**Integrated Crop Protection Management**

## A PROJECT REPORT

## *Submitted by,*

|  |  |  |
| --- | --- | --- |
| Sl. No. | Roll Number | Student Name |
| 1 | 20211CAI0154 | MANOJ J R |
| 2 | 20211CAI0068 | VEERESH B |
| 3 | 20211CAI0100 | K SAINATH |
| 4 | 20221LCA0008 | KUSHAL M P |

### *Under the guidance of,*

**Dr Mohammadi Akheela Khanum**

School of Computer Science, Presidency University, Bengaluru

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING,**

**ARTIFICAL INTELLIGENCE AND MACHINE LEARNING**



**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the Project report **“Integrated Crop Protection Management”** being submitted by **“MANOJ JR, VEERESH B, KUSHAL MP, K SAINATH”**, bearing roll numbers **“20211CAI0154”, “20211CAI0068”, “20221LCA0008”, “20211CAI0100”** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering(Artificial Intelligence and Machine Learning) is a bonafide work carried out under my supervision.

|  |  |
| --- | --- |
| **Dr. Mohammadi Akheela Khanum**  Professor  School of CSE&IS  Presidency University | **Dr. Zafar Ali Khan**  Associate Professor & HoD  School of CSE&IS  Presidency University |

|  |  |  |
| --- | --- | --- |
| **Dr. L. SHAKKEERA**  Associate Dean  School of CSE  Presidency University | **Dr. MYDHILI NAIR**  Associate Dean  School of CSE  Presidency University | **Dr. SAMEERUDDIN KHAN**  Pro-Vc School of Engineering  Dean -School of CSE&IS  Presidency University |

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled **Integrated Crop Protection Management** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering(Artificial Intelligence and Machine Learning)**, is a record of our own investigations carried under the guidance of **Dr. Mohammadi Akheela Khanum, PROFESSOR,** **School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of

any other Degree.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | |  |  |  | | --- | --- | --- | | NAME | ROLL NUMBER | SIGNATURE | | MANOJ J R | 20211CAI0154 |  | | VEERESH B | 20211CAI0068 |  | | K SAINATH | 20211CAI0100 |  | | Kushal M P | 20221LCA0008 |  | | |

**ABSTRACT**

Agriculture is still the backbone of the Indian economy, employing more than half of the population and significantly contributing to the nation's GDP. However, the sector is marred by critical challenges such as unpredictable monsoons, inefficient crop planning, soil degradation, and limited access to advanced farming tools and technologies. These persistent issues lead to reduced agricultural productivity, increased operational costs, and diminished profitability for farmers, further exacerbating their financial and social vulnerabilities. These issues are to be addressed by conceptualizing AgroDoc, a novel mobile-based solution that aims at transforming traditional farming practices through cutting-edge technologies like Artificial Intelligence and Machine Learning. The platform provides predictive insights and actionable recommendations to farmers, integrating a monsoon prediction module for accurate weather forecasts, a soil health analysis system to assess nutrient levels and recommend suitable fertilizers, and a smart crop recommendation feature that considers soil conditions, climatic factors, and market dynamics. AgroDoc also provides real-time weather updates, market sentiment analysis for optimal pricing decisions, and warehouse location assistance to reduce post-harvest losses and ensure better storage solutions.

By filling gaps such as reliance on non-scientific soil analysis, absence of reliable weather predictions, and inadequate crop selection strategies, AgroDoc empowers farmers to make informed decisions, mitigate risks posed by unpredictable environmental conditions, and achieve higher productivity and profitability. Being user-friendly and designed to be accessible for a wide number of farming communities, the platform ensures the widespread adoption and inclusion of small and marginal farmers. Other positive features of AgroDoc are the stimulation of sustainable agricultural practices by practicing equal fertilizer application, optimal timings for irrigation, and diversified crops. AgroDoc provides a linkage between traditional farming methods and modern technological developments by giving the farmers greater economic stability and prolonged stability in the agriculture sector. By promoting knowledge sharing, community-driven learning, and access to advanced tools, AgroDoc will not only improve livelihoods but also contribute significantly to environmental preservation and food security. With this holistic approach, AgroDoc is poised to revolutionize agriculture and ensure a sustainable, profitable, and technologically empowered future for farmers.

