

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК
СЕКЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРОЕКТУВАННЯ

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «Web-додаток визначення схильності до конфліктної поведінки»

за спеціальністю 122 «Комп'ютерні науки»,
освітньо-професійна програма «Інформаційні технології проектування»

Виконавець роботи: студент групи ІТ-72/8 Отенко Дмитрій Сегійович

**Кваліфікаційна робота бакалавра
захищена на засіданні ЕК
зоцінкою**

_____ «» червня 2021 р.

Науковий керівник

(підпис)

к.т.н., доц., Парфененко Ю. В.

(науковий ступінь, вчене звання, прізвища ініціали)

Голова комісії

(підпис)

Шифрін Д. М.

(науковий ступінь, вчене звання, прізвища ініціали)

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

Студент _____
(підпис)

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

ЗАТВЕРДЖУЮ

Зав. секцією ІТП

_____ В. В. Шендрик
«__» _____ 2021р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА СТУДЕНТУ**

Отенко Дмитрій Сергійович

- 1 Тема проекту** Web-додаток визначення схильності до конфліктної поведінки
керівник роботи Парфененко Юлія Вікторівна, к.т.н., доцент,
затверджена наказом по університету від «14» квітня 2021 р. №0181-VI
- 2 Строк подання студентом роботи** «05» червня 2021 р.
- 3 Вхідні дані до проекту** Технічне завдання на розробку web-додатку, контент
- 4 Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)** аналіз предметної області, проектування web-додатку визначення схильності до конфліктної поведінки, розробка проектування web-додатку визначення схильності до конфліктної поведінки
- 5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)** актуальність, дослідження аналогів, мета та задачі, функціональні вимоги до web-додатку, засоби реалізації, моделювання роботи web-додатку, діаграма варіантів використання, база даних, архітектура web-додатку, приклади реалізації web-додатку, висновки

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 випускної проекту	Термін виконання етапів проекту	Примітка
1	Аналіз предметної області	До 01.03.2021	
2	Аналіз конкурентів	До 02.04.2021	
3	Постановка задачі та вибір методів розробки	До 07.04.2021	
4	Розробка дизайну	До 04.05.2021	
5	Проектування бази даних	До 12.05.2021	
6	Розробка функціональних блоків	До 20.05.2021	
7	Тестування	До 31.05.2021	

Студент _____

Отенко Д.С.

Керівник роботи _____

к.т.н., доц. Парфененко Ю.В.

РЕФЕРАТ

Тема кваліфікаційної роботи бакалавра «Web-додаток визначення схильності до конфліктної поведінки».

Метою проекту є розробка web-додатку як повноцінної платформи для створення тестових завдань, проведення тестувань задля встановлення стану користувачів та подальша робота із зібраними даними.

У web-додатку визначення схильності до конфліктної поведінки буде реалізовано тестування за ігровою методикою. За необхідності викладач зможе додати звичайний тест.

В першому розділі проведено аналіз обраної предметної області. У цьому розділі виконано: огляд актуальних досліджень та публікацій, аналіз аналогів, постановку задачі.

В другому розділі було сформовано мету та задачі дослідження. Даний етап також включає в себе вибір засобів реалізації. В даному розділі описано етап проектування web-додатку. Було виконано побудову діаграм в нотації IDEF0, діаграми варіантів використання web-додатку визначення схильності до конфліктної поведінки та спроектована база даних.

В останньому, третьому розділі, було виконано реалізацію та детальний опис використання web-додатку зі сторони користувачів та адміністратора.

Результатом проведеної роботи є розроблений web-додаток визначення схильності до конфліктної поведінки.

Кваліфікаційна робота містить 94 сторінки, 8 таблиць, 46 рисунків, список літератури з 24 найменувань, 3 додатки.

Ключові слова: web-додаток, сервіс, товар, учень, викладач, адміністрування, тестування, гра, агресія, додаток, звіт, робота, проект.

ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Огляд застосування інформаційних технологій при проведенні психологічних досліджень.....	7
1.2 Аналіз web-додатків по тестуванню.....	8
1.3 Постановка задачі.....	12
2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-ДОДАТКУ	15
2.1 Моделювання процесу визначення схильності до конфліктної поведінки з використанням web-додатку.....	15
2.2 Моделювання варіантів використання web-додатку	16
2.3. Проектування бази даних web-додатку.....	18
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ	21
3.1 Архітектура web-додатку	21
3.2 Програмна реалізація	22
3.3 Використання web-додатку зі сторони користувача	25
3.4 Адміністрування сайту.....	32
ВИСНОВКИ	38
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	39
ДОДАТОК А	43
ДОДАТОК Б.....	54
ДОДАТОК В.....	68

ВСТУП

Гнів та агресія – це складні проблеми. Агресія стосується будь-якої поведінки, спрямованої на іншу особу, яка здійснюється з безпосереднім наміром заподіяти шкоду. Для виявлення схильності до агресивної поведінки є декілька методів, а саме тестування та консультація з фаховим спеціалістом. Використання інформаційних технологій при проведенні психологічних тестувань дозволяє спростити процес проходження та обробки тестів.

Метою кваліфікаційної роботи бакалавра є розробка web-додатку для визначення схильності людини до конфліктної поведінки. Для реалізації поставленої мети буде потрібно вирішити такі задачі дослідження:

- аналіз предметної області;
- формування технічного завдання та розробка прототипу web-додатку
- моделювання роботи web-додатку;
- програмна реалізація web-додатку та тестування.

Створений web-додаток орієнтований на розробку повноцінної системи тестування для виявлення схильності до конфліктної поведінки. Web-додаток буде розроблений для зручного використання та експлуатації вчителями, для розробки тестів та проведення тестування. Використання даного web-додатку не буде залежати від локації користувача чи того, хто проводить тестування.

Практичне значення web-додатку полягає у його використанні як повноцінної платформи для створення тестових завдань, проведення тестувань задля встановлення стану користувачів та подальшої роботи із зібраними даними.

Web-додаток по виявленню схильності до конфліктної поведінки після реалізації повинен бути викладений на сервер для перегляду, тестування та користування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд застосування інформаційних технологій при проведенні психологічних досліджень

Можна виділити кілька рівнів залежно від рівня включення комп'ютерних технологій у процедуру психологічного дослідження. Високий рівень комп'ютеризації методів передбачає вибір методів, підготовку психологічних висновків та застосування комп'ютерів до процесу змістовної інтерпретації результатів тесту. Середній рівень комп'ютеризації включає участь комп'ютера в тесті. Комп'ютеризація методів низького рівня включає зберігання даних обстеження та обробку первинних результатів діагностики, наприклад, обчислення показників тесту [1].

Розвиток технології комп'ютеризації високого рівня в даний час представляє значні наукові проблеми. Існуючі комп'ютерні алгоритми для побудови психологічних висновків ще не рекомендуються для широкомасштабного використання, з особливим акцентом на аналізі багатofакторних методів [2].

Традиційно цю групу методів називали комп'ютерним та апаратним тестуванням. За допомогою комп'ютерів можна моделювати різноманітні складні види діяльності, приводячи таким чином модель з точки зору складності та динаміки обставин, що змінюються, і лише результатів [3-5]. Ця якість особливо важлива для професійних методів психологічної діагностики, які орієнтуються на конкретні критерії.

Початковий рівень комп'ютеризації методів передбачає залучення комп'ютерних технологій до обробки та зберігання результатів психологічних досліджень. Додаткова стандартизація такої комп'ютеризованої технології не потрібна, оскільки комп'ютер не використовується в процедурі тестування.

У більшості випадків комп'ютерні методи психологічної діагностики - це автоматичні тести та анкети.

Сьогодні автоматизація психологічної діагностики базується на ідеї формалізації психологічних явищ та їх вимірювального потенціалу. У зв'язку з цим, є психологічні симптоми, які є найбільш корисними для автоматизації [6].

Оцінюючи широту та глибину гніву чи агресії дитини, викладач або психолог може розглянути поведінку дитини не лише в певній ситуації. Це включає отримання інформації від батьків та вчителів, перегляд академічних, медичних та поведінкових записів, проведення індивідуальних співбесід з дитиною та батьками та безпосереднє проведення тестування [7].

Отже, для більш зручного проведення тестування буде розроблено web-додаток. Його використання також полегшить аналіз зібраних даних.

1.2 Аналіз web-додатків по тестуванню

Для проведення тестувань у психології застосовуються різні типи додатків – настільні, мобільні, web-додатки [8-10]. Для проведення аналізу було знайдено веб-сторінки за схожою тематикою. Вже при пошуку виникла проблема – відсутність аналогів web-додатків проведення психологічних тестувань. Всі приклади складають з себе інформаційні сторінки й не більше, на яких користувачу відображається тест, але немає можливості пройти тестування в автоматичному режимі.

Було розглянуто web-додаток «OppositionalDefiantDisorder». Для проходження тесту не потрібно реєструватися. Це також говорить про те, що результати не зберігаються, а одноразово з'являються на сторінці.

На рис.1.1 представлено зображення сторінки екрану з тестами. З першого погляду зрозуміло, що на даному web-додатоку не варто очікувати цікавих ефектів та яскравої подачі для школярів та дітей.

Даний приклад є ілюстрацією стандартного типу тестування з вибором однієї відповіді з декількох можливих варіантів.

Наступний приклад, web-додаток «IDRlabs.com», відрізняється від інших саме наявністю функціонала для проходження тестування (рис.1.3). Навігація на сайті є досить зручною для використання.

При розробці web-додатку варто пам'ятати про зручність навігації сайту. Можливо, потенційний відвідувач сайту зацікавиться послугами і матиме бажання пройти тестування але через відсутність правильної системи навігації людина може не знайти потрібну інформацію, наприклад, сторінку з тестом, який його цікавить.



IDRlabs.com
INDIVIDUAL DIFFERENCES RESEARCH

TESTS TYPES ARTICLES NEWS MEMBERS SEARCH Q

Aggression Test

Anger, aggression, and irritation are complex emotions and are still not fully mapped out by psychology. This Aggression Spectrum Test represents a fusion of several prior efforts to research aggression and measures your aggressiveness profile across 8 different domains.

What is your aggressiveness profile like? For each of the following questions, indicate your level of agreement below.

Question 1 of 40

I spend a lot of time thinking about whether something is really another person's fault before I take my frustrations out on them.

Disagree Agree

NEXT >

Why Use This Test?

1. **Free.** This aggression spectrum test is delivered to you free of charge and will allow you to

Рисунок 1.3 – Головна сторінка сайту «IDRlabs.com»

Проаналізувавши веб-сторінки по тестуванню та виявлення схильності до конфліктної поведінки, було проведено порівняльний аналіз (табл.1.1), де було виявлено всі позитивні та негативні сторони розглянутих web-додатків. Дані пункти будуть враховані при створенні web-додатку в рамках виконання кваліфікаційної роботи бакалавра.

До недоліків можна віднести:

- не лаконічне розподілення інформації на сайті;
- примітивний дизайн сторінок тестування;

- відсутність функціоналу для збереження;
- До переваг сайтів-аналогів можна віднести:
- розподіл інформації по категоріям;
 - виведення результатів тестування на сторінку;
 - зручність навігації на сайті;
 - додаткові інформаційні блоки.

Таблиця 1.1 – Аналіз web-додатків тестування

Критерії	Oppositional DefiantDisorder	Саморозвиток	IDRlabs.com	Власний web- додаток
Відсутність реклами	–	+	+	+
Розподіл інформації по категоріям	+	–	+	+
Виведення результатів тестування	–	–	+	+
Зручність навігації	+	+	+	+
Лаконічний дизайн	+	+	–	+
Додаткові інформаційні блоки	+	+	–	+

Отже, web-додаток повинен мати зручну навігацію, лаконічний дизайн та реалізовувати увесь спектр функцій тестування. Особливу увагу потрібно звернути на збереження результатів тестування, реєстрацію як вчителю чи учню та можливість створення власного тестування.

1.3 Постановка задачі

Метою проекту є створення web-додатку як повноцінної платформи для створення тестових завдань, проведення тестувань задля встановлення емоційного стану користувачів та подальша робота із зібраними даними.

У web-додатку визначення схильності до конфліктної поведінки буде реалізовано тестування за ігровою методикою. За необхідності викладач зможе додати інший тест.

Система матиме декілька типів користувачів системи, а саме – адміністратор, викладач та учень. При реєстрації можна обрати між категоріями користувачів «Викладач» та «Учень». Управління контентом буде вестися через адміністративну панель. До контенту відноситься усе, що пов'язане з тематикою роботи.

Функціонал користувача типу «Учень»:

- проходження тестування;
- перегляд історії пройдених тестів та результати;
- виконувати фільтрацію та пошук на сайті.

Функціонал користувача типу «Викладач»:

- створення тесту за допомогою конструктора;
- перегляд результатів тестування за створеним тестом;
- налаштування створеного тесту;
- виконувати фільтрацію та пошук на сайті.

При розробці тестування користувач матиме можливість створити 4 етапи, а саме – відео, тест, рекомендації та залишення коментарю про тестування. Тести будуть розроблені у ігровій формі, що допоможе школярам не перебувати у нервовому стані та з легкістю відповісти на всі запитання. До кожного запитання можна додати декілька зображень чи відео.

Отже, цільова аудиторія web-додатку:

- вчителі;
- шкільні психологи;

- учні та діти до 18 років;

Структура web-додатку по виявленню схильності до конфліктної поведінки являтиме собою набір сторінок:

- Головна сторінка;
- Реєстрація;
- Авторизація;
- Тестування;
- Створення тесту;
- Налаштування створеного тесту;
- Налаштування аккаунту;

Розглянемо етапи розробки web-додатку. Перший етап, веб-розробка інтерфейсу – це етап, який реалізує веб-дизайн за допомогою HTML, CSS та JavaScript. Для роботи над проектом було обрано фреймворк VueJS.

VueJS - це вдосконалений фреймворк JavaScript, який використовується для створення користувацького інтерфейсу [11]. На відміну від багатьох аналогів, VueJS розроблений з нуля, щоб його можна було поступово застосовувати. Також даний фреймворк досить легкий та зрозумілий у використанні.

Другим етапом є розробка функціональної частини web-додатку. Це серверна сторона розробки, де в першу чергу розробка зосереджена на тому, як працює web-додаток [12-14].

Для розробки даної частини було обрано мову програмування PHP, а саме фреймворк Laravel. Обраний фреймворк є одним із популярних веб-фреймворків нового покоління.

Цей фреймворк було обрано за результатами аналізу можливості його використання з іншим популярним фреймворком VueJS.

Перелік головних переваг та умов використання розглянутих фреймворків представлений у табл.1.2.

Таблиця 1.2 – Перелік переваг використання VueJS та Laravel

№	Назва	Опис
1	Все відбувається на візуальній частині	VueJS дозволяє розробникам Laravel створювати інтерфейс таким чином, що їх додатку не доведеться перезавантажувати сторінки при кожному оновленні.
2	Додаток на одній сторінці	Повні ресурси програми завантажуються один раз, і більша частина їх кешується. Оскільки користувач взаємодіє з вашим додатком, все, що він робить, - це запит даних, які зазвичай потребують низької пропускну здатності [15].
3	Легкий у вивченні та використанні	VueJS досить легко використовувати. Як розробник, він надає дуже мало варіантів і має багато абстрактних матеріалів. Використовуючи VueJS можна створити простий додаток із простим JavaScript, який залишається чинним у VueJS.
4	Створення оптимальних складних інтерфейсних сторінок	Коли додаток розробляється з компонентами VueJS, залежності кожного компонента під час його відтворення відстежуються автоматично. Через це система точно знає, який компонент потрібно актуалізувати насправді при зміні даних.

Програмна частина, реалізована з Laravel, слугуватиме повноцінним функціональним ядром web-додатку, тоді як використання VueJS забезпечить інтерактивний користувальницький інтерфейс для web-додатку[15-19].

Отже, було вирішено використовувати VueJS та Laravel для розробки web-додатку, орієнтованого на проведення тестування для виявлення схильності до конфліктної поведінки.

2 МОДЕЛЮВАННЯ ТА ПРОЕКТУВАННЯ WEB-ДОДАТКА

2.1 Моделювання процесу визначення схильності до конфліктної поведінки з використанням web-додатку

Проведено моделювання процесу визначення схильності до конфліктної поведінки в нотації IDEF0 [20-22]. Діаграма нульового рівня відображена на рис.2.1. Дана діаграма відображає всі дані та вхідну інформацію, яка використовується в web-додатку по тестуванню учнів та дітей.



Рисунок 2.1 – Контекстна діаграма в нотації IDEF0

Діаграма декомпозиції в нотації IDEF0 із детальним розкриттям поетапності для даного проекту з точки зору користувача, який проходить тестування, представлена на рис. 2.2.

Декомпозиція виконує розбиття процесу на такі функціональні блоки: «Авторизація у системі», «Проведення тестування», «Формування результатів».

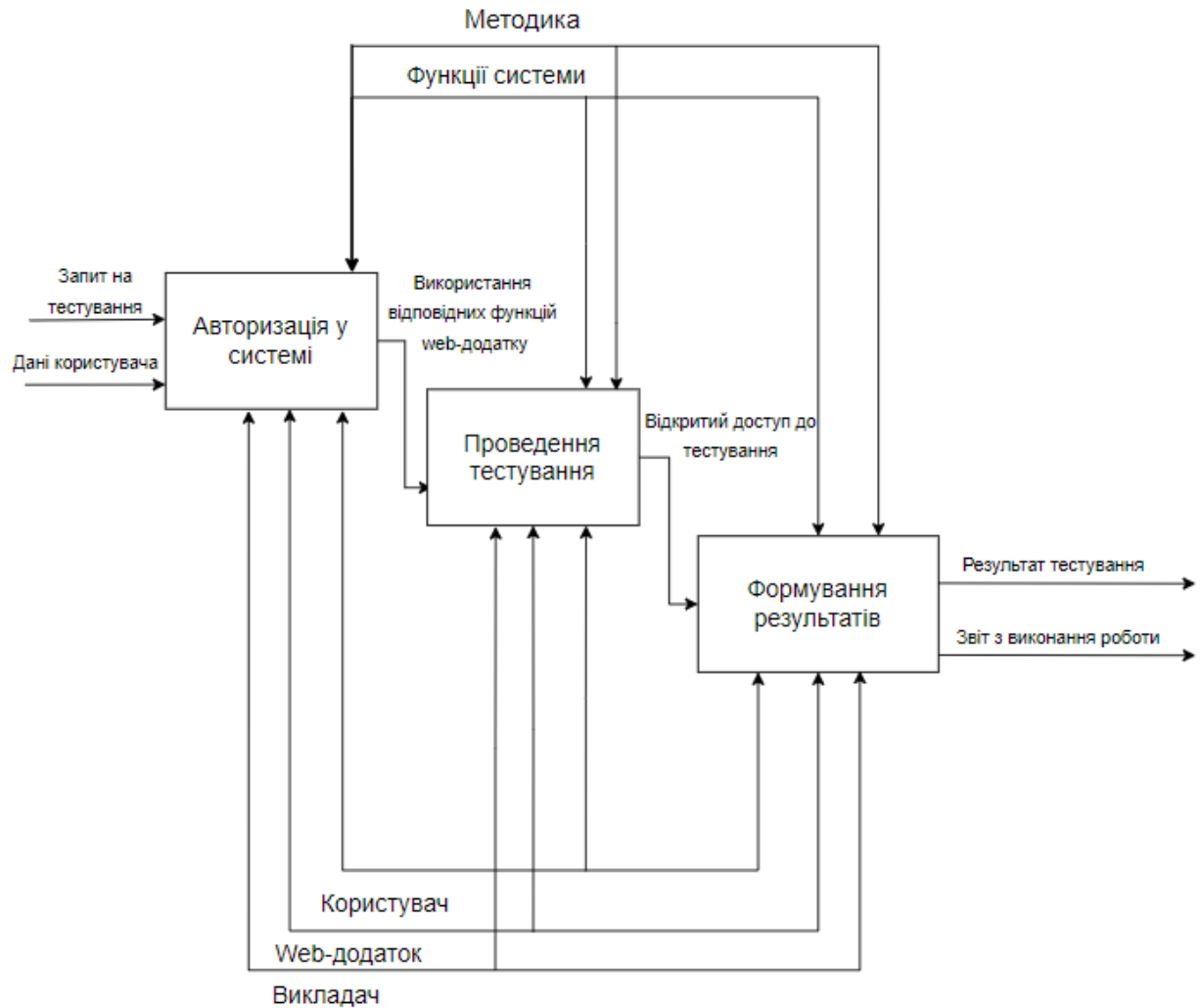


Рисунок 2.2 – Декомпозиція контекстної діаграми в нотатції IDEF0

2.2 Моделювання варіантів використання web-додатку

Представлено інформацію про акторів розроблюваного web-додатку (табл.2.1) та опис варіантів використання (табл.2.2)[23-24].

