EXERCISE

1. Construct a Software Requirement Specifications (SRS) document for a Software System of your choice using LaTeX. Attach the printout of the .tex file and .pdf file.

```
\documentclass[12pt]{article}
\usepackage{graphicx}
\usepackage{imakeidx}
\usepackage{fancyhdr}
\usepackage[a4paper,width=150mm,top=25mm,bottom=25mm]{geometry}
\usepackage{times}
\usepackage{anyfontsize}
\usepackage{t1enc}
\usepackage{hyperref}
\usepackage{blindtext}
\usepackage{setspace}

\title{
\begin{flushleft}
  \vspace{1cm}
  \includegraphics[width=4.5cm, height=5cm]{botani_scan_logo-main.png}
\end{flushleft}

\begin{flushright}
  \vspace{-6cm}
  {
{\fontfamily{ptm}\fontsize{32}{40}\selectfont \textbf{Software}\\}
   {\fontfamily{ptm}\fontsize{32}{40}\selectfont \textbf{Requirements}\\}
{\fontfamily{ptm}\fontsize{32}{40}\selectfont \textbf{Specification}\\}
  \vspace{1cm}
  \Huge \textbf{For}\\
```

```
  \vspace{1cm}
{\fontfamily{ptm}\fontsize{32}{40}\selectfont \textbf{Botani Scan}\\}
  }
  {\\ \vspace{1cm}\textbf{Version 1.0}}
    {\\ \vspace{1cm} \textbf{Prepared by}\\
\textbf{{\Large
Izma Shafqat (SE-21003)\\
Aqsa Zaib (SE-21013) \\
Inshara Iqbal (SE-21018) \\
Mahnoor Iqbal (SE-21027) \\
Sahil Krishna (SE-21046) \\
\\ \vspace{1cm}
Internal Advisor: Ms. Sana Fatima}
}}
\end{flushright}}
\date{\hfill \textbf{03/11/2023}}
\makeindex[columns=1]
\pagestyle{fancy}
\fancyhf{}
\fancyfoot[C]{---------------------------------------------- Page \thepage ----------------------------------
--------------}
% \renewcommand{\headrulewidth}{0pt}
\fancyhead[C]{SRS For Botani Scan - Plant Disease Detection}
% \centering
\renewcommand*\contentsname{\centering{Table Of Contents}}
\urlstyle{same}
\begin{document}
\maketitle
\newpage
```

\tableofcontents

\newpage

\section{\underline{INTRODUCTION}}

\subsection{PURPOSE OF REQUIREMENT DOCUMENT}

\index{B!\\Botani Scan\hfill page}

This Software Requirements Specification provides a complete description of all the functions and specifications of the BotaniScan. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate.

\\ \\ The purpose of the BotaniScan software is to provide a powerful and user-friendly tool for the real-time identification of plants, the detection of diseases in plant leaves, and the estimation of leaf water levels. This software aims to cater to the needs of both hobbyist gardeners and professional botanists, offering valuable insights into plant health and disease management through the utilization of computer vision and machine learning techniques.

\subsection{PROJECT SCOPE}

\index{S!\\Scope of Project\hfill page}

\subsubsection{Description}

BotaniScan is an AI based application that can identify plant by their leaves, detect any diseases, and estimate the leaf water level using computer vision and machine learning techniques. This project will serve as a valuable tool for both hobbyist gardeners and professional botanists by providing real-time information about plant, diseases and health.

\subsubsection{Benefits}

\index{B!\\Benefits\hfill page}

BotaniScan offers a powerful combination of AI technology and practical features, making it an indispensable tool for plant enthusiasts and professionals alike, improving plant care, research, and maintenance. With an intuitive user-friendly interface, it offers accurate and real time information about plants, their health and common diseases, empowering both amateur gardeners and professional botanists to effectively care for their plants and ensure optimal plant health. Additionally, BotaniScan's commitment to user guide and comprehensive documentation ensures easy adoption and understanding for plant care and research.

