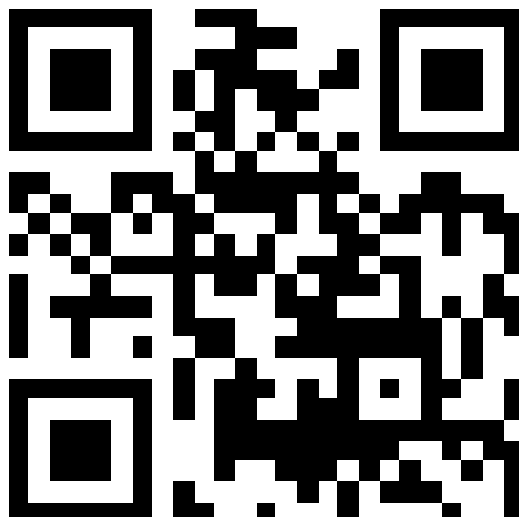


Рисунок В.1 – QR-код з посилання на «Saber»

В.2 Посилання на відео-демонстрацію web-системи



Рисунок В.2 – QR-код з посилання на відео-демонстрацію «Saber»