Quiz Master App

(Modern Application Development - 1)

Author

Name: Krish Gupta
Roll number: 23f2000792
Student Email: 23f2000792@ds.study.iitm.ac.in
I am a diploma level student at BS in Data Science and Applications at IIT Madras.

Description

This project aims to create a quiz management system where admins can create and manage quizzes, while users can register, attempt quizzes, and track their scores. It involves designing a structured database, implementing secure APIs, and developing a user-friendly web interface for seamless interaction.

Technologies used

1. **Flask** – A Python web framework used to build the web application.
2. **Flask-SQLAlchemy** – For managing the database, providing an Object-Relational Mapping (ORM) interface.
3. **SQLite** – A file-based database for storing quiz-related data.
4. **Matplotlib** – Used to visualise quiz results and generate graphical reports.
5. **Flask-Session** – Enables user session management for login authentication.
6. **HTML, CSS** – Used to build the front-end interface.
7. **Jinja2** – used for templating to render dynamic HTML content.
8. **Datetime** – Used to handle timestamps in quiz attempts.

DB Schema Design

1. User Class
   a. **id:** integer, primary key
   b. **username:** string, unique, cannot have a null value
   c. **password:** string, cannot have a null value
   d. **full_name:** string, cannot have a null value
   e. **qualification:** string, can have a null value
   f. **dob:** date, cannot have a null value
   g. **is_admin:** boolean, default is False

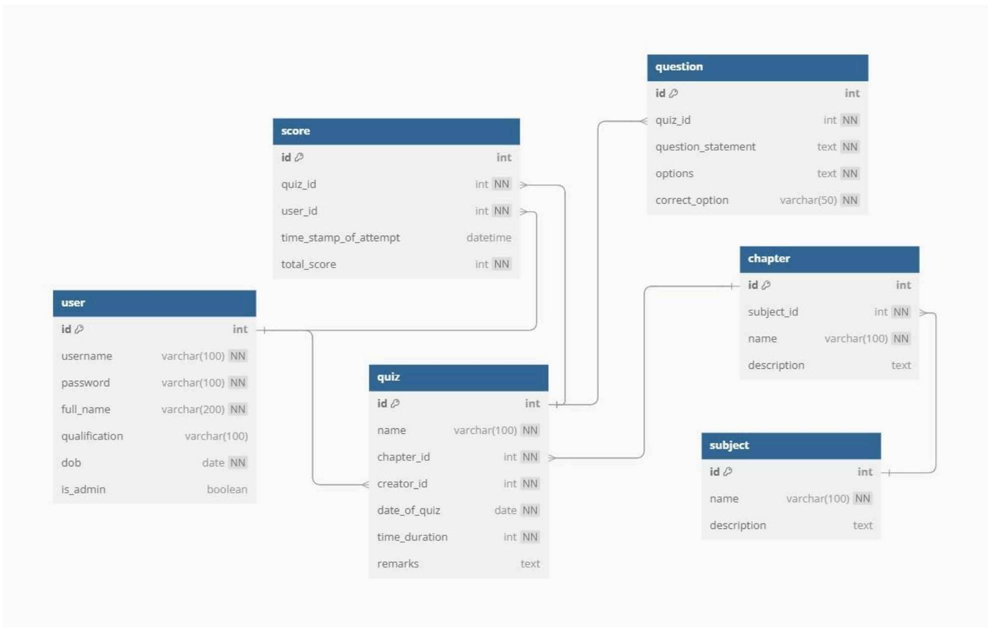
2. Subject Class
   a. **id:** integer, primary key
   b. **name:** string, cannot have a null value
   c. **description:** text, can have a null value

3. Chapter class
    a. **id:** integer, primary key
    b. **subject_id:** integer, foreign key (subject.id)
    c. **name:** string, cannot have a null value
    d. **description:** text, can have a null value

