AI APP FOR FARMER

A PROJECT REPORT

Submitted by

AJAY.A	**(20202002)**
KARRAN.J	**(20202027)**
NAVEEN.A	**(20202033)**

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

PAAVAI ENGINEERING COLLEGE, NAMAKKAL

(AUTONOMOUS)

ANNA UNIVERSITY: CHENNAI – 600025

APRIL 2024

BONAFIDE CERTIFICATE

Certified that this project report "**AI APP FOR FARMER**" is the work of "**A.AJAY (20202002)**, **J.KARRAN (20202027)** and **NAVEEN.A (20202033)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. B. VENKATESAN, M.E., Ph.D.,	**Mrs. B.DEEPA, M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Associate Professor,	Assistant Professor,
Department of Information Technology,	Department of Information Technology,
Paavai Engineering College,	Paavai Engineering College,
Namakkal - 637 018.	Namakkal - 637 018**.**

Submitted for End Semester Examinations held on ___________________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

A.AJAY (20202002), J.KARRAN (20202027), A.NAVEEN (20202033) hereby declare that the project report titled "**AI APP FOR FARMER**" done by us under the guidance of **Mrs.B.DEEPA, M.E.**, Assistant Professor at Paavai Engineering College, Pachal, Namakkal is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology in Information Technology.Certified further that, to the best of our knowledge, the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other occasion.

1.

2.

DATE : **3.**

PLACE: **SIGNATURE OF THE CANDIDATES**

ACKNOWLEDGEMENTS

A great deal of arduous work and effort has been spent in implementing this project work. Several special people have guided us and have contributed significantly to this work and so this becomes obligatory to record our thanks to them.

We express our profound gratitude to our honorable **Chairman, Shri.CA.N.V.NATARAJAN, B.Com., F.C.A.,** and also to our **Correspondent, Smt. MANGAI NATARAJAN, M.Sc.,** for providing all necessary facilities for the successful completion of our project.

We wish to express our sincere thanks to our respected **Director-Administration, Dr. K. K. RAMASAMY, M.E., Ph.D.,** for all the blessing and help provided during the period of project work.

We would like to thank our respected **Principal, Dr. M. PREM KUMAR, M.E., Ph.D.,** for allowing us to do this project and providing required time to complete the same.

We wish to express our deep gratitude to **Dr.B.VENKATESAN, M.E., Ph.D., Head of the Department,** for the able guidance and useful suggestions, which helped us for completing project work in time.

We express our sincere thanks to **Dr.G.MADASAMY RAJA, M.E, M.B.A, Ph.D.,** Project Coordinator for the useful suggestions, which helped us to complete the project work in time.

We would like to extend our sincere thanks to **Mrs.B.DEEPA, M.E.,** Supervisor, for giving us this opportunity to do this project and also for his inspiring guidance, generous help, and support.

We would like to extend our sincere thanks to **all our department staff members and to our parents** for their advice and encouragement to do the project work with full interest and enthusiasm.

TABLE OF CONTENTS

ABSTRACT

The agricultural sector plays a crucial role in sustaining livelihoods and ensuring food security worldwide. However, farmers face various challenges such as unpredictable weather patterns, pest infestations, market fluctuations, and limited access to agricultural knowledge. To address these challenges, we propose Smart Farm, an AI-assisted mobile application designed to empower farmers with actionable insights and recommendations for optimizing agricultural practices and crop selection. Smart Farm leverages the capabilities of artificial intelligence (AI) to provide personalized recommendations tailored to each farmer's specific needs and local conditions. Through the integration of machine learning algorithms, weather data analysis, soil health monitoring, and crop modeling, the application offers real-time guidance on optimal planting times, crop varieties, irrigation schedules, and pest management strategies. By integrating these components, the proposed system aims to revolutionize traditional farming practices, providing farmers with actionable insights for optimizing inputs and improving crop yields through the power of AI and data analytics. Collect diverse agricultural data, including soil composition, historical crop performance, weather conditions, and farm specific details. Utilize IoT sensors and external APIs to gather real-time data on weather patterns and soil health. Employ advanced analytics tools to process and analyze the collected data. Utilize machine learning algorithms for predictive analytics to understand relationships between different variables and make informed recommendations.

KEY INDEX: API, Data analysis, Machine learning, IOT Monitoring.

LIST OF FIGURES

LIST OF ABBREVIATIONS

S.NO	ABBREVIATION	EXPANSION
1	SVM	Support Vector Machines
2	CNN	Convolutional Neural Networks
3	RNN	Recurrent Neural Networks
4	GBM	Gradient Boosting Machines
5	PCA	Principal Component Analysis
6	IOT	Internet Of Things
7	ANN	Artificial Neural Networks
8	FL	Fuzzy Logic

CHAPTER 1
INTRODUCTION

It is ultimate solution for revolutionizing agricultural practices through the power of artificial intelligence. Designed specifically for farmers, Farm AI combines cutting-edge technology with deep agricultural expertise to optimize every aspect of your farming operation.

These are essential components of modern farming, and Farm AI's Crop Monitoring and Analysis feature is designed to provide farmers with valuable insights into the health, growth, and productivity of their crops. With Farm AI, farmers can monitor their crops in real-time, receiving updates on a regular basis about various crop parameters such as vegetation indices, soil moisture levels, and temperature gradients. This continuous monitoring enables early detection of potential issues and allows for timely intervention to prevent crop losses. Say goodbye to unpredictable weather woes. Farm AI leverages weather data and predictive analytics to forecast local weather conditions with high accuracy.

Plan your planting, irrigation, and harvesting schedules with confidence, minimizing weather-related risks. Using advanced analytics and machine learning algorithms, Farm AI analyzes historical weather patterns and current atmospheric conditions to generate highly accurate weather forecasts for your specific location. These forecasts can range from short-term predictions covering the next few hours to long-term forecasts spanning weeks or months. Farm AI gathers real-time weather data from various sources, including meteorological stations, satellites, and weather APIs. This data includes information such as temperature, humidity, precipitation, wind speed, and direction.

Precision farming, also known as precision agriculture, is a farming management concept that utilizes technology to optimize the efficiency, productivity, and sustainability of agricultural practices. Farm AI's Precision Farming feature integrates a range of technologies and data analytics to enable farmers to make informed decisions tailored to their specific fields and crops. Farm AI generates high-resolution soil maps using satellite imagery and soil sensors, providing farmers with detailed insights into soil fertility, moisture levels, pH levels, and other soil properties. By analysing these soil maps, farmers can optimize fertilizer application, irrigation scheduling, and crop selection to maximize yields and minimize environmental impact.

Crop disease identification is a critical aspect of modern farming, and Farm AI's Crop Disease Identification feature leverages advanced image recognition technology to help farmers detect and diagnose plant diseases accurately and efficiently. Farm AI utilizes sophisticated image recognition algorithms trained on vast datasets of crop images to identify signs of diseases, pests, and nutrient deficiencies in plants. These algorithms analyse images captured by drones, smartphones, or other imaging devices to detect visual symptoms indicative of specific diseases. Farm AI's image recognition algorithms can identify a wide range of symptoms associated with crop diseases, including leaf spots, lesions, discoloration, wilting, stunted growth, and deformities. By analysing the size, shape, colour, texture, and distribution of these symptoms, it can accurately classify the type of disease affecting the crop.

Farm AI aggregates real-time market data from various sources, including commodity exchanges, market reports, trade publications, and government databases. This data encompasses information about commodity prices, supply and demand trends, export-import statistics, and geopolitical factors affecting agricultural markets. It conducts historical analysis of market data to identify patterns, trends, and seasonality in commodity prices. By analysing historical price data over multiple years, Farm AI can provide farmers with insights into long-term market trends and cyclical patterns, enabling them to anticipate market movements and plan accordingly. Farm AI helps farmers manage price risk by providing them with actionable recommendations for hedging strategies, forward contracts, and marketing decisions. By understanding the potential impact of price fluctuations on their revenue and profitability, farmers can mitigate price risk and protect their financial interests.

1.1 MACHINE LEARNING ALGORITHMS

Integrating machine learning algorithms into an app for farmers can offer valuable insights and decision support across various aspects of farm management.

1.1.1 Decision Trees

Decision trees can analyse historical data on crop performance, soil characteristics, climate conditions, and market trends to recommend the most suitable crops for a farmer's specific location and conditions. The decision tree model can classify crops based on factors such as soil type, temperature, precipitation, and crop rotation requirements, providing personalized recommendations to farmers.

Decision trees can be trained on sensor data (e.g., images of crops, environmental variables) to detect symptoms of diseases, pests, or nutrient deficiencies in crops. By analysing features such as leaf colour, shape, and texture, the decision tree model can classify crop health status and provide early warnings to farmers about potential threats to their crops. Decision trees can predict weather conditions and assess the risk of extreme weather events (e.g., droughts, floods, frost) based on historical weather data and meteorological forecasts.

