SYSTEM REQUIREMENT REPORT

Team Members

NUMBER	*NAME*
211805077	Mustafa Kağan Bal
221805035	Deniz Arda Çelik
231805084	Dilara Akkuzu
191805073	Arda Pala
211805073	Emir Özcan
211805034	Melike Çelik
211805108	Muhammet Korkmaz
211805027	Nazan Sonal
171805027	Özgür Taş
211805009	Utku Arda Kurtay
211805057	Utku Deniz Uğuş
221805097	Ilia Bohloulvand

Second-Hand Car Valuation System

1.SYSTEM REQUIREMENTS

Hardware Requirements	• **Memory (RAM):** The system must run with at least 8 GB of RAM to handle large datasets and multiple operations simultaneously. • **Storage:** A minimum of 100 GB of free disk space is required to store car data and scraped records in CSV format. • **Processor**:A Multi-core processor .
Software Requirements	• **Operating System:** The system must be compatible with Windows 10, macOS, and Linux. • **Programming Language:** The system must be developed using **Python 3.9**+. • **Development Environment:** The preferred integrated development environment (IDE) is **Spyder**. **Libraries:** • **BeautifulSoup:** For web scraping and extracting data from HTML pages. • **Requests:** For handling HTTP requests to access car listing websites. • **CSV Module:** For saving and managing data in CSV format. • **PyQt5:** Used for developing the user interface (UI). • **Constraint:** Only open-source Python libraries should be used to avoid licensing issues.
Security Requirements	• **Data Privacy:** The system must not collect any personal or sensitive data during the scraping process. • **Compliance:** The system must comply with data privacy laws such as **GDPR**, especially if scraping from European sources. • **Constraint:** The system must not engage in activities that violate the target websites' terms of use or data protection laws.

2.USER REQUIREMENTS

Data Collection	• The system shall allow users to collect car data from specified websites (sahibinden.com and dod.com) by initiating a web scraping process with a single click. • Users shall be able to define the specific parameters (brand, model, price range) for data collection before scraping.
Data Validation	• The system shall validate the collected data to ensure accuracy and completeness, including checks for realistic values in mileage and price.
Data Storage	• The system shall save all collected data in a structured CSV format, organizing it by relevant attributes such as brand, model, year, mileage, price, city, and crash report. • Users shall be able to export the stored data to a CSV file for external analysis or reporting.
User Interface	• The system shall provide a user-friendly graphical user interface (GUI) developed using PyQt5, allowing for easy navigation and interaction.
Data Retrieval and Analysis	• The system shall provide users with the capability to filter and search through the collected data based on various criteria (brand, model, price). • Users shall be able to generate basic statistical reports (average price by brand, mileage distribution) based on the collected data.

3.FUNCTIONAL REQUIREMENTS

Data Input	• The system must scrape data from sites like sahibinden.com and dod.com, collecting car details like brand, model, year, mileage, price, city, and crash report.
Data Validation	• The system must validate the correctness and completeness of collected data (ensuring no fields are missing).
Data Storage	• The system must save the collected data in CSV format.
Data Retrieval	• Users must be able to retrieve data based on specific criteria, such as cars from a particular brand or within a price range.
User Interface	• The system must provide a user-friendly interface for users to specify the data they want to collect.

4.NON-FUNCTIONAL REQUIREMENTS

Performance	• The system should be able to scrape and store large amounts of data efficiently without causing significant delays.
Usability	• The user interface should be intuitive and easy to use for non-technical users. • Error messages should be clear and informative to help users fix issues without external help.
Scalability	• The system should be able to handle growing datasets without significant performance degradation.
Reliability	• The system should ensure data integrity and avoid data loss during scraping, storage, or retrieval. • The system should be able to recover from any unexpected crashes or interruptions.

5.DOMAIN REQUIREMENTS

Data Sources	• The system must gather car data only from reputable and legal sources such as sahibinden.com and dod.com. • The football match result data must be collected only from trustworthy sources or official league websites.
Feature Relevance	• For the second-hand car valuation system, the collected features such as brand, model, year, mileage, and price should be relevant to evaluating the car's market value.
Compliance	• The system must comply with any legal requirements for scraping data, including avoiding overloading website servers.

Football Match Result Prediction System

1.SYSTEM REQUIREMENTS

Hardware Requirements	• **Memory (RAM):** The system must run with at least 8 GB of RAM to handle large datasets and multiple operations simultaneously. • **Storage:** A minimum of 100 GB of free disk space is required to store data and scraped records in CSV format. • **Processor**: A Multi-core processor .
Software Requirements	• **Operating System:** The system must be compatible with Windows 10, macOS, and Linux. • **Programming Language:** The system must be developed using **Python 3.9**+. • **Development Environment:** The preferred integrated development environment (IDE) is **Spyder**. **Libraries:** • **BeautifulSoup:** For web scraping and extracting match data from HTML pages. • **Requests:** For handling HTTP requests to access football match data websites. • **CSV Module**: For saving and managing data in CSV format. • **PyQt5:** Used for developing the user interface (UI). • **Constraint:** Only open-source Python libraries should be used to avoid licensing issues.
Security Requirements	• **Data Privacy:** The system must not collect any personal or sensitive data during the scraping process. • **Compliance:** The system must comply with data privacy laws such as GDPR, especially if scraping from European sources.

