Project Report: QuizMaster

Author Details:
Name : Gautham Krishna S
Roll No : 23F2000466
Email: 23F2000466@ds.study.iitm.ac.in

1. Setup Instructions:

1.Setup Venv:

```
python -m venv venv
```

2. Install dependencies using:

```
pip install -r requirements.txt
```

3. Run the Flask application:

```
flask run
```

4. Access the application at http://127.0.0.1:5000/.

2. Project Statement:

It is a multi-user app (requires one admin and other users) that acts as an exam preparation site for multiple courses.

3. Approach and Execution:

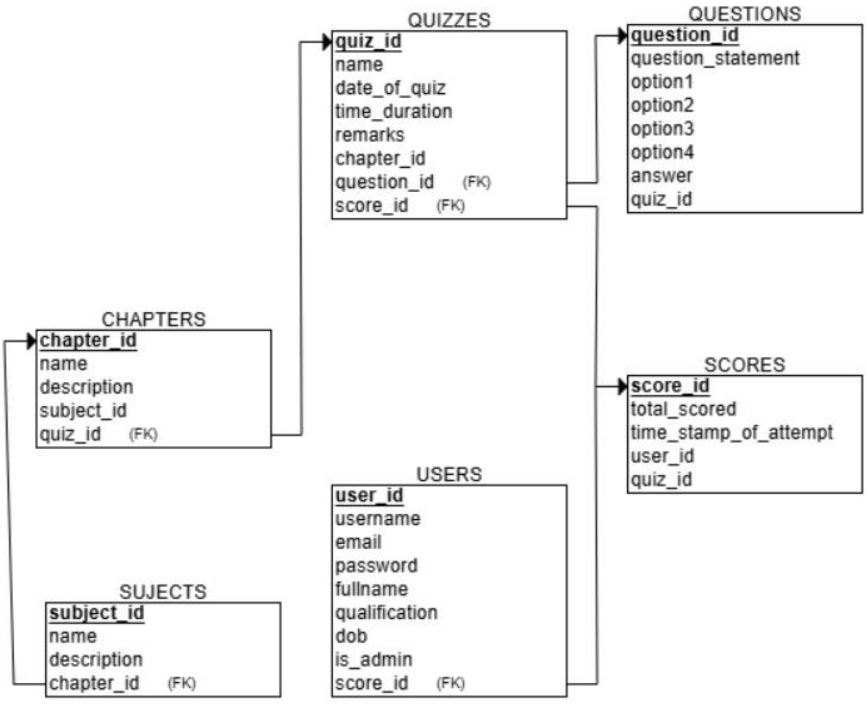
Step 1: Listing Functionalities and Requirements
I meticulously outlined the key features and requirements:

- Admin Management: Admin has complete control over the platform, including user management and quiz creation.
- User Registration & Authentication: Users can register, log in, and manage their accounts.
- Quiz Management: Admin can create subjects, add chapters, and include quiz questions under each chapter.
- User Quiz Participation: Users can select subjects, attempt quizzes, and view their scores.
- Data Security: Secure user authentication and role-based access.
- Database Integrity: Ensuring all CRUD operations maintain consistency.

Step 2: Step-by-Step Development Process
I approached the project methodically, focusing on one functionality at a time:

- User Authentication: Implemented user registration and login using Flask and SQLite.
- Admin Dashboard: Created an exclusive dashboard for the quiz master to manage users and quizzes.
- Quiz Setup: Admin can create subjects, chapters, and quizzes efficiently.
- User Interface: Built dynamic pages using Jinja2 templating for a seamless user experience.
- Quiz Functionality: Enabled users to attempt quizzes and receive scores instantly.
- Performance Tracking: Users can track quiz performance via a scoring system.
- API Development: Developed RESTful APIs for efficient data interaction.

4. Database Schema



5. API Endpoints (Controller)

The API endpoints, defined in `controllers.py`, facilitate CRUD operations. Key endpoints include:

- User Management: Register, Login, and Profile Management.
- Quiz Management: Create, Update, and Delete Quizzes.
- Question Handling: Add and Retrieve Questions.
- Score Tracking: Fetch User Scores and Quiz Performance.

6. Frameworks Used:

Backend: Flask (Python)

Frontend: Jinja2, HTML5, CSS, Bootstrap

Database: SQLite

Libraries: Flask-SQLAlchemy, Flask-RESTful, Matplotlib (for data visualization

7. Video link:

https://drive.google.com/file/d/1Ugx-WuqCiKHpBV9PR4rEtNDhB-rKgHD9/view?usp=drive_link