# Plant leaf Disease detection Using CNN and TensorFlow(Phytora: Plant Health’s Future)

- BY Team SALAAR

# Table of Contents

# 01

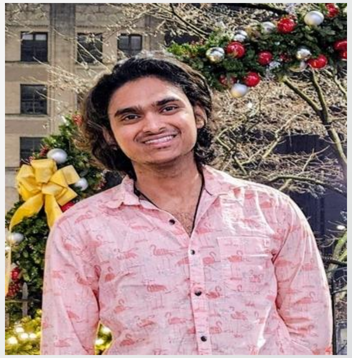
# Team Member Roles and Responsibilities

# Team Members Roles and Responsibilities

Manoj Reddy
Scrum Master

Paul
Sr. Front End Developer

Krishna Kishore
Sr. Back End Developer
& Team Lead



Gopi Krishna
Jr. Back End Developer

Karthik
Back End QA

Sai Priya
Front End
Team Lead

Nikitha
Frond-End QA

NagaLakshmi
Jr. Front End
Developer

# 02

## Problem Statement

# Problem Statement

- Apple trees are prone to diseases like scab and rust, impacting yield if not detected early. Traditional methods are costly and labor-intensive, while AI-powered CNNs offer faster, more accurate detection. However, challenges like background noise and variable disease appearance hinder accuracy. A robust deep learning model is needed to efficiently detect and classify apple diseases, ensuring timely intervention and reducing agricultural losses.

# 03

Project Description

# Project Description

| | |
|---|---|
| **Project Name:** | Phytora |
| **Team:** | Salaar |
| **Project Description:** | **For** farmers and agricultural researchers<br>**who** need an efficient and cost-effective method to detect apple tree diseases**,**<br>**the** AI-based plant disease detection system<br>**is a** deep learning-powered solution<br>**that** identifies diseases like scab and rust early, preventing crop loss.<br>**Unlike** traditional manual inspection methods**,**<br>**our application** leverages Convolutional Neural Networks (CNNs) to provide fast, accurate, and automated disease classification, overcoming challenges like background noise and variable disease appearances**.** |
| | |
| **Benefit Outcomes:** | • **Early Detection**: Enables quick identification of apple tree diseases, reducing yield loss.<br>• **Cost-Effective**: Eliminates the need for frequent manual inspections, saving labor costs.<br>• **High Accuracy**: Uses AI and deep learning models to minimize misdiagnosis.<br>• **Scalability**: Can be applied to large farms and integrated into existing agricultural systems.<br>• **User-Friendly**: Provides a simple interface for farmers to upload images and get instant results. |
| **Github Link:** | https://github.com/htmw/2025S-SALAAR/wiki |

# 04

Personas

# PERSONAS

- **Farmer – John Carter**
- **Age:** 45
- **Occupation:** Apple Orchard Owner
- **Location:** Washington, USA
- **Technology Proficiency:** Low to Moderate
- **Farming Experience:** 20+ years
- **Pain Points:**
  - Relies on manual disease inspection, which is time-consuming and error-prone.
  - Finds it hard to differentiate between early-stage symptoms of apple scab and rust.
  - High costs for hiring agricultural experts and consultants.
  - Crop losses due to late detection of infections, leading to lower profits.
- **Needs & Expectations:**
  - A mobile-friendly app where he can take a picture of infected leaves and receive instant results.
  - An AI-driven advisory system suggesting treatment solutions for different disease stages.
  - Offline functionality, since farms often have poor internet connectivity.
  - A cost-effective solution that doesn't require expensive hardware.
- **How Our Product Helps:**
  - Automated disease detection reduces guesswork and ensures timely intervention.
  - Provides step-by-step guidance on disease management, reducing dependency on external experts.
  - Saves time and labor costs, increasing farm productivity and yield.