2.USER REQUIREMENTS

Data Collection	• The system shall allow users to collect match result data from specified leagues (Turkey, England, Spain, Italy, Germany, and France) by initiating a web scraping process with a single click. • Users shall be able to define specific parameters (date range, home team, away team, league) for data collection before scraping.
Data Validation	• The system shall validate the collected data to ensure accuracy and completeness, including checks for realistic values in match scores and dates. • Users shall be notified if any errors or discrepancies are detected in the scraped data.
Data Storage	• The system shall save all collected match data in a structured CSV format, organizing it by relevant attributes such as date, home team, away team, score, and league. • Users shall be able to export the stored data to a CSV file for external analysis or reporting.
User Interface	• The system shall provide a user-friendly graphical user interface (GUI) developed using PyQt5, allowing for easy navigation and interaction. • The interface shall include buttons for starting the data scraping process, accessing stored data, and exporting data in CSV format.
Data Retrieval and Analysis	• The system shall provide users with the capability to filter and search through the collected match data based on various criteria (league, date, teams). • Users shall be able to generate basic statistical reports (win/loss records, average scores) based on the collected data.
Constraints	• The system shall operate within the legal boundaries defined by the target websites' terms of service and comply with data protection regulations (GDPR). • The system shall only scrape data from approved and reputable websites to ensure data integrity and accuracy.

3.FUNCTIONAL REQUIREMENTS

Data Input	• The system must scrape data from specified sports websites and APIs, collecting match details such as date, home team, away team, score, and league.
Data Validation	• The system must validate the correctness and completeness of the collected data, ensuring no fields are missing and that scores and dates are realistic.
Data Storage	• The system must save the collected match data in CSV format, organizing it by relevant attributes such as date, home team, away team, score, and league.
Data Retrieval	• Users must be able to retrieve data based on specific criteria, such as matches from a particular league, date range, or specific teams.
User Interface	• The system must provide a user-friendly interface that allows users to specify the parameters for data collection, such as date range and leagues, and to view and analyze the collected match data.

4.NON-FUNCTIONAL REQUIREMENTS

Performance	• The system should be able to scrape and store large amounts of data efficiently without causing significant delays.
Usability	• The user interface should be intuitive and easy to navigate for non-technical users, allowing for a smooth user experience. • Error messages should be clear and informative, enabling users to resolve issues without needing external assistance.
Scalability	• The system should handle growing datasets seamlessly, ensuring performance remains stable as data volume increases.
Reliability	• The system should ensure data integrity and avoid data loss during the scraping, storage, or retrieval processes. • It should implement mechanisms to recover from unexpected crashes or interruptions, preserving the collected data.

5.DOMAIN REQUIREMENTS

Data Sources	• The system must gather match result data only from reputable and legal sources, such as official league websites and trusted sports news outlets. • Data scraping should avoid sites that are known to have restrictions against automated data collection.
Feature Relevance	• The collected features, such as date, home team, away team, score, and league, should be relevant for analyzing match outcomes and performance trends.
Compliance	• The system must comply with all legal requirements for scraping data, including adherence to the target websites' terms of service and avoiding overloading their servers. • The system must respect any copyright and data usage policies associated with the collected match data to ensure ethical use.

Voice Command Database Creation System

1.SYSTEM REQUIREMENTS

Hardware Requirements	• **Memory (RAM):** The system must run with at least 8 GB of RAM to handle large datasets and multiple operations simultaneously. • **Storage:** Audio files, and associated metadata in CSV format. • **Processor:** A multi-core processor is required to ensure smooth processing of audio files and real-time analysis.
Software Requirements	• **Operating System:** The system must be compatible with Windows 10, macOS, and Linux. • **Programming Language:** The system must be developed using **Python 3.9**+. • **Development Environment:** The preferred integrated development environment (IDE) is **Spyder**. • **Libraries:** • **CSV Module:** For saving and managing data in CSV format. • **PyQt5:** Used for developing the user interface (UI). • **Constraint:** Only open-source Python libraries should be used to avoid licensing issues.
Security Requirements	• **Data Privacy:** The system must not collect any personal or sensitive data during the voice recording process. Recorded commands must be anonymized. • **Compliance:** The system must comply with data privacy laws such as **GDPR**, especially if collecting data from users in Europe. • **Constraint:** The system must not engage in activities that violate data protection laws or the terms of use of any third-party services it may interact with.