By analysing factors such as temperature, humidity, wind speed, and precipitation, the decision tree model can help farmers plan their irrigation, pest control, and harvesting activities more effectively and mitigate the impact of adverse weather conditions on their crops. Decision trees can analyse soil sensor data (e.g., pH, moisture, nutrient levels) and historical crop performance data to diagnose soil health issues and recommend appropriate soil management practices. By considering factors such as soil type, fertility, and drainage, the decision tree model can provide tailored recommendations for soil amendments, fertilizer applications, and crop rotation strategies to improve soil health and crop yields. Decision trees can predict market prices for agricultural commodities based on historical price data, supply-demand dynamics, economic indicators, and geopolitical factors. By analysing features such as seasonal trends, price volatility, and market sentiment, the decision tree model can help farmers make informed decisions about when to sell their crops and optimize their revenue generation.

Decision trees can forecast crop yields and harvest timing based on factors such as weather conditions, crop growth stages, and historical yield data. By considering features such as temperature, rainfall, growing degree days, and crop phenology, the decision tree model can provide farmers with accurate predictions of their expected harvest yields and timing, allowing them to plan their harvesting activities more efficiently.

1.6.2 Support Vector Machines (SVM)

Support Vector Machines (SVM) are powerful machine learning algorithms commonly used for classification and regression tasks.

SVM can classify crops based on various features such as soil type, climate conditions, and historical crop performance data. By analysing these features, SVM can help farmers identify the most suitable crops to plant in their fields, taking into account factors like soil fertility, water availability, and climate suitability. SVM can classify crops as healthy or diseased based on sensor data such as images of crop leaves or environmental variables. By analysing features like leaf color, texture, and shape, SVM can detect symptoms of diseases or pest infestations early, allowing farmers to take prompt action to mitigate the spread of diseases and minimize crop losses. SVM can predict weather conditions and assess the risk of extreme weather events such as droughts, floods, or frost. By analyzing historical weather data and meteorological forecasts, SVM can help farmers anticipate potential weather-related risks and plan their farming activities accordingly, such as adjusting irrigation schedules or implementing protective measures for their crops.

SVM can analyze soil sensor data to assess soil health and fertility. By considering features such as pH levels, nutrient concentrations, and soil texture, SVM can help farmers identify soil deficiencies or imbalances and recommend appropriate soil management practices, such as fertilization or soil amendments, to improve soil health and optimize crop yields. SVM can predict market prices for agricultural commodities based on historical price data, supply-demand dynamics, and economic indicators.

By analyzing features like seasonal trends, market volatility, and external factors such as government policies or trade agreements, SVM can help farmers make informed decisions about when to sell their crops and maximize their profits. SVM can forecast crop yields and harvest timing based on factors such as weather conditions, crop growth stages, and historical yield data. By analyzing features like temperature, rainfall, and growing degree days, SVM can provide farmers with accurate predictions of their expected harvest yields and timing, allowing them to plan their harvesting activities more efficiently.

1.1.3 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNNs) are a class of deep learning models commonly used for image recognition and classification tasks.

CNNs can analyze images of crop leaves or fruits to detect symptoms of diseases, pests, or nutrient deficiencies. By learning features such as patterns, textures, and colors associated with different diseases or pests, CNNs can accurately classify the health status of crops and provide early warnings to farmers about potential threats. CNNs can identify and classify weed species in agricultural fields based on images captured by drones or cameras. By distinguishing between crops and weeds, CNNs can help farmers implement targeted weed control measures, such as precision spraying or mechanical removal, to reduce weed competition and minimize herbicide use. CNNs can analyze satellite or drone imagery to monitor crop growth and development throughout the growing season.

By tracking features such as plant height, canopy density, and leaf color, CNNs can assess crop health, identify areas of stress or nutrient deficiency, and provide insights into crop management practices, such as irrigation or fertilization. CNNs can estimate crop yields by analyzing images of fields or individual plants captured by drones or smartphones. By correlating visual features with yield indicators such as plant density, fruit count, or grain size, CNNs can provide accurate predictions of expected yields, helping farmers make informed decisions about harvest planning and marketing strategies. CNNs can analyze aerial or satellite imagery to detect signs of soil erosion, such as gullies, rills, or sediment deposition.

By identifying areas of erosion risk, CNNs can assist farmers in implementing soil conservation practices, such as contour ploughing, cover cropping, or terracing, to protect soil health and minimize erosion. CNNs can assess the quality of harvested crops by analyzing images of fruits, vegetables, or grains. By recognizing visual characteristics such as size, shape, color, and defects, CNNs can classify crops into different grades or categories based on predefined quality standards, helping farmers sort and market their produce more effectively.

1.1.4 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are a class of deep learning models commonly used for sequence data processing tasks

RNNs can analyze historical weather data, such as temperature, humidity, and precipitation, to forecast future weather conditions. By capturing temporal dependencies in the data, RNNs can learn patterns and trends in weather patterns, enabling farmers to make informed decisions about irrigation, pest control, and other farming activities. RNNs can process time-series data on crop growth parameters, such as plant height, leaf area, and chlorophyll content, to monitor the development of crops throughout the growing season. By tracking changes in crop phenology over time, RNNs can help farmers assess crop health, predict harvest timing, and optimize management practices. RNNs can analyze historical data on crop yields, weather conditions, soil properties, and management practices to predict future crop yields.

By capturing complex relationships between these factors, RNNs can provide accurate estimates of expected yields, enabling farmers to plan harvest logistics, negotiate contracts, and manage financial risks. RNNs can process time-series data on pest and disease occurrences, environmental conditions, and crop management practices to forecast the likelihood of pest and disease outbreaks. By identifying patterns and correlations in the data, RNNs can provide early warnings to farmers, allowing them to implement timely interventions and minimize crop losses. RNNs can analyze historical price data for agricultural commodities, market trends, economic indicators, and external factors to predict future market prices.

By capturing temporal dependencies and nonlinear relationships in the data, RNNs can provide accurate forecasts of commodity prices, helping farmers make informed decisions about crop marketing and sales. RNNs can analyze historical data on crop rotations, soil health indicators, and yield outcomes to optimize crop rotation schedules. By learning from past experiences and adapting to changing conditions, RNNs can help farmers plan crop rotations that improve soil fertility, reduce pest and disease pressure, and maximize long-term productivity.

1.1.5 Gradient Boosting Machines (GBM)

Gradient Boosting Machines (GBM) is a machine learning technique that builds predictive models in the form of an ensemble of weak prediction models, typically decision trees.

GBM can analyze historical data on crop yields, weather conditions, soil properties, and management practices to predict future crop yields. By capturing complex relationships and interactions between these factors, GBM models can provide accurate estimates of expected yields, enabling farmers to plan their harvest logistics, negotiate contracts, and manage financial risks more effectively. GBM can analyze sensor data or images of crops to detect symptoms of diseases, pests, or nutrient deficiencies. By learning from labeled examples, GBM models can identify patterns indicative of diseases or pest infestations and provide real-time alerts to farmers, enabling them to take timely action to protect their crops.

GBM can predict weather conditions and assess the risk of extreme weather events such as droughts, floods, or frost based on historical weather data. By capturing nonlinear relationships and interactions between weather variables, GBM models can provide accurate forecasts, helping farmers anticipate and mitigate potential risks to their crops. GBM can analyze soil sensor data and historical crop performance to diagnose soil health issues and recommend soil management practices. By capturing nonlinear relationships between soil properties and crop yields, GBM models can provide personalized recommendations for soil amendments, fertilizer applications, and crop rotation strategies to improve soil health and optimize crop productivity.

1.1.6 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique commonly used in machine learning and statistics.

PCA can analyze soil sensor data, such as pH levels, nutrient concentrations, and soil texture, to identify underlying patterns and relationships between different soil properties. By reducing the dimensionality of the data while retaining as much variance as possible, PCA can help farmers visualize and interpret complex soil datasets, allowing them to assess soil health and fertility more effectively. PCA can analyze spectral data from remote sensing imagery or handheld sensors to assess the quality of harvested crops. By reducing the dimensionality of the spectral data and extracting meaningful features, PCA can help farmers classify crops into different quality grades or categories based on factors such as moisture content, sugar content, or disease incidence, enabling them to sort and market their produce more effectively.

1.2 Implementation

First collect relevant data from various sources, including agricultural databases, weather stations, remote sensing satellites, soil sensors, and market databases. Preprocess the collected data to clean, normalize, and transform it into a suitable format for analysis and modeling. Split the data into training, validation, and test sets for model development and evaluation. Split the data into training, validation, and test sets for model development and evaluation.

Choose appropriate machine learning algorithms and techniques based on the app's objectives and data characteristics. Develop and train machine learning models for each feature of the app, such as decision trees, support vector machines, convolutional neural networks, Gradient Boosting Machines, Principal Component Analysis Fine-tune the models and optimize hyperparameters to improve performance and generalization ability. Integrate the machine learning models and algorithms into the app's backend infrastructure, allowing for real-time data processing and analysis.