\subsubsection{Corporate Goals}

\index{C!\\Corporate Goals\hfill page}

BotaniScan have core corporate goals encompass providing a user-friendly, accurate, and responsive system for indivduals and professional botanists. We strive to offer real-time, reliable

information about plant names through advanced machine-learning techniques. Disease detection, precise leaf water level estimation, and an extensive plant dataset are integral to our commitment to plant care. Ensuring compatibility with diverse devices, scalability, and robust user guide are key components of our mission. Our dedication to excellence is underpinned by a relentless pursuit of accuracy, reducing the risk of misdiagnosis and enhancing plant care practices.

\subsection{DEFINITIONS, ACRONYMS \& ABBREVIATIONS}

\index{D!\\DEFINITIONS, ACRONYMS \& ABBREVIATIONS\hfill page}

\index{K!\\Kaggle\hfill page}

\textbf{\large Kaggle} \\ \\

Kaggle is a well-known online platform for data science competitions, machine learning, and data analytics. It provides datasets, tools, and a community of data scientists and machine learning practitioners to collaborate, compete, and share insights. Kaggle hosts a wide range of data-related challenges, from predictive modeling to computer vision tasks.

\index{O!\\OpenCV\hfill page}

\\ \\

\textbf{\large OpenCV} \\ \\

OpenCV stands for Open Source Computer Vision. \\ \\

OpenCV Library, is an open-source computer vision and image processing library. It offers a wide range of tools and functions for tasks like image manipulation, object recognition, and machine learning. OpenCV is widely used in computer vision applications, robotics, and real-time image and video processing.

\\ \\

\textbf{\large YOLO } \\ \\

YOLO stands for You Only Look Once. \\ \\

YOLO is an object detection algorithm that is designed to efficiently detect and locate objects in images and videos. What sets YOLO apart is its speed and accuracy. It can process images in real-time, making it suitable for applications like self-driving cars, surveillance, and more.

\index{T!\\TensorFlow\hfill page}

\\ \\

\textbf{\large TensorFlow} \\ \\

TensorFlow is an open-source machine learning framework developed by Google. It is widely used for building and training deep learning models, including neural networks. TensorFlow

provides a comprehensive ecosystem for machine learning, offering tools, libraries, and community support.

\index{C!\\CNN\hfill page}

\\ \\

\textbf{\large CNN } \\ \\

CNN stands for Convolutional Neural Network. \\ \\

CNN is a type of deep neural network commonly used in image and video analysis. CNNs are designed to automatically and adaptively learn patterns from data, making them well-suited for tasks like image recognition, object detection, and image classification.

\\ \\

\textbf{\large Google Colab} \\ \\

Google Colab is a cloud-based service provided by Google that allows users to run Python code in a Jupyter Notebook environment. It provides free access to GPUs, which makes it a popular choice for machine learning and data analysis tasks, as it allows users to execute code and run machine learning models on Google's infrastructure.

\\ \\

\textbf{\large Google Cloud} \\ \\

Google Cloud is a cloud computing platform offered by Google. It provides a variety of cloud-based services, including computing, storage, databases, machine learning, and data analytics. Google Cloud offers tools and infrastructure to develop and deploy applications in a scalable and reliable manner.

\\ \\

\textbf{\large React JS} \\ \\

React JS, often referred to as React, is an open-source JavaScript library for building user interfaces. Developed by Facebook, React is commonly used for creating dynamic and interactive web applications. It simplifies the process of building UI components and efficiently updating them when data changes.

\index{F!\\Flask\hfill page}

\\ \\

\textbf{\large Flask} \\ \\

Flask is a micro web framework for Python. It is designed to be lightweight and easy to use, making it an excellent choice for building web applications and APIs. Flask provides the essential features needed for web development without imposing too many constraints.

\index{P!\\Python\hfill page}

\\ \\

\textbf{\large Python} \\ \\

Python is a high-level, interpreted programming language known for its simplicity and readability. It is versatile and widely used in various domains, including web development, data science, machine learning, and automation. Python's extensive library ecosystem and community support make it a popular choice for a wide range of applications.