**ACKNOWLEDGEMENT**

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakkeera L and Dr. Mydhili Nair,** School of Computer Science Engineering & Information Science, Presidency University, and Dr. Zafar Ali Khan, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Dr Mohammadi Akheela Khanum, Designation, Professor** and Reviewer **Dr Alamelu J Mangai Designation, Professor**, School of Computer Science Engineering & Information Science, Presidency University for their inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman,** department Project Coordinators AFROZ PASHA and Git hub coordinator **Mr. Muthuraj.** We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**MANOJ J R**

**VEERESH B**

**KUSHAL MP**

**K SAINATH**

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Table Name** | **Table Caption** | **Page No.** |
| 1  2 | Table 1.1  Table 1.2 | Software modules  Literature Survey | x  7 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Figure Name** | **Caption** | **Page No.** |
| 1 | Figure 1 | Architecture Diagram | 16 |

2 Figure 2 Time Line 23

3 Figure 3 Final output 54

4 Figure 4 SGD Mappings 55

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| CHAPTER NUMBER | TITLE | PAGE NUMBER |
|  | ABSTRACT | iv |
|  | ACKNOWLEDGEMENT | v |
|  | SOFTWARE MODULES | x |
|  | TECHNOLOGICAL BACKBONE | xi |
|  | ADVANTAGES OF AGRODOC | xii |
| 1 | INTRODUCTION | 1 |
|  | 1.1 Overview |  |
|  | 1.2 Challenges |  |
|  | 1.3 Approaches |  |
|  | 1.3.1 Technological Solution |  |
|  | 1.3.2 Soil Health Management |  |
|  | 1.3.3 Monsoon analysis |  |
|  | 1.3.4 Crop Recommendation |  |
| 2 | LITERATURE REVIEW | 4 |
| 3 | RESEARCH GAPS OF EXISTING METHODS | 8 |
| 4 | PROPOSED METHODOLOGY | 11 |
|  | 4.1 Monsoon analysis |  |
|  | 4.2 AI-Based Soil Analysis |  |
|  | 4.3 Fertilizer Recommendations |  |
|  | 4.4 Smart Crop Recommendation |  |
|  | 4.5 User-Friendly Mobile Application |  |
|  | 4.6 Sustainability and Environmental Focus |  |
| 5 | OBJECTIVES | 15 |
|  | 5.1 Monsoon Trends for Better Crop Planning |  |
|  | 5.2 Crop Recommendations |  |
|  | 5.3 Fertilizer Recommendations |  |
| 6 | SYSTEM DESIGN AND IMPLEMENTATION | 16 |
|  | 6.1 System Design |  |
|  | 6.1.1 Architecture |  |
|  | 6.1.2 User Interface (UI) Layer |  |
|  | 6.1.3 Core Functional Modules |  |
|  | 6.1.4 Monsoon Prediction |  |
|  | 6.1.5 Soil Health Analysis |  |
|  | 6.1.6 Smart Crop Recommendations |  |
|  | 6.2 Implementation Plan |  |
|  | 6.2.1 Requirement Gathering & Planning |  |
|  | 6.2.2 System Design & Prototyping |  |
|  | 6.2.3 Development & Integration |  |
| 7 | TIMELINE FOR EXECUTION OF PROJECT | 23 |
| 8 | OUTCOMES | 25 |
|  | 8.1 Technological Outcomes |  |
|  | 8.2 Economic Outcomes |  |
|  | 8.3 Environmental Outcomes |  |
|  | 8.4 Social Outcomes |  |
| 9 | RESULTS AND DISCUSSIONS | 27 |
| 10 | CONCLUSION | 29 |
|  | REFERENCES | 31 |
|  | PSUEDO CODE | 33 |
|  | SCREENSHOTS | 54 |
|  | ENCLOSURE | 55 |

