MAKERERE UNIVERSITY

DEPARTMENT OF NETWORKS

SCHOOL OF COMPUTING AND INFORMATICS TECHNOLOGY

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

SECOND YEAR RECESS PROJECT

DETAILED REPORT ON THE AGRICURE SYSTEM

Github link: https://github.com/blackbody256/Agricure

Kaggle link: https://www.kaggle.com/code/karagwaanntreasure/cnn-model

Model Github link: https://github.com/Karagwa/Crop-Disease-model

Dataset: https://www.kaggle.com/datasets/karagwaanntreasure/plant-disease-detection

BY: GROUP J (DAY)

Name	Registration Number	Student Number
KARAGWA ANN TREASURE	23/U/09034/PS	2300709034
AYAN MUSTAFA ABDIRAHMAN	23/X/22948/EVE	2300722948
WANYANA SELINA MASEMBE	23/U/1529	2300701529
AMABE JUNIOR HUMPHREY	23/U/05942/PS	2300705942
AKANGA ANDREW	23/U/05555/PS	2300705555

Table of Contents

1.INTRODUCTION

This report summarizes the Agricure System as an AI-powered, data-driven application designed to support farmers and agronomists in diagnosing crop diseases and receiving actionable treatment recommendations. It details the system objectives, functionalities, architecture, and user interactions.

1.1 Aim

The system aims to enhance crop health management, improve yields, and reduce losses caused by plant diseases by leveraging machine learning and predictive analytics.

1.2 Goals and Objectives

The Agricure system is built with the following key objectives:

i. To facilitate secure user registration and authentication for farmers, agronomists and administrators.
ii. To allow users to upload images of affected crops via a web interface.
iii. To accurately diagnose crop diseases using a trained AI/ML model.
iv. To generate treatment recommendations based on diagnosis and environmental conditions.
v. To track historical disease reports and provide analytics on disease trends.
vi. To enable administrators to manage users, update AI models, and maintain system content.
vii. To enable agronomists to label newly uploaded datasets and provide personalized feedback to farmers who opt for the in-app chat support option.

2. SCOPE

The system encompasses the following functionalities:

i. User Management: Account registration, login, password reset, profile management.
ii. Image-Based Diagnosis: ML-based detection of crop diseases from uploaded images.
iii. Advisory System: Treatment suggestions and recommendations.
iv. Historical Analytics: Diagnosis records, trend visualization, and analytics dashboards.
v. Admin Dashboard: User and content management.

2.1 User Roles and Access

Table 1: Table shows the user roles and access rights

User role	Description	Access Rights
Farmer	This is the primary user who uploads crop images and receives advice.	Register, log in and manage their profiles Upload crop images View diagnosis and recommendations Access personal diagnosis history
Agronomist	This is an expert who can monitor trends, provide manual feedback, or validate AI results.	Access diagnosis trends Provide feedback to farmers Label new datasets for retraining
Administrator	This is a super user who manages other users, models, content, system configurations, and uploads new datasets for retraining the model.	Manage user accounts and roles View all system data Retrain and update AI model Handle reported issues

3.2 Crop Disease Classification

This section outlines the deep learning model for automated crop disease classification using transfer learning with EfficientNetB0. The model effectively handles class imbalance and achieves robust performance for agricultural applications, demonstrating production-ready capabilities.

3.2.1 Model Architecture

i. Transfer Learning Foundation: EfficientNetB0 is used as the pre-trained feature extraction backbone, with its layers frozen (base_model.trainable = False).

ii. Custom Classification Head: Appended to the base model, it consists of GlobalAveragePooling2D, two Dropout layers (0.5 and 0.3), and two Dense layers (256 units with 'relu' and num_classes units with 'softmax').

iii. Model Compilation: Optimized with Adam (learning rate=0.001), sparse_categorical_crossentropy loss, and accuracy metric.