\\ \\

\textbf{\large RESTful API} \\ \\

RESTful API stands for Representational State Transferful Application Programming Interface \\ \\

A RESTful API is an architectural style for designing networked applications. It is based on the principles of Representational State Transfer (REST), which uses standard HTTP methods to perform operations on resources. RESTful APIs are commonly used for building web services that can be easily consumed by clients, allowing for interoperability and scalability.

\\

\subsection{REFERENCES}

\index{R!\\References\hfill page}

\begin{itemize}

\item IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

\item \url{https://www.scribd.com/doc/9138468/Software-Requirement-Specification-Srs-Midtems}

\item \url{http://www.processimpact.com/process_assets/srs_template.doc}

\end{itemize}

\index{O!\\Overview\hfill page}

\subsection{OVERVIEW}

The rest of this SRS is organized as follows:

\begin{itemize}

\item Section 2 and 3 gives an overall description of the software. It gives what level of proficiency is expected of the user, some general constraints while making the software and some assumptions and dependencies that are assumed.

\item Section 4 contains most important features presented with detailed description, and requirements. It gives specific requirements which the software is expected to deliver. Functional requirements are given in this section. This section is written primarily for the developers and describes in technical terms the details of the functionality of the product and about safety and performance.

\end{itemize}

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

\section{\underline{GENERAL DESCRIPTION}}

\subsection{PRODUCT PERSPECTIVE}

\index{P!\\Product Perspective\hfill page}

BotaniScan, an AI-based application, enhances plant management for hobbyist gardeners and professional botanists. Through intuitive user interfaces and computer vision techniques, it identifies plant, detects diseases, and estimates leaf water levels. The system interfaces with the device's camera, external plant datasets, and machine learning models. It operates in the context of plant and garden management, assuming access to a reliable compatibility with various devices and operating systems. By providing real-time information about plants, BotaniScan serves as a valuable tool for plant enthusiasts and experts, contributing to more informed and efficient plant care.

\subsection{PRODUCT FUNCTION}

\index{P!\\Product Function\hfill page}

\subsubsection{Development Team}

This team is responsible for designing, developing, and maintaining the BotaniScan application. It comprises software developers, machine learning engineers, and computer vision experts who work on the core functionality, algorithms, and code base. They ensure the application's robustness and performance with various platforms

\subsubsection{Quality Assurance (QA) Team}

The QA team conducts thorough testing to identify and report any issues, bugs, or inconsistencies in the performance. They play a critical role in ensuring that BotaniScan provides accurate results and a seamless user experience.

\subsubsection{Machine Learning and AI Team}

\index{M!\\Machine Learning and AI Team\hfill page}

This team focuses on building and training machine-learning models for plant species classification and disease detection. They continually update and improve these models to enhance the accuracy of the application.

\subsubsection{Technical Support and Maintenance Team}

This team provides ongoing technical issues and troubleshooting.They also work on maintenance tasks, including updates, bug fixes, and optimizing the performance.

\subsection{USER CLASSES AND CHARACTERISTICS}

We will be having different classes for following users.

% \subsubsection{User}

% BotaniScan users encompass farmers, hobbyist gardeners, and common plant enthusiasts, benefiting from plant identification, disease detection, and water level care.

\subsubsection{For Individuals (Hobbyist Gardeners)}

BotaniScan is designed for hobbyist gardeners, regardless of age or gender, with varying levels of expertise. These users are motivated by their love for gardening, landscaping, and plants.

Their specific needs include plant identification and disease detection to maintain plant health. Usage frequency ranges from occasional to regular.

\\

\subsubsection{For Professionals (Botanists and Agricultural Experts)}

BotaniScan caters to professionals in botany and agriculture, facilitating research and precise disease management. These experts, with advanced degrees, use the app regularly, requiring advanced features to support their specialized work.

\subsection{GENERAL CONSTRAINTS}

\index{G!\\GENERAL CONSTRAINTS\hfill page}

\subsubsection{Hardware Limitations}

\index{H!\\Hardware Limitations\hfill page}

The performance and accuracy of the BotaniScan application may be influenced by the quality of the camera on the user's device. Higher-resolution cameras are likely to provide better leaf image input for analysis.