**Agricultural Researcher – Dr. Emmy Zhao**

- **Age:** 38
- **Occupation:** Plant Pathologist & Researcher
- **Location:** California, USA
- **Technology Proficiency:** High
- **Research Focus:** AI-based plant disease prediction, crop disease resistance
- **Pain Points:**
- Needs large, high-quality datasets to improve disease classification models.
- Variability in disease symptoms (color, shape, size) makes model training challenging.
- Lack of real-time field data for studying disease spread patterns.
- Difficulty in integrating AI-based tools with existing agricultural research systems.
- **Needs & Expectations:**
- Access to a labeled image dataset for improving AI accuracy.
- A cloud-based dashboard with insights into disease occurrences across different regions.
- Ability to contribute research data and collaborate with other scientists.
- A customizable AI model that can be adapted for different crops and environmental conditions.
- **How Our Product Helps:**
- Provides real-time disease detection data, aiding in research and policy recommendations.
- Offers a machine-learning training platform for researchers to refine detection algorithms.
- Helps predict disease outbreaks, leading to more proactive agricultural strategies.

- Agri-Tech Startup Founder – Michael Rodriguez
  - **Age:** 32
  - **Occupation:** CEO of a Smart Farming Startup
  - **Location:** Texas, USA
  - **Technology Proficiency:** High
  - **Business Focus:** AI-driven farm automation, precision agriculture
  - **Pain Points:**
    - Farmers are hesitant to adopt AI-based solutions due to trust issues and cost concerns.
    - Needs a scalable AI model that can integrate with existing smart farming hardware.
    - Difficulty in monetizing AI-based agricultural solutions while keeping them affordable.
    - Requires accurate, real-world disease detection data to improve product effectiveness.
  - **Needs & Expectations:**
    - A lightweight AI model that can be embedded in farm drones and IoT devices.
    - A subscription-based business model that balances affordability and profitability.
    - Market validation data to demonstrate AI's effectiveness in reducing crop losses.
    - A seamless API to integrate disease detection with smart irrigation and pesticide control systems.
  - **How Our Product Helps:**
    - Provides real-time analytics and reporting tools for farm management.
    - Enables precision agriculture, optimizing pesticide use and reducing costs.

# 05

## Technologies

# Tools & Technologies

TensorFlow is a software library for
machine learning and
artificial intelligence.
https://www.tensorflow.org/
.

Tensor Flow

A versatile, high-level programming
language.

https://www.python.org/

Python

Scikit-learn is probably the most useful
library for machine learning in Python.
https://scikit-learn.org/stable/

Scikit

Streamlit For building interactive
web applications.
https://streamlit.io/

Stream lit

# Cont..

Visual Studio Code for Development
https://code.visualstudio.com

VS Code

Jira for project management and issue tracking
https://www.atlassian.com

Jira

Anaconda for Python-based machine learning IDE
https://www.anaconda.com/

Anaconda

My SQL Cluster enables users to meet the database challenges of next generation
https://www.mysql.com/

# Cont..

Allows developers to create, store, manage, and share their code
https://github.com/

Working together is easier with Microsoft Teams
https://www.microsoft.com/en-us/microsoft-teams/group-chat-software

# 06

Algorithms

# Algorithms

- Our system utilizes EfficientNet, a state-of-the-art CNN model, for highly accurate and optimized plant disease detection. EfficientNet balances speed and accuracy by scaling depth, width, and resolution efficiently. Transfer learning enhances performance, while FastAPI enables real-time predictions, ensuring fast, precise, and resource-efficient disease identification for farmers and researchers.

# 07

## Project Schedule

# Project Schedule

- Sprint 0: Initialization & Planning
- Sprint 1: UI & Model Training
- Sprint 2: Full Integration & Testing
- Sprint 3: Optimization & Deployment

# 08

## Team Working Agreement

**Team Working Agreement – Phytora**

**1. Purpose & Scope**

This agreement establishes collaboration guidelines for the development of Phytora by Team Salaar, an AI-powered plant disease detection system. It ensures smooth execution, accountability, and efficient teamwork while maintaining software development best practices.

