VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Report on Extra-Curricular activities done as a part of

ACXC154N Computer in Society

Submitted by

Alisha Bandyopadhyay
21BCE2569

To

Dr. GOVINDA K

Fall Semester 2023-2024

TABLE OF ACTIVITIES

Byte Bistro 2.0

Event Description:

As a member of CSI, I had the pleasure of organizing Byte Bistro 2.0, featuring guest speaker Tanya Warrier, a strategic cloud engineer at Google and a proud CSI alumna. The event took place on March 7th, 2024, from 11:00 AM to 01:00 PM, with 47 attendees. During the session, Tanya delved into the concepts of Generative AI and the various components of Large Language Models (LLMs). She highlighted the impact of Gen-AI on the daily operations of companies and discussed its diverse use cases across different industries. Tanya also shed light on future technologies and how LLMs can be leveraged to handle them. She provided insights into Google's future prospects and wrapped up with a Q&A session to benefit the attendees. Tanya concluded the event by sharing her experiences at VIT, encouraging participants to remain active, curious, and eager to explore technology. The entire session was seamlessly coordinated with the valuable guidance and support of our faculty coordinators, Dr. Sendhil Kumar K. S and Dr. Govinda K.

Event Photos:

Time Spent:

2 Hours Event

4 Hours in Preparation

Init with CSI: Flexbox Files

Event Description:

As a member of CSI-VIT, I had the opportunity to help organize "Init with CSI: Flexbox Files," a comprehensive hands-on workshop focused on frontend web development specifically tailored for students enrolled in the Computers in Society - EXC course. This workshop took place on March 28th, 2024, from 11:00 AM to 5:00 PM at the VOC Gallery. The primary aim was to enhance participants' practical knowledge in web development, offering them a valuable opportunity to gain hands-on experience in creating responsive and interactive websites.

The workshop was meticulously structured into distinct segments, each covering a range of web development topics. Manas provided an in-depth overview of HTML, discussing both basic and advanced topics to ensure a solid foundation. Keshav introduced the fundamentals of CSS, with a particular focus on effective styling and layout techniques. Additionally, Manas explored the Flexbox model in detail, emphasizing its importance in creating flexible and responsive layouts essential for modern web design.

Together, Manas and Keshav also covered essential JavaScript topics, including DOM manipulation, inheritance, and the prototype chain. This comprehensive approach ensured that participants gained a holistic understanding of frontend web development, equipping them with the skills necessary to create dynamic and visually appealing websites. The workshop concluded with a Q&A session, allowing students to clarify doubts and deepen their understanding of the concepts discussed.

Time Spent:

6 Hours Event

10 Hours in Preparation

Flexbox Files: 2.0

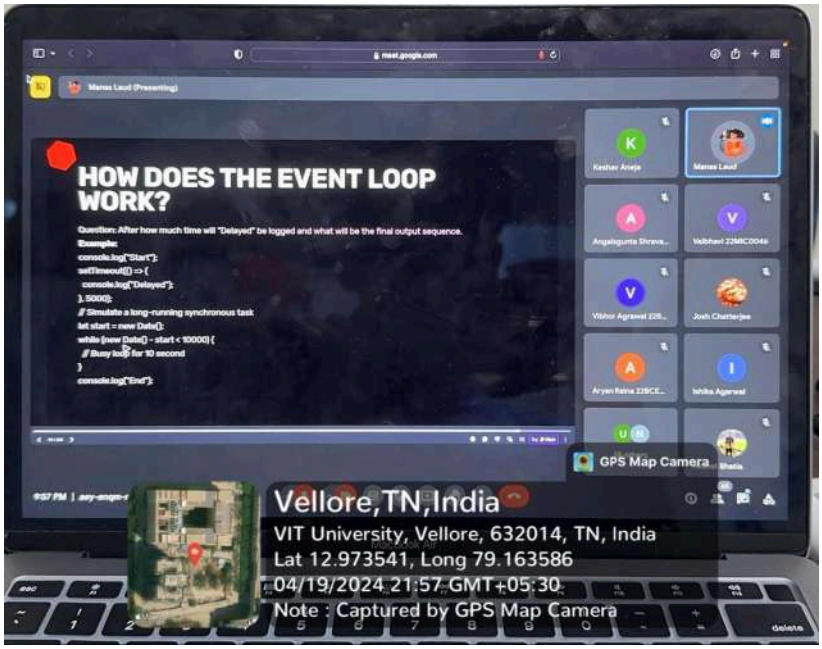
Event Description:

As a member of CSI-VIT, I was involved in organizing Day 2 of the "Flexbox Files" workshop, which continued to focus on advanced JavaScript, APIs, and storage mechanisms. This online workshop aimed to deepen participants' understanding and skills in these critical areas. Manas began by covering various storage mechanisms essential for web development, discussing cache storage, cookies, local storage, and session storage, and providing participants with practical insights into their applications.

Keshav then delved into advanced JavaScript topics, introducing asynchronous JavaScript and threads. He demonstrated file reading techniques and highlighted the execution flow, illustrating how the code is executed and the function of the call stack. Manas continued by explaining callbacks and promises, providing examples to clarify these concepts. He concluded with an overview of APIs, emphasizing their importance in modern web development.