1.3 Objective

The objective of implementing an AI app for farmers is to leverage advanced technologies to address key challenges and optimize agricultural practices. The app aims to provide farmers with valuable insights, decision support, and actionable recommendations across various aspects of farm management. By integrating machine learning algorithms and artificial intelligence techniques, the app can analyze diverse datasets, including weather data, soil health indicators, crop performance metrics, and market trends, to deliver personalized recommendations and predictions tailored to the specific needs and conditions of individual farmers. Through intuitive user interfaces, interactive features, and real-time updates, the app seeks to empower farmers with the knowledge.

CHAPTER 2

LITERATURE SURVEY

2.1 Title: Enabling Scalable Smart Vertical Farming with IoT and Machine Learning Technologies

Author: Technologies Hegedus, Csaba & Frankó, Attila & Varga, Pal & Gindl, Stefan & Tauber, Markus.

Year: (2023)

As global challenges of population growth, climate change, and resource scarcity intensify, the agricultural landscape is at a critical juncture. Sustainable vertical farming emerges as a transformative solution to address these challenges by maximizing crop yields in controlled environments. This paradigm shift necessitates the integration of cutting-edge technologies, with Artificial Intelligence (AI) at the forefront. The paper provides a comprehensive exploration of the role of AI in sustainable vertical farming, investigating its potential, challenges, and opportunities. The review synthesizes the current state of AI applications, encompassing machine learning, computer vision, the Internet of Things (IoT), and robotics, in optimizing resource usage, automating tasks, and enhancing decision-making. It identifies gaps in research, emphasizing the need for optimized AI models, interdisciplinary collaboration, and the development of explainable AI in agriculture. The implications extend beyond efficiency gains, considering economic viability, reduced environmental impact, and increased food security. The paper concludes by offering insights for stakeholders and suggesting avenues for future research, aiming to guide the integration of AI technologies in sustainable vertical farming for a resilient and sustainable future in agriculture.

2.2.Title: Artificial Intelligence Enhanced - Plant and Soil Monitoring Systems.

Author: Aouthithiye Barathwaj SR Y; Haariharan N C; Krishnakanth L; M. Soumya

Year:2023

In today's fast-paced world, where technological advances continue to transform various industries, agriculture is no exception. With the growing demand for sustainable food production and efficient resource management, there is an urgent need for innovative solutions that can transform traditional agricultural practices. Enter the world of artificial intelligence (AI) and machine learning (ML), which offer unprecedented possibilities in plant monitoring systems. Gone are the days of manual observations and guesswork to understand plant health and growth patterns. With the help of AI and ML, farmers and scientists can now immerse themselves in a new era of precision agriculture, where the well-being of each plant is monitored, analyzed and optimized in real-time.

2.3. Title: Artificial Intelligence Cultivation: Transforming Agriculture for a Smart and Sustainable Future.

Author: Bhushan Fulkar, Samir Mendhe, Pawan Patil.

Year:2022

The growing global population is posing unprecedented challenges for the agriculture industry, requiring it to produce more food while also addressing resource constraints and environmental concerns. Artificial intelligence (AI) has changed the game in agriculture by offering creative solutions to increase productivity, optimize resource use, and promote sustainability. Artificial intelligence (AI) has been used in the agriculture sector more and more recently. A few of the problems the industry faces in attempting to maximize its produce are improper soil treatment, disease and pest infestation, the use of driverless tractors, big data requirements, and the knowledge gap between farmers and technology. This work provides an overview of the applications of artificial intelligence in crop, weed, and disease management. The proposed work aims to review various AI techniques, including artificial neural networks (ANN) and fuzzy logic (FL).

2.4. Title: AI and ML for Enhancing Crop Yield and Resource Efficiency in Agriculture

Author: Ehtesham Siddiqui, Mohammed Siddique, Safeer Pasha M, Prasanthi Boyapati, Pavithra G, Natrayan L

Year:2021

We investigate how AI and ML might revolutionize the agricultural industry, particularly with regard to increasing crop output while decreasing input costs. Applying AI and ML technology has promise in a society struggling with population increase, climate change, and resource constraints. This study highlights the practical advantages of AI and ML in agriculture via a wellcrafted research process, including data gathering, model creation, and assessment. The results show that AI and ML models are useful for forecasting agricultural yields, identifying illnesses, allocating resources efficiently, and assisting farmers with decision-making based on empirical evidence. Results like this highlight the importance of these technologies in advancing goals of efficiency, sustainability, and food safety. Additionally, the study acknowledges the significance of addressing ethical problems in AI deployment, guaranteeing equal access to these advancements. We should expect to see more research into cutting-edge methods, Internet of Things (IoT) integration, and accessible tools for subsistence farmers as we go further in the use of AI and ML in the agricultural sector. The full promise of AI and ML in designing a resilient, productive, and sustainable agricultural future requires collaborative efforts across stakeholders. In the struggle to feed the globe while protecting its resources, this study shines a bright light of optimism.

2.5 Tilte: A Review of Artificial Intelligence-Based Techniques in the Diagnosis of Chronic Obstructive Pulmonary Disease

Author: Jasneet Chawla, Navpreet Kaur Walia

Year:2021

The Chronic Respiratory Diseases are characterized by obstructed and inflamed air passages. Chronic obstructive pulmonary disease has emerged as a socioeconomic health burden globally. The severity of the disease is projected to escalate in the upcoming decades. Techniques based on Artificial Intelligence have proved useful in the healthcare industry. The tremendous amount of heterogeneous data accumulated in the repositories of hospitals, if incorporated wisely, can be used to build up tech-aided systems. In this work, the author has reviewed the Artificial Intelligence-based techniques available in the literature that has made the effective diagnosis of the disease evident.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Proposed system

Agriculture is one of the most important sectors of the global economy, and it is essential to ensure food security for the growing population. However, agriculture is facing several challenges, including climate change, water scarcity, and the need to increase productivity. Artificial Intelligence (AI) is emerging as a promising solution to address these challenges. AI can help farmers make smarter decisions, reduce waste, and increase productivity. We will explore the advantages of AI in agriculture, including Pest and Disease Detection and Control, Efficient Resource Utilization, and Enhanced Crop Monitoring and Remote Sensing.

3.1.1 Advantages

- Precision Farming
- Crop Monitoring and management
- Field Insights
- Feedback mechanism

3.2 Economical feasibility:

Cost-Benefit Analysis: Conduct a cost-benefit analysis to assess the financial viability of developing and deploying the AI application. Consider factors such as development costs, hardware/software expenses, maintenance costs, and potential revenue or cost savings. Return on Investment (ROI): Estimate the expected ROI of the AI application in terms of increased productivity, yield improvements, cost reductions, and other tangible benefits for farmers.

3.3 Technical Feasibility

Assess the availability and quality of data required for training AI models. This includes data on weather patterns, soil conditions, crop types, historical yields, pest and disease occurrences. Determine if the proposed AI application can be deployed on existing hardware and software platforms commonly used by farmers, or if additional infrastructure upgrades are needed.

3.4 Environmental Feasibility

Environmental Impact: Evaluate the environmental implications of the AI application, including its potential to reduce resource consumption, minimize chemical use, and promote sustainable farming practices.

3.5 Social Feasibility

This study consider the social impact of the AI application on farmers' livelihoods, employment opportunities, and community well-being. Ensure that the technology benefits all stakeholders and promotes inclusive growth.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 SOFTWARE SYSTEM CONFIGURATION

- Operating System: Windows 10 or 11
- Application : Mobile Application
- IDE : Flutter & VSCODE, Android Studio

4.2 HARDWARE SYSTEM CONFIGURATION

- CPU : 4 core
- RAM : 8 GB (min)

- Hard Disk : 20 GB

4.3 SOFTWARE ENVIROMENT

To write and compile Flutter code for desktop, you must have the following version of Windows and the listed software packages.

4.3.1 Operating system

Flutter supports 64-bit version of Microsoft Windows 10 or later. These versions of Windows should include the required Windows PowerShell 5.0 or later.

4.3.2 Development tools

Download and install the Windows version of the following packages:

- Git for Windows 2.27 or later to manage source code.
- Visual Studio 2022 to debug and compile native C++ Windows code. Make sure to install the **Desktop development with C++** workload. This enables building Windows app including all of its default components. **Visual Studio** is an IDE separate from **Visual Studio *Code***.

You can build apps with Flutter using any text editor or integrated development environment (IDE) combined with Flutter's command-line tools.

Using an IDE with a Flutter extension or plugin provides code completion, syntax highlighting, widget editing assists, debugging, and other features.

Popular options include:

- Visual Studio Code 1.77 or later with the Flutter extension for VS Code.
- Android Studio 2023.1 (Hedgehog) or later with the Flutter plugin for
- IntelliJ.