|  |  |
| --- | --- |
| **CATEGORY** | **DETAILS** |
| **Module OVERVIEW** |  |
| Number of modules | 4 key modules |
| Total functionality | Comprehensive agricultural support including weather, soil, crop, fertilizer and market analysis |
| **MODULE SPECIFICATIONS** |  |
| Weather Forecasting | Monsoon prediction and real-time weather updates for better planning |
| Soil Health Analysis | Nutrient level assessment and fertilizer recommendations |
| Crop Recommendation | Smart suggestions based on soil, climate, and market trends |
| Market Analysis | Insights on crop pricing, demand trends, and buyer information |
| **DATA FORMAT** |  |
| Input type | API-integrated datasets and user-entered data |
| Output type | Interactive dashboards, notifications, and detailed reports |
| **SUITABILITY** |  |
| Adaptability | Handles diverse farming needs through modular, scalable design |
| Sustainability Impact | Promotes eco-friendly farming practices for long-term agricultural viability |

# Software Modules

### TECHNOLOGICAL BACKBONE

### The AgroDoc platform is underpinned by a robust technological framework that integrates cutting-edge tools and methodologies to deliver seamless, data-driven solutions for farmers. Its technological backbone is designed to ensure scalability, reliability, and accessibility, addressing the diverse and complex challenges of modern agriculture.

**1. Artificial Intelligence and Machine Learning**

At the core of AgroDoc is the integration of AI and ML algorithms, which enable predictive insights and actionable recommendations. These technologies power:

* **Weather Prediction Models:** Machine learning models analyze historical weather data and real-time updates to provide accurate monsoon forecasts and weather alerts.
* **Soil Health Analysis:** AI-driven systems process soil test data to assess nutrient levels and recommend fertilizers and soil treatments tailored to specific conditions.
* **Smart Crop Recommendation:** ML algorithms analyze soil quality, climatic conditions, and market trends to suggest optimal crop choices for maximum yield and profitability.

#### **2. Cloud Computing Infrastructure:**

The platform leverages cloud-based services for data storage, processing, and analytics. Cloud computing ensures scalability and accessibility, enabling farmers to access AgroDoc from any location with internet connectivity. It also facilitates seamless integration with external APIs for weather updates and market trends

#### **Security and Privacy**:

Data security is a priority for AgroDoc. Advanced encryption techniques are employed to protect user data, ensuring confidentiality and compliance with data protection regulations. Role-based access control mechanisms further enhance security.

### ADVANTAGES OF AGRODOC

1. **High-Quality Transcriptions:** The dataset includes accurate transcriptions, which are essential for training reliable speech recognition and accent detection models.
2. **Diverse Accents:** The inclusion of multiple regional and national accents provides a robust foundation for developing systems capable of handling real-world linguistic variability.
3. **Optimized Clip Lengths:** The short duration of the clips ensures efficient processing while maintaining sufficient information for analysis.
4. **Flexible Format:** The .mp3 audio format supports seamless integration into existing machine learning pipelines without compromising on audio quality.

### Applications Enabled by the Dataset

**1.Enhanced Decision-Making:**

Provides data-driven insights for crop selection, soil management, and market trends, enabling informed decisions that optimize productivity and profits.

**2.Accurate Weather Forecasting:**

Predicts monsoon patterns and provides real-time weather updates, reducing risks associated with unpredictable weather conditions and ensuring better planning.

**3.Improved Soil Health Management:**

Conducts detailed soil health analysis, recommending appropriate fertilizers and soil treatments, leading to sustained soil fertility and improved crop yields.

**4.Smart Crop Recommendations:**

Suggests the best crops based on soil type, climate, and market trends, ensuring alignment with regional conditions and profitability goals.

**5.Market Connectivity:**

Provides farmers with real-time market insights, including price trends, demand forecasts, and buyer information, facilitating better trade opportunities.

**6.Continuous Learning and Updates:**

AI-driven models continuously learn and adapt, ensuring that farmers receive the most current and relevant recommendations.

