Indian Institute of Technology, Madras | BSCS2003P
Kavisha Tankle
23F1000041
23f1000041@ds.study.iitm.ac.in
I am a Data Science student exploring new concepts and applying them in real projects. This web app project, Quizdom, helps me understand web development basics while building something practical.

Project Report: Quizdom

A Multi-User Web App for Exam Preparation
Modern Application Development I Project

DESCRIPTION

Quizdom is a web-based quiz platform designed for both users and administrators. Users can take quizzes, check their scores, and track their progress, while admins manage quizzes, users, and performance data. I built the backend using Flask and stored all user and quiz data in an SQLite database. For the frontend, I used HTML, CSS, Bootstrap, and Jinja2 templates to create a clear and easy-to-navigate interface.

TECHNOLOGIES USED

Backend:

1. **Flask** for handling web requests and application logic.
2. **Flask-SQLAlchemy** provides an **ORM (Object-Relational Mapping)** to efficiently manage SQLite databases, handle data relationships, and simplify queries.
3. **Werkzeug Security** for secure password hashing.

Frontend:

4. **HTML, CSS, Bootstrap** to design a responsive and user-friendly interface.
5. **Jinja2** for rendering dynamic content.

Database Storage, Data Visualization & Analytics::

6. **SQLite** to store user, quiz, and score data.
7. **OS Module** to manage file paths and directories.
8. **Datetime** to track quiz timestamps and user activity.
9. **Matplotlib & Seaborn** to generate quiz performance charts.

DB SCHEMA DESIGN

I structured the database to efficiently handle **users, quizzes, questions, and scores** with proper relationships, efficient retrieval and data integrity.

1. **User Table:** Manages authentication with unique `email` and `username`, hashed passwords, and an `is_active` flag for account status.
2. **Subject & Chapter Tables:** Organize quizzes into structured categories using `subject_id` as a **foreign key** in Chapters.
3. **Quiz Table:** Links quizzes to chapters via `chapter_id`, with `date_of_quiz` for scheduling and `time_duration` for enforcing time limits.
4. **Question Table:** Stores multiple-choice questions with `quiz_id` as a **foreign key**, ensuring proper quiz-question mapping.
5. **Score Table:** Tracks user performance with `user_id`, `quiz_id`, `score`, `date_taken`, and `completed` status for quiz attempts.

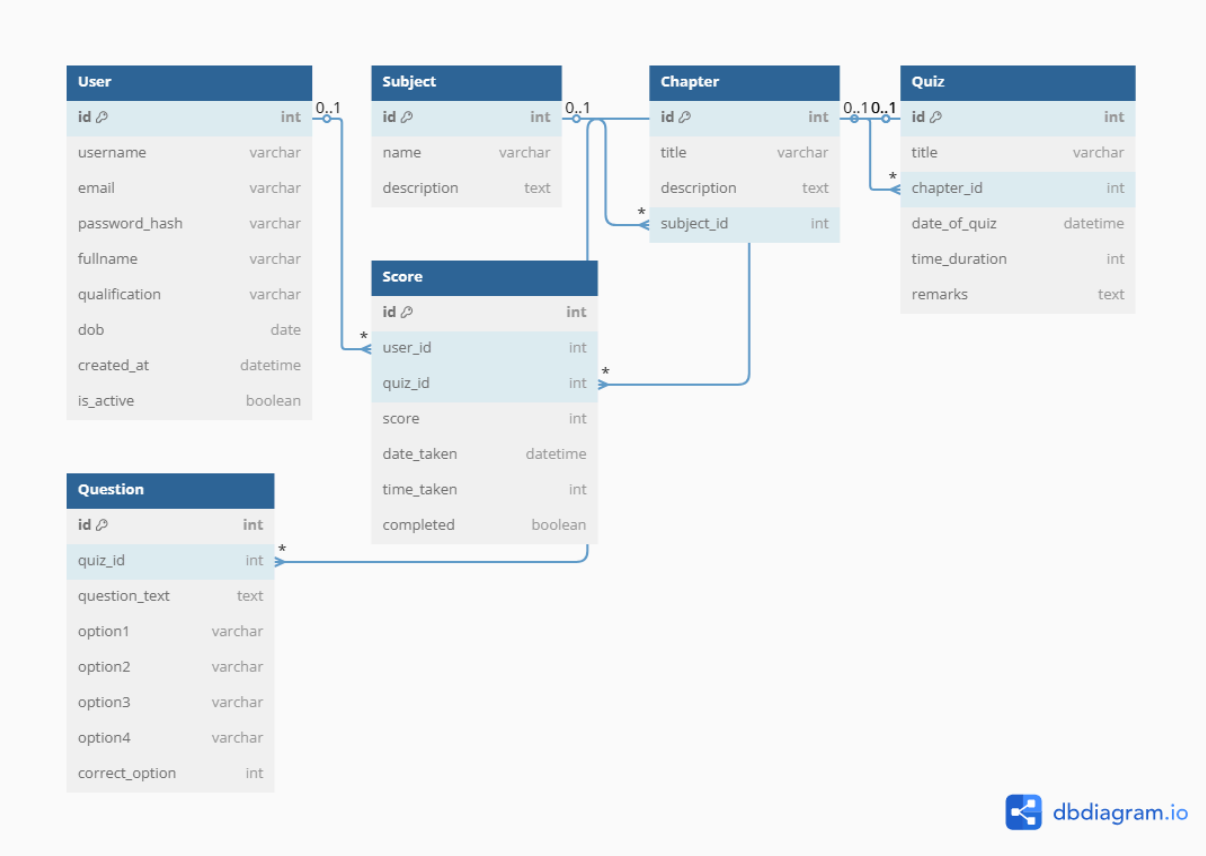

Figure 1: Entity-Relationship (ER) Diagram of Quizdom [Link]

ARCHITECTURE & FEATURES

The app is organized using the **MVC (Model-View-Controller) structure. Flask** manages the routes (controllers) in `main.py`, handling user login, quizzes, and scores. **Jinja2 templates (views)** are stored in the `templates` folder to create web pages using **HTML**, **Bootstrap**, and **CSS**. **SQLAlchemy (models)** connects to the database for storing user, quiz and score data. Static files like **CSS**, **JavaScript** and **generated chart images** are kept in the `static` folder while the `instance` folder holds the **SQLite database**. The app is being served at http://127.0.0.1:5000.

Features Implemented:

1. **User Authentication:** Users can register, log in, and have secure password storage.
2. **Admin Login:** A separate **admin login page** allows the admin to access management features. An admin account is **automatically created** when the app starts.
3. **Quiz Management:** Admin users can create, edit, and delete quizzes, questions, and subjects.
4. **Quiz Attempting & Scoring:** Users can take quizzes, and scores are automatically calculated.
5. **Performance Tracking:** User scores and quiz history are stored, with **graphs generated for analysis.**
6. **Admin Controls:** The admin can manage quizzes and users through dedicated routes.

Flask's built-in features like **session management** and **form validation** help handle user logins and quiz submissions, while **custom features** enable quiz tracking, score storage, and data visualization.

Video Presentation

A full walkthrough of the app, demonstration of its features and navigation by me.