Project Report

Author

Name: G Venkatesh, Roll no: 23f1001210, Email: 23f1001210@ds.study.iitm.ac.in,
I am currently pursuing diploma in programming from IITM and also a B.tech 3rd year student in sreenidhi institute of science and technology

Description

Objective was to build a web application for attempting quizzes as a part of their exam preparation for students focusing on using Flask as the backend framework for the application. The application is similar to the other exam preparation sites like “Prep” and “Testbook”

Technologies used

1. Flask: for request handling, rendering templates, routes to the application
2. Flask Sqlachemy: defining models, query operations on the database, committing changes to the database.
3. Jinja: for templating.
4. SQLite: for database tables
5. Javascript: For implementing timer for quiz and summary charts using chart.js

DB Schema Design

The Database has 7 Tables:

1. User(user_id INTEGER PRIMARY KEY, email STRING, password STRING, full_name STRING, qualification STRING, date_of_birth STRING)
2. Admin(id INTEGER PRIMARY KEY, username STRING NOT NULL UNIQUE, password STRING NOT NULL UNIQUE)
3. Subject(id INTEGER PRIMARY KEY, name STRING NOT NULL UNIQUE, description TEXT)
4. Chapter(id INTEGER PRIMARY KEY, name STRING NOT NULL UNIQUE, description TEXT, subject_id INTEGER NOT NULL REFERENCES subject(id))
5. Quiz(id INTEGER PRIMARY KEY, chapter_id INTEGER NOT NULL REFERENCES chapter(id), date_of_quiz DATE NOT NULL, duration STRING, remarks TEXT)
6. Question(id INTEGER PRIMARY KEY, quiz_id INTEGER NOT NULL REFERENCES quiz(id), question_title STRING NOT NULL, question_statement TEXT NOT NULL, option1 STRING NOT NULL, option2 STRING NOT NULL, option3 STRING NOT NULL, option4 STRING NOT NULL, correct_option STRING NOT NULL,image_path TEXT,audio_path TEXT)

7. Score(id INTEGER PRIMARY KEY AUTOINCREMENT, quiz_id INTEGER NOT NULL REFERENCES quiz(id), user_id INTEGER NOT NULL REFERENCES user(user_id), timestamp DATETIME, total_score INTEGER NOT NULL)

→This is the optimal schema design to perform all the database operations. It is design based on the requirement of the application.

Architecture and Features

Architecture

The whole project is inside the 'MAD_1' folder. Following is the flask app structure.

1. The "model.py" containing the database schema related definitions.
2. Templates folder is used to serve the html files.
3. Static folder contains the .jpg images files for background.
4. The "Instance" folder has the database defined.
5. The 'application' folder acts as the core package where the main logic is implemented.
6. The '__init__.py' file initializes the Flask application and connects it to the database.
7. The 'forms.py' file defines Flask-WTF forms for implementing search feature.
8. The 'routes.py' file contains the controller logic. Manages user authentication, quiz flow, admin functionalities, and more.

Features

1. User signup and login with validation (Signup only works if username doesn't already exist, returns alert if already exists)
2. Admin can manage subjects, chapters, quizzes, questions and manages the registered users.
3. The operations admin can perform are add, delete, edit.
3. Users can register and login to the user dashboard and can attempt any quiz of their choice.
4. Summary charts are included to check the performance of the users and the admin.
5. Search feature is implemented for admin and users.
6. Timer is included with the quiz for real time exam experience.
7. Users can check their previous score of the attempted quizzes.
8. Users can check the solutions of the attempted quizzes
9. Admin can attach images and audio with the questions.

ER diagram

https://drive.google.com/file/d/1IUPp22rzC4NNWC4fYsHNgb72IHTcXOJ9/view?usp=drive_link

Video

https://drive.google.com/file/d/1LKFWZoBal7nCJ0zhnypVfBoC-m9XndHG/view?usp=drive_link