**7.Cost-Effectiveness:**

Reduces input costs by optimizing fertilizer and pesticide usage and minimizes post-harvest losses through better storage and planning.

**CHAPTER-1**

**INTRODUCTION**

## 1.1 Overview

Agriculture is the back bone of the Indian economy, employing over 50% of the workforce and a significant share of the nation's GDP. Although it is crucial to the nation, the sector faces a host of challenges that prevent it from reaching its potential. Unpredictable monsoons, inadequate soil management, suboptimal crop selection, and limited access to reliable market data are but a few of the obstacles that farmers face every day. These problems lead to low productivity, high expenses, and financial insecurity for millions of farmers, mostly small and marginal farmers. The quest for an all-encompassing, technology-based solution has never been more urgent.

To tackle these pressing concerns, AgroDoc is conceived as a revolutionary platform that uses the latest technologies, such as Artificial Intelligence (AI), Machine Learning (ML), and the Internet of Things (IoT) to empower farmers. AgroDoc is poised to transform conventional farming through a suite of tools and services that can be used to make better decisions, utilize resources optimally, and gain more profitability. The critical functionalities include monsoon prediction, soil health analysis, smart crop recommendations, and market trend insights. By harnessing these features, AgroDoc not only improves farm productivity but also sustainability and economic stability.

AgroDoc provides personalized, data-driven recommendations for each farmer based on their specific needs. The soil health analysis module checks the nutrient levels and recommends targeted interventions. The crop recommendation system takes into account the soil conditions, climatic patterns, and market trends to suggest the most profitable crops. The real-time weather updates and monsoon predictions on the platform help farmers mitigate risks associated with unpredictable climatic conditions.

Market insights are another critical component of AgroDoc. The platform connects farmers to real-time pricing information, demand trends, and potential buyers, empowering them to make informed decisions and secure better deals for their produce. AgroDoc also addresses logistical challenges by suggesting nearby storage facilities and optimizing post-harvest management to reduce losses.

AgroDoc is designed with accessibility in mind. It has a multilingual mobile application for India's diverse farmer community. The app will be able to work offline in regions with limited internet connectivity, making it a reliable tool for farmers across rural and remote areas. Future iterations of AgroDoc will incorporate blockchain technology to enhance supply chain transparency and GIS-based tools for precision agriculture, further strengthening its impact.

Summarily, AgroDoc embodies a whole approach toward upgrading the Indian agriculture paradigm. This platform bridges traditional practices with advanced technology, overcoming the longstanding barriers and facilitating better productivity for farmers to improve their economic livelihoods. AgroDoc goes much more than just being a tool-it becomes a step toward building a sustainable, resilient, and prosperous agricultural ecosystem.

**1.2 CHALLENGES:**

Addressing these challenges will be crucial for AgroDoc's success in revolutionizing traditional farming practices**.**

**1.Comprehensive Soil Health Analysis**

Difficulty in acquiring real-time soil health data and accurately identifying nutrient deficiencies using AI-driven analysis tools.

**2.Integration of Real-Time Data from Multiple Sources**

Challenge: Seamlessly integrating data from various sources (e.g., weather APIs, soil sensors, market trends) and ensuring its accuracy and timeliness. Maintaining a robust and scalable system to process and analyze large datasets is a technical hurdle.

Impact: Delays or inaccuracies in data integration could undermine the reliability of recommendations and reduce farmers' trust in the platform.

**3.Accurate and Localized Weather Predictions**

Challenge: Developing highly accurate, region-specific monsoon and weather forecasts using AI and ML models. Ensuring precision in predicting events like unseasonal rains or droughts is critical for effective crop planning and irrigation management.

## APPROACHES

#### **1.Accurate and Localized Weather Predictions:**

To tackle the challenge of accurate and localized weather predictions, AgroDoc will utilize advanced machine learning models, such as Long Short-Term Memory (LSTM) networks and time-series analysis. These models will analyze historical weather data, including rainfall patterns, temperature fluctuations, and humidity levels, to identify trends and predict monsoon behavior. By training these models on region-specific datasets, the system can provide highly localized forecasts, reducing the risks associated with unpredictable weather events like droughts or excessive rainfall. Additionally, the platform will incorporate ensemble methods, which combine multiple predictive models to improve reliability and accuracy.

