

Project Report

Author

Name: Jatin Nahata, Roll no: 23f2001705, Email: 23f2001705@ds.study.iitm.ac.in,
I am currently a student at the diploma level.

Installation Guide

To run the application install create a virtual environment by using 'python -m venv venv'

Now, enable created the environment you have, by running "venv\Scripts\activate" command on terminal.

After creating a virtual environment, run command 'pip install -r requirements.txt' and then all required packages will be installed. and then run the "app.py" file using the "python" command on the terminal.

Or

Simply, run command 'pip install -r requirements.txt' and then all required packages will be installed. and then run the "app.py" file using the "python" command on the terminal.

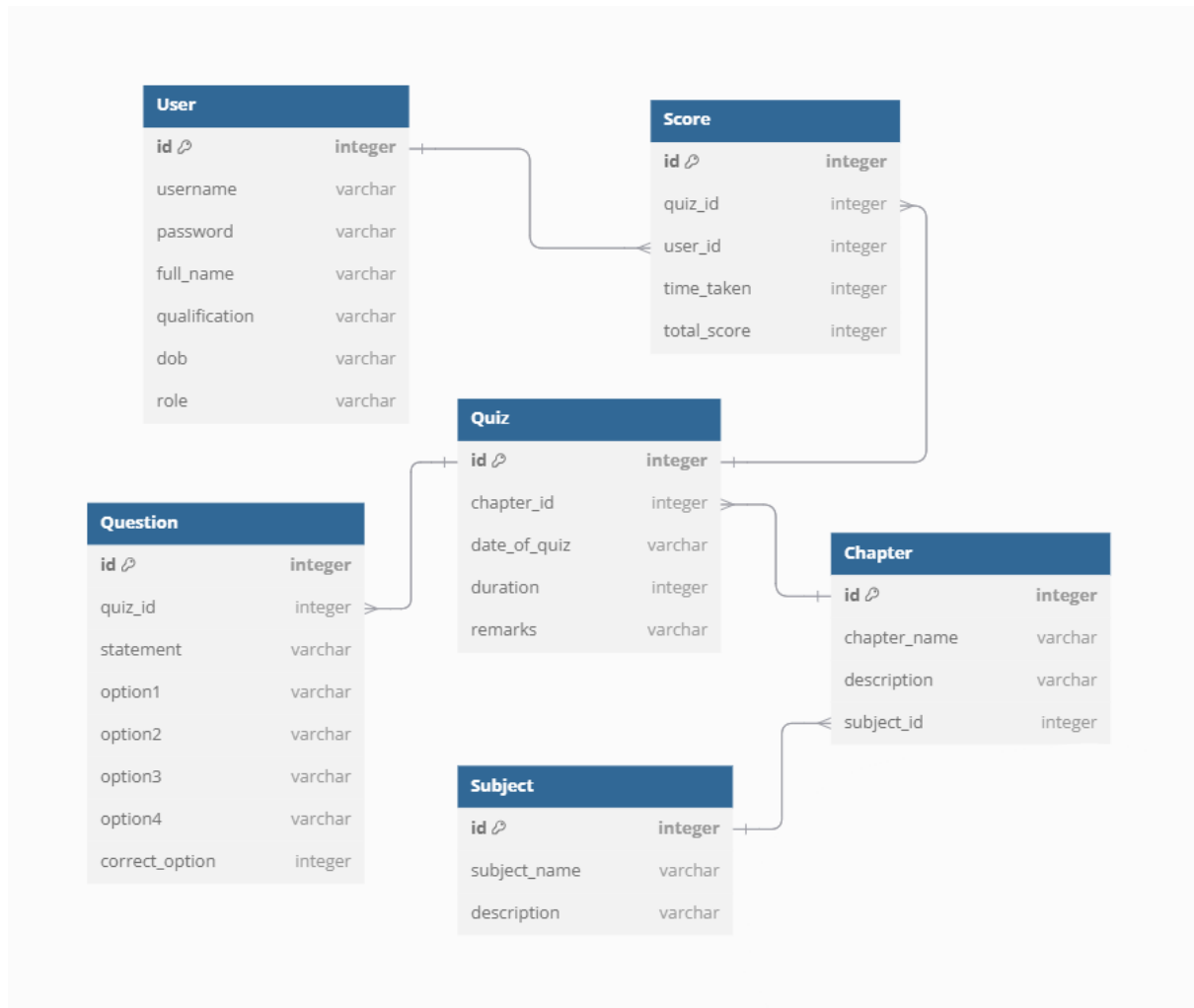
Description

Quest Mantra is a multi-user web application designed to help students prepare for various courses through structured quizzes. The platform supports two types of users: Administrator and Users (students). Administrator can manage course content, chapter, quizzes, and questions, while students engage in self-paced learning and assessments.

Technologies used

- I. **Flask**: for request handling, rendering templates, defining views/routes to the application
- II. **Flask Sqlalchemy**: defining models, doing query operations on the database, committing changes to the database.
- III. **Jinja2 (automatically integrated by flask)**: for templating, provides more flexibility to the html document.
- IV. **Sqlite**: for database tables
- V. **Datetime**: useful to track timer
- VI. **Flask Restful**: useful for building APIs

DB Schema Design (ERD Diagram)



API Design

The **User** field supports **create operation** and other field like **Subject, Chapters, Quiz and Questions** supports **read operation**.

Architecture and Features

Architecture

Following the recommended flask app structure.

- A. The *"model.py"* containing the **database schema** related definitions is divided into separate modules.
- B. The *controller folder* is containing the **controllers of the application** (divided into **Admin** and **User**).
- C. *Templates folder* is used to serve the **HTML files**.
- D. *Static folder* contains the **CSS, images and Api Documentation**.
- E. The *"Instance folder"* has the **database** defined (automatically created when we run application).
- F. *Venv* contains the *required python libraries* used to build the application.
- G. The *"app.py"* setup a base to **run application**.

Features

- **Login System:** Login form for **Admin** and **User** with username and password fields.
- **Register System:** User can register themselves with unique username.
- **Authentication:** Admin and users are authenticated and redirect admin to **Admin Dashboard** and users to **User Dashboard**.
- **Admin Dashboard:** Admin can create, edit, and delete subjects, chapters, and quizzes and can also see MCQ questions inside quizzes.
- **Quiz Management:** Admin can creates, edits, and deletes quizzes and MCQ questions.
- **User Dashboard:** Users can attempt available quizzes and view their scores.
- **Quiz Timer:** Quizzes have a timer for time-limited attempt.
- **User Score Tracking:** Scores are automatically recorded after quiz completion.
- **Search Functionality:** Admin can search users, subjects, and chapters whereas Users can search for quizzes by entering the name of the chapter or subject that the quiz belongs to.
- **Summary Charts:** Admin and users can view summary charts for performance and activity.

Video

For watching presentation: [click here](#)