Student Admission System

Project Overview

This web application manages student admissions. Students can submit their applications, and admins can approve or reject them. Upon approval, an admission letter PDF is generated and available for download.

Installation Instructions

1. **Clone the repository:**

   ```
   git clone https://github.com/your-username/student-admission.git
   cd student-admission
   ```

2. **Create a virtual environment:**

   ```
   python -m venv venv
   ```

3. **Activate the virtual environment:**

   ```
   venv\Scripts\activate
   ```

4. **Install dependencies:**

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

5. **Set up MySQL Database:**

   Create a database in MySQL:

   ```
   CREATE DATABASE studentmanagement;
   ```

6. **Run the application:**

   ```
   python app.py
   ```

The application will be accessible at `http://127.0.0.1:5000/`.

Usage Instructions

1. **Student Registration:**
    - Visit the **Home Page** and click on **Student Registration**.
    - Fill out the form with your details and upload the required files (degree certificate, ID proof).
    - After submission, you'll be redirected back to the home page.
2. **Admin Review:**
    - Admins can visit the **Admin Dashboard** to see all student applications.
    - Admins can approve or reject applications.
    - Upon approval, an admission letter PDF is generated.
3. **Student Login:**
    - After registration, students can log in with their email and phone number to view their application status.
    - If the application is approved, students can download their admission letter PDF.

Testing

Running Tests

- Write and run unit tests to ensure the application works as expected.
- You can use tools like **unittest** or **pytest** to write tests for the following components:
    - Registration Form
    - Admin Approve/Reject
    - PDF Generation
    - Student Login

Run the tests using:

```
pytest
```

Technology Stack

- **Backend:** Flask
- **Database:** MySQL
- **PDF Generation:** FPDF

- **File Uploads:** Flask's `request.files`
- **Frontend:** HTML/CSS (Basic for this project)

Assumptions

- MySQL is set up and running locally.
- The system will handle one admin and multiple student registrations.
- Users (students) are authenticated based on their email and phone.

Final Steps:

1. Push your project code to GitHub.
2. Complete the test cases and functional documentation.
3. Test the application thoroughly before submission.