IITM Modern Application Development-1

QUIZ Master APP

Author

Akshay KS
24f1001761
24f1001761@ds.study.iitm.ac.in

Description

Quiz Master - V1 is a multi-user exam preparation platform built using Flask. The application includes an admin panel for managing users, subjects, quizzes, and questions. It allows users to register, log in, take quizzes, and view scores. The system also includes search functionality and summary charts for better visualisation.

Technologies Used

1. **Flask** – A lightweight web framework that provides flexibility for backend development. It allows for easy routing, session management, and API creation, making it suitable for building the Quiz Master application.

2. **SQLite** – A lightweight, serverless database that simplifies data storage. It is ideal for this project as it does not require additional configurations and can efficiently store user, quiz, and result data.

3. **Jinja2** – A templating engine that enables dynamic content rendering in HTML pages. It simplifies passing data from Flask to the frontend, allowing the creation of user-specific dashboards and quiz interfaces.

4. **Bootstrap** – A responsive CSS framework that ensures a visually appealing and mobile-friendly UI. It simplifies styling and layout design, improving user experience without requiring extensive custom CSS.

5. **Chart.js** – A JavaScript library used to create visual representations of quiz statistics and scores. It helps in displaying performance insights through interactive charts.

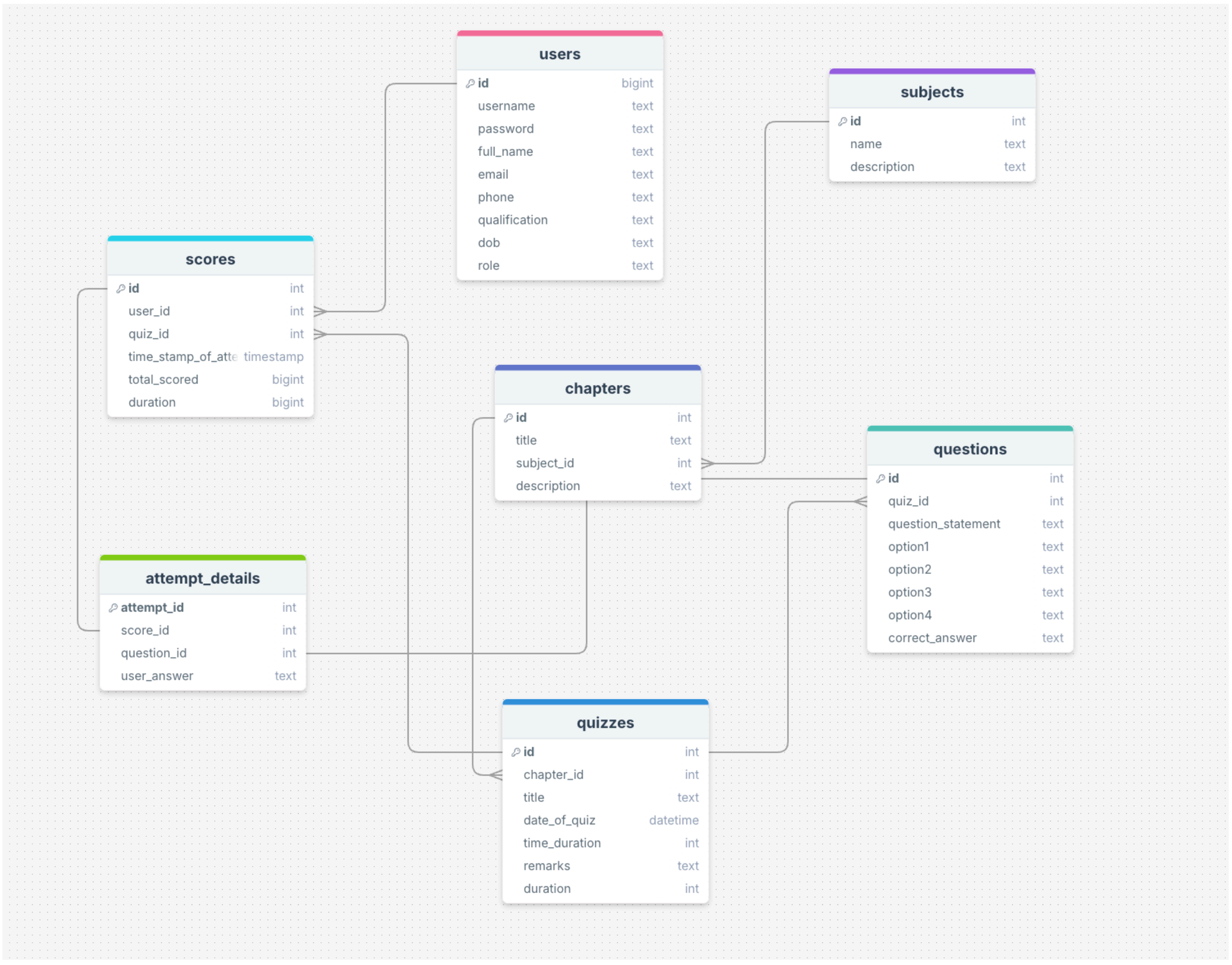
6. **JavaScript** – Enhances interactivity, form validation, and UI responsiveness. It is used for features like login validation, handling UI interactions, and integrating client-side logic with Flask routes.

