

Author

Name : Astitva Agarwal

Roll_no : 23f2004131

S_Email_id : 23f2004131@ds.study.iitm.ac.in

Greetings everyone, I am Astitva Agarwal, currently enrolled in the BS in Data Science and Programming program at IIT Madras. As part of the Modern Application Development II Project, I have designed and developed Quiz Master – V2, a full-stack quiz platform that enables efficient quiz management, secure access control, and automated performance reporting.

Project Description

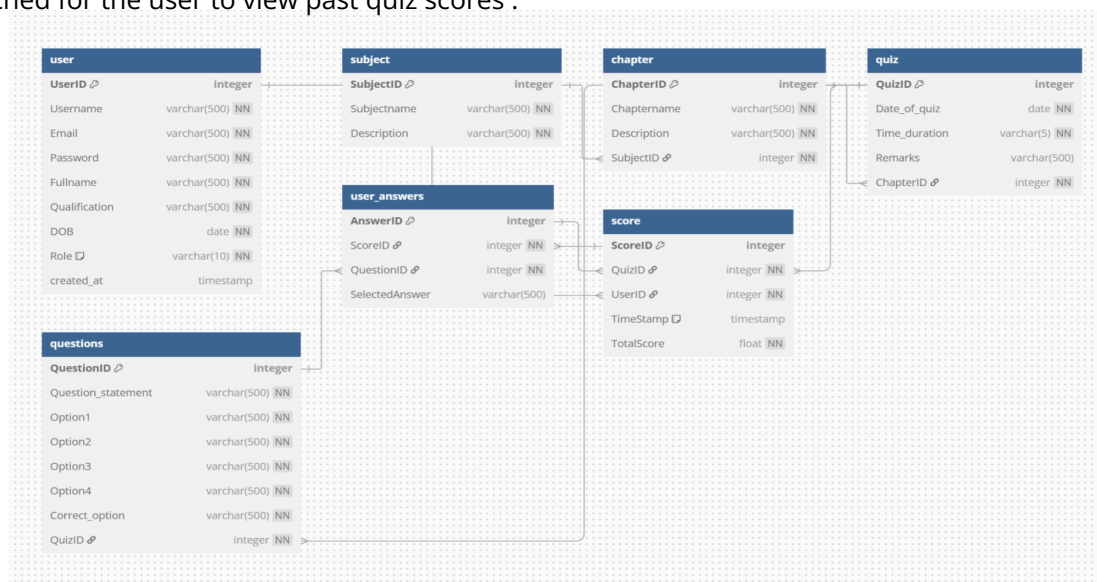
Quiz Master V2 is a multi-user, full-stack web application where admins (Quiz Masters) can manage quizzes and users can register, attempt quizzes, and view results. The application introduces advanced scheduling features, user analytics, email notifications, and frontend/backend separation.

Technologies Used

- **Backend:** Flask (Python)
- **Database:** SQLite (from flask_sqlalchemy :- SQLAlchemy)
- **Frontend:** Vue.js, HTML, CSS, Bootstrap
- **Email Testing:** MailHog (SMTP)
- **Task Scheduler (Batch jobs):** Celery Beat , Redis
- **Caching :** Redis
- **Flask Extensions:** RESTful API with Flask-RESTful ,Flask-JWT-Extended, Flask-SQLAlchemy

DB Schema Design

The schema is organized relationally to ensure data integrity and efficient querying. Foreign keys establish clear relationships between entities. User's answers are efficiently saved and can be fetched for the user to view past quiz scores .



Architecture and Features

Architecture : In my project folder named quiz-master-v2 ,

Frontend (VueJS): Handles all user interactions including user/admin login, user signup , user/admin dashboard, quiz attempts, and score views , summary .

Routes (Flask):Manages the API endpoints and database interactions, including user authentication, CRUD operations for subjects , chapters , quiz by admin , user quiz details .

Database (SQLite): Stores data related to users info, quiz , and scores .

Redis & Celery: For caching and scheduling batch jobs like reminders and monthly reports.

General Features :

- Separate User and Admin login , Signup for new Users , Convenient logout button .
- User Authentication: Role-based access control (RBAC) for admin, user , and user/admin logins also used JWT authentication for token generation , and used hash functions for password hashing.
- Responsive UI: Developed with VueJS and Bootstrap for smooth and dynamic user experience.
- Caching: Implemented via Redis to enhance API performance and reduce database load.
- Search functionality within the user/admin dashboard, offering filters for Subject and chapters.

Admin Functionalities :

- Admin can manage Subjects , chapters, quizzes , questions on Dashboard with all the CRUD Operation .
- View Users option is placed on navigation bar clicking which admin will be redirected to user info page on which admin can see every registered user with its username , email , qualification and role (admin/user)
- Summary page shows charts for Quiz statistics (number of attempts for each quiz) , and User Quiz Participation (number of quizzes each user took) .


User Functionalities :

- Users can see available subjects and Quiz on the dashboard .
- Users can take the Quizzes according to the date given to the quiz if the deadline is passed user cant give that quiz .
- Timer is implemented for every quiz for the time duration specified by admin , if the user does not submit the quiz in time the quiz will auto submit after the timer reaches 0.
- Users can view scores and answer reviews for all attempted quizzes, including chosen and correct options. If the score is below 40%, a Retake Quiz option appears; otherwise, Passed is shown. An Export CSV button lets users download their quiz history and details.
- Summary page shows User performance over time for the attempted quiz .

Backend Jobs:

- **Daily Reminders:** Sent to users who have pending or incomplete quizzes.
- **Monthly Activity Reports:** Summary of user quizzes performance emailed to users .

VIDEO Link : Video demonstration of my project is available here:-

 [23f2004131_quiz_master_v2_MAD_project_video.mp4](#)