**2.Integration of High-Resolution Meteorological Data:**

AgroDoc will also integrate real-time data from high-resolution meteorological APIs and satellite feeds. This integration allows the system to continuously update weather forecasts, ensuring accuracy and timeliness. Alerts for extreme weather conditions, such as heavy rainfall, storms, or sudden temperature drops, will be delivered via push notifications to farmers. By providing actionable insights, AgroDoc empowers farmers to make informed decisions about sowing, irrigation, and harvesting, thereby mitigating potential crop losses due to adverse weather

**Centralized and Scalable Data Management System:**

AgroDoc will implement a centralized data architecture that consolidates information from diverse sources, such as soil health sensors, market trend APIs, and weather data feeds. Using technologies like cloud databases and distributed systems, the platform will ensure scalability and efficient data storage. This system will enable the seamless processing and analysis of large datasets, providing farmers with timely, accurate insights. Additionally, the use of standardized data formats and protocols will facilitate smooth integration across various modules, minimizing potential errors.

**Farmer-Centric Interface Design:**

A key approach to driving adoption is designing an intuitive and user-friendly interface that caters to farmers with varying levels of technological proficiency. AgroDoc will use large icons, simple navigation, and visually engaging dashboards to present information clearly. The app will also feature interactive tutorials and guided workflows, helping users quickly understand how to input data, interpret insights, and apply recommendations. To ensure inclusivity, the interface will support multiple regional languages, making it accessible to a diverse user base across India.

**CHAPTER-2**

**LITERATURE REVIEW**

In short, weather forecasting has been very essential for agriculture. Now the AI and the machine learning advances have been doing wonders to ensure the monsoons' forecasts accuracy. Long Short-Term Memory (LSTM) networks as well as the time-series model are commonly employed to predict rain patterns by making use of previous weather data analysis. Studies reflect that such models are more robust than traditional statistical methods, owing to the inclusion of non-linear dependencies and seasonal patterns in the climatic datasets. However, there is still a challenge in providing hyper-localized forecasts. The generalized data provided by existing weather prediction platforms, such as IMD (India Meteorological Department) and AccuWeather, does not necessarily meet the specific needs of agriculture.

Adding real-time data through meteorological APIs and satellite images to the weather predictions was also proposed. This will enhance the utility of such forecasts by giving a more precise picture of sudden changes in the weather, for instance, unseasonal rains or droughts, thus helping the farmers prepare well. However, the literature states that region-specific models are essential and should be designed according to microclimatic conditions. AgroDoc combines AI-driven models with local data to produce accurate and actionable weather forecasts.

So, soil health has a very direct role in the yield of the crop. Traditionally, soil testing is done using chemical analysis. This is both time-consuming and inaccessible to small-scale farmers. Recent advancements in AI have brought about the development of soil health assessment tools which analyze parameters like pH, moisture content, and nutrient levels. Machine learning models, especially SVM and decision trees, have been used to predict soil fertility and provide fertilizers. It has been found that in studies on AI-driven soil analysis, accuracy improves while it also saves the land from overuse of fertilizers.

However, the above technoAdvances limit the use of soil health platforms on large scales due to their high cost and technical complications. Most studies indicate that research on affordable mobile-based solutions that can provide real-time insights on soil health would be an area of high priority in the near future. AgroDoc uses AI models in analyzing its own soil data for fertilizer application recommendations to ensure cost-effectiveness and sustainability. Its integration with sensor data and predictive algorithms resolves the traditional soil testing approach limitations.

An appropriate crop selection plays a vital role in maximizing yields and profitability. The existing systems for crop recommendations mainly take into account the types of soil and weather conditions alongside the historical data regarding yield. Several studies have proved that certain machine learning algorithms, such as random forests and k-nearest neighbors, are appropriate for identifying ideal crops for any given region. These systems normally neglect dynamic factors such as the level of demand in the markets and price volatilities since they greatly determine farmers' choices.

