

Project Report for Final Project

Modern Application Development - I

Author

Karthik Kalleti

Roll Number: 23f2001791

Email: 23f2001791@ds.study.iitm.ac.in

I am a college student who enjoys learning about programming and data science. This project was an exciting experience, and while there were challenges, overcoming them was rewarding.

Description

In this project, we developed Quiz Master, an interactive quiz application using Flask, RESTful APIs, SQLAlchemy, and Bootstrap. The application supports two types of users: Admins and Users. Admins can manage subjects, chapters, quizzes, and questions, while users can take quizzes and track their scores.

Key features include:

- User authentication (signup/login)
- Quiz management (create, edit, delete quizzes)
- Multiple-choice question format
- Score tracking and performance evaluation
- Time logging for quiz attempts

Technologies Used

1. Flask - Web application framework for Python.
2. Flask-SQLAlchemy - ORM for handling database operations.
3. Flask-RESTful - Extension for building REST APIs.
4. Flask-Login - User authentication and session management.
5. Datetime - Used for timestamping quiz attempts.
6. Security - User authentication and password storage.
7. Requests - Used for handling HTTP requests and responses.

Database Schema Design

- User: Users can have multiple quiz attempts. Relationship: One-to-many.
- Quiz: A quiz consists of multiple questions but belongs to a single subject. Relationship: One-to-many.
- Question: Each question belongs to a quiz. Relationship: Many-to-one.

User Table:

- username
- email
- password

Quiz Table:

- quiz_id (Primary Key)
- subject_id (Foreign Key)
- quiz_name

Question Table:

- question_id (Primary Key)
- quiz_id (Foreign Key)
- question_text
- options
- correct_answer

API Design

- User API: Handles user authentication and data retrieval.
- Quiz API: Fetches quiz details and user progress.
- Question API: Retrieves questions for a quiz and records user responses.

Architecture and Features

The project follows a modular structure:

- Backend: Flask application with APIs and database models.
- Frontend: HTML, CSS, and Bootstrap templates.
- Database: SQLite for storing user data, quizzes, and scores.

File Structure:

- app.py - Main entry point for the Flask app.
- backend/models.py - Defines database models.
- backend/controllers.py - Implements API logic.
- templates/ - HTML pages for UI.
- static/ - CSS and assets for styling.

Key Routes:

- / - Home page
- /login - User login
- /register - User registration
- /dashboard - User dashboard
- /quiz/<quiz_id> - Take a quiz

- /admin - Admin dashboard

This project helped reinforce concepts of web development, database management, and API integration. "Quiz Master" is a functional and interactive platform that simplifies quiz creation and evaluation for users.

Video link :  MAD-1 QM.mp4