

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра програмного забезпечення автоматизованих систем

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Інженерія програмного забезпечення»

«Допущений до захисту»

Завідувач кафедри

_____ проф. Приходько С.Б.

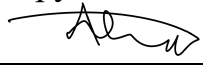
«___» _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти «бакалавр»

на тему: "Розробка програмного забезпечення вебсайту «Quiz»".

Виконав: студент групи 4151

Архіпов Д.В. 

(підпис, ПІБ)

Керівник роботи:

Приходько С.Б.

(посада, науковий ступень вчене звання)

(підпис, ПІБ)

Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра програмного забезпечення автоматизованих систем
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»
Гарант освітньої програми
_____ доц. Макарова
Л.М.
(підпис)
«__» _____ 2022 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти «бакалавр»

Студенту _____ Архипову Денису Владиславовичу _____
(Прізвище, ім'я, по батькові)

1. Тема роботи: "Розробка програмного забезпечення вебсайту «Quiz» ". «Developing the software for the Quiz website»

Керівник роботи _____ Приходько Сергій Борисович

Затверджені наказом ректора №256-уч від «11» травня 2022 р.

2. Термін подання роботи: 2022 р. _____

3. Вихідні дані по роботі: _____

4. Перелік питань, що належать до розробки (найменування розділів) _____
Титульний аркуш; Завдання на кваліфікаційну роботу; Анотація (українською та англійською мовами); Зміст; Перелік умовних позначень, символів, одиниць та термінів (при необхідності); Вступ; Опис предметної галузі та постановка задачі; Проект програмного забезпечення (ескізний, технічний та робочий); Результати розробки програмного забезпечення; Розділ з охорони праці; Висновки; Список використаних джерел; Додатки (технічне завдання, текст програми, опис програми, інструкція користувача, програма та методика випробувань програмного забезпечення).

5. Перелік презентаційних матеріалів Тема дипломного проєкту, Мета дипломного проєкту, Функції програмного забезпечення, Аналіз існуючих технологій вирішення задачі, аналіз та порівняння існуючих веб-додатків аналогів, Постановка задачі, Use case diagram, Діаграма діяльності для процесу «Авторизація», Діаграма діяльності для процесу «Створення Квізу», Тестування з боку користувача, Логічна модель БД, Фізична модель БД, Діаграма класів, Результат розробки програмного забезпечення з квіз тематикою, Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Гурець Н.В., ст. викладач		

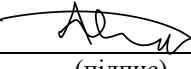
7. Дата видачі завдання 11.05.2022 р. _____

КАЛЕНДАРНИЙ ПЛАН

Номер	Назва етапів роботи	Терміни виконання	Примітка
1.	Підготовка розділу Вступ	24.03.2022	ПП*
2.	Опис та аналіз предметної галузі	28.03.2022	ПП*
3.	Постановка задачі	31.03.2022	ПП*
4.	Ескізний проєкт програмного забезпечення	27.04.2022	
5.	Технічний проєкт програмного забезпечення	06.05.2022	
6.	Робочий проєкт програмного забезпечення	13.05.2022	
7.	Підготовка розділу Результати розробки програмного забезпечення	17.05.2022	
8.	Підготовка розділу з охорони праці	18.05.2022	ПП*
9.	Підготовка розділу Висновки	19.05.2022	
10.	Оформлення списку використаних джерел та додатків	20.05.2022	
11.	Подання на кафедру ПЗАС тексту остаточного варіанту роботи, підписаного її керівником, у роздрукованому та електронному форматі разом із заявами щодо самостійності виконання роботи та ідентичності друкованої та електронної версій роботи (Додатки 1 і 2 «Порядку здійснення заходів з перевірки робіт на наявність текстових збігів/ідентичності/схожості із використанням програмно-технічних засобів», який введений в дію наказом ректора НУК за №20 від 20.01.2020 р.)	10.06.2022	не пізніше, ніж за 14 діб до захисту (згідно п.4.1 зазначеного Порядку)
12.	Підготовка презентації та доповіді	12.06.2022	
13.	Попередній захист роботи на засіданні кафедри ПЗАС	14.06.2022	
14.	Подання на кафедру ПЗАС електронних версій наступних документів у форматі pdf: кваліфікаційної роботи; файлу-опису кваліфікаційної роботи (згідно Додатку до наказу ректора НУК за №287-уч від 19.05.2020 р.); презентації доповіді	24.06.2022	

* - за результатами переддипломної практики (ПП), яка була з 21.03.2022 до 08.05.2022 р.

Студент



 (підпис)

Архіпов Д.В.

(ПБ)

Керівник роботи

 (підпис)

Приходько С.Б.

(ПБ)

АНОТАЦІЯ

Кваліфікаційна робота присвячена розробці програмного забезпечення вебсайту «Quiz».

В ході роботи необхідно провести аналіз предметної галузі, за результатами якого провести постановку задачі, розробити проєкт програмного забезпечення, провести його тестування та визначити норми охорони праці. Розв'язувати задачу інженерії програмного забезпечення процесу створення розважального сервісу з тематикою вікторин.

Мета роботи: розробка програмного забезпечення вебсайту з системою квіз, який дозволить швидко та зручно створювати, редагувати, видаляти нові квізи освітнього та розважального виду.

Задачі роботи: провести аналіз аналогічних вебсайтів та додатків зі сторони користувача; розробити інформаційну модель предметної області та визначити основні напрямлення сайтів вікторин; провести аналіз вимог до програмного забезпечення; виконати проєктування, реалізацію й документування програмного забезпечення.

В роботі застосовані методи теорії прийняття рішень (дерева рішень).

Проєктування програмного забезпечення провести у середовищі draw.io.

Використати середовище розробки PyCharm і СУБД PostgreSQL з використанням DataGrip.

ABSTRACT

The thesis is devoted to the development of software for the website "Quiz".

In the course of the work, it is necessary to analyze the subject area, based on the results of which to set task, develop a software project, test it and determine occupational safety standards. Solve the problem of software engineering in the process of creating an entertainment service with quizzes.

Purpose: development of website software with a system of quizzes, which will allow you to quickly and easily create, edit, delete new quizzes of educational and entertainment type.

Tasks of the work: to analyze similar websites and applications by the user; to develop an information model of the subject area and determine the main directions of quiz sites software design, implementation and documentation were performed software.

The methods of decision theory (decision trees) are used in the work. Design software in draw.io.

Use the PyCharm and PostgreSQL database development environment using DataGrip.

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ВЕБСАЙТІВ З КВІЗ ТЕМАТИКОЮ.....	9
1.1 Процес аналізу предметної галузі	9
1.2. Види квіз-сайтів	9
1.3. Мови програмування.....	11
1.4 Система управління базами даних.....	14
1.5 Аналоги, переваги та недоліки.....	15
1.6. Постановка задачі на розробку проєкту програмного забезпечення вебсайту «Quiz».	32
2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ З КВІЗ ТЕМАТИКОЮ.....	34
2.1 Ескізний проєкт програмного забезпечення	34
2.2 Технічний проєкт програмного забезпечення.....	54
2.3 Робочий проєкт програмного забезпечення.....	56
2.3.3 Випробування програми	61
3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З КВІЗ ТЕМАТИКОЮ.....	68
4 ОХОРОНА ПРАЦІ.....	69
4.1. Основні положення та норми охорони праці.	69
4.2. Аналіз небезпечних і шкідливих факторів під час роботи з екранними пристроями.....	71
4.3. Характеристика умов праці програміста	73
4.4. Забарвлення, коефіцієнти віддзеркалення, освітлення	74
4.5. Параметри мікроклімату. Шум та вібрації.....	76
4.6. Протипожежний захист	77
ВИСНОВКИ.....	80
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	81
ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ	83
ДОДАТОК Б – ТЕКСТ ПРОГРАМИ.....	92

ДОДАТОК В – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ	120
ДОДАТОК Г – ОПИС ПРОГРАМИ	122
ДОДАТОК Д – ІНСТРУКЦІЯ КОРИСТУВАЧА	126

ВСТУП

Вікторина – це вид гри, він полягає у відповіді на питання з різних галузей знань. Зазвичай вікторини відрізняються за правилами, кількістю гравців, способом визначення переможця. Квіз - це тип вікторин який став популярний завдяки інтернету. Дослівно з англійської мови квіз перекладається як вікторина.

Формат квіза також підходить для корпоративних заходів, коли Ви плануєте провести час із користю та не перетворити захід на звичайне застілля. Квіз – це свого роду інтелектуальний тимбілдинг, який допоможе згуртувати колектив, навчить злагоджено працювати у команді та весело провести час разом.

Квіз – це не тільки вікторина, також можна виділити ще кілька видів:

Опитування у цьому квізі немає правильних та неправильних відповідей. Опитування націлене на отримання емоційної реакції, збирання візуальних уподобань або описових характеристик користувача, як у цьому квізі дизайнерської платформи Houzz про інтер'єр.

Тестовий формат часто використовують у рекламних інтеграціях. Не завжди у споживача є уявлення у тому, як працює конкретний продукт чи послуга. Призначення тесту – освітнє. Він допомагає в ігровій формі розповісти про корисні властивості продукту, контекст його використання, можливості самого користувача. У такому форматі можна і розповісти про асортимент страв у меню, допомогти читачкам розібратися в течії фемінізму або переваги різних моделей автомобільного бренду.

Метою роботи є створення розважального квіз-сайту, з спрощеною системою створення своїх квізів ігор.

Актуальність теми полягає у великій сфері застосування квіз сайтів. Сучасний світ за останні роки був змушений перейти на дистанційне спілкування, освіту, роботу. Квіз сайти дозволяють організувати різні освітні або розважальні заходи, не вимагаючи учасникам в одному приміщенні. Останнім часом також активно розвиваються лендінги у форматі квіз. Подання у вигляді інтерактивного

опитування, тесту чи вікторини квіз захоплює увагу потенційного клієнта та стимулює його на конкретні дії.

1 АНАЛІЗ ВЕБСАЙТІВ З КВІЗ ТЕМАТИКОЮ

1.1 Процес аналізу предметної галузі

Процес аналізу предметної області в розробці інформаційних систем передбачає виділення основних і допоміжних бізнес-процесів, які покликані забезпечити виробництво продукту чи послуги. Але, поряд з цим, виділення і розгляд бізнес-процесів надає можливість визначитися з бізнес-елементами та структурами даних, які повинні брати участь в обробці даних. Такі можливості вимагають від розробника інформаційної системи в моделюванні бази даних відштовхуватися не тільки від документів, які використовуються в діяльності предметної області, а й оточення кожного бізнес-процесу і функцій, що включає визначення бізнес-елементів, об'єктів даних, виконавців обробки, власників процесів і функцій, попередніх і наступних функцій, що ініціюють і результируючих подій, інші елементи. Глибина розгляду бізнес-процесів і функцій дає максимально повну інформацію про процеси, що відбуваються в предметній області, і дозволяє краще розуміти завдання, які необхідно реалізувати при розробці веб сервісу, до яких відносяться моделювання структури веб сервісу, проектування застосунку мовою UML, створення мокапів та інше.

1.2. Види квіз-сайтів

У ході вивчення предметної області та поставленої теми, було встановлено, що квіз сайти можуть бути будь-якої тематики та виду від платформи де створюються вікторина на загально популярні теми, новини та події до вузькоспеціалізованих спрямованих на створення контенту з певної теми або області, наприклад: вивчення іноземної мови, математики, космос тощо.

Під час аналізу більшості найпопулярніших сайтів, основною тематикою

яких є вікторини, було встановлено певні види квіз-сайтів.

Існують освітні, розважальні, маркетингові, статистичні види квіз сайтів, а також багато сервісів для самонавчання використовують систему квіз.

Освітні - орієнтуються на учнів шкіл та університетів, а також людей, які займаються самонавчанням. За допомогою таких сайтів можна організувати завдання для вирішення учнів на дистанційному навчанні. По суті більшість таких сайтів є платформами-конструкторами, на яких можна створити свою вікторину для навчання або проходження тесту, після чого надіслати посилання учням або студентам.

Розважальні - насамперед варто розуміти, що розважальні вікторини прийшли в інтернет спираючись на телевізійні аналоги. Наприклад, одна з найпопулярніших платформ "Своя гра" є копією телевізійної версії, в неї гравці можуть завантажувати свої збірники питань різних типів: музичні, зображення, відео, усні, ребуси тощо.

Головна мета розважальних сайтів із квіз системою змагання гравців між собою за більшу кількість правильних відповідей або набрану кількість очок. Зазвичай система має на увазі наявність ведучого і мінімум 2-х гравців. Управління здійснюється із сайту квіз-ігри, де у гравців є кнопка для відповіді, ведучий вислуховує відповідь на поставлене запитання та зараховує або забирає бали.

Маркетингові – квіз-маркетинг, бренди використовують ігрову динаміку квізів, щоб привернути увагу аудиторії та при цьому не відлякати їх прямою рекламою свого продукту. Це високоефективна маркетингова стратегія, користувачі набагато частіше діляться результатами пройденої вікторини, оскільки вона не займає багато часу і допомагає розважитися. Відповідаючи на запитання тесту, людина отримує індивідуальні рекомендації, варіанти розв'язання своєї проблеми за допомогою рекламованого продукту, подарунок чи знижку. За допомогою опитувань, тестів або вікторин розміщених у соціальних мережах, на сторінках сайту, поштових розсилках або на окремо створених квіз-лендінгах виконується робота з розвантаження маркетологів та зниження вартості

клієнта шляхом збільшення часу знаходження користувачів на сайті, короткому огляді на товар чи послугу, велике охоплення рахунків характерного поширення таких квізів, допомагає клієнту ознайомиться з пропозицією і підводить його до покупки.

Статистичні - онлайн вікторини, опитування та різні тести розміщені на популярних інтернет ресурсах допомагають збирати величезну базу даних переваг користувачів, існують сервіси, які за допомогою опитувань створюють аркуш популярності, наприклад письменників, фільмів чи політичних діячів. За своєю суттю будь-який квіз опитувальник чи тест, можна використовувати як статистичний, цим часто користуються сайти з тестами на профорієнтацію, психологію тощо.

Сервіси самонавчання - зазвичай будуються на моделі теоретичні знання + інтерактивні випробування. Це високоефективний метод, допомагає, наприклад, заучувати правила іноземної мови, слова, терміни. Полегшує перевірку помилок та оцінку свого результату. Такі сайти дозволяють вибрати план навчання в області, що цікавиться, і проходити тести у вільний час, для вивчення нової інформації та закріплення вже вивченого матеріалу. У таких додатках та сайтах часто використовується гейміфікація, коли користувачі за проходження завдання та щоденну старанність отримують бали, бонуси, рейтинг та покращують свою шкалу прогресу, можливі навіть змагальні елементи, у деяких додатках можна змагатися з іншими користувачами у знанні вивченого матеріалу. Такі змагання відбуваються в онлайн форматі й на них потрібно вирішувати тести за відведений час, отримуючи очки за перемогу в загальному заліку або втрачаючи їх. Таким чином, сервіси для самонавчання адаптували ігровий формат вікторини для утримання уваги користувача та заохочення щоденного користування своїм продуктом.

1.3. Мови програмування.

Для розробки даного програмного забезпечення було вирішено використати

мову програмування python.

Python — це потужна мова програмування, якою легко опанувати. Вона має ефективні структури даних високого рівня та простий, але ефективний підхід до об'єктноорієнтованого програмування. Простий синтаксис, динамічна типізація роблять цю мову ідеальною для новачків і спрощує написання скриптів. У Python дуже велика стандартна бібліотека, що підходить під основні операційні системи. Велика кількість модулів, програм, улітіліт, документацій дозволяють налаштувати під себе середовище розробки. Інтерпретатор мови Python може бути легко розширений функціями та типами даних, розробленими на C чи C++

Серед основних її переваг можна назвати такі:

- чистий синтаксис (для виділення блоків слід використовувати відступи);
- переносність програм;
- стандартний дистрибутив має велику кількість корисних модулів;
- Найзручніші IDE, одним з яких є PyCharm;
- зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор);
- управління пам'яттю - цілком автоматичне, не потрібно хвилюватися щодо розподілу або звільнення пам'яті. Немає загрози “небезпечної посилання”.
- типи пов'язані з об'єктами, а не зі змінними. Це означає, що змінній може бути призначене значення будь-якого типу, і що (наприклад) масив може містити об'єкти різних типів.
- велика спільнота та багато тематичних форумів та чатів.

Всі ці переваги роблять Python ідеальним вибором навіть для створення вебсайту. Один недолік Python, у порівнянні з найбільш традиційними мовами, полягає в тому, що це не цілком компільована мова; замість цього вона частиною трансліює програму до внутрішньої форми байт-коду, і цей байт-код виконується інтерпретатором Python. Однак, у перспективі – сучасні комп'ютери мають так багато невикористовуваного обчислювального потенціалу, що для 90% додатків швидкодія пов'язана з вибором.

HTML - це мова розмітки, використовується для зображення вебсторінки та лежить в основі World Wide Web. HTML заснований на стандарті SGML гіпертекстова мова розмітки документів для їх подання в Web. Стандарти мови HTML, одного з ключових стандартів Web, розробляються і підтримуються консорціумом W3C. Засновником цього міжнародного консорціуму є Тім Бернес Лі. Консорціум крім створення стандартів форматування, є центром розробки (семантична мережа). Засобами мови HTML забезпечується форматна розмітка документів, визначаються гіперзв'язку між документами та / або їх фрагментами [20].

Весь контент HTML-сайти зберігають у статичних файлах. Тобто, сайти на чистому HTML-кодi складаються з окремих сторінок.

Переваги сайтів на чистому HTML-кодi

- менша вага;
- Економія ресурсів сервера;
- не потрібне оновлення двигуна або модулів;
- Високий рівень безпеки;
- Висока стабільність вебсайту;

CSS – це інструмент і одна з основних складових практично будь-якого вебсайту. Без CSS просто неможливо уявити сучасну діяльність веброзробників.

У ньому є мінуси, такі як різне відображення верстки в різних браузерах або застарілих браузера або щільний зв'язок з HTML, через що доводиться виправляти не тільки CSS файли, складність рефакторингу впливає з цього.

Однак плюсів набагато більше, наприклад таких як:

- Простота самої мови CSS разом із принципом відділення оформлення;
- кілька варіантів дизайну сторінки для перегляду на різних пристроях;
- CSS надає додаткові можливості форматування;
- підвищення сумісності з різними платформами шляхом використання веб-стандартів;
- доступність крос-браузерної функціональності;

1.4 Система управління базами даних

Система управління базами даних (СУБД) — це набір програмного забезпечення, який дозволяє працювати з базами даних: створювати, видаляти та редагувати їх.

Що робить СУБД:

- Допомагає вносити нову інформацію. Також структурує інформацію;
- дозволяє швидко оновити інформацію;
- зменшує ризик помилки;
- структура таблиці вже задана при створенні бази, тому інформація потрапить у потрібний розділ;

Основні типи СУБД:

- реляційні - вони є набір таблиць зі стовпцями й рядками;
- об'єктоорієнтовані – вони працюють за принципом об'єктоорієнтованого програмування: вся інформація представлена у формі об'єкта;
- NoSQL, друга назва – нереляційні. У таких базах зберігаються неструктуровані дані; сховища даних. Ці бази націлені на те, щоб максимально швидко виконувати запити та аналізувати їх.

Для проєкту було прийнято рішення використати реляційну базу даних PostgreSQL

PostgreSQL – це реляційна СУБД з відкритим кодом та вільним поширенням. База даних Postgre базується стандартною мовою запитів SQL.

Переваги PostgreSQL:

- успадкування;
- велику кількість документації;
- потужний інструмент;
- підтримує бази даних необмежених розмірів, що дозволяє використовувати її у великих проєктах або на сайтах, де користувачі своїми діями роблять записи в базу даних, наприклад, як створення вікторин користувачами;

- гнучка взаємодія з основними мовами програмування;
- Безплатний;

1.5 Аналоги, переваги та недоліки.

На сьогодні існує багато сервісів та сайтів, основною тематикою яких є вікторини, опитування чи тести. Для отримання кращих результатів у ході аналізу, будуть проаналізовані не тільки сайти розважального характеру, але також і суміжні. У ході аналізу та вивчення аналогічних сервісів, було виявлено, що більшість популярних сервісів недоступна, це вплинуло на кінцеву вибірку розважальних квіз сайтів.

webquiz.net - вебсайт, на якому зібрано велику кількість тематичних ігор, які можна завантажити та використати у "своїй грі" і так само можливо пройти прямо на сайті.

Інтерфейс домашньої сторінки сайту представлений на Рисунку 1.1.

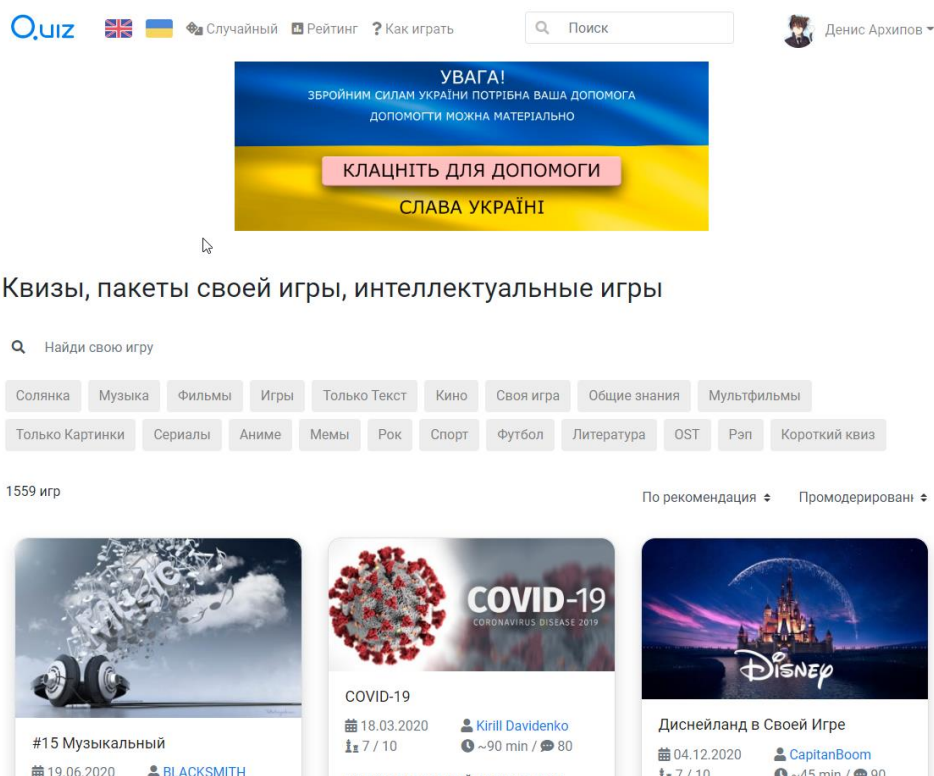


Рисунок 1.1 – Домашня сторінка webquiz.net

Проаналізувавши цей сервіс, ми можемо бачити, що даний сайт надає змогу:

- Пошуку за категоріями або назвою;
- Переглянути рейтинг гравців, приклад на Рисунку 1.2;

Там побачимо таблицю з рейтингом гравців за різний проміжок часу, ім'я гравця, кількість квізів, кількість балів, правильних відповідей

- Обрати гру з плитки на сторінці;
- Правила гри. Приклад на Рисунку 1.3;

Сторінка з правилами гри оформлена за допомогою картинок та пояснення до них.

- Обрати гру навмання;
- Профіль користувача. Рисунок 1.4;

У профілі користувача ми можемо побачити списки квізів (завершено, не завершено, мої квізи, вибрані, скарги на мої квізи та мої скарги), кнопку редагування профілю (аватар, підпис)

Також з іконки профілю можна вийти або перейти у створення квізів. Приклад сторінки створення квізів на Рисунку 1.5. На сторінці створення свого квізу, знаходяться правила та описання умовних позначень, а також кнопки управління, які потрібні для додавання нових блоків (тег, тема, додавання зображення, аудіо, медіа, зберегти).

Рейтинг ТОП 100

Рейтинг гравців за тиждень					
Рейтинг гравців за місяць					
Рейтинг гравців за весь час					
Рейтинг Авторів					
#	Ім'я	Зіграні квізи	Вірних відповідей	Балів	Рейтинг
1	 хоровод1	10	541	179,959	243
2	Гість #494691934	6	414	228,238	225
3	Гість #4175122526	8	435	250,173	199
4	 Александр Королев	2	217	134,790	179
5	 Сергей Михайлов	7	309	180,154	167
6	Гість #2044907848	3	250	35,491	152
7	 Anastasia Nichiporovich	6	289	98,148	98
8	Гість #3785883236	1	102	24,391	81
9	Гість #82825670	2	169	124,134	68
10	 Ruslan Bondar	3	123	11,262	66
11	 Anna Gurevich	1	80	40,636	61
12	Гість #1875809941	2	117	63,347	55
13	 Ivan Ivanov	1	77	13,641	44
14	Гість #3339669622	1	66	18,333	42

Рисунок 1.2 – Сторінка с рейтингом гравців

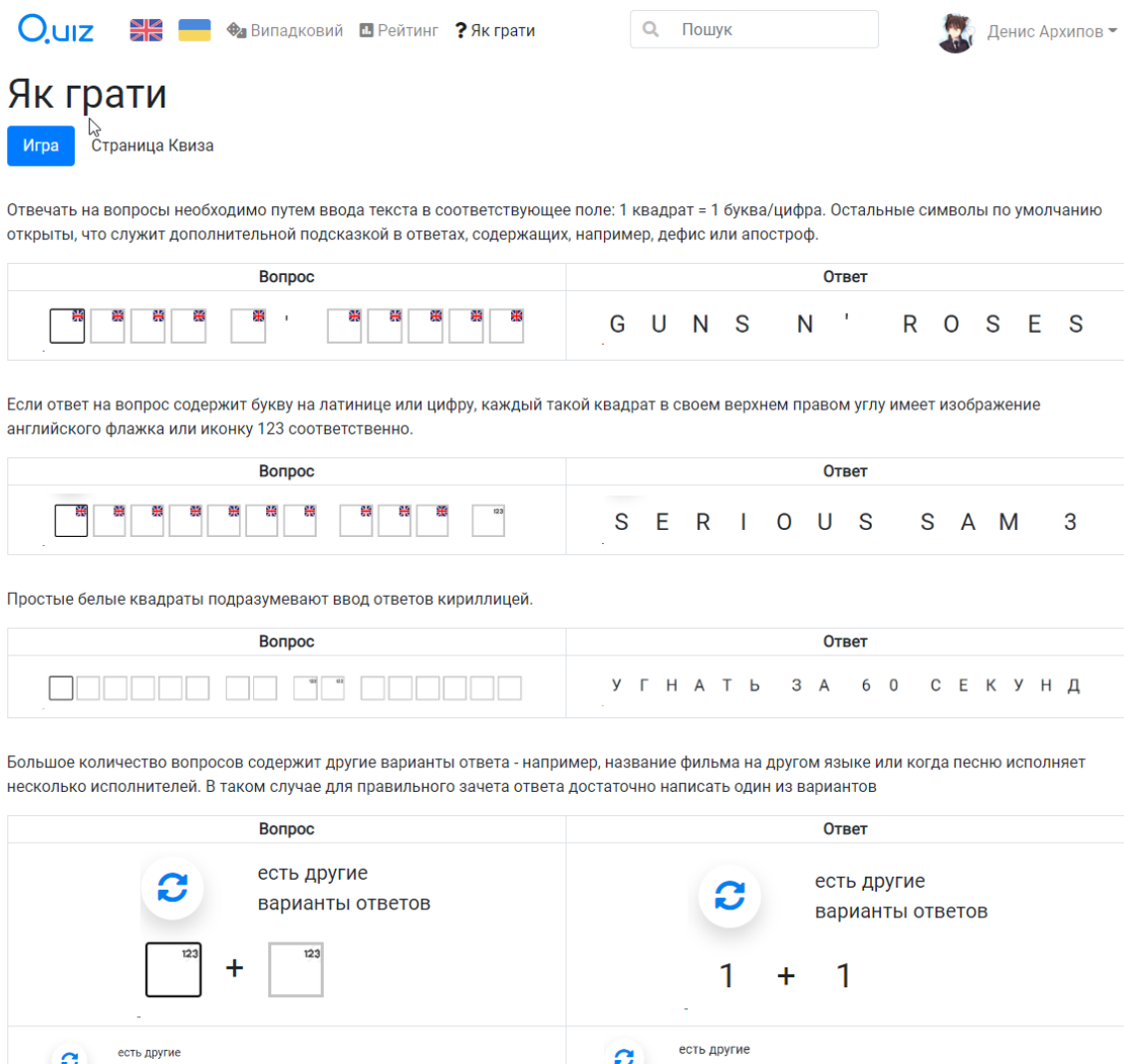


Рисунок 1.3 – Сторінка правил гри

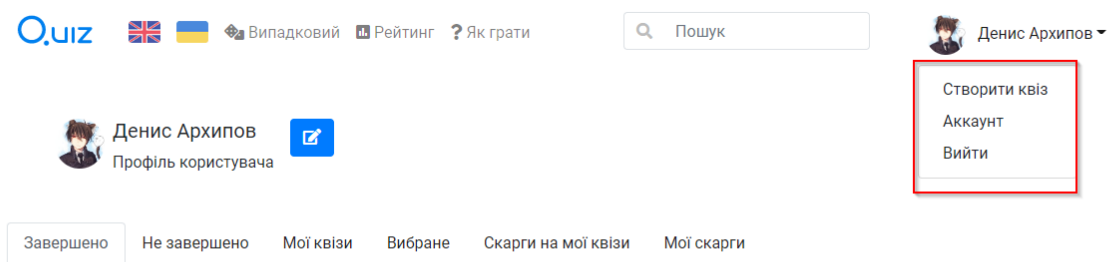


Рисунок 1.4 – Профіль користувача

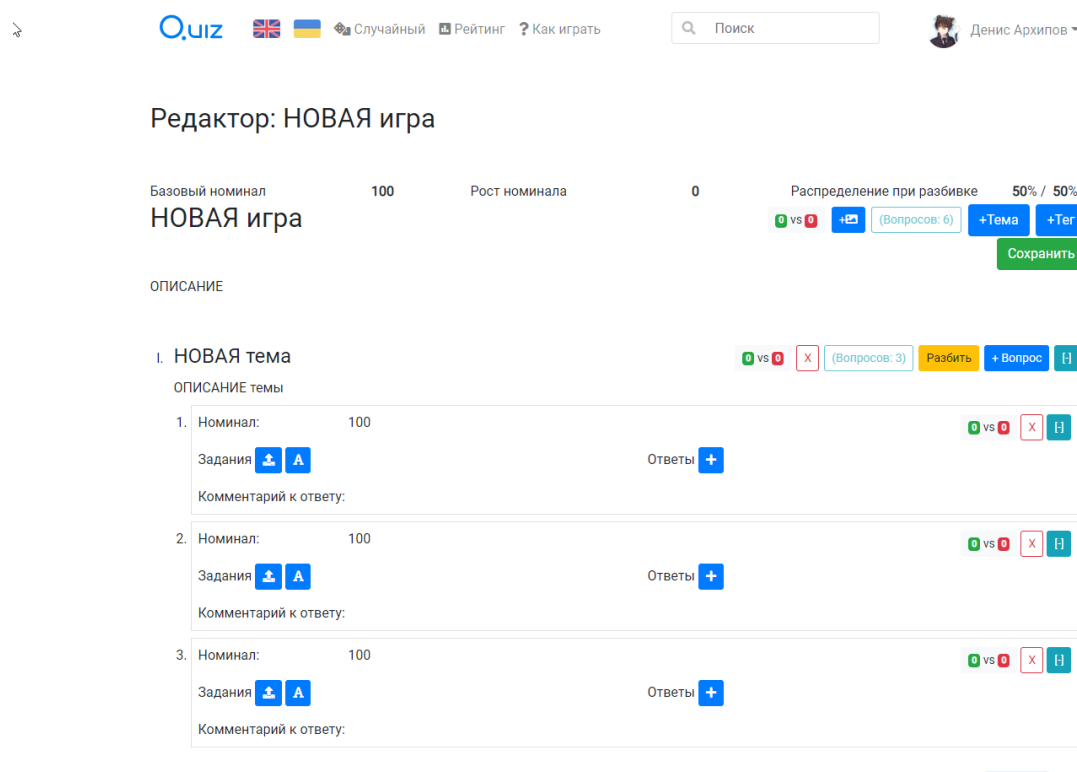


Рисунок 1.5 – Сторінка створення квізу

На сторінці з відкритим квізом знаходиться багато інформації, такої як: тип квізу, категорії, назва, приблизна тривалість гри, дата створення, складність, автор, переважний тип завдань). Приклад сторінки квізу на Рисунку 1.6.