The research in smart farming highlights the need for market trends to be integrated into crop recommendation systems. Platforms such as FarmERP and Kisan Suvidha provide partial solutions by providing market price updates along with agronomic advice. However, they lack real-time decision-making capabilities and holistic integration of data sources. AgroDoc addresses these gaps by incorporating market sentiment analysis and dynamic crop modeling. The system gives farmers crop recommendations based on real-time environmental and economic conditions, ensuring both productivity and profitability.

AI and machine learning have been transformative for agriculture, supporting predictive analytics and precision farming. Deep learning models, including convolutional neural networks (CNNs), are being used for such tasks as pest detection and yield estimation. The deployment of IoT devices further expands data collection and monitoring capabilities to paint a more comprehensive picture of the farm's condition. Despite these advancements, the adoption of AI technologies is still low in rural areas mainly because of bad internet connectivity and high costs in addition to limited technical expertise.

AgroDoc has designed a user-friendly, comprehensive mobile application with offline functionality and multilingual support to meet these challenges. The product is designed for rural farmers, maximally accessible and user-friendly, and, through collaboration with local cooperatives and agricultural extension programs, will promote the uptake of technology and provide farmers with the opportunity to make informed, data-driven decisions.

Agriculture, a critical sector in India, faces challenges such as unpredictable weather, declining soil fertility, and inefficient crop selection, which directly impact farmers’ productivity and profitability. AgroDoc, an AI-driven mobile application, seeks to address these issues by integrating monsoon trend analysis, soil health assessment, and smart crop recommendations. This literature review examines existing research in weather forecasting, soil analysis, and crop recommendation systems, highlighting their strengths and limitations. Additionally, it explores relevant studies in AI and machine learning applications in agriculture, providing a foundation for AgroDoc’s proposed solutions.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SL NO | **Title of the paper and year** | **Authors** | **Methodology used** | **Advantages** |
| **1** | Soil Quality Prediction Using Deep Learning,  **YAER: 2023** | **C. V. Suresh BabuYadavamuthiah K** | The paper employs deep learning techniques, combining Long Short-Term Memory networks and Convolutional Neural Networks to predict soil quality. The LSTM captures temporal relationships among variables, while the CNN extractsfeatures from soil samples. The dataset includes various physical, chemical, and biological soil parameters. The model uses cross-validation. | High accuracy in predicting soil  2) quality Ability to capture non-linear relationships in complex soil data Can process a variety of data sources |
| **2** | Soil Quality Prediction Using Deep Learning  **YEAR: 2023** | **C. V. Suresh Babu**  **Yadavamuthiah K,** | The paper utilizes deep learning models, specifically a combination of Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNNs). LSTMs capture temporal dependencies in data, while CNNs extract features from soil samples | High accuracy in predicting soil quality compared to conventional machine learning methods Ability to capture complex, non-linear relationships in the data. |
| **3** | **Weather Forecasting Framework for Time Series Data using Intelligent Learning Models"**, published in **December 2021**. | **Raksha S**,  **Graceline Jasmine S**  **L. Jani Anbarasi**, |  | High accuracy: Random Forest achieved 89%, which is higher than other models tested, including KNN and SVM.Scalability: The use of Hadoop Distributed File System (HDFS) ensures efficient handling of large datasets. |

**CHAPTER- 3**

**RESEARCH GAPS OF EXISTING METHODS**

**Localized and Accurate Weather Predictions**

Although AI and machine learning have enhanced weather forecasting, the current systems are not precise enough for hyper-localized predictions. Generalized forecasts from platforms such as the India Meteorological Department (IMD) and AccuWeather do not account for microclimatic variations that are critical for specific agricultural regions. This limitation results in suboptimal crop planning and increased vulnerability to unpredictable weather events such as sudden rainfall or droughts. There is a need for region-specific models that integrate local environmental data with historical weather trends to provide actionable insights for farmers.