Layer (type)	Output Shape	Param #
input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
efficientnetb0 (Functional)	(None, 7, 7, 1280)	4,049,571
global_average_pooling2d_1 (GlobalAveragePooling2D)	(None, 1280)	0
dropout_2 (Dropout)	(None, 1280)	0
dense_2 (Dense)	(None, 256)	327,936
dropout_3 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 23)	5,911

3.2.2 Advanced Training Controls

i. ModelCheckpoint: Saves the best model based on val_accuracy to "best_model.keras".

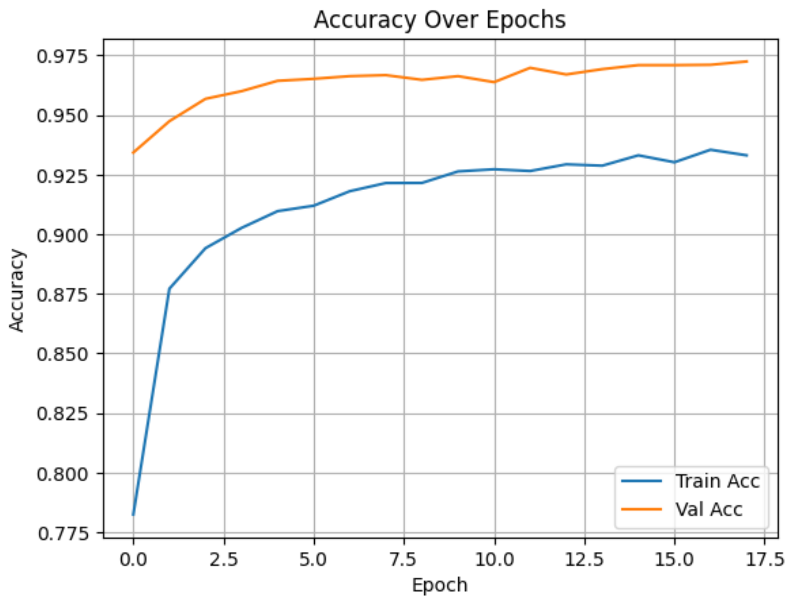
ii. EarlyStopping: Monitors val_loss and stops training if no improvement for 3 epochs, restoring the best weights.

iii. Training Execution: The model is trained for a maximum of 20 epochs on the prepared datasets, applying the computed class_weights.

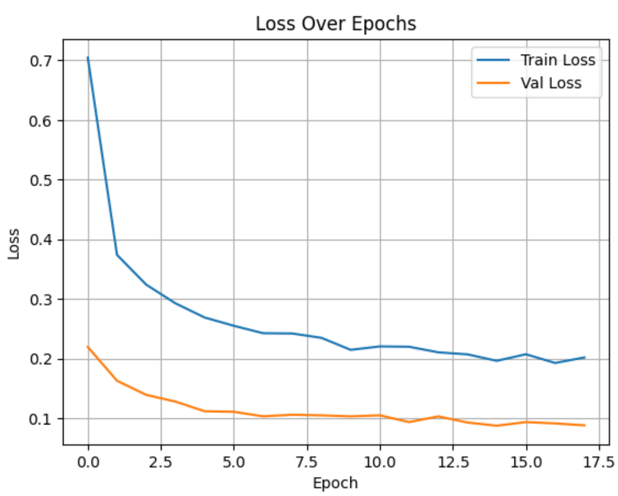
3.2.3 Model Evaluation (Summary)

The model's performance is comprehensively evaluated using:

Training Performance Visualization: Plots for accuracy and loss (training and validation).



The graph above shows the accuracy of a machine learning model over 17.5 epochs, showing that training accuracy (blue line) increases from around 0.775 to approximately 0.925, while validation accuracy (orange line) rises from about 0.80 to around 0.975, with both stabilizing towards the later epochs.



