Project Report: Quiz Management System

Modern Application Development - I

Author

Name: Ayan Hussain
Roll Number: 23f1000932
Email: 23f1000932@ds.study.iitm.ac.in

About Me

I am a student of IIT Madras passionate about building practical web applications. This project challenged me to integrate backend logic with frontend design while ensuring data consistency and user-friendly workflows. Overcoming hurdles like database relationships and real-time score tracking deepened my understanding of full-stack development.

Project Description

The **Quiz Management System** is a web application designed for creating, managing, and attempting quizzes. Key features include:

- User Authentication : Secure login / register .
- Role - Based Access : Separate dashboards for admins (create/manage quizzes) and users (attempt quizzes).
- Quiz Workflow : Admin create subjects, chapter, quizzes and questions, User attempt quizzes with time tracking and instant scoring.
- Analysis : Users can view scores and performance charts , Admin monitor quiz attempts and average scores.
- Database management : CRUD operations for quizzes, questions, and user scores.

Technology Used

1. **Flask**: Backend framework for routing and templating.
2. **SQLAlchemy**: ORM for database management (SQLite).
3. **Jinja2**: Dynamic HTML templating.
4. **Matplotlib**: Score visualization via charts.
5. **HTML/CSS**: Frontend design and styling.
6. **SQLite**: Lightweight relational database.

DB Schema Design

Models :

1. **User** (user_id, email, password, scores)
2. **Subject** (subject_id, name, chapters)
3. **Chapter** (chapter_id, name, subject_id, quizzes)

4. **Quiz** (quiz_id, chapter_id, time_duration, questions)
5. **Question** (question_id, quiz_id, options, correct_answer)
6. **Score** (score_id, user_id, quiz_id, score, timestamp)

Relationships:

- **One-to-Many**:
  - Subject → Chapters → Quizzes → Questions
  - User → Scores
- **Many-to-One**:
  - Quiz → Scores (Multiple users can attempt the same quiz)

Endpoints and Routes

User Routes:

- **/login, /register :** Authentication
- **/user_dashboard/<user_id> :** List of quiz
- **/start_quiz/<quiz_id> :** Quiz interface
- **/Score/<user_id>:** Score tracking

Admin Routes:

- **/admin_dashboard :** Manage subject and chapter
- **/add_quiz, /add_question :** Quiz creation
- **/quiz_management:** CRUD operation for quizzes
- **/admin_summary :** Analytics dashboard

Architecture and Features

Project Structure:

- app.py (Main application logic)
- models.py (Database models)
- templates/ (HTML pages)
- static/ (Images/Charts)

Key Features:

Dynamic Dashboards: Users can view upcoming quizzes, scores, and performance charts , whereas Admins can manage content and view analytics.

Search Functionality: Filter quizzes by chapter or keyword.

Automated Scoring: Immediate results after quiz submission.

Cascading Deletes: Ensures data integrity (e.g., deleting a subject removes its chapters/quizzes).

Video

🎬 Recording 2025-03-25 231057.mp4