The interactive online sessions led by Manas and Keshav encouraged participants to engage actively, ask questions, and apply their knowledge in practical exercises. This hands-on approach ensured that attendees not only understood the theoretical aspects but also gained practical experience in using advanced JavaScript, APIs, and storage mechanisms effectively.

Event Photos:



Time Spent:

1.5 Hours Event

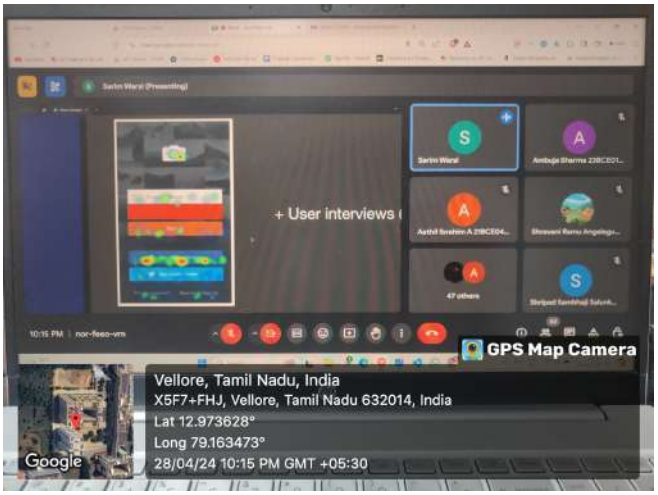
6 Hours in Preparation

Init with CSI: Dabbling in design

Event Description:

CSI-VIT hosted an online workshop titled "Init with CSI: Dabbling in Design," led by Sarim Warsi, the current design head. This workshop aimed to introduce participants to front-end web design. Sarim began by covering the fundamentals of design, gradually moving towards aesthetics and usability, and discussing how to balance the two. He provided a brief overview of UI design, discussing popular tools such as Sketch, Figma, and Adobe XD, and exploring the scope of this sub-domain. The session then transitioned to UX design, focusing on user psychology and tools for gathering and analyzing user data, such as UX cam, heatmaps, and user interviews. The workshop also addressed interaction design, product thinking, and product design, emphasizing the importance of studying the market and aligning organizational goals with user needs. The event concluded with an engaging discussion on the visual aspects of design, including graphic and motion graphic design.

Event Photos:



Time Spent:

1.5 Hours Event

6 Hours in Preparation

AI- Powered Leaf Analysis System for
Plant Species Identification and Disease Detection - Project Report

Duration : 79 Hours

INTRODUCTION:

In the dynamic landscape of agriculture and botanical research, advancements in technology have become integral to addressing challenges and optimizing processes. The LeafSense project emerges within this domain as a pioneering effort to fuse artificial intelligence (AI) and machine learning (ML) with botanical science, aiming to revolutionize plant analysis, species identification, and disease detection.

Domain Overview:

The domain of agriculture and botanical research encompasses a broad spectrum of activities, ranging from crop cultivation and plant breeding to biodiversity conservation and ecological restoration. In this domain, the understanding of plant species, their characteristics, and health status plays a pivotal role in ensuring agricultural productivity, environmental sustainability, and ecosystem resilience. Traditional methods of plant analysis and disease diagnosis often require extensive manual labor and expertise, leading to inefficiencies and limitations in scalability. However, with the advent of AI and ML technologies, there is immense potential to overcome these challenges and unlock new avenues for innovation in plant science.

Project Overview:

The LeafSense project is a groundbreaking initiative that harnesses the power of AI and ML to transform plant analysis and disease detection. At its core, LeafSense is designed to serve as an intelligent platform for plant species identification and disease diagnosis, catering to a diverse range of users including farmers, researchers, botanists, and environmentalists. By leveraging advanced algorithms and image processing techniques, LeafSense offers users the ability to accurately identify plant species and detect diseases in plant leaves with unprecedented speed and accuracy.

Objectives of the Project:

1. Plant Species Identification: LeafSense aims to provide users with a reliable and efficient means of identifying plant species based on leaf morphology and other botanical characteristics. This functionality is essential for various applications such as biodiversity conservation, ecosystem monitoring, and horticultural management.

2. Disease Detection: LeafSense seeks to enhance disease diagnosis in plants by leveraging machine learning algorithms to analyze images of diseased leaves. By detecting early signs of diseases such as fungal infections, viral pathogens, and nutrient deficiencies, LeafSense empowers users to implement timely interventions and mitigate the spread of diseases, thereby safeguarding crop yields and preserving plant health.

Significance of the Project:

The LeafSense project holds immense significance in the domains of agriculture, botany, and environmental science. By providing accessible and accurate plant analysis tools, LeafSense has the potential to revolutionize plant health management, empower individuals and communities, and foster a more sustainable and resilient agricultural ecosystem. Through its innovative approach, LeafSense aims to contribute to global efforts towards food security, environmental conservation, and sustainable development.