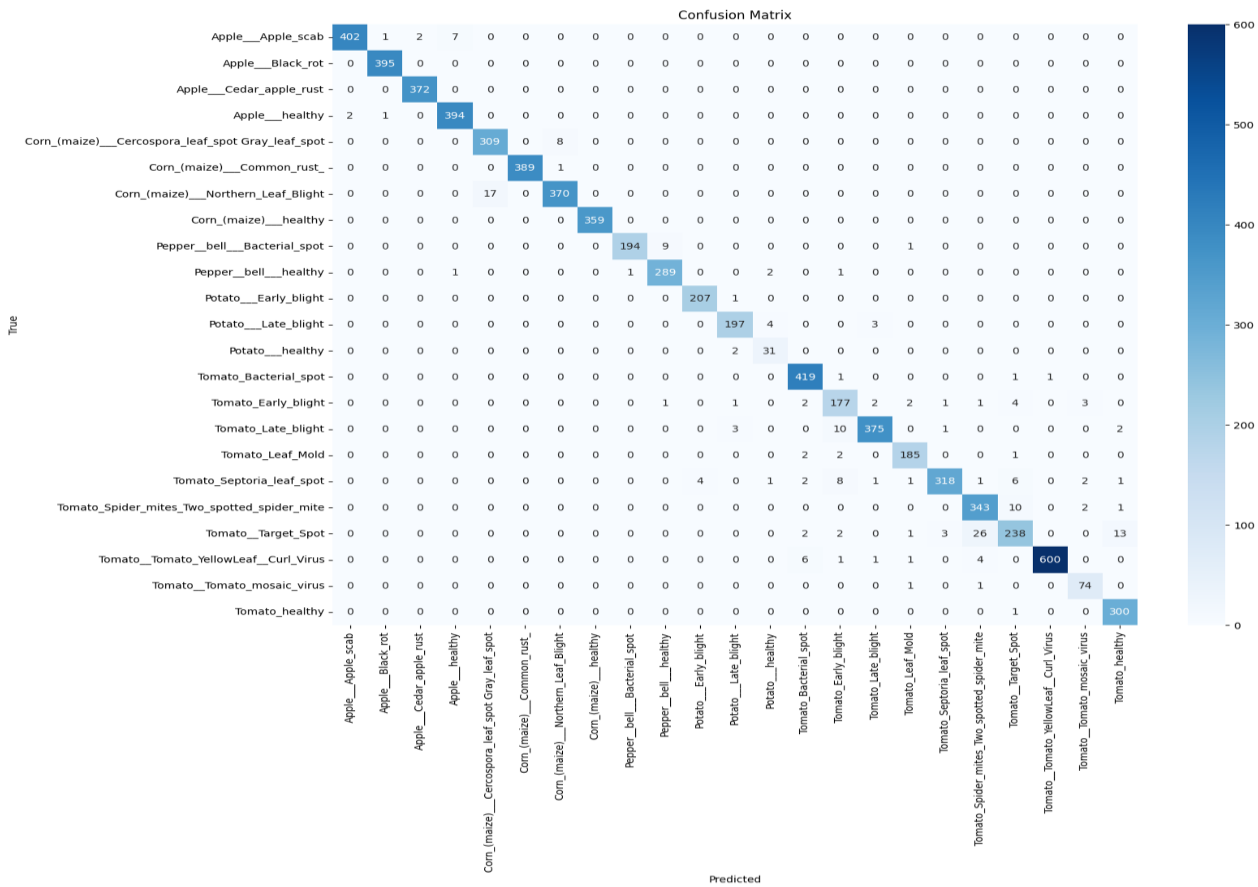
The graph depicts the loss of a machine learning model over 17.5 epochs, showing that training loss (blue line) decreases sharply from 0.7 to around 0.2 and then stabilizes, while validation loss (orange line) drops from approximately 0.2 to about 0.1 and remains relatively constant towards the end.

Together, these graphs suggest the model is learning effectively, with improving accuracy and decreasing loss, and no significant overfitting as validation metrics remain close to training metrics.

