### Author

Fazal Iqbal V V

23f1003024

23f1003024@ds.study.iitm.ac.in

A undergrad student who majors in Computer Science. Currently in 3rd year.

### Description

This project is a webapp to design a Quiz Master app where the admin can can issue and edit the quizzes and the users can attend it. The admins can add subjects, chapters and quizzes and also view the summary of the top score and quizzes attended while the users are able to attend quiz and view the score. They also have the capabiity to view the summary of quizzes they attended.

### Technologies used

* **Flask:** Web framework for building the app.
* **SQLAlchemy:** ORM for database interaction.
* **SQLite:** Lightweight, file-based database.
* **Jinja2:** Template engine for rendering dynamic HTML.
* **Bootstrap:** Frontend framework for responsive UI.
* **SQLite:** Lightweight, file-based database.
* **RESTful:** For creating APIs.
* **JavaScript:** For enhanced experience and for generating charts.

### DB Schema Design

* **User**: id, username, full\_name, password, qualification, dob, role (admin/user)
* **Subject**: id, name, description
* **Chapter**: id, name, subject\_id (FK), description
* **Quiz**: id, chapter\_id (FK), name, date\_of\_quiz, time\_duration, description
* **Question**: id, quiz\_id (FK), chapter\_id(FK), question\_title, question\_statement, option1,option2, option3,option4, correct\_option

### API Design

The application includes a RESTful API for managing chapters, quizzes, and questions. The API is built using Flask and Flask-RESTful, providing endpoints for CRUD operations.

### Chapters API

* **POST /chapters**: Add a new chapter.
* **PUT /chapters**: Edit an existing chapter.
* **GET /chapters/<id>**: Retrieve chapter details by ID.
* **DELETE /chapters/<id>**: Delete a chapter by ID.

### Quizzes API

* **POST /quizzes**: Add a new quiz.
* **GET /quizzes/<id>**: Retrieve quiz details by ID.
* **DELETE /quizzes/<id>**: Delete a quiz by ID.

### Questions API

* **POST /questions**: Add a new question.
* **GET /questions/<id>**: Retrieve question details by ID.
* **PUT /questions**: Edit an existing question.
* **DELETE /questions/<id>**: Delete a question by ID.

### Architecture and Features

The project follows a well-structured MVC (Model-View-Controller) architecture using Flask. The **controllers** are handled through Flask routes and Flask-RESTful resources, organized into separate classes for chapters, quizzes, and questions. The **models** are defined using SQLAlchemy, with relationships established between subjects, chapters, quizzes, and questions. **Templates** are rendered using Jinja2, and Bootstrap is used for styling and responsiveness. All dynamic content is fetched and displayed through API calls, with proper error handling and flash messages to provide feedback to users.

Default features include quiz creation, question management, and quiz participation, while additional features like date-based quiz availability and real-time quiz duration tracking enhance the user experience. Users can view and attend upcoming or ongoing quizzes, with past quizzes automatically disabled. Quiz durations, question counts, and results are handled dynamically. The architecture uses Flask for the backend, SQLAlchemy, JavaScript and Jinja2 templates for dynamic HTML rendering.

### Video

<<Link to your online video of not more than 3 minutes length>>