- IntelliJ IDEA 2023.1 or later with the Flutter plugin for IntelliJ.
- Download the following installation bundle to get the latest stable release of the Flutter SDK.
- For other release channels, and older builds, check out the SDK archive.

Run Flutter doctor

The flutter doctor command validates that all components of a complete Flutter development environment for Windows.

1.Open PowerShell.

2.To verify your installation of all the components, run the following command. **C: \> flutter doctor**

4.4 Flutter

Flutter is an open-source UI software development kit (SDK) created by Google for building natively compiled applications for mobile, web, and desktop from a single codebase.

Flutter allows developers to write code once and deploy it across multiple platforms, including iOS, Android, web, and desktop. This is achieved through a single codebase, reducing development time and effort compared to building separate apps for each platform. Flutter utilizes a reactive, widget based architecture where everything is a widget, including structural elements like buttons and containers, as well as complex layouts and entire screens. This makes it easy to compose rich and interactive user interfaces using prebuilt and custom widgets. One of Flutter's standout features is hot reload, which allows developers to instantly see changes made to the code reflected in the app UI without restarting the app or losing its state.

This rapid iteration cycle significantly speeds up the development process and enhances productivity. Flutter apps are compiled to native machine code, providing near-native performance and fluid animations. The Flutter framework includes a high-performance rendering engine called Skia, which ensures smooth UI rendering across different devices and platforms. Flutter comes with a comprehensive set of customizable widgets and libraries for building beautiful and feature-rich user interfaces. These include material design widgets for Android and Cupertino widgets for iOS, as well as third-party packages for additional functionality.

Flutter provides access to plat form specific APIs and features through platform channels, allowing developers to integrate native functionality such as camera, location, and sensors into their Flutter apps. This enables developers to leverage the full capabilities of each platform while maintaining a single codebase. Flutter has a vibrant and rapidly growing ecosystem with a thriving community of developers, contributors, and third-party packages. The Flutter ecosystem includes tools, plugins, and resources to streamline development, enhance productivity, and address various use cases and industries. As an open-source project backed by Google, Flutter benefits from regular updates, improvements, and support from the tech giant. Google actively contributes to the development of Flutter and promotes its adoption across different platforms and industries.

4.5 Android studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development. Based on the powerful code editor and developer tools from IntelliJ IDEA , Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system

- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Live Edit to update composable in emulators and physical devices in real time
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support

Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

This page provides an introduction to basic Android Studio features. For a summary of the latest changes, see the Android Studio release notes.

4.5.1 Project structure:

Each project in Android Studio contains one or more modules with source code files and resource files. The types of modules include:

- Android app modules
- Library modules
- Google App Engine modules

By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files. All the build files are visible at the top level, under Gradle Scripts.

java: Contains the Kotlin and Java source code files, including JUnit test code.res: Contains all non-code resources such as UI strings and bitmap images. The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select Project instead of Android from the Project menu.

4.5.2 Gradle build system

Android Studio uses Gradle as the foundation of the build system, with more Android-specific capabilities provided by the Android Gradle plugin. This build system runs as an integrated tool from the Android Studio menu and independently from the command line. You can use the features of the build system to do the following:

Customize, configure, and extend the build process. Create multiple APKs for your app with different features, using the same project and modules. Reuse code and resources across source sets. By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named build.gradle.kts if you use Kotlin (recommended) or build.gradle if you use Groovy. They are plain text files that use the Kotlin or Groovy syntax to configure the build with elements provided by the Android Gradle plugin. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

Build variants: The build system can help you create different versions of the same app from a single project. This is useful when you have both a free version and a paid version of your app or if you want to distribute multiple APKs for different device configurations on Google Play. Multiple APK support: Multiple APK support lets you efficiently create multiple APKs based on screen density or ABI. For example, you can create separate APKs of an app for the hdpi and mdpi screen densities, while still considering them a single variant and letting them share test APK, javac, dx, and ProGuard settings   Resource shrinking: Resource shrinking in Android Studio automatically removes unused resources from your packaged app and library dependencies. For example, if your app uses Google Play services to access Google Drive functionality, and you are not currently using Google Sign-In, then resource shrinking can remove the various drawable assets for the Sign-In-Button buttons.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

Software architecture involves the high-level structure of software system abstraction, by using decomposition and composition, with architectural style and quality attributes. A software architecture design must conform to the major functionality and performance requirements of the system, as well as satisfy the non-functional requirements such as reliability, scalability, portability, and availability. Software architecture must describe its group of components, their connections, interactions among them and deployment configuration of all components.

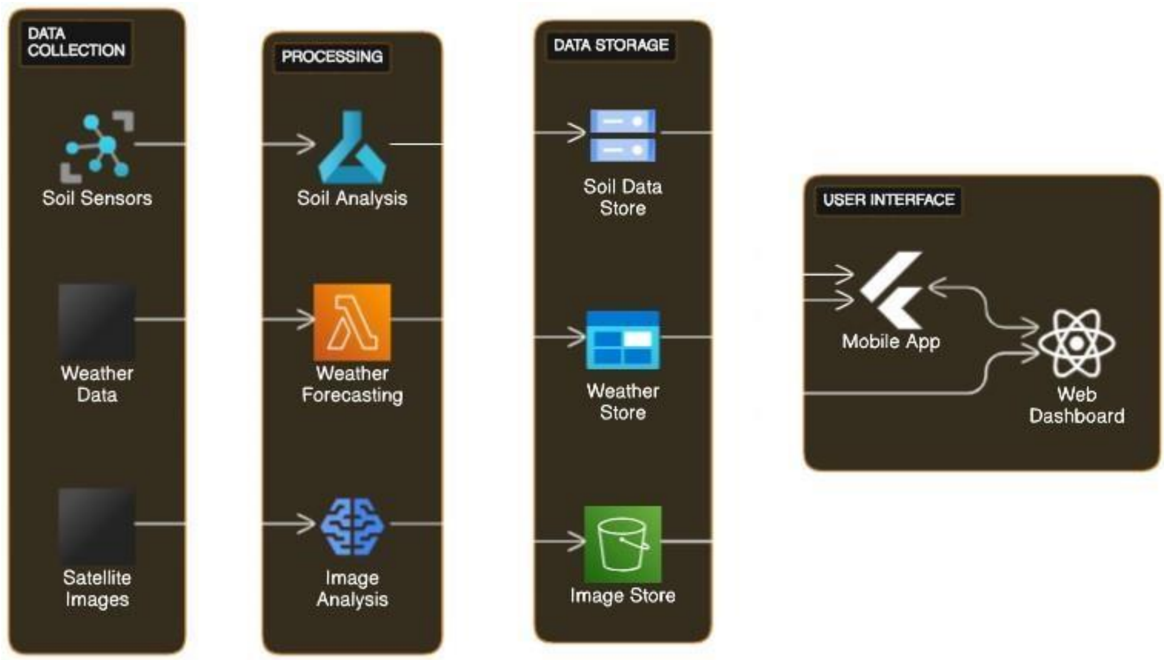

Fig 5.1 Software architecture

5.2.1 Data Collection Layer

Sensors and IoT Devices: Collect data from various sources such as soil moisture sensors, weather stations, drones, satellite imagery, and farm machinery equipped with sensors. Clean, aggregate, and preprocess raw data to make it suitable for analysis. This may involve data filtering, normalization, and feature extraction.

5.2.2 Data Storage Layer

Store pre processed data in a centralized database or data warehouse. Use scalable and efficient database systems capable of handling large volumes of data. Optionally, store raw data in data lakes for long-term storage and future analysis.

5.2.3 AI Model Development Layer

Set up infrastructure for training AI models, including hardware accelerators (e.g., GPUs, TPUs) and scalable compute resources. Annotate training data for supervised learning tasks. Use tools and platforms for data labelling to generate high-quality labelled datasets. Develop pipelines for training AI models using frameworks like TensorFlow, PyTorch, or scikit-learn. Implement distributed training for scalability. Use techniques such as grid search or Bayesian optimization to tune hyperparameters and optimize model performance.

5.2.4 Application Layer

Develop user-friendly interfaces for farmers to interact with the AI application. This may include web dashboards, mobile apps, or command-line interfaces. Integrate with external systems such as farm management software, supply chain management systems, and weather forecasting services. Implement mechanisms for collecting feedback from users to improve the accuracy and relevance of AI predictions and recommendations.

5.2.5 Security and Compliance Layer

Implement encryption, access controls, and authentication mechanisms to protect sensitive data from unauthorized access. Ensure compliance with relevant data privacy regulations and industry standards for data security and ethics.

5.2.6 Monitoring and Analytics

Monitor system performance metrics, such as response time, throughput, and resource utilization, to identify bottlenecks and optimize performance. Provide farmers with analytics and reporting features to track key performance indicators (KPIs), evaluate the impact of AI recommendations, and make data driven decisions.

