

QUIZMASTER

A full-featured quiz management system built with Flask that allows administrators to create and manage subjects, chapters, quizzes, and questions, while enabling users to take quizzes and view their performance statistics.

Features

Admin Features

- User management
- Subject management (add, edit, delete)
- Chapter management (add, edit, delete)
- Quiz management (add, edit, delete)
- Question management
- Admin dashboard with comprehensive statistics and visualizations
- Search functionality

User Features

- User registration and login
- Browse subjects and chapters
- Take quizzes
- View performance summaries with visual charts
- Search for subjects and quizzes

Installation and Setup

Prerequisites

- Python 3.6+
- pip (Python package manager)

Setup Instructions

1. Clone the repository

```
...  
git clone <repository-url>  
cd quiz  
...
```

2. Create and activate a virtual environment (recommended)

```
...  
python -m venv venv
```

On Windows

```
venv\Scripts\activate
```

```
# On macOS/Linux
source venv/bin/activate
...
```

3. Install dependencies

```
...
pip install flask flask-login matplotlib
...
```

4. Initialize the database

```
...
python app.py
...
```

The application will create the SQLite database file (`storage.db`) on first run.

5. Run the application

```
...
python app.py
...
```

The application will be available at `http://localhost:5000`

Usage

Admin Account Setup

1. Register a new account at `/register` with the role set to 'admin'
2. Log in with your admin credentials
3. You'll be redirected to the admin dashboard

Creating Quiz Content (Admin)

1. First, create subjects via the "Manage Subjects" section
2. Add chapters to subjects via the "Manage Chapters" section
3. Create quizzes within chapters using the "Manage Quizzes" section
4. Add questions to quizzes by clicking on "Manage Questions" for a specific quiz

Taking Quizzes (User)

1. Register a new account at `/register` with the role set to 'user'
2. Log in with your user credentials
3. Browse available subjects and chapters from the user dashboard
4. Select a quiz to take
5. Submit your answers and view your score
6. Check your performance statistics via the "View Summary" option

Project Structure

```
...
quiz/
├── static/          # Static files (CSS, JS, images)
├── templates/       # HTML templates
├── app.py           # Main application file
├── model.py         # Database models
└── storage.db       # SQLite database
...
```

Database Schema

The application uses the following models:

- User: Stores user account information
- Subject: Represents a subject area
- Chapter: Represents chapters within subjects
- Quiz: Represents quizzes within chapters
- Question: Stores quiz questions with multiple-choice options
- Score: Records user quiz attempts and scores

Visualizations

The application includes various visualizations:

- User performance summary with bar and pie charts
- Admin dashboard with participation and performance charts
- Overall accuracy visualization

Troubleshooting

- If you encounter database issues, try deleting the `storage.db` file and restarting the application to reinitialize the database
- For visualization issues, ensure matplotlib is properly installed
- If styles are not loading, check that your browser can access the static files

License

[Specify your license here]

Tanish kaur mehta

[Your name/organization]