У процесі гри потрібно вирішити завдання і вписати його у квадратні секції внизу сторінки, так само є можливість перейти до іншого завдання, натиснути на кнопку "пас". Приклад сторінки з процесом гри Рисунок 1.7

Поминки у Деда

УВАГА!
ЗЕРОЙНИМ СИЛАМ УКРАЇНИ ПОТРІБНА ВАША ДОПОМОГА
ДОПОМОГТИ МОЖНА МАТЕРІАЛЬНО

КЛАЦНІТЬ ДЛЯ ДОПОМОГИ

СЛАВА УКРАЇНІ

★★★★★

Іграти

Інформація про гру



Создано
23.03.2021

Автор
6axXx14

Сложность: 6 / 10

Продолжительность: ~20 min

Солянка

- Видео
0 вопросов
- Аудио
33 вопросов
- Картинки
52 вопросов
- Текст
0 вопросов



Вопросов: 85

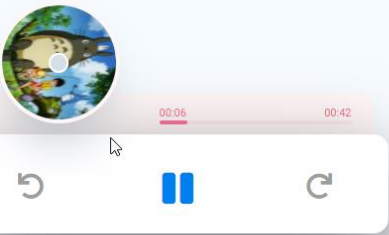
Сыграли: 50

Скачали: 148

Категорія	Вопросов
Фильмы	6
Игры	6
Серики	6
Чудеса света	7
Мультфильмы	6
Рок	6
Реп	5
Классическое	5
Все девочки школы плачут под это	5
Немного опенинг онимэ	5
Магнитофон Бати	7
Игры	6
Аниме	5
Mobile Gaeming	5
Джетикс	5

Рисунок 1.6 – Сторінка квізу

3. Аниме на русском тв по опенингу (10)



Назовите аниме.

Пас

ОТВЕТ

Рисунок 1.7 – Сторінка з процесом гри

Також, даний сервіс має і свої недоліки, наприклад:

- Створення квізу реалізовано повністю на одній сторінці, тому при створенні пакета гри з більш ніж 3-4 раундами гортати таку велику сторінку стає незручно;
- Весь сервіс зріс тільки на одиночну гру, рейтингова система швидко набридає і не викликає мотивації, через те, що немає жодних нагород, наприклад системи досягнень;
- Відсутність темної теми в сучасному світі темна тема сайту є вже обов'язковим елементом дизайну.

vladimirkhil.com/si/online/ - SiGame або «Своя гра». Створена по аналогу однойменної телепередачі. Повністю розважальний продукт, у якому можливо вибрати пакет (набір), і брати участь в іграх з іншими гравцями чи комп'ютером.

Паки з питаннями для цієї гри створюються в окремому додатку, мають формат файлів .siq. Саме додаток є застарілим та не зручним для користувачів, які не знайомі з ним, так що його дизайн теж застарілий і сама ідея в створенні контенту для гри в у іншому додатку виглядає не практично та відштовхує потенціальних користувачів. Приклад інтерфейсу редактору наборів на Рисунок

1.8

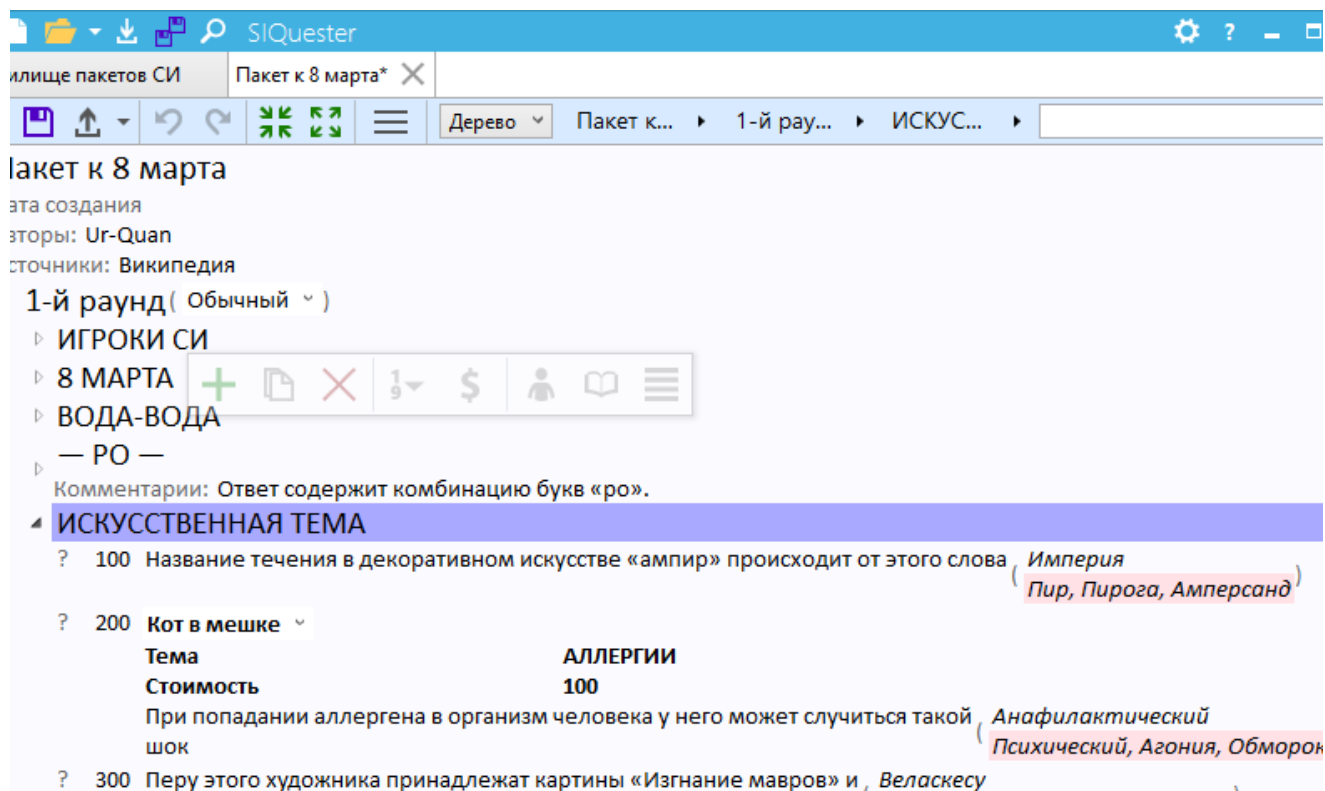


Рисунок 1.8 – Редактор SiGame

Розглянемо інтерфейс SiGame. При переході на сайт нас зустрічає вікно, де потрібно ввести ім'я користувача і відразу можна увійти в гру. Також є сторінка з описом товару. Вікно входу зображено на Рисунку 1.9.

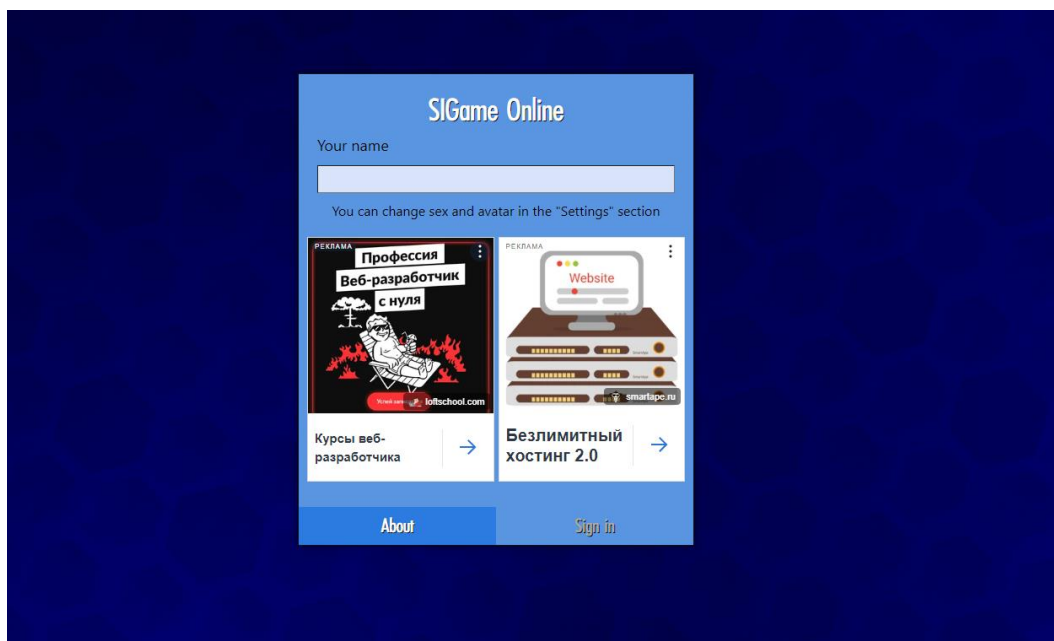


Рисунок 1.9 – Вікно входу SiGame

Після входу на сайт (введення імені) нас зустрічає головне меню, приклад на Рисунку 1.10, воно складається з інформації про сервер та 4 пункти:

- Грати одному (з ботами).

При переході на цю сторінку нам відкривається сторінка створення гри з ботами. Рисунку 1.11. Там ми вибираємо ім'я гри, правила гри (класичні або спрощені), роль творця гри (спостерігач, гравець або ведучий), кількість ботів, а також ігровий набір, можна вибрати випадкові, завантажити файл або відкрити бібліотеку з наборами. Бібліотека з наборами містить фільтри та методи сортування за авторами, категоріями, складністю, темами. Так само на сторінці бібліотеки розміщуються набори та показуватися як плитки, з характеристикою квізу. Приклад бібліотеки наборів (квізів) показано на Рисунку 1.12;

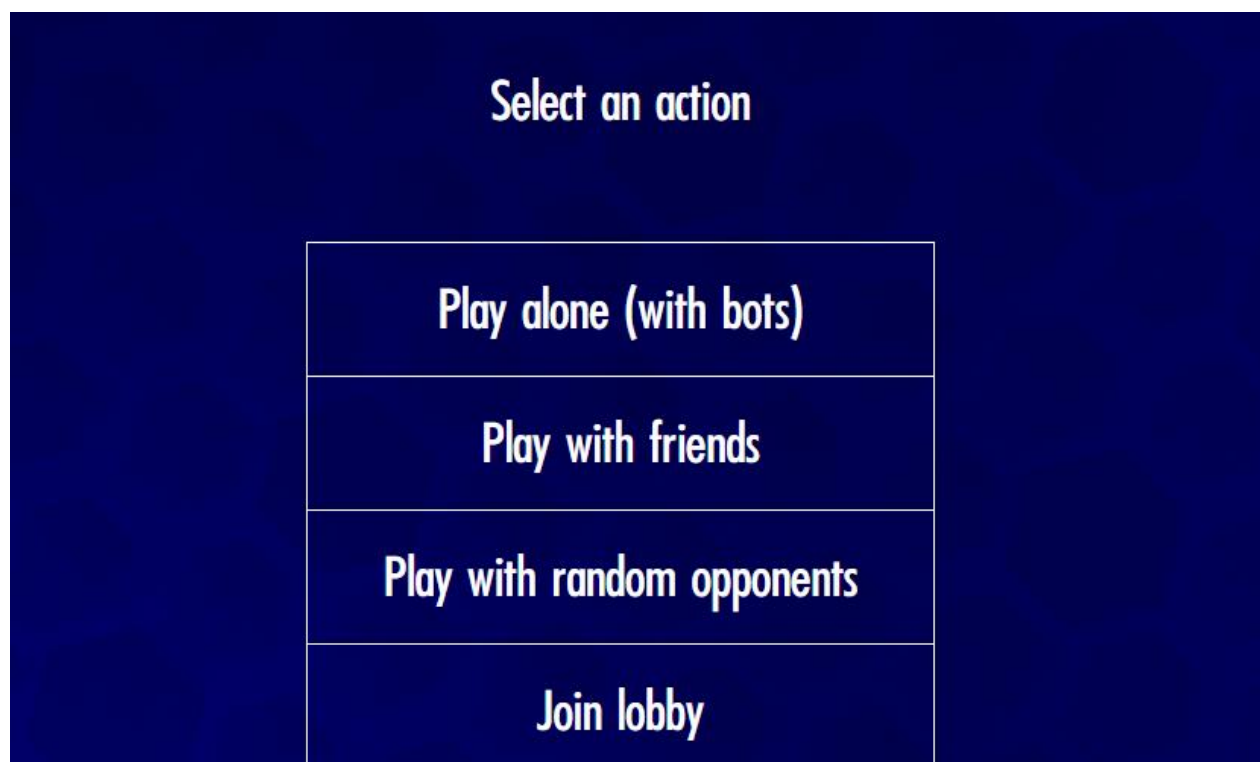


Рисунок 1.10 – Головне меню SiGame

New game

Game name

Question package

Game type

Role

Players

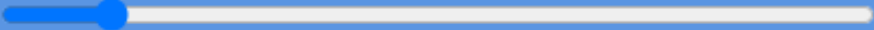
Total: 3 

Рисунок 1.11 – Сторінка створення гри з ботами SiGame

Questions library

Topic: (all) Restriction: (not set)

Difficulty: greater than or equal to 1 Filter:

Publisher: (all) Sort: Name ascending

Author: (all)

Packages (396)

"На вечерок" Topic: "На вечерок" Difficulty: 5 Publisher: Sova igra (Germania) Author: Maldini Restriction: none Publish date: 9/24/2017 Select	«Своя игра» из Германии Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Михаил Перлин, Евгений Копейка, Михаил Лещинер, Николай Пятаев, Александр Ланин и др. Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии 2 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Борис Шойхет и др. Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии 3 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Александр Ланин, Михаил Перлин, Борис Шойхет, Александр Шенгелия, Алёна Шенгелия Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии 4 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Александр Ланин, Михаил Перлин, Борис Шойхет, Александр Шенгелия, Алёна Шенгелия Restriction: none Publish date: 1/1/1970 Select
«Своя игра» из Германии. Игры 10, 11 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Михаил Перлин, Борис Шойхет Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии. Игры 11, 12 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Михаил Перлин, Борис Шойхет Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии. Игры 11, 12 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Михаил Перлин, Борис Шойхет Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии. Игры 14, 15 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Александр Ланин, Михаил Лещинер, Николай Пятаев, Дмитрий Жарков Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии. Игры 16, 17 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Александр Ланин, Михаил Перлин, Борис Шойхет, Александр Шенгелия, Алёна Шенгелия Restriction: none Publish date: 1/1/1970 Select
«Своя игра» из Германии. Игры 19, 20 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Александр Ланин, Михаил Перлин, Борис Шойхет, Николай Пятаев Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии. Игры 2, 3 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Михаил Перлин Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии. Игры 20, 21 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Александр Ланин, Михаил Перлин, Борис Шойхет, Александр Шенгелия, Алёна Шенгелия Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии. Игры 22, 23 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Михаил Перлин Restriction: none Publish date: 1/1/1970 Select	«Своя игра» из Германии. Игры 24, 25 Topic: "Своя игра" Difficulty: 10 Publisher: Sova igra (Germania) Author: Борис Шойхет, Алексей Богословский Restriction: none Publish date: 1/1/1970 Select

Рисунок 1.12 – Бібліотека наборів SiGame

- Грати з друзями.

Меню створення гри з друзями майже не відрізняється від меню створення гри проти ботів. Додана можливість встановити пароль, вибрати хто буде ведучим людина або комп'ютер, а також ще один повзунок кількості учасників. Як і на сторінці створення гри для одного гравця, в нижньому правому кутку сторінки є кнопка налаштувань гри та її правил. Приклад сторінки створення гри з друзями наведено на Рисунку 1.13;

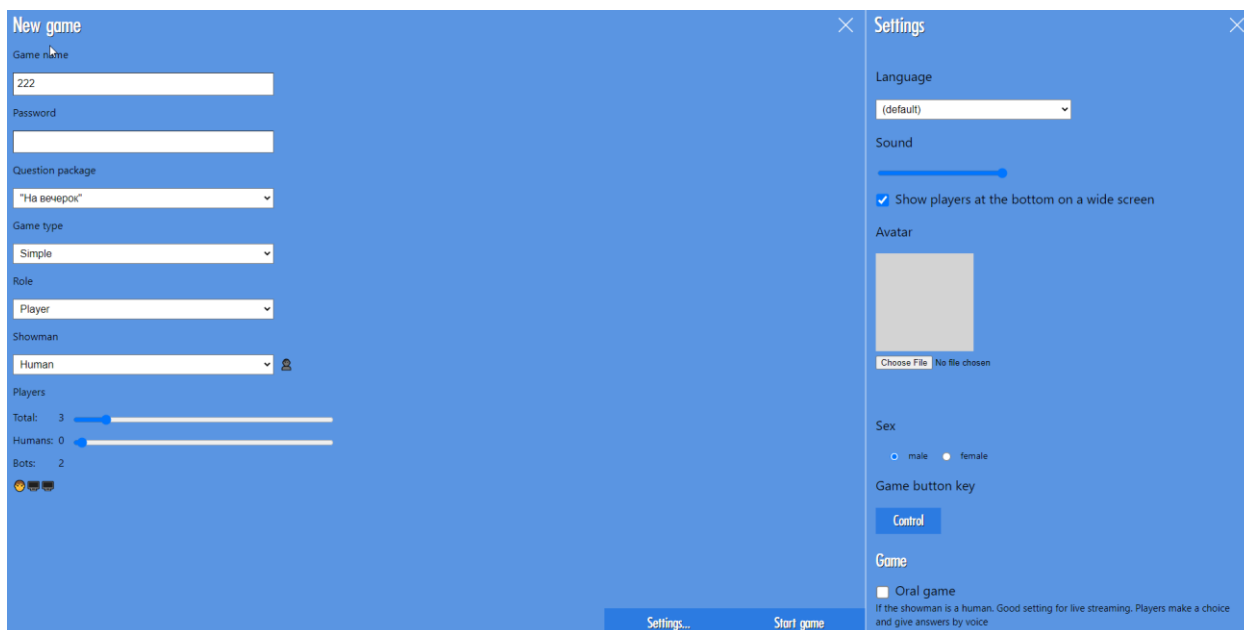


Рисунок 1.13 – Сторінка створення гри з друзями SiGame

- Створити кімнату з випадковими супротивниками.

Після переходу, користувач відразу потрапляє на сторінку гри, сторінка гри всі залежності від режиму майже не відрізняється, є портрети та імена учасників, екран з питанням та категоріями, кількість балів, червона кнопка відповіді та кнопки паузи, виходу та пропуску питання. Приклад сторінки гри на Рисунку 1.14;

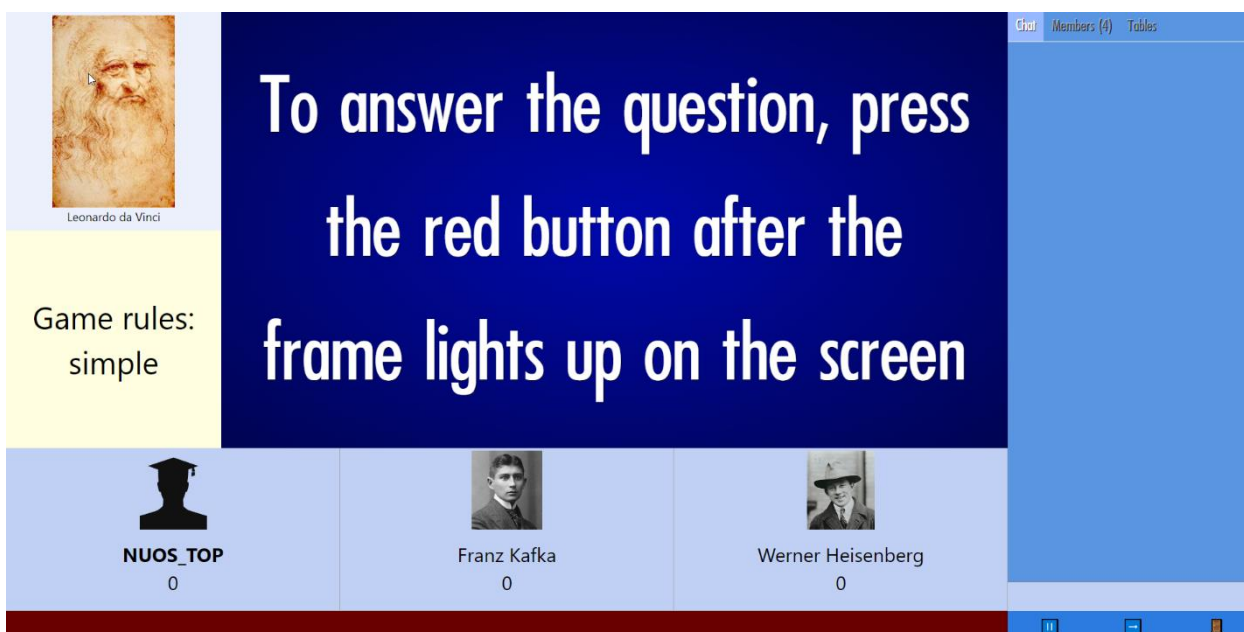


Рисунок 1.14 – Сторінка гри SiGame

- Приєднається до створеного лобі.

На сторінці з лобі, ми можемо побачити список ліворуч, при виділенні кімнати по центру з'являється короткий опис правил і настрою кімнати, праворуч ми бачимо чат для гравців. На цій сторінці можна створити своє лобі або виконати автоматичний пошук опонентів. Інтерфейс цієї сторінки показано на Рисунок 1.15;

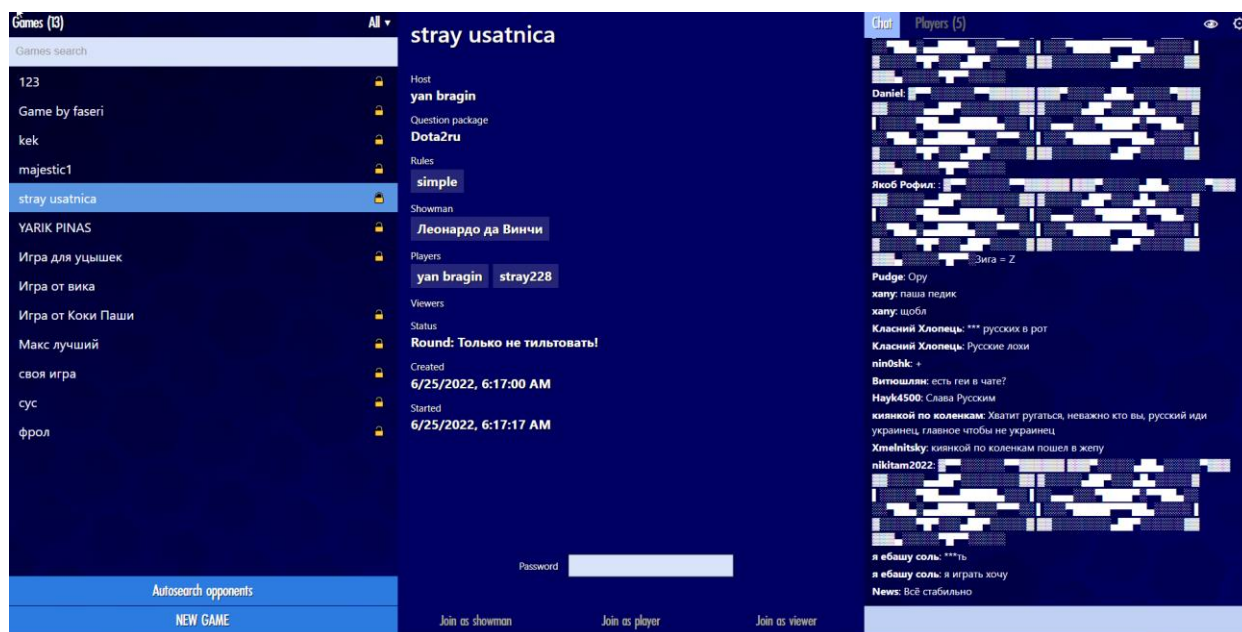


Рисунок 1.15 – Сторінка пошуку лоббі SiGame

В результаті аналізу вебсайту vladimirkhil.com/si/online/ було виявлено такі мінуси:

- Застарілий та не інтуїтивний дизайн
- Безліч зайвих налаштувань та повзунків
- Відсутність профілю користувача
- Створення ігрових квізів в іншому додатку
- Недороблений штучний інтелект у ведучого під керуванням комп'ютера.

Дослідження показало, що даний сервіс користується популярністю та користувачів, пов'язано це в першу чергу з тим, що є можливість грати у компанії друзів. Також є додаток на комп'ютер, воно майже не відрізняється від вебверсії, але підключення гравців виконується локальною мережею до хоста, в ролі якої

виступає комп'ютер творця партії.

playbuzz.com – це онлайн-платформа, яка дозволяє публікувати та створювати різні види квізів (опитування, тести, вікторини, списки-статті та інші). Цей ресурс орієнтований на маркетологів, блогерів та інші компанії. На ньому можна знайти квізи на різні теми. Головна сторінка являє собою рядок пошуку, перемикач мови, кнопку на створення свого квізу, основне місце займає контент із популярними темами та квізами. Приклад наведено на Рисунку 1.16

Самі квізи створюються на платформі app.ex.co і являють собою різні шаблони з блоковим редагуванням, приклад наведений на Рисунку 1.17. Об'ємний, але простий інструментарій, секцію зі створеними сторінками (Рисунок 1.18), вкладку монетизації (Рисунку 1.19), обліковий запис користувача (Рисунку 1.20) та велику базу знань (Рисунку 1.21) для навчання користування редактором.

Сам редактор складається з покрокового блочного проектування, приклад на Рисунку 1.22.

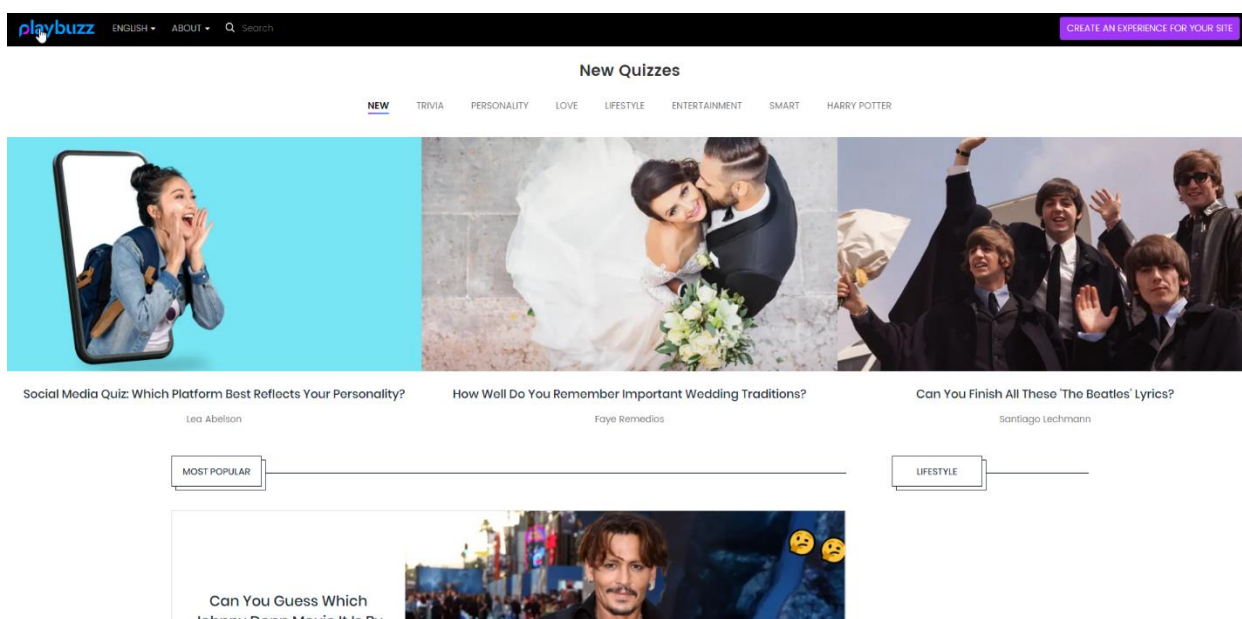


Рисунок 1.16 – Головна сторінка playbuzz.