Таблиця 2.1 – Опис акторів

№	Назва	Опис
1	Адміністратор	Адміністратор має можливість редагувати інформацію.
2	Учень	Користувач із можливістю проходження тестування.
3	Викладач	Користувач із можливістю створення тестів.

Таблиця 2.2 – Опис варіантів використання

№	Назва	Опис
1	Авторизація	Функція входу в систему.
2	Реєстрація	Новий користувач має можливість реєстрації у системі
3	Перегляд новин та загальної інформації	Перегляд користувачами інформації на сайті
4	Редагування даних на сторінці	Можливість редагування користувача власних даних
5	Редагування інформації на сайті	Адміністратор має можливість редагувати дані на сайті.
6	Додавання тестування	Користувач із роллю «Викладач» має можливість створення тестів
7	Проходження тестування	Користувач із роллю «Учень» має можливість проходження відкритих тестів

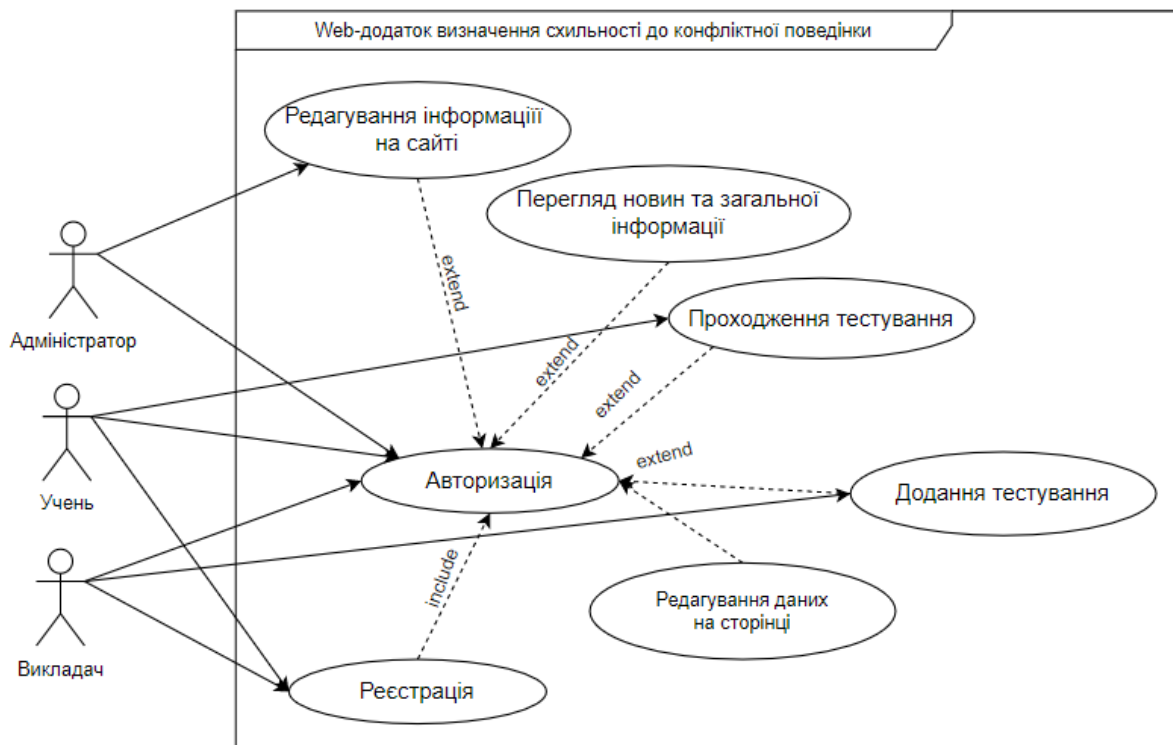


Рисунок 2.3 – Діаграма варіантів використання

2.3. Проектування бази даних web-додатку

Розроблена база даних web-додатку показана в ER-діаграмі (рис.2.4). Проаналізувавши сутність, використані в моделі web-додатку, представлена реалізація структури бази даних. Детальна інформація про кожну із таблиць знаходяться в табл.2.3.

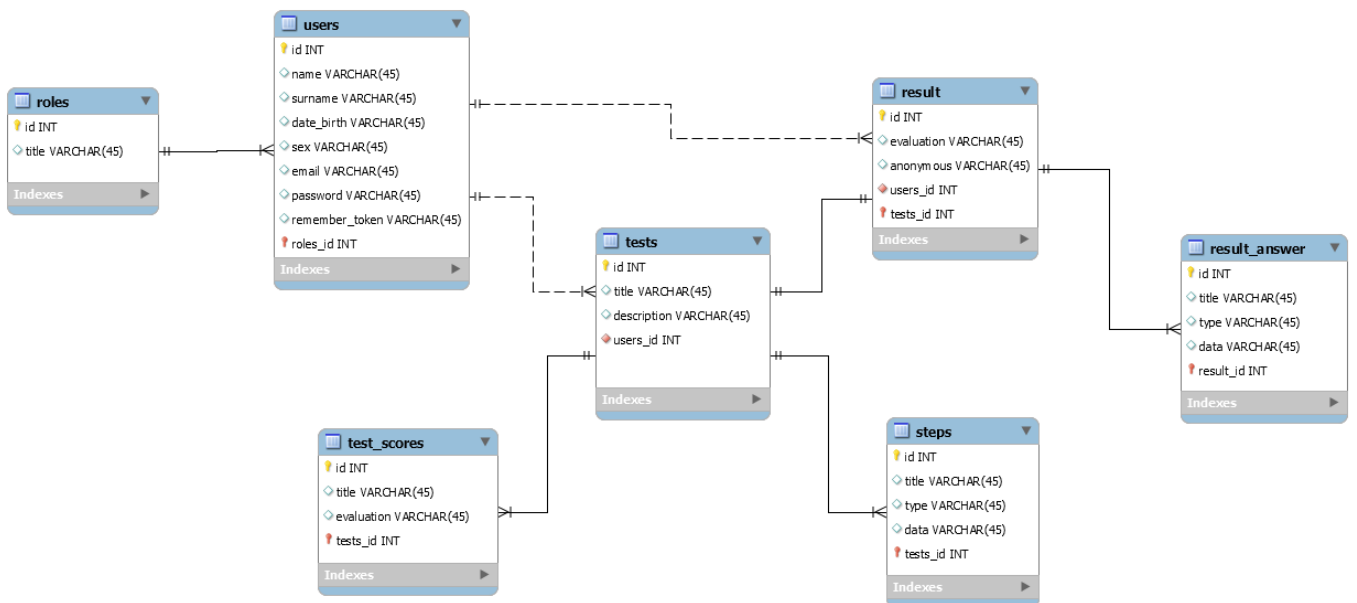


Рисунок 2.4 – ER-діаграма веб-додатку

Таблиця 2.3 – Опис таблицьбази даних

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
1	role	id	Ідентифікатор ролі користувача	INTEGER	PK	Не пустий
		title	Назва ролі користувача	VARCHAR(45)		Не пустий
2	rusers	id	Ідентифікаторкористувача	INTEGER	PK	Не пустий
		name	Ім'я користувача	VARCHAR(45)		Не пустий
		surname	Прізвище користувача	VARCHAR(45)		Не пустий
		date	Дата народження користувача	VARCHAR(45)		Не пустий
		sex	Стать користувача	VARCHAR(45)		
		email	Електронна адреса користувача	VARCHAR(45)		Не пустий
		password	Пароль користувача	VARCHAR(45)		Не пустий
		remember_token	Токет	VARCHAR(45)		
		role_id	Ідентифікатор ролі користувача	INTEGER	FK	Не пустий
3	tests	id	Ідентифікатор тесту	INTEGER	PK	Не пустий
		title	Назва тесту	VARCHAR(45)		Не пустий
		description	Опис тесту	VARCHAR(45)		Не пустий
		user_id	Ідентифікатор користувача	INTEGER	FK	Не пустий
4	steps	id	Ідентифікатор кроку	INTEGER	PK	Не пустий

Продовження таблиці 2.3

№	Таблиця	Поле	Зміст	Тип	Ключі	Обмеження
	steps	title	Назва кроку	VARCHAR(45)		Не пустий
		type	Тип кроку	VARCHAR(45)		Не пустий
		date	Дані створеного кроку	VARCHAR(45)		Не пустий
		test_id	Ідентифікатор тесту	INTEGER	FK	Не пустий
5	result	id	Ідентифікатор результату	INTEGER	PK	Не пустий
		evaluation	Загальна оцінка за тест	VARCHAR(45)		Не пустий
		anon	Тип проходження	VARCHAR(45)		
		user_id	Ідентифікатор користувача	INTEGER	FK	Не пустий
		test_id	Ідентифікатор тесту	INTEGER	FK	Не пустий
6	result_ansew	id	Ідентифікатор відповіді	INTEGER	PK	Не пустий
		title	Відповідь	VARCHAR(45)		Не пустий
		type	Тип відповіді	VARCHAR(45)		Не пустий
		date	Вага відповіді	VARCHAR(45)		Не пустий
		resulr_id	Ідентифікатор результату	INTEGER	FK	Не пустий

ЗПРАКТИЧНА РЕАЛІЗАЦІЯ WEB-ДОДАТКУ

3.1 Архітектура web-додатку

При розробці web-додатку визначення схильності до конфліктної поведінки було спроектовано архітектуру за шаблоном MVC (рис.3.1).

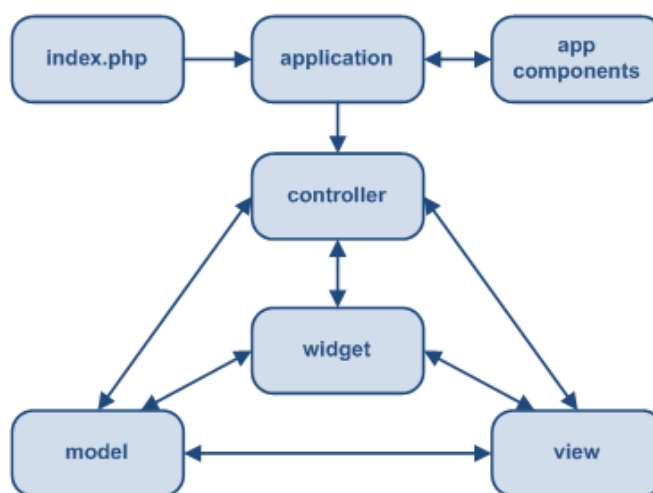


Рисунок 3.1 – Структура шаблону MVC

Web-додаток, реалізований на основі шаблону MVC, складається з трьох основних компонентів:

- модель – відповідає за розроблені об'єкти бази даних та призначена для роботи із контролером;
- контролер – виконує взаємодію з моделлю та представленням;
- представлення – частина для роботи із користувачем та отримання потрібних даних.

Дані модулі є незалежними один від одного, що можуть бути використані окремо, в той же час залежать один від одного при використанні web-додатку.

Життєвий цикл web-додатку на MVC починається з користувача, який входить на початкову сторінку web-додатку визначення схильності до конфліктної поведінки з певними параметрами в своєму браузері. Користувач повідомляє, виходячи із заданих параметрів, який тип сторінки потрібно відобразити.

3.2 Програмна реалізація

Структура програми Laravel за замовчуванням, має на меті забезпечити єдину точку входу для web-додатку. Маршрутизацію налаштовано відповідно до логіки роботи web-додатку.

Було розроблено базу даних web-додатку. Для цього був використаний додаток phpMyAdmin. Представлені створені таблиці на рис.3.2-8.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT
2	result_id	bigint(20)		UNSIGNED	Нет	Нет		
3	title	varchar(255)	utf8mb4_unicode_ci		Да	NULL		
4	type	varchar(255)	utf8mb4_unicode_ci		Нет	Нет		
5	data	json			Нет	Нет		

Рисунок 3.2 – Таблица «result_answer»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT
2	test_id	bigint(20)		UNSIGNED	Нет	Нет		
3	user_id	bigint(20)		UNSIGNED	Нет	Нет		
4	evaluation	int(11)			Нет	Нет		
5	anonymous	tinyint(1)			Нет	0		
6	created_at	timestamp			Да	NULL		

Рисунок 3.3 – Таблица «result»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🔑	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT
2	title	varchar(255)	utf8mb4_unicode_ci		Нет	Нет		

Рисунок 3.4 – Таблица «roles»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🔑	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT
2	test_id 🔑	bigint(20)		UNSIGNED	Нет	Нет		
3	title	varchar(255)	utf8mb4_unicode_ci		Да	NULL		
4	type	varchar(255)	utf8mb4_unicode_ci		Нет	Нет		
5	data	json			Нет	Нет		

Рисунок 3.5 – Таблица «steps»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🔑	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT
2	test_id 🔑	bigint(20)		UNSIGNED	Нет	Нет		
3	title	text	utf8mb4_unicode_ci		Нет	Нет		
4	evaluation	int(11)			Нет	Нет		

Рисунок 3.6 – Таблица «test_scores»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🔑	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT
2	user_id 🔑	bigint(20)		UNSIGNED	Нет	Нет		
3	title	text	utf8mb4_unicode_ci		Нет	Нет		
4	description	text	utf8mb4_unicode_ci		Нет	Нет		
5	created_at	timestamp			Да	NULL		

Рисунок 3.7 – Таблица «tests»

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🔑	bigint(20)		UNSIGNED	Нет	Нет		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_unicode_ci		Нет	Нет		
3	surname	varchar(255)	utf8mb4_unicode_ci		Нет	Нет		
4	date_birth	varchar(255)	utf8mb4_unicode_ci		Нет	Нет		
5	sex	tinyint(1)			Нет	Нет		
6	photo	varchar(255)	utf8mb4_unicode_ci		Нет	/img/no-image.png		
7	email 🔑	varchar(255)	utf8mb4_unicode_ci		Нет	Нет		
8	password	varchar(255)	utf8mb4_unicode_ci		Нет	Нет		
9	role_id 🔑	bigint(20)		UNSIGNED	Нет	Нет		
10	remember_token	varchar(100)	utf8mb4_unicode_ci		Да	NULL		
11	created_at	timestamp			Да	NULL		

Рисунок 3.8 – Таблица «users»

Файлова структура розробленого web-додатку по визначення схильності до конфліктної поведінки представлено на рис.3.9.

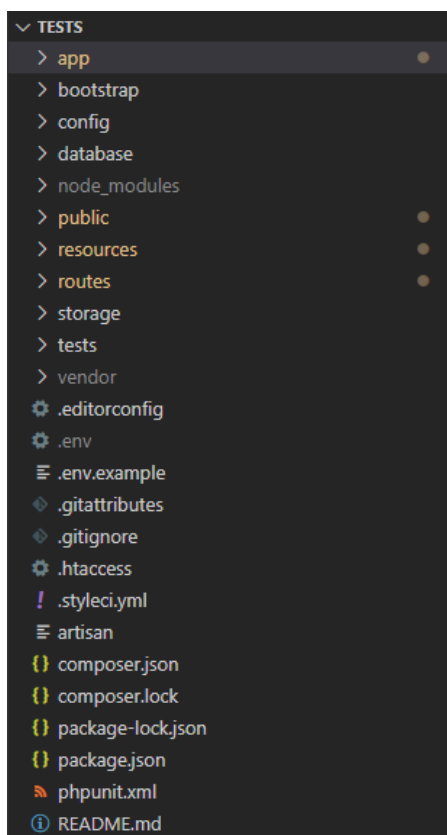


Рисунок 3.9 –Файлова структура web-додатку

3.3 Використання web-додатку зі сторони користувача

Для початку користування web-додатком, а саме проходження тесту у виді гри, в якості користувача необхідно зареєструватись або пройти авторизацію, якщо вже акаунт існує (рис.3.10-3.12). Реєструватись можна в якості вчителя або як учень.

ВІТАЄМО!

Вже зареєстрований? [Вхід](#)

Ім'я * Прізвище *

Дата народження *

Стать *

Чоловіча Жіноча

Тип користувача *

Викладач Учень

Пошта

Пароль

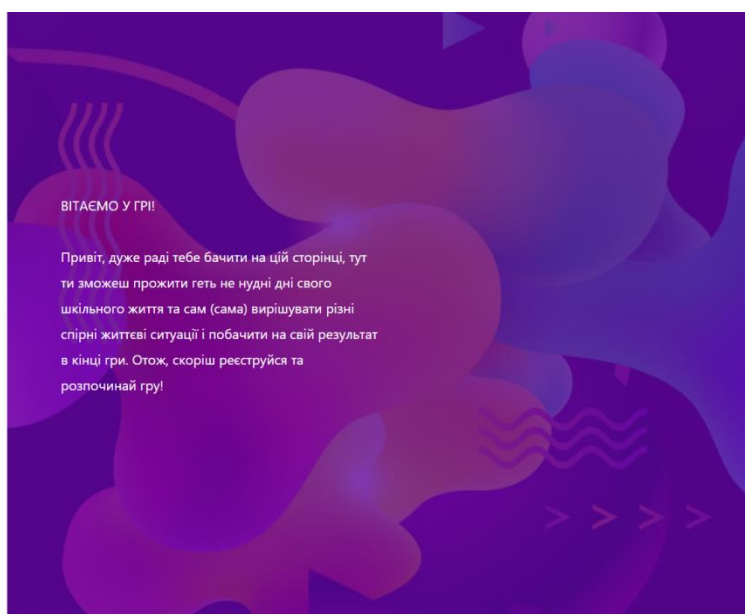


Рисунок 3.10 – Реєстрація

ВІТАЄМО!

Ще не зареєстрований? [Зареєструватися](#)

Пошта

Пароль

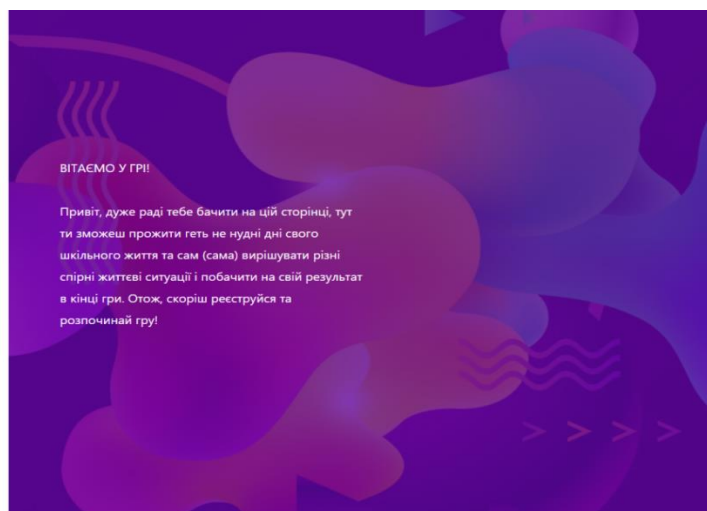


Рисунок 3.11 – Авторизація

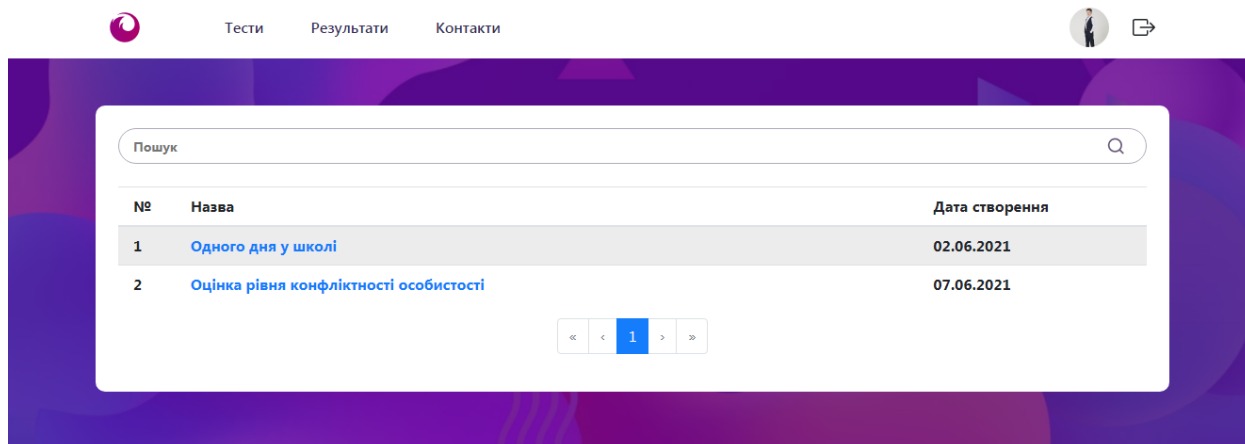


Рисунок 3.12 – Авторизація в ролі учня

Проходження тесту відбувається в декілька етапів. Першим етапом є відображення сторінки, яка передбачає перегляд відео-запрошення до проходження тесту у виді "гри" (рис.3.13).