CHAPTER 6

PROJECT DESCRIPTION

6.1 MODULES DESCRIPITION

A module is a separate unit of software or hardware. Typical characteristics of modular components include portability, which allows them to be used in a variety of systems and interoperability**.**

Module 1: Login Page

Module 2: Home Page

Module 3: Farm management

Module 4: IoT monitoring

Module 5: Diseases detect

Module 6: My profile

6.1.1 LOGIN PAGE

- If you're new to Farm AI and ready to revolutionize your farming practices, click on the Sign-Up button to create your account and unlock the power of AI for your farm.

- Thank you for choosing Farm AI.
- Together, let's cultivate a smarter, more sustainable future for agriculture.

Login

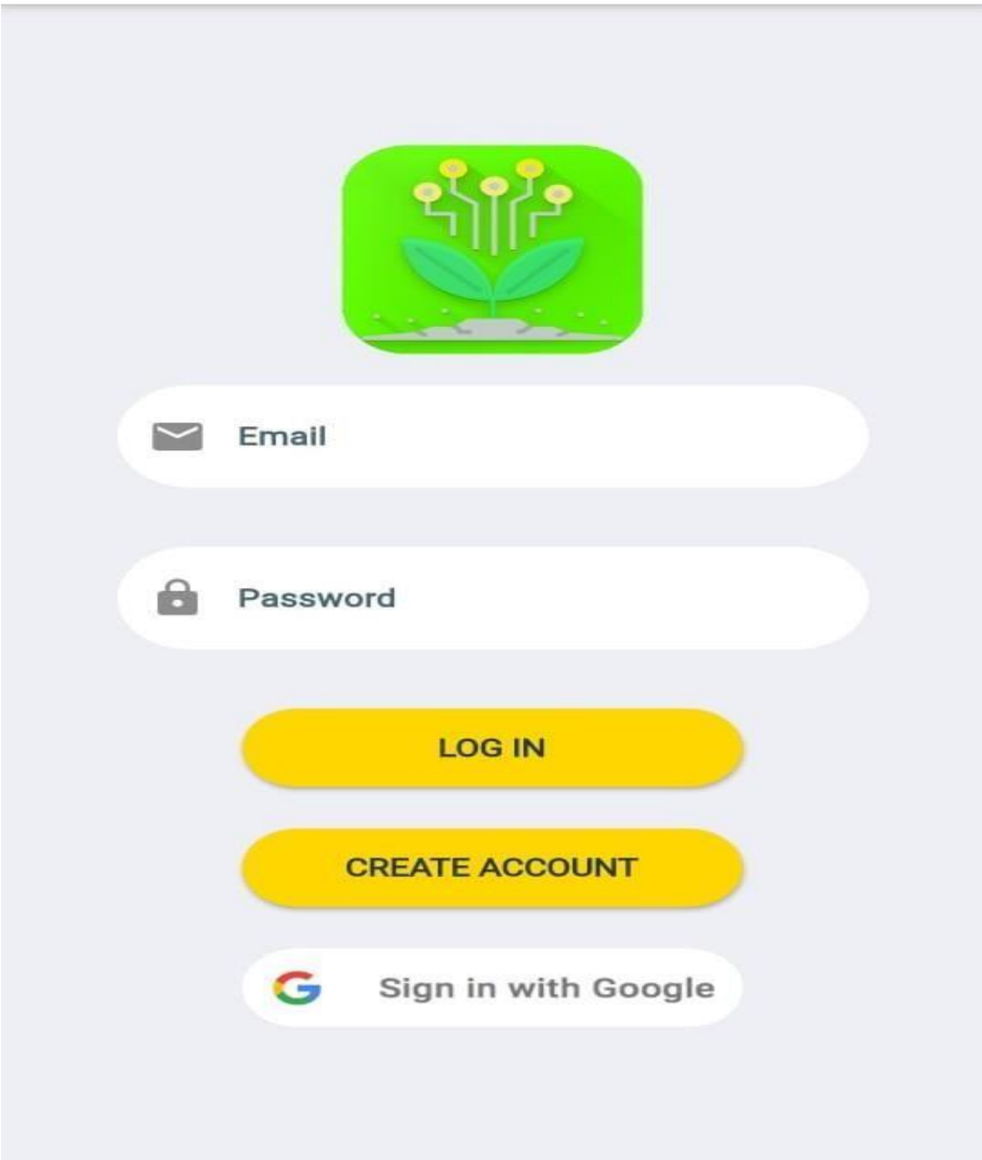

FIG 6.1 LOGIN PAGE

6.1.2 FARM MANAGEMENT

By implementing effective farm management practices, farmers can optimize production, enhance profitability, mitigate risks, and promote environmental sustainability to ensure the long-term success and viability of their agricultural operations. Farm management begins with strategic planning, which involves setting goals, defining objectives, and developing long-term plans for the farm's operations. This includes decisions about crop selection, livestock management, land use, equipment investment, and market positioning based on factors such as soil quality, climate, market demand, and financial resources.

- To know the harvesting and store related activities
- Knowledge about current agriculture technology
- The statistics based on your harvesting

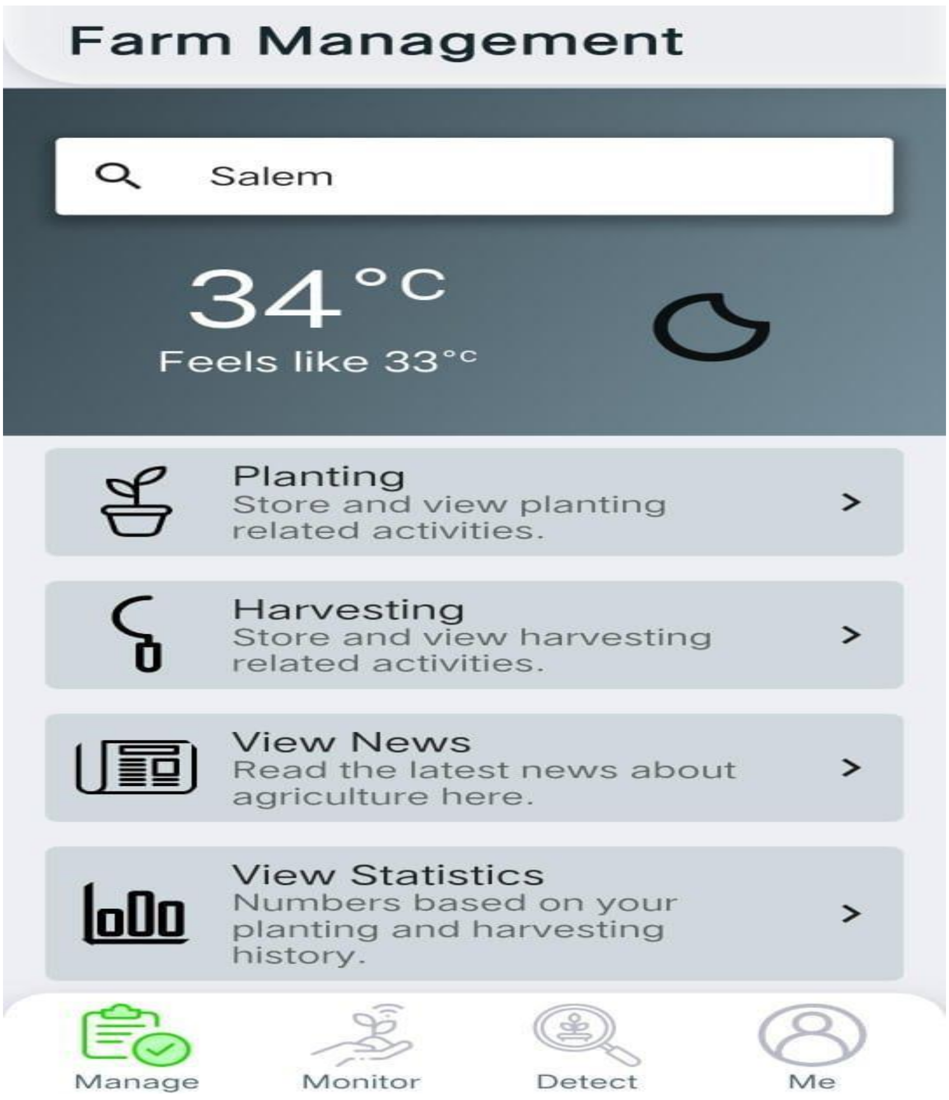

FIG 6.2 FARM MANAGEMENT PAGE

6.1.3 MANAGE PLANTING

- To create the planting record
- Able to view the planting and their previous record

