

Author

- ✦ **Name:** *Rajveer Rahul kharadepati*
 - ✦ **Roll Number:** *23f3000717*
 - ✦ **Student Email:** *23f3000717@ds.study.iitm.ac.in*
-

✦ Project Description

This project is a **Flask-based web application** that provides:

- **Users:** Can register, log in, take quizzes, track scores, and search for subjects/quizzes.
 - **Admins:** Can manage users, quizzes, subjects, chapters, and block/unblock users.
 - **Role-Based Access:** Admins and users have different dashboards.
 - **Quiz System:** Time-based quizzes, score tracking, and user history.
 - **Search & Filters:** Users can search for quizzes, subjects, and their results.
-

✦ Technologies Used

Technology	Purpose
Python & Flask	Core framework for backend
Flask-SQLAlchemy	Database ORM for efficient query handling
Flask-Session	Session management for authentication
Flask-WTF	Form handling for registration & login
SQLite	Database for storing users, quizzes, and scores
Bootstrap	Frontend styling for UI enhancement
Jinja2	Templating engine for rendering dynamic HTML pages
Chart.js	Used for rendering graphs and user analytics

✦ Database Schema Design

The database consists of **six main tables**:

Table Name Description

User	Stores user details (ID, username, email, password, role, is_blocked)
Subject	Stores subjects (ID, name, description)
Chapter	Each subject has multiple chapters
Quiz	Stores quizzes associated with chapters

Table Name Description

Question Stores multiple-choice questions for quizzes

Score Stores user quiz scores, attempt date, correct/incorrect responses

◆ Constraints Implemented:

- email is **unique** in the User table.
- foreign key constraints exist between related tables.
- is_blocked boolean field is used to manage user status.

Design Reasoning:

- **Relational design** to ensure scalability.
 - **Separation of concerns** (subjects, chapters, quizzes, users).
 - **Efficient querying** using SQLAlchemy ORM.
-

✚ API Design

◆ The application includes **RESTful API endpoints** to:

- Fetch quizzes, subjects, and scores.
 - Admin management (add/edit/delete quizzes, subjects).
 - User authentication (login, registration).
-

✚ Architecture and Features

- **Controllers:** Flask **Blueprints** for authentication, admin, and user functionalities.
- **Templates:** HTML & Jinja2 templates inside the /templates folder.
- **Static Files:** CSS, JavaScript, images inside the /static folder.
- **Database Models:** Stored in /models.py, managed via Flask-SQLAlchemy.

◆ Features Implemented: ✓ **User & Admin Authentication** – Secure login/logout using sessions.

✓ **Quiz System with Timer** – Users can take time-based quizzes.

✓ **Score Management** – Users can track past performance.

✓ **Search & Filter Options** – Admin & users can search for quizzes & subjects.

✓ **User Blocking System** – Admin can block/unblock users from login.

✓ **Charts & Analytics** – Admin can view user performance in visual graphs.

✚ Video Submission

🎥 Project Demo Video:

<https://drive.google.com/file/d/1PSjxcuh9ER7rVd6sHGjDuHTlI2xNV9OD/view?usp=sharing>
