MAD1 PROJECT: Quiz Master V1

Roll Number: 21F1003882
Email: 21f1003882@ds.study.iitm.ac.in
Name: Arvind S

1. Introduction

QuizApp is a web application built with Flask that allows administrators to create and manage subjects, chapters, quizzes, and questions. Registered users can then browse available quizzes, attempt them under timed conditions, and view their performance history and summary statistics. The application features distinct roles for Admins and regular Users, a responsive web interface using Bootstrap 5, and a RESTful API for programmatic management (documented with Swagger UI).

2. Features

2.1. Core Features

- **Role-Based Access Control:** Clear distinction between Administrators (full content management) and Users (quiz taking and viewing own results).
- **Session-Based Authentication:** Secure user login, registration, and session management using Flask-Login. Includes basic (stubbed) "Forgot Password" functionality.
- **Responsive UI:** User interface built with Bootstrap 5, designed to work across different screen sizes.

2.2. Admin Features

- **Admin Dashboard:** Interactive dashboard using accordions to manage educational content.
    - **Subject Management:** Create, Read, Update, Delete (CRUD) subjects.
    - **Chapter Management:** CRUD chapters nested within their respective subjects.
    - **Quiz Management:** View quizzes nested within chapters, Add/Edit/Delete quizzes (linking to dedicated forms/pages).
- **Quiz Question Management:** Dedicated interface (linked from quiz lists) to CRUD Multiple Choice Questions (MCQ) with exactly 4 options for any given quiz. Mark the correct answer.
- **All Quizzes View:** A separate page listing all quizzes in the system with filtering/sorting options (implicit).

- **Admin Summary & Reports:**
    - Visual charts (Bar, Doughnut via Chart.js) showing:
        - Highest scores achieved per subject across all users.
        - Total quiz attempts per subject.
        - Number of quizzes per subject.
    - Content overview statistics (total users, subjects, chapters, etc.).
    - User activity ranking table based on the number of quiz attempts, with links to detailed user activity.
    - List of quizzes that currently have no questions added.
- **Search Functionality:** Dedicated search page to find Users (by username/email), Subjects (by name/description), and Quizzes (by title).

2.3. User Features

- **User Dashboard:** Browse available *active* quizzes, organized by Subject and Chapter using an accordion interface. View highest achieved score on previously attempted quizzes.
- **Quiz Taking:**
    - Interactive interface for attempting quizzes one question at a time.
    - Countdown timer based on the quiz duration.
    - Immediate answer checking ("Check" button) with visual feedback (correct/incorrect highlighting).
    - Navigation between questions (Previous/Next).
    - Final submission (manual or automatic on timeout).
- **User Summary:**
    - View history of past quiz attempts, including score, percentage, date, and time taken.
    - View personal performance charts (e.g., highest scores per subject, attempts per subject).

2.4. API & Documentation

- **RESTful API:** Provides endpoints (`/api/...`) for programmatic CRUD operations on most data models (Subjects, Chapters, Quizzes, Questions, Users, Roles, Attempts).
- **Swagger UI:** Integrated API documentation available at `/apidocs/`, automatically generated using Flasgger.

3. Technology Stack

- **Backend:** Python 3
- **Framework:** Flask

- **Database:** SQLAlchemy ORM (Defaulting to SQLite in development)
- **Authentication:** Flask-Login (Session-based)
- **Password Hashing:** Werkzeug Security Utilities
- **Forms:** Flask-WTF (integrating WTForms)
- **API:** Flask-RESTful
- **API Documentation:** Flasgger (Swagger UI)
- **Frontend:** Jinja2 (Templating), Bootstrap 5 (CSS Framework), JavaScript (for quiz interaction, charts)
- **Charting:** Chart.js

See `requirements.txt` for specific Python package dependencies.

5. Setup and Installation

1. **Prerequisites:**
   - Python 3.8+
   - `pip` package installer
2. **Create Virtual Environment:**
   ```
   python -m venv venv

   source venv/bin/activate  # Linux/macOS
   ```
3. **Install Requirements:**
   ```
   pip install -r requirements.txt
   ```
4. **Configuration:**
   - `config.py`. Key settings are loaded from environment variables or defaults:
     - `SECRET_KEY`: Essential for sessions and security. **Set a strong, unique secret key.**
     - `SECURITY_PASSWORD_SALT`: Used for password hashing. **Set a unique salt.**
     - `SQLALCHEMY_DATABASE_URI`: Defaults to SQLite in `instance/app.db`.
     - `ADMIN_USERNAME` (defaults to 'admin')
     - `ADMIN_EMAIL` (defaults to 'admin@example.com')
     - `ADMIN_PASSWORD` (defaults to 'ChangeMeNow!')
5. **Database Initialization:**
   - The application currently uses `db.create_all()` and `create_initial_roles_and_admin()` within the `if __name__ == '__main__':` block in `app.py`.
   - This means when you run `python app.py` for the *first time*, it will create the SQLite database file (`instance/app.db`) and tables, and attempt to create the default 'admin' and 'user' roles and the initial admin user.

