Quizzo Project Report

By: Prajakta Nitin Chinavalkar
Email: 21f3001858@ds.study.iitm.ac.in

---

1. Introduction

Quizzo is an interactive web-based quiz application designed to provide users with an engaging and structured quiz-taking experience. The system allows users to attempt quizzes, track their scores, and improve their knowledge across various subjects. The project is built using Flask, SQLite, and Bootstrap for the frontend.

---

2. Objectives

- Develop an intuitive platform for quiz-based learning.
- Track user performance over time.
- Provide a structured database for managing quizzes, questions, and scores.

---

3. System Architecture

Technologies Used:

- **Backend:** Flask (Python)
- **Database:** SQLite
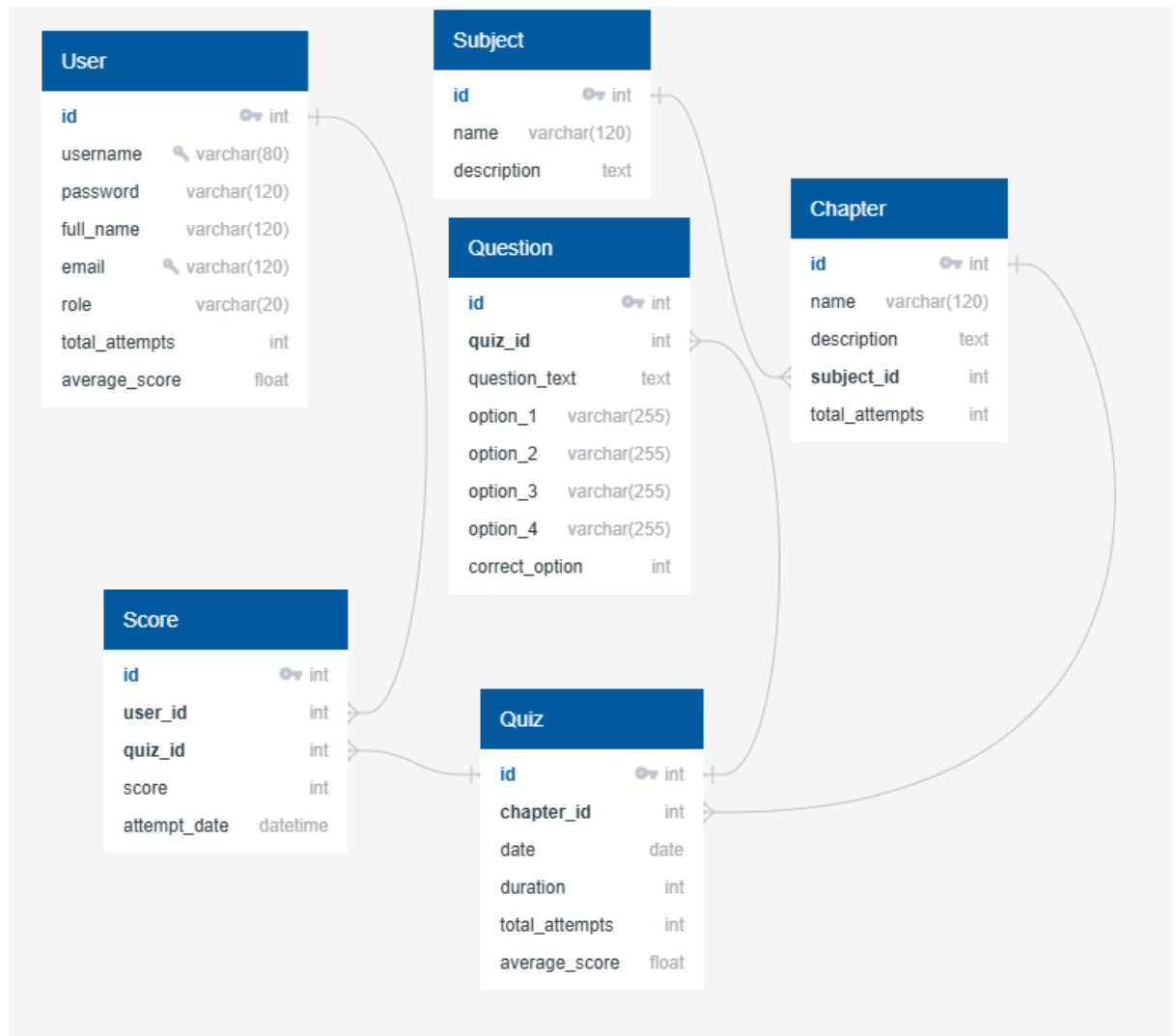- **Frontend:** HTML, Bootstrap, Jinja2 Templates
- **ORM:** SQLAlchemy

System Components:

- **User Authentication:** Login and role-based access.
- **Quiz Management:** Users can take quizzes, and admins can create and manage quizzes.

