**Project Documentation**

**Project Overview**

This project is a Flask-based web application for processing and classifying Excel files. It allows users to upload data files, classify items by department, generate reports, and download results in Excel format.

**Features**

* Upload Excel files for item classification and department mapping.
* Process and summarize pharmacy reports by sales, advance, and credit.
* Generate downloadable Excel reports.
* Flash messaging for user feedback.
* Organized result storage in a designated folder.

**Technologies Used**

* Python
* Flask
* Pandas
* HTML, CSS (for frontend templates)
* Bootstrap (for styling, if included)

**Directory Structure**

project\_root/

│── uploads/ # Folder for uploaded files

│── processed/ # Folder for processed files

│── static/results/ # Folder for storing result files

│── templates/ # HTML templates for rendering views

│── app.py # Main Flask application

│── requirements.txt # Dependencies

**Installation and Setup**

1. Clone the repository:
2. git clone <repository-url>
3. cd project\_root
4. Create a virtual environment and install dependencies:
5. python -m venv venv
6. source venv/bin/activate # On Windows use: venv\Scripts\activate
7. pip install -r requirements.txt
8. Run the Flask application:
9. python app.py

**API Endpoints**

**Home Page**

**Endpoint:** /

* **Methods:** GET, POST
* **Description:** Handles file uploads for item classification.
* **Response:** Redirects to /results upon successful processing.

**Results Page**

**Endpoint:** /results

* **Methods:** GET
* **Description:** Displays categorized report files in the results folder.

**Download Report**

**Endpoint:** /download/<filename>

* **Methods:** GET
* **Description:** Downloads a processed report file.

**Pharmacy Report Processing**

**Endpoint:** /pharmacy\_report

* **Methods:** GET, POST
* **Description:** Uploads, processes, and summarizes pharmacy data based on selected file type (sales, advance, credit).

**Data Processing Flow**

1. Users upload item\_file.xlsx and doctor\_list\_file.xlsx.
2. The system reshapes the doctor list, aggregates item quantities, and merges data.
3. Department-wise summaries are created and saved.
4. Remaining items not classified are identified and saved separately.
5. Pharmacy reports categorize data into summarized and excluded items.

**File Naming Convention**

* Classified reports: Classified\_YYYYMMDD.xlsx
* Remaining items: Remaining\_Items\_YYYYMMDD.xlsx
* Pharmacy reports:
  + Summarized: Summarized\_<type>\_YYYY-MM-DD.xlsx
  + Excluded: Excluded\_<type>\_YYYY-MM-DD.xlsx

**Error Handling**

* Invalid or missing files trigger flash messages.
* Missing required columns in uploaded files result in an error message.
* If no excluded items are found, an appropriate message is saved instead of an empty file.

**Future Improvements**

* Implement authentication for file uploads.
* Add frontend enhancements with better UI/UX.
* Improve file validation and handling of edge cases.

**License**

This project is licensed under the MIT License.