

# Project Report

## Author

Pranav Pathak

21f1005483

**21f1005483@ds.study.iitm.ac.in**

Pursuing MCA from BVICAM Delhi and BSc in Data Science and Programming from IIT Madras.

Love to code + play cricket + badminton. Currently life revolves around DSA and development.

Aspiring to make a career in Data Science.

**Description** This project is a multi-user web application designed as an exam preparation platform for multiple courses. It features two types of users: administrators who manage subjects, chapters, quizzes, and users, and students who can attempt quizzes, track their progress, and view results. The platform ensures a structured quiz-taking experience with recorded scores and history tracking.

## Technologies Used

- **Backend:** Flask (Python)
- **Frontend:** Jinja2, HTML, CSS, Bootstrap
- **Database:** SQLite
- **Flask Extensions:** Flask-SQLAlchemy, Flask-Login

## Purpose of Technologies:

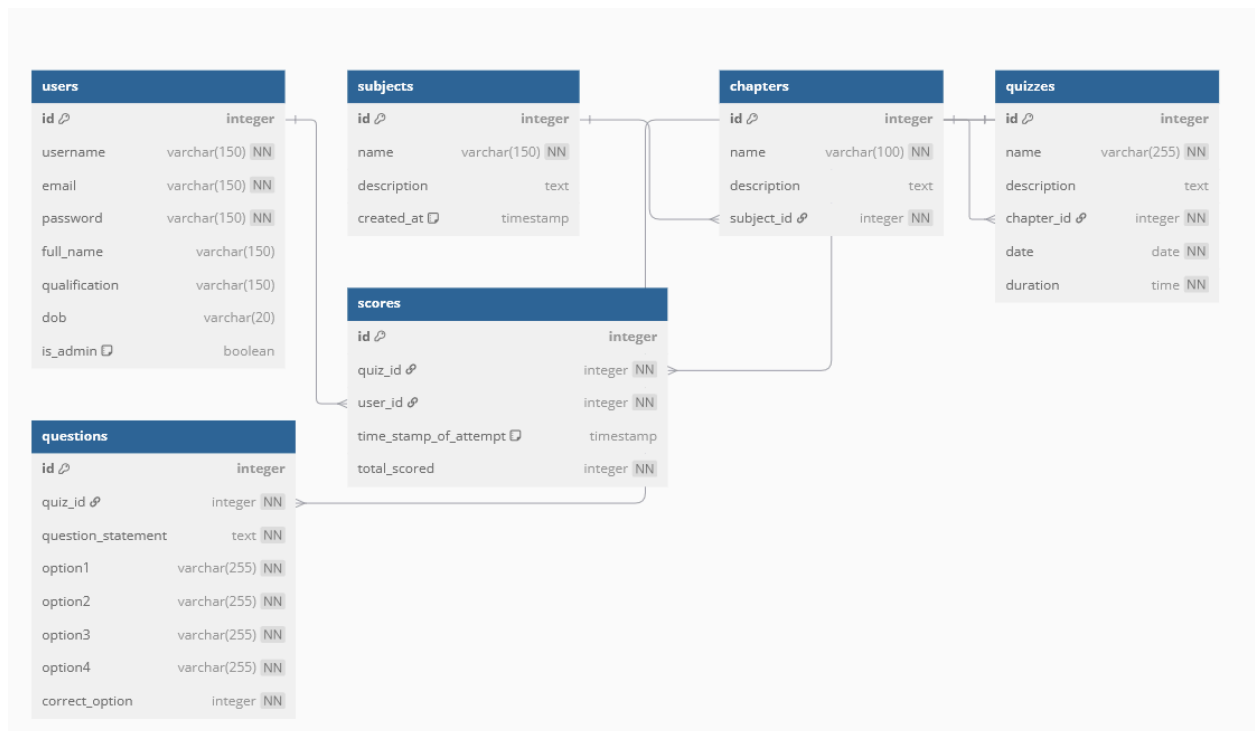
- Flask was chosen for its simplicity and efficiency in building web applications.
- SQLite was used for lightweight database management.
- Flask-SQLAlchemy was used as an ORM to manage database operations easily.
- Flask-Login provided authentication functionalities.
- Bootstrap provided a clean and responsive UI design.

## DB Schema Design

The database consists of multiple tables to efficiently manage users, subjects, chapters, quizzes, questions, and scores.

- **User:** Stores user details with fields `id`, `username`, `email`, `password`, `full_name`, `qualification`, `dob`, and `is_admin`. `username` and `email` are unique.