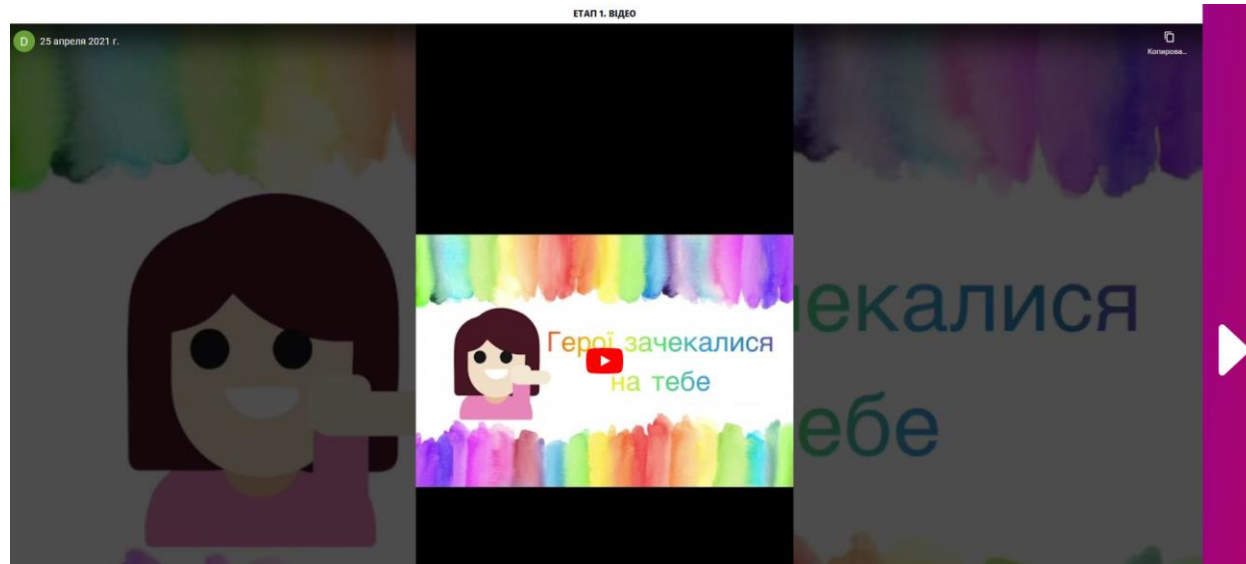


Рисунок 3.13 – Перший етап

Другий етап – це, власне, сама гра-тест в якій є декілька питань з варіантами відповідей, за кожен з яких зараховується певна кількість балів (рис.3.14).

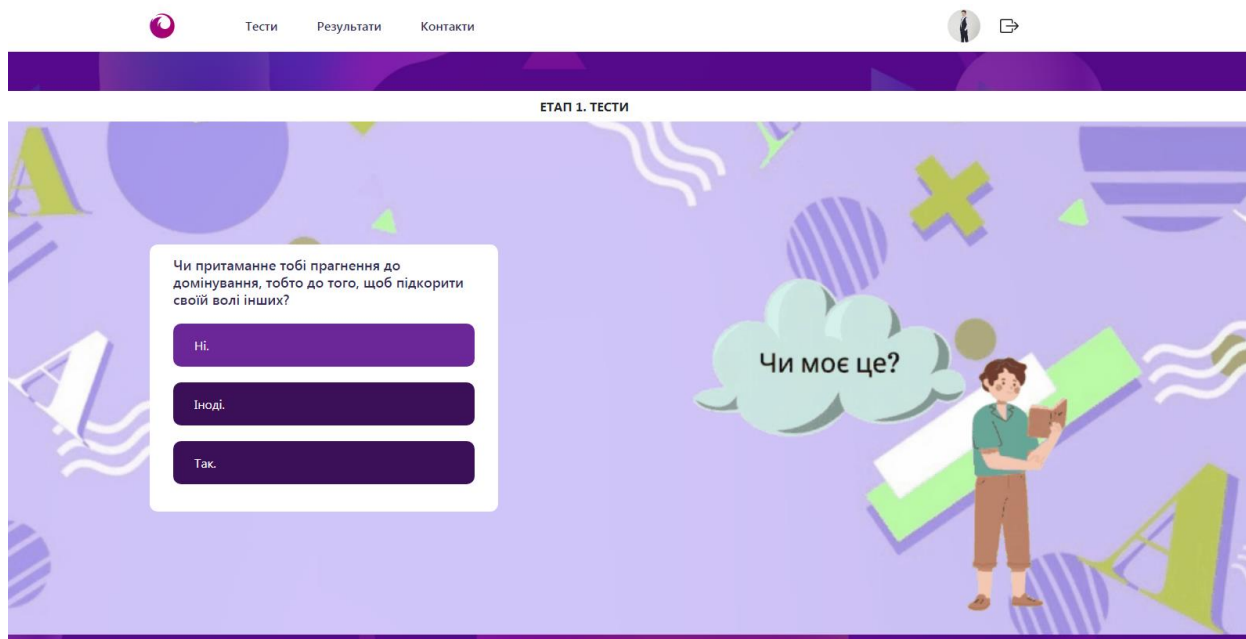


Рисунок 3.14 – Другий етап

Після проходження тесту можна переглянути результати. На екран виводиться кількість набраних балів та висновок згідно результату (рис.3.15).

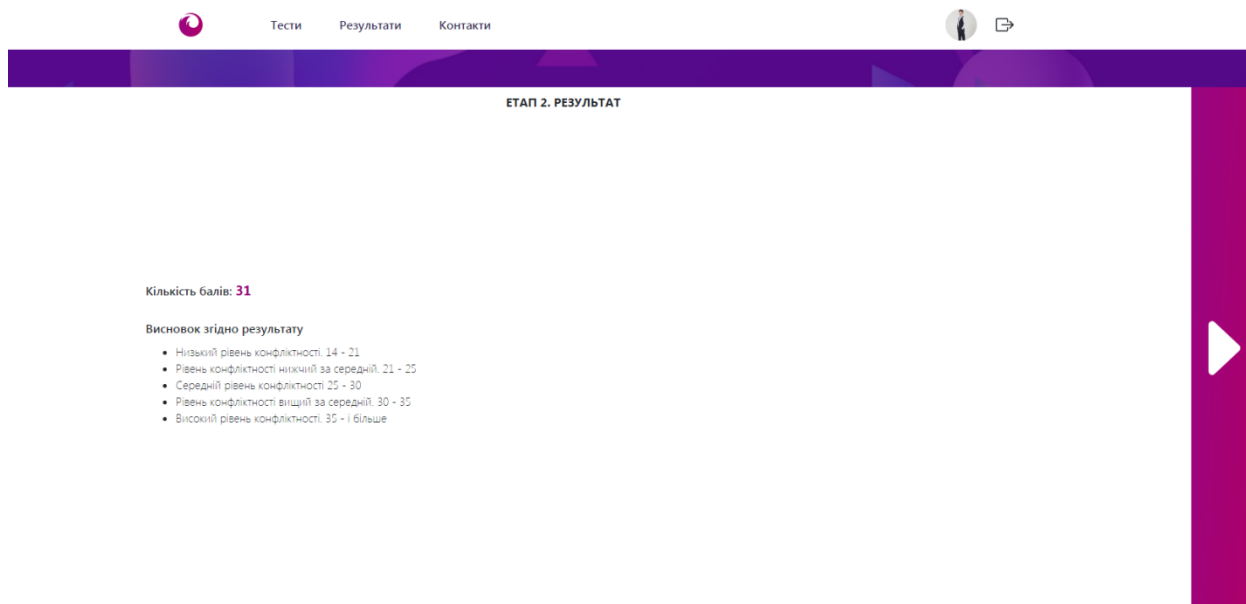


Рисунок 3.15 – Третій етап

Наступний етап – рекомендації з приводу того, як необхідно поводити себе в стресових ситуаціях, під час спілкування з іншими людьми (рис.3.16).

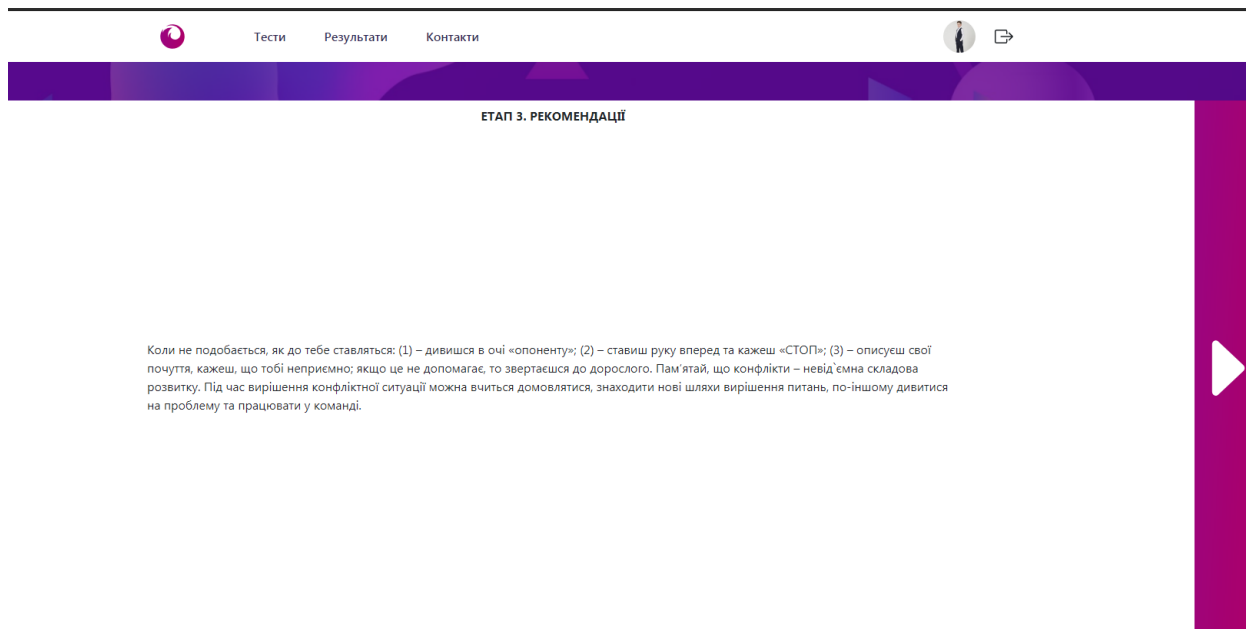


Рисунок 3.16 – Етап чотири

Після рекомендацій надаються ще декілька питань для винесення власних висновків про свої результати та можливість оцінити гру (рис.3.17).

Рисунок 3.17 – П'ятий етап

Переглянути результати пройдених тестів можна на особистій сторінці (рис.3.18).

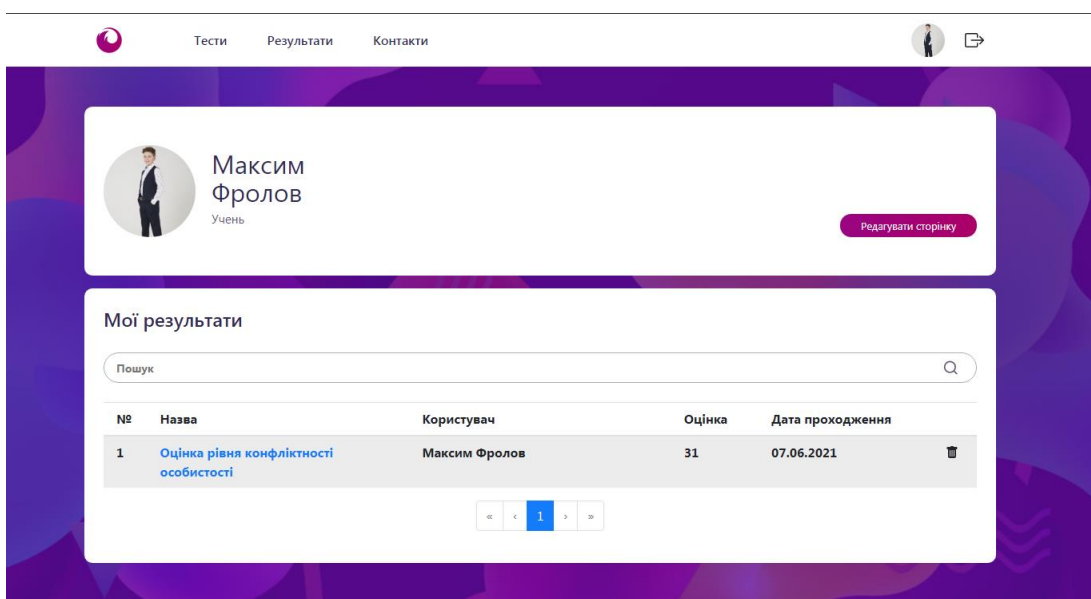


Рисунок 3.18 – Особиста сторінка учня

Натиснувши на назву тесту, можна переглянути детальну інформацію про пройдені тести, кількість відповідей та їх зміст, свою оцінку тесту та кількість набраних балів (рис.3.19).

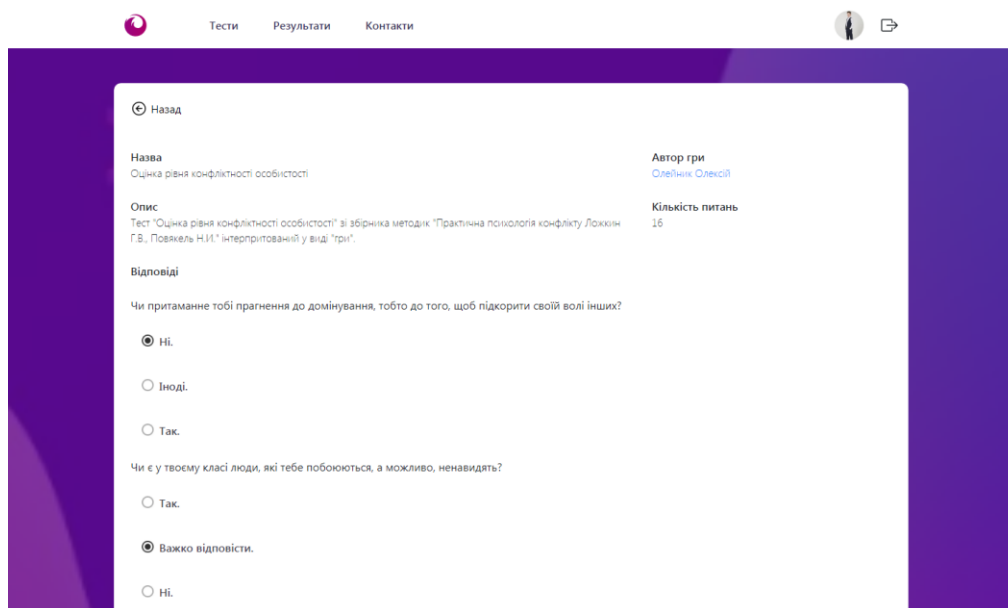


Рисунок 3.19 – Інформація про тест

В особистому кабінеті користувач може змінити інформацію про себе. Змінити ім'я, прізвище, додати фотографію та дату народження, вказати стать та прив'язати сторінку до пошти. При завантаженні фото його можна автоматично кадрувати. (рис.3.20-23).

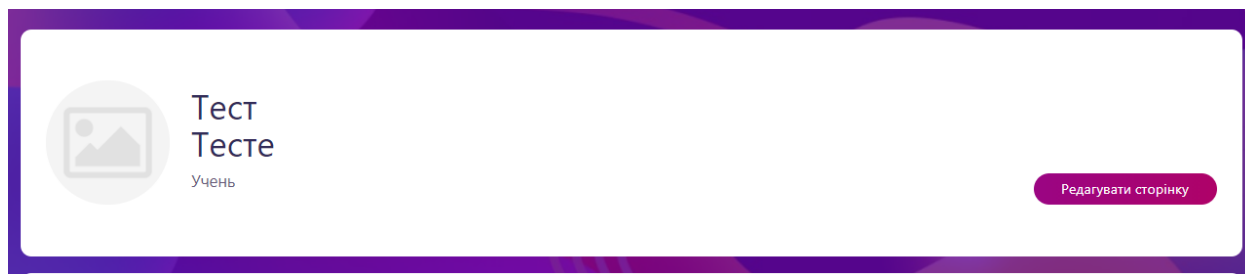


Рисунок 3.20 – Особиста сторінка

The image shows a form for editing a user profile. On the left, there is a large square area for a profile picture, currently showing a placeholder icon. Below it is a file selection area with the text "No file chosen" and a "Browse" button. To the right of the picture area, there are several input fields: "Ім'я" (Name) with the value "Тест", "Прізвище" (Surname) with the value "Тесте", "Дата народження" (Date of birth) with the value "02.05.1999" and a calendar icon, "Email" with the value "st@gmail.com", "Стать" (Gender) with a dropdown menu showing "Жіноча", and "Новий пароль" (New password) with an empty field. At the bottom of the form is a green button with white text that says "Зберегти" (Save).

