Author

Nirmaleshwaran S B
22f3000637
22f3000637@ds.study.iitm.ac.in
I has a HSC qualification, currently pursuing IITM BS degree program at diploma level as a Standalone degree

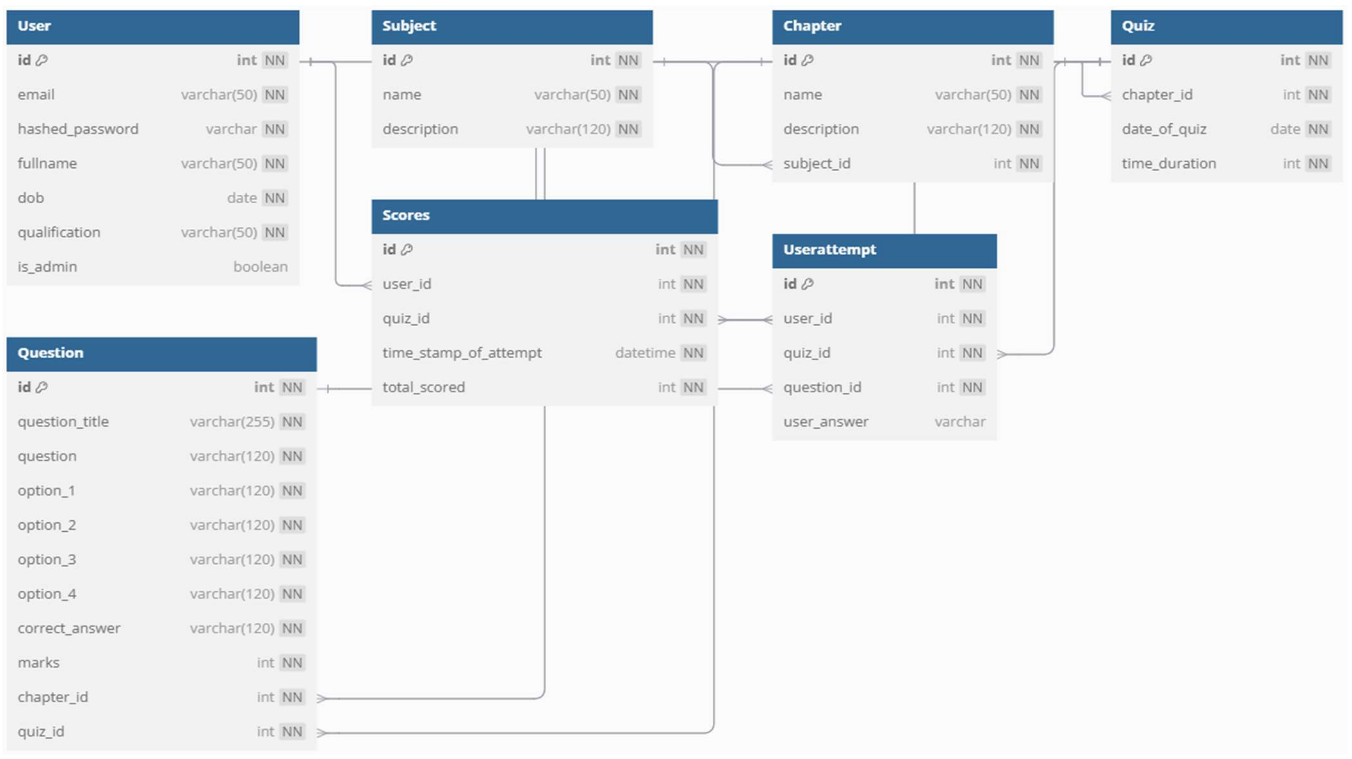
Description

Objective was to build a quiz master app in which users can take part in quizzes scheduled, track their scores accordingly. My idea is similar to the exam centres web app's layout.

