Author

Snehal Satwik

23F3000660

23f3000660@ds.study.iitm.ac.in

Description

The project aims to develop a Quiz Master Web App using Flask and Python, allowing users to take quizzes on various subjects while enabling admins to manage subjects, chapters, quizzes, and questions. It includes user authentication, score tracking, a leaderboard, and data visualization for performance analysis. The system ensures a seamless quiz experience with a well-structured database and an interactive UI.

Technologies used

- **Python** – The core programming language used for backend development.
- **Flask** – A lightweight web framework to build the application.
- **Flask-SQLAlchemy** – ORM to interact with the database efficiently.
- **Flask-Login** – Manages user authentication and session handling.
- **Flask-WTF** – Helps in handling forms securely with CSRF protection.
- **SQLite** – A lightweight database for storing user data, quizzes, and scores.
- **HTML, CSS, Bootstrap** – Used for creating a responsive front-end UI.
- **Jinja2** – Flask's templating engine to render dynamic web pages.
- **Matplotlib** – Used for visualizing quiz performance through charts.

Purpose of These Technologies

- **Flask & Extensions** provide a modular and scalable web development approach.
- **SQLite** ensures easy database management without requiring a separate server.
- **Bootstrap & Jinja2** enhance the UI and improve user experience.
- **Matplotlib** adds visual representation of quiz data, improving insights.

DB Schema Design

User (1)---<(≈)>---Score (≈)--->(1)Quiz (1)|∨Question (≈)(≈)∧Chapter (1)∧|(≈)Subject (1)

Reasons Behind This Design

1. **Normalization**: The schema follows a relational approach to minimize redundancy.
2. **Efficient Queries**: Foreign keys ensure easy retrieval of data by joining tables.
3. **Data Integrity**: Constraints (e.g., NOT NULL, UNIQUE) prevent incorrect or duplicate entries.
4. **Scalability**: The modular design allows easy extension (e.g., adding more subjects or quizzes).

API Design

Authentication:

- /api/login (POST): Verifies credentials via flask_login and returns a response.
- /api/signup (POST): Hashes and stores user details in SQLite.

Architecture and Features

Project Organization

The project follows the **Model-View-Controller (MVC)** architecture using **Flask** as the backend framework.

- **Models (model.py)**: Defines the database schema using **SQLAlchemy ORM**. This includes tables for **Users, Subjects, Chapters, Quizzes, Questions, and Scores**.
- **Controllers (app.py)**: Handles request routing, user authentication, form submissions, and database interactions. Each route corresponds to a specific function that processes user input and returns appropriate responses.
- **Templates (templates/ folder)**: Contains **HTML files** using **Jinja2** for dynamic content rendering. Key templates include **login, user dashboard, admin dashboard, manage subjects, quizzes, questions, leaderboard, etc.**
- **Static Files (static/ folder)**: Stores **CSS, JavaScript, and images** for styling and interactive frontend elements.

Features Implemented

Subject and Chapter Management *(Admin Feature)*

- Admins can **create, update, and delete** subjects and chapters from the dashboard.
- Relationship maintained using **foreign keys** (one subject has multiple chapters).

Quiz and Question Management *(Admin Feature)*

- Admins can **add, modify, and delete** quizzes and questions.
- Questions are stored with **multiple-choice options** and a correct answer.

Taking a Quiz *(User Feature)*

- Users can attempt quizzes and submit answers.
- The system calculates the score and stores it in the database.

Leaderboard & Score Tracking

- A **leaderboard** ranks users based on quiz scores.
- Users can view **subject-wise performance charts** using **Matplotlib**.

Flash Messaging for Alerts

- Displays success/error messages using Flask's **flash()** functionality.

Summary Dashboard with Charts

- Users can see **subject-wise and time-wise quiz performance** using **bar and pie charts**.

Video

https://drive.google.com/file/d/1obLEHj4fwaa0wo4d85tMgBN9foDGVEv1/view?usp=sharing