4. Quiz class
    a. **id:** integer, primary key
    b. **name:** string, cannot have a null value
    c. **chapter_id:** integer, foreign key (chapter.id)
    d. **creator_id:** integer, foreign key (user.id)
    e. **date_of_quiz:** date, cannot have a null value
    f. **time_duration:** integer, cannot have a null value
    g. **remarks:** text, can have a null value

5. Question class
    a. **id:** integer, primary key
    b. **quiz_id:** integer, foreign key (quiz.id)
    c. **question_statement:** text, cannot have a null value
    d. **options:** text, cannot have a null value
    e. **correct_option:** string, cannot have a null value

6. Score class
    a. **id:** integer, primary key
    b. **quiz_id:** integer, foreign key (quiz.id)
    c. **user_id:** integer, foreign key (user.id)
    d. **time_stamp_of_attempt:** datetime, default to current timestamp, cannot have a null value
    e. **total_score:** integer, cannot have a null value



API Design

The API in this project is designed for user authentication, quiz creation, and management of subjects, chapters, questions, and scores.

1. **User Authentication**
   - Registration (/register)
   - Login (/login)
   - Logout (/logout)
2. **Admin Functionalities**
   - Accessing the admin dashboard (/admin_dashboard)
   - Creating subjects (/create_subject)
3. **Quiz and Question Management**
   - Creating and managing quizzes (/create_quiz)
   - Adding questions to quizzes (/add_question)
   - Retrieving quizzes and questions for users
4. **Score Tracking**
   - Storing and retrieving user scores

The implementation uses Flask as the backend framework, with Flask-SQLAlchemy for database interactions and Flask session management for authentication control.

Architecture and Features

The project follows an MVC (Model-View-Controller) architecture.

- Controllers (app.py): It handles authentication, quiz management, and user interactions.
- Templates (templates/): It contains modular HTML files for user and admin dashboards, quizzes, authentication, and score tracking.
- Static Files (static/): It includes CSS (style.css) and images for styling and assets.
- Instance Directory: It stores configuration settings and the database instance.

Features Implemented:

Default Features:

- User Authentication: Secure login and registration.
- Quiz Management: Users take quizzes; admins create and manage them.
- Score Tracking: Users can view their scores in the dashboard.
- Subject & Chapter Management: Admins handle subjects and chapters.

Additional Features:

- Search Functionality: Users can search for quizzes and subjects.
- User & Admin Dashboards: Displays performance insights and management tools.
- Dynamic Quiz System: Interactive quiz-taking and result viewing.

```
└── quiz_master_23f2000792
    ├── codes/
    │   ├── instance
    │   ├── static/
    │   │   ├── style.css
    │   │   └── images
    │   ├── templates/
    │   │   ├── admin_dashboard.html
    │   │   ├── admin_navbar.html
    │   │   ├── admin_summary.html
    │   │   ├── base.html
    │   │   ├── create_chapter.html
    │   │   ├── create_question.html
    │   │   ├── create_quiz.html
    │   │   ├── create_subject.html
    │   │   ├── edit_chapter_.html
    │   │   ├── edit_question.html
    │   │   ├── home.html
    │   │   ├── login.html
    │   │   ├── messages.html
    │   │   ├── register.html
    │   │   ├── quiz_management.html
    │   │   ├── quiz_results.html
    │   │   ├── search_results.html
    │   │   ├── start_quiz.html
    │   │   ├── user_dashboard.html
    │   │   ├── user_navbar.html
    │   │   ├── user_scores.html
    │   │   ├── user_search_results.html
    │   │   ├── user_summary.html
    │   │   ├── view_quiz_user.html
    │   │   ├── view_quiz.html
    │   │   ├── view_subject_user.html
    │   │   ├── view_subject.html
    │   │   └── view_user.html
    │   └── app.py
    └──  project_report.pdf
```

Video

https://drive.google.com/file/d/1EPTSk8-kr3WE4P67ro96SJKSxQssohUE/view?usp=sharing