Running the Application (Development):

```bash
python app.py
```

6. The application should be accessible at `http://127.0.0.1:5000`.

6. Usage

6.1. Admin User

1. Navigate to `http://127.0.0.1:5000/login` (or the root `/`).
2. Log in using the configured admin credentials (default: `admin` / `ChangeMeNow!`).
3. You will be redirected to the **Admin Dashboard** (`/admin/dashboard`).
4. **Manage Content:**
   - Use the "Add New Subject" form.
   - Expand subjects using the accordion buttons.
   - Add/Edit/Delete Chapters within a subject's expanded view.
   - Click the "Quizzes" button next to a chapter to expand the quiz list for that chapter.
   - Add/Edit/Delete Quizzes within the chapter's quiz list, or via the "All Quizzes" page.
   - Click the "Questions" button next to a quiz (in the nested list or on the "All Quizzes" page) to navigate to the Question Management page for that quiz.
   - Add/Edit/Delete MCQs with 4 options on the Question Management page.
5. **View Reports:** Navigate to the "Summary" page via the navbar to see charts and statistics. View "All Quizzes" via the navbar link. Use the "Search" page.

6.2. Regular User

1. Navigate to `http://127.0.0.1:5000/login`.
2. Click "Create an Account" to open the registration modal. Fill in details and register. You will be logged in automatically.
3. (Or) Log in with existing user credentials.
4. You will be redirected to the **User Dashboard** (`/dashboard`).
5. Browse Subjects/Chapters using the accordions.
6. Click "Attempt Quiz" on an active quiz card.
7. **Take the Quiz:**
   - Note the timer.
   - Select an answer for the current question.
   - Click "Check Answer" to see immediate feedback (correct/incorrect highlighting). Radios will be disabled.
   - Click "Next" (or "Previous" after checking) to navigate.

    - Click "Submit Quiz" on the last question or let the timer expire for auto-submission.
8. After submission, you are redirected to the **My Summary** page (`/dashboard/summary`).
9. View past attempts and performance charts on the summary page.