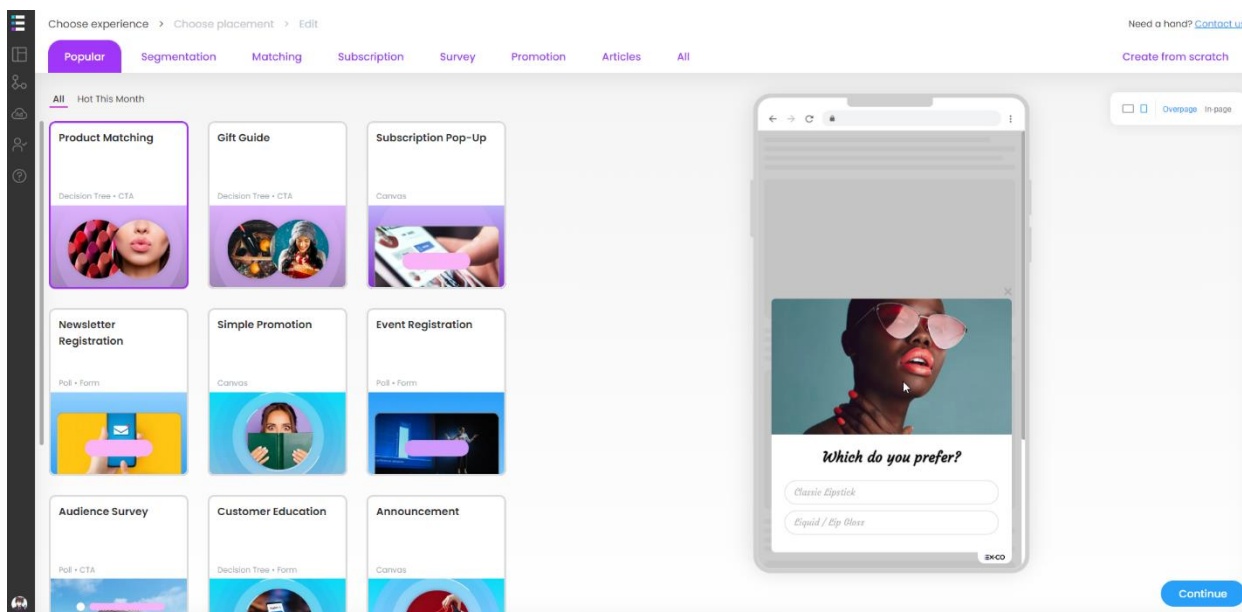


Рисунок 1.17 – Сторінка створення квіз лендінгу app.ex.co

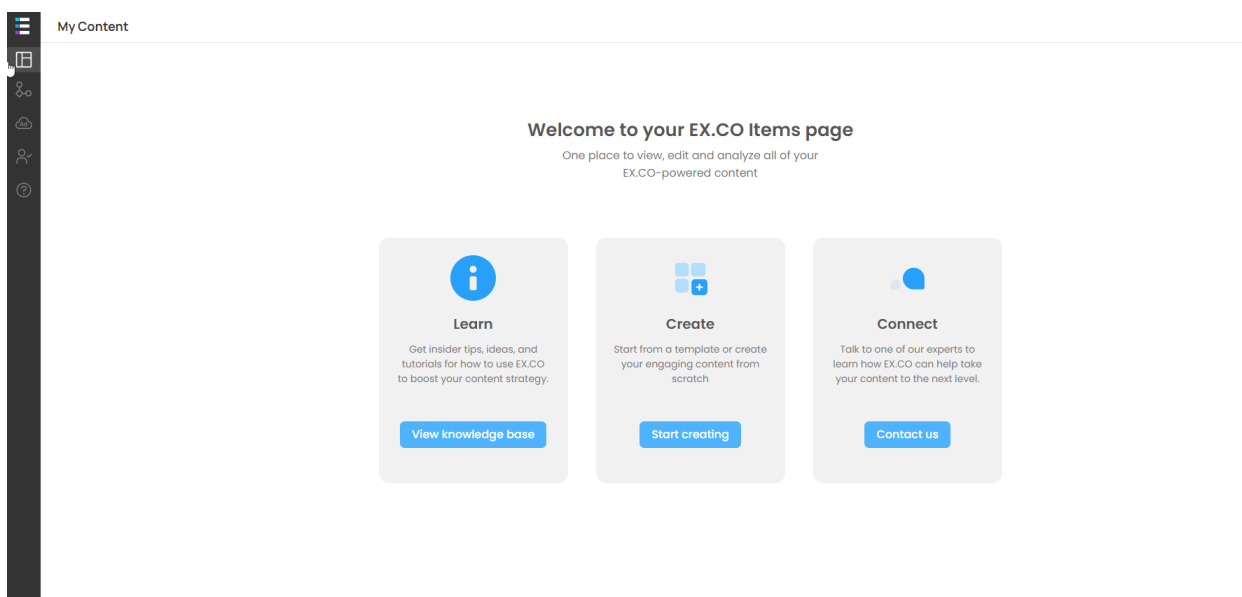


Рисунок 1.18 – Секція створених лендінг-сторінок.

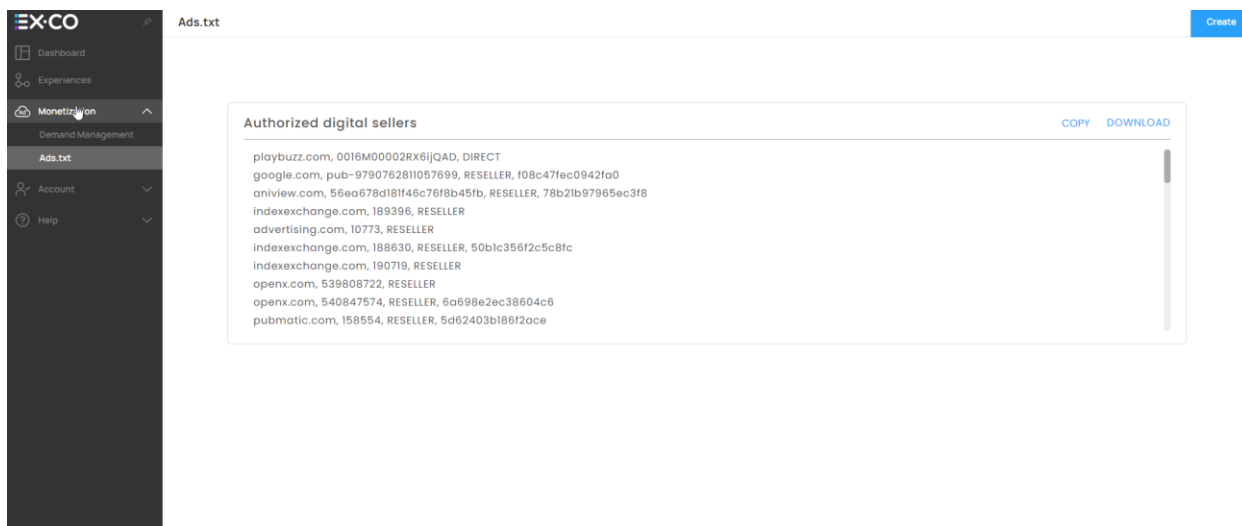


Рисунок 1.19 – Сторінка налаштування монетизації app.ex.co

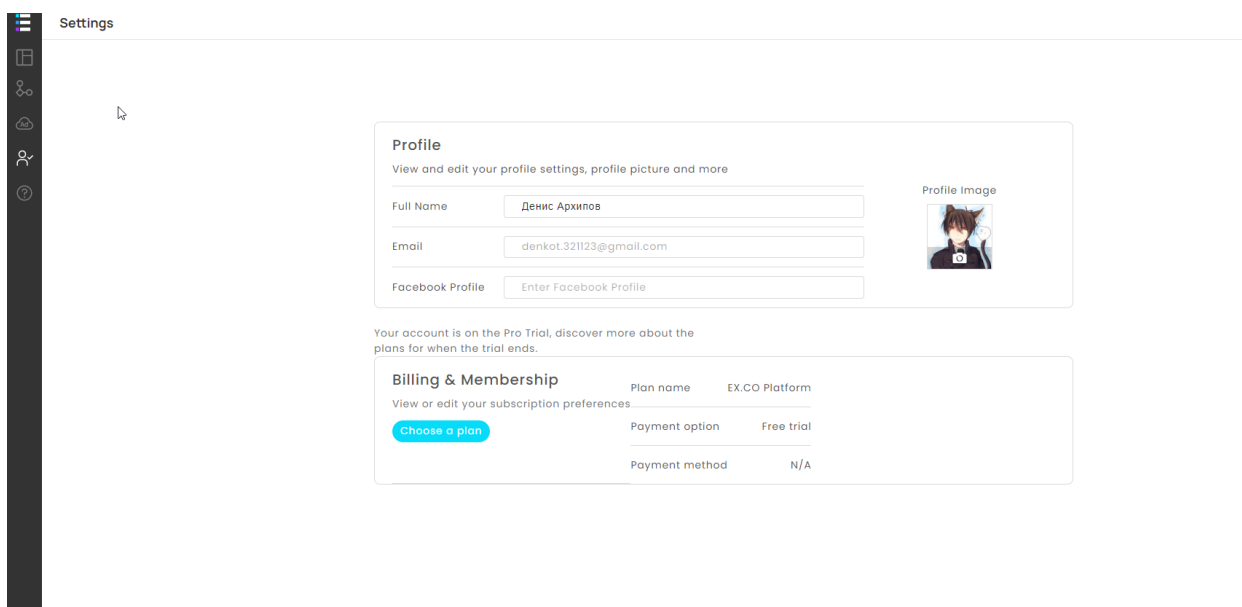


Рисунок 1.20 – Обліковий запис користувача

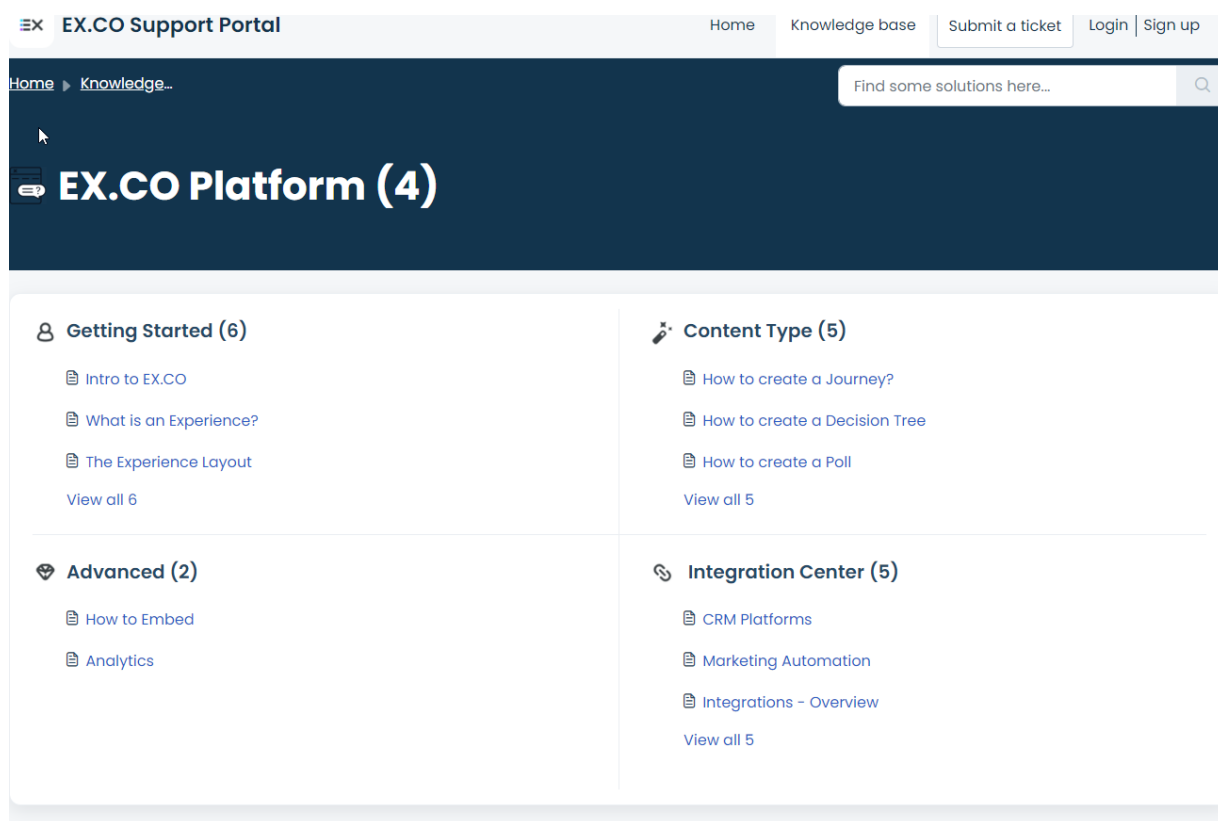


Рисунок 1.21 – База знань ex.co

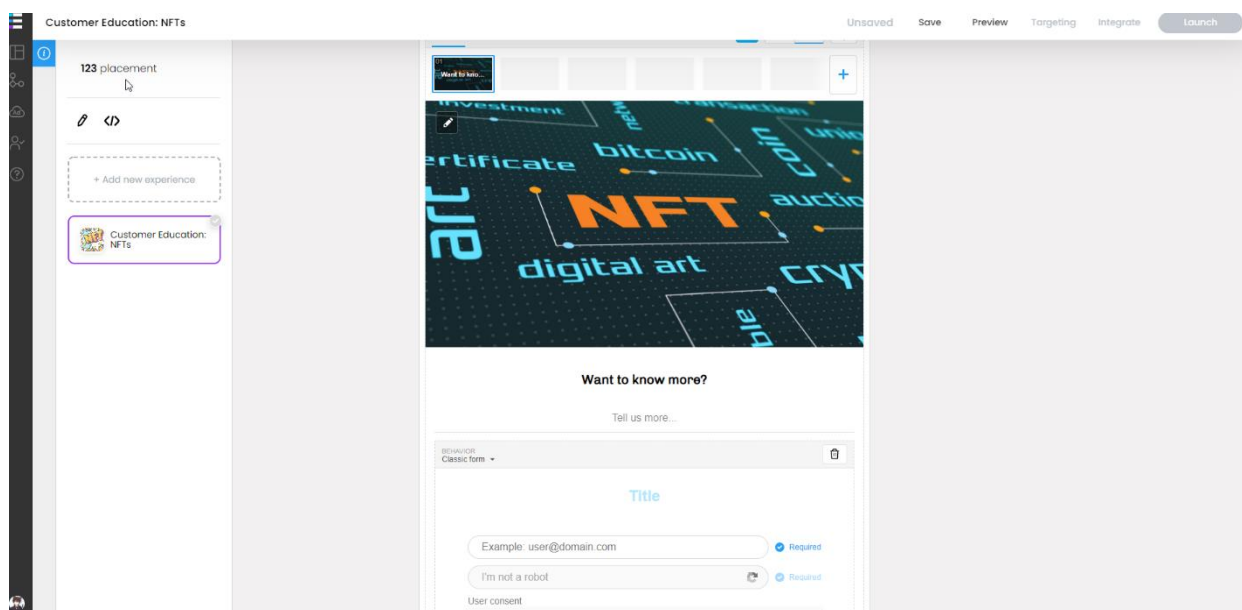


Рисунок 1.22 – Процес блочного створення сторінки

На досліджуваній онлайн платформі можна якісно, швидко та зручно створювати як прості опитування та тести, так і складні квізи для лендінг сторінок. Головне завдання даної платформи – дати можливість користувачеві

створювати свої лендінги для маркетингових цілей. У ході дослідження було отримано знання та досвід роботи в сучасному редакторі зі зручним дизайном.

Посилаючись на основне завдання проєкту було виявлено наступні недоліки:

- складні для новачків інструменти та принципи створення сторінки;
- велика кількість функцій, які необхідно вивчити;
- платна підписка;

1.6. Постановка задачі на розробку проєкту програмного забезпечення вебсайту «Quiz».

Проаналізувавши предметну галузь, наявні аналоги, їх переваги та недоліки, можна зробити висновок, що розробка нового сервісу, який буде мати в собі всі переваги наявних аналогів, та буде позбавлений всіх недоліків, є оправданим та вигідним.

Програмне забезпечення має забезпечувати користувачів можливість виконання перерахованих нижче функцій:

- перегляд ігрових паків та наборів за категоріями (назва набору, опис набору, тривалість гри);
- створення профілю;
- створення свого пакета, набору, квіза (назва набору, дата створення, опис набору, тривалість гри);
- створення питання;
- додавання варіантів відповіді;
- перегляд створених паків, наборів, квізів;
- зручний інтерфейс створення нової гри;
- редагування вже створених ігрових наборів;
- видалення ігрового набору (id набору).

Програмне забезпечення має забезпечувати для модератора можливість виконання перерахованих нижче функцій:

- перегляд списку наборів (назва набору, опис набору, тривалість гри);
- видалення набору (id набору).

Докладніше вимоги до програмного забезпечення наведено у технічному завданні у Додатку А.

2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВЕБСАЙТУ З КВІЗ ТЕМАТИКОЮ

2.1 Ескізний проєкт програмного забезпечення

У результаті проведення аналізу вимог до програмного забезпечення був розроблений ескізний проєкт за допомогою мови моделювання UML. Для визначення та аналізу вимог було розроблено діаграму варіантів використання, а також виконано структурування та специфікацію варіантів використання.

Для формалізації представлення сценаріїв варіантів використання були побудовані діаграми діяльності для кожного з них, а також розроблена інформаційна модель за допомогою діаграми класів.

2.1.1 Побудова моделі варіантів використання

Проаналізувавши функціональні вимоги до програмного забезпечення, були виділені основні варіанти використання:

- авторизація;
- переглянути список квізів;
- створити квіз;
- створити питання;
- редагувати квіз;
- редагувати питання
- видалити квіз;
- обрати квіз для гри;

На рисунку 2.1 наведено діаграму варіантів використання.

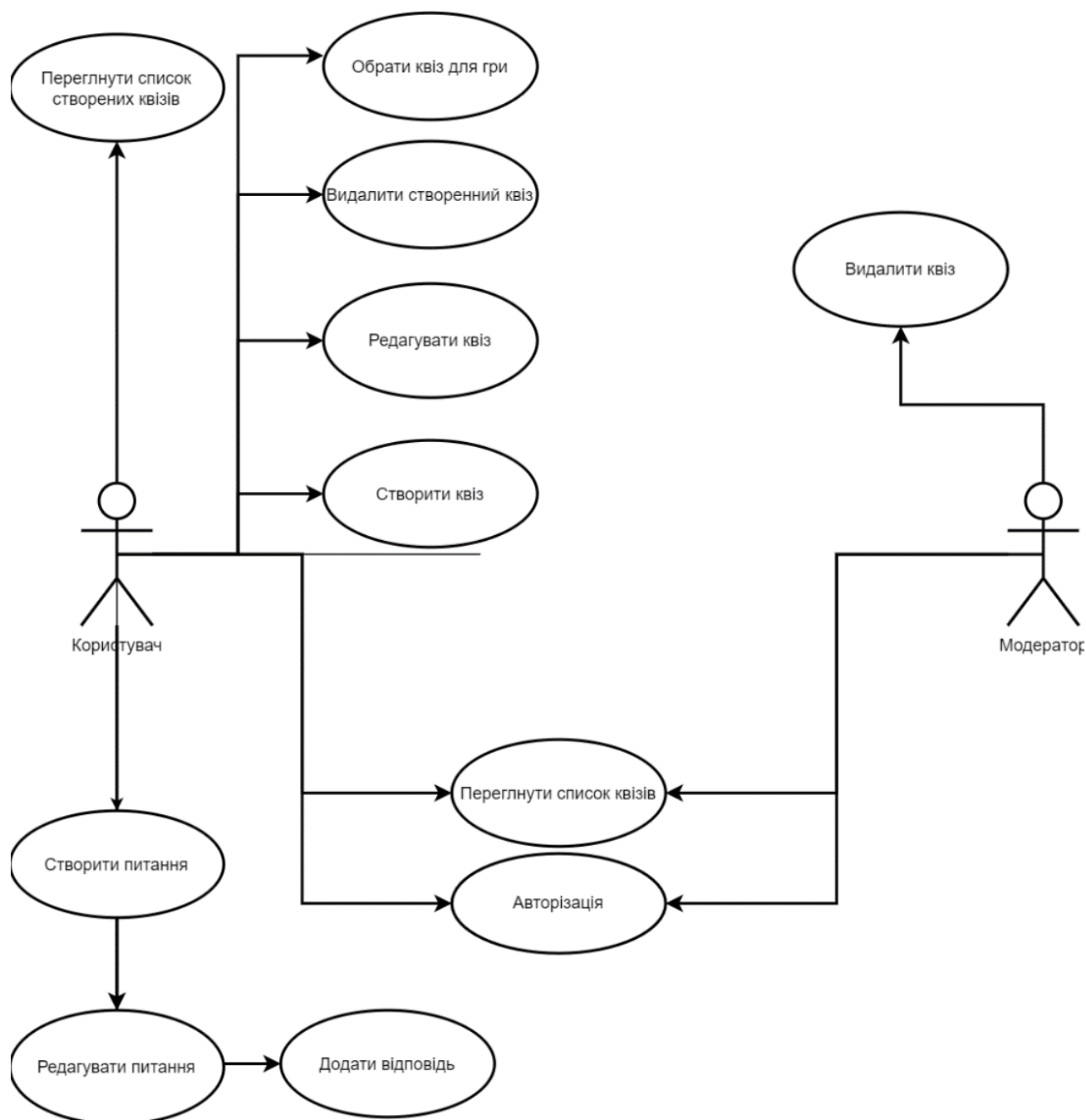


Рисунок 2.1 – Діаграма варіантів використання

Діаграма варіантів використання дуже схожа на графи, вона показує асоціацію між акторами та прецедентами.

2.1.2 Специфікації варіантів використання

Проаналізувавши діаграму варіантів використання, проведемо специфікації для кожного з варіантів використання для користувачів. Варіанти використання

описують не тільки взаємодії між користувачами й сутністю, але також реакції сутності на одержання окремих повідомлень від користувачів і сприйняття цих повідомлень за межами сутності. Варіанти використання можуть містити в собі опис особливостей способів реалізації сервісу й різних виняткових ситуацій, таких як коректна обробка помилок системи. Нижче представлений реєстр варіантів використання, який зображений на таблиці 1.

Таблиця 1 – Реєстр варіантів використання

Код	Основна дійова особа (актор)	Назва варіанту використання	Короткий опис
1	2	3	4
A1	Користувач (гравець)	Переглянути список квізів	Прецедент дозволяє переглянути список квізів, відсортувати за категоріями, здійснити пошук квізів за ключовим словом.
A2	Користувач (гравець)	Створити квіз	Прецедент дозволяє створити квіз.
A3	Користувач (гравець)	Редагувати квіз	Прецедент дозволяє редагувати квіз.
A4	Користувач (гравець)	Видалити квіз	Прецедент дозволяє видалити квіз.
A5	Користувач (гравець)	Створити питання	Прецедент дозволяє створити питання.
A6	Користувач (гравець)	Редагувати питання	Прецедент дозволяє редагувати питання.
A7	Користувач (гравець)	Додати відповідь	Прецедент дозволяє додати відповідь до питання.
A8	Користувач (гравець)	Обрати квіз для гри	Прецедент дозволяє обрати квіз для гри та запустити його.

A9	Користувач (гравець)	Переглянути список створених квізів	Прецедент дозволяє переглянути список квізів
A10	Користувач	Авторизація	Прецедент дозволяє ідентифікувати користувача як гравця або модератора
B1	Модератор	Переглянути список квізів	Прецедент дозволяє переглянути список квізів
B2	Модератор	Видалити квіз	Прецедент дозволяє видалити квіз користувача.

Продовження таблиці 1

Конкретизуємо варіанти використання.

A1. Переглянути список вакансій

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: Модератор.

Передумови: Користувач ініціював перегляд списку квізів.

Короткий опис

Прецедент дозволяє переглянути списки квізів, відсортувати за певними критеріями, здійснити пошук квізу за ключовими словами.

A2 Створити квіз

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював створення квізу.

Короткий опис

Прецедент дозволяє створити квіз.

A3 Редагувати квіз

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював редагування квізу.

Короткий опис

Прецедент дозволяє редагувати квіз.

A4 Видалити резюме

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював видалення створеного квізу.

Короткий опис

Прецедент дозволяє видалити створеного квізу.

A5 Створити питання

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював створення питання.

Короткий опис

Прецедент дозволяє створити питання.

A6 Редагувати питання

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював редагування питання.

Короткий опис

Прецедент дозволяє редагувати питання.

A7 Додати відповідь

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював додавання варіанту відповіді питання.

Короткий опис

Прецедент дозволяє додати відповідь до питання.

A8 Обрати квіз для гри

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював гру квізу зі списку.

Короткий опис

Прецедент дозволяє обрати квіз для гри та запустити його.

A9 Переглянути список створених квізів

Основна дійова особа: Користувач (гравець).

Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював відкриття списку створених квізів.

Короткий опис

Прецедент дозволяє переглянути список створених квізів цим користувачем.

A10 Авторизація

Основна дійова особа: Користувач Інші учасники прецеденту: відсутні. .

Передумови: Користувач ініціював авторизацію на сайті.

Короткий опис

Прецедент дозволяє ідентифікувати користувача як гравця чи модератора.

B1 Переглянути список вакансій

Основна дійова особа: Модератор

Інші учасники прецеденту: гравець. .

Передумови: Модератор ініціював перегляд списку квізів.

Короткий опис

Прецедент дозволяє переглянути список створених користувачами квізів.

B2 Видалити квіз

Основна дійова особа: Модератор.

Інші учасники прецеденту: відсутні. .

Передумови: Модератор ініціював видалення квізу користувача.

Короткий опис

Прецедент дозволяє видалити квіз створений користувачем.

2.1.3 Сценарії варіантів використання

На даній стадії процесу розробки програмного забезпечення створюються діаграми діяльності. Доцільно створити діаграми діяльності для основних варіантів використання.

Діаграма діяльності показує кроки управління від початку до кінця, показуючи різні шляхи прийняття рішень, які існують під час виконання дії. Ми можемо показати як послідовну обробку, так і одночасну обробку діяльності, використовуючи діаграми діяльності. Він використовується в моделюванні бізнесу та процесів, де основним використанням є відображення динамічного характеру системи. Діаграма діяльності дуже схожа на блок-схему.

На рисунку 2.2 наведено діаграму діяльності для варіанту використання «Авторизація»:

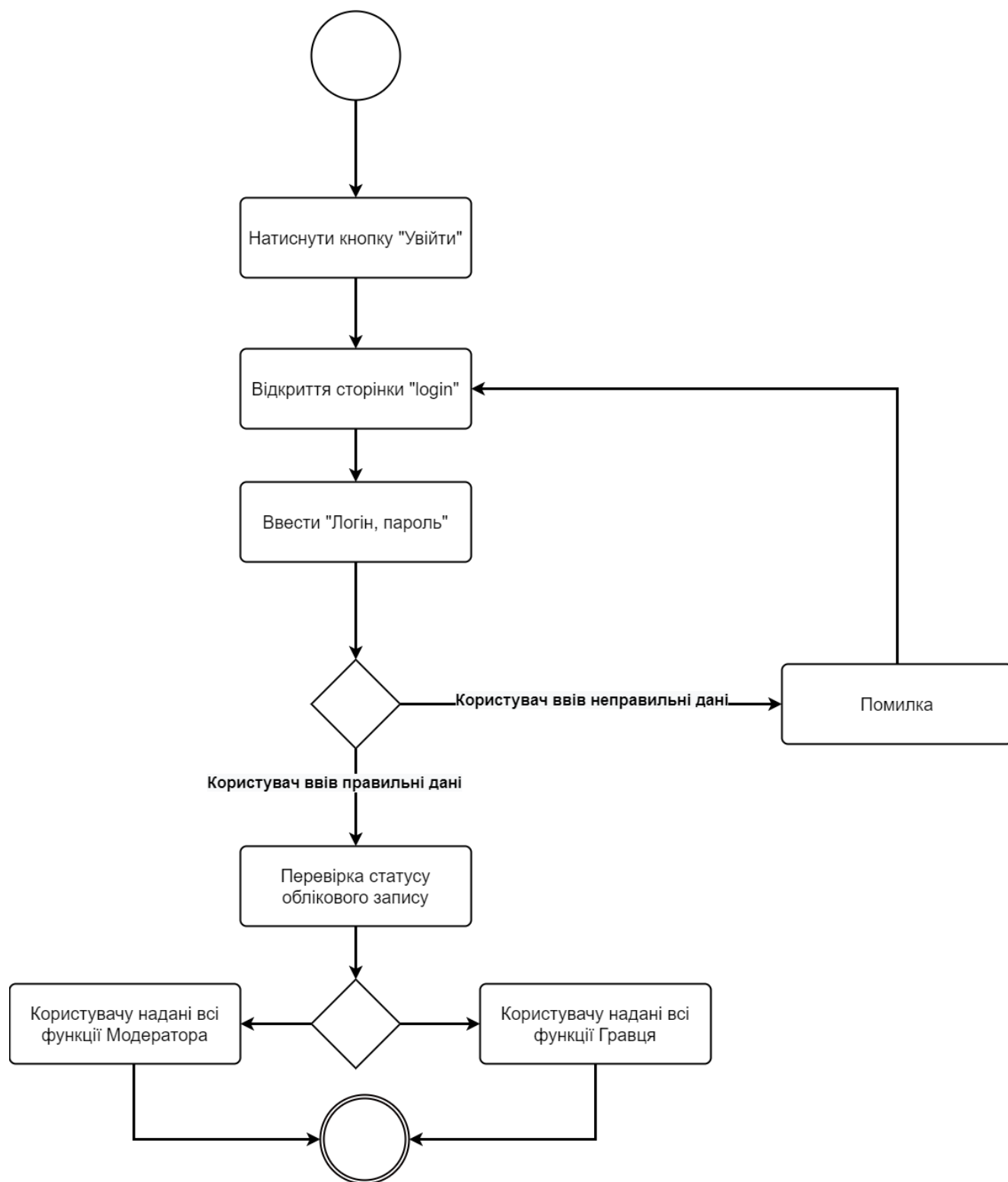


Рисунок 2.2 - Діаграма діяльності для варіанту використання «Авторизація»

На рисунку 2.3 наведено діаграму діяльності для варіанту використання «Створити резюме»:

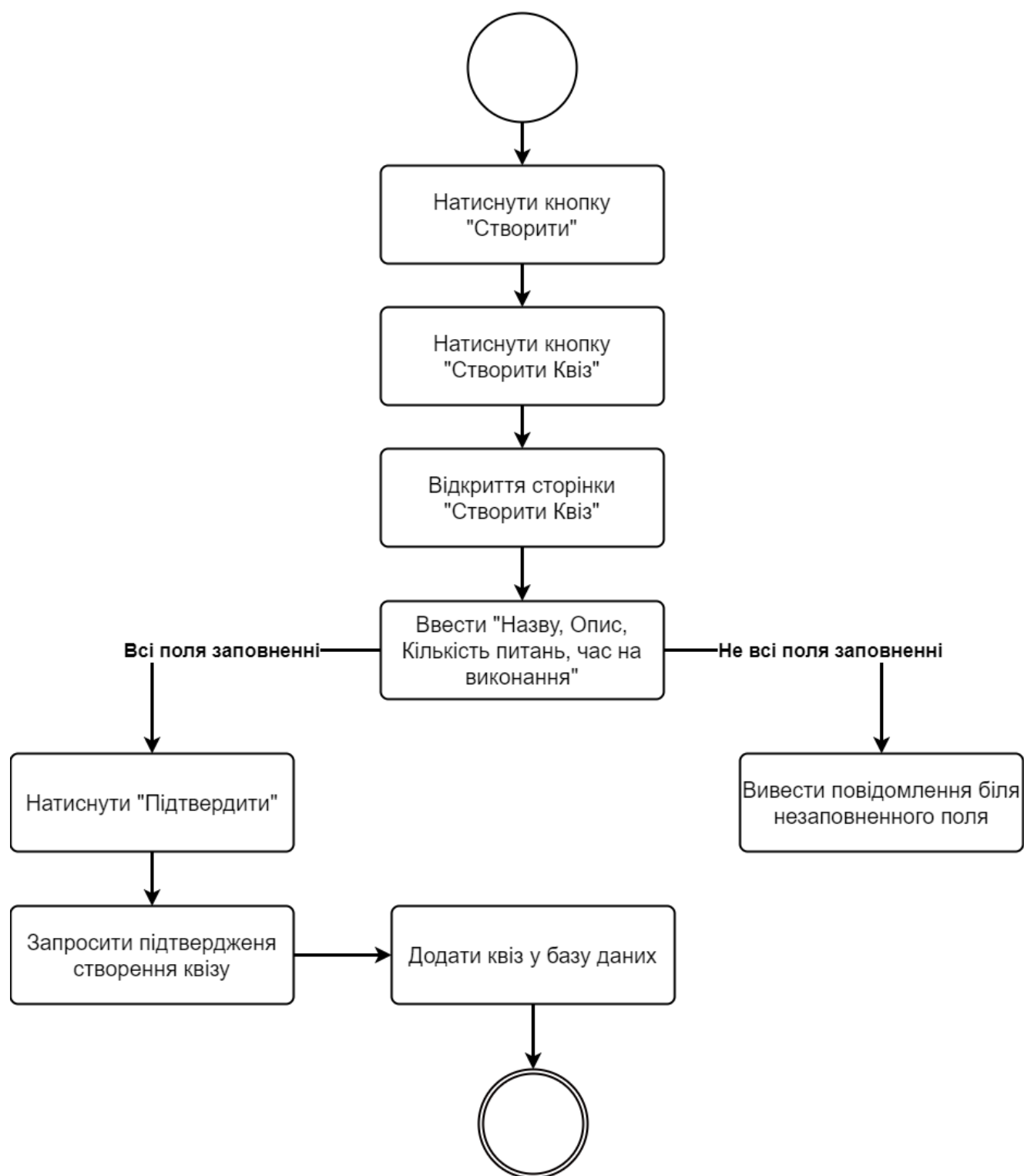


Рисунок 2.3 – Діаграма діяльності для варіанту використання «Створення квізу»

На рисунку 2.4 наведено діаграму діяльності для варіанту використання «Видалити квіз» для модератора:

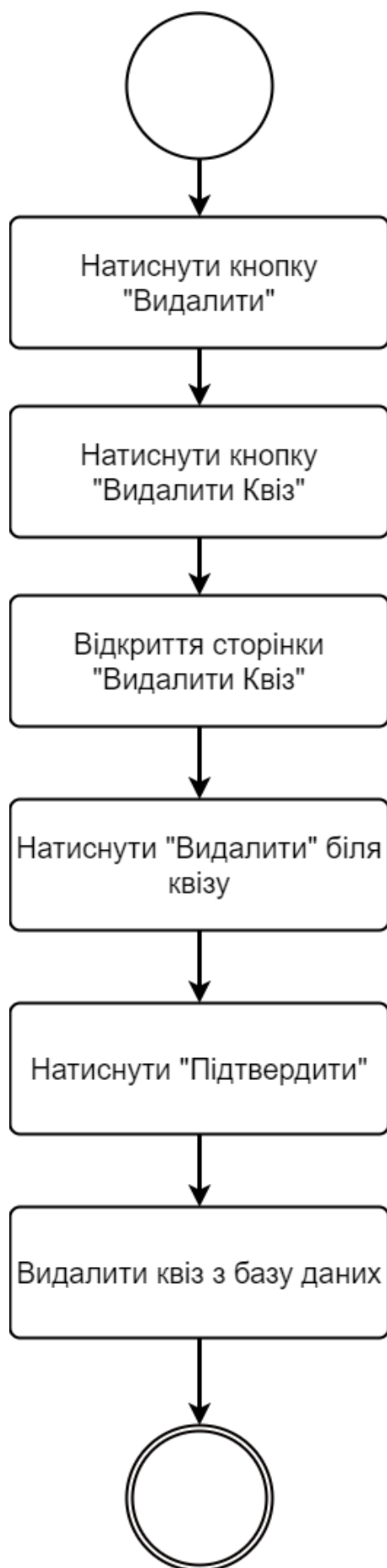


Рисунок 2.4 – Діаграма діяльності для варіанту використання «Видалити квіз»

Створені діаграми діяльності дозволяють деталізувати особливості алгоритмічної та логічної реалізації.