ABSTRACT:

The LeafSense project is an innovative endeavor that merges artificial intelligence (AI) and machine learning (ML) with botanical science to revolutionize plant analysis, species identification, and disease detection. In the dynamic domain of agriculture and botanical research, where the understanding of plant species and their health status is crucial for ensuring agricultural productivity and environmental sustainability, LeafSense emerges as a pioneering solution.

At its core, LeafSense is designed to serve as an intelligent platform for plant species identification and disease diagnosis. By harnessing advanced algorithms and image processing techniques, LeafSense empowers users to accurately identify plant species and detect diseases in plant leaves with unprecedented speed and accuracy. This functionality is essential for a wide range of applications, including biodiversity conservation, agricultural management, and ecological restoration.

The objectives of the LeafSense project are twofold: plant species identification and disease detection. Through its AI-powered analysis capabilities, LeafSense provides users with a reliable means of identifying plant species based on leaf morphology and other botanical characteristics. Additionally, LeafSense enhances disease diagnosis in plants by leveraging machine learning algorithms to analyze images of diseased leaves, enabling users to detect early signs of diseases and implement timely interventions.

The significance of the LeafSense project extends beyond its technical capabilities; it embodies a vision of environmental stewardship, agricultural resilience, and global food security. By providing accessible and accurate plant analysis tools, LeafSense aims to empower individuals and communities to become proactive stewards of their plant ecosystems, fostering a harmonious relationship between humanity and the natural world.

In summary, the LeafSense project represents a convergence of technological innovation, scientific inquiry, and environmental consciousness. Through its innovative approach, LeafSense seeks to redefine the landscape of plant health management, paving the way for a greener, healthier future for generations to come.

SOFTWARE REQUIREMENTS SPECIFICATION:

1. Introduction:

The Software Requirements Specification (SRS) document outlines the functional and non-functional requirements of the LeafSense project, providing a comprehensive understanding of the system's capabilities and constraints.

2. Purpose:

The purpose of this document is to define the software requirements for LeafSense, an AI-powered leaf analysis system for plant species identification and disease detection. It serves as a blueprint for the development team, guiding the design, implementation, and testing phases of the project.

3. Scope:

LeafSense is designed to cater to a wide range of users, including plant enthusiasts, farmers, researchers, and professionals in the agricultural sector. The system aims to provide accurate and accessible plant analysis tools, enabling users to identify plant species and detect diseases in plant leaves.

4. Functional Requirements:

4.1 User Authentication:
- Users must be able to create an account and log in securely.
- User authentication mechanisms such as username/password or social media login should be provided.

4.2 Image Upload:
- Users should be able to upload images of plant leaves for analysis.
- Supported image formats include JPEG, PNG, and BMP.

4.3 Species Identification:
- LeafSense must accurately identify plant species based on leaf morphology and other botanical characteristics.
- The system should provide detailed information about the identified species, including common name, scientific name, and relevant botanical information.

4.4 Disease Detection:

- LeafSense should detect diseases in plant leaves based on image analysis.
- Supported diseases include fungal infections, viral pathogens, and nutrient deficiencies.

4.5 Feedback Submission:
- Users should have the option to submit feedback or report any issues with the system.
- A feedback form or mechanism should be provided within the application.

4.6 Contact Options:
- LeafSense should provide users with options to contact the support team for assistance or collaboration.
- Contact information such as email address or contact form should be available.

5. Non-Functional Requirements:

5.1 Performance:
- LeafSense should be able to handle multiple user requests simultaneously without significant latency.
- Response times for image analysis and result display should be minimal.

5.2 Security:
- User authentication and data transmission should be encrypted to ensure data privacy and security.
- Access control mechanisms should be in place to prevent unauthorized access to sensitive information.

5.3 Usability:
- The user interface should be intuitive, user-friendly, and accessible to users with varying levels of technical expertise.
- Help documentation or tooltips should be provided to assist users in navigating the application.

5.4 Reliability:
- LeafSense should be reliable and available for use at all times, with minimal downtime.
- Automated backups and data recovery mechanisms should be implemented to prevent data loss.

6. External Interfaces:

6.1 APIs:

- LeafSense may integrate with external APIs for accessing plant databases, image processing libraries, and other third-party services.
- API documentation and usage guidelines should be provided for developers.

7. Constraints:

7.1 Hardware:

- LeafSense should be compatible with a wide range of devices, including desktop computers, laptops, tablets, and smartphones.
- Minimum hardware requirements, such as processor speed, memory, and storage space, should be specified.

7.2 Software:

- LeafSense should support multiple operating systems, including Windows, macOS, and Linux.
- Compatibility with popular web browsers such as Google Chrome, Mozilla Firefox, and Safari should be ensured.

8. Assumptions:

8.1 Data Availability:

- It is assumed that a sufficient dataset of plant images with corresponding species and disease labels will be available for training and testing the machine learning models.

8.2 Internet Connectivity:

- LeafSense requires an internet connection for accessing external resources such as plant databases and APIs.