\subsubsection{Machine Learning Model Size}

The size of the dataset used for leaf identification and disease detection may affect the response time and performance of our application.

\subsection{ASSUMPTIONS AND DEPENDENCIES}

% \index{A!\\Assumptions\hfill page}

\subsubsection{Assumptions}

\begin{itemize}

\item \textbf{User-Provided Images:} The project assumes that users will provide clear and well-captured images of plant leaves. The accuracy of identification and disease detection is highly dependent on the quality of the input images. \\

\item \textbf{Legal Considerations:} The project assumes that there are no legal issues related to using images of plants(dataset), and there are no copyright violations regarding the dataset.

\end{itemize}

\subsubsection{Dependencies}

\index{D!\\Dependencies\hfill page}

\begin{itemize}

\item \textbf{Availability of Quality Datasets:} The project relies on comprehensive datasets of plant images and diseases. A dependency is that such datasets are available, and they need to be regularly updated to ensure the system's accuracy.

\item \textbf{Machine Learning Models:} The successful implementation of machine learning models for plant identification and disease detection depends on the availability of suitable algorithms and tools. Dependencies include the selection of the right machine-learning frameworks and libraries.

\item \textbf{Server Infrastructure:} To process and analyze images and deliver real-time results, the application depends on server infrastructure with sufficient computational power and storage capacity.

\end{itemize}

\section{\underline{SPECIFIC REQUIREMENT}}

\index{S!\\Specific Requirement\hfill page}

\subsection{FUNCTIONAL REQUIREMENT}

\index{F!\\FUNCTIONAL REQUIREMENT\hfill page}

\begin{itemize}

\item \textbf{Plant Identification:} The application should be able to identify plants by analyzing their leaves using computer vision and machine learning techniques.

\item \textbf{Disease Detection:} The application should be capable of detecting diseases in plant leaves.

\item \textbf{Leaf Water Level Estimation:} BotaniScan should provide an estimate of the leaf water level, which can be useful for assessing plant health and hydration.

\item \textbf{Real-time Information:} The system should offer real-time information about the identified plants, estimate their water levels and any detected diseases.

\item \textbf{Image Input:} Users should be able to input images of plant leaves through various means, such as uploading pictures from their devices or capturing images using the camera.

\item \textbf{Plant Dataset:} The system should have a comprehensive dataset of plant including images and information about plants and common diseases.

\item \textbf{Disease Dataset:} BotaniScan should maintain a dataset of common plant diseases.

\item \textbf{Image Processing:}

\index{I!\\Image Processing\hfill page}

Utilize computer vision techniques to process and analyze leaf images accurately.

\end{itemize}

\subsection{NON FUNCTIONAL REQUIREMENT}

\index{N!\\Non Functional Requirement\hfill page}

\subsubsection{Product Requirements}

\begin{itemize}

\item \textbf{Accuracy:} The system should provide accurate identification of plants and diseases with a better level of confidence.

\item \textbf{Response Time:} The application should have low latency in processing user inputs and returning results, ensuring a smooth and responsive user experience.

\item \textbf{Scalability:} System should be able to handle a large number of users and images simultaneously.

\index{U!\\Usability\hfill page}

\item \textbf{Usability:} The user interface should be intuitive, easy to navigate, user-friendly and should provide responsive user interface.

\item \textbf{Reliability:} The system should be robust and reliable, with minimal downtime or technical issues.

\item \textbf{Compatibility:} The web application should be compatible with various devices, including smartphones, tablets, and desktop computers, and across different operating systems.

\item \textbf{Documentation:} Comprehensive documentation for users and developers should be available, explaining how to use the application and its underlying technologies.

\end{itemize}

\subsubsection{Domain-Specific Requirements}

\begin{itemize}

\item \textbf{Botanical Dataset:} Provide an extensive dataset of plant, diseases and watering level information.

\item \textbf{Botanical Knowledge Base:} Incorporate botanical expertise to improve plant identification accuracy.