2.1.4 Інформаційна модель

Інформаційна модель програмного забезпечення представлена діаграмою класів. Діаграма UML, приведена нижче, показує моделі, які були визначені в цьому випадку. Діаграма класів визначає типи класів системи та різного роду статичні зв'язки, які існують між ними. На діаграмах класів зображуються також атрибути класів, операції класів та обмеження, що накладаються на зв'язки між класами. На Рисунку 2.5 наведено діаграму класів

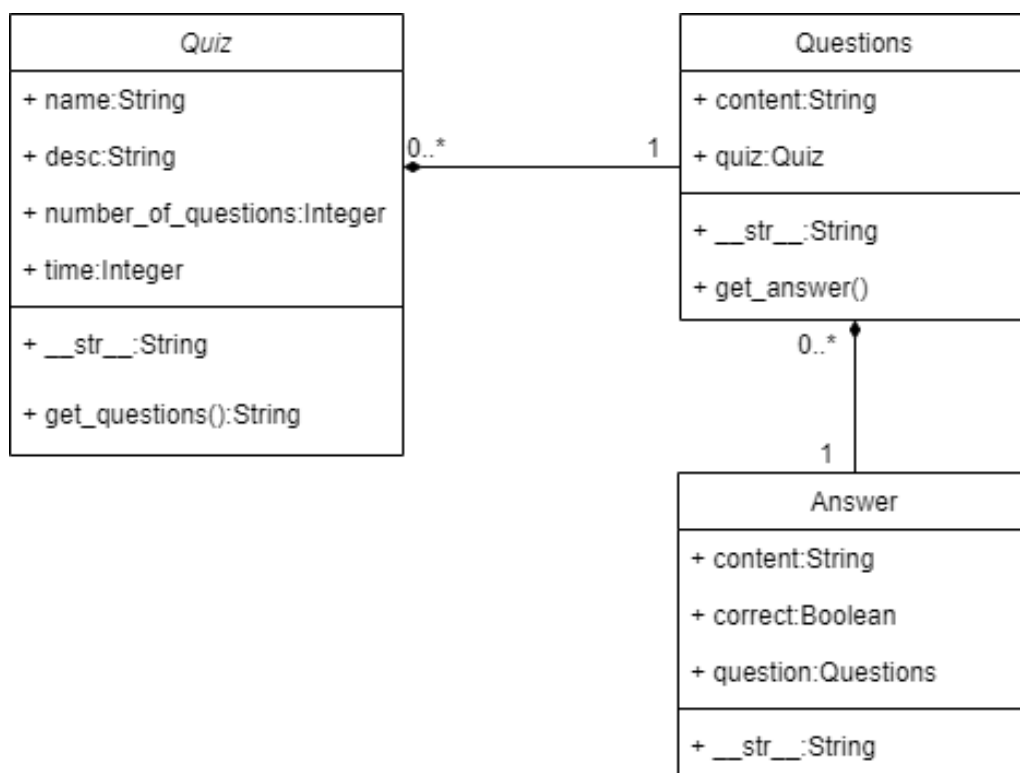


Рисунок 2.5 – Діаграма класів

2.1.5 Логічна модель бази даних

Схема "сутність-зв'язок" (також ERD або ER-діаграма) - це різновид блок-схеми, де показано, як різні "сутності" (люди, об'єкти, концепції й так далі)

пов'язані між собою всередині системи. ER-діаграми найчастіше застосовуються для проєктування та налагодження реляційних баз даних у сфері освіти, дослідження та розробки програмного забезпечення та інформаційних систем для бізнесу. Логічна модель бази даних представлена на Рисунку 2.6

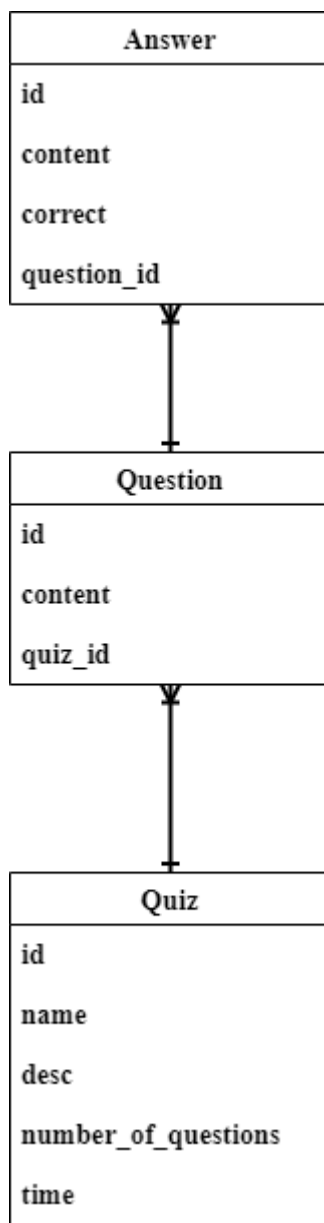


Рисунок 2.6 – Логічна модель бази даних

ER-діаграми (або ER-моделі) покладаються на стандартний набір символів, включаючи прямокутники, ромби, овали та сполучні лінії для відображення сутностей, їх атрибутів та зв'язків.

2.1.6 Фізична модель бази даних

Фізична модель бази даних визначає розміщення даних у зовнішній пам'яті. Вона ще називається внутрішньою моделлю системи й форма її представлення залежить від обраної СУБД. Для системи вебсайту "Quiz" це СУБД PostgreSQL. Фізична модель бази даних представлена на рисунку 2.7.

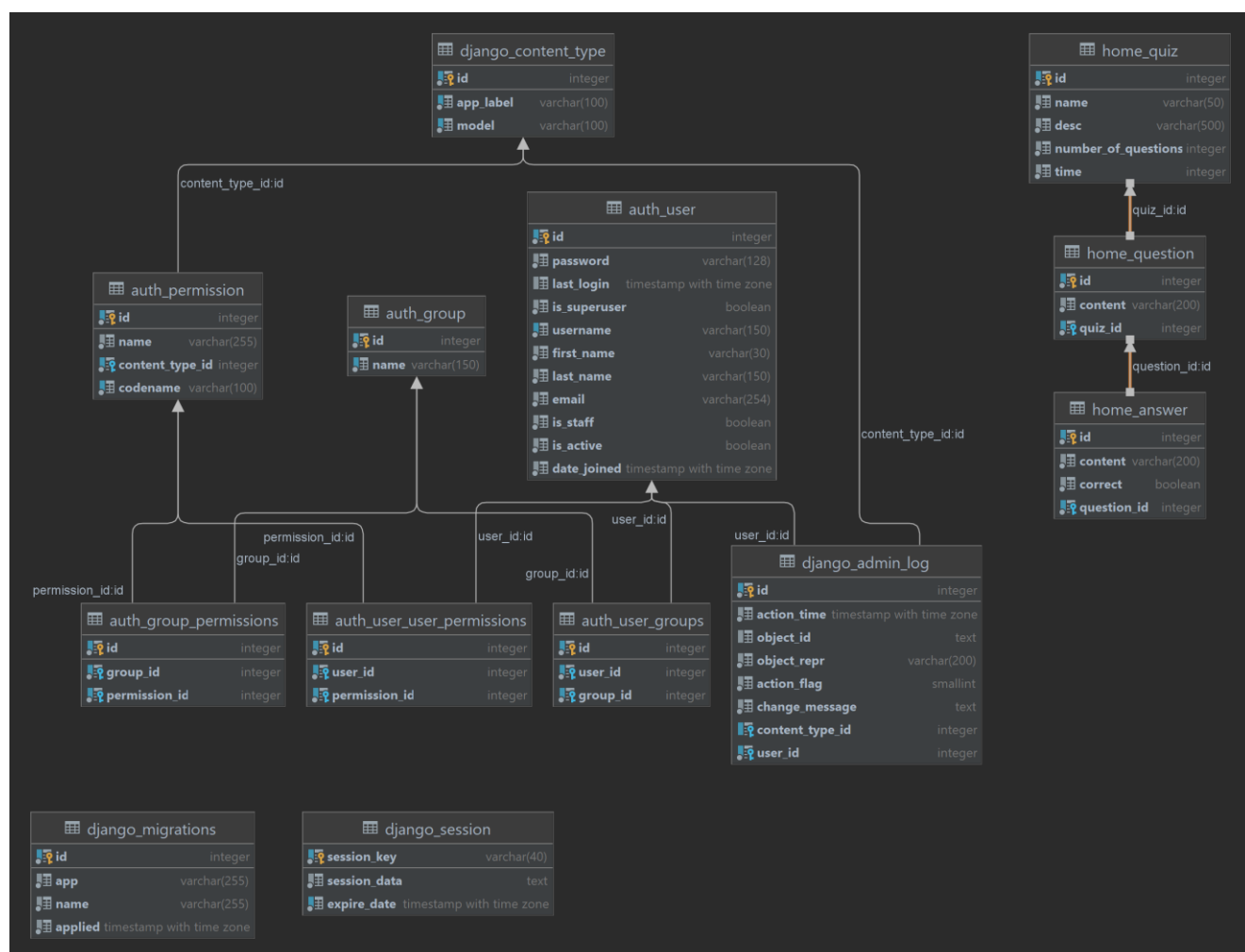


Рисунок 2.7 – Фізична модель бази даних

Фізична модель бази даних створена за допомогою додатку DataGrip.

2.1.7 Розробка інтерфейсу програмного забезпечення

У цьому розділі представлені схеми взаємодії користувача з вебсайтом.

Під час відкриття сторінки перед нами з'явиться домашня сторінка з навігаційною панеллю та кнопками "Увійти" та "Реєстрація", а також список квізів, які є на сайті. Домашню сторінку зображено на малюнку 2.8.

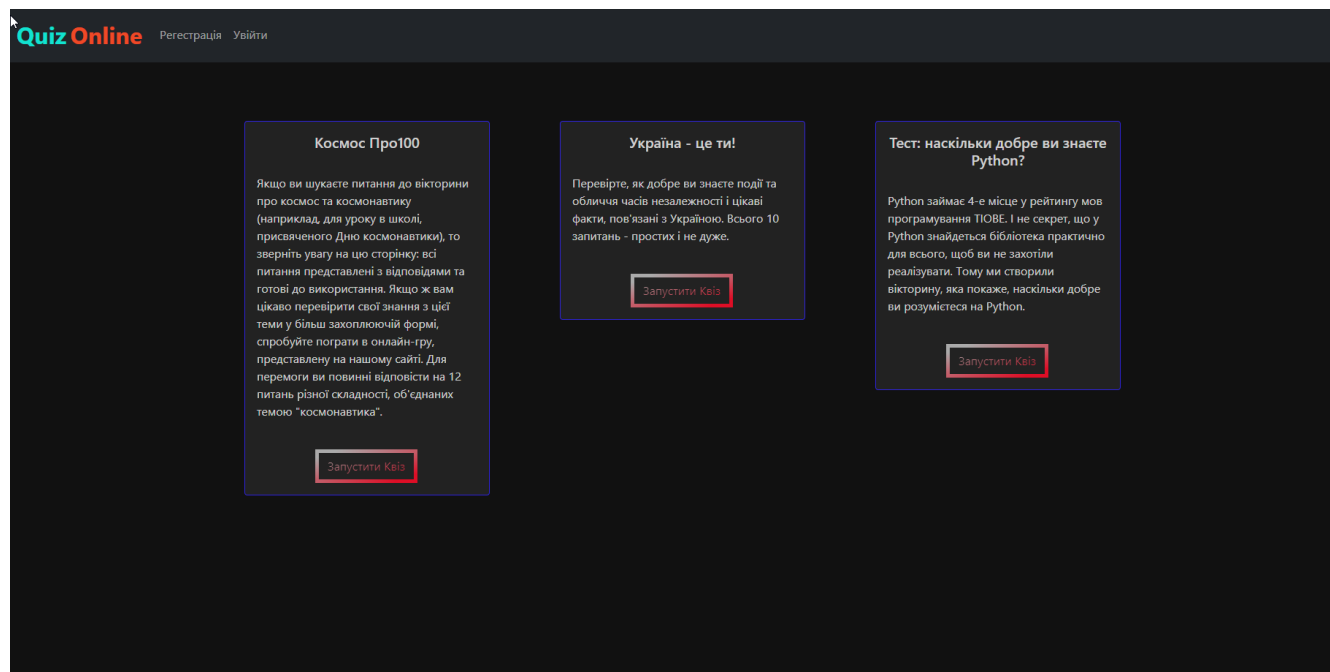


Рисунок 2.8 – Домашня сторінка

При натисканні кнопки реєстрація, ми потрапляємо на сторінку реєстрації, де потрібно заповнити відведені для реєстрації поля (ім'я, прізвище, пошта, логін, пароль), Сторінка реєстрації представлена на Рисунку 2.9. Після натискання на кнопку "Підтвердити" ми одразу потрапляємо на сторінку авторизації, де потрібно ввести логін та пароль для входу, приклад сторінки авторизації представлений на Рисунку 2.10.

Quiz Online [Регистрація](#) [Увійти](#)

ДЕТАЛІ РЕГІСТРАЦІЇ

Ім'я

Введіть Ім'я

Прізвище

Введіть Прізвище

Логін

Введіть Логін

Пошта

Введіть вашу пошту

Пароль

Введіть Пароль

Повторіть Пароль

Повторіть Пароль

Зареєструватись

Рисунок 2.9 – Сторінка Реєстрації

Quiz Online [Регистрація](#) [Увійти](#)

ДЕТАЛІ ВХОДУ

Логін

admin

Пароль

.....

Увійти

Рисунок 2.10 – Сторінка Авторизації

Після входу в систему як адміністратор у нас з'являється можливість видаляти квізи та запитання. Для цього потрібно перейти по меню, що розкривається, приклад якого наведено на Рисунку 2.11.

На Рисунку 2.12 та Рисунку 2.13 показано сторінки видалення квізу та питання. Також приклад підтвердження видалення квізу наведено на Рисунку 2.14.

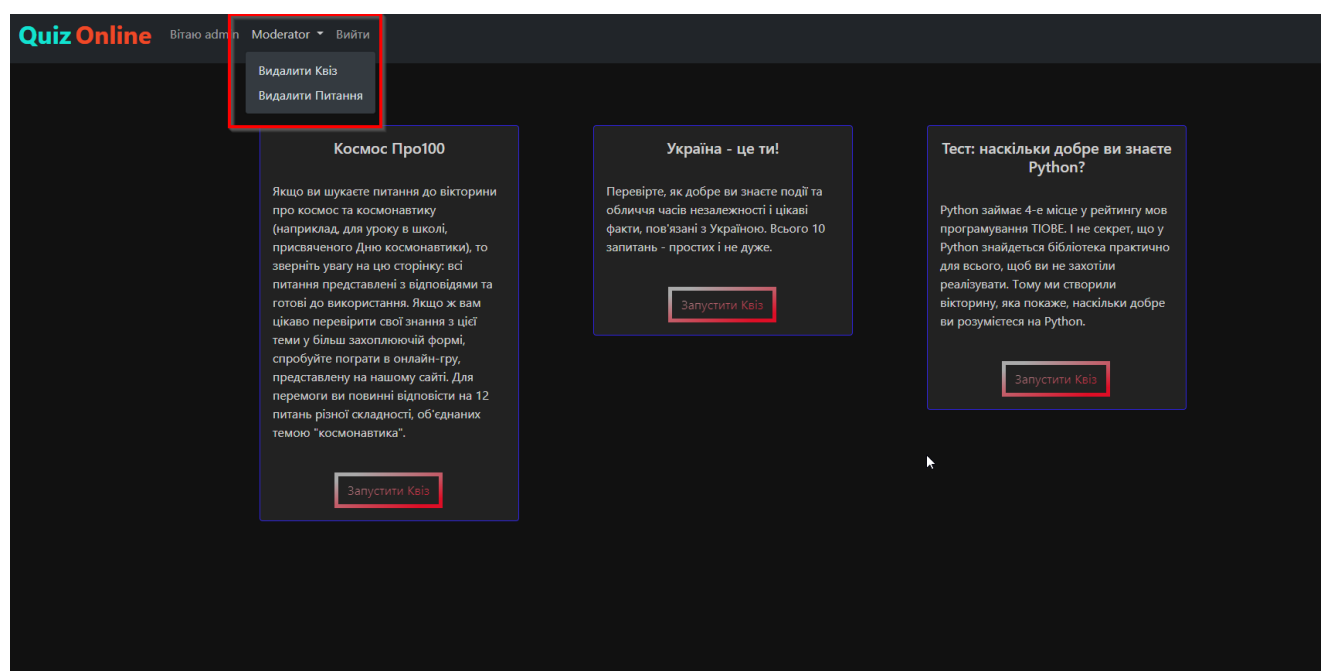


Рисунок 2.11 – Меню, що розкривається

Квіз	Опис	Видалити
Космос Про100	Якщо ви шукаєте питання до вікторини про космос та космонавтику (наприклад, для уроку в школі, присвяченого Дню космонавтики), то зверніть увагу на цю сторінку: всі питання представлені з відповідями та готові до використання. Якщо ж вам цікаво перевірити свої знання з цієї теми у більш захоплюючій формі, спробуйте пограти в онлайн-гру, представлену на нашому сайті. Для перемоги ви повинні відповісти на 12 питань різної складності, об'єднаних темою "космонавтика".	Видалити
Україна - це ти!	Перевірте, як добре ви знаєте події та обличчя часів незалежності і цікаві факти, пов'язані з Україною. Всього 10 запитань - простих і не дуже.	Видалити
Тест: наскільки добре ви знаєте Python?	Python займає 4-е місце у рейтингу мов програмування ТЮБЕ. І не секрет, що у Python знайдеться бібліотека практично для всього, щоб ви не захотіли реалізувати. Тому ми створили вікторину, яка покаже, наскільки добре ви розумієтеся на Python.	Видалити

Рисунок 2.12 – Сторінка видалення квіза

Питання	Квіз	Видалити
тут	Тест: наскільки добре ви знаєте Python?	Видалити
Як називається найближча до Сонця планета?	Космос Про100	Видалити
Який астронавт був другим (слідом за Леоновим) вийшов у відкритий космос?	Космос Про100	Видалити
Як звали людину, яка першою висадилася на Місяць?	Космос Про100	Видалити
Хто став першим "космічним туристом"?	Космос Про100	Видалити
Що є причиною утворення кратерів на Місяці?	Космос Про100	Видалити
На якій максимальній висоті був корабель під час польоту Гагаріна?	Космос Про100	Видалити
Які планети сонячної системи обертаються у напрямку, протилежному до Землі?	Космос Про100	Видалити
Що в перекладі з грецької означає "комета"?	Космос Про100	Видалити
Який вчений винахідник космічної ракети?	Космос Про100	Видалити
Як називався корабель, на якому 12 квітня 1961 року Юрій Гагарін здійснив перший політ у космос?	Космос Про100	Видалити

Рисунок 2.13 – Сторінка видалення питання

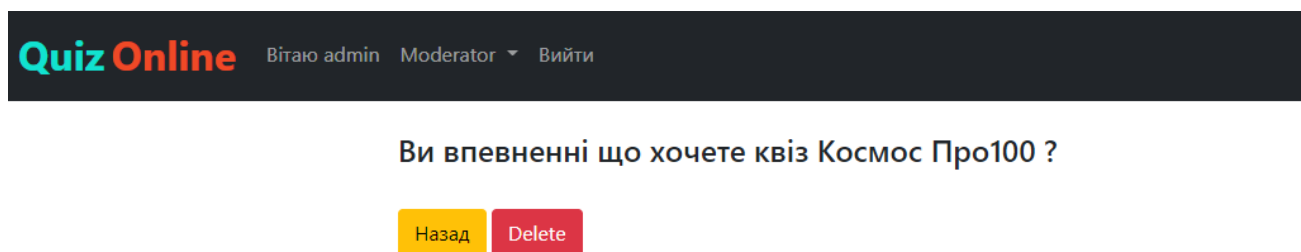


Рисунок 2.14 – Сторінка підтвердження видалення

Після входу під обліковий запис зазвичай користувач (гравця), ми можемо створити квіз і додати йому запитання. Приклад сторінки гравця, наведено на Рисунку 2.15.

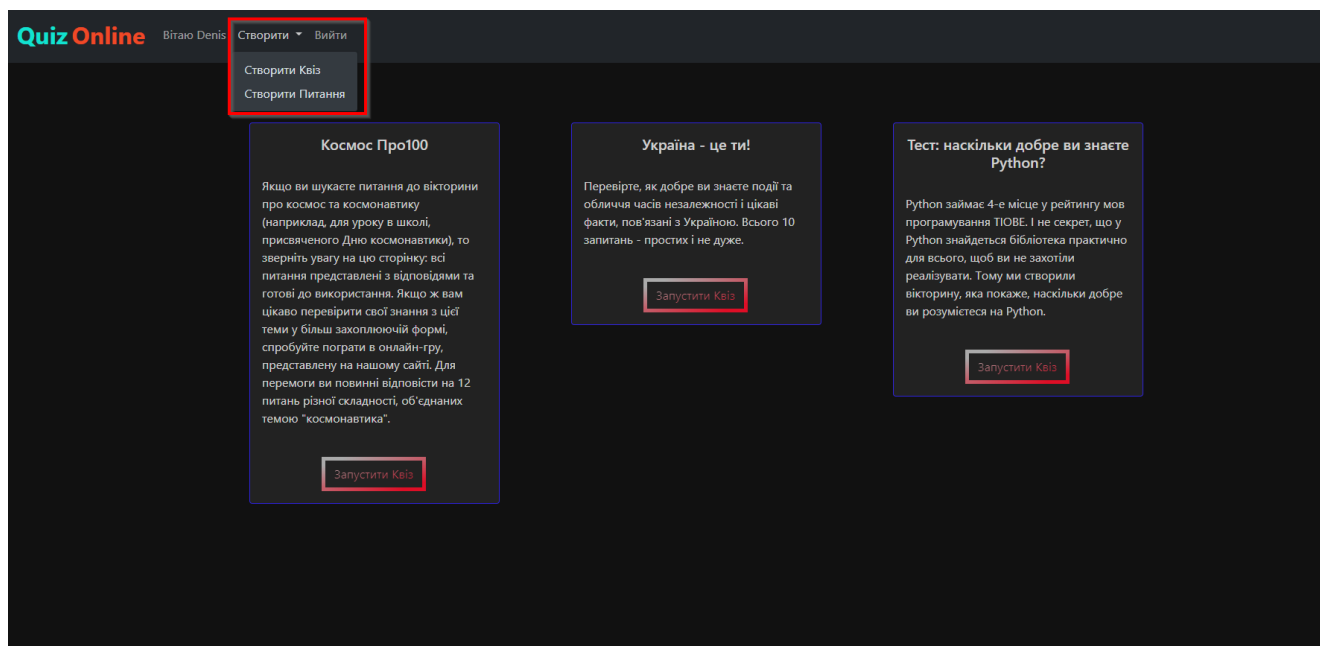


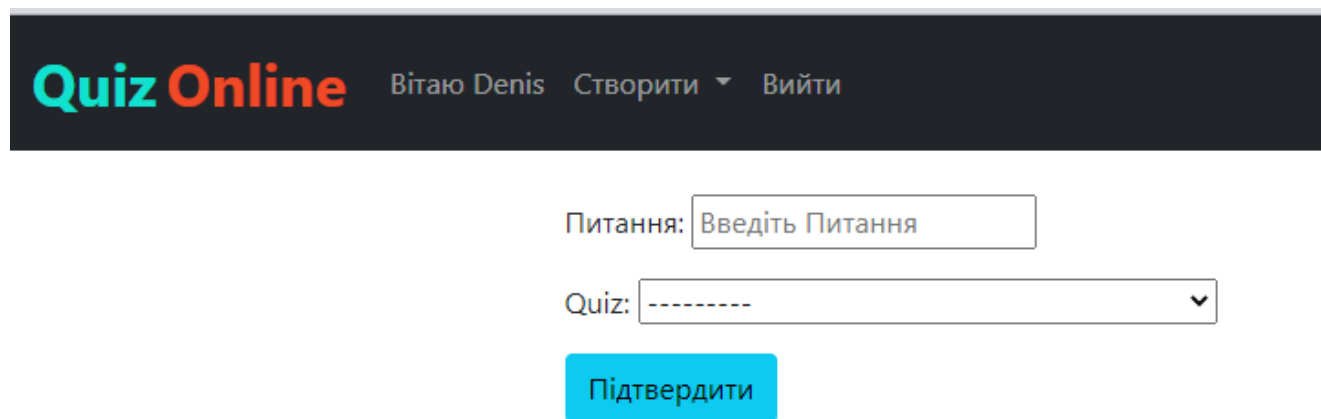
Рисунок 2.15 – Головна сторінка гравця

Для створення квіза потрібно ввести назву квіза, опис, кількість питань та тривалість проходження. Приклад створення квізу на Рисунку 2.16.

Квіз	Опис	Видалити
Космос Про100	Якщо ви шукаєте питання до вікторини про космос та космонавтику (наприклад, для уроку в школі, присвяченого Дню космонавтики), то зверніть увагу на цю сторінку: всі питання представлені з відповідями та готові до використання. Якщо ж вам цікаво перевірити свої знання з цієї теми у більш захоплюючій формі, спробуйте пограти в онлайн-гру, представлену на нашому сайті. Для перемоги ви повинні відповісти на 12 питань різної складності, об'єднаних темою "космонавтика".	Видалити

Рисунок 2.16 – Сторінка створення квіза

На сторінці створення питання потрібно вказати питання та до якого квізу воно належить, приклад цієї сторінки на Рисунку 2.17. Після створення питання відкривається сторінка додавання варіантів відповіді, де потрібно вказати варіант відповіді та поставити галочку в чек боксі з правильним варіантом, інтерфейс сторінки додавання варіантів відповіді на Рисунку 2.18.

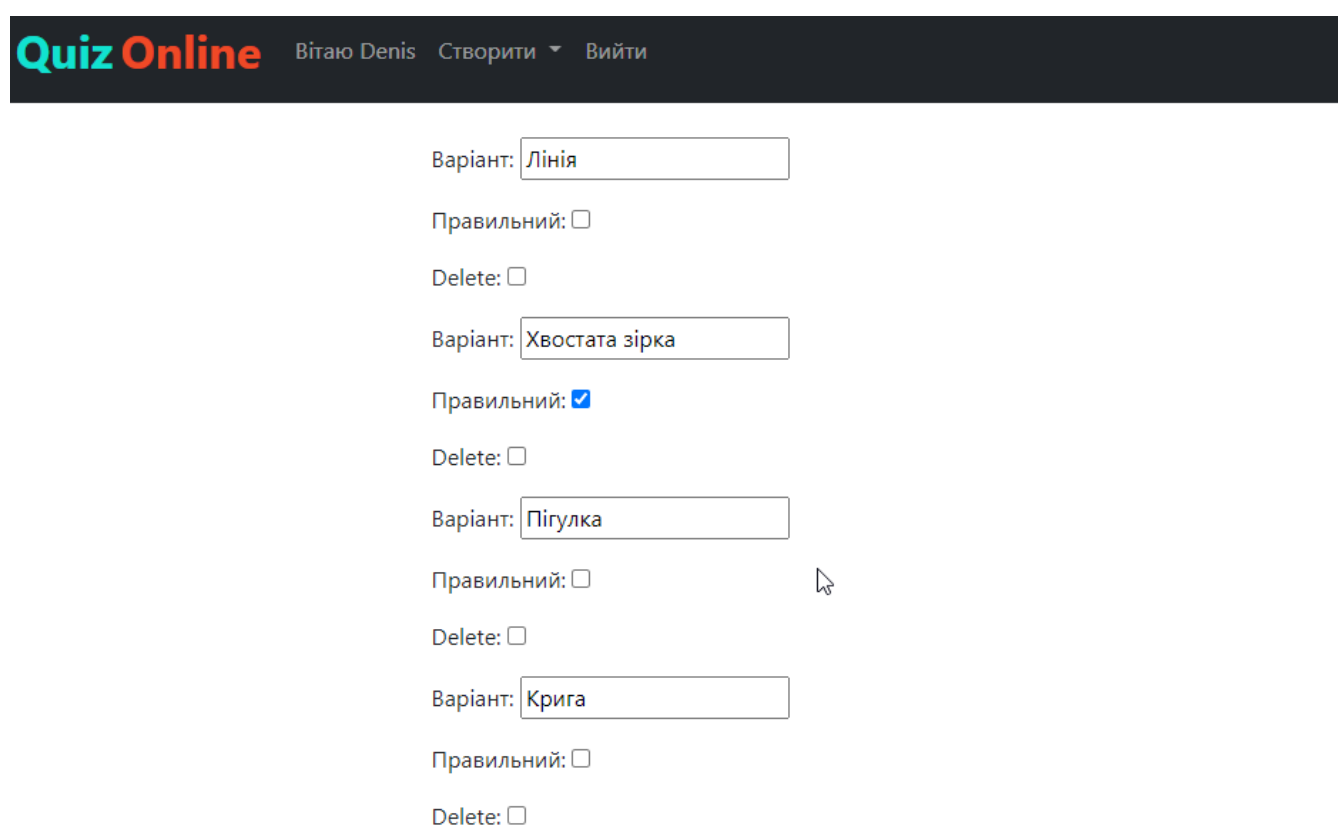


Quiz Online Вітаю Denis Створити Вийти

Питання:

Quiz:

Рисунок 2.17 – Сторінка створення питання



Quiz Online Вітаю Denis Створити Вийти

Варіант:

Правильний: ☐

Delete: ☐

Варіант:

Правильний: ☒

Delete: ☐

Варіант:

Правильний: ☐

Delete: ☐

Варіант:

Правильний: ☐

Delete: ☐

Рисунок 2.18 – Сторінки додавання варіантів відповіді

Гравець може почати проходити квіз після натискання на головній сторінці на кнопку "Запустити Квіз" з'являється поп-ап з правилами квізу, після чого гравець підтверджує початок квізу натисканням кнопки "Почати Квіз!". Приклад правил квізу на Рисунку 2.19.