UI DESIGN:

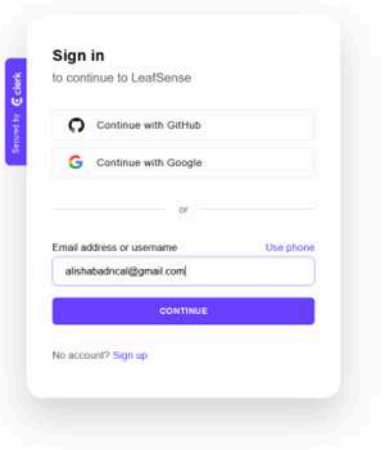
1) Landing Page:

Provides a brief introduction to LeafSense and options to proceed.

---

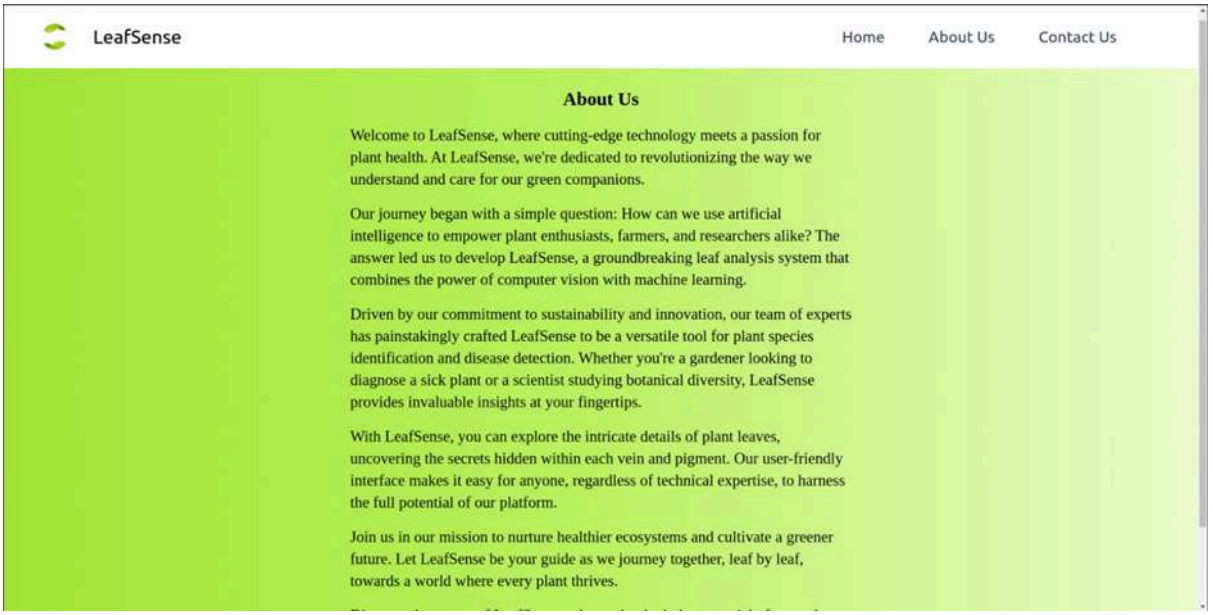
2) Login Page:

The login page grants access to LeafSense's array of plant health analysis tools and personalized features by users to login with their credentials.



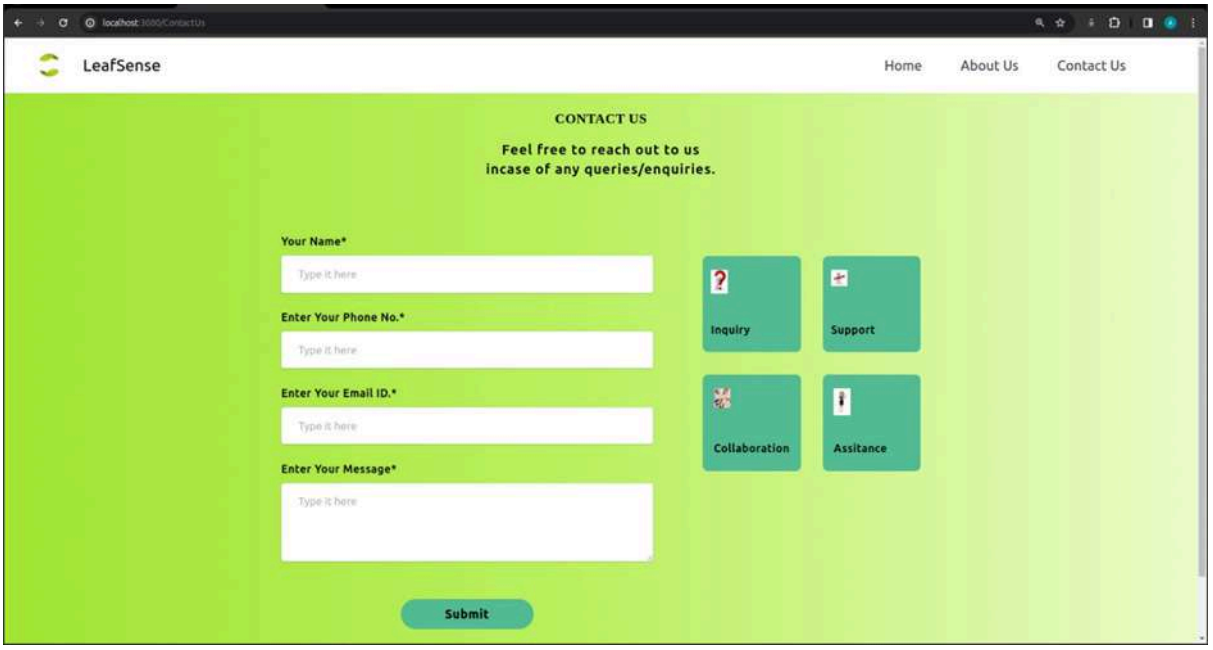
3) About Us:

Users can access this screen to learn more about the LeafSense project, its mission, and the team behind its development.



4) Contact Us:

For any inquiries, assistance requests, or collaboration opportunities, users can utilize this screen to get in touch with the LeafSense team.

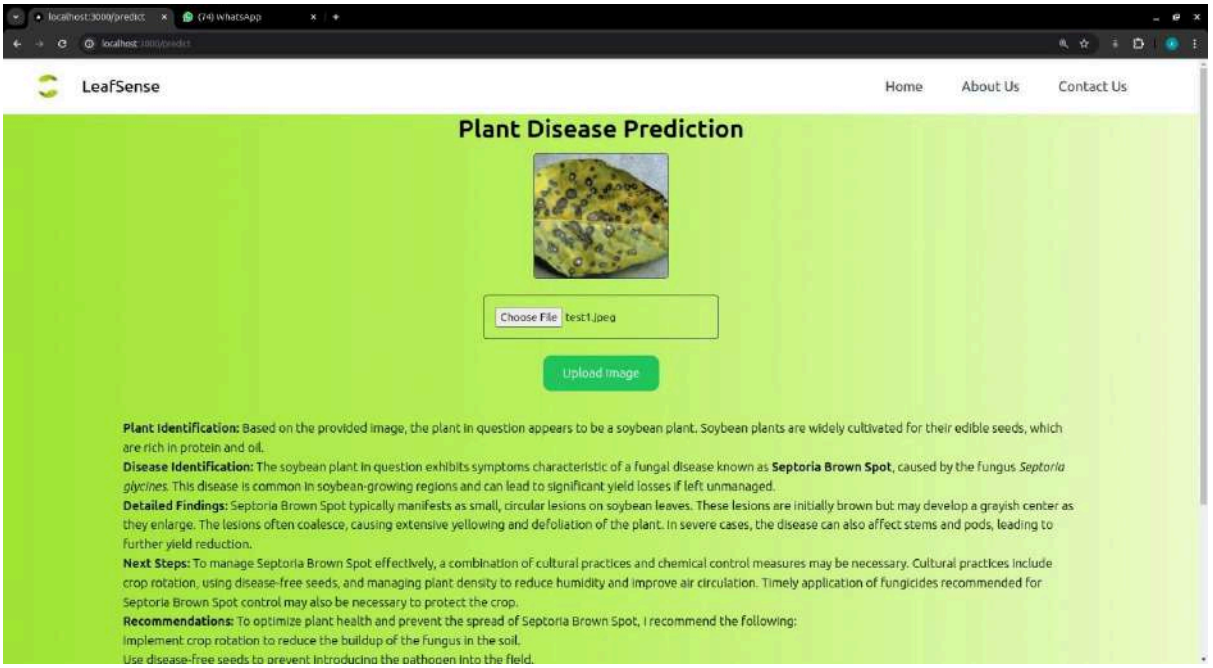


5) Upload Image:

Users are presented with an interface to upload an image of a plant leaf, which serves as the input for species identification and disease detection.

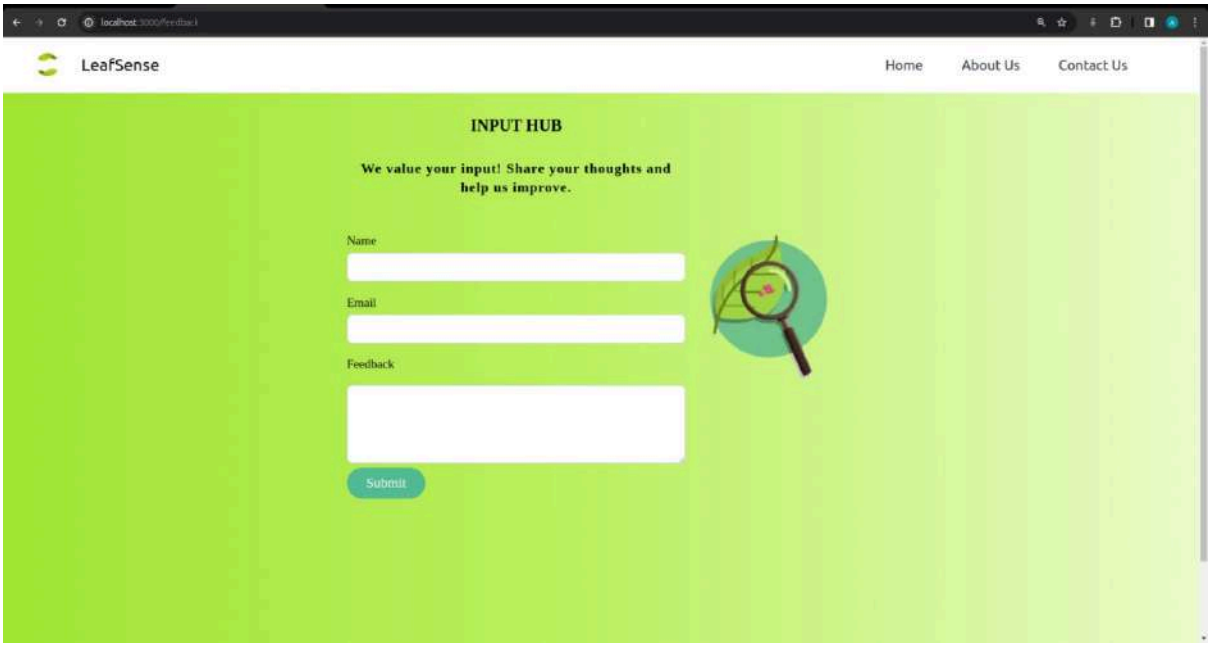
6) Result Page:

Upon uploading an image, users are presented with the identified species of the plant based on the analysis performed by LeafSense. This screen displays the outcomes of disease detection analysis, indicating whether any diseases are detected in the plant leaf and providing relevant details. It also gives recommendations on how to cure the disease as well.

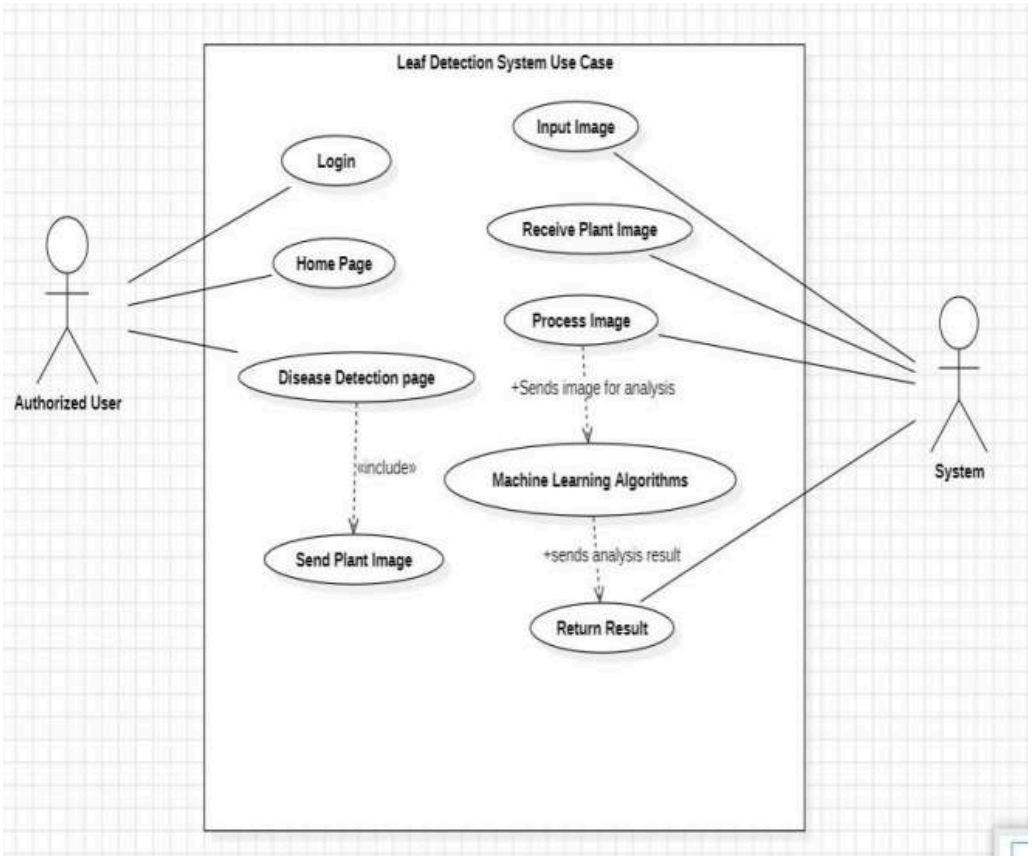


7) Feedback Page:

LeafSense encourages user feedback and offers a dedicated page for users to submit their thoughts, suggestions, or report any issues encountered while using the system.