Detailed Performance Analysis: A classification report (precision, recall, f1-score, support per class) linked below

CNN Model Classification Report

Confusion matrix



3.2.4 Overall Metrics

Accuracy: 97.10%

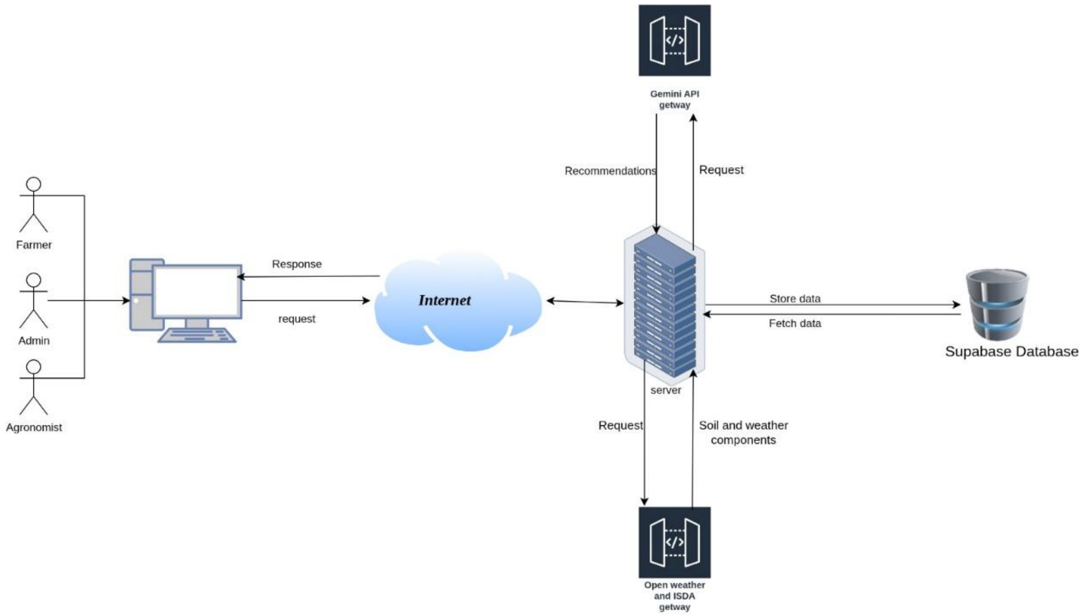
Loss: 0.0912

Best Epoch: 1

4. SYSTEM ARCHITECTURE

4.1 System Overview

This section outlines the high-level architecture of the Agricure System.