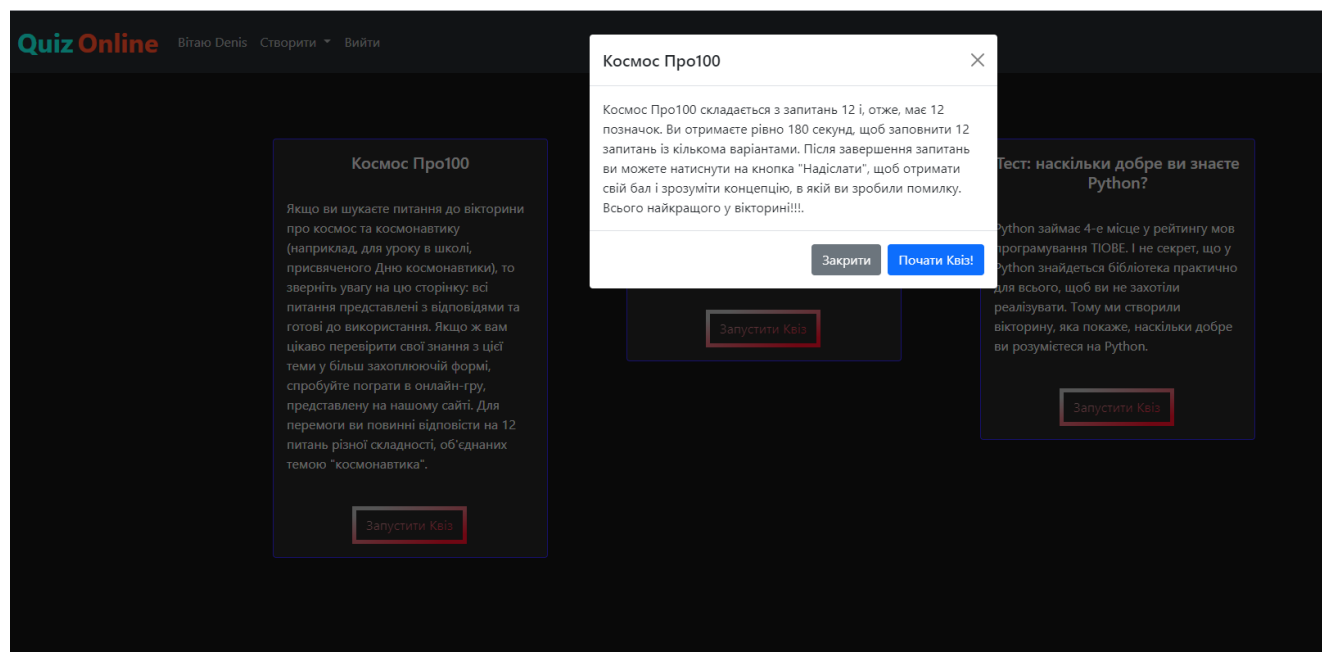


Рисунок 2.19 – Поп-ап правил квізу

На сторінці квіза представлені питання та варіанти відповіді, де потрібно виділити правильні варіанти за відведений час, після чого натиснути кнопку "Надіслати". Приклад сторінки квізу на Рисунку 2.20.

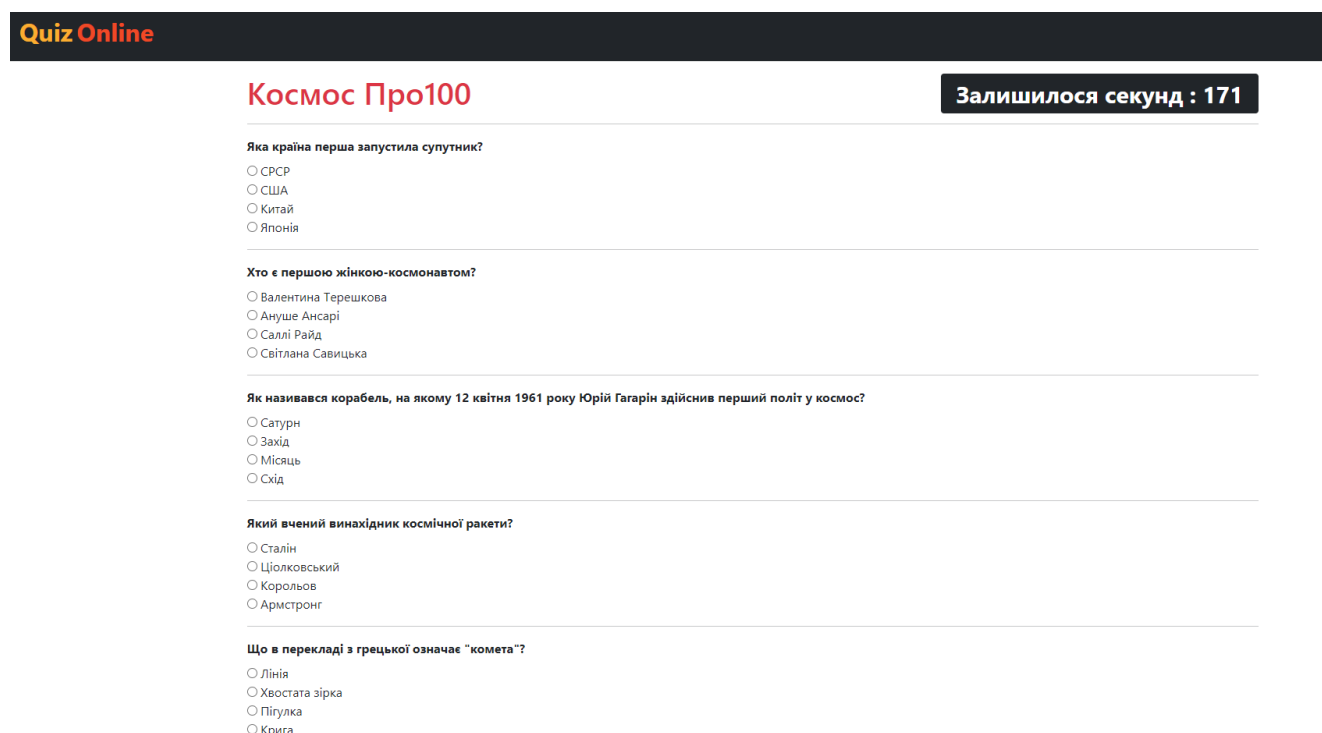


Рисунок 2.20 – Сторінка квізу

Після здачі квізу з'являється результат із кількістю правильних відповідей із кількості питань, детальніше продемонстровано на Рисунку 2.21.

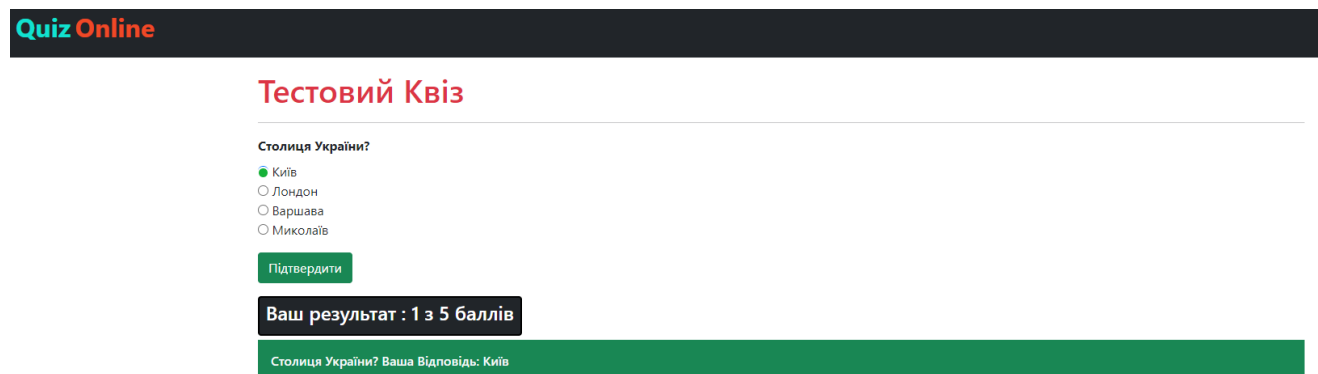


Рисунок 2.21 – Результат проходження квізу

Результат складається з правильних відповідей відносно заданої кількості питань у квізі.

2.2 Технічний проєкт програмного забезпечення

Проаналізувавши ескізний проєкт, був розроблений технічний проєкт програмного забезпечення шляхом побудови статичної й динамічної моделей. Статична модель представлена діаграмою класів, а динамічна – діаграмою взаємодії.

2.2.1 Структура класів

На основі моделі варіантів використання, інформаційної моделі та моделі програмного інтерфейсу була розроблена статична модель, представлена діаграмою класів, яка наведена на рисунку 2.22:

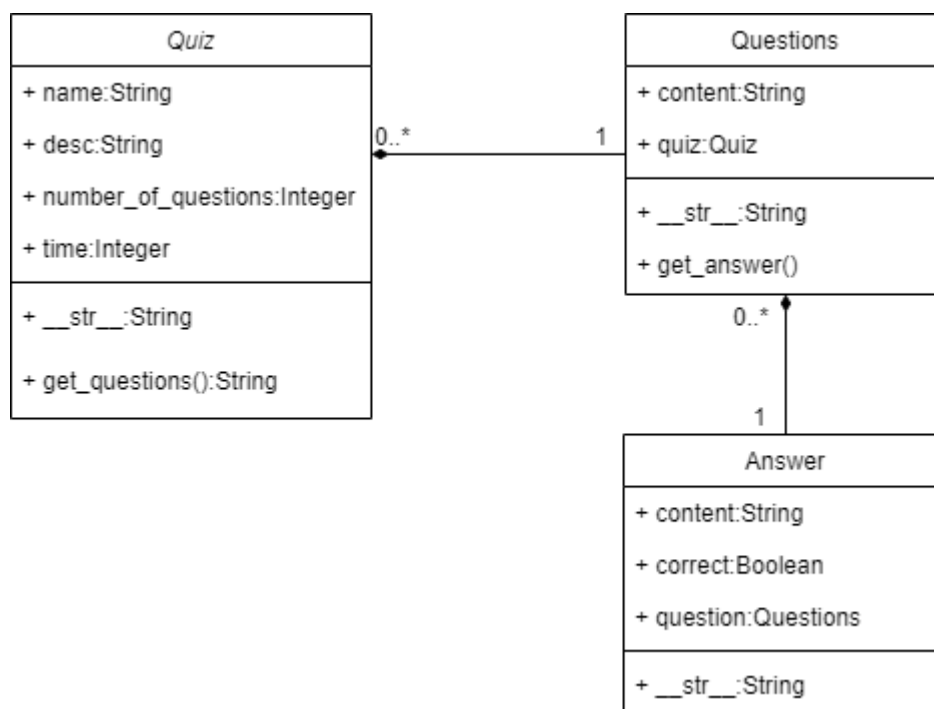


Рисунок 2.22 – Статична модель, представлена діаграмою класів

Специфікації класів Клас: «Quiz»

Відповідальність: визначає структуру даних, що зберігаються, включаючи типи полів і, можливо, їх максимальний розмір, значення за замовчуванням, параметри списку вибору. Основний клас квізу.

Атрибути: name, desc, number_of_questions, time.

Нижче представлені функції класу

- __str__() – повертає строку. Відповідає за вивід інформації до інтерфейсу;
- get_questions() – відповідає за створення зв'язку з класом «Questions».

Клас: «Question»

Відповідальність: Клас відповідальний за опис питань.

Атрибути: content, quiz

Нижче представлені функції класу:

- __str__() – повертає строку. Відповідає за вивід інформації до інтерфейсу;
- get_answer() - відповідає за створення зв'язку з класом «Answer».

Клас: «Answer»

Відповідальність: Клас, що описує варіанти відповіді.

Атрибути: content, correct, question.

Нижче представлені функції класу

- `__str__()` – повертає строку. Відповідає за вивід інформації до інтерфейсу;

2.3 Робочий проєкт програмного забезпечення

2.3.1 Фізична модель даних

Діаграма розгортання показує топологію системи та розподіл компонентів системи її вузлами, і навіть сполуки - маршрути передачі між апаратними вузлами. На рисунку 2.23 наведена діаграма розгортання:

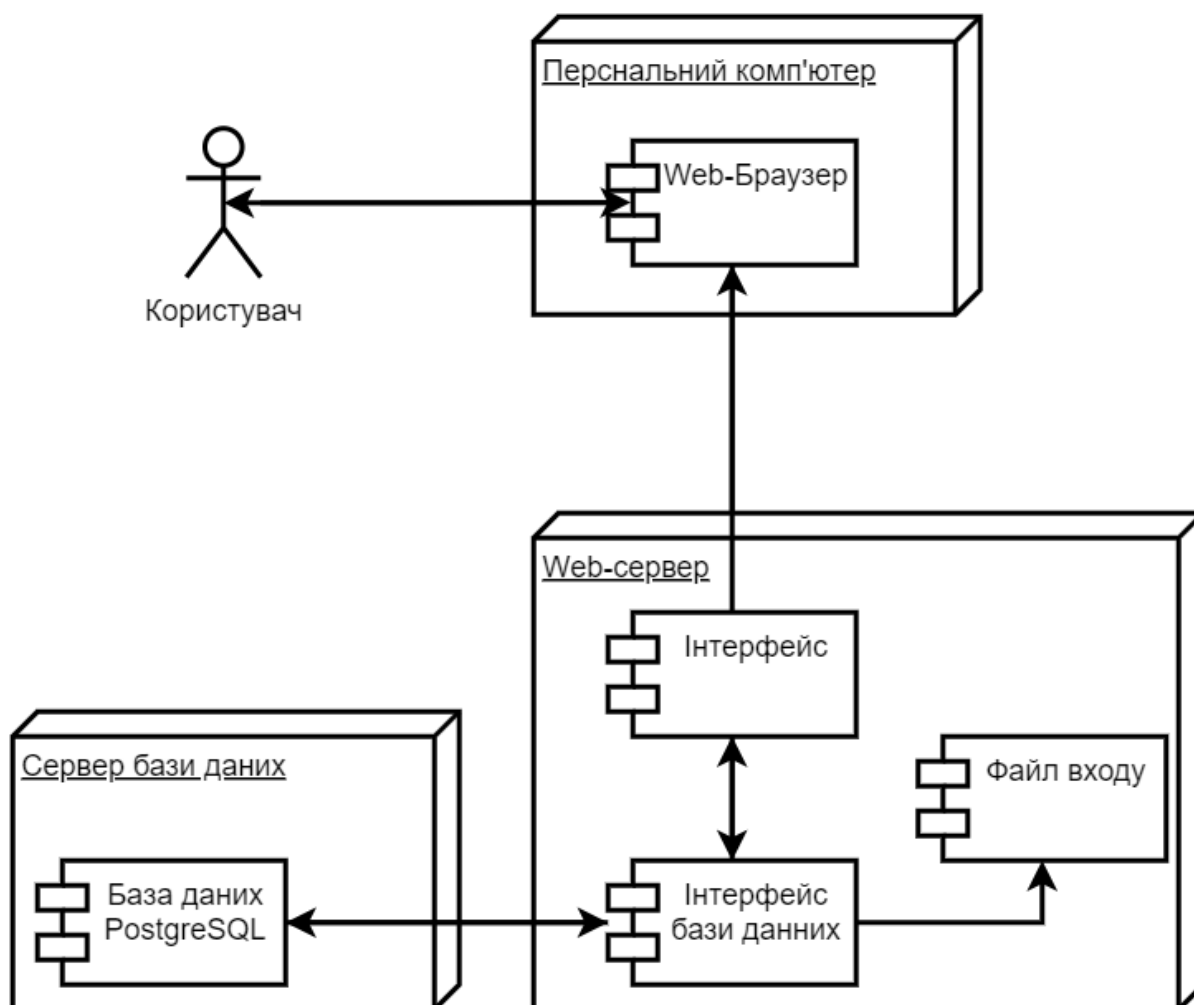


Рисунок 2.23 - Діаграма розгортання.

Діаграми розгортання допомагають моделювати апаратну топологію системи проти іншими типами UML-діаграм, які переважно описують логічні компоненти системи.

2.3.2 Тестування варіантів використання

Для тестування програмного забезпечення були обрані лише основні функції, а саме такі варіанти використання, як: «Створити квіз», «Авторизація».

Тестування чорного ящика або поведінкове тестування - стратегія (метод) тестування функціональної поведінки об'єкта (програми, системи) з точки зору зовнішнього світу, при якому не використовується знання про внутрішній устрій (код) тестованого об'єкта. Інакше кажучи, тестуванням чорного ящика займаються тестувальники, що не мають доступ до вихідного коду програми. Під стратегією розуміються систематичні методи відбору і створення тестів для тестового набору. Стратегія поведінкового тесту виходить з технічних вимог і їх специфікацій [21].

Тестування варіанта використання «Авторизація»

На даному вебсайті реалізована функція авторизації як «гравець», «модератор».

Тест 1.

Тестування авторизації користувача як гравця.

Для авторизації користувача як гравець, він повинен натиснути на кнопку «увійти» та ввести логін та пароль які він вказував при реєстрації. Сайт повинен надати йому доступ як авторизованого гравця. Після чого він отримає можливість створювати квізи, питання та проходили квізи. Результат тестування представлений на рисунку 2.24:

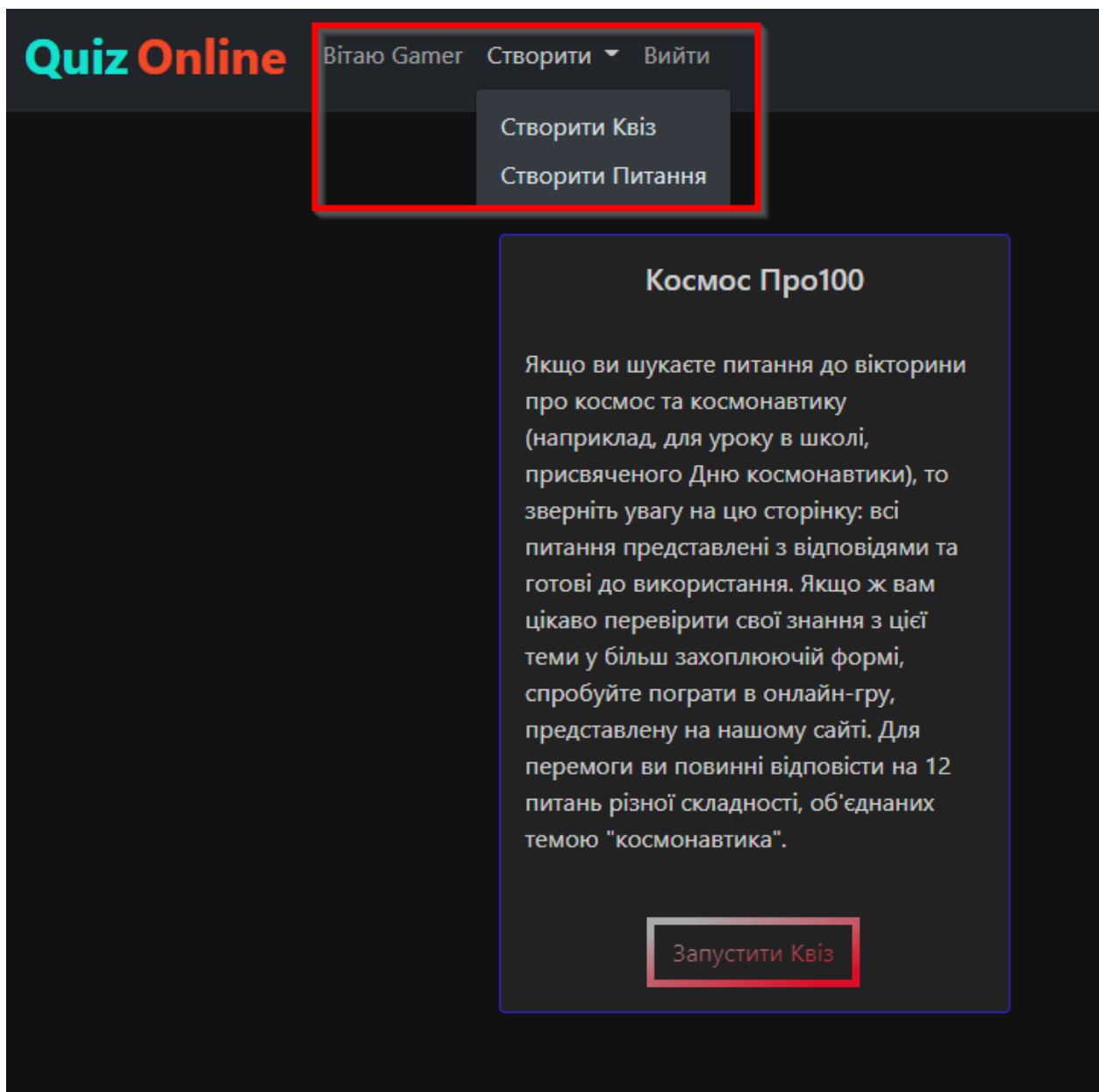


Рисунок 2.24 – Результат тестування варіанту використання «Авторизація»
(тест 1)

В результаті тестування можна зробити висновок, що функція «Авторизація» працює нормально.

Тест 2.

Тестування авторизації користувача як модератора.

Для авторизації користувача як роботодавця він повинен отримати права від адміністратора сайту, при створенні профілю модератора він отримує права управління контентом, приклад створення такого користувача на Рисунку 2.25.

Вебсайт повинен авторизувати його як модератора та надати всі необхідні функції. Результат тестування на рисунку 2.26.

The screenshot shows the Django administration interface for the 'Moderator' user. The 'Change user' form is displayed, with the following details:

- Username:** Moderator
- Password:** algorithm: pbkdf2_sha256 iterations: 320000 salt: Krahkq***** hash: FZyOZ*****
- Personal info:**
 - First name: (empty)
 - Last name: (empty)
 - Email address: (empty)
- Permissions:**
 - ☒ Active
 - ☒ Staff status
 - ☐ Superuser status
- Groups:** Available groups (empty), Chosen groups (empty)

Рисунок 2.25 – Створення модератора

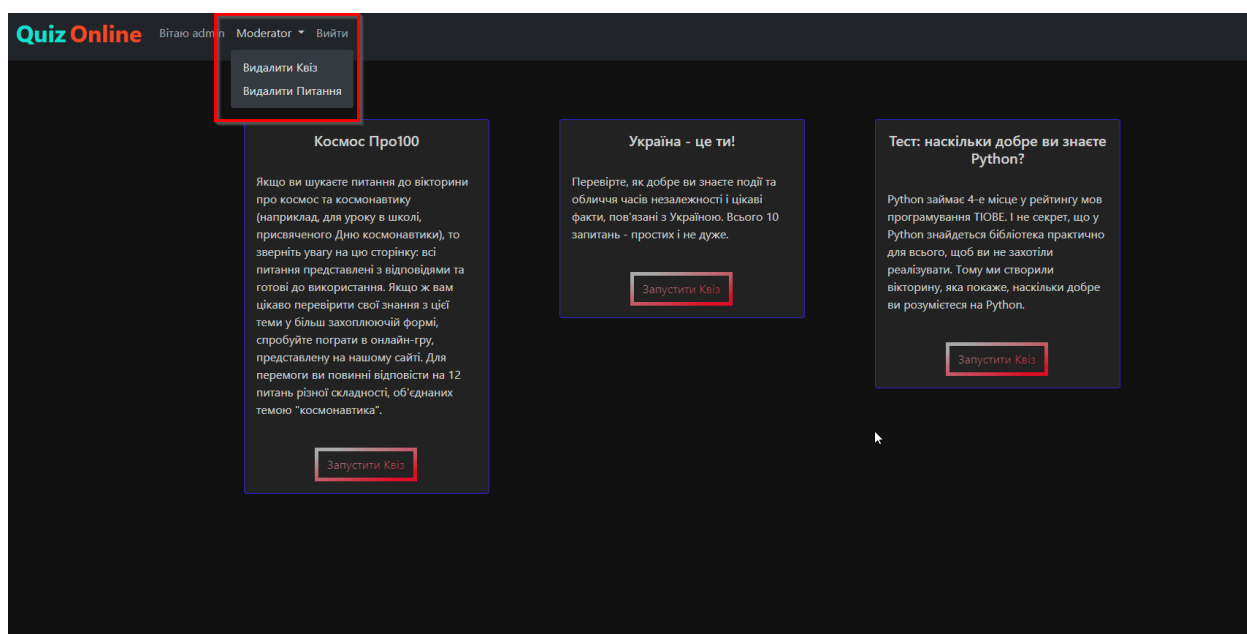


Рисунок 2.26 – Результат тестування варіанту використання «Авторизація»
(тест 2)

Тестування варіанта використання «Створення квізу»

На даному вебсайті реалізована функція створення квізу.

Тест

Тестування створення квізу під обліковим записом гравця.

Для створення квізу гравець, повинен натиснути на кнопку «Створити» та обрати «Створити Квіз». Після чого він перейде на сторінку створення квізу, яка наведена на Рисунку 2.16. Для створення квізу він повинен ввести назву, опис, вказати кількість питань та час відведений на проходження квізу. Коли квіз буде створений, навіть без питань він повинен з'явитися на головній сторінці. Під час тесту створимо квіз з вхідними даними:

- Назва: «5 фактів про Apple»;
- Опис: «Можна довго розповідати про заснування компанії, про її засновників, про перший комп'ютер, але далі мова піде про найцікавіші факти з життя цієї унікальної компанії. Подивимося чи знаєте ви їх.»;
- Кількість питань: 5;
- Час: 240;

Результат тестування представлений на рисунку 2.27:

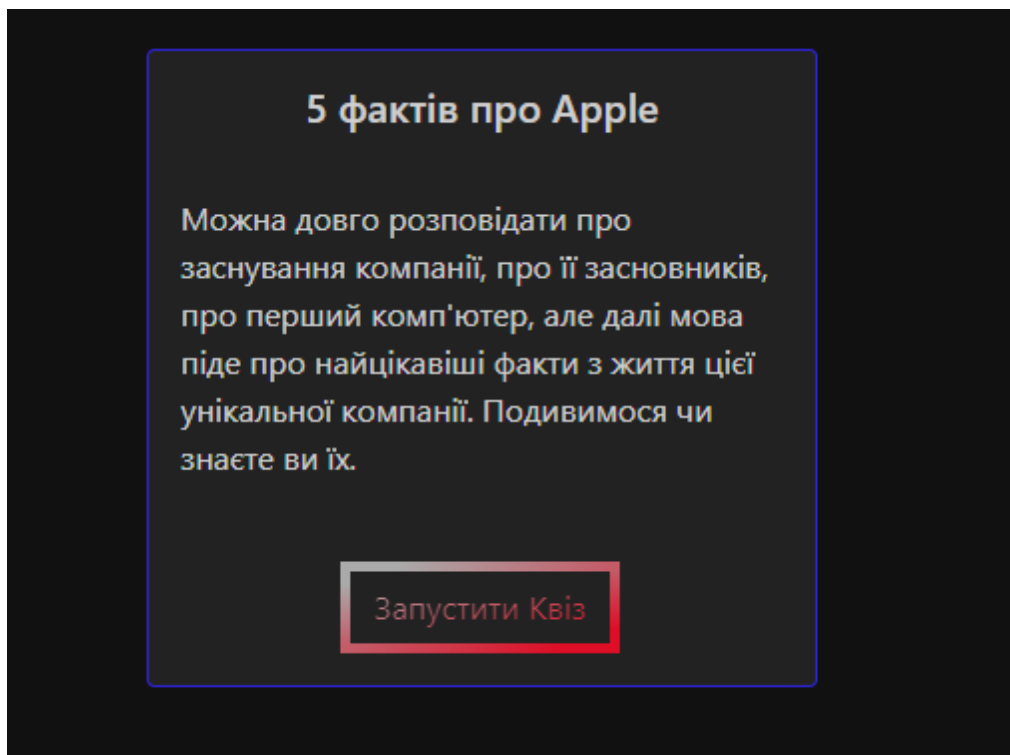


Рисунок 2.27 – Результат тестування варіанту використання «Створите квіз»

В результаті тестування можна зробити висновок, що функція «Створення квізу» працює нормально.

2.3.3 Випробування програми

Програму і методику випробувань розробленого програмного забезпечення наведено в Додатку В. На основі інструкцій, що наведені в цьому додатку, проведемо випробування розробленого програмного забезпечення. Наше завдання Зареєструватись, авторизуватися, створити квіз, створити питання, додати варіанти відповідей, пройти квіз.

При запуску головної сторінки вебсайту, ми бачимо доступні квізи та кнопки реєстрації та входу.

Натискаємо на кнопку «Реєстрація», потрапляємо на сторінку реєстрації, де потрібно ввести дані (ім'я, прізвище, пошта, логін, пароль, повторити пароль) після введення цих даних, натискаємо «Підтвердити» та переходимо на сторінку входу, вводимо логін та пароль. Процес реєстрації та авторизації показаний на Рисунках 2.28, 2.29, 2.30.