- **Subject:** Contains `id`, `name` (unique), `description`, and `created_at`.
- **Chapter:** Each chapter belongs to a subject (`subject_id` as FK) and has `id`, `name`, and `description`. A unique constraint prevents duplicate chapter names under the same subject.
- **Quiz:** Contains `id`, `name` (unique), `description`, `chapter_id` (FK), `date`, and `duration`. Deleting a chapter removes its associated quizzes.
- **Question:** Stores `id`, `quiz_id` (FK), `question_statement`, four answer options (`option1` to `option4`), and `correct_option`. Deleting a quiz removes its questions.
- **Scores:** Tracks quiz attempts with `id`, `quiz_id` (FK), `user_id` (FK), `time_stamp_of_attempt`, and `total_scored`.



## API Design

### Dashboard

- **GET** /dashboard  
**Description:** Renders the admin dashboard.  
**Access:** Admin only

### Subjects Management

- **GET** /subjects  
**Description:** View all subjects.  
**Access:** Admin only
- **GET, POST** /subjects/add  
**Description:** Add a new subject.  
**Access:** Admin only
- **GET, POST** /subjects/edit/<int:id>  
**Description:** Edit a subject by ID.  
**Access:** Admin only
- **POST** /subjects/delete/<int:id>  
**Description:** Delete a subject by ID.  
**Access:** Admin only

### Chapters Management

- **GET** /subjects/<int:subject\_id>/chapters  
**Description:** View all chapters under a specific subject.  
**Access:** Admin only
- **GET, POST** /subjects/<int:subject\_id>/chapters/add  
**Description:** Add a new chapter under a subject.  
**Access:** Admin only
- **GET, POST** /chapters/edit/<int:id>  
**Description:** Edit a chapter by ID.  
**Access:** Admin only

- **POST** /chapters/delete/<int:id>  
**Description:** Delete a chapter by ID.  
**Access:** Admin only

### Quizzes Management

- **GET** /quizzes  
**Description:** View all quizzes.  
**Access:** Admin only
- **GET** /quizzes/select\_chapter  
**Description:** Select a chapter to manage quizzes.  
**Access:** Admin only
- **GET** /chapters/<int:chapter\_id>/quizzes  
**Description:** View all quizzes under a chapter.  
**Access:** Admin only
- **GET, POST** /chapters/<int:chapter\_id>/quizzes/add  
**Description:** Add a new quiz to a chapter.  
**Access:** Admin only
- **GET, POST** /quizzes/edit/<int:id>  
**Description:** Edit a quiz by ID.  
**Access:** Admin only
- **POST** /quizzes/delete/<int:id>  
**Description:** Delete a quiz by ID.  
**Access:** Admin only

### Questions Management

- **GET** /quizzes/<int:quiz\_id>/questions  
**Description:** View all questions in a quiz.  
**Access:** Admin only
- **GET, POST** /quizzes/<int:quiz\_id>/questions/add  
**Description:** Add a new question to a quiz.  
**Access:** Admin only

- **GET, POST** /questions/edit/<int:id>

**Description:** Edit a question by ID.

**Access:** Admin only

- **POST** /questions/delete/<int:id>

**Description:** Delete a question by ID.

**Access:** Admin only

## Architecture and Features

- **Project Structure:**

- **app.py** – Handles the whole backend of the web application
- **controllers/auth\_routes.py** – Handles user authentication routes.
- **controllers/admin\_routes.py** – Handles admin-related routes.
- **controllers/user\_routes.py** – Handles user-related routes
- **controllers/decorators.py** – Handles admin authentication (prevents unauthorized access to admin dashboard)
- **controllers/routes.py** – Contains the route to home page
- **models/models.py** – Defines database schema.
- **templates/** – Contains Jinja2 templates for UI rendering. (ex: admin\_dashboard.html, login.html, register.html)
- **static/** – Stores css (mainly app designed with bootstrap)

## Features Implemented:

- **Authentication:** Secure login system for admins and users.
- **Admin Dashboard:** Subject, chapter, quiz, and user management.
- **Quiz Management:** Add, edit, delete quizzes and questions.
- **User Dashboard:** View available quizzes, attempt quizzes, and track scores.
- **Performance Analytics:** Score tracking and visual analytics.
- **Search Functionality:** Admin can search users efficiently.
- **Debug Messages:** Notifications for various actions (in console currently)
- **Minimalist UI:** Clean and user-friendly interface for ease of use.

Video Link:

[https://drive.google.com/file/d/1st\\_mktU9E0C02l-vQzfxA9wrIZkGBISU/view?usp=sharing](https://drive.google.com/file/d/1st_mktU9E0C02l-vQzfxA9wrIZkGBISU/view?usp=sharing)