

Author

SANTHOSH A S

24f2001811

24f2001811@ds.study.iitm.ac.in

Santhosh, a Computer Science student at Amrita Vishwa Vidyapeetham and a Data Science learner at IIT Madras, is passionate about web development and creating impactful software Solutions.

Description

This Flask-based quiz app has an admin managing subjects, quizzes, and users, while users register, attempt quizzes, and track scores. SQLite is used for storage, with Jinja2 and Bootstrap for UI.

Technologies used

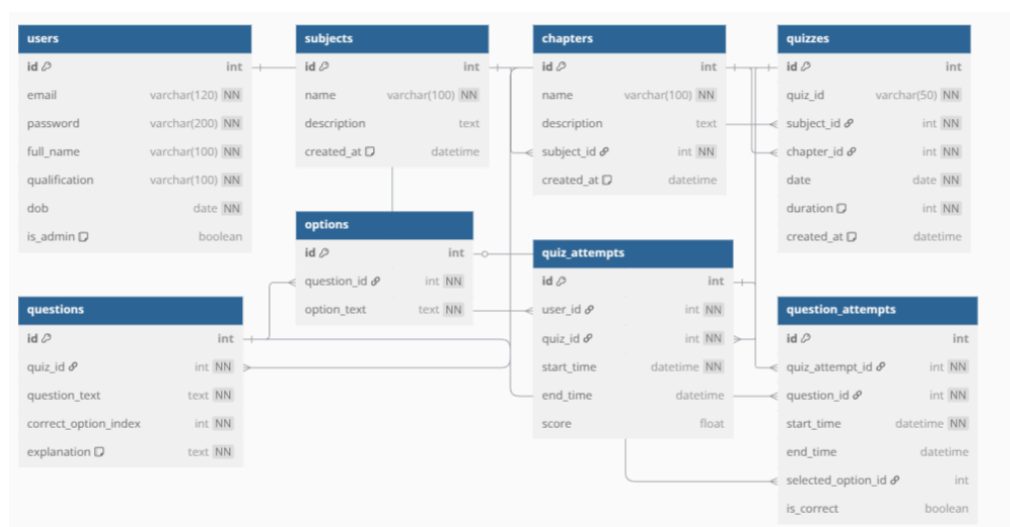
Backend (Flask and Extensions)

1. **Flask** – Web framework for handling routes, requests, and rendering templates.
2. **Flask-SQLAlchemy** – ORM for interacting with the SQLite database.
3. **Werkzeug Security** – Provides password hashing and authentication.
4. **Functools (wraps)** – Used to create decorators for access control.
5. **Datetime** – Handles date and time functionalities.
6. **SQLAlchemy Functions (func)** – Used for database queries, such as aggregation functions.

Frontend (HTML, CSS, JS, and Libraries)

1. **Jinja2** – Flask's templating engine to dynamically render HTML pages.
2. **Bootstrap 5** – CSS framework for responsive design and UI components.
3. **Tailwind CSS** – Utility-first CSS framework for modern styling.
4. **FontAwesome** – Provides icons for UI enhancement.

DB Schema Design



API Design

Authentication & User Management

Endpoints: /login, /register, /logout, /profile, /profile/edit, /users, /admin/user/add, /admin/user/<int:user_id>/edit, /admin/user/<int:user_id>/delete

Implementation: Supports user authentication, profile management, and admin user controls.

Dashboard & Summary

Endpoints: /dashboard, /admin/dashboard, /user/dashboard, /summary, /user/summary, /admin/summary

Implementation: Provides role-based dashboards and quiz performance summaries.

Subjects & Chapters Management (Admin)

Endpoints: /admin/subjects, /admin/subject/add, /admin/subject/<int:subject_id>/edit, /admin/subject/<int:subject_id>/delete, /admin/subject/<int:subject_id>/chapters, /admin/subject/<int:subject_id>/chapter/add, /admin/chapter/<int:chapter_id>/edit, /admin/chapter/<int:chapter_id>/delete

Implementation: Admins can create, modify, and delete subjects and chapters.

Quiz Management (Admin & User)

Endpoints: /admin_quiz, /admin/quizzes, /admin/quiz/<int:quiz_id>/view, /admin/quiz/<int:quiz_id>, /admin/quiz/<int:quiz_id>/delete, /quiz/<int:quiz_id>/start, /quiz/<int:quiz_id>, /quiz/result/<int:attempt_id>

Implementation: Enables quiz creation, editing, deletion, and results viewing.

Quiz API (Question, Answer, Submission, Analysis)

Endpoints: /api/subject/<int:subject_id>/quizzes, /api/chapter/<int:chapter_id>/quizzes, /api/quiz/<int:quiz_id>/questions, /api/quiz/<int:quiz_id>/state, /api/quiz/<int:quiz_id>/question/<int:question_id>/answer, /api/quiz/<int:quiz_id>/submit, /api/quiz/<int:quiz_id>/attempts, /api/quiz/<int:quiz_id>/time-analysis

Implementation: Handles quiz interactions, submissions, and analytics.

User Performance & Statistics

Endpoints: /api/user/statistics, /api/user/performance-comparison

Implementation: Provides user performance insights and comparisons.

Architecture and Features

Architecture

The project follows a well-organized structure for maintainability and scalability. All HTML files are housed in the **templates** directory, with three subfolders: **admin** for admin-related pages, **user** for user-specific pages, and **login** for authentication pages like login and registration. Static assets, including CSS files, are stored in the **static** folder for better separation of concerns. The backend is managed by **app.py**, ensuring smooth functionality, while user-related data is securely stored in **user.db** for efficient data handling.

Features

Admin Login, User Registration and Login, Admin Dashboard, User Dashboard, Subject Management (Create/Edit/Delete), Chapter Management (Create/Edit/Delete), Quiz Management (Create/Edit/Delete), Question Management (Create/Edit/Delete), Quiz Attempt by Users, Quiz Timer (Optional), Score Recording, User Quiz Attempt History, Summary Charts for Admin, Summary Charts for Users, Admin Search for Users/Subjects/Quizzes, Database Initialization with Predefined Admin, Programmatic Database Creation (No Manual DB Creation), Frontend with Jinja2, HTML, CSS, Bootstrap, Flask for Backend, SQLite as Database, Basic Authentication (No Security Concern), MCQ-Based Questions (Single Correct Option), Quiz Date and Duration Specification.

Video

<https://drive.google.com/file/d/1ZoKeF6ZwergPOKpoQSzVJSGfbO2QHcEc/view?usp=sharing>