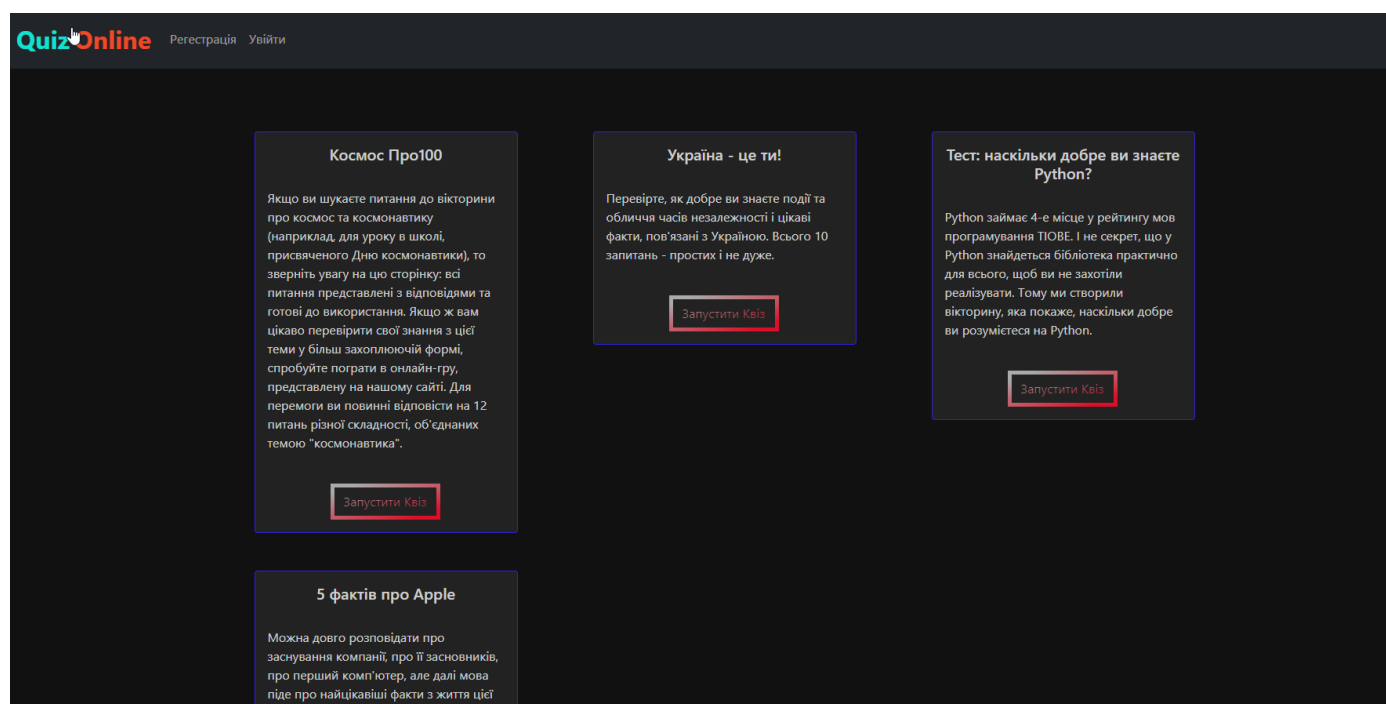


Рисунок 2.28 – Головна сторінка вебсайту

Quiz Online [Регистрація](#) [Увійти](#)

ДЕТАЛІ РЕГІСТРАЦІЇ

Ім'я

Tester

Прізвище

Tester

Логін

Tester

Пошта

Tester@gmail.com

Пароль

.....

Повторіть Пароль

.....

Зареєструватись

Рисунок 2.29 – Сторінка реєстрації

Quiz Online [Регистрація](#) [Увійти](#)

ДЕТАЛІ ВХОДУ

Логін

Tester

Пароль

.....

Увійти

Рисунок 2.30 – Сторінка авторизації

Після входу в систему як Гравець, нам потрібно натиснути на кнопку «Створити» і з меню вибрати «Створити Квіз». Після цього ми потрапляємо на сторінку зі створенням квізів, де вводимо дані нового квізу (назва, опис, кількість питань, час виконання), натискаємо кнопку «Підтвердити», підтверджуємо створення нового квіза. Процес створення нового квіза зображено на Рисунках 2.31, 2.32.

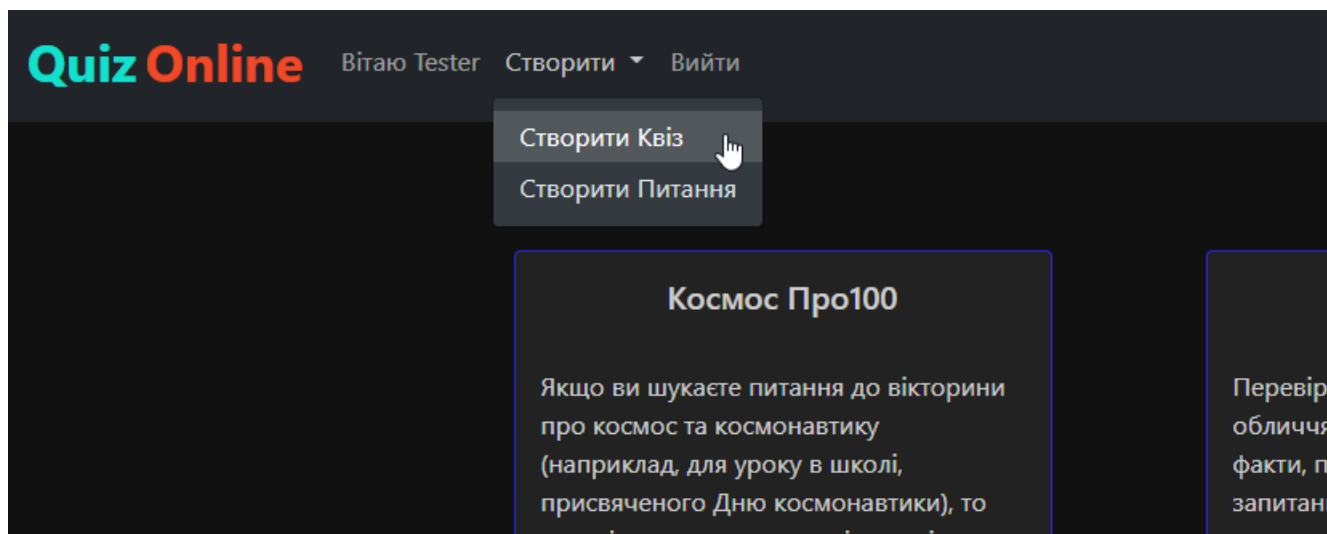


Рисунок 2.31 – Сторінка авторизованого користувача

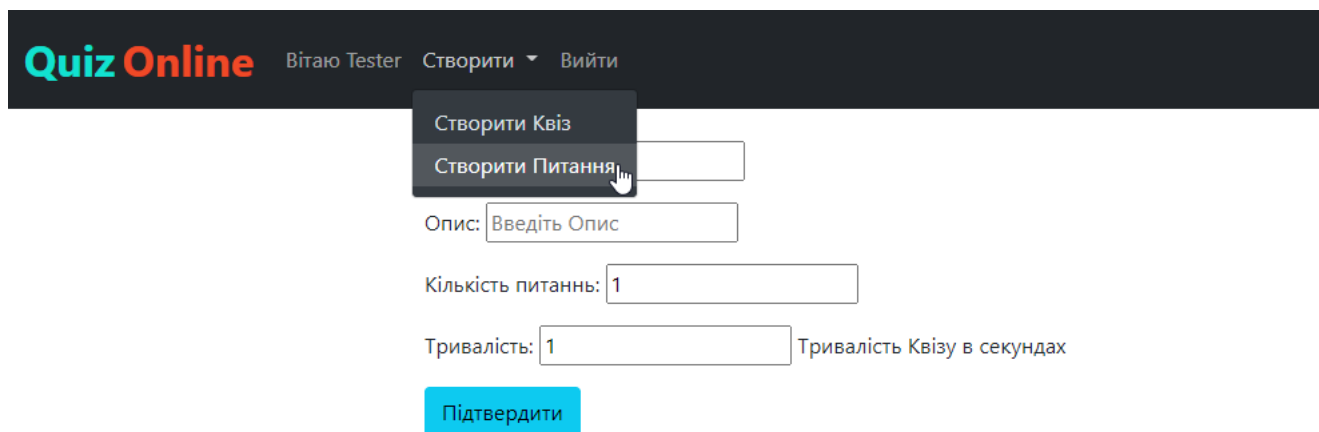
 The screenshot shows the 'Create Quiz' form in the 'Quiz Online' application. The top navigation bar is identical to the previous screenshot. Below the bar, there are four input fields: 'Назва:' with the value 'Тестовий Квіз', 'Опис:' with the value 'Lorem ipsum dolor sit am', 'Кількість питань:' with the value '5', and 'Тривалість:' with the value '240'. To the right of the 'Тривалість' field is the text 'Тривалість Квізу в секундах'. At the bottom of the form is a blue button labeled 'Підтвердити'.

Рисунок 2.32 – Процес створення квізу

Наступний крок, натискаємо «Створити», вибираємо «Створити питання», переходимо на сторінку створення питань. Вводимо питання та обираємо до якого

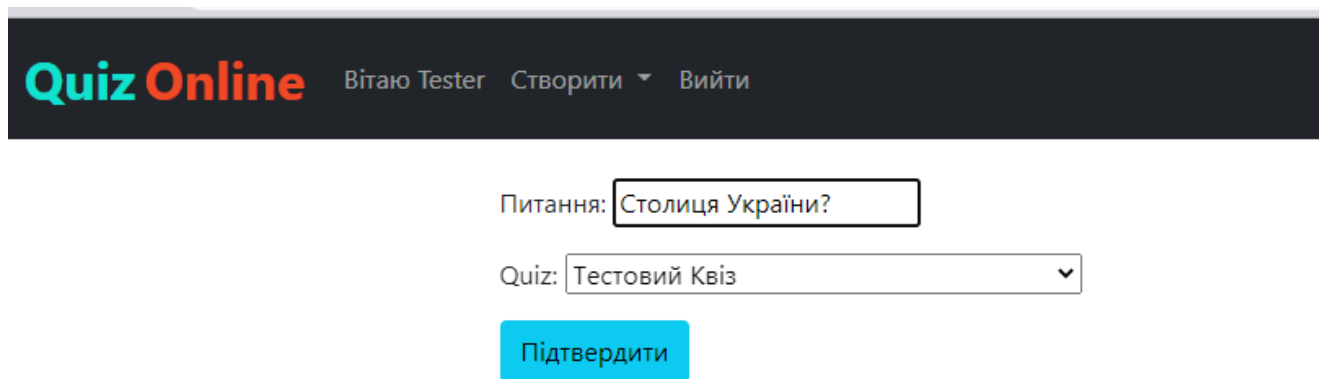
квізу він належить. Підтверджуємо. Вибираємо питання зі списку та відкриваємо його, додаємо 4 варіанти відповіді та вибираємо, який з них буде правильним, натискаємо «Підтвердити».

Процес створення питання та варіантів відповіді продемонстровано на Рисунках 2.33, 2.34, 2.35.



The screenshot shows the 'Quiz Online' header with navigation links: 'Вітаю Tester', 'Створити' (with a dropdown arrow), and 'Вийти'. The 'Створити' dropdown menu is open, showing two options: 'Створити Квіз' and 'Створити Питання', with the latter being selected. Below the menu, there is a form with the following fields: 'Опис:' with a placeholder 'Введіть Опис', 'Кількість питань:' with the value '1', and 'Тривалість:' with the value '1' and a label 'Тривалість Квізу в секундах'. A blue button labeled 'Підтвердити' is at the bottom.

Рисунок 2.33 – Перехід на сторінку «Створити Питання»



The screenshot shows the 'Quiz Online' header with navigation links: 'Вітаю Tester', 'Створити' (with a dropdown arrow), and 'Вийти'. The 'Створити' dropdown menu is open, showing two options: 'Створити Квіз' and 'Створити Питання', with the latter being selected. Below the menu, there is a form with the following fields: 'Питання:' with the text 'Столиця України?', 'Quiz:' with a dropdown menu showing 'Тестовий Квіз', and a blue button labeled 'Підтвердити' at the bottom.

Рисунок 2.34 – Процес створення питання

Quiz Online Вітаю Tester Створити Вийти

Варіант:

Правильний: ☒

Delete: ☐

Варіант:

Правильний: ☐

Delete: ☐

Варіант:

Правильний: ☐

Delete: ☐

Варіант:

Правильний: ☐

Delete: ☐

Рисунок 2.35 – Процес створення варіантів відповіді до питання

Переходимо на головну сторінку, натиснувши на назву сайту, вибираємо квіз який ми створили та натискаємо «Запустити Квіз». Читаємо правила квіза та натискаємо «Почати Квіз!». На сторінці з квізом вибираємо правильну відповідь та натискаємо «Підтвердити». Процес вибору та запуску квіза показаний на Рисунках 2.36, 2.37, 2.38. Отримуємо результат. Результат продемонстровано на Рисунку 2.39.

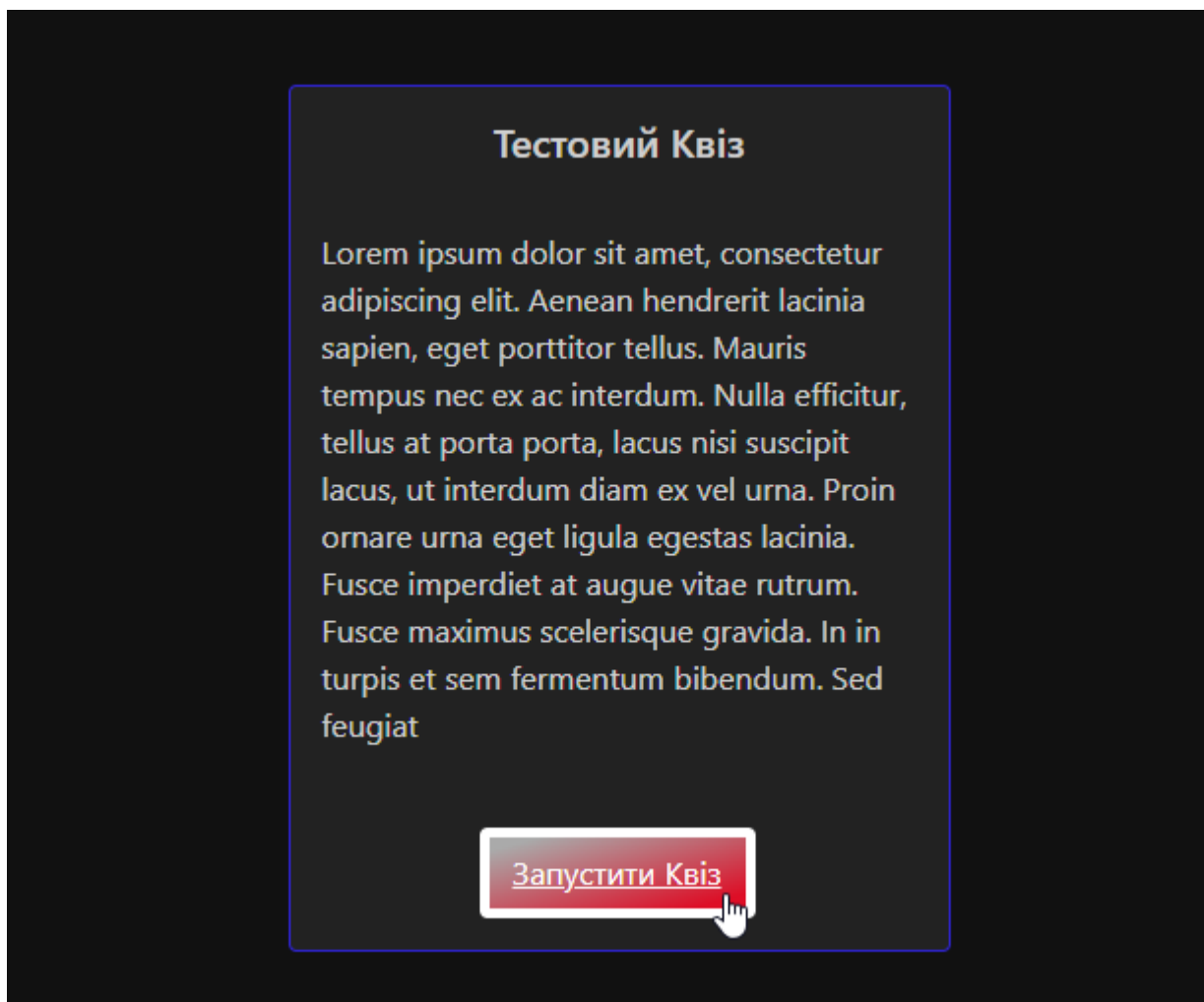


Рисунок 2.36 – Процес запуску квізу

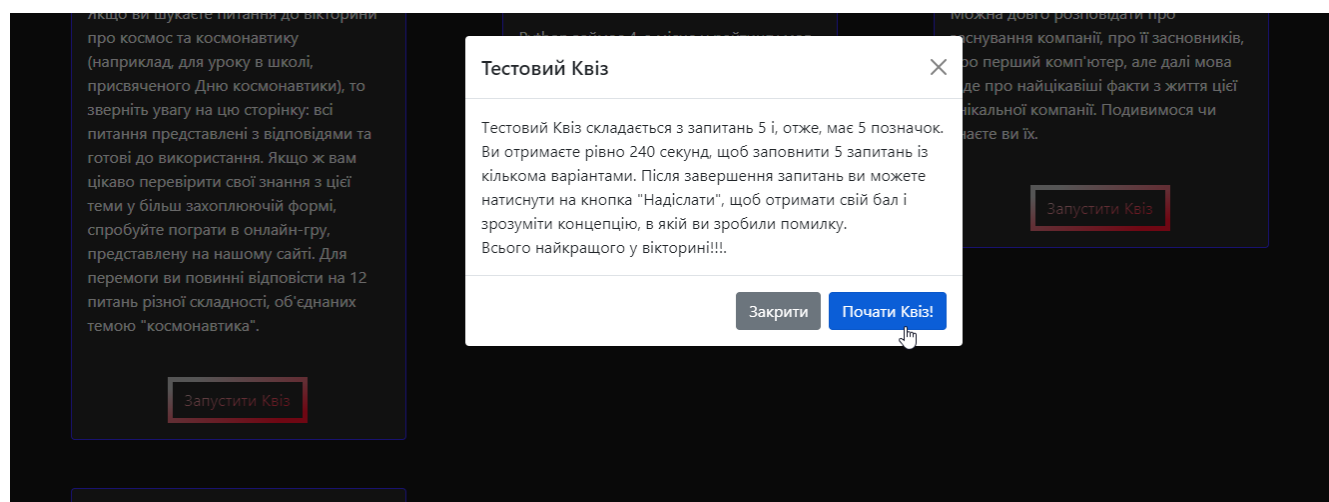


Рисунок 2.37 – Поп-ап з правилами квізу

Quiz Online

Тестовий Квіз

Столиця України?

☐ Київ
☐ Лондон
☐ Варшава
☐ Миколаїв

Підтвердити

Залишилося секунд : 238

Рисунок 2.38 – Процес проходження квізу

Quiz Online

Тестовий Квіз

Столиця України?

☒ Київ
☐ Лондон
☐ Варшава
☐ Миколаїв

Підтвердити

Ваш результат : 1 з 5 баллів

Столиця України? Ваша Відповідь: Київ

Рисунок 2.39 – Результати проходження квізу

Під час перевірки було проведено поетапне тестування вебсайту та основних функцій. В результаті тестування можна зробити висновок, що вебсайт працює без помилок і всі функції, що тестуються виконуються нормально.

3 РЕЗУЛЬТАТИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З КВІЗ ТЕМАТИКОЮ

В процесі виконання дипломного проєкту було розроблено програмне забезпечення вебсайт з системою квіз. Розроблене програмне забезпечення задовольняє вимогам, які поставлені у технічному завданні.

Програмне забезпечення розроблено мовою програмування Python з використанням Django.

Проведене тестування програмного забезпечення продемонструвало повну працездатність програми, відповідність її характеристик та виконуваних функцій технічному завданню, готовність до впровадження. Випробування були виконані згідно з програмою та методикою випробувань, яку наведено у Додатку В.

Програмне забезпечення сервера вимагає встановлення інтерпретатора Python.

Для роботи з даним вебсайтом користувач може використовувати будь-який браузер.

Інструкція користувача для роботи з вебсайтом наведена в Додатку Г.

4 ОХОРОНА ПРАЦІ

4.1. Основні положення та норми охорони праці.

Охорона праці – система забезпечення безпеки життя і здоров'я робітників в процесі трудової діяльності, включаючи правові, соціально-економічні, санітарно гігієнічні та лікувально-профілактичні, реабілітаційні та інші заходи.

Функціями охорони праці є дослідження санітарії та гігієни праці, проведення заходів щодо зниження впливу шкідливих факторів на організм працівників в процесі праці.

Основним методом охорони праці є використання техніки безпеки. При цьому вирішуються два основних завдання: створення машин та інструментів, при роботі з якими виключена безпека для людини, та розробка спеціальних засобів захисту, що забезпечують безпеку людини в процесі праці, також проводиться навчання працюючих безпечним прийомам праці та використання засобів захисту, створюються умови для безпечної роботи.

Основна мета покращення умов праці – досягнення соціального ефекту, тобто забезпечення безпеки праці, збереження життя та здоров'я працюючих, скорочення кількості нещасних випадків та захворювань на виробництві.

Покращення умов праці дає і економічні результати: ріст прибутку (у зв'язку з підвищенням продуктивності праці); скорочення витрат, пов'язаних з компенсацією за роботу з шкідливими та важкими умовами праці, зменшення втрат пов'язаних з травматизмом, професійною захворюваністю, зменшенням плинності кадрів тощо.

Робоче місце – просторова зона, оснащена необхідними засобами, в якій здійснюється трудова діяльність робітника або групи робітників, які спільно виконують виробничі завдання. Робоче місце є частиною виробничо-технологічної структури підприємства (організації), воно призначене для виконання частини технологічного (виробничого) процесу та визначається на основі трудових та

інших діючих норм та нормативів.

Робоча зона – простір, обмежений по висоті двома метрами над рівнем підлоги та площадки, яких знаходяться місця постійного або непостійного (тимчасового) перебування працюючих. До постійних належать місця, на яких працюючий знаходиться більше 50% робочого часу протягом зміни або більше двох годин безперервно. Якщо робота здійснюється в різних пунктах робочої зони, то постійним робочим місцем вважається вся робоча зона.

Умови праці – сукупність факторів виробничого середовища, що впливають на здоров'я та працездатність людини в процесі праці. Дослідження умов праці показали, що факторами виробничого середовища в процесі праці є:

- санітарно-гігієнічна обстановка, що визначає зовнішнє середовище в робочій зоні – мікроклімат, механічне коливання, випромінювання, температуру, освітлення та інші;
- психофізіологічні елементи: робоча поза, фізичне навантаження, нервово-психологічна напруга та інші, які обумовлені процесом праці;
- соціально-психологічні елементи, що складають характеристику так званого психологічного клімату.

Професійним захворюванням називають захворювання, викликане впливом шкідливих умов праці. До них належать: хронічні пилові бронхіти, вібраційна хвороба, отруєння різними токсичними речовинами та інші професійні захворювання в залежності від тяжкості та строків виявлення, можуть супроводжуватися та не супроводжуватися втратою працездатності. У важких випадках вони можуть призвести до інвалідності.

До основних функцій управління охороною праці належать:

- облік показників стану умов і безпеки праці;
- аналіз та оцінка стану умов і безпеки праці;
- прогнозування і планування робіт, їх фінансування;
- організація та координація робіт;
- контроль за функціонуванням СУОП.

Основні завдання управління охороною праці:

- навчання працівників безпечним методам праці та пропаганда питань охорони праці;
- забезпечення безпечності технологічних процесів, устаткування, будівель і споруд;
- професійної підготовки і підвищення кваліфікації працівників з питань охорони праці, пропаганди безпечних методів праці;
- вибору оптимальних режимів праці і відпочинку працівників;
- професійного добору виконавців для визначених видів робіт.

Правові та організаційні питання:

- нормалізація санітарно-гігієнічних умов праці;
- забезпечення працівників засобами індивідуального захисту;
- забезпечення оптимальних режимів праці та відпочинку;
- організація лікувально-профілактичного обслуговування;
- удосконалення нормативної бази з питань охорони праці.

4.2. Аналіз небезпечних і шкідливих факторів під час роботи з екранними пристроями.

Шкідливим фактором називається такий фактор, вплив якого на робітника в певних умовах призводить до захворювання або зниження працездатності.

Небезпечний фактор – фактор, дія якого за певних умов може призвести до травм або іншого раптового погіршення здоров'я працівника.

Щодня перед початком роботи необхідно очищати екранні пристрої від пилу та інших забруднень.

Мінімальні вимоги безпеки під час роботи з екранними пристроями:

- після закінчення роботи екранні пристрої слід відключати від електричної мережі;
- у разі виникнення аварійної ситуації необхідно негайно відключити екранний пристрій від електричної мережі;
- не допускається: виконувати технічне обслуговування, ремонт і налагодження екранних пристроїв безпосередньо на робочому місці працівника під час роботи з екранними пристроями;
- відключати захисні пристрої, самочинно проводити зміни у конструкції та складі екранних пристроїв або їх технічне налагодження;
- працювати з екранними пристроями, у яких під час роботи виникають нехарактерні сигнали, нестабільне зображення на екрані та інші несправності;
- під час виконання робіт операторського типу, пов'язаних з нервово-емоційним напруженням, у приміщеннях під час роботи з екранними пристроями, на пультах і постах керування технологічними процесами та в інших приміщеннях мають дотримуватися оптимальні умови мікроклімату відповідно до вимог ДСН 3.3.6.042-99.

Мінімальні вимоги безпеки до екранних пристроїв:

- екранні пристрої не мають бути джерелом ризику для працівників;
- усе випромінювання, за винятком видимої частини електромагнітного спектра, має бути зведене до незначного рівня з погляду безпеки й охорони здоров'я працівників;
- символи на екранних пристроях мають бути чіткими, відповідного розміру. Між символами та рядками символів має бути належна відстань;
- зображення на екрані має бути стабільним, без миготінь або інших видів нестабільності;
- яскравість та/або контрастність символів має легко регулюватися працівником під час роботи з екранними пристроями, а також швидко адаптуватися до навколишніх умов;
- вибираючи екрани, слід надавати перевагу таким екранам, які легко та вільно повертаються і нахиляються відповідно до потреби працівника;

- за необхідності може використовуватись окрема підставка або регульований стіл для розміщення екрана;
- екран не має відблискувати або відбивати світло, щоб не викликати дискомфорту у працівника під час роботи з екранними пристроями;
- вибираючи клавіатуру, слід надавати перевагу такій клавіатурі, яка відкидається і є автономною (відокремленою від екрана), щоб працівник міг вибрати зручну робочу позу й уникнути втоми рук (кисті та верхньої частини руки);
- поверхня клавіатури має бути матовою, щоб уникнути віддзеркалювання. Розташування клавіш і самі клавіші мають полегшувати роботу із клавіатурою. Позначення клавіш повинно бути достатньо контрастним і розбірливим;
- устаткування, яке входить до робочої станції, не має виділяти надлишкового тепла, що може спричинити незручності працівникам під час роботи з екранними пристроями;
- під час розробки, вибору, замовлення та модифікації програмного забезпечення, а також під час розробки завдань, що передбачають використання устаткування з екранними пристроями, роботодавець має керуватися таким програмним забезпеченням, яке відповідає розв'язуванню завданням і є простим у використанні, а де необхідно - адаптованим до рівня знань і досвіду працівника [14].

4.3. Характеристика умов праці програміста

Науково-технічний прогрес приніс серйозні зміни в умови виробничої діяльності робітників розумової праці. Їх праця стала більш інтенсивним, напруженим, які вимагають значних витрат розумової, емоційної та фізичної енергії. Це зажадало комплексного розв'язання проблем ергономіки, гігієни й організації праці, регламентації режимів праці та відпочинку.

В цей час комп'ютерна техніка широко застосовується у всіх областях діяльності людини. При роботі з комп'ютером людина піддається дії ряду

небезпечних і шкідливих виробничих факторів: електромагнітних полів (діапазон радіочастот: ВЧ, УВЧ і СВЧ), інфрачервоного та іонізуючого випромінювання, шуму і вібрації, статичної електрики та ін [6].

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи й достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ. Велике значення має раціональна конструкція і розташування елементів робочого місця, що важливо для підтримки оптимальної робочої пози людини-оператора.

У процесі роботи з комп'ютером необхідно дотримувати правильний режим праці та відпочинку. В іншому випадку у персоналу наголошуються значна напруга зорового апарату з появою скарг на незадоволеність роботою, головні болі, дратівливість, порушення сну, втому і хворобливі відчуття в очах, в поясниці, в області ший і руках.

4.4. Забарвлення, коефіцієнти віддзеркалення, освітлення

Забарвлення приміщень і меблів повинні сприяти створенню сприятливих умов для зорового сприйняття, гарного настрою.

Джерела світла, такі як світильники й вікна, які дають віддзеркалення від поверхні екрана, значно погіршують точність знаків і тягнуть за собою перешкоди фізіологічного характеру, які можуть виразитися в значній напрузі, особливо при тривалій роботі. Віддзеркалення, включаючи віддзеркалення від вторинних джерел світла, повинне бути зведено до мінімуму. Для захисту від надмірної яскравості вікон можуть бути застосовані штори та екрани [10].

Залежно від орієнтації вікон рекомендується наступне фарбування стін і підлоги:

- вікна орієнтовані на південь: - стіни зеленувато-блакитного або світло-блакитного кольору; підлога - зелений;
- вікна орієнтовані на північ: - стіни світло-оранжевого або оранжево-

жовтого кольору; підлога - червонувато-оранжевий;

- вікна орієнтовані на схід: - стіни жовто-зеленого кольору, підлога зелена або червонувато-оранжевий;

- вікна орієнтовані на захід: - стіни жовто-зеленого або голубувато-зеленого кольору; підлога зелена або червонувато-оранжевий.

У приміщеннях, де знаходиться комп'ютер, необхідно забезпечити наступні величини коефіцієнта віддзеркалення: для стелі: 60 ... 70%, для стін: 40 ... 50%, для підлоги: близько 30%. Для інших поверхонь і робочих меблів: 30 ... 40%

Правильно спроектований і виконане виробниче освітлення покращує умови зорової роботи, знижує стомлюваність, сприяє підвищенню продуктивності праці, добродійно впливає на виробниче середовище, надаючи позитивну психологічну дію на працюючого, підвищує безпеку праці і знижує травматизм.

Недостатність освітлення приводить до напруги зору, ослабляє увагу, приводить до настання передчасної стомленості. Надмірно яскраве освітлення викликає засліплення, роздратування і різь в очах. Неправильний напрямок світла на робочому місці може створювати різкі тіні, відблиски, дезорієнтувати працюючого. Всі ці причини можуть призвести до нещасного випадку або профзахворювань, тому такий важливий правильний розрахунок освітленості.

Існує три види освітлення - природне, штучне і поєднане (природне і штучне разом) [11].

Природне освітлення - освітлення приміщень денним світлом, що потрапляє через світлові прорізи в зовнішніх огорожуючих конструкціях приміщення. Природне освітлення характеризується тим, що змінюється в широких межах залежно від часу дня, пори року, характеру області і ряду інших чинників.

Штучне освітлення застосовується при роботі в темний час доби і вдень, коли не вдається забезпечити нормовані значення коефіцієнта природного освітлення (похмура погода, короткий світловий день). Освітлення, при якому недостатнє за нормами природне освітлення доповнюється штучним,

називається змішаним освітленням.

Штучне освітлення підрозділяється на робоче, аварійне, евакуаційне, охоронне. Робоче освітлення, у свою чергу, може бути загальним або комбінованим. Загальне - освітлення, при якому світильники розміщуються у верхній зоні приміщення рівномірно, або, як розташоване устаткування. Комбіноване - освітлення, при якому до загального додається місцеве освітлення.

Згідно СНіП II-4-79 в приміщень обчислювальних центрів необхідно застосувати систему комбінованого освітлення.

При виконанні робіт категорії високої зорової точності (найменший розмір об'єкту розрізнення 0,3 ... 0,5 мм) величина коефіцієнта природного освітлення (КЕО) повинна бути не нижче 1,5%, а при зоровій роботі середньої точності (найменший розмір об'єкту розрізнення 0,5 ... 1,0 мм) КЕО повинен бути не нижче 1,0%. В якості джерел штучного освітлення звичайно використовуються люмінесцентні лампи типа ЛБ, або ДРЛ, які попарно об'єднуються в світильники, які повинні розташовуватися рівномірно над робочими поверхнями [10].

Вимоги до освітленості в приміщеннях, де встановлені комп'ютери, наступні: при виконанні зорових робіт високої точності загальна освітленість повинна складати 300лк, а комбінована - 750лк; аналогічні вимоги при виконанні робіт середньої точності - 200 і 300лк відповідно.

Крім того все поле зору повинне бути освітлено достатньо рівномірно - ця основна гігієнічна вимога. Іншими словами, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими, оскільки яскраве світло в районі периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності.

