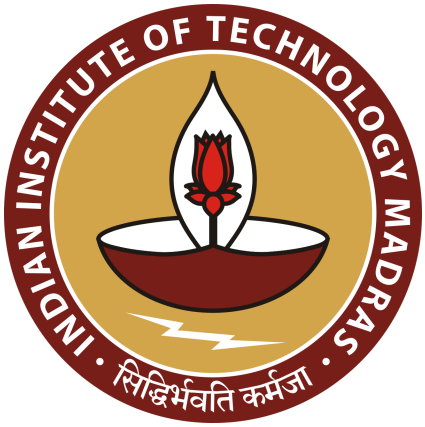

Indian Institute of Technology - Madras
BS in Data Science
Modern Application Development - 1
(Project)

By:
Chittari Sri Siva Siddhartha Varma
24f1002705
IITM (Diploma)

Author:

Name: Chittari Sri Siva Siddhartha Varma
Roll: 24f1002705
Student E-mail: 24f1002705@ds.study.iitm.ac.in
About me:
I am a student currently pursuing B.Tech (Aerospace Engineering) and BS (Data Science). I am from Hyderabad. I am a resident of Noida. I love solving problems and creating solutions using principles of engineering. I am skilled in Programming, CAD, Structural and Fluid Simulations (FEA & CFD)

Description:

This project is to create a platform for users to manage, plan and analyse their studying by testing. This platform enables creating that structure for students, and helps them with the rest by creating and being able to attempt tests. It also helps admin add, edit and delete the study content and the overall study structure.

Technologies used:

- Python (Basic Programming Language)
- HTML (markup, layout)
- Bootstrap (UI, front-end)
- CSS (Styling)
- Flask (Templating and Application)
- Sqlite3 (DBMS)
- Visual Studio Code (IDE)
- Chrome (Browser)
- Git/GitHub (Version Control

DB Schema Design:

- CREATE TABLE subjects (
    - id INTEGER PRIMARY KEY AUTOINCREMENT,
    - name TEXT NOT NULL UNIQUE,
    - description TEXT);
- CREATE TABLE chapters (
    - id INTEGER PRIMARY KEY AUTOINCREMENT,
    - subject_id INTEGER NOT NULL,
    - name TEXT NOT NULL,
    - description TEXT,
    - FOREIGN KEY(subject_id) REFERENCES subjects(id) ON DELETE CASCADE);
- CREATE TABLE scores (
    - id INTEGER PRIMARY KEY AUTOINCREMENT,

    - quiz_id INTEGER NOT NULL,
    - user_id INTEGER NOT NULL,
    - time_stamp_of_attempt TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    - total_scored INTEGER NOT NULL,
    - FOREIGN KEY(quiz_id) REFERENCES quizzes(id) ON DELETE CASCADE,
    - FOREIGN KEY(user_id) REFERENCES users(id) ON DELETE CASCADE);
- CREATE TABLE users (
    - id INTEGER PRIMARY KEY AUTOINCREMENT,
    - name TEXT UNIQUE NOT NULL,
    - username TEXT UNIQUE NOT NULL,
    - password TEXT NOT NULL,
    - Qualification TEXT NOT NULL);
- CREATE TABLE questions (
    - id INTEGER PRIMARY KEY AUTOINCREMENT,
    - name TEXT NOT NULL,
    - quiz_id INTEGER NOT NULL,
    - chapter_id INTEGER NOT NULL,
    - subject_id INTEGER NOT NULL,
    - question_statement TEXT NOT NULL,
    - option1 TEXT NOT NULL,
    - option2 TEXT NOT NULL,
    - option3 TEXT NOT NULL,
    - option4 TEXT NOT NULL,
    - correct_option INTEGER NOT NULL,
    - FOREIGN KEY(quiz_id) REFERENCES quizzes(id) ON DELETE CASCADE,
    - FOREIGN KEY(chapter_id) REFERENCES chapters(id) ON DELETE CASCADE);
- CREATE TABLE quizzes (
    - id INTEGER PRIMARY KEY AUTOINCREMENT,
    - name TEXT NOT NULL,
    - subject_id INTEGER NOT NULL,
    - chapter_id INTEGER NOT NULL,
    - date_of_quiz DATE NOT NULL,
    - time_duration TEXT NOT NULL,
    - remarks TEXT, FOREIGN KEY(chapter_id) REFERENCES chapters(id) ON DELETE CASCADE,
    - FOREIGN KEY(subject_id) REFERENCES subjects(id) ON DELETE CASCADE);
- CREATE TABLE admins (
    - id INTEGER PRIMARY KEY AUTOINCREMENT,
    - username VARCHAR[20],
    - password VARCHAR[20]);

Reasons for design:

- To be able to quickly extract for each subject or any other attribute on every table.
    - Advantages: Quicker, takes one query

    - Disadvantages: On a very large scale, this will be a highly inefficient use of space, It makes more sense to execute more commands, but since this is a small project, I have chosen to use this method of excessive redundant columns for every table to references the other tables.

API Design:

- None Implemented - Full control logic written into a singular app file

Architecture and Features:

- Architecture:
  The architecture is a modular flask architecture with some elements mixed or combined for ease of design:
    - Templates: Follow a common format with changes in main content of the page based on the page and dynamic content managed by flask app
      All the templates are located in the templates folder
    - Controllers: Use the routing decorators and formatting
      All the controllers located within the flask app file
    - Static files:
        - Styles: No separate style file, most style implement within the element or using bootstrap classes
        - Images: No images
        - Script: No JavaScript
    - User handling: Usage of Flask session to store user data across requests.
- Features:
    - Admin:
        - Subject: Add, edit, delete
        - Chapter: Add, edit, delete
        - Quiz: Add, edit, delete
        - Questions: Add, edit, delete
        - See Summary
    - User:
        - Quiz - attempt on given date
        - See upcoming quizzes
        - View past scores
        - View summary

Video:

https://drive.google.com/file/d/1O4CV89VsJDX15RzDxpTuu2TwEiA1SXET/view?usp=sharing