Project Report

- **Student details:**
  - Name - Sanish kumar
  - Email id - 23f3001252@ds.study.iitm.ac.in
  - Current level - Diploma(Jan25)
  - Registered course & Project - MAD2 Theory & MAD1 Project

1. Project Overview

- **Question Statement (Quiz Master v1) :**

  The goal of this project is to develop a Quiz Management System using Flask. The system allows two types of users: Admin and Users.

- **Admin functionalities:**
  - Create/Edit/Delete Subjects, Chapters, Quizzes, and Questions.
  - View top scores and summary charts.
  - Manage users (view and delete users).
  - Search for users, subjects, quizzes, and questions.
- **User functionalities:**
  - View the quizzes.
  - Attempt quizzes with time stamps.
  - View quiz scores and attempt history.
  - View summary charts for quiz attempts (by subject and month).
  - Search for subjects and quizzes.

2. Approach to Problem Statement

We approached the problem with the following structured plan:

- **Planning and Database Design:**
  - Identified all entities (User, Subject, Chapter, Quiz, Question, Score).
  - Designed database relationships (one-to-many, many-to-one)

- **Backend Development:**
    - Used Flask to set up routes and views.
    - Implemented RESTful routes for CRUD operations.
    - Used SQLAlchemy for ORM and database operations.
- **Frontend Development:**
    - Built HTML templates using Jinja2.
    - Used Bootstrap for responsive design.
- **API Development:**
    - Created endpoints to fetch subjects, quizzes, and scores.
    - Ensured APIs return JSON responses.
- **Data Visualization:**
    - Integrated Chart.js for summary charts (bar and pie charts).
- **Error Handling:**
    - Added custom error pages (404, 500).
    - Used Flask's flash messaging for user feedback.

3. Frameworks and Libraries Used

Library/Framework	Purpose
Flask	Web framework for routing and views
Flask-SQLAlchemy	ORM for database interactions
Flask-restful	Building RESTful APIs

Jinja2	Templating engine for HTML
Bootstrap	Frontend styling and responsiveness
Chart.js	Visualizing data with bar and pie charts
SQLite	Database management

5. API Resource Endpoints

Method	Endpoint	Results
GET	/api/subjects	Get all subjects
GET	/api/chapters	Get all chapters
GET	/api/quizzes	Get all quizzes
GET	/api/scores	Get all scores

4. ER Diagram

- **Entities and Relationships:**
  - **User** (id, name, username, password_hash, is_admin, type)
  - **Subject** (id, name, description)
  - **Chapter** (id, name, description, subject_id)
  - **Quiz** (id, title, chapter_id, date_of_quiz, time_duration, remarks)
  - **Question** (id, quiz_id, question_statement, option1, option2, option3, option4, correct_option)
  - **Score** (id, quiz_id, user_id, time_stamp_of_attempt, total_scored)

- **Relationships:**

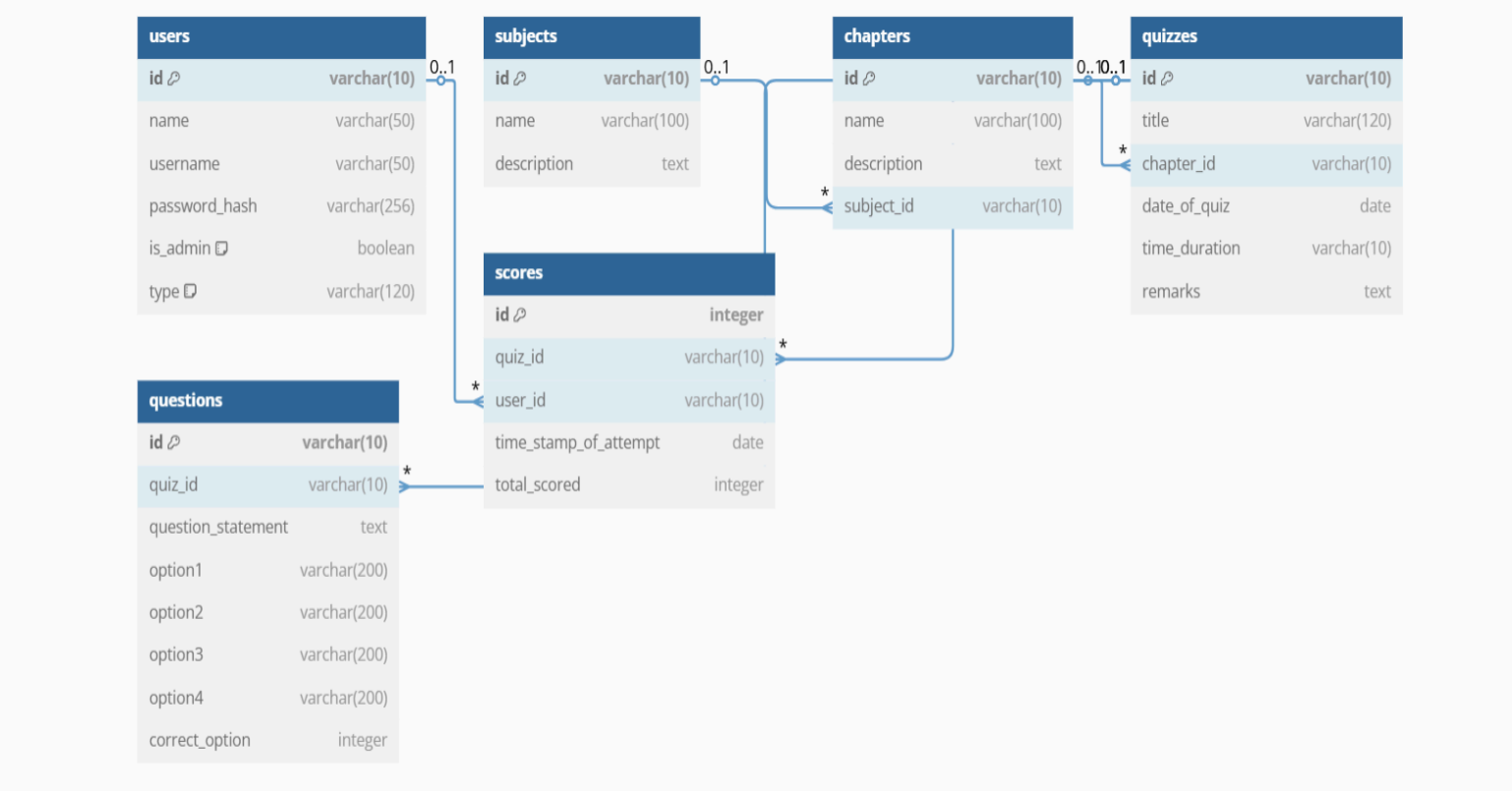
  Subject → Chapter (One-to-Many)

  Chapter → Quiz (One-to-Many)

  Quiz → Question (One-to-Many)

  Quiz → Score (One-to-Many)

  User → Score (One-to-Many)



- **Drive link of the presentation video:** [Presentation video link]