Author

Name **:** A.O. Navin Kumar
Roll Number **:** 23f1001686
Email **:** 23f1001686@ds.study.iitm.ac.in

Description

The project is a multi-user quiz app where admins create subjects, chapters and quizzes, while users attempt quizzes, view scores, and access a summary of their results.

Technologies used

1. Flask: It is used to build the Backend application which renders the JINJA templates, fetches the required data, stores the user input to the database and defines different views and routes.
2. Flask SQLalchemy: It is used for defining the data models and to perform all required query operation on the database
3. JINJA: for building HTML templates which make it more flexible to add required data and render it using python.
4. SQLite: It is a free and open-source relational database engine used for storing all required data as tables

DB Schema Design

1). User

1.) user_id – Integer, not null, Primary Key, Auto Increment
2.) username – String, not null, unique
3.) password – String, not null
4.) fullname – String, not null
5.) qualification – String
6.) dob – Date

2). Admin

1.) admin_id – Integer, not null, Primary Key, Auto Increment
2.) username – String, not null, unique
3.) password – String, not null
4.) fullname – String, not null

3). Subject

1.) subject_id – Integer, not null, Primary Key, Auto Increment
2.) name – String, not null, unique
3.) description – String

4). Chapter

1.) chapter_id – Integer, not null, Primary Key, Auto Increment
2.) subject_id – Integer, Foreign Key referencing subject_id from subject
3.) name – String
4.) description – String

5). Quiz

1.) quiz_id – Integer, not null, Primary Key, Auto Increment
2.) chapter_id – Integer, Foreign Key referencing chapter_id from chapter
3.) name – String
4.) date – Date, not null
5.) time_in_mins = Integer, not null

6). Question

1.) question_id – Integer, not null, Primary Key, Auto Increment
2.) quiz_id – Integer, Foreign Key referencing quiz_id from quiz
3.) question – String, not null
4.) option_a – String, not null
5.) option_b – String, not null
6.) option_c – String, not null
7.) option_d – String, not null
8.) answer – String, not null

7). Scores

1.) score_id – Integer, not null, Primary Key, Auto Increment
2.) quiz_id – Integer, Foreign Key referencing quiz_id from quiz
3.) user_id – Integer, Foreign Key referencing user_id from user
4.) score – Integer, not null
5.) question_ids – String, not null
6.) answers – String, not null
7.) timestamp – DateTime, not null

Each table contains an auto incrementing id field as primary key. Chapter table reference the subject_id it's a part of. Similarly quiz and question references chapter_id and quiz_id respectively. This creates a hierarchy where each question belongs to a particular quiz which in turn belongs to a chapter and a subject. The scores table references the quiz_id and user_id to keep track of which user submitted which quiz.

API Design

REST API:

- The timer for the quiz is implemented using REST API. It work by timestamping the start time and using that to calculate remaining time in the backend to make sure there are no malpractices.

- The data required to make the summary graphs for user and admins is processed and delivered using REST APIs. The overall summary statistics for users and admins is also implemented REST APIs. The required data is queried from the database and delivered as JSON which is compatible with chartJS

Web API:

- These Web APIs are used to add, edit and delete subjects, chapter, quizzes and questions.
- Scores are calculated in the backend and the results are stored to the database

Architecture and Features

Architecture:

- The model.py contains the schema of the database
- The controller folder contains 3 different controller modules auth_controller.py, admin_controller.py and user_controller.py. The auth controller contains the business logic for admin/user verification and user registration. The admin and user controller controls the routes for admin and user views respectively.
- The template folder contains the JINJA templates required for the front end
- The static folder contains the images and CSS
- The instance folder contains the database instance

Features:

- The admin can create, edit and delete subjects, chapter, quizzes and questions from each quiz. He/she can also schedule the quiz to start at a particular time and set a time limit for each quiz and look at the summary of the student s' performance in each quiz.
- The user can attempt a quiz check results and summary of their performance.
- The admin can check the summary of overall performance of each student like Average percentage and standard deviation.

Video

https://drive.google.com/file/d/1sZDlj7-dp7vXGRqEf1LPs8jjlG3lB6Uh/view?usp=sharing