● **Performance Tracking:** Stores quiz scores, attempts, and performance trends.

4. Database Design

● **Entity-Relationship (ER) Diagram**

5. Features

User Roles:

● **Admin:** Can create quizzes, manage users, and analyze reports.

- **User:** Can take quizzes and track their performance.

Quiz Features:

- Create, edit, and delete quizzes.
- Each quiz contains multiple-choice questions.
- All quiz attempts are stored.

Performance Tracking:

- Stores total attempts per quiz and chapter.
- Tracks average score and improvements over time.

---

6. Implementation Details

Flask Models:

- **User Model:** Handles authentication and user management.
- **Quiz Model:** Manages quizzes.
- **Question Model:** Stores quiz questions.
- **Score Model:** Tracks user quiz performance.
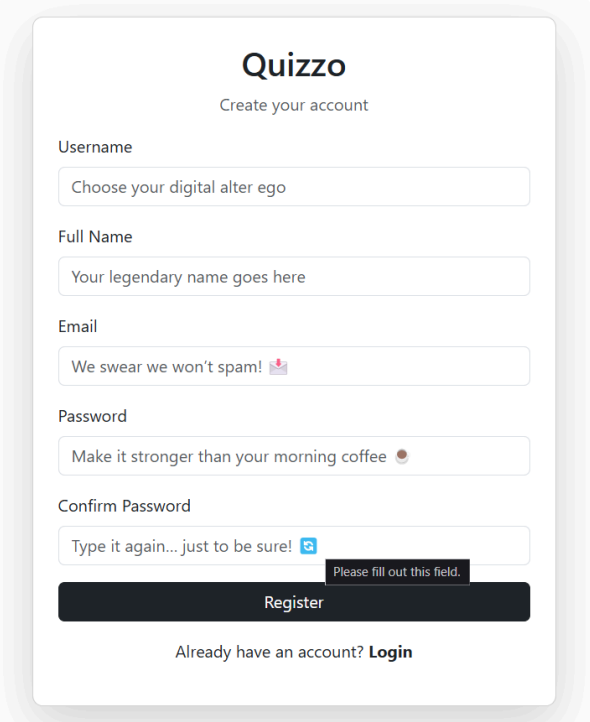
Routes & Templates

1. User Routes

- `/` → Home page (Landing page) → `index.html`
- `/register` → User registration page → `register.html`
- `/login` → User login page → `login.html`
- `/ud`→ User dashboard (Quiz summary, scores, and available quizzes) → `us_dash.html`
- `/quiz/<int:quiz_id>` → Page to attempt a quiz → `us_quiz.html`
- `/quiz/<int:quiz_id>/submit` → Submit quiz answers (backend logic only) → redirect to score page.
- `/score` → View quiz history and scores → `us_scores.html`
- `/user_summary/<int:user_id>` → View summary charts and statistics → `us_summary.html`
- `/logout` → Logout user and redirect to login page (backend logic only)

2. `Admin` Routes

- `/ad` → Admin dashboard → `ad_dash.html`
- `/admin/users` → Manage users (View, search users) → `ad_users.html`
- `/admin/subjects` → Manage subjects (CRUD operations) → `ad_subject.html`
- `/admin/chapters` → Manage chapters (CRUD operations) → `ad_chapter.html`
- `/admin/quizzes` → Manage quizzes (CRUD operations) → `ad_quiz.html`
- `/admin/questions/<int:quiz_id>`→ Manage quiz questions (CRUD operations) → `ad_questions.html`
- `/admin/summary` → View quiz statistics → `ad_summary.html`

---

7. Results & Screenshots

Video Representation -
https://drive.google.com/file/d/1vbvrac8pEBj11BYALYRSZcrnWRy5uzmV/view?usp=sharing

8. Conclusion

The Quizzo project successfully implements a structured quiz-taking system with user authentication and performance tracking. Future enhancements could include:

- **Leaderboard System** – Displaying top performers.
- **Question Bank Expansion** – More categories and dynamic difficulty levels.