

# **MODERN APPLICATION DEVELOPMENT - 1**

## **QUIZ MASTER APP**

### **PROJECT REPORT**

**Roll No : DS23F3001827**

**Name : Kaushik. S**

#### **Question Statement**

Design and implement a comprehensive web-based Quiz Master Application that allows administrators to create, manage, and analyze quizzes while providing users with an interactive platform to take quizzes and track their performance.

#### **Key Objectives**

##### **1. Admin Functionality:**

- Allow admins to create, edit, and delete subjects, chapters, and quizzes.
- Enable question management (MCQ-based) with correct answer validation.
- Provide performance analytics (user-wise and subject-wise) using visual charts.

##### **2. User Functionality:**

- Allow users to register, log in, and attempt available quizzes.
- Display quiz results immediately after submission.
- Show historical performance through a dashboard with score trends.

##### **3. Technical Implementation:**

- Develop a Flask-based backend with SQLAlchemy for database management.
- Use Bootstrap 5 for responsive frontend design.

- Implement Chart.js for performance visualization.
- Ensure secure authentication and role-based access control.

#### **4. Expected Outcomes:**

- A fully functional quiz platform where admins can manage content efficiently.
- Users can take quizzes, receive instant feedback, and track progress.
- Data-driven insights for both users and administrators.

## **Solution Approach :**

To address the problem of creating a comprehensive Quiz Master Application, I adopted a structured development approach, focusing on modularity, user experience, and data integrity. The solution was divided into three core components:

### **1. Backend Development with Flask & SQLAlchemy:**

- Designed a relational database schema with models for Users, Subjects, Chapters, Quizzes, Questions, and Attempts to ensure structured data storage.
- Implemented role-based authentication (Admin/User) using Flask-Login, with restricted access to admin functionalities like quiz creation/deletion.
- Used SQLAlchemy ORM for efficient database operations, ensuring atomic transactions for deletions (e.g., cascading deletes for chapters/quizzes).

### **2. Frontend with Bootstrap 5 & Chart.js:**

- Built responsive UIs for dashboards (Admin/User) using Bootstrap 5, ensuring mobile compatibility.
- Integrated Chart.js to visualize performance metrics (e.g., subject-wise scores) dynamically fetched via Flask routes.
- Added real-time feedback for quiz attempts, with immediate scoring and historical results.

### **3. Problem-Solving & Optimization:**

- Challenge: Deleted subjects appeared as "Unknown" in user charts.  
Solution: Modified queries to filter out deleted subjects by cross-referencing active records.
- Challenge: Flash messages displayed on login pages.

Solution: Ensured redirects pointed to relevant pages (e.g., admin dashboard) and added flash message rendering in base.html.

- Challenge: Data integrity during deletions.

Solution: Implemented transactional deletes (e.g., deleting a chapter removes its quizzes/questions via ``db.session``).

The result is a scalable, secure, and user-friendly quiz platform that meets the problem statement's requirements, with clear separation of concerns (MVC architecture) and extensibility for future features like timed quizzes or multimedia questions.

## **Frameworks And Libraries Used :**

### **1. Frontend**

- Jinja2 templating
- HTML and CSS for basic styling
- Bootstrap for responsive design

### **2. Backend**

- Flask
- Javascript

### **3. Database**

- SQLite3
- DB Browser (for checks)

### **4. Authentication**

- Flask login
- Werkzeug security

### **5. Analysis**

- Charts JS

## **Presentation Video :**

Drive link :

<https://drive.google.com/file/d/14xomxTD6drzA5EMc7KM2co64FAD4frCv/view?usp=sharing>

## **Admin Login Username and Password :**

Username : [admin@example.com](mailto:admin@example.com)

Password :

scrypt:32768:8:1\$l6chYkrL2UIYF7Ai\$6eebedce64bf199a483ff  
c22b6cf88242a409063f74da3dd791d3cf33bfa57c387bf75ae788  
57b35f132ddb5241cb5249d18e051eb204d405beceb4c0ec7e4d

## **ER Diagram :**

