**Subject :** **Software Development and Applications**

**Title :** Smart Data Organization Model for Language, Gender, and Age Segmentation

**Prof. Lukasz Rojek**

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Name** | **Matriculation No.** |
| **1** | **Abhishek Yashwant Malwadkar** | **100001234** |

**1.Purpose :**

The primary purpose of this application is to **efficiently sort and categorize demographic data** provided in a dataset on the basis of **language, gender, and age** of the global population. The model is designed to **simplify the data sorting process** by automatically organizing the input dataset into separate, well-structured data sheets for each category. This structured segregation will enhance **data accessibility, usability, and analysis** for further research, reporting, and decision-making processes.

**2. Implementation and structure :**

**2.1 Implementation**

The proposed system is implemented in the following steps:

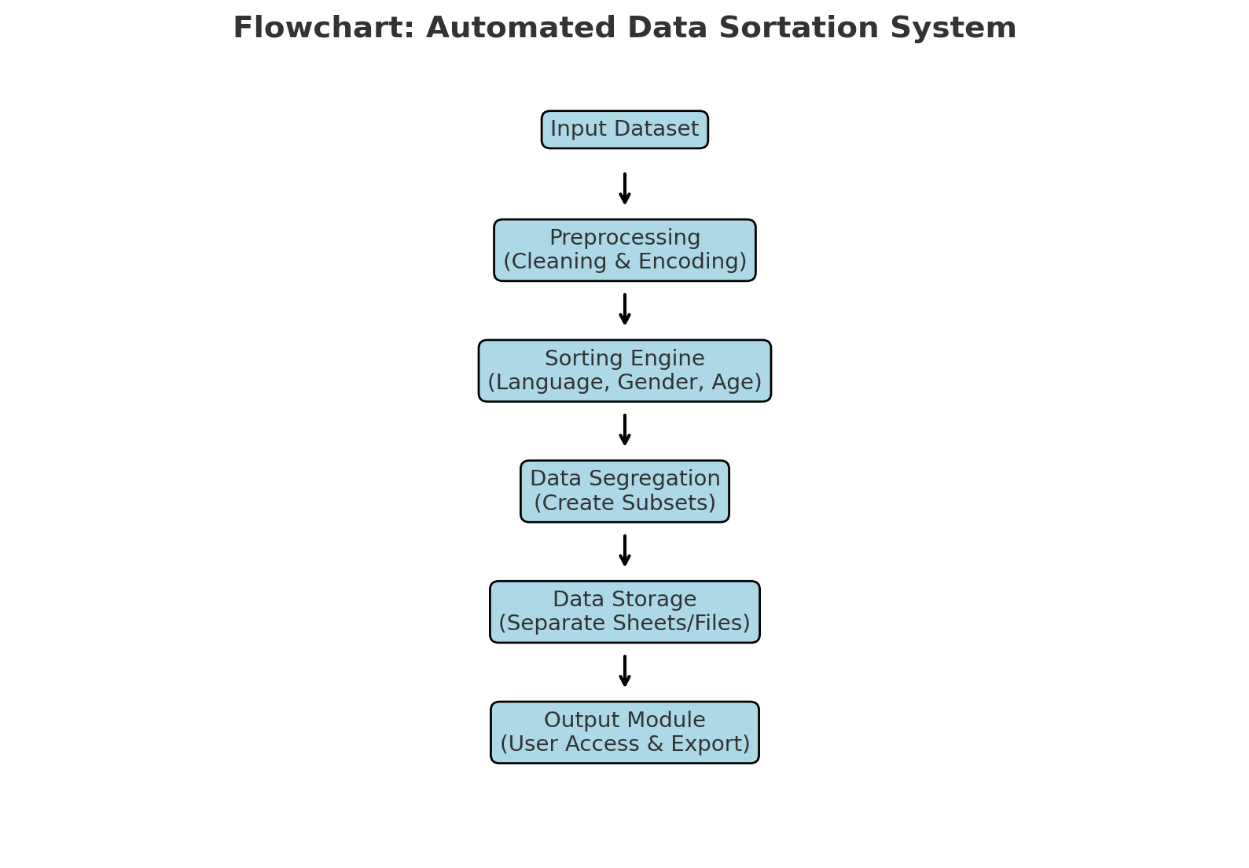
1. **Data Collection**
   * Import the global population dataset containing attributes such as **Name, Age, Gender, and Language**.
   * Ensure data consistency and completeness by preprocessing (removing duplicates, handling missing values, and standardizing formats).
2. **Data Preprocessing**
   * Normalize age values (e.g., convert to integer format).
   * Encode categorical data such as **Gender** and **Language** if needed for computational handling.
   * Validate the dataset for errors before sorting.
3. **Sorting Algorithm**
   * Implement sorting and filtering mechanisms based on three attributes:
     + **Language-based Sortation** → Groups data by spoken/written language.
     + **Gender-based Sortation** → Separates records into Male, Female, and Other categories.
     + **Age-based Sortation** → Organizes individuals into age brackets (e.g., 0–12, 13–19, 20–35, 36–60, 60+).
4. **Data Segregation and Storage**
   * Create **separate data sheets/files** for each sorting criterion.
   * Export sorted data into structured formats such as **CSV, Excel**  for easy retrieval and analysis.
5. **Output and Verification**
   * Verify correctness of data in each sorted category.
   * Ensure no data loss during splitting.
   * Provide user-friendly access to sorted data sheets for further processing.

