Project Report
Modern Application Development I

Author

Chandan Kumar
23f1003124
23f1003124@ds.study.iitm.ac.in

I am currently at the diploma level of this BS Degree in Data Science and Application. I am doing this degree as a standalone. Enrolling in MAD 1 along with its project was a challenging yet rewarding experience. Coming from a non tech background and having minimal knowledge of coding, developing this web app was like a rollercoaster ride for me. I remember there were days I just couldn't see any progress and moments when I doubted my decision of taking both MAD 1 and its project together. However, perseverance and continuous learning led me to finally draft this project report.

Description

It is a multi-user app (one requires an administrator and other users) that acts as an exam preparation site for multiple courses. We were asked to use certain mandatory frameworks which includes **Flask** for application backend, **Jinja2** templating, **HTML**, **CSS** and **Bootstraps** for application front-end, and **SQLite** for database to build the quiz master application. Here admin is the superuser of the application and requires no registration. The admin will manage all the other users. Admin can create a new subject under which he can also create as many chapters as he wishes to. Then the admin can conduct a quiz for a particular chapter of a particular subject to assess the user's understanding of the chapter. He can add various questions under a quiz. A user can attempt any quiz of his/her own interest.

Technologies Used :-

1. Flask: A lightweight web framework that handles URL routing, HTTP request/response management, and session handling
2. Flask-SQLAlchemy: An extension that integrates the SQLAlchemy ORM into Flask.
3. SQLite: A file-based relational database used to store the application's data (e.g., users, subjects, chapters, quizzes, and scores).
4. Jinja2:The templating engine used by Flask to dynamically generate HTML pages.
5. Flask Session Management: Flask's built-in session handling mechanism is used to maintain state across HTTP requests (such as storing the logged-in user's ID).

DB Design: -

Relation - A user can attempt any quiz of his/her own interest. A subject can have multiple chapters however, a chapter can not exist without a subject which tells that the relation between subject table and chapter table is one to many. Similarly, a chapter can have more than one quizzes nevertheless a quiz can't be created without a chapter which tells that the relationship between chapter table and quizzes table is one to many.

API Design: -

Didn't use any api.

Wireframe Link: - https://drive.google.com/file/d/1_aEVCqAT6iidqbYnk92kNiPgc1tg2cCM/view

The following are the controllers that I have used: -

- @app.route('/')
- @app.route('/login', methods = ['GET', 'POST'])
- @app.route('/register', methods = ['GET', 'POST'])
- @app.route('/admin', methods = ['GET', 'POST'])
- @app.route('/subject', methods = ['GET', 'POST'])
- @app.route('/chapter/<int:subject_id>', methods = ['GET', POST'])
- @app.route('/quiz_management', methods = ['GET', 'POST'])
- @app.route('/add_quiz', methods = ['GET', 'POST'])
- @app.route('/add_question/<int:quiz_id>', methods = ['GET', 'POST'])
- @app.route('/user_dashboard')
- @app.route('/view/<int:quiz_id>')
- @app.route('/start_Exam/<int:quiz_id>')
- @app.route('/edit_question/<int:question_id>', methods = ['GET', 'POST'])
- @app.route('/delete_question/<int:question_id>')
- @app.route('/submit/<int:quiz_id>', methods = ['GET', 'POST'])
- @app.route('/edit_chapter/<int:chapter_id>', methods = ['GET', 'POST'])
- @app.route('/delete_chapter/<int:chapter_id>', methods = ['GET', 'POST'])
- @app.route('/scores')
- @app.route('/summary_user')
- @app.route('/logout')
- @app.route('/summary_admin')
- @app.route('/search')
- @app.route('/edit_quiz/<int:quiz_id>', methods = ['GET', 'POST'])
- @app.route('/delete_quiz/<int:quiz_id>')
- @app.route('/edit_subject/<int:subject_id>', methods = ['GET', 'POST'])

Video: -

Link : https://drive.google.com/file/d/1a_ClUj2bLjEwbx4styJL2w1umjs_qT6B/view?usp=sharing