Рисунок 3.21 - Форма для редагування

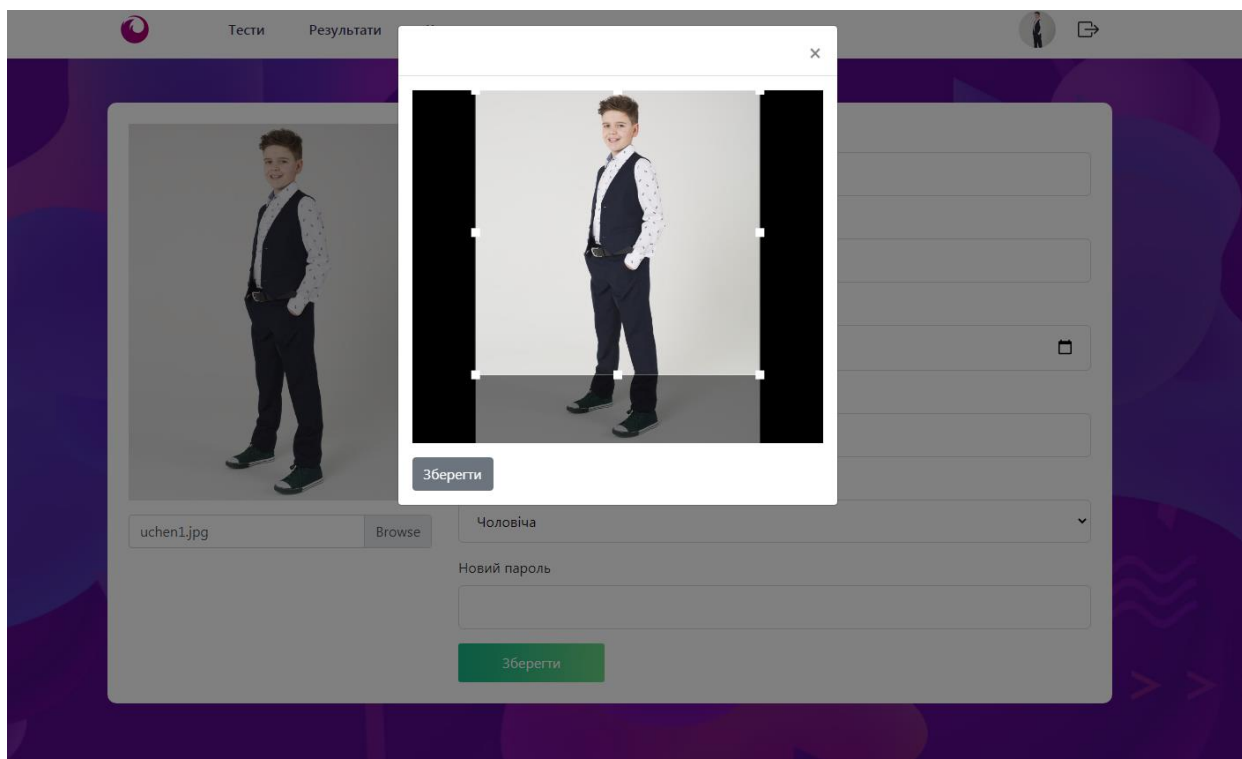


Рисунок 3.22 – Завантаження фотографії

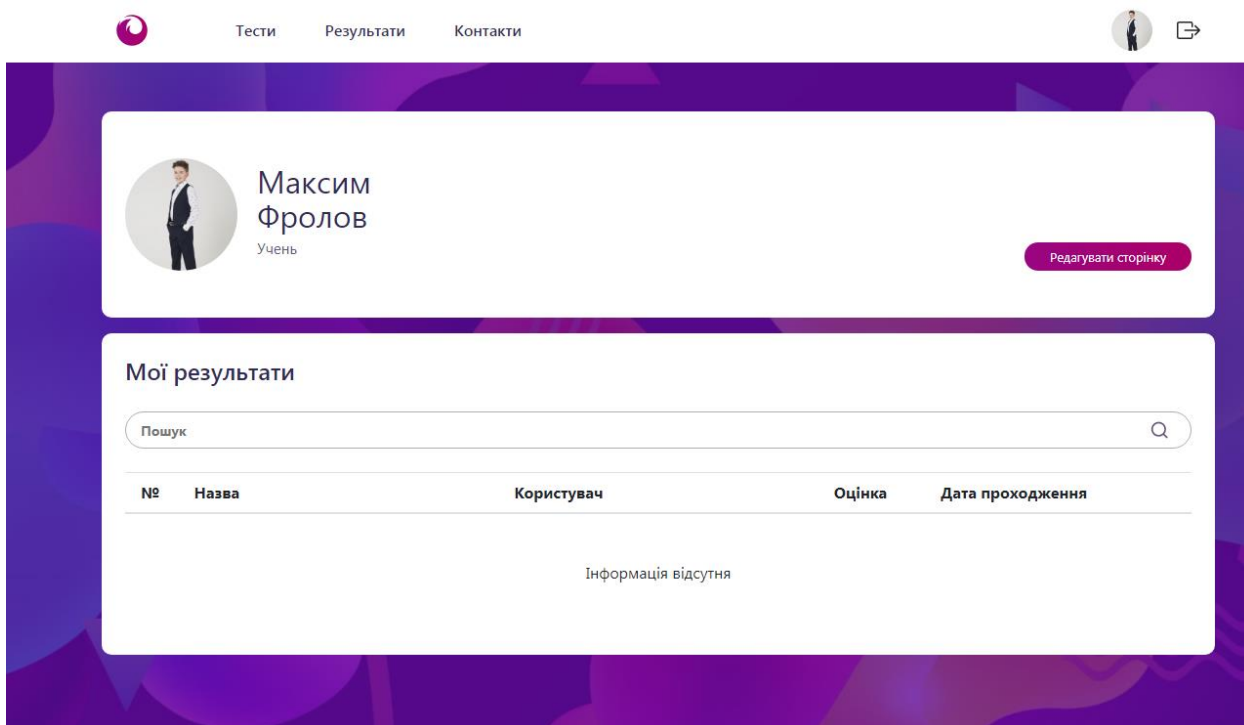


Рисунок 3.23 - Результат редагування

3.4 Адміністрування сайту

Для користування сайтом в ролі адміністратора необхідно, як і користувачу, авторизуватись (рис.3.24).

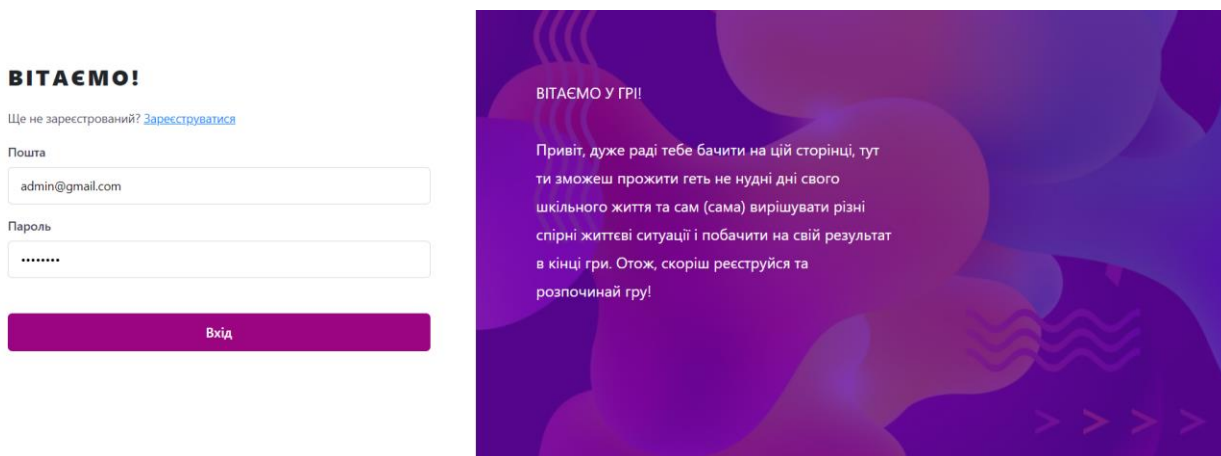


Рисунок 3.24 – Авторизація адміністратора

Адміністратор може переглядати, редагувати та видаляти всі наявні тести. Також йому наданий доступ до відповідей кожного користувача на будь-який пройдений ним тест (рис.3.25-26).

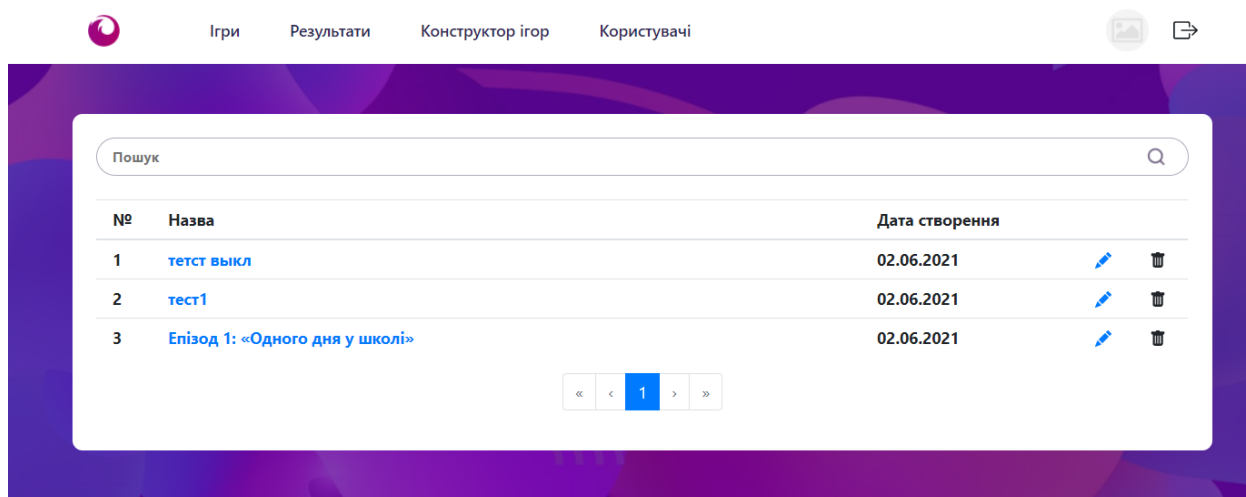


Рисунок 3.25 – Список тестів

№	Назва	Користувач	Оцінка	Дата проходження
1		Тест Тест	0	30.05.2021
2	Епізод 1: «Одного дня у школі»	Admin Admin	22	02.06.2021
3	Епізод 1: «Одного дня у школі»	Карл О	22	02.06.2021
4	Епізод 1: «Одного дня у школі»	Admin Admin	20	02.06.2021
5	тест1	Admin Admin	5	02.06.2021
6	тест1	викл 1	5	02.06.2021
7	тетст выкл	e e	4	02.06.2021
8	Епізод 1: «Одного дня у школі»	Анонім	20	05.06.2021

Рисунок 3.26 – Результати

Адміністратору надаються вповноваження для створення нових тестів. Для цього є певна форма для заповнення, що наведена на рис.3.27-3.29. В неї записується назва тесту, короткий опис, вигляд виведення варіантів, самі варіанти відповідей та мінімальна оцінка. Прикладом був обраний тест зі збірника методик Практична психологія конфлікту авторів Ложкін Г.В., Пов'якель Н.І. [25].

СТВОРЕННЯ ТЕСТУ

Назва тесту

Оцінка рівня конфліктності особистості

Опис тесту


Тест "Оцінка рівня конфліктності особистості" зі збірника методик "Практична психологія конфлікту Ложкин Г.В., Повякель Н.І." інтерпритований у виді "гри".


Виведення результату

Вдаєть максимальну оцінку при якій буде введено результат.
Наприклад: 90 - відмінно, 69 - задовільно, 29 - незадовільно

Варіанти відповідей	Мінімальна оцінка
Низький рівень конфліктності.	14
Рівень конфліктності нижчий за середній.	21
Середній рівень конфліктності	25
Рівень конфліктності вищий за середній.	30

Рисунок 3.27 – Форма створення тесту


Рівень конфліктності вищий за середній. 30 

Високий рівень конфліктності. 35 


[Додати рекомендацію](#)

Розробка - ЕТАП 1


Назва етапу


Тести Тести 14 [Додати](#) 


Запитання 1

Чи притаманне тобі прагнення до домінування, тобто до того, щоб підкорити своїй волі інших? [Додати](#) 

Варіанти відповідей

Ні. 1 

Іноді. 2 

Так. 3 

Додаткове зображення Browse /uploads/60be579de1cee.gif Browse /uploads/60be58062ea6a.gif Browse





Рисунок 3.28 – Створення тесту

Розробка - ЕТАП 2

Назва етапу


Рекомендації Рекомендація [Додати](#) 

Рекомендація


Коли не подобається, як до тебе ставляться:
 (1) – дивишся в очі «опоненту»;
 (2) – ставиш руку вперед та кажеш «СТОП»;
 (3) – опишуєш свої почуття, кажеш, що тобі неприємно; якщо це не допомагає, то звертаєшся до дорослого.
 Пам'ятай, що конфлікти – невід'ємна складова розвитку. Під час вирішення конфліктної ситуації можна вчиться домовлятися, знаходити нові шляхи вирішення питань, по-іншому дивитися на проблему та працювати у команді.


Розробка - ЕТАП 3

Назва етапу

Відкриті питання Відкриті пита 2 [Додати](#) 

Запитання

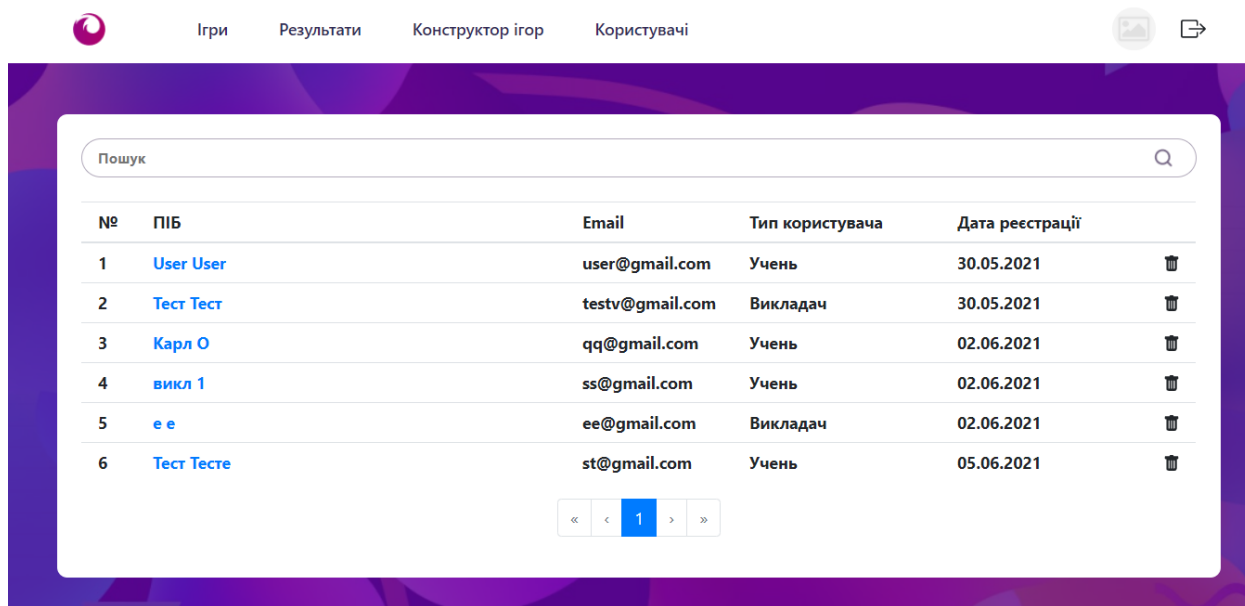
Що ти для себе виніс(ла)? 

Чи сподобалося тобі проходження даного тесту? 

[Додати етап](#) [Зберегти](#)

Рисунок 3.29 – Створення тесту

Список користувачів, який може переглянути адміністратор представлений на рис.3.30. В ньому надається ПІБ користувача, його електронна адреса, викладач це чи учень та дата його реєстрації. Адміністратор може видалити будь-кого з цього списку.



№	ПІБ	Email	Тип користувача	Дата реєстрації	
1	User User	user@gmail.com	Учень	30.05.2021	🗑
2	Тест Тест	testv@gmail.com	Викладач	30.05.2021	🗑
3	Карл О	qq@gmail.com	Учень	02.06.2021	🗑
4	викл 1	ss@gmail.com	Учень	02.06.2021	🗑
5	е е	ee@gmail.com	Викладач	02.06.2021	🗑
6	Тест Тесте	st@gmail.com	Учень	05.06.2021	🗑

Рисунок 3.30 – Список користувачів

Викладачу представлені всі зроблені ним тести на його власній сторінці та надана можливість переглядати результати (рис.3.31-3.32).

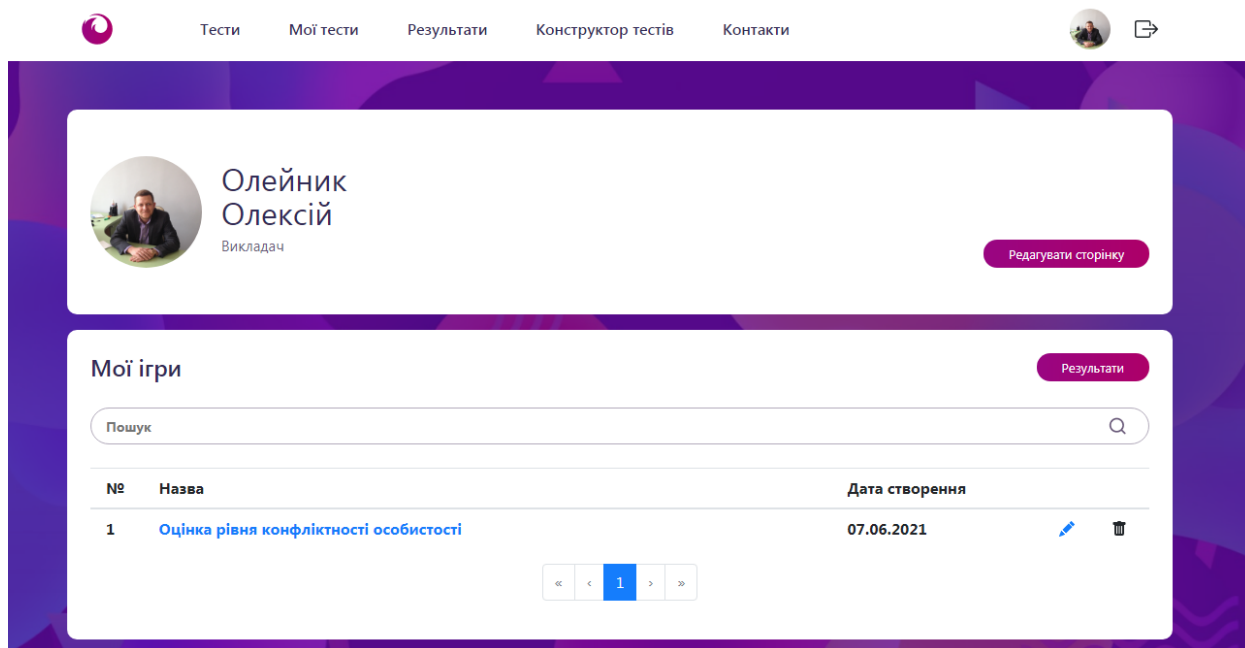


Рисунок 3.31–Особиста сторінка викладача

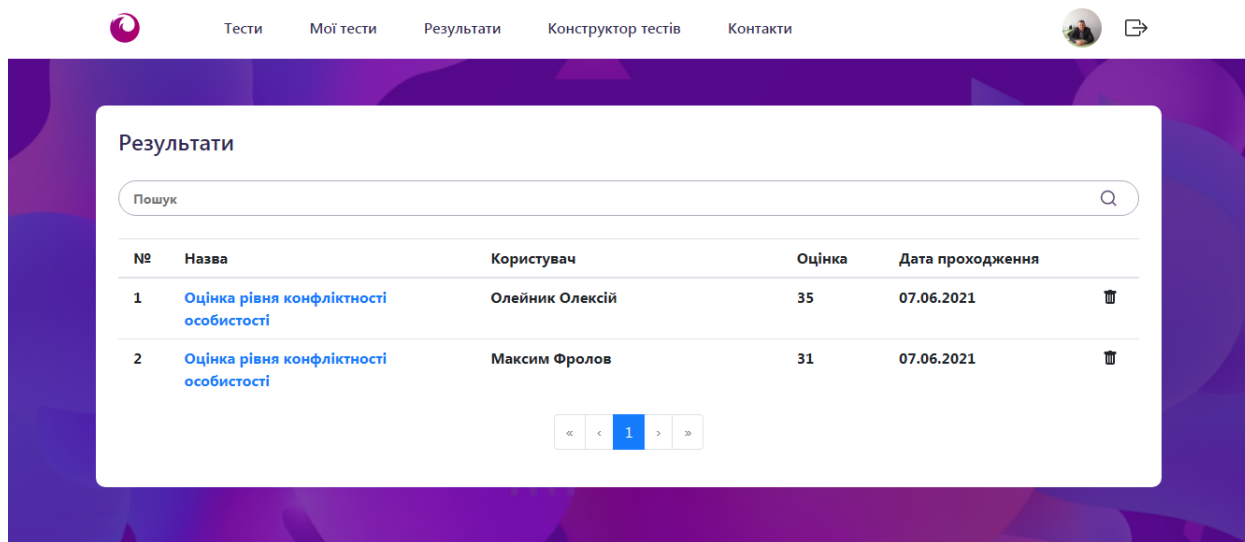


Рисунок 3.32 - Результати

Особиста сторінка адміністратора має такий самий вигляд, як і будь-якого іншого користувача (рис.3.33).

Ігри Результати Конструктор ігор Користувачі

Admin Admin
Адміністратор

Редагувати сторінку

Мої ігри Результати

Пошук

№	Назва	Дата створення		
1	тест1	02.06.2021		
2	Епізод 1: «Одного дня у школі»	02.06.2021		

« < 1 > »

Рисунок 3.33 – Особиста сторінка адміністратора

ВИСНОВКИ

При виконанні роботи було проаналізовано аспекти використання інформаційних технологій при проведенні психологічного тестування. На базі аналогів web-додатків для проходження психологічних тестувань були виявлені їх недоліки та переваги, які допомогли в проектуванні web-додатку визначення схильності до конфліктної поведінки.