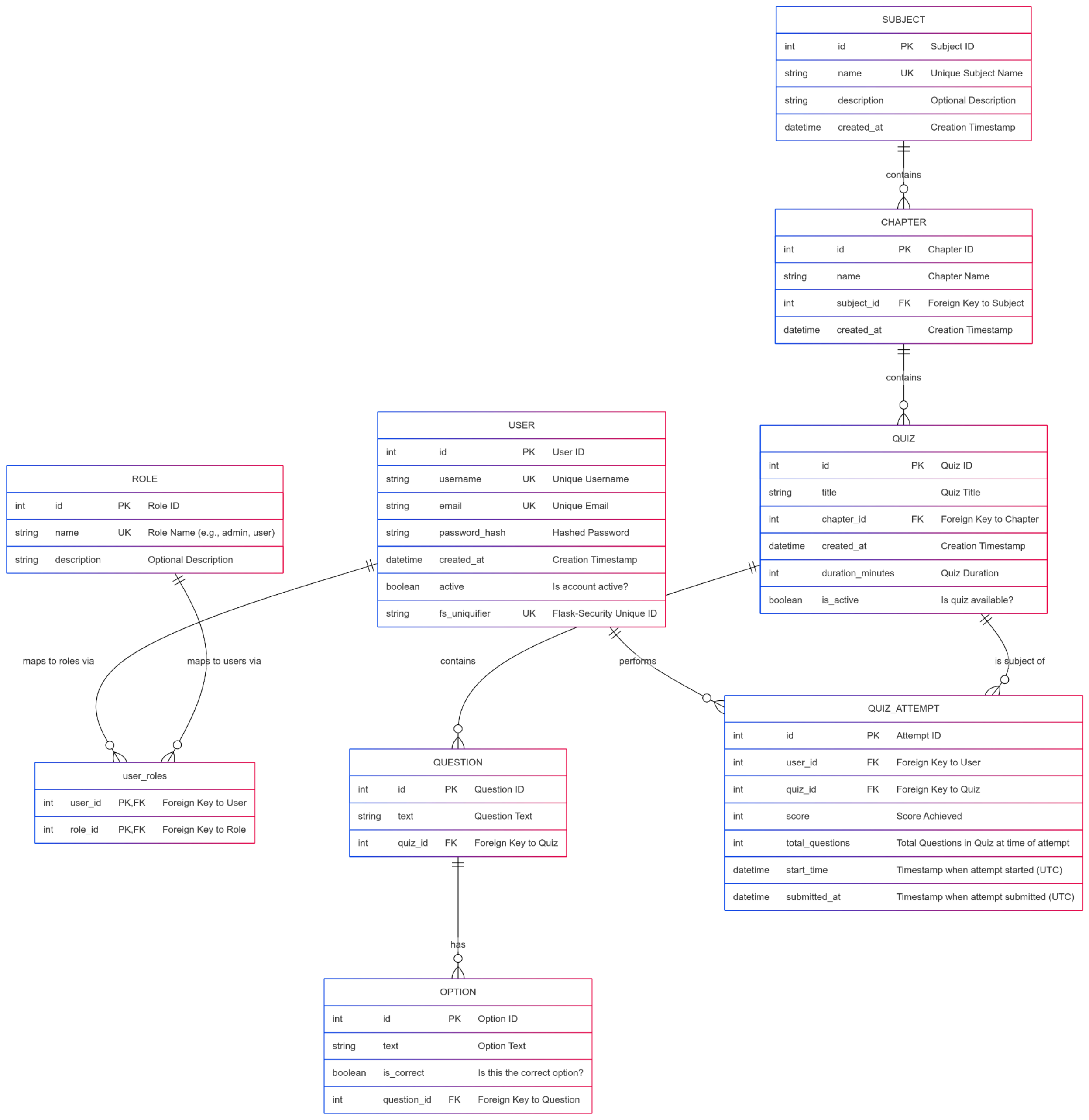
7. API Documentation

- A RESTful API is provided for programmatic access to resources.
- **Authentication:** The API currently relies on the same **session-based authentication** as the web application. Clients need to authenticate via the web login (`/login`) first and then include the obtained `session` cookie in the `Cookie` header of subsequent API requests. CSRF protection is disabled for the API blueprint prefix (`/api`).
- **Swagger UI:** Interactive API documentation is available at `/apidocs/`. Use this interface to explore available endpoints, view request/response schemas, and test API calls directly.
- **Base URL:** `http://127.0.0.1:5000/api`
- **Main Resources:**
    - `/subjects`
    - `/chapters`
    - `/quizzes`
    - `/quizzes/{quiz_id}/questions`
    - `/questions/{question_id}`
    - `/attempts`
    - `/users` (Admin only)
    - `/roles` (Admin only)

8. Bulk Data Upload

- A script `upload_initial_data.py` is provided to bulk-load Subjects, Chapters, Quizzes, and Questions from corresponding `.csv` files.
- **Usage:**
    1. Ensure the Flask app is running.
    2. Log in as admin via the web browser.
    3. Copy the `session` cookie value from browser developer tools.
    4. Run the script: `python upload_initial_data.py`
    5. Paste the session cookie value when prompted.
- The script requires the `requests` library (`pip install requests`).

ER Diagram of the Database



SUBJECT			
int	id	PK	Subject ID
string	name	UK	Unique Subject Name
string	description		Optional Description
datetime	created_at		Creation Timestamp

CHAPTER			
int	id	PK	Chapter ID
string	name		Chapter Name
int	subject_id	FK	Foreign Key to Subject
datetime	created_at		Creation Timestamp

USER			
int	id	PK	User ID
string	username	UK	Unique Username
string	email	UK	Unique Email
string	password_hash		Hashed Password
datetime	created_at		Creation Timestamp
boolean	active		Is account active?
string	fs_uniquifier	UK	Flask-Security Unique ID

ROLE			
int	id	PK	Role ID
string	name	UK	Role Name (e.g., admin, user)
string	description		Optional Description

QUIZ			
int	id	PK	Quiz ID
string	title		Quiz Title
int	chapter_id	FK	Foreign Key to Chapter
datetime	created_at		Creation Timestamp
int	duration_minutes		Quiz Duration
boolean	is_active		Is quiz available?

user_roles			
int	user_id	PK,FK	Foreign Key to User
int	role_id	PK,FK	Foreign Key to Role

QUESTION			
int	id	PK	Question ID
string	text		Question Text
int	quiz_id	FK	Foreign Key to Quiz

QUIZ_ATTEMPT			
int	id	PK	Attempt ID
int	user_id	FK	Foreign Key to User
int	quiz_id	FK	Foreign Key to Quiz
int	score		Score Achieved
int	total_questions		Total Questions in Quiz at time of attempt
datetime	start_time		Timestamp when attempt started (UTC)
datetime	submitted_at		Timestamp when attempt submitted (UTC)

OPTION			
int	id	PK	Option ID
string	text		Option Text
boolean	is_correct		Is this the correct option?
int	question_id	FK	Foreign Key to Question

Drive Link of the presentation video

https://drive.google.com/file/d/13o_Pky4Xx0Lem_7kkJtPIKv8OvG_wPq9/view?usp=drive_link