

Author

Kapishankar Sanjay Ashtankar

21f3000371

21f3000371@ds.study.iitm.ac.in

I graduated with a degree in mathematics from DACN, Nagpur, Maharashtra. My interests include finance and biology, and I love to read and swim in my free time.

Description

Building an exam preparation website for students. A user can sign up and log in. A student can attempt a quiz within the time specified by the admin. Admin is a super user of the app who can add subjects, and chapters, and schedule a quiz.

Technologies Used

- Frontend: Vue.js with CLI for advanced components
- Backend: Flask for API development
- Database: SQLite
- Caching: Redis
- Batch Processing: Celery with Redis as the message broker
- Styling: Bootstrap for responsive design and custom CSS
- Templating: Jinja2 for dynamic content rendering in backend reports
- Authentication: JWT(javascript web token) for role-based access control

Architecture and Features

The Quiz Application follows a structured architecture with Flask as the backend framework, Vue.js for the front end, and Celery for handling background tasks like CSV export. The backend manages user authentication, quiz creation, question storage, and result processing. The database is managed using SQLAlchemy, where users, quizzes, and scores are stored efficiently. The front end communicates with the backend through RESTful APIs, enabling a seamless user experience.

The application implements several key features. **User authentication** ensures that only registered users can access the platform, with JWT-based authorization for security. **Quiz management** allows the admin to create and manage quizzes, while students can attempt them within a specified time. **Scoring and performance tracking** store quiz results and generate performance reports. A **CSV export feature** is implemented using Celery, allowing admins to export user quiz data asynchronously. To download quiz performance reports, the application triggers a Celery task that processes the data and returns a downloadable CSV file. The Vue.js

frontend provides an intuitive interface for students to browse quizzes, attempt questions, and view their results. Additionally, the admin panel offers functionalities to manage users, quizzes, and export reports. This architecture ensures scalability, efficiency, and a smooth user experience across different roles.

API Resources

Quiz Master **API** runs on **port 5001** and provides a structured set of endpoints for managing authentication, subjects, chapters, quizzes, users, and results. Below is an overview of the key resources and their functionalities:

1. Authentication & User Management

- `http://127.0.0.1:5001/api/login` → Handles user authentication.
- `http://127.0.0.1:5001/api/signup` → Allows new users to register.
- `http://127.0.0.1:5001/api/users` → Manages user-related operations (fetch, delete).
- `http://127.0.0.1:5001/api/admin` → Admin-specific functionalities (admins cannot register themselves).

2. Quiz Structure & Content

- `http://127.0.0.1:5001/api/subject` → Manages subjects for quizzes(create, update, and delete).
- `http://127.0.0.1:5001/api/chapter` → Handles chapters under subjects.
- `http://127.0.0.1:5001/api/quiz` → Admin can create, update, and delete quizzes.
- `http://127.0.0.1:5001/api/question` → Admin can add, update, or remove quiz questions.

3. User Dashboard & Exam Handling

- `http://127.0.0.1:5001/api/user_dash` → Fetches user-specific data and progress.
- `http://127.0.0.1:5001/api/exam/<int:id>` → Handles quiz attempts and submissions.

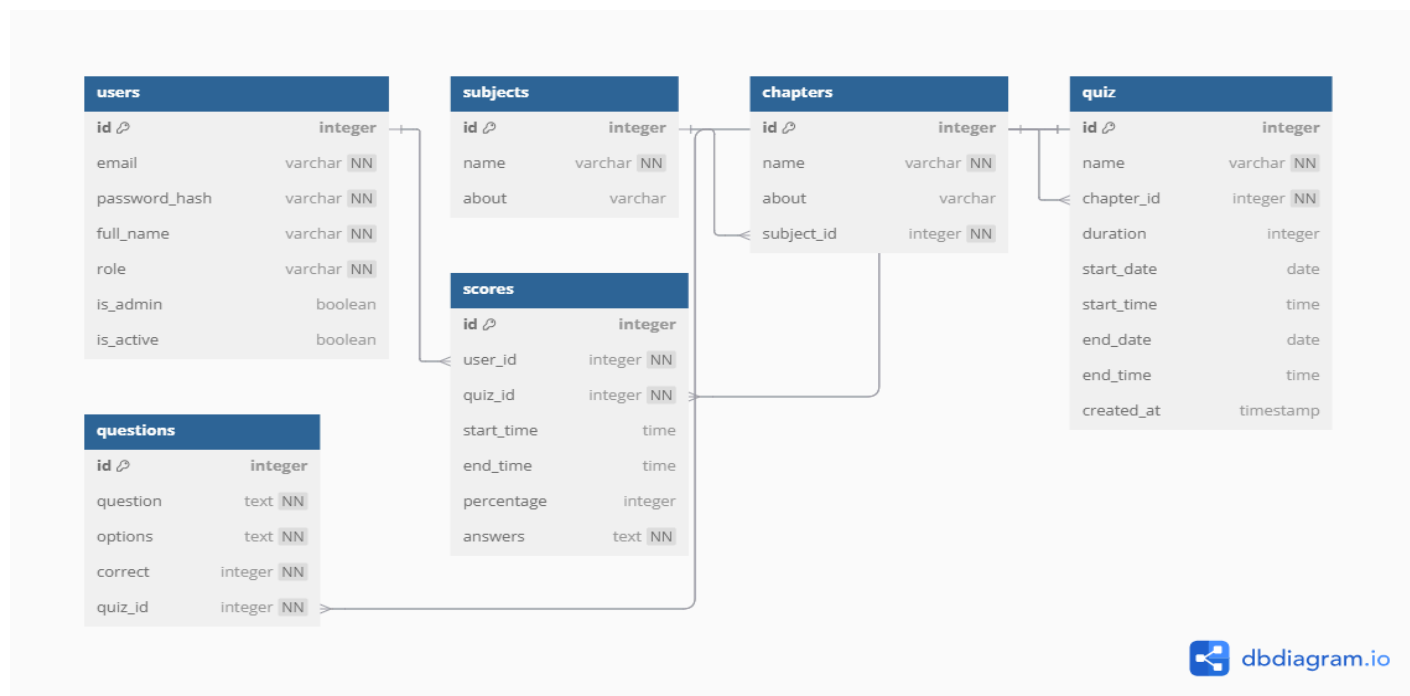
4. Scoring & Performance Tracking

- `http://127.0.0.1:5001/api/result/<int:score_id>` → Fetches quiz results for users.
- `http://127.0.0.1:5001/api/export/data` → Admins can export user quiz performance data (handled via Celery for async processing).

5. Charts & Analytics

- `http://127.0.0.1:5001/api/user_chart/image` → Generates user-specific performance charts.
- `http://127.0.0.1:5001/api/admin_bar_chart/image` → Provides bar chart analytics for admins.
- `http://127.0.0.1:5001/api/admin_pie_chart/image` → Displays pie chart analytics for admin insights.

DB Schema Design



Video

[Click here](#)