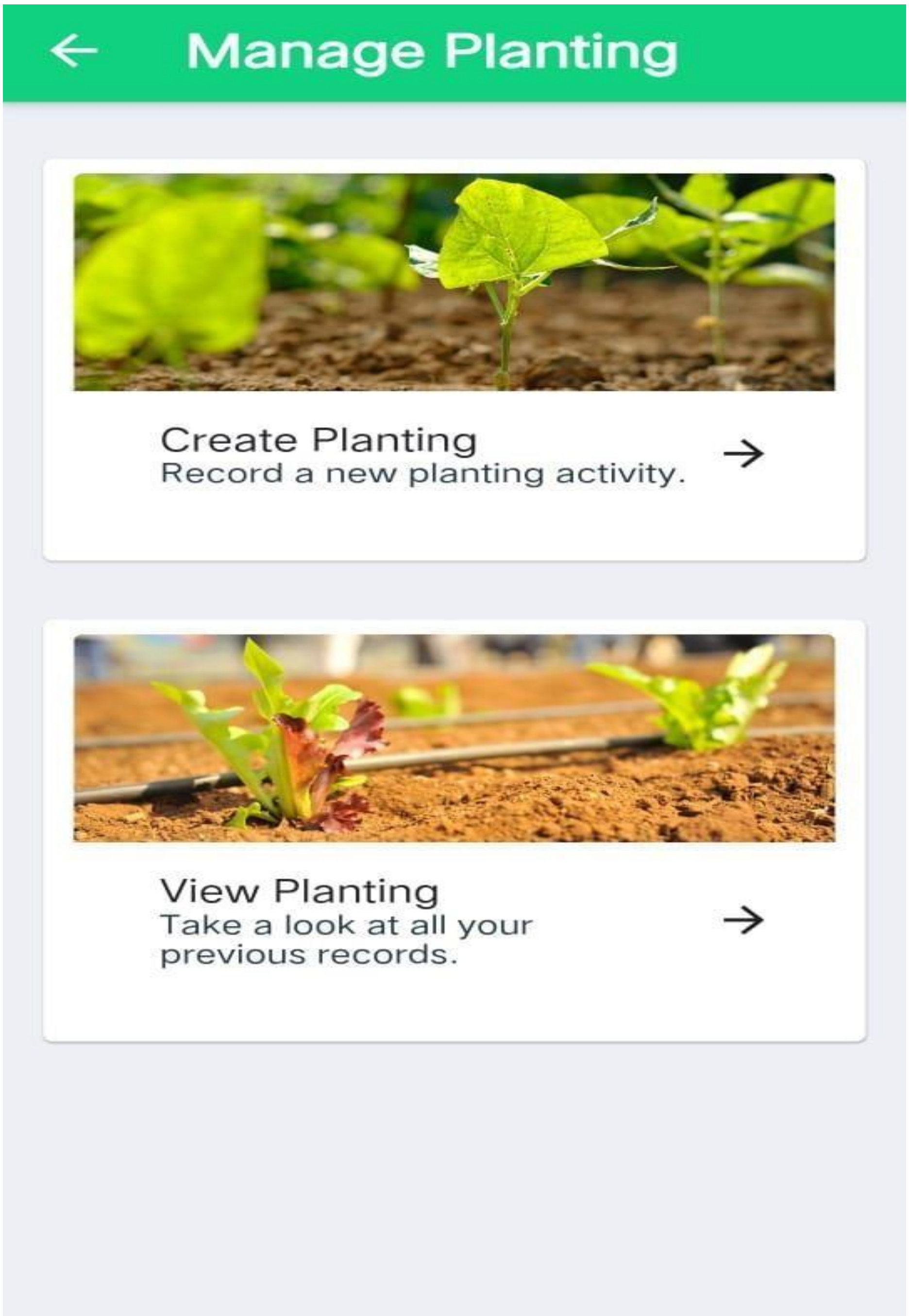

FIG 6.3 MANAGE PLANTING PAGE

6.1.4 IOT MONITIRING

IoT (Internet of Things) monitoring refers to the process of remotely observing and managing connected devices and sensors in real-time to collect data, analyse performance, and ensure proper functioning.

- It can help to prevent from us weather condition.
- Likewise, air temperature and air humidity and rain.

FIG 6.4 IOT MONITIRING

6.1.5 DISEASES DETECTION

Disease detection using IoT involves leveraging connected sensors, devices, and data analytics to identify symptoms or patterns indicative of diseases in various contexts, such as healthcare, agriculture, and environmental monitoring.

- This page can able to see the diseases of the plant
- At the same time to give predication for future treatment for the diseases plant

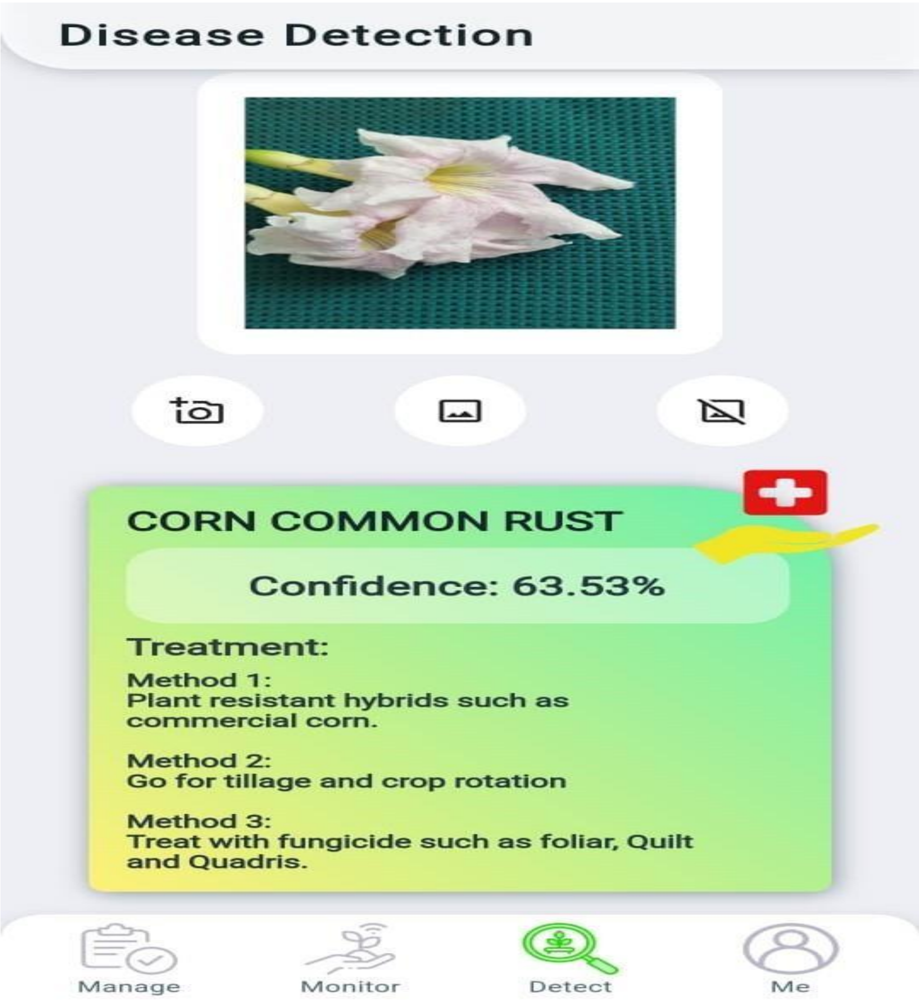

FIG 6.5 DISEASES DETECTION

6.1.5 DISEASES DETECTION

- This page can able to see the diseases of the plant
- At the same time to give predication for future treatment for the diseases plant