USE CASE DIAGRAM:



ALGORITHM – CODE:

Algorithm: Image Analysis with Flask and GenerativeAI

Step 1: Initialize Flask App and Configure GenerativeAI API
1.1 Import necessary modules and libraries.
  1.1.1 Import Flask, request, jsonify.
  1.1.2 Import CORS from flask_cors.
  1.1.3 Import Image, io, base64 from PIL.
  1.1.4 Import genai from google.generativeai.
  1.1.5 Import load_dotenv from dotenv.
  1.1.6 Import os.
1.2. Load environment variables from .env using load_dotenv().
1.3. Configure GenerativeAI API key using os.getenv("GOOGLE_API_KEY").
1.4.
1.4.1 Set up model configuration for text generation.
1.4.2 Define generation_config dictionary.
1.5. Set up safety settings for content generation.
  1.5.1 Define safety_settings list.
1.6. Initialize GenerativeModel.
  1. 6.1 Initialize GenerativeModel with model_name, generation_config, safety_settings.

Step 2: Define Input Prompt
2.1. Define input_prompt multiline string with analysis guidelines.

Step 3: Define Flask Route for Image Analysis
3.1. Define route '/analyze' using app.route().
  3.1.1 Specify HTTP methods allowed (POST).
3.2. Handle POST requests for image analysis.
  3. 2.1 Extract image file from request.
        3 .2.1.1 Check if 'image' is in request.files.
    3.2.1.2 If not, return JSON error response.
 3. 2.2 Open image file using Image.open().
3.3. Generate content using GenerativeAI API.
  3.3.1 Generate content with model.generate_content().
3.4. Return generated response as JSON using jsonify.

Step 4: Run Flask App
4.1. Check if the script is executed directly.
  4.1.1 If true, run Flask app with debug mode enabled using app.run()

Backend Code:

```
from flask import Flask, request, jsonify
from flask_cors import CORS

from PIL import Image
import io
import base64
import google.generativeai as genai
from dotenv import load_dotenv
import os

# Load environment variables from a .env file
load_dotenv()
# Configure the GenerativeAI API key using the loaded environment variable
genai.configure(api_key=os.getenv("GOOGLE_API_KEY"))
# Set up the model configuration for text generation
generation_config = {
   "temperature": 0.4,
   "top_p": 1,
   "top_k": 32,
   "max_output_tokens": 4096,
}
# Define safety settings for content generation
safety_settings = [
                        {"category": f"HARM_CATEGORY_{category}", "threshold":
                "BLOCK_MEDIUM_AND_ABOVE"}
       for category in ["HARASSMENT", "HATE_SPEECH", "SEXUALLY_EXPLICIT",
                "DANGEROUS_CONTENT"]
]
# Initialize the GenerativeModel with the specified model name, configuration, and safety
                settings
                model = genai.GenerativeModel(
                   model_name="gemini-pro-vision",
                   generation_config=generation_config,
   safety_settings=safety_settings,
)
input_prompt = """
```

As a highly skilled plant pathologist, your expertise is indispensable in our pursuit of maintaining optimal plant health. You will be provided with information or samples related to plant diseases, and your role involves conducting a detailed analysis to identify the specific issues, propose solutions, and offer recommendations.

Analysis Guidelines:

1. **Plant Identification:** Identify the plant species and variety based on the provided information or samples.

2. **Disease Identification:** Examine the provided information or samples to identify and characterize plant diseases accurately.

3. **Detailed Findings:** Provide in-depth findings on the nature and extent of the identified plant diseases, including affected plant parts, symptoms, and potential causes.

4. **Next Steps:** Outline the recommended course of action for managing and controlling the identified plant diseases. This may involve treatment options, preventive measures, or further investigations.

5. **Recommendations:** Offer informed recommendations for maintaining plant health, preventing disease spread, and optimizing overall plant well-being.

6. **Important Note:** As a plant pathologist, your insights are vital for informed decision-making in agriculture and plant management. Your response should be thorough, concise, and focused on plant health.

Disclaimer:
"Please note that the information provided is based on plant pathology analysis and should not replace professional agricultural advice. Consult with qualified agricultural experts before implementing any strategies or treatments."

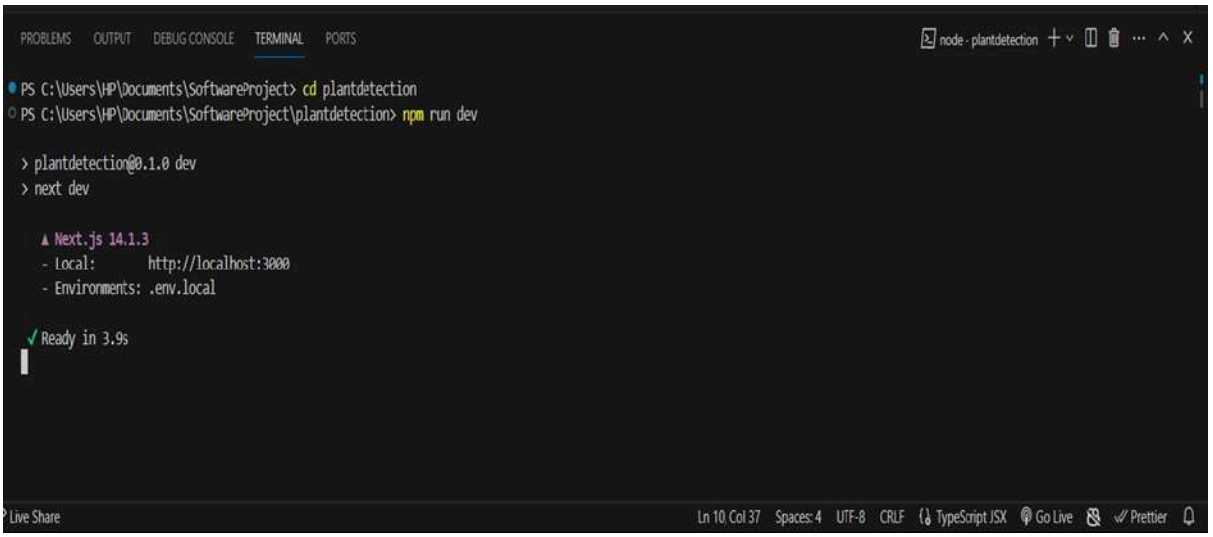
Your role is pivotal in ensuring the health and productivity of plants. Proceed to analyze the provided information or samples, adhering to the structured
"""

```
app = Flask(__name__)
CORS(app)

@app.route('/analyze', methods=['POST'])
def analyze():
    if 'image' not in request.files:
        return jsonify({'error': 'No image provided'})
```

