### Author

Prakhar Swaroop

23f2002240

23f2002240@ds.study.iitm.ac.in

I am a diploma level student at IIT Madras.

### Description

In this project, we need to implement a quiz app in which there is an admin who has access to adding, editing and deleting subjects, quizzes, chapters and questions respectively and users which can attempt the quiz that the admin has created for them.

### Technologies used

Flask- It is the main web framework used to create the application.

Flask-login- used for managing user sessions, login/logout functionality

Flask-SQLAlchemy - to integrate SQL-Alchemy with flask so that database models and queries can be managed

Flask-WTF- It has been used I the process of creating forms for functionalities like login, register, etc.

Jinja2- it has been used in rendering templates, conditional rendering and to use python like logic in templates

Dotenv – used to load environment variables from .env file into the application

Werkzeug- used for creating password hashes and checking them

SQL-Alchemy- used to interact with the databases through python classes rather than writing sql queries. It is used as an ORM(Object Relational Mapper).

ChartJS- used to visualise the summary data on the admin dashboard.

### DB Schema Design

A diagram of a computer

AI-generated content may be incorrect.

Following is the structure of my Database:

The users table:

The users table contains the columns id,username, password\_hash, fullname, qualification, dob and there is a scores relationship which shows that a user can have multiple scores. The Id has been set as a primary key and username has been set to unique so that no two users can have the same username.

The subjects table:

The subjects table shows the different subjects in the system. It consists of id, name, description and chapters columns respectively. The chapters relationship tells us that a subject can have more than one chapter. Each subject must have a name, the name cannot be null

The Chapter table:

The chapter table consists of the id, name, description and subject\_id columns. The subject\_id column acts as a foreign key to link the chapter with its subject. The name and subject\_id cannot be null which ensures each subject has a name and is linked to a subject. Also, the quizzes relationship indicates that a chapter can have multiple quizzes.

The Quiz table:

The quiz table consists of the id,name of quiz, date of quiz, time-duration and chapter-id columns. Chapter id helps to connect the quiz with a particular chapter. The chapter id and name cannot be null as the quiz must have a chapter associated with it and it should have a name. The questions relations helps to state that a quiz can have multiple questions.

The Question Table:

The question table consists of the id, question statement of question, respective question numbers, the column for correct option and the quiz id column. quiz id helps to connect the question with a particular quiz. The correct option and respective choice options cannot be null. Quiz id helps to connect the question with a particular quiz.

The Scores Table:

The scores table consists of the id, total scored, timestamp, quiz id and user id columns. The quiz id acts as a foreign key to establish connection with the quiz model while the user id acts as a foreign key and helps in assigning a score to a specific user. The total scored, quiz id and user id fields cannot be null as we need to ensure that every score is assigned to a user and a quiz does not have a invalid score value.

### Architecture and Features

The project is contained in the root folder quiz\_ master\_23f2002240. The templates are present inside the templates folder which is present inside the app folder within the root directory. The controller is the routes.py file which is contained inside the app folder inside the root folder. The models.py file is also present inside the app folder. The form.py file which represents the forms used in the project is also located in the app folder. The main run.py file used to run the quiz app is located within the root directory itself.

The features that the admin can perform are managing the subjects, chapters, quizzes and quiz questions along with managing the users. The admin can create, edit and delete the subjects, chapters and quizzes and add quiz questions to the quiz. All these features have been implemented by defining routes for them in the routes.py file and rendering their templates accordingly. The values submitted in their respective forms are retrieved and updated in the database accordingly and then added to the templates. Thus, this enables the CRUD operations to be performed properly. The features available to the user are to attempt the quiz from the user dashboard and to filter the quiz based on the subject and chapter that they would like to attempt the quiz for.

### Video

<https://drive.google.com/file/d/1BkCvtjCzztFPSPF0v8oGAx44iLFF_3Ji/view?usp=sharing>