Purpose of These Technologies

- **Flask** was chosen due to its simplicity and flexibility, allowing easy integration with other technologies while keeping the backend lightweight.
- **SQLite** was selected for its ease of use, requiring no additional setup while still providing powerful querying capabilities.
- **Jinja2** was used to dynamically generate HTML content, enabling seamless data integration into web pages.

- **Bootstrap** ensures a modern, mobile-friendly design, reducing the effort needed for UI development.
- **Chart.js** provides interactive visualisations to make quiz results more insightful for users.
- **JavaScript** was incorporated to enhance the user experience with interactive elements and client-side validation.

DB Schema Design



Users Table: Stores user details, including role (admin/user). This allows role-based access control.
Subjects & Chapters Tables: Subjects contain multiple chapters, forming a hierarchical structure (subject_id as a foreign key in chapters).
Quizzes Table: Each quiz is linked to a specific chapter (chapter_id as a foreign key). It includes details like date_of_quiz and duration for scheduling.
Questions Table: Stores MCQs linked to a quiz (quiz_id as a foreign key), ensuring each quiz has multiple questions.
Scores Table: Captures quiz attempts by users, storing total_scored and duration for performance tracking.
Attempt Details Table: Records user responses per question, linking score_id and question_id to track correctness and review attempts.

API Design

Authentication APIs: register, login (POST)
User APIs: profile, dashboard, attempt_quiz, view_report, leaderboard, edit_profile (GET, POST, EDIT)
Admin APIs:

- manage_subjects, manage_chapters, manage_quizzes, manage_questions (GET, POST, EDIT, DELETE)
- search_users, search_subjects, search_quizzes, search_chapters (GET)
- admin_dashboard, admin_profile (GET)

Admin Privileges: Can **add, edit, and delete** all entities in the system.

Implementation:

- Routes are defined using Flask and handled in app.py.
- API responses are returned in JSON format.
- Form validation ensures correct input.
-

Architecture and Features

- **Project Organization:**
  - app.py: Main Flask application logic.
  - init_db.py: Initialise the database
  - quiz_master.db : SQLite database file
  - templates/: Contains Jinja2 HTML templates.
  - static/: Holds CSS, JavaScript, and image assets.
  - Schema/database.sql: Schema of the database tables.
- **Implemented Features:**
  - User Registration & Login
  - Role-based Access Control (Admin/User)
  - Subject, Chapter, and Quiz Management (Admin)
  - Multiple-choice Quiz Attempt (Users)
  - Score Calculation and Leaderboard
  - Search Functionality for Users, Subjects, and Quizzes
  - Summary Charts for Quiz Performance

Default Features: Basic user authentication, quiz management, admin features and attempt tracking .

Additional Features: Admin dashboard, analytics, and responsive UI enhancements ,leaderboard of all users.

Video Link : https://drive.google.com/file/d/12DB29uSWpq794bnoWjzAtqc0BMc_5ptA/view?usp=drive_link