For our project, each of the three data collection systems will have same requirements.

Here's an outline:

1. Second-hand Car Valuation

- **Web Scraping**: We will use Python libraries such as BeautifulSoup and Selenium to scrape data from websites like sahibinden.com and arabam.com.
- **Data Structure**: For each car listing, we'll need to extract attributes like brand, model, year, mileage, price, city, and crash report.
- **Saving Data**: We will store the data in a CSV file with the following columns like :

  Brand, Model, Year, Mileage, Price, City, Crash Report

- **Frequency of Data Collection**: Depending on how often these websites update their listings, we might want to run the scraping script periodically but this is optinal.

2. Football Match Result Prediction

- **API or Web Scraping**: To gather football match results, we could use football data APIs like football-data.org or scrape websites that store match results (e.g., soccerstats.com, flashscore.com ,maçkolik.com).
- **Data Structure**: For each match, the data would include:

  Date, Home Team, Away Team, Score, League

- **Leagues**: Focus on the top leagues from Turkey, England, Spain, Italy, Germany, and France.
- **Historical Data**: Ensure the collection goes back to the year 2000, which might require pulling data from archives or specialized sports databases we will figure it out in the road.

3. Voice Command Database Creation

- **Audio Recording**: We will use Python libraries like sounddevice or pyaudio to record voice commands.
- **Data Structure**: The dataset should consist of:

  Command, Language, Speaker ID, Audio File Path

- **Diversity**: Ensure to collect samples for 50 commands, with recordings in both English and Turkish, and from at least 10 different individuals for each command,as we said before this little details might change .
- **Storage**: Save audio files in a specific format (e.g., .wav) and reference their paths in the CSV.

System Requirements for the Data Collection Project

1. Second-hand Car Valuation System

Hardware Requirements

- **Computer/Server**: A system capable of running Python scripts, with at least 8 GB of RAM and 2 GHz processor. This will support web scraping and data processing tasks.
- **Storage**: At least 10 GB of free storage for storing the scraped data in CSV format. Storage needs may increase depending on the volume of data.
- **Network Connectivity**: A stable internet connection for accessing car listing websites (e.g., sahibinden.com, dod.com).
- **For these requirements we will use a desktop pc that at least satisfies these requirements.**

Software Requirements

- **Python 3.x**: The project will be developed using Python, so Python 3.x must be installed.
- **Libraries**:
  - BeautifulSoup, Selenium: For web scraping tasks.
  - pandas: For processing and storing data in CSV format.
  - requests: To handle HTTP requests for scraping.
  - CSV: Native Python CSV module for reading/writing CSV files.

Functional Requirements

- **Scraping Automation**: The software should be able to scrape car listings from websites automatically, pulling details like brand, model, year, mileage, price, city, and crash report.
- **Data Parsing**: The software should parse the scraped data and store it in a structured CSV format.
- **Error Handling**: The system should handle network errors, CAPTCHA challenges, and website structure changes without breaking.
- **Frequency**: The scraping process should be executed at regular intervals, and must allow for dynamic website updates.

Feasibility and Justification

- **Web Scraping**: Since car listing websites generally offer data in a structured format (HTML), it is feasible to scrape them with Python. The system will need to handle large datasets efficiently, making Python's data manipulation libraries a good fit.
- **Data Storage**: Storing the scraped data in CSV format is suitable as CSV files are lightweight and easy to manipulate for valuation purposes.

---

2. Football Match Result Prediction System

Hardware Requirements

- As we said before an old desktop could do the work.
- **Computer/Server**: At least 8 GB RAM and 2 GHz processor to handle data fetching from multiple sources and process historical data.

- **Storage**: Minimum of 10 GB storage for saving the historical match data in CSV format.
- **Network Connectivity**: High-speed internet connection for fetching football data through APIs or web scraping.

Software Requirements

- **Python 3.x**: Required for development and execution.
- **Libraries**:
  - requests: For accessing football match data APIs.
  - pandas: For data processing and storage in CSV format.
  - BeautifulSoup or Selenium: For scraping data from websites.
  - **APIs** (optional): football-data.org for accessing structured football data if scraping isn't available.

Functional Requirements

- **Data Collection**: The system should be able to fetch football match results from the year 2000 onward for leagues in Turkey, England, Spain, Italy, Germany, and France.
- **Data Storage**: Store the match data in a CSV format, with fields for date, home team, away team, score, and league.
- **Historical Data**: Support retrieval of historical data from the year 2000 onward, covering six major European leagues.
- **Data Processing**: Allow easy querying and manipulation of data for statistical analysis and prediction model training.

Feasibility and Justification

- **Data Availability**: Football match data is publicly available either through APIs or accessible websites, making it feasible to collect this data programmatically .
- **Processing Capability**: Python is well-suited to handle this kind of structured, historical data and can efficiently store it in CSV files for future use in predictive models.

---

3. Voice Command Database Creation System

Hardware Requirements

- We probably use the same old desktop.
- **Microphone**: Good quality microphones to record clear audio in both English and Turkish.
- **Computer**: A system with at least 4 GB RAM and basic processing capabilities for recording and saving audio files.
- **Storage**: Minimum 10 GB free space for storing the recorded audio files and metadata in CSV format. This will increase based on the number of speakers and command repetitions.

Software Requirements

- **Python 3.x**: Required for development.
- **Libraries**:
  - sounddevice or pyaudio: To capture voice recordings.
  - pandas: For organizing metadata in CSV format.
  - wave: For saving and handling audio files in .wav format.

Functional Requirements

- **Voice Recording**: The system must capture and store voice commands from at least 10 different individuals in both English and Turkish.
- **Command Variations**: For each of the 50 commands, voice recordings must be captured for both languages, ensuring diversity of pronunciation and tone.
- **Metadata**: The system must store metadata (command, speaker ID, language, and audio file path) in a CSV file.
- **Audio Quality Control**: Ensure all recordings meet a minimum quality threshold (e.g., clarity, absence of noise).

Feasibility and Justification

- **Voice Data Collection**: Python libraries like pyaudio and sounddevice allow for high-quality audio recording, making this system feasible. Storing audio files and metadata in CSV format is straightforward.
- **Data Diversity**: Ensuring recordings from different individuals ensures the robustness of the voice command dataset, which is essential for creating a valuable training dataset.

Conclusion:

Each system has well-defined technical requirements and functional constraints to ensure the data collection process is efficient and streamlined. By using a desktop environment for web scraping, we can manage larger volumes of data more effectively, ensuring performance stability and consistency throughout the process. Additionally, utilizing GitHub for version control will allow the team to track changes, collaborate efficiently, and maintain full transparency across all stages of development. Python’s extensive ecosystem of libraries makes it an ideal choice for handling diverse tasks like web scraping, data collection, voice recording, and CSV file management, ensuring that all three systems are practical, scalable, and fully aligned with the project's objectives.