**2. Roles & Responsibilities**

- Frontend Developer (FD): Builds and optimizes the Next.js UI, ensuring seamless interaction with the backend.
- Backend Developer (BD): Develops Node.js APIs, integrates with FastAPI, and handles system architecture.
- Machine Learning Engineer (MLE): Trains and fine-tunes EfficientNet, manages datasets, and ensures accurate model predictions.
- Quality Assurance (QA): Conducts testing, detects bugs, ensures performance optimization, and validates deployment readiness.

All members are responsible for delivering assigned tasks, maintaining quality, and proactively communicating challenges.

**3. Sprint Planning & Retrospective**

- Sprint Ends – Monday: Retrospective to review achievements, blockers, and process improvements.
- Sprint Planning – Wednesday: Define tasks, assign responsibilities, and set objectives for the new sprint.

**4. Communication & Collaboration**

- WhatsApp – Daily updates and quick discussions.
- Google Meet – Sprint planning, retrospectives, and issue resolution meetings.
- Task Management – Tracked Trello for visibility and accountability.

**5. Development Workflow & Code Management**

- Version Control – Use GitHub with feature branching and PR reviews before merging.
- Testing & Quality Assurance – Implement unit and integration tests, followed by manual validation.
- Deployment – Continuous testing and incremental deployment for stability.

**6. Accountability & Conflict Resolution**

- Team members must proactively report blockers.
- Issues should be discussed and resolved within the team.
- Repeated delays or unresponsiveness may result in task reassignment.

**7. Work Hours & Availability**

- Commitment: 10-15 hours per week per team member.
- Work flexibility, but tasks must align with sprint goals.

**8. Agreement & Commitment**

All members agree to uphold this agreement to ensure efficient collaboration, high-quality development, and successful delivery of Phytora.


SALAAR

# 09

Retrospective

# Retrospective

**What Went Well?**

Team Formation & Role Clarity
Manoj - Scrum Master - Organized roles in the team.
Sai Priya Frontend Lead & Krishna Kishore Backend Lead - Smooth Coordination

Jira & Git Setup Completed
Manoj - Set up Jira with backlog items.
Paul & Naga Lakshmi (Frontend), Krishna Kishore & Gopi Krishna (Backend) - Set up the Git repository.

Daily Scrum Calls Established
10:00 PM calls were initiated. Most of the team members updated about the progress in Jira before the meeting.
Project & Personas Finalized
Sai Priya added Farmer and Agricultural Expert personas in Jira.

Presentation Video Created & Uploaded
Sai Priya and Arpula Nikitha - Frontend QA designed the video.
Paul and Karthik - Backend QA uploaded and checked it on Git.

# Cont..

**Areas of Improvement**

Jira Task Clarity
Some tasks were too broad and needed better breakdown.
Action: Manoj and Sai Priya will refine task definitions in Sprint 1.

Delayed Peer Reviews
Some pull requests were not reviewed on time.
Action: Arpula Nikitha and Karthik will enforce 24-hour review deadlines.

Git Collaboration Issues
Merge conflicts occurred due to an unclear branching strategy.
Action: Paul and Krishna Kishore will create a Git workflow guide.

Scrum Call Participation
Gopi Krishna and Naga lakshmi have forgotten to update Jira before certain calls.
Action: The members should update Jira before 9.30 PM daily.

Presentation Video Quality
More structure of script required before recording.
Action: The content would be reviewed before recording by Sai Priya and Paul.

**Sprint 1 Action Plan**

Smoothen Jira Workflow
Manoj and Sai Priya will ensure tasks are well-defined with clear acceptance criteria.

Standardize Git Workflow
Paul and Krishna Kishore will document the branching strategy: main → dev → feature-branch.

Increase Scrum Discipline
Manoj will enforce 15-minute standups.
Gopi Krishna All developers and QA members will update Jira before 9:30 PM.

Enforce Deadlines for Peer Review
Arpula Nikitha and Karthik will ensure pull requests are reviewed within 24 hours.

Improve Video and Documentation
Paul and Sai Priya will do a review before the videos go up.

Clearly, this retro summary encapsulates the essence of Sprint 0 and helps refine the execution at Sprint
Please suggest any modification…..?

# 10

Wikipage

# WikiPage

- https://github.com/htmw/2025S-SALAAR/wiki