**2.2 System Structure**

The structure of the proposed application consists of the following modules:

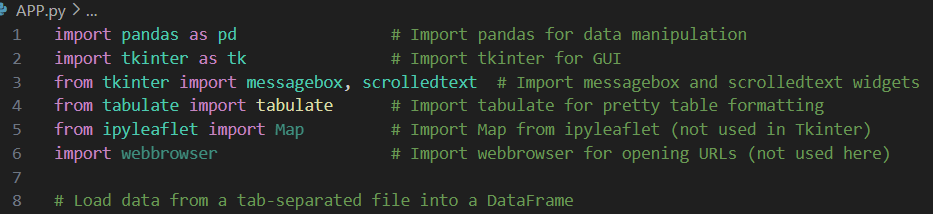
1. **Input Module**
   * Accepts dataset from external sources (Excel, CSV, Database).
   * Performs validation checks before processing.
2. **Preprocessing Module**
   * Cleans, formats, and encodes data.
   * Handles missing values and standardizes input.
3. **Sorting Engine**
   * Core logic that sorts and categorizes data into subsets based on **Language, Gender, and Age**.
4. **Data Storage Module**
   * Stores sorted results into **separate sheets/files**.
   * Ensures organized storage for quick access and reusability.
5. **Output Module**
   * Presents sorted datasets to the user.
   * Allows export in CSV format .

**2.3 Flow chart**



3 . Steps to run software:

1. **Download the Project Files**
   * Extract the contents of the provided **Final Project.zip** file.
2. **Open in Visual Studio**
   * Launch **Visual Studio** on your system.
   * Open the extracted project folder inside Visual Studio.
3. **Make sure all required packages are installed .**



1. **Verify Folder Structure**
   * Ensure that the folder structure appears exactly as shown in the reference image provided in the documentation. **Languages.txt** will be your data set.
   * Confirm that **all required files** are present within the same project directory for smooth execution.

A screenshot of a black screen

AI-generated content may be incorrect.

1. **Open the Main Application File**
   * Locate the file named **APP.py** in the project folder.
   * Double-click on **APP.py** to open it in the Visual Studio editor.
   * This file contains the interface logic and will serve as the entry point of the software.
2. **Run the Application**
   * Click on the **Run** button in Visual Studio to start execution.
3. **Verify Successful Launch**
   * Once executed, the **software interface window** will appear on the screen, as illustrated in the image provided in the documentation.

A screenshot of a computer screen

AI-generated content may be incorrect.

1. Next insert the data for eg->

**Gender** = MALE

**Age** = 33

**Country** = RUSSIA

**Language** = RUSSIAN

1. Click on Filter and Show Output.
2. Next step is to open the file named **map.ipynb.** This file will show the output on the map indicating the region.

A screenshot of a computer

AI-generated content may be incorrect.