4.2 User Interface

4.2.1 Farmer Interactions

Figure 1: Figure shows Farmer dashboard

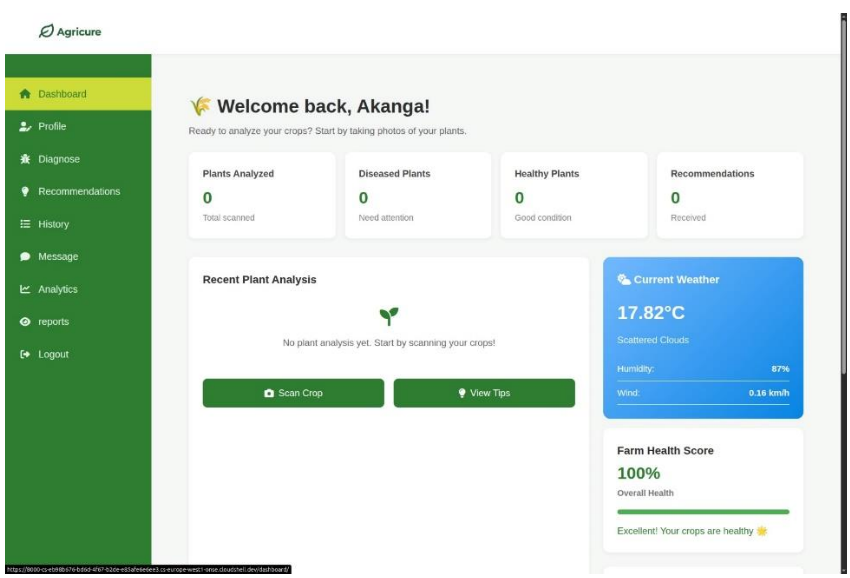


Figure 2: Figure shows the Farmer Diagnosis page

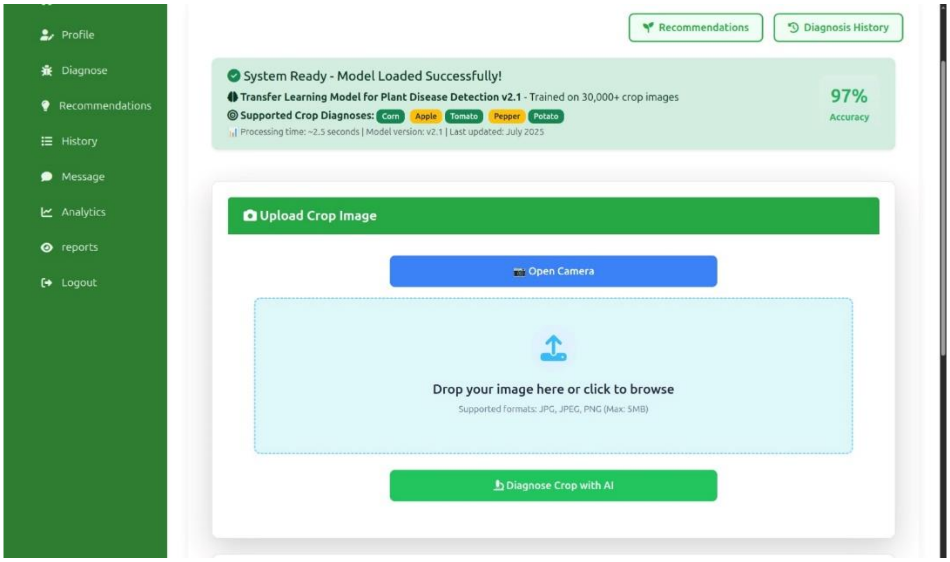
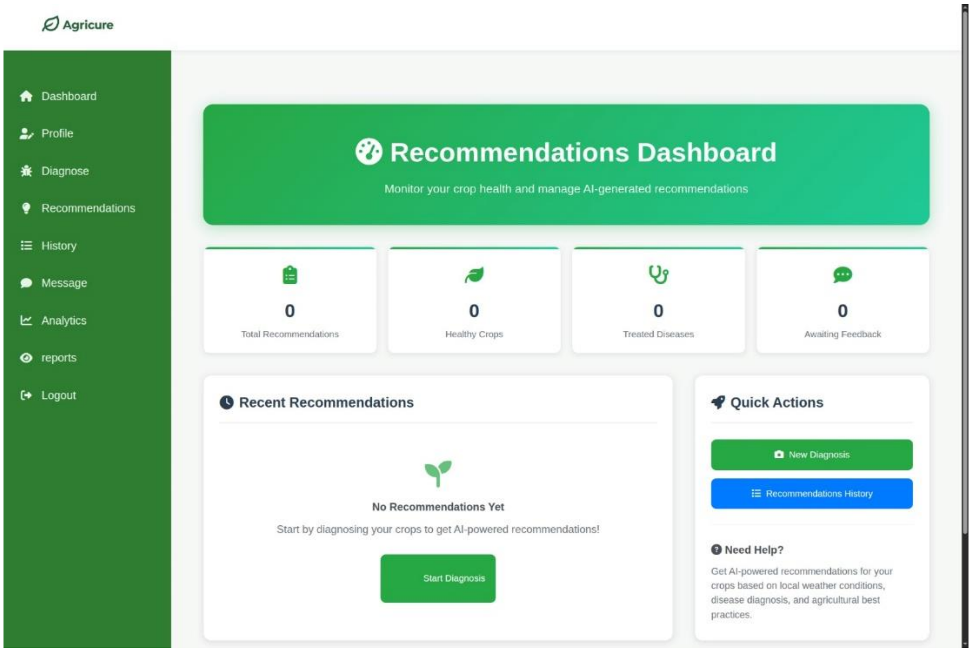