Technologies used

- Python : programming language to drive the development stack
- HTML : for developing the web page
- CSS: for styling
- Javascript : for timer functionality
- Jinja : used for separation of logic and dynamic HTML rendering in the Flask app
- SQLite : database for storing users, subjects, chapters, questions, quizzes and scores data
- Flask : a web framework for building the web application backend
- Flask-SQLAlchemy : used to create ORM databases and query them
- MatplotLib : to create bar and pie charts for admin insights

DB Schema Design



User		
id	int	NN
email	varchar(50)	NN
hashed_password	varchar	NN
fullname	varchar(50)	NN
dob	date	NN
qualification	varchar(50)	NN
is_admin	boolean	

Subject		
id	int	NN
name	varchar(50)	NN
description	varchar(120)	NN

Chapter		
id	int	NN
name	varchar(50)	NN
description	varchar(120)	NN
subject_id	int	NN

Quiz		
id	int	NN
chapter_id	int	NN
date_of_quiz	date	NN
time_duration	int	NN

Scores		
id	int	NN
user_id	int	NN
quiz_id	int	NN
time_stamp_of_attempt	datetime	NN
total_scored	int	NN

Userattempt		
id	int	NN
user_id	int	NN
quiz_id	int	NN
question_id	int	NN
user_answer	varchar	

Question		
id	int	NN
question_title	varchar(255)	NN
question	varchar(120)	NN
option_1	varchar(120)	NN
option_2	varchar(120)	NN
option_3	varchar(120)	NN
option_4	varchar(120)	NN
correct_answer	varchar(120)	NN
marks	int	NN
chapter_id	int	NN
quiz_id	int	NN

User has 7 attributes with id being the primary key.
Subject has 3 attributes with id being the primary key.
Chapter has 4 attributes with id being the primary key and subject_id being the foreign key.
Quiz has 5 attributes with id being the primary key and chapter_id being the foreign key.
Question has 9 attributes with id being the primary key, chapter_id being the foreign key, and quiz_id being the foreign key.
Scores has 4 attributes with id being the primary key, user_id being the foreign key, and quiz_id being the foreign key.
Userattempt has 5 attributes with id being the primary key, user_id being the foreign key, quiz_id being the foreign key, and question_id being the foreign key.

User and Scores have a one-to-many relationship.
User and Userattempt have a one-to-many relationship.
Subject and Chapter have a one-to-many relationship.
Chapter and Quiz have a one-to-many relationship.
Chapter and Question have a one-to-many relationship.
Quiz and Question have a one-to-many relationship.
Quiz and Scores have a one-to-many relationship.
Question and Userattempt have a one-to-many relationship.

API Design

Register a user and login as well.
Subject field supports create, read, update and delete operations.
Chapter field supports create, read, update and delete opertions.
https://app.swaggerhub.com/apis/NirmalS/quiz-master_api/1.0.0

Architecture and Features

Architecture

1. The app.py file is in the root folder creates instances of app and also has models, templates, static, controllers, instance folders in root folder.
2. Controllers folder consist od routes.py and api.py files which handles the logic and api controls of the program
3. Templates folder serves html files
4. Static folder serves css, bar charts, pie charts and some images
5. Instance folder has predefined database file
6. The application follows the standard MVC architecture. The View of the application is created using HTML . The Controller is created using Python and Flask. The Model is created using SQLite/Flask-SQLAlchemy.

Model → SQLite + Flask-SQLAlchemy

View → HTML

Controller → Python + Flask

Features

1. User authentication: Register, Login and Admin Login
2. Admin dashboard: To perform crud operations for subjects and chapters, navigate to summary and quiz management
3. Quiz management: To perform crud operations for quizzes and questions by admin.
4. Search Functionality [Admin]: Can search Subjects, Quizzes and Users
5. Summary[User] : Tracks top scores of users subject wise
6. User dashboard: To attend quizzes in a given time, get results after completing the quiz and track their attempts.
7. Search Functionality [User] : Can search quizzes and results by date and scores.
8. Summary[User] : Tracks Attempts of the user month wise and subject wise.
9. Instant results: View scores after completing quizzes instantly.

Video

https://drive.google.com/file/d/1bEnSmlSZLBOX_xBW4FXSEfTRXnq1bU91/view?usp=drive_link