Project Report - Quizzy Quest

Author

Raghvi Gupta | 23f3001927@ds.study.iitm.ac.in

IITM BS in Data Science and Applications Diploma Level Student

Introduction & Description

Quizzy Quest is a web-based quiz platform with two main roles: **User** and **Admin**. Users can browse, attempt, and track quizzes, while Admins manage quizzes, questions, and analytics. The goal is a seamless quiz experience for users and efficient management for admins.

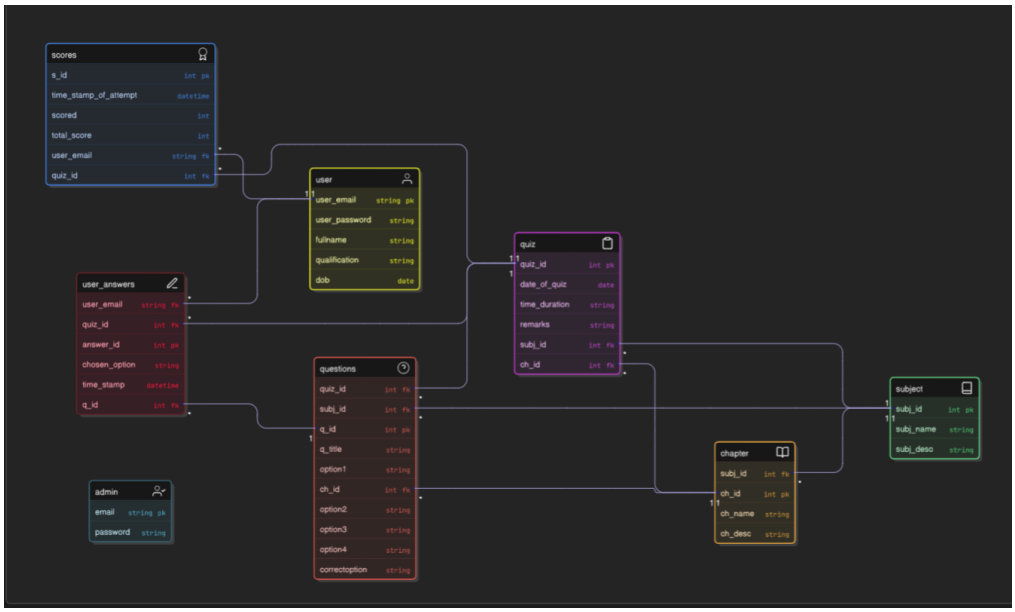
Technologies Used

- **Backend:** Flask, Flask-SQLAlchemy, SQLite, DateTime
- **Frontend:** HTML, CSS, JavaScript, Bootstrap, Chart.js
- **Templates:** Jinja2

Database Schema

The database efficiently manages admins, users, subjects, quizzes, questions, and scores while ensuring data integrity through normalization and foreign keys.

- **Admin Table:** Stores email (Primary Key) and hashed password.
- **User Table:** Stores user email (Primary Key), password, name, qualification, and DOB.
- **Subject Table:** Stores subject ID (Primary Key), name, and description.
- **Chapter Table:** Links chapters to subjects.
- **Quiz Table:** Links quizzes to subjects and chapters, with time limits and schedule.
- **Questions Table:** Stores questions, options, and correct answers, linked to quizzes.
- **Scores Table:** Tracks user performance with timestamps and marks.
- **UserAnswers Table:** Logs users' responses for analytics.



API Design

Admin Endpoints

- /admin/dashboard – View subjects, chapters, and questions per chapter.
- /admin/add_quiz – Create quizzes; /admin/add_question/<quiz_id> – Add questions.
- /admin/edit_question/<quiz_id>/<question_id> – Edit;
  /admin/delete_question/<quiz_id>/<question_id> – Delete.
- /admin/summary – View top scores, attempts per quiz.

User Endpoints

- /user/register, /login, /logout – Authentication.
- /user/quiz-list – View quizzes; /user/quiz/<quiz_id>/question/<question_id> – Attempt quiz.
- /user/quiz/<quiz_id>/submit – Submit answers; /user/quiz-result/<quiz_id> – View results.
- /user/scores, /user/summary – Track quiz history.

Common Endpoints

- / – Landing page; /admin/admin_base – Admin dashboard template.

Architecture & Features

- **MVC Pattern:** Routes in app.py, database models in models.py, templates in templates/, static assets in static/.
- **Key Features:** User authentication, quiz management, automated scoring, leaderboards, and real-time progress tracking.
- **Additional Features:** Search functionality, analytics, and time tracking.

Implementation Details

- **Flask & Flask-SQLAlchemy** handle backend and database operations.
- **Session Handling** ensures authentication.
- **Form Handling & Flash Messages** enhance user experience.

Conclusion

Quizzy Quest is a structured, feature-rich quiz platform optimized for scalability and usability.

Video

https://drive.google.com/file/d/1lA_ZxEwxC8qkSR0uoUIrzeLcU8XCfa0K/view?usp=sharing