4.5. Параметри мікроклімату. Шум та вібрації

Параметри мікроклімату можуть мінятися в широких межах, у той час як необхідною умовою життєдіяльності людини є підтримка постійності температури тіла завдяки терморегуляції, тобто здатності організму регулювати

віддачу тепла в навколишнє середовище. Принцип нормування мікроклімату - створення оптимальних умов для теплообміну тіла людини з навколишнім середовищем.

Шум погіршує умови праці надаючи шкідливу дію на організм людини. Працюючі в умовах тривалої шумової дії випробовують дратівливість, головні болі, запаморочення, зниження пам'яті, підвищену стомлюваність, зниження апетиту, біль у вухах і т.д. Такі порушення в роботі ряду органів і систем організму людини можуть викликати негативні зміни в емоційному стані людини аж до стресових. Під впливом шуму знижується концентрація уваги, порушуються фізіологічні функції, з'являється втома у зв'язку з підвищеними енергетичними витратами і нервово-психічним напруженням, погіршується мовна комутація. Все це знижує працездатність людини і її продуктивність, якість і безпеку праці. Тривала дія інтенсивного шуму [вище 80 дБ (А)] на слух людини приводить до його часткової або повної втрати [12].

4.6. Протипожежний захист

Первинні засоби пожежогасіння призначені для ліквідації невеликих осередків пожеж, а також для гасіння пожеж у початковій стадії їх розвитку силами персоналу об'єктів до прибуття штатних підрозділів пожежної охорони [16].

До первинних засобів пожежогасіння відносяться: вогнегасники (вогнегасник ВП-10—порошковий з місткістю корпусу 10 л; вогнегасник ВВ-5 — вуглекислотний з місткістю корпусу 5 л); пожежний інвентар (покривала з негорючого теплоізоляційного полотна, грубововняної тканини, ящики з піском, бочки з водою, пожежні відра, совкові лопати); пожежний інструмент (гаки, лом, сокири тощо).

Вогнегасники та пожежний інвентар повинні мати червоне пофарбування, а бочки з водою та ящики з піском ще й відповідні написи білою фарбою. Пожежний інструмент фарбується у чорний колір [17].

Бочки для зберігання води з метою пожежогасіння встановлюються у виробничих, складських та інших приміщеннях, будівлях та спорудах у разі відсутності внутрішнього протипожежного водогону та за наявності горючих матеріалів, а також на території підприємств. Їх кількість у приміщеннях визначається з розрахунку установки однієї бочки, місткістю не менше 0,2 м³ на 250—300 м² захищеної площі. Такі бочки повинні бути укомплектовані пожежним відром місткістю не менше 8 л [18].

Ящики для піску повинні мати місткість 0,5, 1,0 або 3,0 м³ та бути укомплектовані совковою лопатою.

Пожежна безпека передбачає такий стан об'єктів народного господарства, при якому виключається можливість пожежі, а у випадку його виникнення попереджається дія на людей шкідливих факторів пожежі та забезпечується захист матеріальних цінностей.

Пожежна безпека забезпечується системами попередження пожежі й пожежного захисту, які включають комплекс організаційних заходів і технічних засобів.

Заходи щодо профілактики протипожежного захисту поділяються на організаційні, експлуатаційні, технічні та режимні.

Організаційні заходи: навчання робочих та службовців правилам пожежної безпеки, організація пожежної охорони, проведення бесід, лекцій, видання необхідних інструкцій, плакатів тощо.

Технічні заходи передбачають дотримання протипожежних правил та норм при улаштуванні систем опалення, вентиляції, кондиціонування повітря, захисту від блискавки, при спорудженні будівлі, установленні технологічного обладнання тощо.

Експлуатаційні заходи передбачають правильну експлуатацію систем опалення, вентиляції та кондиціонування повітря, захисту від блискавки, технологічних машин і обладнання, правильне утримання будівель та території тощо.

Режимні заходи передбачають заборону або граматичне застосування

відкритою вогню в пожежонебезпечних місцях, куріння в невстановлених місцях, обов'язкове дотримання норм і правил при роботі з вогненебезпечними та вибухонебезпечними речовинами.

Заходи протипожежного захисту в системах вентиляції та кондиціонування повітря здійснюють з метою попередження можливості появи вибухонебезпечних концентрацій газо-, паро- і пило- повітряних речовин, як за об'ємом всього приміщення, так і за об'ємом тої чи іншої його частини.

Відповідальність за пожежну безпеку на підприємствах і своєчасне виконання протипожежних заходів покладається на керівників підприємства, цехів, відділів, майстерень, лабораторій та інших ділянок.

ВИСНОВКИ

В результаті виконання дипломного проєкту було розроблено вебсайт який відповідає всім вимогам описаним в Технічному завданні.

Результати дослідження предметної області дозволили:

- визначити актуальність поставленої проблеми та актуальності веб-додатку;
- дослідити та проаналізувати існуючі сайти-аналоги;
- сформулювати мету та задачі дипломного проєкту для використання готового продукту
- проведено детальний аналіз існуючих технологій та обрати раціональний та сучасний спосіб реалізації веб-додатку відповідно до визначених вимог.

Було проведено проєктування веб-додатку. На даному етапі розробки веб-додатку було створено UBL діаграми, а також діаграму варіантів використання, що визначає головних акторів та сценарії їх взаємодії з компонентами веб-додатку. Як результат етапу планування виконання даного проєкту було створено всі необхідні діаграми для спрощення керуванням проєкту.

Система реалізована з використанням сучасних WEB-технологій та програмних засобів розробки WEB-застосувань: Python, Django, технології HTML та CSS.

У розділі з охорони праці було розглянуто основні положення про охорону праці та проаналізовано небезпечні та шкідливі фактори при роботі з екранними пристроями, зібрана інформація про протипожежний захист, визначення умов, що характеризують оптимальну організацію робочого місця інженера - програміста, дозволять зберегти гарну працездатність протягом усього робочого дня, підвищать як в кількісному, так і в якісному відношенні продуктивність праці програміста, що своєю чергою сприятиме якнайшвидшій розробці й налагодженню програмного продукту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. П'ять сервісів для створення онлайн-вікторин [Електронний ресурс]. – Режим доступу: <https://ms.detector.media/>
2. «Что такое СУБД?» [Електронний ресурс]. – Режим доступу: <https://help.reg.ru/>
3. «Что такое PostgreSQL?» [Електронний ресурс]. – Режим доступу: <https://azure.microsoft.com/>
4. Підручник мови Python/Вступ [Електронний ресурс]. – Режим доступу: [https://uk.wikibooks.org/wiki/Підручник_мови_Python/Вступ#:~:text=Пайтон%20\(Python\)%20%20це%20потужна,об'єктноорієнтованого%20програмування.&text=Пайтон%20також%20зручний%20як%20мова,програми%20С%20для%20додаєткових%20налаштувань%20функціональності.](https://uk.wikibooks.org/wiki/Підручник_мови_Python/Вступ#:~:text=Пайтон%20(Python)%20%20це%20потужна,об'єктноорієнтованого%20програмування.&text=Пайтон%20також%20зручний%20як%20мова,програми%20С%20для%20додаєткових%20налаштувань%20функціональності.)
5. Що таке Python? [Електронний ресурс]. – Режим доступу: <http://www.plugin.org.ua/documentation/about-python>
6. Безпека життєдіяльності. / Под ред. Н.А. Белова - М.: Знання, 2000 - 364с.
7. Дубовці В.А. Безпека життєдіяльності. / Учеб. посібник для дипломників. - К.: вид. Кірпи, 1992.
8. «Что такое квиз-маркетинг и как он помогает находить клиентов?» [Електронний ресурс]. - – Режим доступу: <https://skillbox.ru/media/marketing/>
9. Мотузко Ф.Я. Охорона праці. - М.: Вища школа, 1989. - 336с.
10. Самгін Е.Б. Освітлення робочих місць. - М.: МІРЕА, 1989. - 186с.
11. Довідкова книга для проектування електричного освітлення. / Под ред. Г.Б. Кнорринга. - Л.: Енергія, 1976.
12. Боротьба з шумом на виробництві: Довідник / Є.Я. Юдін, Л.А. Борисов; За заг. ред. Є.Я. Юдіна - М.: Машинобудування, 1985. - 400с., Іл.
13. Зінченко В.П. Основи ергономіки. - М.: МГУ, 1979. - 179с.
14. Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями [Електронний ресурс]. –

Режим доступу: <https://zakon.rada.gov.ua/laws/show/z0508-18#Text>

15. ДСТУ 8302:2015. Бібліографічне посилання. Загальні положення та правила складання. [Чинний від 2016–07–01]. Київ: Держспоживстандарт України, 2016. 16 с. (Інформація та документація).
16. Рожков А.П. Пожежна безпека на виробництві.- К.: Основа, 1997.- 448с.
17. Жидецький В.Ц., Джигирей В.С., Мельников О.В. Основи охорони праці. Навчальний посібник.-Вид. 4-те доповнене.- Львів: Афіша, 2000.- 351с.
18. ДБН В.2.5-56:2014 «Системи протипожежного захисту».
19. ДСТУ 3008:2015. Звіти у сфері науки і техніки. Структура і правила оформлення. [Чинний від 2017–07–01]. Київ. Держспоживстандарт України, 2016. 26 с. (Інформація та документація).
20. Гутманс Э., Баккен С., Ретанс Д. РНР 5. Профессиональное программирование. пер. с англ. СПб.: Символ-Плюс, 2006. 704 с.
21. Тестирование по стратегии чёрного ящика [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/>

ДОДАТОК А – ТЕХНІЧНЕ ЗАВДАННЯ

1. Вступ.

Назва програмного продукту: "Quiz".

Позначення програми: Розважальний веб-сервіс.

Створення системи передбачає використання браузера на комп'ютері або мобільному пристрої, пошук матеріалів для заповнення та тестування бази даних.

2. Підстави для розробки.

Підставами для розробки цього веб-сайту є:

- збільшення популярності сервісів із системою квіз;
- відсутність на ринку України та східної Європи аналогів, які могли б відповідати стандартам 2022 року та задовольняти відразу кілька категорій користувачів.

3. Призначення розробки.

3.1. Функціональне призначення

Функціональним призначенням програми - за допомогою гри у вікторини, розвиток пам'яті, зняття стресу, розвиток уяви та творчості при створенні своїх вікторин, тестів, опитувань.

3.2. Експлуатаційне призначення.

Програма повинна експлуатуватися в браузері такими людьми як: користувач, модератор.

- Користувач – повинен мати змогу створити ігровий набір, запустити його на сайті, створити питання до квізу.
- Модератор – повинен мати змогу видалити пак, Перевірка на якість створених наборів.

4. Вимоги до програми або програмного виробу.

4.1. Вимоги до програмних інструментів для розробки.

Вихідні коди програми повинні бути реалізовані на мові Python3. В якості середовища розробки програм повинна бути використана PyCharm. В якості бази даних використовувати СУБД PostgreSQL.

4.2. Вимоги до складу виконуваних функцій.

Програмне забезпечення має забезпечувати користувачів можливість виконання перерахованих нижче функцій:

- перегляд ігрових паків та наборів за категоріями (назва набору, дата створення, опис набору, тривалість гри);
- створення профілю;
- створення свого пакета, набору, квіза (назва набору, дата створення, опис набору, тривалість гри);
- перегляд створених паків, наборів, квізів;
- створення питання до квізу;
- додавання варіантів відповіді;
- редагування вже створених ігрових наборів;
- видалення ігрового набору (id набору).

Програмне забезпечення має забезпечувати для модератора можливість виконання перерахованих нижче функцій:

- перегляд списку наборів (назва набору, дата створення, опис набору,

тривалість гри);

- видалення набору (id набору).

4.3. Вимоги до організації вхідних та вихідних даних.

Введення інформації здійснюється за допомогою кнопок управління на сайті та клавіатури, при введенні даних у спеціально відведені для цього поля. Вихідна інформація відображається на вебсторінках за допомогою їхнього вмісту. Вся інформація зберігається на сервері у відповідній базі даних.

4.4. Вимоги до надійності.

4.4.1. Вимоги до забезпечення надійного (сталого) функціонування програми.

Надійне (стійке) функціонування програми має бути забезпечене виконанням сукупності організаційно-технічних заходів, перелік яких наведено нижче:

- організацією безперебійного живлення технічних засобів;
- використанням ліцензійного програмного забезпечення.

4.4.2. Час відновлення після відмови.

Час відновлення після відмови, викликаной збоєм електроживлення на сервері (іншими зовнішніми факторами), не фатальним збоєм операційної системи, не повинно перевищувати 5-ти хвилин.

Час відновлення після збою, викликаного несправністю технічних засобів, фатальним збоєм операційної системи, не повинно перевищувати час, необхідний для усунення несправностей технічних засобів і переустановлення програмних засобів.

4.4.3. Відмови через некоректність дій користувача.

Відмови програми неможливі внаслідок некоректних дій користувача при відправленні даних. Через те, що при введенні некоректних даних програма не буде виконувати запит користувача.

4.5. Вимоги до видів обслуговування.

Програма не потребує проведення будь-яких видів обслуговування.

4.6. Вимоги до чисельності та кваліфікації персоналу.

Мінімальна кількість персоналу, необхідна для функціонування програми, повинна складати одну особу — модератор.

Модератор повинен мати базові навички роботи з комп'ютером.

4.7. Вимоги до складу і параметрів технічних засобів.

4.7.1. Backend.

В склад технічних засобів повинен входити сервер з наступними параметрами:

- операційна система: Linux;
- процесор: Intel Celeron G4900, 3.1 GHZ, 2 ядра та більше;
- інтернет: 30 Мбит/с та більше.
- оперативна пам'ять: мінімум 4Гб.
- твердий диск: для нормальної роботи системі потрібно 50Мб вільного дискового простору.

4.7.2. Frontend.

В склад технічних засобів повинен входити комп'ютер, або смартфон з мінімальними характеристиками, який підтримує встановлення програмного додатку Google Chrome.

Android:

- версія ОС: Android 10.0 Marshmallow або пізнішої версії.
 - інтернет: потрібно;
 - потрібно вільного місця: 1 Mb.
- IOS:
- версія ОС: iOS 14.0 +;
 - інтернет: потрібно;
 - потрібно вільного місця: 162,6 Mb.

Комп'ютер:

- ОС: Windows XP, 7, Vista, 8, 8.1, 10;
- процесор: Intel Pentium 4/Athlon 64 або пізнішої версії з підтримкою SSE2
- оперативна пам'ять: 512 MB ОЗУ;
- відеокарта: Intel HD Graphics;
- місце на диску: 350 MB.
- інтернет: потрібно;

4.8. Вимоги до інформаційної і програмної сумісності.

4.8.1. Вимоги до інформаційних структур і методів розв'язку.

Вимоги до інформаційних структур (файлів) на вході та виході, а також до методів вирішення не висуваються.

4.8.2 Вимоги до вихідних кодів та мов програмування.

Вихідні коди програми повинні бути реалізовані на мові Python3. В якості середовища розробки програм повинна бути використана PyCharm. В якості в якості СУБД використовувати PostgreSQL.

4.8.3. Вимоги до програмних засобів, що використовуються програмою.

Системні програмні засоби, що використовуються для взаємодії з браузером Google Chrome повинні бути представлені ліцензійною локалізованою версією операційної системи.

4.8.4. Вимоги до захисту інформації та програм.

Відповідно до політики конфіденційності та умови браузера.

4.9. Вимоги до маркування.

Доступ до програмного додатку здійснюється за допомогою інтернет браузера

4.10. Вимоги до транспортування та збереження.

Вимоги до транспортування та збереження програми не пред'являються.

5. Вимоги до програмної документації.

Попередній склад програмної документації повинен включати в себе:

- технічне завдання;
- текст програмних модулів;
- програма та методика випробовування;
- опис програми;
- інструкція користувача.

6. Техніко-економічні показники.

Техніко-економічні показники не розраховуються.

7. Стадії та етапи розробки.

Стадії та етапи розробки представлено в таблиці 1.

Таблиця 1
Стадії та етапи розробки

№	Стадії розробки	Етапи розробки	Зміст робіт по етапах	Строки виконання
1	2	3	4	5
1	Технічне завдання	Розробка, узгодження та затвердження технічного завдання	Постановка задачі; визначення та уточнення вимог до технічних засобів; визначення вимог до програми; визначення стадій, етапів і термінів розробки програми і документації на неї; узгодження і затвердження технічного завдання.	Початок: 14.02.22 Закінчення: 23.03.22
2	Ескізний проект	Розробка та затвердження ескізного проекту	Виявлення вимог, концепція майбутньої системи, складання ескізної частини програмного продукту	Початок: 23.03.22 Закінчення: 31.03.22
3	Технічний проект	Розробка та затвердження технічного проекту	Загальні системні вирішення, алгоритми задач, оцінка економічної ефективності автоматизованої системи та перелік заходів підготовки об'єкта для провадження	Початок: 31.03.22 Закінчення: 13.05.22

4	Робочий проект	Розробка та затвердження робочого проекту	Деталізовані загальносистемні проекти рішення, програми та інструкції для розв'язку завдань, розробка програмного забезпечення.	Початок: 13.05.22 Закінчення: 10.06.22
---	----------------	---	---	---

Продовження таблиці 1

8. Порядок контролю та приймання.

Прийомка та контроль програми повинні проводитися відповідно до узгоджених заздалегідь із замовником програми і методики випробувань

Кожна стадія розробки повинна бути представлена в зазначені строки та узгоджена із замовником (викладачем).

Хід проведення приймально-здавальних випробувань документують за допомогою протоколу проведення випробувань, відповідно до програми й методики випробувань.

Тестування програмного продукту повинно проводитися у відповідності до узгоджених заздалегідь із замовником програмного продукту методики випробувань. Тестування проводиться в зазначені строки.

На підставі протоколу проведення випробувань виконувач сумісно з замовником підписують акт приймання - здачі програмного забезпечення в експлуатацію

У разі знаходження помилок під час прийому програмного виробу складається акт про знайдені помилки, який підписується представниками замовника і розробника та затверджується керівниками організації – замовника та організації – розробника. Розробник повинен на протязі не більше ніж 2-х тижнів виправити зазначені помилки й оповістити замовника про повторне проведення перевірки.

ДОДАТОК Б – ТЕКСТ ПРОГРАМИ

QuizApp/urls.py

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('home.urls')),
```

```
] + static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

urls.py

```
from django.urls import path
from . import views
```

```
urlpatterns = [
    path("", views.index, name="index"),
    path("<int:myid>/", views.quiz, name="quiz"),
    path('<int:myid>/data/', views.quiz_data_view, name='quiz-
data'),
    path('<int:myid>/save/', views.save_quiz_view, name='quiz-
save'),

    path("signup/", views.Signup, name="signup"),
```

```

    path("login/", views.Login, name="login"),
    path("logout/", views.Logout, name="logout"),

    path('add_quiz/', views.add_quiz, name='add_quiz'),
    path('add_question/', views.add_question,
name='add_question'),
    path('add_options/<int:myid>/', views.add_options,
name='add_options'),
    # path('add_quiz_options/<int:myid>/',
views.add_quiz_options, name='add_quiz_options'),
    path('results/', views.results, name='results'),
    path('delete_question/<int:myid>/', views.delete_question,
name='delete_question'),
    path('delete_quiz/<int:myid>/', views.delete_quiz,
name='delete_quiz'),
    path('delete_result/<int:myid>/', views.delete_result,
name='delete_result'),
]

```

views.py

```

from django.shortcuts import render, redirect
from .models import *
from django.http import JsonResponse
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.models import User
from django.contrib.auth.decorators import login_required
from .forms import QuizForm, QuestionForm
from django.forms import inlineformset_factory

```

```

def index(request):
    quiz = Quiz.objects.all()
    para = {'quiz' : quiz}
    return render(request, "index.html", para)

@login_required(login_url = '/login')
def quiz(request, myid):
    quiz = Quiz.objects.get(id=myid)
    return render(request, "quiz.html", {'quiz':quiz})

def quiz_data_view(request, myid):
    quiz = Quiz.objects.get(id=myid)
    questions = []
    for q in quiz.get_questions():
        answers = []
        for a in q.get_answers():
            answers.append(a.content)
        questions.append({str(q): answers})
    return JsonResponse({
        'data': questions,
        'time': quiz.time,
    })

def save_quiz_view(request, myid):
    if request.is_ajax():
        questions = []
        data = request.POST

```

```

data_ = dict(data.lists())

data_.pop('csrfmiddlewaretoken')

for k in data_.keys():
    print('key: ', k)
    question = Question.objects.get(content=k)
    questions.append(question)

user = request.user
quiz = Quiz.objects.get(id=myid)

score = 0
marks = []
correct_answer = None

for q in questions:
    a_selected = request.POST.get(q.content)

    if a_selected != "":
        question_answers =
Answer.objects.filter(question=q)
        for a in question_answers:
            if a_selected == a.content:
                if a.correct:
                    score += 1
                    correct_answer = a.content
            else:
                if a.correct:

```



```

        correct_answer = a.content

        marks.append({str(q): {'correct_answer':
correct_answer, 'answered': a_selected}})
    else:
        marks.append({str(q): 'not answered'})

    Marks_Of_User.objects.create(quiz=quiz, user=user,
score=score)

    return JsonResponse({'passed': True, 'score': score,
'marks': marks})

```

```

def Signup(request):
    if request.user.is_authenticated:
        return redirect('/')
    if request.method=="POST":
        username = request.POST['username']
        email = request.POST['email']
        first_name=request.POST['first_name']
        last_name=request.POST['last_name']
        password = request.POST['password1']
        confirm_password = request.POST['password2']

        if password != confirm_password:
            return redirect('/register')

        user = User.objects.create_user(username, email,

```

```

password)
    user.first_name = first_name
    user.last_name = last_name
    user.save()
    return render(request, 'login.html')
return render(request, "signup.html")

def Login(request):
    if request.user.is_authenticated:
        return redirect('/')
    if request.method=="POST":
        username = request.POST['username']
        password = request.POST['password']

        user = authenticate(username=username, password=password)

        if user is not None:
            login(request, user)
            return redirect("/")
        else:
            return render(request, "login.html")
    return render(request, "login.html")

def Logout(request):
    logout(request)
    return redirect('/')

def add_quiz(request):

```

```

quiz = Quiz.objects.all()
if request.method=="POST":
    form = QuizForm(data=request.POST)
    if form.is_valid():
        quiz = form.save(commit=False)
        quiz.save()
        obj = form.instance
        return render(request, "add_quiz.html", {'obj':obj})
    else:
        form=QuizForm()
    return render(request, "add_quiz.html", {'form':form,
'quiz':quiz})

# def add_quiz_options(request, myid):
#     quiz = Quiz.objects.get(id=myid)
#     if request.method=="POST":
#         form = QuizForm(data=request.POST)
#         if form.is_valid():
#             quiz = form.save(commit=False)
#             quiz.save()
#             obj = form.instance
#             return render(request, "add_quiz_options.html",
{'obj':obj})
#     else:
#         form=QuizForm()
#     return render(request, "add_quiz_options.html",
{'form':form, 'quiz':quiz})

```

```
def delete_quiz(request, myid):
    quiz = Quiz.objects.get(id=myid)
    if request.method == "POST":
        quiz.delete()
        return redirect('/add_quiz')
    return render(request, "delete_quiz.html", {'quiz':quiz})
```

```
def add_question(request):
    questions = Question.objects.all()
    questions = Question.objects.filter().order_by('-id')
    if request.method=="POST":
        form = QuestionForm(request.POST)
        if form.is_valid():
            form.save()
            return render(request, "add_question.html")
    else:
        form=QuestionForm()
    return render(request, "add_question.html", {'form':form,
'questions':questions})
```

```
def delete_question(request, myid):
    question = Question.objects.get(id=myid)
    if request.method == "POST":
        question.delete()
        return redirect('/add_question')
    return render(request, "delete_question.html",
```

```
{'question':question})
```

```
def add_options(request, myid):
    question = Question.objects.get(id=myid)
    QuestionFormSet = inlineformset_factory(Question, Answer,
fields=('content','correct', 'question'), extra=4)
    if request.method=="POST":
        formset = QuestionFormSet(request.POST,
instance=question)
        if formset.is_valid():
            formset.save()
            alert = True
            return render(request, "add_options.html",
{'alert':alert})
        else:
            formset=QuestionFormSet(instance=question)
            return render(request, "add_options.html",
{'formset':formset, 'question':question})
```

models.py

```
from django.db import models
from django.contrib.auth.models import User
import random

class Quiz(models.Model):
    name = models.CharField("Назва", max_length=50)
    desc = models.CharField("Опис", max_length=500)
```

```

        number_of_questions = models.IntegerField("Кількість питаннь",
default=1)

```

```

        time = models.IntegerField("Тривалість", help_text="Тривалість
Квізу в секундах", default="1")

```

```

    def __str__(self):
        return self.name

```

```

    def get_questions(self):
        return self.question_set.all()

```

```

# class Edit(models.Model):

```

```

#     name = models.CharField("Назва", max_length=200)

```

```

#     desc = models.CharField("Опис", max_length=500)

```

```

#     number_of_questions = models.IntegerField("Кількість
питаннь", default=1)

```

```

#     time = models.IntegerField("Тривалість", help_text="Тривалість
Квізу в секундах", default="1")

```

```

#     quiz = models.ForeignKey(Quiz, on_delete=models.CASCADE)

```

```

#

```

```

#     def __str__(self):

```

```

#         return f"quiz: {self.quiz.name}, desc: {self.desc},
numb: {self.number_of_questions}, time: {self.time}"

```

```

class Question(models.Model):

```

```

    content = models.CharField("Питання", max_length=200)

```

```

    quiz = models.ForeignKey(Quiz, on_delete=models.CASCADE)

```

```

def __str__(self):
    return self.content

def get_answers(self):
    return self.answer_set.all()

class Answer(models.Model):
    content = models.CharField("Вариант", max_length=200)
    correct = models.BooleanField("Правильный", default=False)
    question = models.ForeignKey(Question,
on_delete=models.CASCADE)

    def __str__(self):
        return f"question: {self.question.content}, answer:
{self.content}, correct: {self.correct}"

```

forms.py

```

from django import forms
from .models import Quiz, Question, Answer
from django.contrib import admin

class QuizForm(forms.ModelForm):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields["name"].widget.attrs.update({
            'class': 'input512',

```

```

        'type': 'text',
        'name': 'name',
        'placeholder': "Введіть Назву",
    })
    self.fields["desc"].widget.attrs.update({
        'class': 'input1024',
        'type': 'text',
        'name': 'desc',
        'placeholder': "Введіть Опис",
    })
class Meta:
    model = Quiz
    fields = ('name', 'desc', 'number_of_questions', 'time')

class QuestionForm(forms.ModelForm):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields["content"].widget.attrs.update({
            'class': 'input512',
            'type': 'text',
            'name': 'content',
            'placeholder': "Введіть Питання",
        })

class Meta:
    model = Question
    fields = ('content', 'quiz')
```


admin.py

```

from django.contrib import admin

from .models import Quiz, Question, Answer, Marks_Of_User

admin.site.register(Quiz)

class AnswerInline(admin.TabularInline):
    model = Answer

class QuestionAdmin(admin.ModelAdmin):
    inlines = [AnswerInline]

admin.site.register(Question, QuestionAdmin)
admin.site.register(Answer)

admin.site.register(Marks_Of_User)

```

settings.py

```

from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See

```

<https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/>

```
# SECURITY WARNING: keep the secret key used in production
secret!
```

```
SECRET_KEY = 'n+qfzy!zoth6))00$8mn^yrly+7rfq0wygs1kpn7$XH87*as97'
```

```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True
```

```
ALLOWED_HOSTS = []
```

```
# Application definition
```

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'home.apps.HomeConfig',
]
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
```

```

'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

```

```

ROOT_URLCONF = 'QuizApp.urls'

```

```

TEMPLATES = [
    {
        'BACKEND':
'django.template.backends.django.DjangoTemplates',
        'DIRS': ["home/templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',

'django.contrib.messages.context_processors.messages',

            ],
        },
    },
]

```

```

WSGI_APPLICATION = 'QuizApp.wsgi.application'

```

```

# Database

```

<https://docs.djangoproject.com/en/3.1/ref/settings/#databases>

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'df6kqklonfgouo',
        'USER': 'znyxrlecupkpaz',
        'PASSWORD':
'67a3fcb6f24a5aea495850649f331b451dd2a9816095b1d49048f4930f03e042',
        'HOST': 'ec2-99-80-170-190.eu-west-1.compute.amazonaws.com',
        'PORT': '5432',
    }
}
```

Password validation

<https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators>

```
AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
```

```

'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator'
,
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator'
,
    },
]

```

Internationalization

<https://docs.djangoproject.com/en/3.1/topics/i18n/>

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

Static files (CSS, JavaScript, Images)

```
# https://docs.djangoproject.com/en/3.1/howto/static-files/
```

```
STATIC_URL = '/static/'
```

```
##Added Manually
```

```
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "static")
]
```

```
# Managing Media
```

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

```
MEDIA_URL = '/media/'
```

base.html

```
<!doctype html>
```

```
<html lang="en">
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <meta name="viewport" content="width=device-width, initial-
scale=1">
```

```
    <link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/boot-
strap.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
```

```
    <title>{% block title %} {% endblock %}</title>
```

```

{% block css %}
<style>
    .head1 {
    color: #11e2d0;
    font-size: 2rem;
    font-weight: bold;
    height: 50px;
    }

    .head2 {
    color: #EF4926;
    font-size: 2rem;
    font-weight: bold;
    }
</style>
{% endblock %}
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <div class="container-fluid">
            <a class="navbar-brand" href="/"><span
class="head1">Quiz</span> <span class="head2">Online</span></a>
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse"

```

```

id="navbarSupportedContent">
    {% if user.is_authenticated %}
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
            <a class="nav-link" href="#">Вітаю
{{request.user}}</a>
        </li>

        {% if user.is_authenticated %}
        <div class="collapse navbar-collapse"
id="navbarNavDarkDropdown">
            <ul class="navbar-nav">
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle"
href="#" id="navbarDarkDropdownMenuLink" role="button" data-bs-
toggle="dropdown" aria-expanded="false">
                        Створити
                    </a>
                    <ul class="dropdown-menu dropdown-menu-
dark" aria-labelledby="navbarDarkDropdownMenuLink">
                        <li><a class="dropdown-item"
href="/add_quiz/">Створити Квіз</a></li>
                        <li><a class="dropdown-item"
href="/add_question/">Створити Питання</a></li>
                        {% comment %}                <li><a class="dropdown-
item" href="/results/">Results of the Users</a></li>{% endcomment
%}
                    </ul>
                </li>

```



```

        </ul>
    </div>
        <li class="nav-item">
            <a class="nav-link" href="/logout">Вийти</a>
        </li>

        {% endif %}
    </ul>
    {% else %}
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
            <a class="nav-link" href="/signup">Регістрація</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="/login">Увійти</a>
        </li>
    </ul>
    {% endif %}
</div>
</div>
</nav>
{% block body %} {% endblock %}

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstr
ap.bundle.min.js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
```

```
crossorigin="anonymous"></script>
</body>
    {% block js %}{% endblock %}
</html>
```

index.html

```
{% extends 'base.html' %}
{% block title %} Quiz {% endblock %}
{% block css %}
<style>
    .attempt {
        --gradient: linear-gradient(to left top, #de0e26 10%, #aaaaaa
90%) !important;
    }

    body {
        background: #111;
    }

    .card {
        background: #222;
        border: 1px solid #3024dd;
        color: rgba(250, 250, 250, 0.8);
        margin-bottom: 3rem;
    }

    #btn {
        border: 5px solid;
```

```

border-image-slice: 1;
background: var(--gradient) !important;
-webkit-background-clip: text !important;
-webkit-text-fill-color: transparent !important;
border-image-source: var(--gradient) !important;
text-decoration: none;
position: relative;
left: 80px;
}