Figure 3:Figure shows the Recommendation Page for the farmer



4.2.2 Agronomist Interactions

Figure 4: Figure shows the Agronomist dashboard

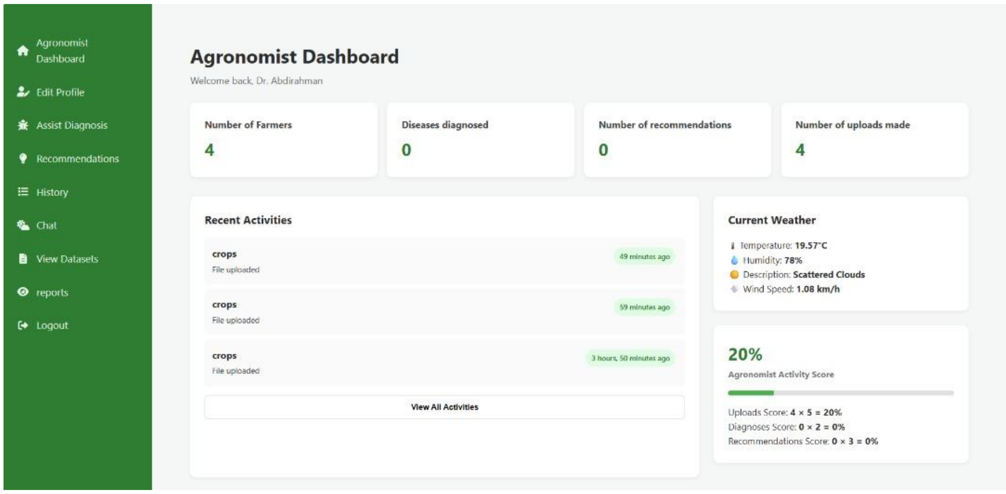
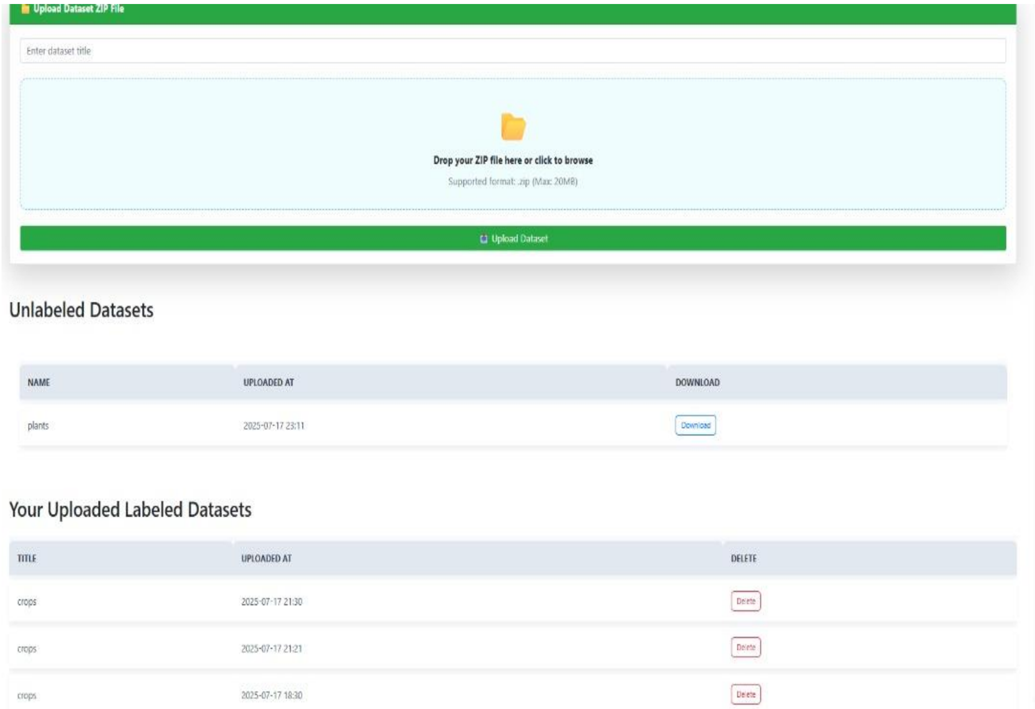