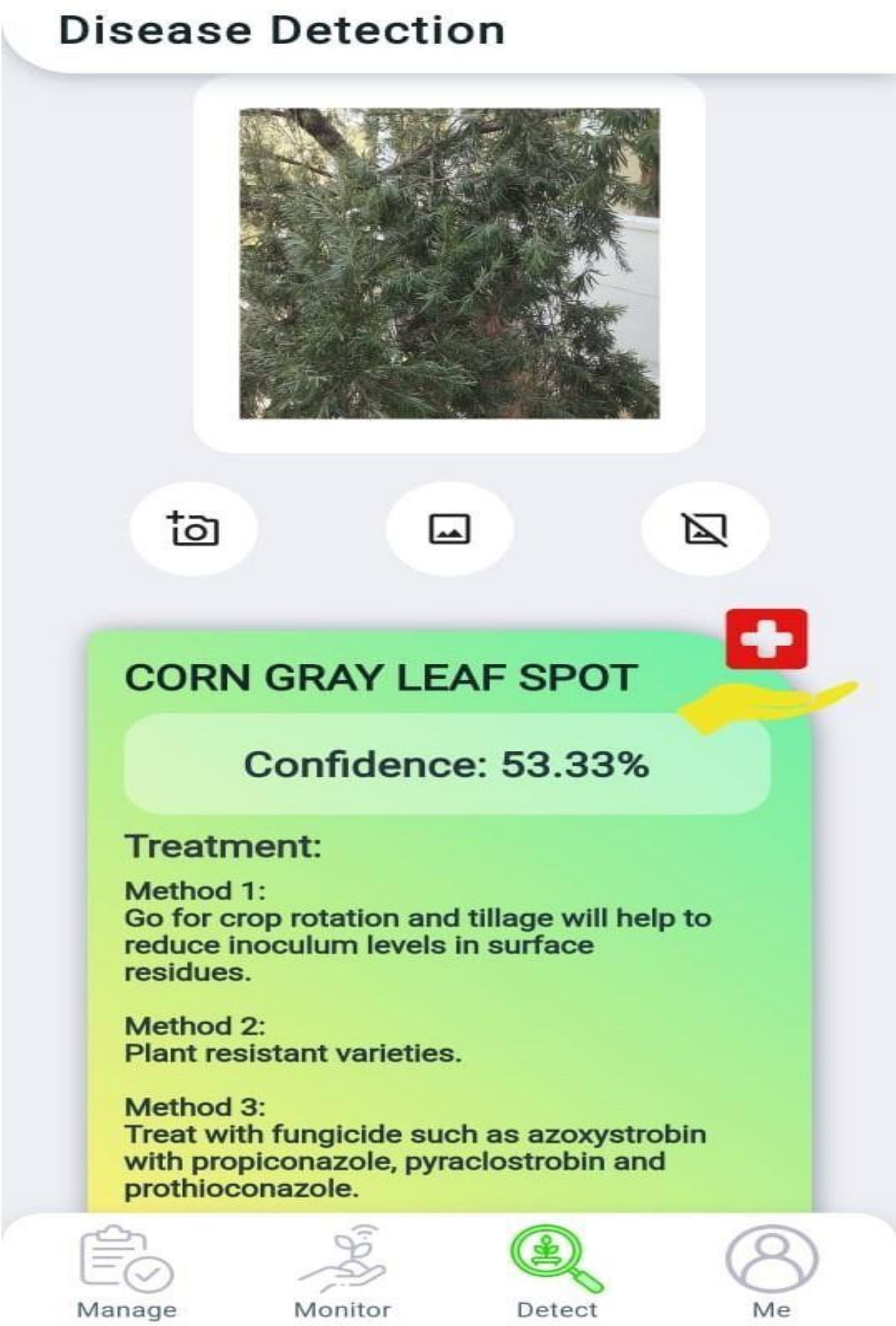

FIG 6.6 DISEASES DETECTION

6.1.6 My Profile

- In this page to see the login details and username.

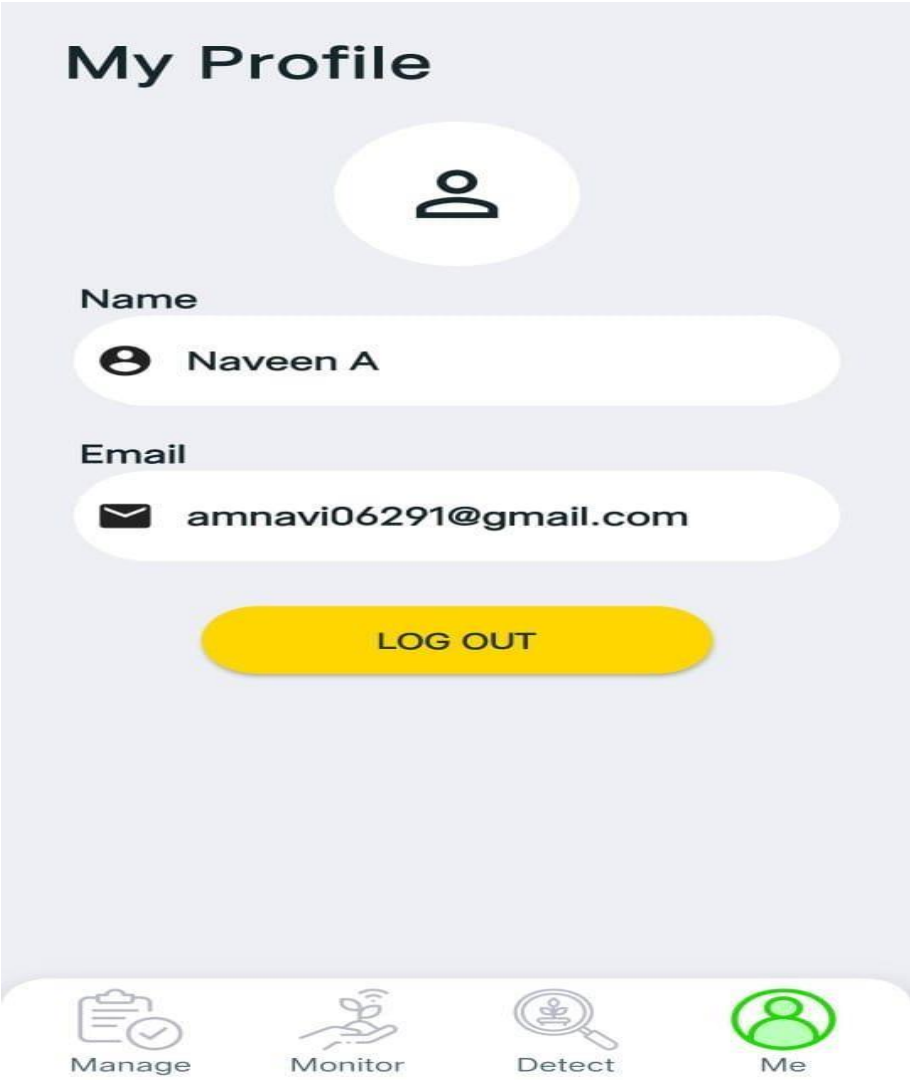

FIG 6.7 My Profile

CHAPTER 7
SYSTEM TESTING

Test each feature and functionality of the app, including crop recommendation, pest detection, weather forecasting, soil health assessment, market price prediction. Ensure that the app accurately processes user inputs, retrieves relevant data, and delivers correct results and recommendations. Assess the app's performance under normal and peak load conditions, ensuring that it responds quickly to user interactions and data requests. Measure the app's response time, processing speed, and resource utilization to identify any performance bottlenecks or scalability issues. Optimize the app's performance by optimizing algorithms, caching frequently accessed data, and leveraging parallel processing and distributed computing resources where applicable. Solicit feedback from farmers and stakeholders through surveys, interviews, and usability testing sessions to identify any usability issues or areas for improvement. Verify that the app remains operational and responsive even under adverse conditions, such as network failures, server outages, or hardware malfunctions. Implement error monitoring and logging mechanisms to track and diagnose any failures or anomalies in the app's behaviour, enabling quick detection and resolution of issues. Identify and address any security vulnerabilities or weaknesses in the app's architecture, codebase, data storage, and communication protocols. Validate the interoperability and compatibility of the app with external systems, APIs, and data sources, such as agricultural databases, weather APIs, soil sensors, and market databases. Test data exchange and communication between the app and external systems, ensuring seamless integration and accurate data synchronization. Engage farmers and end-users in user acceptance testing to validate that the app meets their needs, expectations, and requirements.

CHAPTER 8

APPENDIX

8.1 SOURCE CODE

8.1.1 MAIN.DART

```
import

'package:firebase_core/firebase_core.dart';

import 'package:flutter/material.dart'; import
```

```
'package:flutter/services.dart'; import
'app.dart'; void main() async {
  WidgetsFlutterBinding.ensureInitialized();
await Firebase.initializeApp();

SystemChrome.setSystemUIOverlayStyle(SystemUiOverlaySty
le(     statusBarColor: Colors.transparent,
statusBarIconBrightness: Brightness.dark,
statusBarBrightness: Brightness.dark,
systemNavigationBarColor: Colors.white,
systemNavigationBarDividerColor: Colors.grey,
systemNavigationBarIconBrightness: Brightness.dark,
  ));
  await SystemChrome.setPreferredOrientations(<DeviceOrientation>[
    DeviceOrientation.portraitUp,
    DeviceOrientation.portraitDown
  ]).then((_) => runApp(App()));
}
```