Було розроблене технічне завдання, обрано технічні та програмні засоби для реалізації поставленої мети, виконано планування робіт. Проведено структурно-функціональне моделювання визначення схильності до конфліктної поведінки з використанням web-додатку. Спроектовано базу даних web-додатку. Розроблено web-додаток визначення схильності до конфліктної поведінки та розміщено на хостингу.

Створений web-додаток орієнтований на розробку повноцінної системи тестування для виявлення схильності до конфліктної поведінки. Web-додаток може бути застосований для зручного використання та експлуатації вчителями психології, практичними психологами для розробки тестів та проведення тестування. Використання даного web-додатку не залежить від локації користувача чи того, хто проводить тестування.

Практичне значення web-додатку полягає у його використанні як повноцінної платформи для створення тестових завдань та проведення психологічних тестувань задля встановлення стану користувачів та подальшої роботи із зібраними даними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. HelpingYoungChildrenChannelTheirAggression [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.zerotothree.org/resources/12-helping-young-children-channel-their-aggression> (дата звернення: 25.10.2020);
2. Researchlinkingviolententertainmenttoaggressionretractedafterscrutiny [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.sciencemag.org/news/2021/02/research-linking-violent-entertainment-aggression-retracted-after-scrutiny> (дата звернення: 25.10.2020);
3. ViolentBehaviorinChildrenandAdolescents [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://www.aacap.org/AACAP/Families_and_Youth/Facts_for_Families/FFF-Guide/Understanding-Violent-Behavior-In-Children-and-Adolescents-055.aspx(дата звернення: 25.10.2020);
4. Інформаційні технології в культурі, мистецтві, освіті, науці, економіці та бізнесі: матеріали Міжнародної науково-практичної конференції. – Київ: Видавничий центр КНУКіМ, 2017. - № 21. – 238 с.
5. Інформаційні технології як інструментальні засоби психолога [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: <https://subcase.ru/uk/informacionnye-tehnologii-kak-instrumentalnye-sredstva-psihologa.html> (дата звернення: 29.04.2021);
6. Інформаційні технології як інструментальні засоби психолога [Електронний ресурс]. – 2011. – Режим доступу до ресурсу: <https://subcase.ru/uk/informacionnye-tehnologii-kak-instrumentalnye-sredstva-psihologa.html> (дата звернення: 29.04.2021);
7. Сучасні інформаційні технології в роботі психолога [Електронний ресурс]. – 2020. – Режим доступу до ресурсу:

http://journal.osnova.com.ua/article/48358/Сучасні_інформаційні_технології_в_роботі_психолога(дата звернення: 25.10.2020);

8. ADDITUDES [Електронний ресурс]. – 2020. – Режим доступу до ресурсу:<https://www.additudemag.com/tests-for-oppositional-defiant-disorder-in-children-odd/>(дата звернення: 25.10.2020);

9. Саморозвиток [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://srcaltufevo.ru/en/testys-na-agressivnoe-povedenie-test-agressivnosti-oprosnik-l-g-rochebut.html>(дата звернення: 25.10.2020);

10. IDlabs.com [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.idrlabs.com/aggression/test.php>(дата звернення: 25.10.2020);

11. Angers, IrritabilityandAggressioninKids [Електронний ресурс]. – 2020. – Режим доступу до ресурсу:[https://www.yalemedicine.org/conditions/anger-issues-in-children-and-teens#:~:text=Multiple%20factors%20can%20contribute%20to,harm%20to%20oneself%20or%20another\).&text=For%20children%2C%20anger%20issues%20often,compulsive%20disorder%2C%20and%20Tourette%27s%20syndrome](https://www.yalemedicine.org/conditions/anger-issues-in-children-and-teens#:~:text=Multiple%20factors%20can%20contribute%20to,harm%20to%20oneself%20or%20another).&text=For%20children%2C%20anger%20issues%20often,compulsive%20disorder%2C%20and%20Tourette%27s%20syndrome) (дата звернення: 25.10.2020);

12. Фесенко А.О. Використання інформаційно-комунікативних технологій практичного психолога [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://informatika.udpu.edu.ua/?page_id=1964(дата звернення: 25.10.2020);

13. Критерії оцінки дизайну сайту [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://crystal-digital.ru/blogs/15-kriteriev-htoby-oczenit-dizajn-sajta/> (дата звернення: 25.10.2020);

14. Методи аналізу сайтів [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://webprojects.ru/articles/sites/analiz_dizayna_sayta_kriterii_otsenki/ (дата звернення: 25.10.2020);

15. Whyshouldyouuse Vue.js whenusingLaravel[Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.ongraph.com/vuejs-with-laravel-becoming-popular-choice-build-applications/> (дата звернення: 25.10.2020);

16. TheBestsCombinationfor a PerfectWebApplication[Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://blog.pusher.com/why-vuejs-laravel/>(дата звернення: 20.04.2021);

17. ThisCoupleGettingPopular[Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://papdan.com/2020/07/20/why-vuejs-and-laravel-match-perfectly/>(дата звернення: 21.04.2021);

18. Laraveland Vue.js [Електронний ресурс]. – 2020. – Режим доступу до ресурсу:<https://www.technoexponent.com/blog/laravel-and-vue-js-the-best-combination-for-a-perfect-web-application/>(дата звернення: 25.04.2021);

19. WhatisFrontEndDevelopment? [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://skillcrush.com/blog/skills-to-become-a-front-end-developer/>(дата звернення: 25.04.2021);

20. Key Technologies andConcepts [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.altexsoft.com/blog/front-end-development-technologies-concepts/>(дата звернення: 29.04.2021);

21. BackendDeveloper. SkillsNeedforWebDevelopment [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.guru99.com/what-is-backend-developer.html>(дата звернення: 29.04.2021);

22. Howdoesitwork – IDEF [Електронний ресурс]. – 2020. – Режим доступу до ресурсу:http://www.syque.com/quality_tools/tools/Tools19.htmhttps://stud.com.ua/37674/psihologiya/novi_mozhливosti_perspektivi_kompyuternoї_dіagnostiki (дата звернення: 29.04.2021);

23. AdrienPresley [Текст] TheUseof IDEF0 fortheDesignandSpecificationofMethodologies – 2010. - № 9. –С. 5-7.

24. Нові можливості та перспективи комп'ютерної діагностики [Електронний ресурс].–2020.–Режим доступу до ресурсу: https://stud.com.ua/37674/psihologiya/novi_mozhливosti_perspektivi_kompyuternoї_dіagnostiki (дата звернення: 29.04.2021);

25. Ложкин Г.В. Практична психологія конфлікту/ Г.В.Ложкин,
Н.И.Повякель—Київ : МАУП, 2002. - 256 с.

ДОДАТОК А

ТЕХНІЧНЕ ЗАВДАННЯ
на розробку інформаційної системи
«Web-додаток визначення схильності до конфліктної поведінки»

1 ПРИЗНАЧЕННЯ Й МЕТА СТВОРЕННЯ WEB-ДОДАТКУ

1.1 Призначення web-додатку

Створений програмний продукт орієнтований на розробки повноцінної інформаційної системи тестування та виявлення схильності до конфліктної поведінки. Web-додаток буде розроблений для зручного використання. Дана інформаційна система не залежить від локації користувача чи того, хто проводить тестування.

1.2 Мета створення web-додатку

Головна мета web-додатку – створення повноцінної платформи для створення тестових завдань, проведення тестувань за для встановлення стану користувачів та подальша робота із зібраними даними.

1.3 Цільова аудиторія

До цільової аудиторії web-додатку можна віднести школярів, викладачів, психологів та інших, хто працює чи вивчає дану тематику.

2 ВИМОГИ ДО WEB-ДОДАТКУ

2.1 Вимоги до web-додатку в цілому

2.1.1 Вимоги до структури й функціонування web-додатку

Web-додаток по виявленню схильності до конфліктної поведінки після реалізації повинен бути викладений на сервер для перегляду, тестування та користування.

2.1.2 Вимоги до персоналу

Для роботи з web-додатком не потрібно наймати повноцінну команду для підтримки. Для цього потрібно найняти декількох адміністраторів. Адміністратори відповідають за додавання, редагування та видалення певної інформації.

2.1.3 Вимоги до збереження інформації

Уся інформація надана у web-додатку по виявленню схильності до конфліктної поведінки повинна зберігатися у базі даних реалізованій засобами системи управління базами даних MySQL.

2.1.4 Вимоги до розмежування доступу

Система має декілька типів користувачів системи. А саме – адміністратор, викладач та учень. При реєстрації можна обрати між «Викладач» та «Учень». Розподіл функціоналу детальніше представлений в табл.А.1.

Таблиця А.1 – Функціонал за групами користувачів

№	Користувач	Можливості користувача
1	Адміністратор	Перегляд загальної інформації;
		Підтримка сайту;
		Блокування користувачів;
		Перевірка контенту.

Продовження таблиці А.1

№	Користувач	Можливості користувача
2	Учень	Перегляд інформації про іншого користувача;
		Оформлення підписки на інших користувачів;
		Пройдення тестування;
		Перегляд історії пройдених тестів та їх результатів;
		Виконувати фільтрацію та пошук на сайті.
3	Викладач	Перегляд інформації про іншого користувача;
		Створення тесту за допомогою конструктора;
		Перегляд результатів тестування за тим чи іншим створеним тестом;
		Редагування та налаштування створеного тесту;
		Виконувати фільтрацію та пошук на сайті.

2.2 Структура web-додатку

2.2.1 Загальна інформація про структуру web-додатку

Структура web-додатку по виявленню схильності до конфліктної поведінки являє собою набір сторінок, що представлений у табл.А.2.

Таблиця А.2 – Опис сторінок додатку

№	Назва	Опис сторінки
1	Реєстрація	Сторінка для реєстрації користувача. Користувач повинен ввести такі дані для створення аккаунту: <ul style="list-style-type: none"> – Ім'я; – Прізвище; – Електронна пошта; – Пароль; – Дата народження; – Стать; – Тип аккаунту (Викладач / Учень).
2	Авторизація	Сторінка для авторизації: <ul style="list-style-type: none"> – Електронна пошта; – Пароль.
3	Тестування	Функціонал змінюється в залежності від типу аккаунту. Учень – проходження тестів. Користувач може переглянути результати пройдених тестів. Викладач – створені тести. Також користувач може перейти до конструктора тесту.
4	Створення тесту	Користувач може додати будь-яку кількість запитань.

Продовження таблиці А.2

№	Назва	Опис сторінки
5	Налаштування створеного тесту	Сторінка для редагування уже створеного тесту. На даній сторінці можна редагувати як запитання, так і іншу інформацію.
6	Налаштування аккаунту	Сторінка для редагування основної інформації про користувача та додаткове її налаштування.

2.2.2 Навігація

Навігацією у web-додатку по виявленню схильності до конфліктної поведінки являє собою меню в шапці сторінки. Кожен із пунктів відповідає певній сторінці.

2.2.3 Наповнення web-додатку

Управління контентом буде вестися через адміністративну панель. До контенту відноситься усе, що пов'язане з тематикою роботи.

2.2.4 Дизайн та структура додатку

Інтерфейс web-додатку по виявленню схильності до конфліктної поведінки повинен забезпечувати інтуїтивно зрозумілий вид. При виборі будь-якого з пунктів меню користувачем повинна завантажуватися відповідна йому інформаційна сторінка.

Прототип майбутнього дизайну представлений на рисунку А.1.

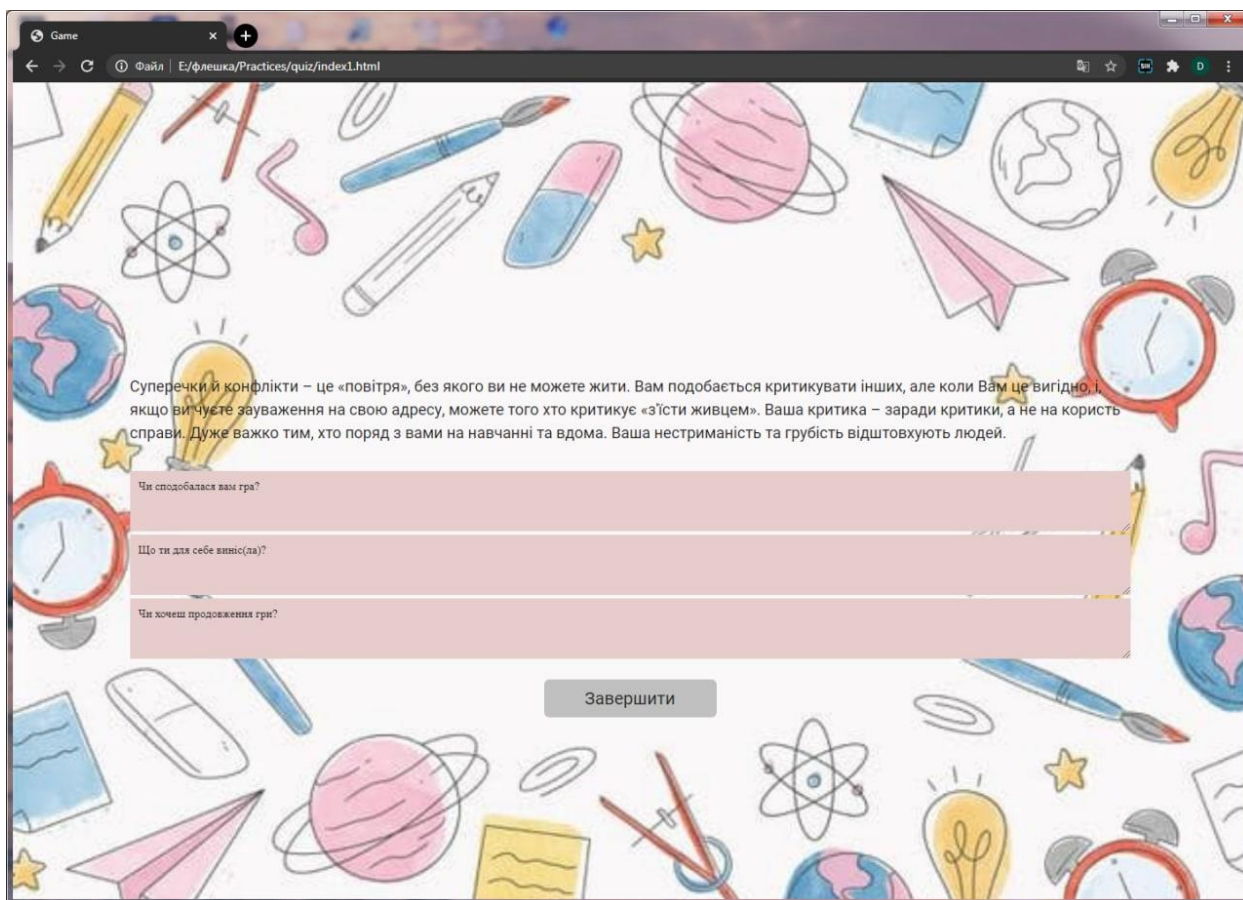


Рисунок А.1 – Прототип сторінки із запитанням

2.2.5 Система навігації (карта web-додатку)

Карта web-додатку по виявленню схильності до конфліктної поведінки зображена на рисунку А.2.

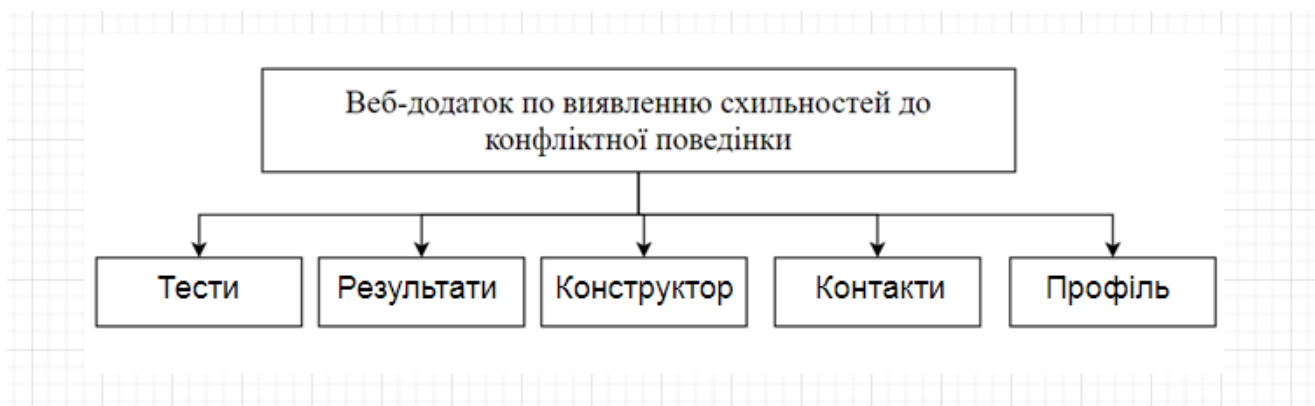


Рисунок А.2 – Карта web-додатку

2.3 Вимоги до функціонування системи

2.3.1 Потреби користувача

Потреби користувача, визначені на основі рішення замовника, представлені у таблиці А.3.

Таблиця А.3 – Потреби користувача

ІД	Потреби користувача	Джерело
UN-01	Реєстрація на сайті як «Вчитель» чи «Учень»	Клієнт
UN-02	Перегляд інформації про доступні тести	Клієнт
UN-03	Можливість зворотного зв'язку	Клієнт
UN-04	Редагування загальних даних	Адміністратор
UN-05	Перегляд інформації про викладача	Клієнт
UN-06	Створення тестування в ролі «Викладач»	Клієнт
UN-06	Розрахунок результатів тестування	Клієнт
UN-06	Відображення всіх результатів тестування для його власника	Клієнт

2.3.2 Функціональні вимоги

На основі потреб майбутніх користувачів web-додатку були визначені такі функціональні вимоги:

- Реєстрація та авторизація користувачів;
- Пошук інформації у web-додатку;
- Відображення результатів пошуку;
- Можливість проходження тестування чи створення власного тесту;
- Адміністрування інформації та контенту web-додатку;
- Можливість перегляду результатів тестування;

– Наявність зв'язку адміністратора web-додатку з зареєстрованим користувачем.

2.3.3 Системні вимоги

Даний розділ визначає, розподіляє та вказує на системні вимоги, визначені розробником. Їх перелік наведений в таблиці А.4.

Таблиця А.4 – Системні вимоги

ID	Системні вимоги	Пріоритет	Опис
SR-01	Наявність модуля створення тесту	М	Надає можливість клієнту типом «Викладач» створювати тест
SR-02	База даних із записами результатів тестування	М	Надає можливість відображати власникам тестів результати їх проходження
SR-03	База даних з контентом	М	Відповідає за заповнення web-додатку контентом
SR-04	Наявність модуля зворотного зв'язку	S	Надає можливість клієнту зв'язатися з адміністратором web-додатку
SR-05	Панель адміністратора	М	Відповідає за подальше наповнення та редагування контенту адміністратором

Умовні позначення в таблиці А.4:

Musthave (M) – вимоги, які повинні бути реалізовані в системі;

Shouldhave (S) – вимоги, які мають бути виконані, але вони можуть почекати своєї черги;

Couldhave (C) – вимоги, які можуть бути реалізовані, але вони не є центральною ціллю проекту.

2.4 Вимоги до видів забезпечення

2.4.1 Вимоги до програмного забезпечення розробки web-додатку

Реалізація web-додатку відбувається з використанням:

- HTML&CSS
- PHP 7.4.4 та його фреймворки
- MySQL 8.0

2.4.2 Вимоги до лінгвістичного забезпечення

Web-додаток має бути виконаний українською мовою.

2.4.3 Вимоги до програмного забезпечення перегляду web-додатку

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам: веб-браузер Chrome 2 і вище.

3 СКЛАД І ЗМІСТ РОБІТ ЗІ СТВОРЕННЯ WEB-ДОДАТКУ

Докладний опис етапів роботи зі створення web-додатку по виявленню схильності до конфліктної поведінки наведено в таблиці А.5.

Таблиця А.5 – Етапи створення web-додатку

№	Склад і зміст робіт	Строк розробки (у робочих днях)
1	Створення технічного завдання	3
2	Розробка прототипу	5
3	Підготовка макету дизайну сайту	2
4	Верстка	15
5	Робота над програмною частиною	20
6	Наповнення сайту контентом	1
7	Перевірка працездатності сайту	8
8	Підключення веб-аналітики	2
Загальна тривалість робіт		56

ДОДАТОК Б

ПЛАНУВАННЯ РОБІТ для розробки інформаційної системи «Web-додаток визначення схильності до конфліктної поведінки»

1 ІДЕНТИФІКАЦІЯ МЕТИ ІТ-ПРОЕКТУ

1.1 Розробка концепції проекту

1.1.1. Ідентифікація ідеї проекту.

