Django File Upload and Summary Generation – Documentation

ARNAV MORE

Project Overview

This project involves developing a Django web application that allows users to upload an Excel/CSV file, processes the file to generate a summary report, and emails the summary. The summary is also displayed on a web page with a professional design using Bootstrap.

Project Structure

- **Forms**
    - forms.py: Defines a form for file upload.
- **Views**
    - views.py: Contains the logic for handling file uploads, processing the data, generating the summary, and sending the email.
- **Templates**
    - upload.html: A simple form for uploading files.
    - summary.html: Displays the summary of the uploaded data in a Bootstrap-styled table.
- **Static Files**
    - **Bootstrap CSS**: Used for styling the pages and tables.

Steps to Complete the Task

1. Web Page Creation

- **File Upload Form:**
    - Created a Django form in forms.py to allow users to upload an Excel or CSV file.
    - The form is rendered in the upload.html template with a simple and user-friendly interface using Bootstrap.

2. Data Processing

- **File Handling:**
    - The uploaded file is processed in the upload_file view located in views.py.
    - The file is read into a pandas DataFrame using pd.read_excel() for Excel files or pd.read_csv() for CSV files.
- **Data Transformation:**
    - Grouped the data by Cust Pin and counted the occurrences of each unique Cust Pin.
    - Assigned the corresponding Cust State to each Cust Pin.

- Reordered the columns in the DataFrame to display Cust State, Cust Pin, and the count (DPD(Count of Cust Pin)).

3. Summary Display

- **Bootstrap Table:**
  - Converted the processed DataFrame to HTML using summary_df.to_html(index=False).
  - Rendered the summary in a Bootstrap-styled table within the summary.html template, ensuring it stretches evenly across the container.

4. Email Functionality

- **Email Summary:**
  - Configured Django's email backend in settings.py to use SMTP (e.g., Gmail).
  - Sent the summary report via email with the subject "Python Assignment - Arnav More" to the specified recipient using the send_summary_email function.

5. Deployment

- **GIT:**
  - Created a branch named Arnav-More and pushed the code to the repository.
- **Deployment:**
  - Deployed the Django application to an open-source server (e.g., Heroku) and provided a live URL for testing.