8.1.2 APP.DART

```
import 'package:farmassist/app_theme.dart';
import 'package:farmassist/bloc/authentication/authentication.dart';
import
  'package:farmassist/data/authentication/authentication_repository.dart';
import 'package:farmassist/data/farm/resources/repository.dart';
import 'package:farmassist/ui/farm/news/bloc/nBloc.dart';
import 'package:farmassist/ui/farm/news/bloc/nEvent.dart';
import 'package:farmassist/ui/farm/news_details/bloc/dBloc.dart';
import 'package:farmassist/ui/farm/statistics/utils/getEvents.dart';
import 'package:farmassist/ui/home_page.dart';
import 'package:farmassist/ui/login/login_page.dart';
import 'package:farmassist/ui/splash_page.dart';
```

```
import 'package:flutter/material.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:provider/provider.dart';
import 'data/farm/view_model/cityEntryViewModel.dart';
import 'data/farm/view_model/weather_app_forecast_viewmodel.dart';
class App extends StatelessWidget {
  App({Key key}) : super(key: key);
  final AuthenticationRepository _authenticationRepository =
  AuthenticationRepository();
  @override
  Widget build(BuildContext context) {
  return MultiProvider(
   providers: [
   BlocProvider<AuthenticationBloc>(
       create: (_) => AuthenticationBloc(
       authenticationRepository: _authenticationRepository)),
        BlocProvider<NewsBloc>(
         create: (_) => NewsBloc(repository: Repository())
         ..add(Fetch(type: 'Science'))),
        BlocProvider<DetailBloc>(
           create: (_) => DetailBloc(null),
        ),
        ChangeNotifierProvider<ForecastViewModel>(
           create: (_) => ForecastViewModel()),
        ChangeNotifierProvider<CityEntryViewModel>(
           create: (_) => CityEntryViewModel()),
      ],
      child: MultiRepositoryProvider(
        providers: [
          RepositoryProvider<AuthenticationRepository>.value(
            value: _authenticationRepository,
          ),
        ],
        child: AppView(),
      ),
    );
  }
}
class AppView extends StatefulWidget {
  @override
  AppViewState createState() => _AppViewState();
}
```

```
class _AppViewState extends State<AppView> {
 final _navigatorKey = GlobalKey<NavigatorState>();
 NavigatorState get _navigator => _navigatorKey.currentState;
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'Farmassist',
     debugShowCheckedModeBanner: false,
     theme: ThemeData(
       primaryColor: AppTheme.nearlyGreen,
       accentColor: AppTheme.nearlyGreen,
       appBarTheme: AppTheme.appBarTheme,
       scaffoldBackgroundColor: AppTheme.background,
       textTheme: AppTheme.textTheme,
       inputDecorationTheme: AppTheme.inputDecorationTheme,
     ),
     navigatorKey: _navigatorKey,
     builder: (context, child) {
       return BlocListener<AuthenticationBloc, AuthenticationState>(
         listener: (context, state) {
           switch (state.status) {
             case AuthenticationStatus.authenticated:
               _navigator.pushAndRemoveUntil<void>(
                 HomePage.route(),
                 (route) => false,
               );
               break;
             case AuthenticationStatus.unauthenticated:
              navigator.pushAndRemoveUntil<void>(
                   LoginPage.route(),
                (route) => false,
                 );
                 break;
               default:
                 break;
             }
           },
           child: child,
         );
       },
       onGenerateRoute: (_) => SplashPage.route(),
     );
```

```
  }
}
import 'dart:ui';
class DiseaseCardItem {
  DiseaseCardItem({
    this.diseaseName,
    this.action,
    this.treatment,
    this.color1,
    this.color2,
    this.imagePath,
  });
  final String diseaseName;
  final String action;
  final String treatment;
  final Color color1;
  final Color color2;
  final String imagePath;
  static List<DiseaseCardItem> diseaseItem = [
    DiseaseCardItem(
      diseaseName: 'DISEASE',
      action: 'Action/Treatment Information',
      treatment: 'Snap or choose an image now for disease detection!',
      color1: const Color(0XFFFFF176),
      color2: const Color(0XFF69F0AE),
      imagePath: 'assets/images/treatment.png',
    ),
    DiseaseCardItem(
      diseaseName: 'TOMATO LEAF MOLD',
      action: 'Treatment:',
      treatment:
      color1: const Color(0XFFFFF176),
      color2: const Color(0XFF69F0AE),
      imagePath: 'assets/images/treatment.png',
    ),
    DiseaseCardItem(
      diseaseName: 'STRAWBERRY LEAF SCORCH',
      action: 'Treatment:',
      treatment:
      color1: const Color(0XFFFFF176),
      color2: const Color(0XFF69F0AE),
      imagePath: 'assets/images/treatment.png',
```

```
  ),
  DiseaseCardItem(
    diseaseName: 'TOMATO MOSAIC VIRUS',
    action: 'Actions:',
    treatment:
    color1: const Color(0XFFFFF176),
    color2: const Color(0XFF69F0AE),
    imagePath: 'assets/images/treatment.png',
  ),
  DiseaseCardItem(
    diseaseName: 'CORN COMMON RUST',
    action: 'Treatment:',
    treatment:
    color1: const Color(0XFFFFF176),
    color2: const Color(0XFF69F0AE),
    imagePath: 'assets/images/treatment.png',
  ),
  DiseaseCardItem(
    diseaseName: 'CORN GRAY LEAF SPOT',
    action: 'Treatment:',
    treatment:
    color1: const Color(0XFFFFF176),
    color2: const Color(0XFF69F0AE),
    imagePath: 'assets/images/treatment.png',
  ),
  DiseaseCardItem(
    diseaseName: 'POTATO EARLY BLIGHT',
    action: 'Actions:',
    treatment:
    color1: const Color(0XFFFFF176),
    color2: const Color(0XFF69F0AE),
    imagePath: 'assets/images/treatment.png',
  ),
  DiseaseCardItem(
    diseaseName: 'HEALTHY',
    action: 'Condition:',
    treatment
    color1: const Color(0XFFFFF176),
    color2: const Color(0XFF69F0AE),
    imagePath: 'assets/images/healthy.png',
  ),
];
```

}

CHAPTER 9
CONCLUSION

9.1 Conclusion

Environmental and social impacts must also be considered, with a focus on promoting sustainable farming practices, minimizing environmental footprint, and fostering inclusivity and social acceptance within farming communities. In conclusion, while there are challenges and complexities to navigate, the potential benefits of an AI application for farmers are immense. By addressing feasibility considerations comprehensively and implementing solutions thoughtfully, stakeholders can unlock the full potential of AI to transform agriculture and contribute to a more sustainable and prosperous future for farmers worldwide.

9.2 Future enhancement

While AI farm apps are promising, there are challenges like initial cost and ensuring accessibility for all farmers. But overall, AI has the potential to be a game-changer for agriculture, boosting production and sustainability for the future. Imagine an app that uses artificial intelligence to be your personal farm advisor.

CHAPTER 10

REFERENCES

[1] Udegbe, Francisca & Ebulue, Ogochukwu & Ebulue, Charles & Ekesiobi, Chukwunonso. (2024). The role of artificial intelligence in healthcare: a systematic review of applications and challenges.

[2] M. Javaid, A. Haleem, I. H. Khan and R Suman, "Understanding the potential applications of Artificial Intelligence in Agriculture Sector", Advanced Agrochem, vol. 2, no. 1, pp. 15-30, 2023.

[3] Hegedus, Csaba & Frankó, Attila & Varga, Pal & Gindl, Stefan & Tauber, Markus. (2023). Enabling Scalable Smart Vertical Farming with IoT and Machine Learning Technologies. 1-4.

[4] M. Wakchaure, B.K Patle and A. K. Mahindrakar, "Application of AI techniques and robotics in agriculture: A review", Artificial Intelligence in the Life Sciences, pp. 100057, 2023.

[5] Singh Parvinder and Amandeep Kaur, "A systematic review of artificial intelligence in agriculture", Deep Learning for Sustainable Agriculture, pp. 57-80, 2022.

[6] A. Sood, R. K. Sharma and A. K. Bhardwaj, "Artificial intelligence research in agriculture: A review", Online Information Review, vol. 46, no. 6, pp. 1054-1075, 2022.

[7] Rayda Ben Ayed and Mohsen Hanana, "Artificial intelligence to improve the food and agriculture sector", Journal of Food Quality, pp. 1-7, 2021.

[8] D. Eisenha uer, “Irrigation Efficiency and Uniformity, and Crop Water Use Efficiency Suat Irmak, Extension Soil and Water Resources and

Irrigation Engineering Specialist, Professor EXTENSION ®.” Accessed: Dec. 23, 2021.

[9] Y. Li and X. Chao, "ANN-based continual classification in agriculture", Agriculture, vol. 10, no. 5, pp. 178, 2020.

[10] N. Chergui, M. T. Kechadi and M. Donnell, "The impact of data analytics in digital agriculture: a review", 2020 International Multi-Conference: “Organization of Knowledge and Advanced Technologies” (OCTA), pp. 1-13, 2020, February.

[11] Shankar Priyamvada et al., "Artificial intelligence driven crop protection optimization for sustainable agriculture", 2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G), 2020.

[12] Zha Jiali, "Artificial intelligence in agriculture", Journal of Physics: Conference Series, vol. 1693, no. 1, pp. 012058, 2020.

[13] V. Partel, S. Charan Kakarla, and Y. Ampatzidis, “Development and evalua tion of a low-cost and smart technology for precision weed management utilizing artificial intelligence,” Computers and Electronics in Agriculture,vol. 157, pp. 339–350, Feb. 2019, doi: 10.1016/j.compag.2018.12.048

[14] R. Mark, "Ethics of using AI and big data in agriculture: The case of a large agriculture multinational", The ORBIT Journal, vol. 2, no. 2, pp. 1-27, 2019.