Головна мета проекту – розробка web-додатку для проходження тесту-гри.

1.1.2. Деталізація мети методом SMART

Таблиця Б.1.1 – Таблиця опису проекту за SMART методом

Specific (конкретна)	Розробка web-додатку, використовуючи який будуть виконуватися поставлені цілі та задачі, будуть формуватися замовлення.
Measurable (вимірювана)	Прогрес та якість буде вимірюватися в кількості користувачів системи, користувачів яки пройшли "гру" їх спроб та результатів, відгуків.
Achievable (досяжна)	Даний web-додаток буде розроблятися з використанням HTML, CSS, JavaScript, JSON та PHP, MySQL.
Relevant (реалістична)	Для виконання проекту є всі необхідні засоби та знання.
Time-framed (обмежена у часі)	Web-додаток розроблюється з обмеженням у часі на основі сформованого календарного плану та матриці відповідальності.

1.2 Техніко-економічне дослідження

1.2.1. Дослідження ІТ-проекту, організації, регіону

Створений web-додаток буде використовуватися та поширюватися серед учнів шкіл. Система оцінювання виконаних дій на основі результату проходження тесту у виді "гри" або звичайного тесту, допоможе викладачу з перевіркою схильності до конфлікту учнів.

1.2.2. Оцінка цінності, життєздатності, економічної ефективності та життєстійкості ІТ-проекту

Модель буде корисною у сфері освіти для проведення тестувань учнів.

Економічна ефективність проекту обґрунтована збільшенням кількості конфліктних ситуацій.

Сталість проекту підтверджується наявністю попиту і актуальності даного проекту на даний момент часу.

2 УПРАВЛІННЯ ВИМОГАМИ

2.1 Загальні вимоги до інформаційної системи

2.1.1 Вимоги до структури й функціонування

Інформаційна система повинна бути реалізована у вигляді web-додатку, доступного в мережі Інтернет. Web-додаток повинен складатися із взаємозалежних розділів із чітко розділеними функціями.

2.1.2. Вимоги до стилістичного оформлення

Інтерфейс web-додатку повинен забезпечувати інтуїтивно зрозуміле уявлення структури розміщеної на ньому інформації, логічний перехід до розділів і сторінок. Посилання на сторінки повинні бути забезпечені заголовками. При виборі будь-якого з пунктів меню користувачем повинна завантажуватися відповідна йому інформаційна сторінка, а в блоці меню відкриватися список підрозділів чи перенаправлення на підрозділ.

2.2 Вимоги до розділів

Web-додаток повинен складатися з наступних розділів:

- Тести;
- Результати;
- Контакти;
- Вхід/Реєстрація
- Особистий кабінет(посилання авторизованим користувачам)

3 ОПИСАННЯ ФАЗИ РОЗРОБКИ ІТ-ПРОЕКТУ

3.1 Планування змісту структури робіт ІТ-проекту

WBS проекту (вона ж WorkBreakdownStructure або ICP, Ієрархічна Структура Робіт) - це розбиття проекту на конкретні результати, які повинні бути досягнуті для досягнення цілей проекту. Важливо розуміти, що в WBS збираються саме результати робіт, а не завдання, які потрібно виконати для отримання цих результатів. WBS є ієрархічною та інкрементною декомпозицією проекту у фазі, кінцеві результати та пакети робіт. Вона є ієрархічною структурою, що показує подальший розподіл необхідних для виконання мети зусиль; наприклад, програма, проект чи договір.

Загальна WBS структура представлена на рисунку Б.2.1.

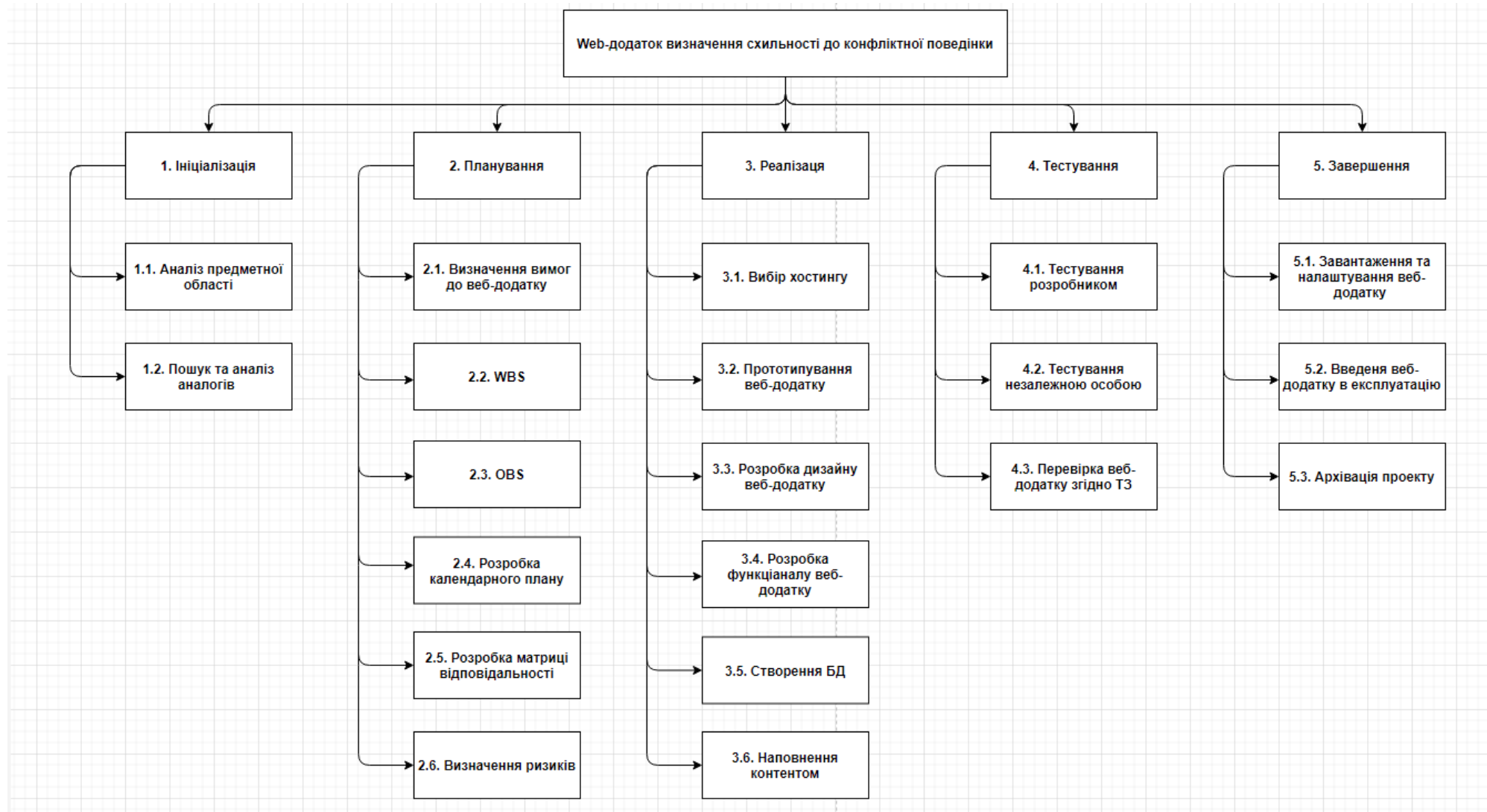


Рисунок Б.2.1 – WBS-структура проекту

3.2. Планування структури організації

Структура розподілу організації або OBS – це ієрархічна модель, що описує встановлені організаційні рамки для планування проектів, управління ресурсами, відстеження часу та витрат, розподілу витрат, звітування про доходи / прибуток та управління роботою.

Організаційна структура представляє собою графічне відображення учасників проекту та їх відповідальних осіб, які задіяні в реалізації проекту. На верхньому рівні OBS розташована команда проекту.

Визначається за переліком пакетів робіт нижнього рівня кожної гілки WBS-структури.

Загальна OBS структура представлена на рисунку Б.2.2.

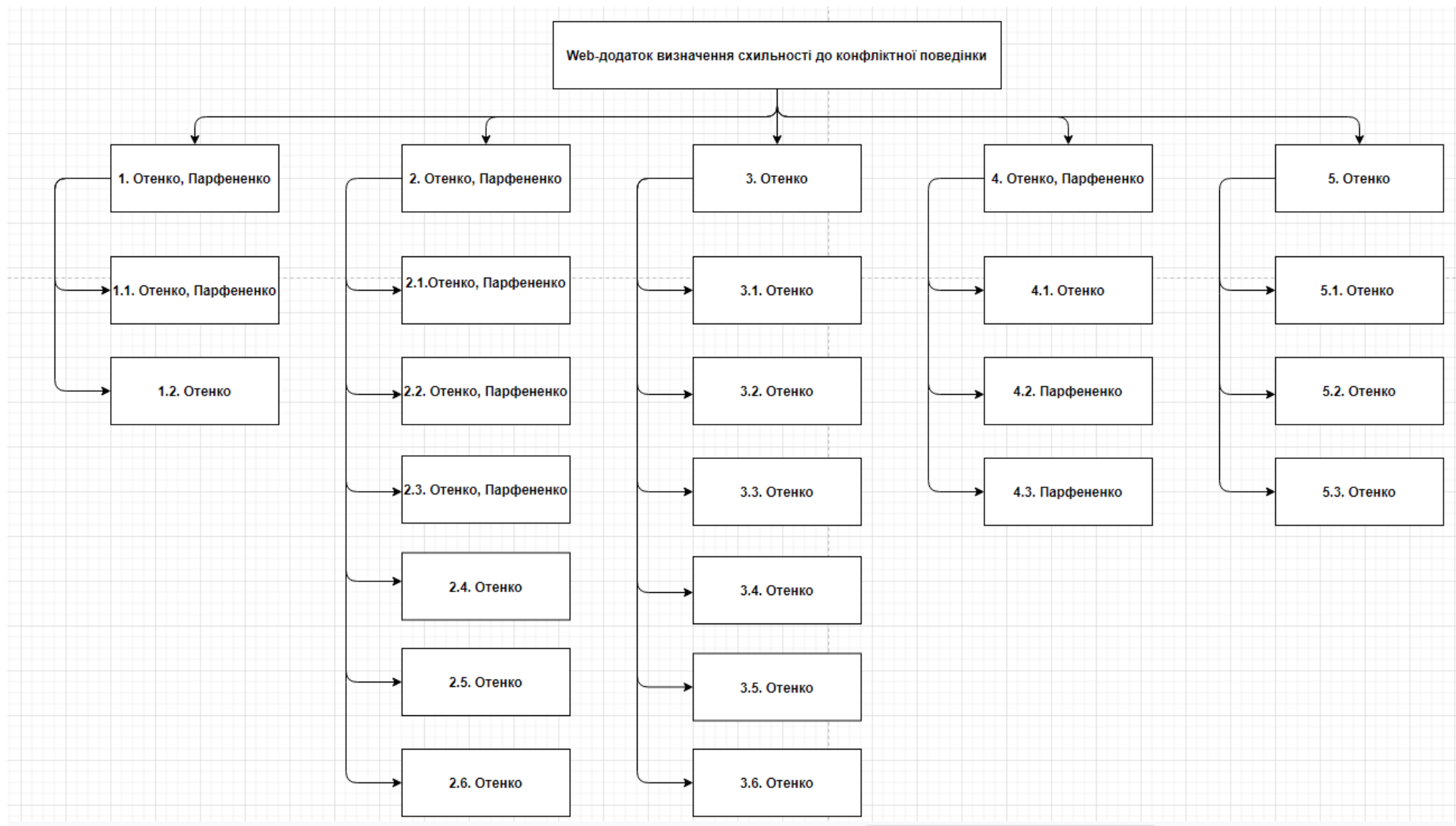


Рисунок Б.2.2 – OBS структура проекту

3.3. Побудова матриці відповідальності

На підставі OBS та WBS структур будується матриця відповідальності проекту. Матриця відповідальності закріплює за кожною елементарною роботою виконавця. Використовується для контролю відповідності розподілу ролей цілям проекту. На верхньому рівні розподіляються ролі та відповідальність по елементах OBS. На нижньому — фази проекту. Матриця відповідальності представлена в таблиці Б.3.1

Таблиця Б.3.1 – Матриця відповідальності

Назва етапу		Отенко Д.С.	Парфененко Ю.В.
1	Ініціалізація	+	+
1.1	Аналіз предметної області	+	+
1.2	Пошук та аналіз аналогів	+	-
2	Планування	+	+
2.1	Визначення вимог до web-додатку	+	+
2.2	WBS	+	+
2.3	OBS	+	+
2.4	Розробка календарного плану	+	-
2.5	Розробка матриці відповідальності	+	-
2.6	Визначення ризиків	+	-
3	Реалізація	+	-
3.1	Вибір хостингу	+	-
3.2	Прототипування web-додатку	+	-
3.3	Розробка дизайну web-додатку	+	-
3.4	Розробка функціоналу web-додатку	+	-
3.5	Створення БД	+	-

Продовження таблиці Б.3.1

Назва етапу		Отенко Д.С.	Парфененко Ю.В.
3.6	Наповнення контентом	+	-
4	Тестування	+	+
4.1	Тестування розробником	+	-
4.2	Тестування незалежною особою	+	+
4.3	Перевірка web-додатку згідно ТЗ	+	+
5	Завершення	+	-
5.1	Завантаження та налаштування web-додатку	+	-
5.2	Введення web-додатку в експлуатацію	+	-
5.3	Архівація проекту	+	-

3.4. Побудова календарного графіку виконання ІТ-проекту

Діаграма Ганта – це горизонтальна діаграма з тимчасовою шкалою, яка використовується для ілюстрації плану робіт за проектом з прив'язкою до часу.

За допомогою діаграм Ганта керівники проектів і менеджери по продукту розбивають проекти на робочі завдання для зручності управління, підтримують порядок в роботі і роблять залежності між завданнями наочними.

Діаграми Ганта дозволяють спростити складові проекти. За допомогою цього засобу можна досить наочно і зручно для узагальнення представити велику кількість даних. Завдяки цій гістограмі велика кількість зацікавлених осіб, команд або їх учасників не стане проблемою для запису завдань, як і часті зміни обсягу роботи. Ще одна перевага використання діаграми Ганта полягає в тому, що вона дає загальне уявлення про проект в цілому, в тому числі про всі контрольних точках і терміни виконання. Діаграму Ганта можна уявити як ефективний засіб раннього попередження(рис.Б.3.1-2).







		Название задачи	Длительность	Начало	Окончание
1		Аналіз предметної області	7 дней	Пн 18.01.21	Вт 26.01.21
2		Пошук та аналіз аналогів	5 дней	Ср 27.01.21	Вт 02.02.21
3		Визначення вимог до веб-додатку	4 дней	Ср 03.02.21	Пн 08.02.21
4		WBS	3 дней	Вт 09.02.21	Чт 11.02.21
5		OBS	3 дней	Пт 12.02.21	Вт 16.02.21
6		Розробка календарного плану	4 дней	Ср 17.02.21	Пн 22.02.21
7		Розробка матриці відповідальності	4 дней	Вт 23.02.21	Пт 26.02.21
8		Визначення ризиків	6 дней	Пн 01.03.21	Пн 08.03.21
9		Вибір хостингу	4 дней	Вт 09.03.21	Пт 12.03.21
10		Прототипування веб-додатку	5 дней	Пн 15.03.21	Пт 19.03.21
11		Розробка дизайну веб-додатку	7 дней	Пн 22.03.21	Вт 30.03.21
12		Розробка функціоналу веб-додатку	15 дней	Ср 31.03.21	Вт 20.04.21
13		Створення БД	6 дней	Ср 21.04.21	Ср 28.04.21
14		Наповнення контентом	1 день	Чт 29.04.21	Чт 29.04.21
15		Тестування розробником	2 дней	Пт 30.04.21	Пн 03.05.21
16		Тестування незалежною особою	7 дней	Вт 04.05.21	Ср 12.05.21
17		Перевірка веб-додатку згідно ТЗ	3 дней	Чт 13.05.21	Пн 17.05.21
18		Завантаження та налаштування веб-додатку	2 дней	Вт 18.05.21	Ср 19.05.21
19		Введення веб-додатку в експлуатацію	1 день	Чт 20.05.21	Чт 20.05.21
20		Архівація проекту	1 день	Пт 21.05.21	Пт 21.05.21

Рисунок Б.3.1 – Діаграма Ганта (інформація)

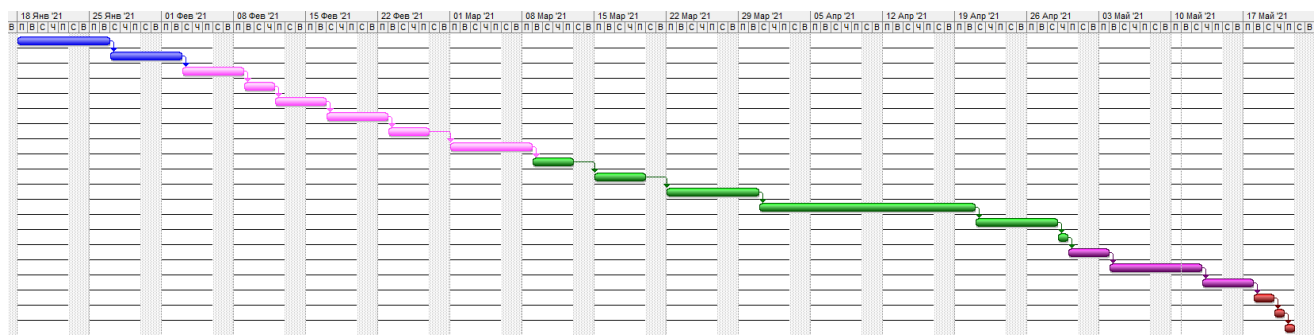


Рисунок Б.3.2 – Діаграма Ганта(графік)

4 УПРАВЛІННЯ РИЗИКАМИ ІТ-ПРОЕКТУ

Ризиком є ймовірна подія, яка у випадку її виникнення може як негативно, так і позитивно вплинути на конкретний проект. Управління ризиком – це процес зміни ризиків та реагування на події під час виконання проекту. При реалізації проекту важливою частиною є моніторинг ризиків. Отже, в даному випадку можна виділити деякі ризики. Основні ризики під час розробки web-додатку:

- R1 - зміна ТЗ на етапі розробки;
- R2 - недотримання календарного плану;
- R3 - збій роботи програмного забезпечення;
- R4 - хвороба або сімейні обставини;
- R5 - недостатня кількість ресурсів;
- R6 - помилка в роботі хостингу;
- R7 - помилка в роботі БД;
- R8 - зміна дати здачі проекту;

Під час процесу аналізу для визначення числових значень ймовірності появи ступеня впливу застосовувалась шкала ймовірності виникнення та величину втрат.

Таблиця Б.4.1 – Шкала оцінювання ймовірності виникнення та величини витрат.

Оцінка	Ймовірність виникнення	Величина втрат
1	Слабоймовірна	Мінімальна
2	Малоймовірна	Низька
3	Ймовірна	Середня
4	Дуже ймовірна	Висока
5	Майже ймовірна	Максимальна

На основі цих даних була проведена класифікація ризиків для даного проекту, що наведено в Таблиці Б.4.2.

Таблиця Б.4.2 – Класифікація ризиків.

№	Назва ризику	Ймовірність	Величина втрат
R1	Зміна ТЗ на етапі розробки	2	4
R2	Недотримання календарного плану	2	3
R3	Збій роботи програмного забезпечення	3	3
R4	Хвороба або сімейні обставини	3	4
R5	Недостатня кількість ресурсів	2	4
R6	Помилка в роботі хостингу	1	5
R7	Помилка в роботі БД	3	3
R8	Зміна дати здачі проекту	1	3

Використовуючи дані з таблиці класифікацій ризиків було створено матрицю ризиків, що показано в таблиці Б.4.2.

Таблиця Б.4.2 – Матриця ризиків.