2 . USER REQUIREMENTS

Data Collection	• The system shall allow users to collect voice data for at least 50 different command in both English and Turkish by initiating a recording process with a single click.
Data Validation	• The system shall validate the collected voice recordings to ensure clarity and completeness, including checks for background noise and volume levels. • Users shall be notified if any recordings are deemed unsuitable or if there are discrepancies in the expected number of recordings per command.
Data Storage	• The system shall save all collected voice data in a structured format, organizing it by relevant attributes such as command, participant ID, language, and recording quality. • Users shall be able to export the stored data to a CSV file or audio format for external analysis or further processing.
User Interface	• The system shall provide a user-friendly graphical user interface (GUI) developed using PyQt5, allowing for easy navigation and interaction. • The interface shall include buttons for starting the recording process, accessing stored voice data, and exporting data in the desired format.
Data Retrieval and Analysis	• The system shall provide users with the capability to filter and search through the collected voice data based on various criteria (command, language, participant). • Users shall be able to generate basic statistical reports (number of recordings per command, participant contribution) based on the collected data.
Constraints	• The system shall operate within the legal boundaries defined by data protection regulations like GDPR and ensure that participants give informed consent before recording their voice. • The system shall only collect voice data from participants who are aware of the purpose of the data collection to ensure ethical standards are maintained.

3 . FUNCTIONAL REQUIREMENTS

Data Input	• The system must collect voice data for 50 different commands in both English and Turkish from at least 10 different individuals for each command.
Data Validation	• The system must validate the collected voice data for clarity and completeness, ensuring that each command is recorded with adequate audio quality and no missing entries.
Data Storage	• The system must save the collected voice recordings and associated metadata (command text, language) in a structured CSV format for easy access and analysis.
Data Retrieval	• Users must be able to retrieve and filter voice command data based on specific criteria, such as command text, language, or user demographics.
User Interface	• The system must provide a user-friendly interface that allows users to specify the commands they want to collect, manage recordings, and access stored data easily.

4 . NON FUNCTIONAL REQUIREMENTS

Performance	• The system should be able to record and process voice commands efficiently without causing significant delays.
Usability	• The user interface should be intuitive and easy to use for non-technical users, allowing them to easily record and manage voice commands. • Error messages should be clear and informative to help users troubleshoot issues without needing external assistance.
Scalability	• The system should be able to handle increasing volumes of voice command recordings without significant performance degradation.
Reliability	• The system should be capable of recovering from unexpected crashes or interruptions, ensuring minimal disruption to users.

5 . DOMAIN REQUIREMENTS

Data Sources	• The system must gather voice command data only from reputable sources and platforms that allow for the collection of voice samples. • The voice commands must be collected from diverse and legally compliant sources to ensure a wide representation of accents and dialects.
Feature Relevance	• Voice samples must accurately represent the intended commands, ensuring that they cover various contexts and usages.
Compliance	• The system must comply with all legal requirements for data collection, including obtaining explicit user consent before recording voice samples to adhere to data protection regulations like GDPR. • The system must ensure that all voice data is handled securely and responsibly, protecting user privacy and preventing unauthorized access.

Feasibility and Justification

Second-Hand Car Valuation System	Football Match Result Prediction System	Voice Command Database Creaction System
Feasibility Web scraping tools and libraries BeautifulSoup make data extraction from websites technically feasible. Storing data in CSV format is straightforward, supported by Python's built-in libraries, making data handling highly feasible. The requirement to gather key attributes like brand, model, year, mileage, price, and crash report is realistic, as these data points are readily available on car listing sites.	Feasibility Gathering data from popular sports APIs or public databases of match results from major leagues is highly feasible using Python libraries like requests or APIs like football-data.org. Stored efficiently in CSV format and processed using Python's data manipulation tools.	Feasibility Voice recording can be handled using Python libraries like PyDub or SpeechRecognition, and storing voice data alongside metadata (speaker, language, command type) is feasible with a well-structured directory system and CSV for metadata. Collecting data from 10 individuals for 50 commands is feasible within the project's scope, considering the widespread availability of recording devices and willing participants.
Justification The need for automated car valuation systems is high, given the dynamic pricing in the second-hand car market. Storing data in CSV allows for ease of access and compatibility with analytical tools, providing flexibility for future use in machine learning models or pricing trend analysis.	Justification With the growing interest in predictive analytics in sports, this project can provide valuable insights to stakeholders, including sports analysts and bettors. The system will be flexible and scalable, allowing the addition of more leagues or seasons, which justifies its potential long-term usability.	Justification The demand for voice recognition technology is increasing, and creating a diverse, multilingual voice command database will contribute to advancements in voice-activated systems. Storing voice commands and metadata in a structured format ensures that the database will be usable for various machine learning models or research.