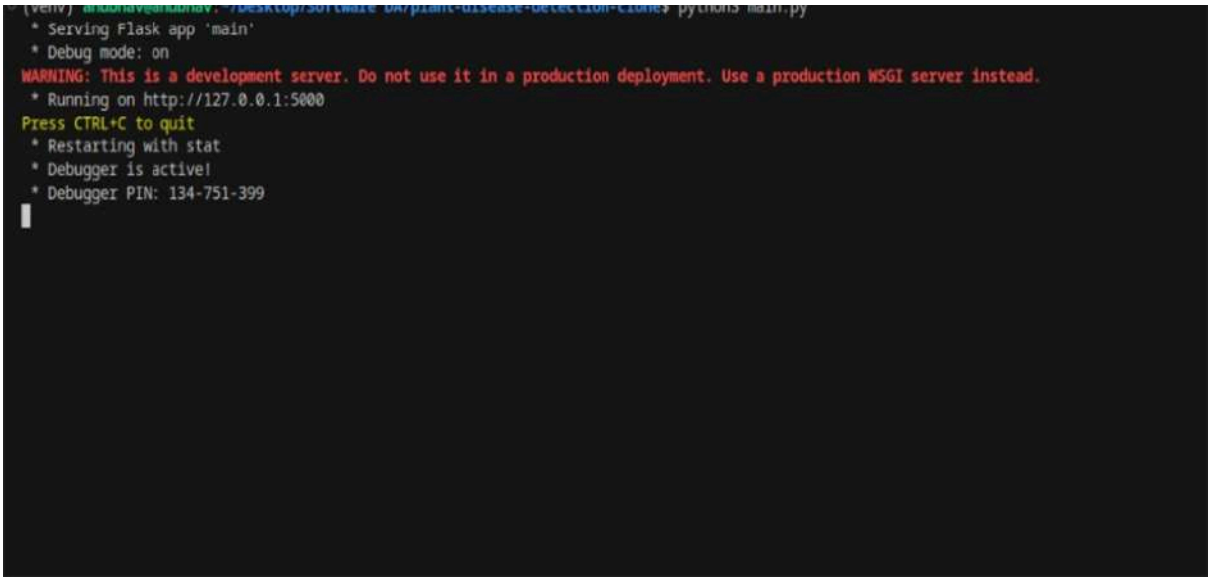
```
    image = request.files['image']
    image = Image.open(image)
    response = model.generate_content([input_prompt,image])
    print(response.text)
    return jsonify({'response': response.text})

if __name__ == '__main__':
    app.run(debug=True)
```

IMPLEMENTATION:

To run the front-end:



To run the back-end:



CONCLUSION:

In conclusion, the LeafSense project represents a significant advancement in the field of plant pathology and agriculture by leveraging AI and ML technologies to develop an innovative leaf analysis system. By harnessing the power of computer vision and machine learning, LeafSense offers a sophisticated solution for plant species identification and disease detection, providing farmers, agronomists, and researchers with valuable insights into plant health and management.

What sets LeafSense apart from existing systems is its comprehensive approach to leaf analysis, combining species identification and disease detection in a single platform. Unlike traditional methods that rely on manual inspection or limited diagnostic tools, LeafSense offers an automated and efficient solution that can accurately identify plant species and diagnose diseases based on leaf images. This automation not only saves time and effort but also improves the accuracy and reliability of plant health assessments.

Furthermore, LeafSense's integration of AI and ML algorithms enables continuous learning and adaptation, allowing the system to improve over time and stay updated with emerging plant diseases and species variations. This dynamic nature ensures that LeafSense remains relevant and effective in addressing the evolving challenges faced by farmers and agricultural professionals.

Overall, the LeafSense project represents a significant advancement in agricultural technology, offering a transformative solution for plant health monitoring and management. Its ability to provide accurate and timely insights into plant species and diseases sets it apart from existing systems, making it a valuable tool for enhancing crop productivity, optimizing resource utilization, and promoting sustainable agriculture.