5					
4					
3			R3, R7	R4	
2			R2	R1, R5	
1			R8		R6
Ймовірність Вел. втрат	1	2	3	4	5

На основі матриці ризику було створено комплексний план управління ризиками який наведений у Таблиці Б.4.3.

Таблиця Б.4.3 – Комплексний план управління ризиками.

№	Назва ризику	Рівень ризику	План дій
R1	Зміна ТЗ на етапі розробки	Середній	Отримати чіткі правки від замовника та виконати їх.
R2	Недотримання календарного плану	Середній	Перебудова стратегії реалізації проекту щодо загального терміну проекту.
R3	Збій роботи програмного забезпечення	Середній	Перевстановлення або встановлення нового ПО.
R4	Хвороба або сімейні обставини	Середній	Організувати бесіду, щоб розв'язати дану проблему.
R5	Недостатня кількість ресурсів	Середній	Виділити час на опрацювання наукової літератури.
R6	Помилка в роботі хостингу	Середній	Зв'язатися з технічною підтримкою хостингу для вирішення цієї проблеми. Якщо не вдається вирішити, обрати інший хостинг.
R7	Помилка в роботі БД	Середній	Провести аналіз БД та виправити помилки.

ДОДАТОК В. КОД РЕАЛІЗАЦІЇ WEB-ДОДАТКУ

2014_10_12_000000_create_users_table.php

Міграція створення таблиці користувачів.

```
<?php

useIlluminate\Database\Migrations\Migration;
useIlluminate\Database\Schema\Blueprint;
useIlluminate\Support\Facades\Schema;

classCreateUsersTableextendsMigration
{
publicfunctionup()
{
Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('surname');
    $table->string('date_birth');
    $table->boolean('sex');
    $table->string('photo')->default('/img/no-image.png');
    $table->string('email')->unique();
    $table->string('password');
    $table->foreignId('role_id');
    $table->rememberToken();
    $table->timestamps();
});

Schema::table('users', function (Blueprint $table) {
    $table->index('role_id');
    $table->foreign('role_id')->references('id')->on('roles')->onDelete('cascade');
});
}

/**
 * Reversethemigrations.
 *
 * @return void
 */
publicfunctiondown()
{
Schema::dropIfExists('users');
}
}
```

2021_04_18_111725_create_tests_table.php

Міграція створення таблиці тестів.

```
<?php

useIlluminate\Database\Migrations\Migration;
useIlluminate\Database\Schema\Blueprint;
useIlluminate\Support\Facades\Schema;

classCreateTestsTableextendsMigration
{
```

```

publicfunctionup()
{
Schema::create('tests', function (Blueprint $table) {
    $table->id();
    $table->foreignId('user_id');
    $table->text('title');
    $table->text('description');
    $table->timestamps();
});

Schema::table('tests', function (Blueprint $table) {
    $table->index('user_id');
    $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
});
}

/**
 * Reversethemigrations.
 *
 * @return void
 */
publicfunctiondown()
{
Schema::dropIfExists('tests');
}
}

```

2021_04_18_111757_create_steps_table.php

Міграція створення таблиці кроків для кожного тесту.

<?php

```

useIlluminate\Database\Migrations\Migration;
useIlluminate\Database\Schema\Blueprint;
useIlluminate\Support\Facades\Schema;

classCreateStepsTableextendsMigration
{
publicfunctionup()
{
Schema::create('steps', function (Blueprint $table) {
    $table->id();
    $table->foreignId('test_id');
    $table->string('title')->nullable();
    $table->string('type');
    $table->json('data');
    $table->timestamps();
});

Schema::table('steps', function (Blueprint $table) {
    $table->index('test_id');
    $table->foreign('test_id')->references('id')->on('tests')->onDelete('cascade');
});
}

/**
 * Reversethemigrations.
 *
 * @return void
 */
publicfunctiondown()

```

```

    {
    Schema::dropIfExists('steps');
    }
}

```

2021_04_18_111941_create_result_table.php

Міграція створення таблиці результатів тестів.

```
<?php
```

```

useIlluminate\Database\Migrations\Migration;
useIlluminate\Database\Schema\Blueprint;
useIlluminate\Support\Facades\Schema;

```

```
classCreateResultTableextendsMigration
```

```

{
publicfunctionup()
{
Schema::create('result', function (Blueprint $table) {
    $table->id();
    $table->foreignId('test_id');
    $table->foreignId('user_id');
    $table->integer('evaluation');
    $table->boolean('anonymous')->default(0);
    $table->timestamps();
});

```

```

Schema::table('result', function (Blueprint $table) {
    $table->index('test_id');
    $table->foreign('test_id')->references('id')->on('tests')->onDelete('cascade');
});

```

```

Schema::table('result', function (Blueprint $table) {
    $table->index('user_id');
    $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
});
}

```

```

/**
 * Reversethemigrations.
 *
 * @return void
 */

```

```

publicfunctiondown()
{
Schema::dropIfExists('result');
}
}

```

Tests.php

Модель тестів.

```
<?php
```

```
namespaceApp\Models;
```

```

useIlluminate\Database\Eloquent\Factories\HasFactory;
useIlluminate\Database\Eloquent\Model;

```

```
classTestsextendsModel
```

```

{
useHasFactory;

protected $table = 'tests';

protected $fillable = [
    'user_id',
    'title',
    'description'
];

protected $casts = [
    'created_at' => 'datetime:d.m.Y'
];

functionuser() {
return $this->belongsTo('App\Models\User', 'user_id');
}

functionscores() {
return $this->hasMany('App\Models\Scores', 'test_id');
}

functionsteps() {
return $this->hasMany('App\Models\Steps', 'test_id');
}
}

```

Result.php

Модель результатів тестів.

```

<?php

namespaceApp\Models;

useIlluminate\Database\Eloquent\Factories\HasFactory;
useIlluminate\Database\Eloquent\Model;

classResultextendsModel
{
useHasFactory;

protected $table = 'result';

protected $fillable = [
    'test_id',
    'user_id',
    'evaluation',
    'anonymous'
];

protected $casts = [
    'created_at' => 'datetime:d.m.Y'
];

functionuser() {
return $this->belongsTo('App\Models\User', 'user_id');
}

functiontest() {
return $this->belongsTo('App\Models\Tests', 'test_id');
}
}

```

```

    }

functionanswer() {
return $this->hasMany('App\Models\ResultQuestion', 'result_id');
}
}

```

User.php

Модель користувачів.

```

<?php

namespaceApp\Models;

useIlluminate\Foundation\Auth\UserasAuthenticatable;
useIlluminate\Notifications\Notifiable;
useLaravel\Passport\HasApiTokens;

classUserextendsAuthenticatable
{
useNotifiable, HasApiTokens;

protected $table = 'users';

protected $fillable = [
    'name',
    'surname',
    'date_birth',
    'sex',
    'email',
    'role_id',
    'photo',
    'password'
];

protected $casts = [
    'created_at' => 'datetime:d.m.Y'
];

protected $hidden = [
    'password'
];

functionrole() {
return $this->belongsTo('App\Models\Roles', 'role_id');
}

functiontests() {
return $this->HasMany('App\Models\Tests', 'user_id');
}

functionresults() {
return $this->HasMany('App\Models\Result', 'user_id');
}
}

```

TestController.php

Контролер тестів.

```

<?php

```



```

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\Tests;
use App\Models\Steps;
use App\Models\Scores;
use App\Models\Result;
use App\Models\ResultQuestion;

use Illuminate\Support\Facades\Auth;

class TestController extends Controller
{
    protected $publicStorage = "/uploads/";

    function saveImg(Request $request) {
        $name = $this->publicStorage . uniqid().'.'.$request['img']->getClientOriginalExtension();
        $request['img']->move(public_path().$this->publicStorage, $name);
        return response($name);
    }

    function deleteTest($id) {
        Tests::find($id)->delete();
        Result::where('test_id', $id)->delete();
        return response('ok', 200);
    }

    function updateTest(Request $request, $id) {
        $modelTest = Tests::find($id);
        $modelTest->update([
            'title' => $request->title,
            'description' => $request->description
        ]);

        Scores::where('test_id', $id)->delete();
        foreach ($request->scores as $key => $value) {
            $modelSteps = new Scores();
            $modelSteps->create([
                'test_id' => $id,
                'title' => $value['title'],
                'evaluation' => $value['evaluation']
            ]);
        }

        Steps::where('test_id', $id)->delete();
        foreach ($request->steps as $key => $value) {
            $modelSteps = new Steps();
            $modelSteps->create([
                'test_id' => $id,
                'title' => $value['title'],
                'type' => $value['type'],
                'data' => $value['data']
            ]);
        }

        return response('ok', 200);
    }

    function postTest(Request $request) {

```

```

$modelTest = newTests();
$responseTest = $modelTest->create([
    'user_id' => Auth::id(),
    'title' => $request->title,
    'description' => $request->description
]);

foreach ($request->scoresas $key => $value) {
    $modelSteps = newScores();
    $modelSteps->create([
        'test_id' => $responseTest['id'],
        'title' => $value['title'],
        'evaluation' => $value['evaluation']
    ]);
}

foreach ($request->stepsas $key => $value) {
    $modelSteps = newSteps();
    $modelSteps->create([
        'test_id' => $responseTest['id'],
        'title' => $value['title'],
        'type' => $value['type'],
        'data' => $value['data']
    ]);
}

returnresponse('ok', 200);
}

functionget() {
    $model = Tests::get();
    returnresponse()->json($model);
}

functiongetMyTests() {
    $model = Tests::with('user');
    // if(Auth::user()->role_id == 2) {
    $model->where('user_id', Auth::id());
    // }
    $model = $model->get();
    returnresponse()->json($model);
}

functiongetResult() {
    $model = Result::with('test', 'user');
    if(Auth::user()->role_id == 2) {
        $id = Auth::id();
        $model->whereHas('test', function($q) use ($id) {
            $q->where('user_id', $id);
        });
    }
    if(Auth::user()->role_id == 3) {
        $model->where('user_id', Auth::id());
    }
    $model = $model->get();
    returnresponse()->json($model);
}

functiongetAllResult() {
    $model = Result::with('test', 'user')->get();
    returnresponse()->json($model);
}

```

```

    }

functiongetResultId($id) {
    $model = Result::with('answer', 'test.user')->find($id);
    returnresponse()->json($model);
}

functiongetId($id) {
    $model = Tests::with('user', 'scores', 'steps')->find($id);
    returnresponse()->json($model);
}

functiondeleteResult($id) {
    Result::find($id)->delete();
    returnresponse('ok', 200);
}

functionsaveResult(Request $request) {
    $modelResult = newResult();
    $response = $modelResult->create([
        'test_id' => $request->id,
        'user_id' => Auth::id(),
        'evaluation' => $request->evaluation,
        'anonymous' => $request->anonymous
    ]);
    foreach ($request->stepsas $key => $step) {
        if($step['type'] != 'result') {
            $modelSteps = newResultQuestion();
            $modelSteps->create([
                'result_id' => $response['id'],
                'title' => $step['title'],
                'type' => $step['type'],
                'data' => $step['data']
            ]);
        }
    }
    returnresponse('ok', 200);
}
}

```

UserController.php

Контролер користувачів.

```

<?php

namespaceApp\Http\Controllers;

useIlluminate\Http\Request;
useIlluminate\Support\Facades\Auth;
useApp\Models\User;
useIlluminate\Support\Facades\Hash;

useImage;

classUserControllerextendsController
{
    protected $publicStorage = "/uploads/";

    functionprofile() {
        $model = User::with('role')->find(Auth::id());
    }
}

```

```

returnresponse()->json($model);
}

functiongetUser($id) {
    $model = User::with('role', 'tests.user', 'results.test', 'results.user')->find($id);
    returnresponse()->json($model);
}

functionupdate(Request $request) {
    $model = User::find(Auth::id());
    $data = $request->all();

    if($request->new_photo != 'null') {
        $name = $this->publicStorage . uniqid() . '.' . $request['new_photo']->getClientOriginalExtension();
        $data['photo'] = $name;
        $this->cropImg(
            $request->file('new_photo'),
            $name,
            $request->cropWidth,
            $request->cropHeight,
            $request->cropLeft,
            $request->cropTop
        );
    }

    if($request->new_password != "") {
        $data['password'] = Hash::make($request->new_password);
    }

    $model->update($data);
    returnresponse('ok', 200);
}

functiongetUsers() {
    $model = User::with('role')->where('role_id', '!=', 1)->get();
    returnresponse()->json($model);
}

functiondeleteUser($id) {
    User::find($id)->delete();
    returnresponse('ok', 200);
}

functioncropImg($file, $name, $width, $height, $left, $top) {
    $image = Image::make($file->getRealPath());
    $image->crop($width, $height, $left, $top)->resize(300, 300, function ($constraint) {
        $constraint->aspectRatio();
    })->save(public_path() . '/' . $name);
}
}

```

CreateTest.vue

Сторінка створення тестів.

```

<template>
<b-containerclass="block p-4">
<h2>Створення гри</h2>
<div class="form-group">
<label>Назва гри</label>
<inputtype="text" placeholder="Введіть текст" v-model="data.title">

```

```

</div>

<div class="form-group">
<label>Опис гри</label>
<textareaplaceholder="Введіть текст" v-model="data.description"></textarea>
</div>

<div class="form-group">
<label>Виведення результату</label>
<p>
Введіть максимальну оцінку при якій буде введено результат. <br>
Наприклад: 90 - відмінно; 69 - задовільно; 29 - незадовільно
</p>
</div>

<div class="form-group">
<b-row>
<b-colcols="9">
<label>Варіанти відповідей</label>
</b-col>
<b-colcols="2">
<label>Мінімальна оцінка</label>
</b-col>
<b-colcols="1"></b-col>
</b-row>
<b-row v-for="(score, scoreIndex) in data.scores" :key="scoreIndex" class="mb-2">
<b-colcols="9">
<inputtype="text" v-model="score.title">
</b-col>
<b-colcols="2">
<inputtype="text" v-model="score.evaluation">
</b-col>
<b-colcols="1">
<buttonclass="trash" @click="deleteItem('scores', scoreIndex)"><b-iconicon="trash-fill" aria-hidden="true"></b-
icon></button>
</b-col>
</b-row>
</div>

<buttonclass="add-img" @click="addItem('scores', scoreDefault)">Додати рекомендацію</button>

<hr>

<div v-for="(step, stepIndex) in data.steps" :key="stepIndex">
<h3>Розробка - ЕТАП {{ stepIndex + 1 }}</h3>

<div class="form-group">
<b-row>
<b-colcols="6">
<label>Назва етапу</label>
</b-col>
<b-colcols="2"></b-col>
<b-colcols="1"></b-col>
<b-colcols="3"></b-col>
</b-row>
<b-row>
<b-colcols="6">
<inputtype="text" v-model="step.title">
</b-col>
<b-colcols="2">
<select v-model="step.questionType" :disabled="step.data.length > 0">

```

```

<optionvalue="video">Відео</option>
<optionvalue="tests">Тести</option>
<optionvalue="inputs">Відкриті питання</option>
<optionvalue="recommendation">Рекомендація</option>
</select>
</b-col>
<b-colcols="1">
<inputtype="text" v-model="step.count" :disabled="step.questionType == 'video' || step.questionType ==
'recommendation">
</b-col>
<b-colcols="2">
<buttonclass="add-question" @click="addQuestion(stepIndex)">Додати</button>
</b-col>
<b-colcols="1">
<buttonclass="trash" @click="deleteItem('steps', stepIndex)"><b-iconicon="trash-fill" aria-hidden="true"></b-
icon></button>
</b-col>
</b-row>
</div>

<hr>

<div class="form-group" v-if="step.type == 'tests'">
<b-row>
<b-colcols="12" v-for="(test, testIndex) instep.data" :key="testIndex" class="mb-4">
<b-row>
<b-colcols="9">
<label>Запитання {{ testIndex + 1 }}</label>
</b-col>
<b-colcols="2"></b-col>
<b-colcols="1"></b-col>
</b-row>
<b-rowclass="mb-3">
<b-colcols="9">
<inputtype="text" placeholder="Введіть текст" v-model="test.text">
</b-col>
<b-colcols="2">
<buttonclass="add-question" @click="addVariant(stepIndex, testIndex)">Додати</button>
</b-col>
<b-colcols="1">
<buttonclass="trash" @click="delTest(stepIndex, testIndex)"><b-iconicon="trash-fill" aria-hidden="true"></b-
icon></button>
</b-col>
</b-row>
<b-row>
<b-colcols="9">
<label>Варіанти відповідей</label>
</b-col>
<b-colcols="2">
<label>Вара</label>
</b-col>
<b-colcols="1"></b-col>
</b-row>
<b-row v-for="(variant, variantIndex) instest.variants" :key="variantIndex" class="mb-2">
<b-colcols="9">
<inputtype="text" v-model="variant.title">
</b-col>
<b-colcols="2">
<inputtype="text" v-model="variant.evaluation">
</b-col>
<b-colcols="1">

```

```

<buttonclass="trush" @click="delVariant(stepIndex, testIndex, variantIndex)"><b-iconicon="trash-fill" aria-
hidden="true"></b-icon></button>
</b-col>
</b-row>
<b-row>
<b-col>
<b-form-file
      @change="saveImg($event, stepIndex, testIndex, 'img')"
      :placeholder="test.img ? test.img : 'Додаткове зображення'"
></b-form-file>

</b-col>
<b-col>
<b-form-file
      @change="saveImg($event, stepIndex, testIndex, 'img_female')"
      :placeholder="test.img_female ? test.img_female : 'Додаткове зображення для дівчат'"
></b-form-file>

</b-col>
<b-col>
<b-form-file
      @change="saveImg($event, stepIndex, testIndex, 'img_male')"
      :placeholder="test.img_male ? test.img_male : 'Додаткове зображення для хлопців'"
></b-form-file>

</b-col>
</b-row>
</b-col>
</b-row>
</div>

<div class="form-group" v-if="step.type == 'video'">
<b-row>
<b-col>
<label>Посилання на відео</label>
<inputtype="text" placeholder="Посилання на відео" v-model="step.data.text">
</b-col>
</b-row>
</div>

<div class="form-group" v-if="step.type == 'inputs'">
<b-row>
<b-col>
<label>Запитання</label>
</b-col>
<b-col></b-col>
</b-row>
<b-row v-for="(question, questionIndex) instep.data" :key="questionIndex" class="mb-2">
<b-colcols="11">
<inputtype="text" placeholder="Введіть текст" v-model="question.text">
</b-col>
<b-colcols="1">
<buttonclass="trush" @click="delInput(stepIndex, questionIndex)"><b-iconicon="trash-fill" aria-hidden="true"></b-
icon></button>
</b-col>
</b-row>
</div>

<div class="form-group" v-if="step.type == 'recommendation'">
<b-row>
<b-col>

```

```

<label>Рекомендація</label>
<textareaplaceholder="Введіть текст" v-model="step.data.text"></textarea>
</b-col>
</b-row>
</div>
</div>

<b-alert
  :show="dismissCountDown"
  dismissible
  variant="danger"
  @dismissed="dismissCountDown=0"
  @dismiss-count-down="countDownChanged"
  class="my-3"
>{{ error }}</b-alert>

<b-row>
<b-col>
<buttonclass="add-step py-3" @click="addStep()">Додати етап</button>
</b-col>
<b-col>
<buttonclass="add-question py-3" @click="save()">Зберегти</button>
</b-col>
</b-row>

</b-container>
</template>
<script>
exportdefault {
  data() {
    return {
      error: "",
      dismissSecs: 5,
      dismissCountDown: 0,
      data: {
        title: "",
        description: "",
        scores: [
          {
            title: "",
            evaluation: ""
          }
        ],
        steps: []
      },
      scoreDefault: {
        title: "",
        evaluation: ""
      }
    },
    created() {
      if(this.$route.name === 'edit') {
        this.getData();
      }
    },
    methods: {
      countDownChanged(dismissCountDown) {
        this.dismissCountDown = dismissCountDown
      },

