

Quiz Master Application

A comprehensive web application for managing and taking quizzes. The application supports both user and admin roles, with distinct functionality for each.

Author

- **Full Name:** Lakshya Raj
- **Roll Number:** 21f1003767
- **Student Email:** 21f1003767@ds.study.iitm.ac.in
- **Current Designation:** Integration Engineer @ Randomwalk.ai

Technologies Used

- **Backend:** Flask, SQLAlchemy
- **Frontend:** HTML, CSS, JavaScript
- **Authentication:** Session-based authentication
- **Form Handling:** Flask-WTF
- **Database:** SQLite (development)
- **Password Security:** Werkzeug Security

Key Features

- Role-based access control (User/Admin)
- Comprehensive quiz management
- Hierarchical organization (Subject > Chapter > Quiz > Question)
- Quiz scheduling with time constraints
- Detailed score tracking and reporting
- Search functionality

Application Structure

Unset

```
quiz-master-app/

├─ app/                                # Application package
|
|   ├─ models/                        # Database models
|   |   └─ models.py                 # SQLAlchemy model definitions
|   └─ routes/                       # Route definitions
|       ├─ admin.py                  # Admin routes
|       ├─ auth.py                   # Authentication routes
|       └─ user.py                   # User routes
|
|   └─ static/
|       └─ style.css                 # Static files (CSS, JS, images)
|
|   └─ templates/                   # HTML templates
|       ├─ admin/                   # Admin templates
|       ├─ auth/                    # Authentication templates
|       └─ user/                    # User templates
|           └─ base.html
|           └─ index.html
|
|   └─ forms.py                     # Form definitions using Flask-WTF
|
|   └─ __init__.py                  # Application factory
|
├─ create_db.py                    # Script to create and initialize database
├─ requirements.txt                 # Project dependencies
├─ run.py                           # Script to run the application
└─ SwaggerFile.json                 # Swagger API documentation
```

Database Schema

User

- id: Integer (Primary Key)
- username: String (Unique)
- password_hash: String
- full_name: String
- qualification: String
- dob: Date
- created_at: DateTime
- Relationship: scores → Score

Admin

- id: Integer (Primary Key)
- username: String (Unique)
- password_hash: String

Subject

- id: Integer (Primary Key)
- name: String
- description: Text
- created_at: DateTime
- Relationship: chapters → Chapter

Chapter

- id: Integer (Primary Key)
- name: String
- description: Text
- subject_id: Integer (Foreign Key to Subject)

- created_at: DateTime
- Relationship: quizzes → Quiz

Quiz

- id: Integer (Primary Key)
- chapter_id: Integer (Foreign Key to Chapter)
- title: String
- date_of_quiz: Date
- time_duration: String
- remarks: Text
- created_at: DateTime
- Relationships:
 - questions → Question
 - scores → Score

Question

- id: Integer (Primary Key)
- quiz_id: Integer (Foreign Key to Quiz)
- question_statement: Text
- option1: String
- option2: String
- option3: String
- option4: String
- correct_option: Integer

Score

- id: Integer (Primary Key)
- quiz_id: Integer (Foreign Key to Quiz)
- user_id: Integer (Foreign Key to User)
- time_stamp_of_attempt: DateTime

- total_scored: Integer
- total_questions: Integer

Routes Overview

Authentication Routes

- `/login`: User login
- `/admin/login`: Admin login
- `/register`: User registration
- `/logout`: Logout for all users

User Routes

- `/dashboard`: User dashboard with recent scores and subjects
- `/subjects`: List all available subjects
- `/subjects/<subject_id>`: View a specific subject
- `/subjects/<subject_id>/chapters`: List chapters for a subject
- `/chapters/<chapter_id>`: View a specific chapter
- `/chapters/<chapter_id>/quizzes`: List quizzes for a chapter
- `/quizzes/<quiz_id>`: View a specific quiz
- `/quizzes/<quiz_id>/questions`: Get questions for a quiz
- `/quizzes/<quiz_id>/start`: Start a quiz
- `/quizzes/<quiz_id>/submit`: Submit quiz answers
- `/quizzes/<quiz_id>/result`: View quiz results
- `/quizzes/<quiz_id>/report`: View detailed quiz report
- `/quiz-history`: View user's quiz history
- `/search`: Search for subjects, chapters, and quizzes

Admin Routes

- `/dashboard`: Admin dashboard with system statistics
- `/subjects`: List all subjects
- `/subjects/create`: Create a new subject
- `/subjects/<id>`: View a specific subject
- `/subjects/edit/<id>`: Edit a subject
- `/subjects/delete/<id>`: Delete a subject
- `/chapters`: List all chapters
- `/chapters/create`: Create a new chapter
- `/chapters/<id>`: View a specific chapter
- `/chapters/edit/<id>`: Edit a chapter
- `/chapters/delete/<id>`: Delete a chapter
- `/quizzes`: List all quizzes
- `/quizzes/create`: Create a new quiz
- `/quizzes/<id>`: View a specific quiz
- `/quizzes/edit/<id>`: Edit a quiz
- `/quizzes/delete/<id>`: Delete a quiz
- `/quizzes/<quiz_id>/questions`: List questions for a quiz
- `/quizzes/<quiz_id>/questions/create`: Create a new question
- `/questions/edit/<id>`: Edit a question
- `/questions/delete/<id>`: Delete a question
- `/users`: List all users
- `/users/edit/<id>`: Edit details of a user users
- `/search`: Search across all content types

Installation & Setup

1. Clone the repository or extract the zip file shared:

Unset

```
git clone https://github.com/21f1003767/quiz-master-app.git  
  
cd quiz-master-app
```

2. Install dependencies:

Unset

```
pip install -r requirements.txt
```

Note: If you want to run the `check.py` script to verify the source code integrity, you'll need to install the additional dependency:

Unset

```
pip install checksumdir
```

3. Run Checksum check:

Unset

```
python check.py
```

4. Initialize the database:

Unset

```
python create_db.py
```

This script will create the necessary database tables and add an admin account

5. Run the application:

Unset

```
python run.py
```

6. Access the application at <http://localhost:5000>

Unset

Default Login Credentials

After running the database initialization script, you can log in with these credentials:

Admin Account

- **Username:** admin@quizmaster.com
- **Password:** admin123

Demo Video

https://drive.google.com/file/d/1NYw6uVol3WLMb2AoOg2-pcFrY94uVYK-/view?usp=drive_link