Figure 5: Figure shows the Agronomist page for labelling datasets



4.2.3 Administrator Interactions

Figure 6: Figure shows the Administrator dashboard

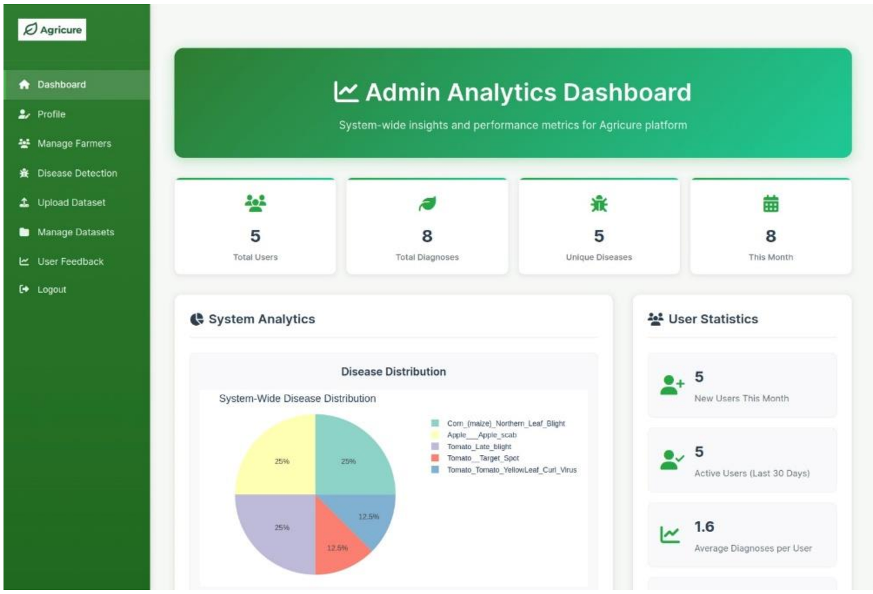
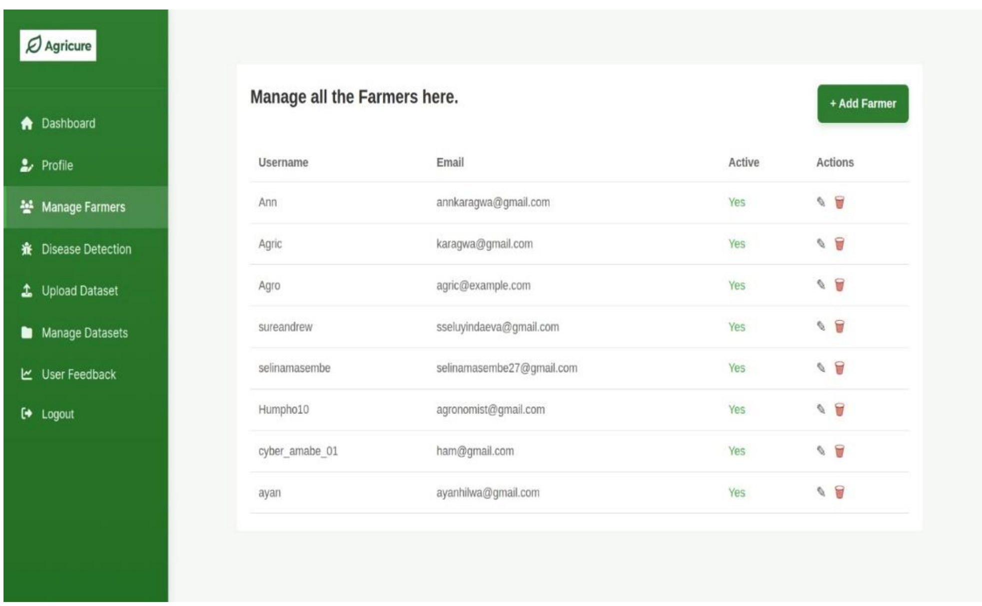


Figure 7: Figure shows the manage farmer's page.



5. CONCLUSION

The Agricure System offers a powerful solution to modernize rural agriculture by bridging technology and traditional practices. Its AI-powered disease detection and advisory features make it a valuable tool in the fight against food insecurity and crop loss.