

Author

Aaditee Iche

23dp3000056

23dp3000056@ds.study.iitm.ac.in

Diploma in Programming, IIT Madras

B. Tech, CE Undergraduate, MKSSS's Cummins College of Engineering for Women, Pune

Description

Quiz Master V1 is a multi-user exam preparation platform built using Flask, Jinja2, and SQLite. The project includes an admin role for managing subjects, chapters, quizzes, and users, and a user role for attempting quizzes and viewing scores. The database is programmatically created, and users interact through a responsive web interface.

Technologies used

Backend: Flask (Python) with Flask-SQLAlchemy

Templating: Jinja2

Frontend: HTML, CSS, Bootstrap

Database: SQLite

Flask Extensions: Flask-SQLAlchemy

Authentication: Custom-built using Flask sessions

DB Schema Design

Tables and Key Fields:

User: id (PK), username, password, full_name, qualification, DOB

Admin: id (PK), username, password

Subject: id (PK), name, description

Chapter: id (PK), subject_id (FK), name

Quiz: id (PK), chapter_id (FK), date_of_quiz, time_duration, remarks

Question: id (PK), quiz_id (FK), question_statement, optionA, optionB, optionC, optionD, correct_option

UserResponse: id (PK), user_id (FK), quiz_id (FK), question_id (FK), selected_answer, is_correct

QuizScore: id (PK), user_id (FK), quiz_id (FK), score, total_questions, timestamp

Architecture and Features

- Templates Folder: Contains all HTML web page files.
- Static Folder: Stores CSS and JavaScript files.
- Database File: quiz_master.db is the SQLite database file.
- Models.py: Contains all the database table models.
- App.py: Implements all controllers and application functionalities.

- **Controllers Folder:** Contains auth.py for authentication

Key Features:

- **User Authentication** – Secure signup and login system with validation to prevent duplicate accounts.
- **Admin Dashboard** – Admins can manage subjects, chapters, quizzes, and users.
- **Quiz Management** – Admins can create, edit, and delete quizzes and questions.
- **Question Format** – Each question supports multiple-choice options with a correct answer field.
- **Quiz Attempt System** – Users can take quizzes with a set time duration.
- **Score Calculation** – User responses are stored and evaluated to generate quiz scores.
- **Print Quiz Feedback** – Users can download or print their quiz feedback directly.
- **Quiz Feedback Report** – Detailed performance breakdown for users after completing a quiz.
- **View Class Average Score** – Users can compare their scores with the class average.
- **User Dashboard** – Users can track past quiz attempts and view performance history.
- **Database-Driven** – All quiz data is stored in an SQLite database with structured relationships.

Approach:

- **Database Design:** Structured tables for users, quizzes, questions, and scores to ensure smooth data management.
- **User Authentication:** Implemented a secure login/signup system using Flask sessions.
- **Admin Dashboard:** Enabled admins to create subjects, chapters, quizzes, and manage users.
- **Quiz Attempt System:** Users can attempt quizzes with a predefined time limit.
- **Feedback & Analysis:** Users can view quiz feedback and compare scores with the class average.

Video

<https://drive.google.com/file/d/1RqvR21gX2DKuPpASti96eUKnRL7kK8dL/view?usp=sharing>