```



```

showAlert() {
this.dismissCountDown = this.dismissSecs
  },
getData() {
axios.get('/api/test/'+this.$route.params.id)
  .then((response) => {
this.data = Object.assign(this.data, response.data);
for (letindex = 0; index<response.data.steps.length; index++) {
this.data.steps[index].questionType = this.data.steps[index].type
  }
  })
  },
saveImg(event, stepIndex, testIndex, field) {
varform = newFormData;
varinput = event.target;
if (input.files&&input.files[0]) {
form.append('img', input.files[0]);
axios.post('/api/save-img', form)
  .then((response) => {
this.data.steps[stepIndex].data[testIndex][field] = response.data;
  });
  }
  },
addStep() {
this.data.steps.push({
title: "",
type: "",
count: 1,
data: []
  });
  },
addQuestion(stepIndex) {
this.data.steps[stepIndex].type = this.data.steps[stepIndex].questionType;
for (letindex = 0; index<this.data.steps[stepIndex].count; index++) {
switch (this.data.steps[stepIndex].questionType) {
case 'tests':
this.data.steps[stepIndex].data.push({
text: "",
variants: [
  {
title: "",
evaluation: 0
  }
  ]
  });
break;
case 'video':
this.data.steps[stepIndex].data = {
text: ""
  };
break;
case 'inputs':
this.data.steps[stepIndex].data.push({
text: ""
  });
break;
case 'recommendation':
this.data.steps[stepIndex].data = {
text: ""
  };
break;

```

```

    }
  }
},
addVariant(stepIndex, testIndex) {
  this.data.steps[stepIndex].data[testIndex].variants.push({
    title: "",
    evaluation: 0
  })
},
delVariant(stepIndex, testIndex, variantIndex) {
  this.data.steps[stepIndex].data[testIndex].variants.splice(variantIndex, 1);
},
delTest(stepIndex, testIndex) {
  this.data.steps[stepIndex].data.splice(testIndex, 1);
},
delInput(stepIndex, questionIndex) {
  this.data.steps[stepIndex].data.splice(questionIndex, 1);
},
// scores
deleteItem(array, index) {
  this.data[array].splice(index, 1);
},
addItem(array, defaultData) {
  this.data[array].push(Object.assign({}, defaultData));
},
save() {
  this.error = "";
  if(!this.data.title) {
    this.showAlert();
    this.error = "Необхідно додати назву!";
    return;
  }
  if(!this.data.description) {
    this.showAlert();
    this.error = "Необхідно додати опис!";
    return;
  }
  if(this.data.scores.length <= 1) {
    this.showAlert();
    this.error = "Необхідно додати хоча б одну рекомендацію!";
    return;
  }
  if(this.data.steps.length == 0) {
    this.showAlert();
    this.error = "Необхідно додати хоча б один етап!";
    return;
  }
  if(this.$route.name == 'edit') {
    axios.post('/api/test/'+this.$route.params.id, this.data)
      .then(() => {
        swal.fire({
          icon: 'success',
          title: 'Збережено!'
        });
      }).catch(() => {
        swal.fire({
          icon: 'error',
          title: 'Помилка!'
        });
      })
  } else {

```

```

axios.post('/api/test', this.data)
  .then(() => {
swal.fire({
icon: 'success',
title: 'Збережено!'
});
}).catch(() => {
swal.fire({
icon: 'error',
title: 'Помилка!'
});
})
}
}
}
}
</script>
<stylelang="css" scoped>
h2 {
padding-top: 50px;
padding-bottom: 20px;
color: #322B57;
text-align: center;
text-transform: uppercase;
font-weight: 900;
font-size: 16px;
}
h3 {
color: #000000;
font-weight: 600;
font-size: 18px;
text-align: center;
padding: 20px 0;
}
button.trush {
height: 100%;
width: 100%;
color: #fff;
border: 0;
background: #B40000;
border-radius: 3px;
}
button.add-question {
height: 100%;
width: 100%;
color: #fff;
border: 0;
background: linear-gradient(93.86deg, #06B685 6.06%, #79DB81 114.45%);
border-radius: 3px;
}
button.add-step {
height: 100%;
width: 100%;
color: #fff;
border: 0;
background: linear-gradient(93.86deg, #9B0582 6.06%, #B20264 114.45%);
border-radius: 3px;
}
button.add-img {
height: 42px;
width: 300px;

```

```

color: #fff;
border: 0;
background: #0012B8;
border-radius: 3px;
}
</style>

```

Test.vue

Сторінка проходження тесту.

```

<template>
<div style="margin-bottom: -50px">
<template v-if="status == null">
<b-containerclass="block p-4">
<Back></Back>
<b-row>
<b-colcols="8">
<div class="text-block">
<div class="title">
Назва
</div>
<div class="content">{{ data.title }}</div>
</div>

<div class="text-block">
<div class="title">
Опис
</div>
<div class="content">{{ data.description }}</div>
</div>

<div class="text-block">
<labelclass="checkbox-block">Пройти гру анонімно
<inputtype="checkbox" v-model="data.anonymous">
<spanclass="checkmark"></span>
</label>
</div>

<buttonclass="start-button" @click="startTest()">Почати гру</button>

</b-col>
<b-colcols="4">
<div class="text-block">
<div class="title">Автор гри</div>
<div class="content">
<router-link :to="/user/"+data.user.id">{{ data.user.name }} {{ data.user.surname }}</router-link>
</div>
</div>

<div class="text-block">
<div class="title">Кількість питань</div>
<div class="content">{{ countUnswers }}</div>
</div>
</b-col>
</b-row>
<br><br>
</b-container>
</template>

<template v-if="status == 'start'">

```

```

<div :style="width:' + (currentStep == (data.steps.length - 1) || data.steps[currentStep].type == 'tests' ? '100%' : '95%')"
class="step-item" v-for="(step, stepIndex) in data.steps" :key="stepIndex" v-show="stepIndex == currentStep">
<h2>ЕТАП {{ stepIndex + 1 }}. {{ step.title ? step.title : stepTitle }}</h2>
<div class="video" v-if="step.type == 'video'">
<iframe height="300px" width="100%" :src="'https://www.youtube.com/embed/' + step.data.text.slice(step.data.text.length -
11, step.data.text.length)" frameborder="0" allowfullscreen></iframe>
</div>

```

```

<div class="test" v-if="step.type == 'tests'" :style="background: url(' + ($store.state.user.sex ?
step.data[currentTest].img_male : step.data[currentTest].img_female) + ') no-repeat">
<b-container>
<b-row style="align-items: center;">
<b-col cols="5">
<div class="left-content">
<p>{{ step.data[currentTest].text }}</p>
<ul>
<li v-for="(variant, variantIndex) in step.data[currentTest].variants" :key="variantIndex" @click="nextTest(step, variant,
variantIndex)">{{ variant.title }}</li>
</ul>
</div>
</b-col>
<b-col cols="7" v-if="step.data[currentTest].img" style="text-align: center">

</b-col>
</b-row>
</b-container>
</div>

```

```

<div class="inputs" v-if="step.type == 'result'">
<b-container>
<div class="text-block">
<div class="title">
Кількість балів: <span>{{ step.text }}</span>
</div>
</div>

```

```

<div class="text-block">
<div class="title mb-2">
Висновок згідно результату
</div>
<div class="content">
<ul>
<li v-for="(score, index) in data.scores" :key="index">{{ score.title }} {{ score.evaluation }} - <span v-
if="data.scores[index + 1]">{{ data.scores[index + 1].evaluation }}</span><span v-else>i більше</span></li>
</ul>
</div>
</div>
</b-container>
</div>

```

```

<div class="inputs" v-if="step.type == 'inputs'">
<b-container>
<b-row>
<b-col cols="6" v-for="(question, indexQuestion) in step.data" :key="indexQuestion">
<div class="form-group">
<label>{{ question.text }}</label>
<input type="text" placeholder="Введіть відповідь" v-model="question.answer">
</div>
</b-col>
</b-row>
</b-container>

```

```

</div>

<div class="inputs" v-if="step.type == 'recommendation'">
  <b-container>
    <p>{{ step.data.text }}</p>
  </b-container>
</div>

<div v-if="currentStep == (data.steps.length - 1)">
  <b-container>
    <button class="end-button" @click="saveResult()">Завершити гру</button>
    <br><br>
  </b-container>
</div>

<div v-if="currentStep < (data.steps.length - 1) && data.steps[currentStep].type != 'tests'" class="next-button"
  @click="currentStep++">
  
</div>
</template>

</div>
</template>
<script>
import Back from '../components/Back';
export default {
  components: {
    Back
  },
  data() {
    return {
      status: null,
      currentStep: null,
      currentTest: 0,
      data: {
        title: "",
        description: "",
        anonymous: false,
        evaluation: 0,
        steps: [],
        scores: [],
        user: {
          name: "",
          surname: ""
        }
      }
    },
  },
  created() {
    this.getData();
  },
  computed: {
    stepTitle() {
      switch (this.data.steps[this.currentStep].type) {
        case 'result':
          return "Результат";
        case 'recommendation':
          return "Рекомендації";
        case 'inputs':

```

```

return "Додаткові запитання";
default:
returnthis.data.title;
  }
  },
countUnswers() {
letcount = 0;
for (letindex = 0; index<this.data.steps.length; index++) {
if(this.data.steps[index].type == 'tests' || this.data.steps[index].type == 'inputs') {
count += this.data.steps[index].data.length
  }
}
returncount;
}
},
methods: {
nextTest(step, variant, index) {
vartest = this.currentTest;
test++;
step.data[test - 1].answer = variant.evaluation;
if(this.currentTest< (step.data.length - 1)) {
this.currentTest++;
  } else {
this.currentStep++;
this.data.steps.splice(this.currentStep, 0, {
type: "result",
text: step.data.map(item =>item.answer).reduce((a, b) => +a + +b, 0)
  });
}
},
getData() {
axios.get('/api/test/'+this.$route.params.id)
  .then((response) => {
this.data = Object.assign(this.data, response.data);
  })
},
startTest() {
this.status = 'start';
this.currentStep = 0;
},
saveResult() {
for (let i = 0; i <this.data.steps.length; i++) {
if(this.data.steps[i].type == 'tests') {
for (let y = 0; y <this.data.steps[i].data.length; y++) {
this.data.evaluation += +this.data.steps[i].data[y].answer;
  }
}
}
}
axios.post('/api/result', this.data)
  .then(() => {
this.$router.push('/results');
  })
}
}
}
</script>
<stylelang="css" scoped>

```

```

.step-item {
background: #fff;
min-height: 400px;

```

```

}
.next-button {
display: flex;
align-items: center;
justify-content: center;
position: absolute;
right: 0;
top: 0;
background: linear-gradient(93.86deg, #9B0582 6.06%, #B20264 114.45%);
width: 5%;
height: 100%;
cursor: pointer;
}
.next-buttonimg {
width: 50%;
}
.step-item h2 {
font-weight: 900;
font-size: 16px;
text-align: center;
text-transform: uppercase;
margin: 0;
padding: 10px 0;
background: #fff;
}
.step-item .inputs p {
background: #fff;
border-radius: 10px;
padding: 18px 27px;
}
.step-item .video {
position: relative;
padding-bottom: 56.25%; /* 16:9 */
height: 0;
}
.step-item .videoiframe {
position: absolute;
top: 0;
left: 0;
width: 100%;
height: 100%;
}
.step-item .test, .step-item .inputs {
background-size: cover !important;
min-height: calc(100vh - 230px) !important;
display: flex;
align-items: center;
}
.step-item .test .left-content {
background: #fff;
border-radius: 10px;
padding: 15px 30px;
margin: 20px 0;
}
.step-item .test .left-content p {
font-weight: 600;
font-size: 18px;
line-height: 22px;
color: #322B57;
margin-bottom: 20px;
}

```



```

.step-item .test .left-contentul {
padding: 0;
margin: 0;
}
.step-item .test .left-contentul li {
padding: 0;
background: #3C0E5A;
border-radius: 10px;
color: #fff;
font-size: 16px;
line-height: 19px;
margin-bottom: 20px;
padding: 18px 27px;
cursor: pointer;
}
.step-item .test .left-contentulli:hover {
background: #6b2699;
}
.text-block {
margin-bottom: 25px;
}
.text-block .title {
font-weight: 600;
font-size: 16px;
}
.text-block .titlespan {
color: #9B0582;
font-weight: 900;
font-size: 18px;
}
.text-block .content {
font-weight: 300;
font-size: 15px;
}
.start-button {
margin-top: 50px;
border: 0;
outline: none;
display: block;
width: 360px;
padding: 12px;
text-align: center;
color: #fff;
font-weight: 600;
font-size: 15px;
background: linear-gradient(93.86deg, #9B0582 6.06%, #B20264 114.45%);
border-radius: 3px;
float: right;
}
.end-button {
margin: 0 auto;
margin-top: 40px;
border: 0;
display: block;
outline: none;
padding: 12px;
text-align: center;
color: #fff;
font-weight: 600;
font-size: 15px;
background: linear-gradient(93.86deg, #9B0582 6.06%, #B20264 114.45%);

```

```

border-radius: 3px;
}
.form-group {
background: #fff;
border-radius: 10px;
padding: 18px 27px;
}
.form-grouplabel {
color: #444150;
font-weight: 600;
font-size: 16px;
line-height: 24px;
}
.form-grouplabelspan {
color: #9B0582;
}
.form-groupinput {
background: #FFFFFF;
padding: 12px 16px;
display: block;
width: 100%;
}
.form-groupinput:focus {
outline: none;
background: rgba(155, 5, 130, 0.1);
padding: 12px 16px;
}

.checkbox-block {
display: block;
position: relative;
padding-left: 35px;
margin-bottom: 12px;
cursor: pointer;
font-weight: 600;
font-size: 16px;
-webkit-user-select: none;
-moz-user-select: none;
-ms-user-select: none;
user-select: none;
}

/* Hidethebrowser'sdefaultcheckbox */
.checkbox-blockinput {
position: absolute;
opacity: 0;
cursor: pointer;
height: 0;
width: 0;
}

/* Create a customcheckbox */
.checkmark {
position: absolute;
top: 0;
left: 0;
height: 25px;
width: 25px;
background-color: #eee;
border-radius: 5px;
}

```

```

/* Onmouse-over, add a greybackgroundcolor */
.checkbox-block:hoverinput ~ .checkmark {
background-color: #ccc;
}

/* Whenthecheckboxischecked, add a bluebackground */
.checkbox-blockinput:checked ~ .checkmark {
background-color: #9B0582;
}

/* Createthecheckmark/indicator (hiddenwhennotchecked) */
.checkmark:after {
content: "";
position: absolute;
display: none;
}

/* Showthecheckmarkwhenchecked */
.checkbox-blockinput:checked ~ .checkmark:after {
display: block;
}

/* Stylethecheckmark/indicator */
.checkbox-block .checkmark:after {
left: 9px;
top: 5px;
width: 7px;
height: 12px;
border: solidwhite;
border-width: 0 3px 3px 0;
-webkit-transform: rotate(45deg);
-ms-transform: rotate(45deg);
transform: rotate(45deg);
}
</style>

```

Cabinet.vue

Сторінка кабінету користувача.

```

<template>
<div>
<b-containerclass="block mb-3 px-4 py-5">
<User :user="user" @update="fetchData()"></User>
</b-container>

<b-containerclass="block p-4" v-if="$store.getters.authUser.role_id == 2 || $store.getters.authUser.role_id == 1">
<div class="head">
<h2>Мої ігри</h2>
<div class="filler"></div>
<PurpureLinklink="/results" title="Результати"></PurpureLink>
</div>
<inputtype="text" class="search" placeholder="Пошук" v-model="search">
<b-table-simplehoversmallcaption-topresponsiveclass="mt-4">
<b-thead>
<b-tr>
<b-thclass="py-2 px-3" width="5%">№</b-th>
<b-thclass="py-2 px-3" width="65%">Назва</b-th>
<b-thclass="py-2 px-3" width="20%">Дата створення</b-th>
<b-thclass="py-2 px-3" width="5%"></b-th>
<b-thclass="py-2 px-3" width="5%"></b-th>

```

```

</b-tr>
</b-thead>
<b-tbody>
<b-tr v-for="(item, index) in filteredList" :key="index">
<b-thclass="py-2 px-3">
    {{ index + 1 }}
</b-th>
<b-thclass="py-2 px-3">
<router-link :to="/test/"+item.id">{{ item.title }}</router-link>
</b-th>
<b-thclass="py-2 px-3">
    {{ item.created_at }}
</b-th>
<b-thclass="py-2 px-3">
<router-link :to="/edit/"+item.id">
<b-iconicon="pencil-fill" aria-hidden="true"></b-icon>
</router-link>
</b-th>
<b-thclass="py-2 px-3">
<b-iconicon="trash-fill" style="cursor: pointer" aria-hidden="true" @click="deleteItem(item)"></b-icon>
</b-th>
</b-tr>
</b-tbody>
</b-table-simple>
<Pagination :count="data.length"></Pagination>
</b-container>

<b-containerclass="block p-4" v-if="$store.getters.authUser.role_id === 3">
<div class="head">
<h2>Мої результати</h2>
</div>
<inputtype="text" class="search" placeholder="Пошук" v-model="search">
<b-table-simplehoversmallcaption-topresponsiveclass="mt-4">
<b-thead>
<b-tr>
<b-thclass="py-2 px-3" width="5%">№</b-th>
<b-thclass="py-2 px-3" width="30%">Назва</b-th>
<b-thclass="py-2 px-3" width="30%">Користувач</b-th>
<b-thclass="py-2 px-3" width="10%">Оцінка</b-th>
<b-thclass="py-2 px-3" width="20%">Дата проходження</b-th>
<b-thclass="py-2 px-3" width="5%"></b-th>
</b-tr>
</b-thead>
<b-tbody>
<b-tr v-for="(item, index) in filteredList2" :key="index">
<b-thclass="py-2 px-3">
    {{ index + 1 }}
</b-th>
<b-thclass="py-2 px-3">
<router-link :to="/result/"+item.id" v-if="item.test">{{ item.test.title }}</router-link>
</b-th>
<b-thclass="py-2 px-3">
<span v-if="item.anonymous === 0">{{ item.user.name }} {{ item.user.surname }}</span>
<span v-else>Анонім</span>
</b-th>
<b-thclass="py-2 px-3">
    {{ item.evaluation }}
</b-th>
<b-thclass="py-2 px-3">
    {{ item.created_at }}
</b-th>

```

```

<b-thclass="py-2 px-3">
<b-iconicon="trash-fill" style="cursor: pointer" aria-hidden="true" @click="deleteItem(item)"></b-icon>
</b-th>
</b-tr>
</b-tbody>
</b-table-simple>
<Pagination :count="data.length"></Pagination>
</b-container>
</div>

```

```

</template>
<script>
importUserfrom './components/User';
importPurpureLinkfrom './components/PurpureLink';
importPaginationfrom './components/Pagination';
exportdefault {
  components: {
    User,
    PurpureLink,
    Pagination
  },
  data() {
    return {
      user: {
        name: "",
        surname: "",
        photo: "",
        role: {
          title: ""
        }
      },
      search: "",
      data: []
    },
    computed: {
      filteredList() {
        returnthis.data.filter(item => {
          return (item.title.toLowerCase().includes(this.search.toLowerCase()))
        })
      },
      filteredList2() {
        returnthis.data.filter(item => {
          return (item.test.title.toLowerCase().includes(this.search.toLowerCase()))
        })
      },
      created() {
        this.getData();
        if(this.$store.getters.authUser.role_id == 2 || this.$store.getters.authUser.role_id == 1) {
          this.getMyTest();
        }
        if(this.$store.getters.authUser.role_id == 3) {
          this.getMyResult();
        }
      },
      methods: {
        getData() {
          axios.get('/api/profile')
            .then((response) => {
              this.user = Object.assign(this.user, response.data);
            })
        }
      }
    }
  }
}

```

```

    })
  },
  getMyTest() {
    axios.get('/api/my-tests')
      .then((response) => {
        this.data = response.data;
      })
  },
  getMyResult() {
    axios.get('/api/results')
      .then((response) => {
        this.data = response.data;
      })
  },
  deleteItem(item) {
    axios.delete('/api/test/'+item.id)
      .then(() => {
        var index = this.data.indexOf(item)
        this.data.splice(index, 1)
      })
  },
  watch: {
    $route(to, from) {
      if(this.$store.getters.authUser.role_id == 2 || this.$store.getters.authUser.role_id == 1) {
        this.getMyResult();
      }
      if(this.$store.getters.authUser.role_id == 3) {
        this.getMyTest();
      }
    }
  },
}
</script>
<stylelang="css" scoped>
.head {
display: flex;
margin-bottom: 20px;
}
.head h2 {
color: #322B57;
font-weight: 600;
font-size: 24px;
}
</style>

```