\end{itemize}

\subsubsection{External Requirements}

\begin{itemize}

\item \textbf{Ethical Considerations:}

\index{E!\\Ethical Considerations\hfill page}

The system should be designed and operated ethically, avoiding any harmful consequences related to plant diagnosis. Ethical considerations also include transparent data usage and user consent.

\end{itemize}

\subsection{EXTERNAL INTERFACE REQUIREMENTS}

\index{E!\\EXTERNAL INTERFACE REQUIREMENTS\hfill page}

\subsubsection{Hardware Interfaces}

\index{H!\\Hardware Interfaces\hfill page}

This section describes the hardware components or devices with which the Plant Disease Detection system needs to interact or be compatible. It includes details about the system's requirements for mobile phones, laptop and desktop computer with minimum 10 mega pixel camera.

\subsubsection{Software Interfaces}

\index{S!\\Software Interfaces\hfill page}

\begin{itemize}

\item \textbf{Kaggle} will serve as a resource for accessing plant-related datasets and machine learning competitions. It will help in training and improving the accuracy of the system's plant identification and disease detection models by providing valuable data and benchmarking opportunities.

\item \textbf{OpenCV} will be utilized to process and analyze plant images. It will enable the system to perform tasks like leaf recognition, disease detection, and image manipulation to enhance the accuracy of plant identification and health assessment.

\item \textbf{YOLO} will be employed for efficient and real-time object detection in plant images. It will enhance the system's ability to quickly and accurately locate and identify plant leaves and diseases within images and video streams.

\item \textbf{TensorFlow} will be the core technology for building and training machine learning models, including convolutional neural networks (CNNs), to improve the accuracy of plant identification and disease detection.

\item \textbf{CNN} built using TensorFlow, will be used for deep learning tasks, such as recognizing plant leaves and detecting diseases, making the system proficient in image-based plant analysis.

\index{G!\\Google Colab\hfill page}

\item \textbf{Google Colab} will provide access to free GPU resources, allowing the system to train machine learning models more efficiently and handle large datasets for plant identification and disease detection.

\index{G!\\Google Cloud\hfill page}

\item \textbf{Google Cloud} services can be used for deploying the system, handling scalability, storing data, and running the application in a reliable and cost-effective manner.

\index{R!\\React JS\hfill page}

\item \textbf{React JS} will be employed to build a user-friendly and interactive web interface, enabling users to easily input images and receive real-time information about plant identification and disease status.

\item \textbf{Flask} will serve as the backend framework for the system, handling API requests and responses. It will ensure smooth communication between the frontend and the machine learning models.

\item \textbf{Python} will be the primary programming language for the system, enabling the integration of various components and libraries for plant identification and disease detection.

\index{R!\\RESTful API\hfill page}

\item \textbf{RESTful API} will be implemented to facilitate communication between the frontend and backend of the system, allowing users to interact with the application and receive plant-related information.

\end{itemize}

% \printindex

```
External Interface Requirements\hfill page 9
\newline
Ethical Considerations\hfill page 9
\subsection*{\Huge F}
Functional Requirements\hfill page 8
\newline Flask\hfill page 5
\subsection*{\Huge G}
General Constraints\hfill page 7
\newline
Google Cloud\hfill page 10
\newline
\newline
Software Interfaces\hfill page 9
\newline
Specific Requirement\hfill page 8
\subsection*{\Huge T}
TensorFlow\hfill page 4

\subsection*{\Huge U}
Usability\hfill page 9
\subsection*{\Huge Y}
YOLO\hfill page 4
```

```
Product Perspective\hfill page 6
\newline
Python\hfill page 5
\subsection*{\Huge R}
RESTful API\hfill page 10
\newline
React JS\hfill page 10
\newline
References\hfill page 5
\subsection*{\Huge S}
Scope of Project\hfill page 3
\newpage
\underline{SCREENSHOTS OF PROJECT
PLAN} \\
  \includegraphics[width=15cm,
height=10cm]{Plan ss 1.jpg}
\\ \\
\vspace{2cm}
  \includegraphics[width=15cm,
height=10cm]{Plan ss 2.jpg}
\end{document}
```