```

```

.head1 {
  color: #11e2d0;
  font-size: 2rem;
  font-weight: bold;
  height: 50px;
}

```

```

.head2 {
  color: #EF4926;
  font-size: 2rem;
  font-weight: bold;
}

```

```
#btn:hover,
```

```

#btn:focus {
  background: var(--gradient) !important;
  -webkit-background-clip: none !important;
  -webkit-text-fill-color: #fff !important;
  border: 5px solid #fff !important;
  box-shadow: #222 1px 0 10px;
}

```

```

    text-decoration: underline;
}

.logo {
    position: relative;
    bottom: 2rem;
}

.container-fluid {
    height: 60px;
}
</style>
{% endblock %}
{% load static %}

{% block body %}

<!-- Modal -->
<div class="modal fade" id="exampleModal" tabindex="-1" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="modal-title"></h5>
                <button type="button" class="btn-close" data-bs-
dismiss="modal" aria-label="Close"></button>
            </div>
            {% if user.is_authenticated %}
            <div class="modal-body" id="modal-body">

```

```

</div>
<div class="modal-footer">
    <button type="button" class="btn btn-secondary" data-bs-
dismiss="modal">Закрити</button>
    <button type="button" id="start-quiz" class="btn btn-
primary">Почати Квіз!</button>
</div>
{% else %}
<div class="modal-body">
    <h3>Для початку увійдіть!</h3>
</div>
{% endif %}
</div>
</div>
</div>
<br><br><br>
<!-- cards -->
<div class="container">
    <div class="container mx-auto mt-2">
        <div class="row">

            {% for i in quiz %}
            <div class="col-md-4">
                <div class="card" style="width: 21rem;">
                    <div class="card-body">
                        <h5 class="card-title" style="text-align:
center;">{{i.name}}</h5>
                        <br>
                        <p class="card-text">{{i.desc}}</p>

```

```

        <br>
        <button data-bs-target="#exampleModal" data-bs-
toggle="modal" data-time="{{i.time}}"
            data-questions="{{i.number_of_questions}}" data-
name="{{i.name}}" data-id="{{i.id}}" id="btn"
            style="justify-content: center;" class="attempt btn
mr-2 modal-button"><i class="fas fa-link"></i>Запустити
            КВіЗ</button>
    </div>
</div>
</div>
</div>
{% endfor %}
</div>
</div>
{% endblock %}
{% block js %}
<script>
    const modalBtns = [...document.getElementsByClassName('modal-
button')]
    const modalBody = document.getElementById('modal-body')
    const modalTitle = document.getElementById('modal-title')
    const startQuiz = document.getElementById('start-quiz')

    const url = window.location.href

    modalBtns.forEach(modalBtn =>
modalBtn.addEventListener('click', () => {
        const id = modalBtn.getAttribute('data-id')

```

```

const name = modalBtn.getAttribute('data-name')
const numberOfQuestions = modalBtn.getAttribute('data-
questions')
const time = modalBtn.getAttribute('data-time')

modaltitle.innerHTML = `${name}`

modalBody.innerHTML = `
    ${name} складається з запитань ${numberOfQuestions} і, отже, має
    ${numberOfQuestions} позначок. Ви отримаєте рівно ${time} секунд,
    щоб заповнити ${numberOfQuestions} запитань із кількома варіантами.
    Після завершення запитань ви можете натиснути на
        кнопка "Надіслати", щоб отримати свій бал і зрозуміти
    концепцію, в якій ви зробили помилку.<br>
        Всього найкращого у вікторині!!!.
    `

startQuiz.addEventListener('click', () => {
    window.location.href = url + id
});
});
</script>
{% endblock %}

```

add_quiz.html

```

{% extends 'base.html' %}
{% block title %} Створити Квіз {% endblock %}

{% block body %}

```

```

<div class="container mt-4">
    <form action="/add_quiz/" method="POST"> {% csrf_token %}
    {{ form.as_p }}
    <button type="submit" class="btn btn-info"
href="/">Підтвердити</button>
    </form>
    <br>
    <table class="table">
        <thead>
            <tr>
                <th>Квіз</th>
                <th>Опис</th>
                <th>Видалити</th>
            </tr>
        </thead>
        {% for quiz in quiz %}
            <tr>
                <td><a
href="/add_quiz_options/{{quiz.id}}/">{{quiz.name}}</a></td>
                <td>{{quiz.desc}}</td>
                <td><a href="/delete_quiz/{{quiz.id}}/" class="btn
btn-danger">Видалити</a></td>
            </tr>
        {% endfor %}
    </table>
</div>
{% endblock %}

```


ДОДАТОК В – ПРОГРАМА ТА МЕТОДИКА ВИПРОБУВАНЬ

1 Об'єкт випробувань

Об'єктом випробувань є вебсайт з системою Quiz.

Областю застосування програмного є забезпечення створення освітніх та розважальних вікторин.

2 Мета випробувань

Перевірка придатності програмного забезпечення до використання за призначенням, її відповідність заданим вимогам і документації. Перевірка працездатності та правильності роботи програмного забезпечення.

3 Вимоги до програми

При проведенні випробувань, функціональні характеристики програми підлягають перевірці на відповідність вимогам, викладеним у пункті «Вимоги до складу виконуваних функцій» Технічного завдання.

4 Вимоги до програмної документації

В комплект програмної документації повинні входити наступні документи:

- технічне завдання;
- інструкція користувача;
- програма і методика випробувань;
- текст програми.

5 Склад та порядок випробувань

Вимоги до складу та параметрів технічних і програмних засобів вказані в Додатку А.

Порядок проведення випробувань:

- виконуються контрольні тести в довільному порядку;
- робиться аналіз отриманих результатів і встановлюється відповідність програмного продукту вимогам і системним документам.

При виявленні помилок у роботі системи складається їх перелік і обговорюється термін їх виправлення розробником. Після цього замовник проводить повторне тестування в повному обсязі (можливе використання нових чи додаткових тестів).

6 Методи випробувань

Методом випробування розробленого програмного забезпечення є тестування чорної скриньки, яке представляє процес виконання програми з метою виявлення помилок. Кроки процесу задаються тестами.

Розглянемо тести для вебсайту з квіз тематикою.

- а) варіанта використання «Створити Квіз»;
- б) варіанта використання «Авторизація»;
- в) тестування циклу від створення квізу до проходження користувачем.

ДОДАТОК Г – ОПИС ПРОГРАМИ

1 Загальні відомості

1.1 Позначення і найменування програми

Найменування: Розробка вебсайту «Quiz».

Позначення: Розваги та освіта.

1.2 Програмне забезпечення, необхідне для функціонування програми

1.2.1 Backend.

В склад технічних засобів повинен входити сервер з наступними параметрами:

- операційна система: Linux;
- процесор: Intel Celeron G4900, 3.1 GHZ, 2 ядра та більше;
- інтернет: 30 Мбит/с та більше.
- оперативна пам'ять: мінімум 4Гб.
- твердий диск: для нормальної роботи системі потрібно 50Мб вільного дискового простору.

2.2.2. Frontend.

В склад технічних засобів повинен входити комп'ютер, або смартфон з мінімальними характеристиками, який підтримує встановлення програмного додатка Google Chrome.

Android:

- версія ОС: Android 10.0 Marshmallow або пізнішої версії.
- інтернет: потрібно;
- потрібно вільного місця: 1 Mb. IOS:
- версія ОС: iOS 14.0 +;
- інтернет: потрібно;
- потрібно вільного місця: 162,6 Mb.

Комп'ютер:

- ОС: Windows XP, 7, Vista, 8, 8.1, 10;
- процесор: Intel Pentium 4/Athlon 64 або пізнішої версії з підтримкою SSE2
- оперативна пам'ять: 512 MB ОЗУ;
- відеокарта: Intel HD Graphics;
- місце на диску: 350 MB.
- інтернет: потрібно;

1.3 Мови програмування

Вихідні коди програми повинні бути реалізовані на мові Python3 та Django. В якості середовища розробки програм повинна бути використана PyCharm. В якості бази даних використовувати СУБД PostgreSQL у середовищі DataGrip.

2 Функціональне призначення

2.1 Класи розв'язуваних завдань та призначення програми Функціональним призначенням програми – за допомогою гри у вікторину, розвиток пам'яті, знання стресу, розвиток уяви та творчості при створенні своїх вікторин, тестів, опитувань.

2.2 Функціональні обмеження

Для роботи з інтернет браузером та вебсайтом необхідно мати підключення до мережі інтернет.

3 Опис логічної структури

3.1 Структура програми

Програмне забезпечення складається з наступних частин. Графічне зображення всіх компонентів можна побачити на рисунку Г.1:

- Вебсервер, на якому запущений скрипт програми. Також на сервері знаходиться інтерфейс бази даних;
- Сервер бази даних;
- Пристрій користувача, із встановленим веббраузером та підключенням до мережі інтернет.

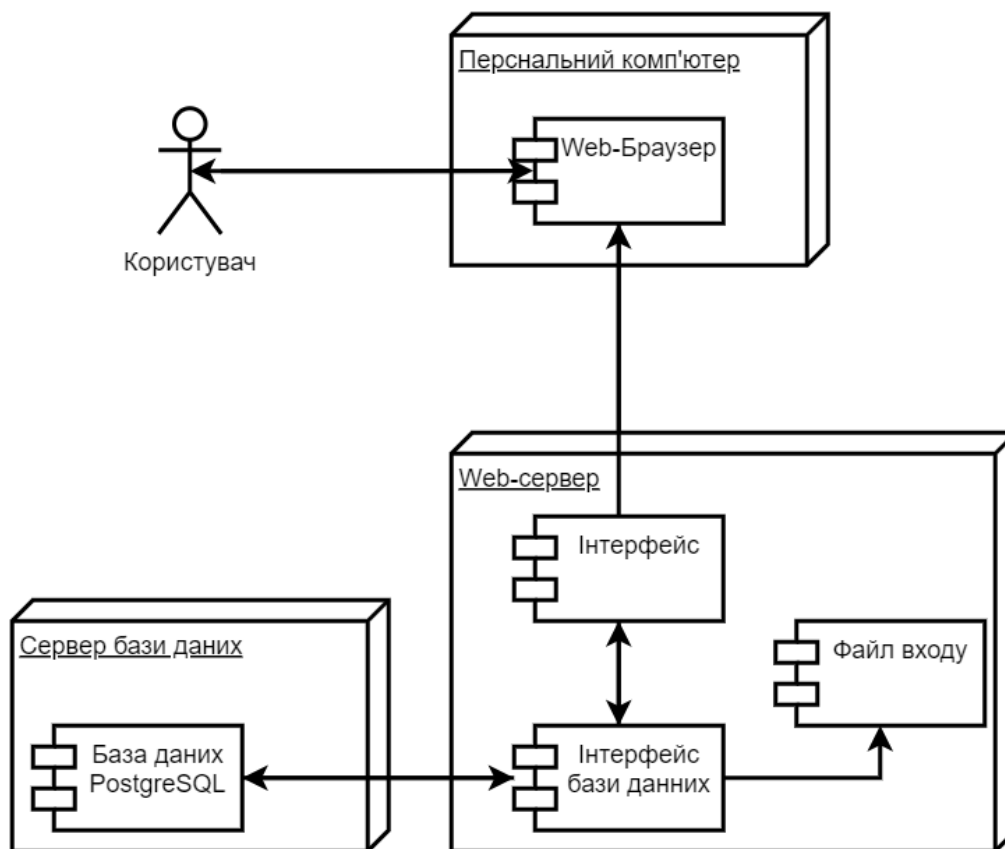


Рисунок Г.1 – Схема взаємодії

3.2 Алгоритм програми

Робота програмного забезпечення представлена взаємодією між клієнтською та серверною частинами, а також базою даних. Алгоритм роботи програми є

наступним:

- 1) Користувач взаємодіє з вебсайтом через веббраузер;
- 2) Веббраузер через Інтернет відсилає запити вебсерверу;
- 3) Сервер обробляє запит;
- 4) Скрипт звертається до бази даних (якщо це потрібно) та відсилає результат;
- 5) Веббраузер відображає отриманий результат у вигляді сторінок.

ДОДАТОК Д – ІНСТРУКЦІЯ КОРИСТУВАЧА

При першому запуску вебсайту ми потрапляємо на головну сторінку зі списком квізів, для взаємодії з основними функціями сайту необхідно зареєструватися. Для цього ми натискаємо на кнопку «Реєстрація», переходимо на сторінку реєстрації та заповнюємо відповідні поля. Зображено на рисунку Д.1 та рисунку Д.2.

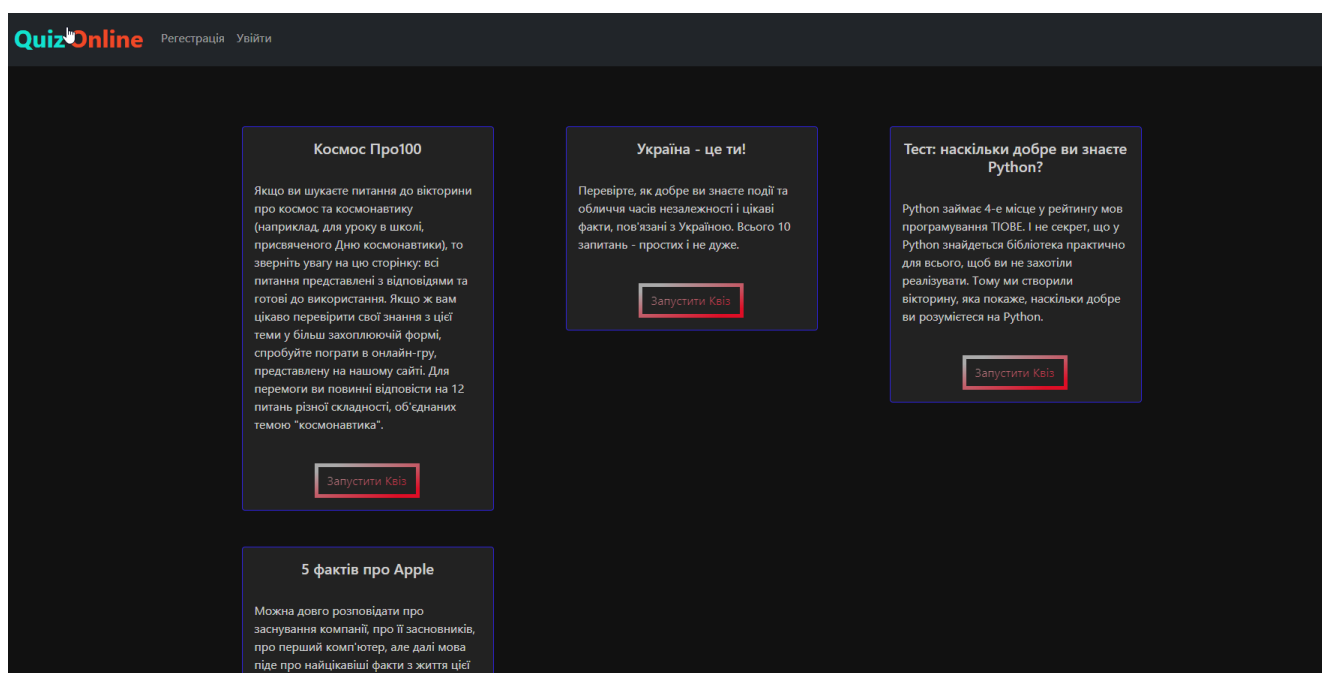


Рисунок Д.1 – Головна сторінка

Quiz Online [Регистрація](#) [Увійти](#)

ДЕТАЛІ РЕГІСТРАЦІЇ

Ім'я

Tester

Прізвище

Tester

Логін

Tester

Пошта

Tester@gmail.com

Пароль

.....

Повторіть Пароль

.....

Зареєструватись

Рисунок Д.2 – Реєстрація

Після успішної реєстрації, щоб отримати повний набір функцій вебсайту, потрібно увійти до системи під своїм логіном та паролем. Зображено на рисунку Д.3.

Quiz Online Регистрація Увійти

ДЕТАЛІ ВХОДУ

Логін

Tester

Пароль

.....

Увійти

Рисунок Д.3 – Авторизація як «Гравець»

Для того, щоб створити новий квіз, потрібно натиснути кнопку «Створити» і з меню вибрати «Створити Квіз». Після цього ми потрапляємо на сторінку зі створенням квізів, де вводимо дані нового квізу (назва, опис, кількість питань, час виконання), натискаємо кнопку «Підтвердити», підтверджуємо створення нового квізу. Зображено на рисунку Д.4.

Quiz Online Вітаю Tester Створити Вийти

Назва: Тестовий Квіз

Опис: Lorem ipsum dolor sit am

Кількість питань: 5

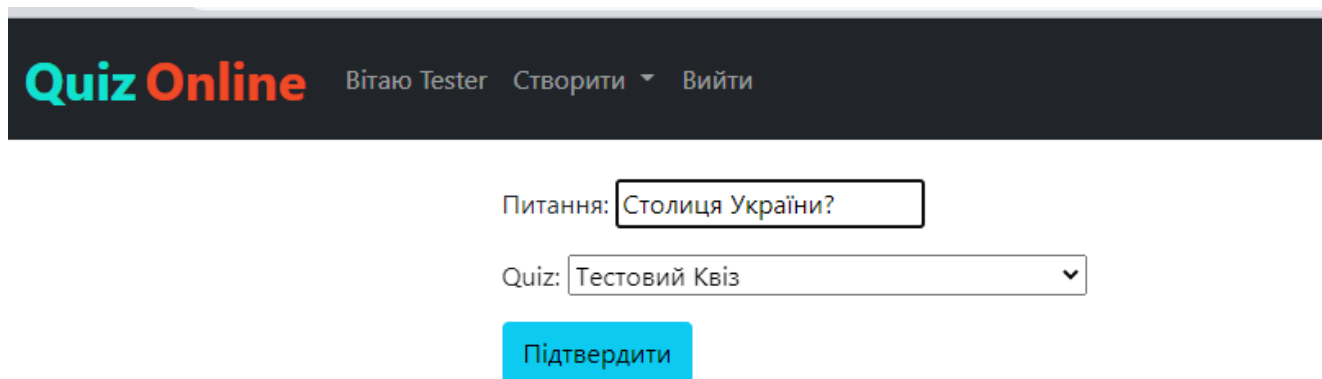
Тривалість: 240 Тривалість Квізу в секундах

Підтвердити

Рисунок Д.4 – Створення квізу

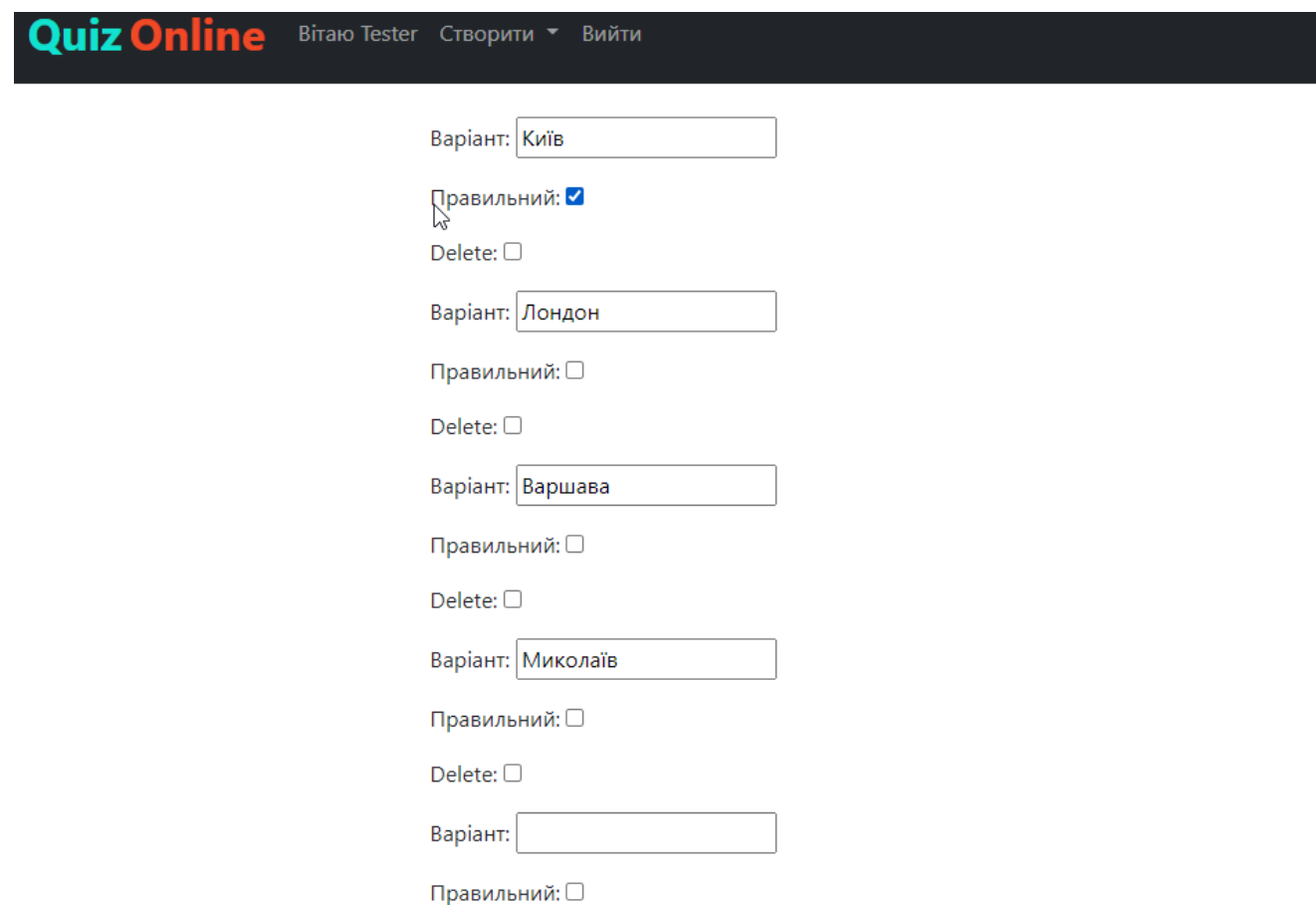
Наступний крок, натискаємо "Створити", вибираємо "Створити питання",

переходимо на сторінку створення питань. Вводимо питання та обираємо до якого квізу він належить. Підтверджуємо. Вибираємо питання зі списку та відкриваємо його, додаємо 4 варіанти відповіді та вибираємо, який з них буде правильним, натискаємо «Підтвердити». Зображено на рисунках Д.5, Д.6.



The screenshot shows the 'Quiz Online' header with navigation links: 'Вітаю Tester', 'Створити' (with a dropdown arrow), and 'Вийти'. Below the header, there is a form for creating a question. It includes a label 'Питання:' followed by a text input field containing 'Столиця України?'. Below this is a label 'Quiz:' followed by a dropdown menu showing 'Тестовий Квіз'. At the bottom of the form is a blue button labeled 'Підтвердити'.

Рисунок Д.5 – Створення питання



The screenshot shows the 'Quiz Online' header with navigation links: 'Вітаю Tester', 'Створити' (with a dropdown arrow), and 'Вийти'. Below the header, there is a list of options for a question. Each option consists of a label 'Варіант:' followed by a text input field, a checkbox labeled 'Правильний:', and a checkbox labeled 'Delete:'. The first option has 'Київ' in the input field, the 'Правильний:' checkbox is checked, and the 'Delete:' checkbox is unchecked. The second option has 'Лондон' in the input field, and both checkboxes are unchecked. The third option has 'Варшава' in the input field, and both checkboxes are unchecked. The fourth option has 'Миколаїв' in the input field, and both checkboxes are unchecked. The fifth option has an empty input field, and both checkboxes are unchecked.

Рисунок Д.6 – Додавання питання

Переходимо на головну сторінку, натиснувши на назву сайту, вибираємо квіз який ми створили та натискаємо «Запустити Квіз». Читаємо правила квізу та натискаємо «Почати Квіз!». На сторінці з квізом вибираємо правильну відповідь та натискаємо «Підтвердити». Зображено на рисунках Д.7, Д.8, Д.9.

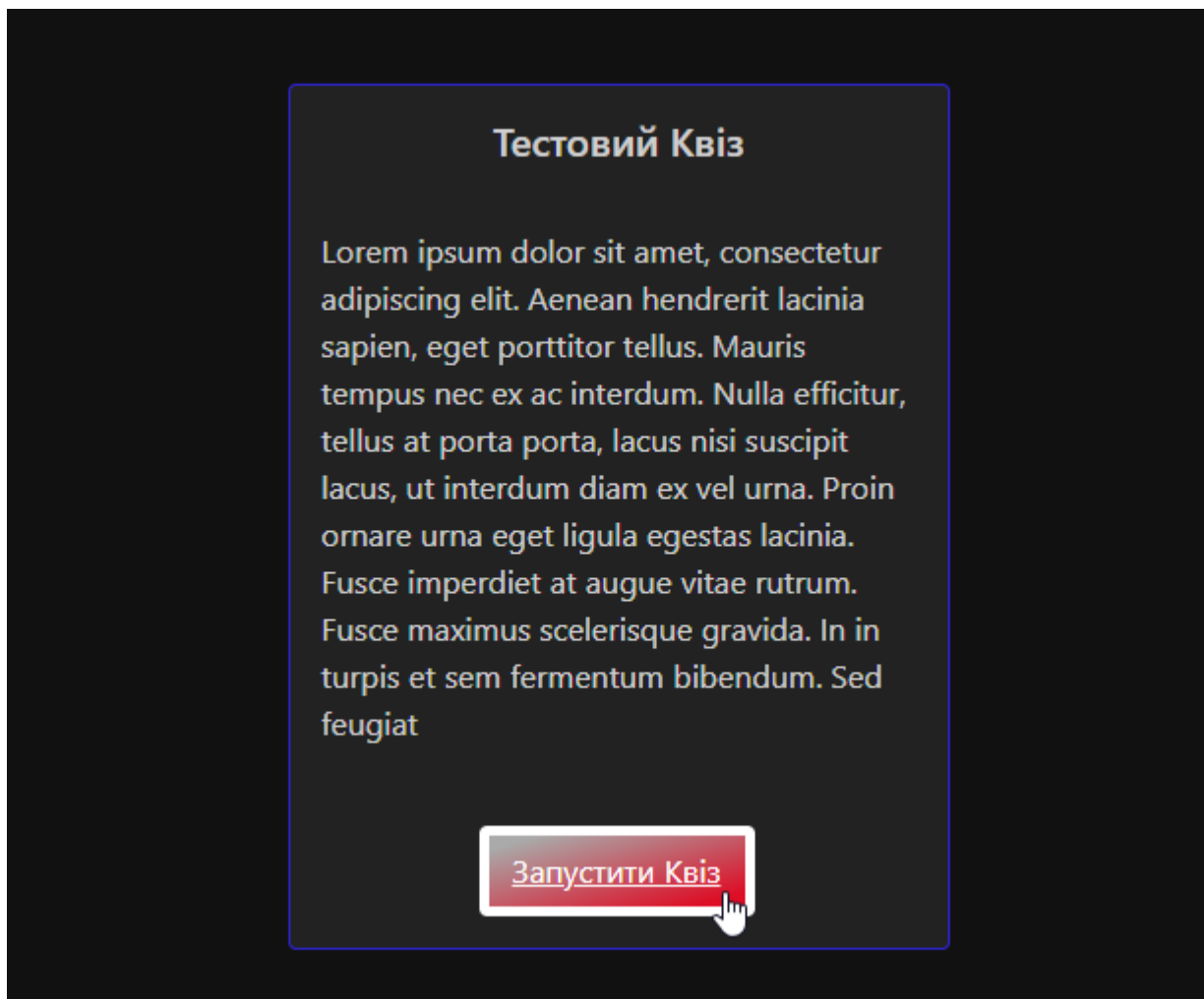


Рисунок Д.7 – Запуск квізу

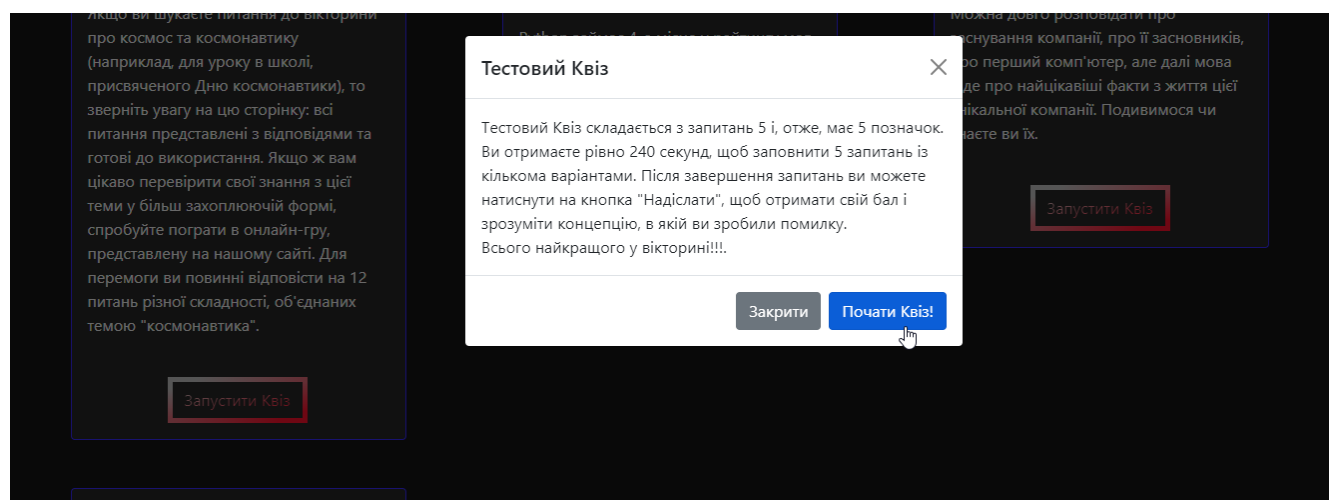


Рисунок Д.8 – Правила квізу

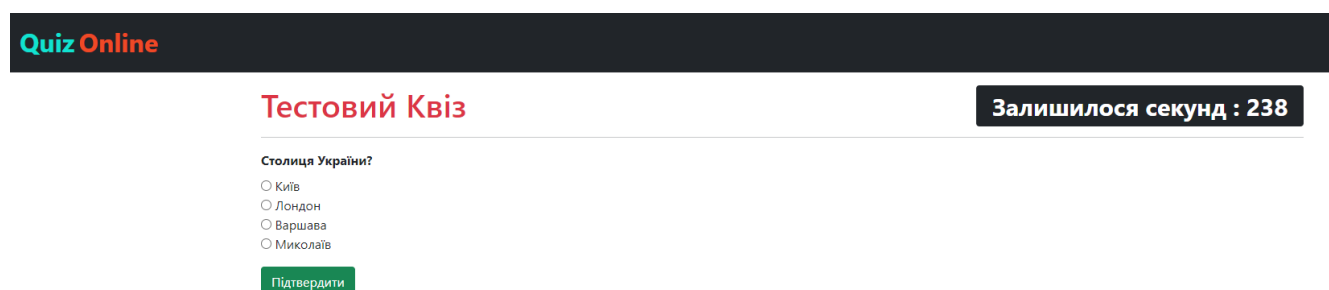


Рисунок Д.9 – Проходження квізу