

## Author

Aditya Basavaraj Narthi

24DP3000019

24dp3000019@ds.study.iitm.ac.in

I am a 3rd-year Electronics and Communication Engineering student, also pursuing a diploma in programming. My passion for building things began in childhood, driven by curiosity about how things work. I enjoy working on projects that blend hardware and software to create innovative solutions.

## Description

This project involves developing a Quiz Management System where the admin can manage subjects, chapters, quizzes, and questions while tracking user progress. Users can attempt quizzes multiple times on scheduled dates and review their past performance.

## Technologies used

**Flask:** Web framework for building backend.

**Flask-SQLAlchemy:** Allows you to interact with databases using Python objects.

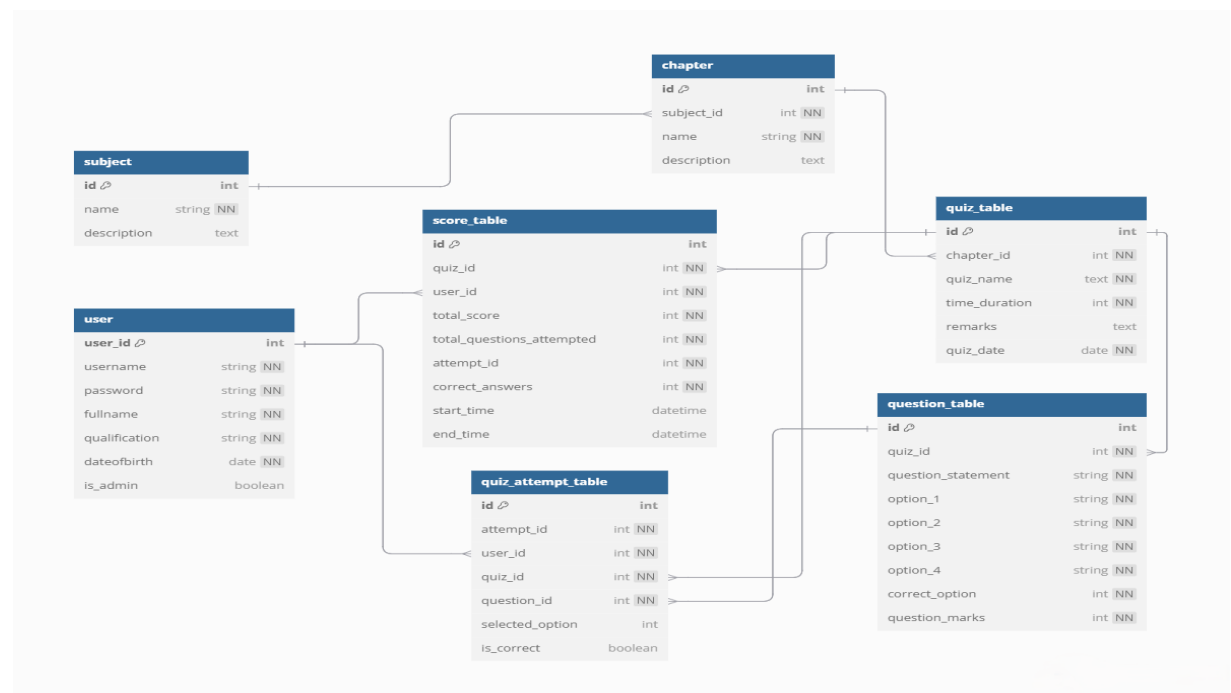
**Jinja2:** Used for embedding python logic within html pages.

**Bootstrap:** Enhances frontend styling and layout consistency.

**SQLite:** Database for storing users quizzes etc.

**Chart.js:** JavaScript library for visualizing data.

## DB Schema Design



- **User:** Stores user details, including login credentials, full name, qualification, and admin status.
- **Quiz\_table:** Holds information about quizzes, including their name, duration, date, and related chapter.
- **Subject:** Represents subjects, each containing multiple chapters.
- **Chapter:** Stores chapters linked to a subject, which contain quizzes.
- **Question\_Table:** Contains quiz questions, options, the correct answer, and the marks assigned to each question.
- **Quiz\_Attempt\_Table:** Records user quiz attempts, including selected answers and correctness.
- **Score\_Table:** Stores quiz results, including total score, number of questions attempted, and time taken.

## Architecture and Features

### QUIZ\_MASTER/

```

├── application/
│   ├── config.py - Sets up configuration settings for application
│   ├── database.py - Manages database connection and initialization
│   ├── models.py - Defines the database tables and relationships
├── controllers/
│   ├── admin_controllers - Multiple python files that handle admin related functionalities
│   ├── user_controllers - Multiple python files that handle user related functionalities
│
├── static/
│   ├── Styling and static images for applications frontend
│
├── templates/
│   ├── admin_templates - Multiple html templates designed for admin interface
│   ├── user_templates - Multiple html templates designed for user interface
│   ├── General login and logout templates
├── app.py

```