MAD 1 PROJECT DOCUMENTATION

Author

Name : Sumit

Roll No.: 22f2000848

mail: 22f2000848@ds.study.iitm.ac.in

Description

Introducing QuizMaster, a comprehensive web application designed for educational assessment and knowledge evaluation. QuizMaster provides a structured platform where administrators can create and manage subject hierarchies, chapters, quizzes, and questions, while users can take quizzes and track their performance over time.

QuizMaster organizes educational content hierarchically, allowing administrators to create subjects, add chapters within subjects, develop quizzes for specific chapters, and formulate multiple-choice questions for each quiz. This structured approach ensures a systematic method for knowledge assessment across various domains.

The application offers a dual-interface system tailored to different user roles. Administrators can manage the entire content structure, create assessment materials, and monitor user performance. Regular users can browse available subjects and chapters, take quizzes, and track their progress through detailed score reports.

Designed with user-friendliness in mind, QuizMaster makes it easy for both administrators to manage educational content and for users to assess their knowledge. The application provides immediate feedback and performance metrics, helping users identify areas for improvement and track their learning journey over time.

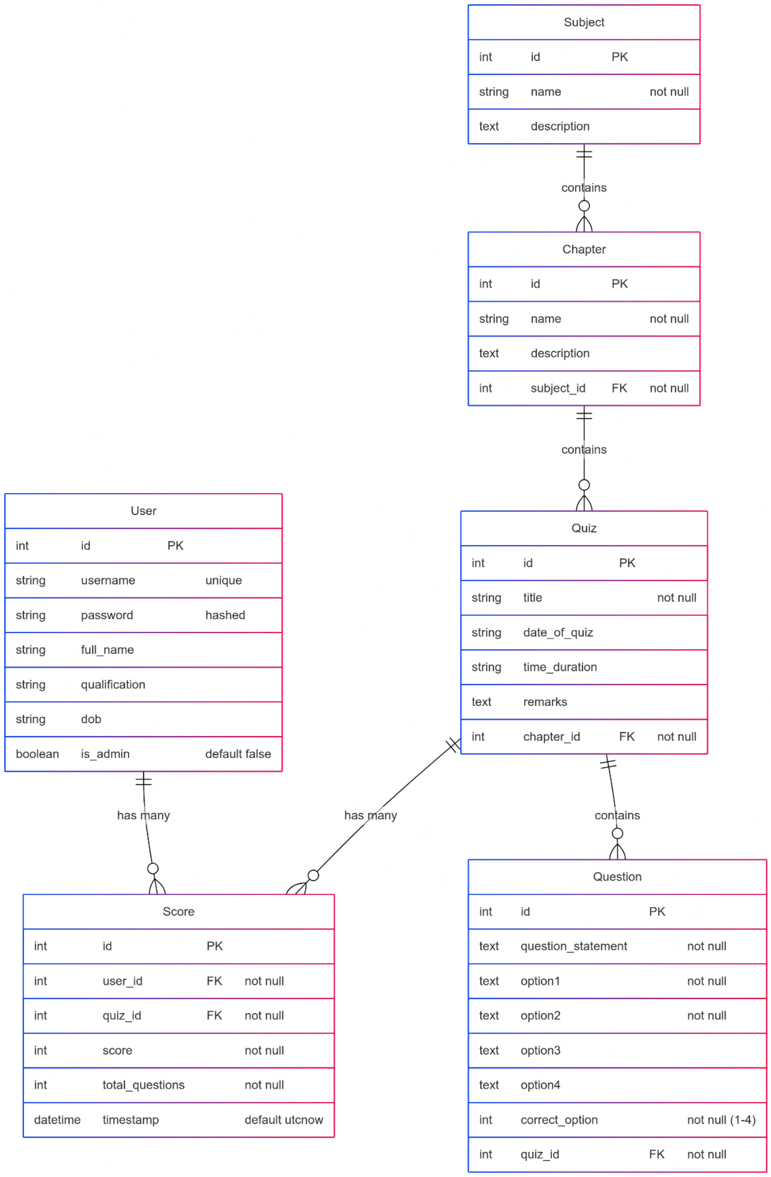
QuizMaster brings a streamlined approach to educational assessment, benefiting both educators who need to evaluate student knowledge and learners who want to test their understanding of various subjects.

Technologies Used

- **Flask**: A micro web framework used to build the web application.
- **Flask-SQLAlchemy**: A Flask extension that simplifies the integration of SQLAlchemy with Flask for database operations.

- **SQLAlchemy**: An Object-Relational Mapping (ORM) library used to interact with the database.
- **Werkzeug Security**: Used for password hashing and verification.
- **HTML, CSS, Bootstrap**: Web development technologies used to create the user interface and styling.
- **Jinja**: A template engine used to render dynamic HTML templates in Flask.
- **SQLite**: A lightweight disk-based database used for data storage.
- **Session Management**: Flask's session functionality for user authentication and state management.

DB Schema :



Subject			
int	id	PK	
string	name		not null
text	description		

Chapter			
int	id	PK	
string	name		not null
text	description		
int	subject_id	FK	not null

Quiz			
int	id	PK	
string	title		not null
string	date_of_quiz		
string	time_duration		
text	remarks		
int	chapter_id	FK	not null

User			
int	id	PK	
string	username		unique
string	password		hashed
string	full_name		
string	qualification		
string	dob		
boolean	is_admin		default false

Score			
int	id	PK	
int	user_id	FK	not null
int	quiz_id	FK	not null
int	score		not null
int	total_questions		not null
datetime	timestamp		default utcnow

Question			
int	id	PK	
text	question_statement		not null
text	option1		not null
text	option2		not null
text	option3		
text	option4		
int	correct_option		not null (1-4)
int	quiz_id	FK	not null

db Schema Design

The database model includes tables for Users, Subjects, Chapters, Quizzes, Questions, and Scores. Users are defined by id, username, password, full_name, qualification, dob, and is_admin flag. Subjects contain basic information with name and description. Chapters are linked to Subjects and include chapter details. Quizzes are connected to Chapters with quiz specifics like title, date, and duration. Questions store multiple-choice options and correct answers for each Quiz. Scores track user performance on quizzes with timestamps. The model supports hierarchical relationships between subjects, chapters, quizzes, and questions, as well as user performance tracking. This schema forms the foundation for storing and managing educational content and assessment data in the QuizMaster application.

Architecture and Features

In the root folder we have the app.py file, a readme.md file, a requirements.txt file, the instance folder with the SQLite database file, the template folder which contains the HTML+CSS templates and the static folder with CSS and JavaScript files.

1. app.py: This file acts as a controller for the app and has all the routes to different pages, database models, and authentication logic.
2. templates/: This directory contains HTML templates separated into admin/ and user/ subfolders for different user interfaces.
3. static/: This directory contains CSS stylesheets, JavaScript files, and any images used in the application.

The app features a dual-interface system with separate dashboards for administrators and regular users. Administrators can create and manage subjects, chapters, quizzes, and questions through an intuitive interface, while users can browse available content, take quizzes, and view their performance history. The application implements user authentication with password hashing and role-based access control to ensure secure operation.

Presentation Video :
https://drive.google.com/drive/folders/1weWf-Roc27I9rDA9zH9CGQEbdjvmzvZW?usp=sharing