Furthermore, most weather prediction systems lack real-time updates and dynamic changes in weather conditions. Delayed updating of forecasts might leave farmers without adequate preparation for extreme weather events. Models developed to provide continuous, real-time updates and incorporate satellite imagery, sensor data, and IoT devices may bridge this gap, ensuring that timely and accurate weather information tailored to specific farming zones is achieved.

**Comprehensive and affordable soil health analysis**

Traditional practices of soil testing are labor-extensive, time-consuming, and expensive. It remains unattainable for small and marginal farmers. AI-driven soil health analysis tools are not yet readily adopted due to the presence of advance sensors and laboratory facilities that may be unapproachable at rural areas. Additionally, such tools lack comprehensive views on soil health. Most of the tools focus on microbial action rather than water-holding capacity or structure and can significantly question the idea of long-run sustainability in agriculture.

The second gap is that there are no real-time monitoring solutions for soil. The current methods provide static data at specific intervals, which is not enough for dynamic decision-making. There is a need to develop mobile-based, low-cost soil health assessment tools that provide instant and continuous updates on the soil conditions and AI-powered fertilizer recommendations to encourage sustainable farming practices and improve yields.

**Integrated and Dynamic Crop Recommendation Systems**

Most of the crop recommendation systems that are available today depend on static information such as soil type, climate, and historical yield records. However, they usually fail to take into account dynamic factors such as fluctuating market demand, price trends, and changing weather patterns. Thus, farmers might choose crops that are not economically viable or appropriate for the prevailing environmental conditions, resulting in losses.

Moreover, the present systems are incapable of providing a personalized recommendation adapted to the unique conditions of every farm. A real-time integration of market and environmental data drastically limits the performance of these tools. For optimizing crop choice and maximizing profits, a more holistic crop recommendation system that considers real-time input from soil health, weather prediction, and market trends is a necessity.

**Fragmented agricultural solutions**

In fact, the majority of existing platforms address either weather forecasting, soil analysis, crop recommendation, etc., in isolation, and therefore, the approach forces farmers to use multiple tools, with less efficiency and increased complexity in decision-making processes. Moreover, a lack of interoperability typically exists among these systems, which prevents fluent data exchange between different platforms.

All-in-one, fully integrated solution incorporating weather data, soil health analysis, crop recommendation, market trend, and post-harvest management for holistic agricultural decision-making is of the utmost importance. Such integration has the potential to simplify, lighten the burden on farmers, and increase overall productivity and profitability in agriculture.

**Limited Scalability and Adaptability of AI Models in Agriculture**

Most models are trained on limited datasets, which restrict their ability to generalise across diverse agricultural scenarios. These models often don't adapt well without extensive retraining to new soil types, crop varieties, and climatic conditions. In addition, existing systems lack scalability, and are thus difficult to implement across large geographic areas with different agricultural needs.

Adaptive and scalable AI models capable of dealing with diverse datasets and incorporating feedback from real-world farming conditions need to be developed. Such models must be able to learn from minimal data and quickly adapt to new environments, ensuring broader applicability and effectiveness.

**Data Privacy and Security Concerns**

With agricultural platforms increasingly relying on data gathering, questions and concerns surrounding privacy and security are raised. Farmers are not going to share sensitive data relating to their land, crop patterns, or financial transactions without clear-cut assurances of data protection. Often, existing platforms lack a robust data privacy policy and communication of the use of the data that is gathered, leading to mistrust and low adoption rates.

For filling this gap, it is very important to have transparent data governance frameworks and use the advanced security measures like encryption and anonymization. Building trust through clear communication and ensuring compliance with data protection regulations will encourage farmers to adopt technology-driven solutions.

**CHAPTER - 4**

## PROPOSED METHODOLOGY

## The methodology of AgroDoc involves providing a farmer with an overall, AI-based platform for optimizing agricultural practices that can be able to tackle dramatic challenges including unpredictable weather, soil health management, and inefficient crop choice. It integrates six significant components like monsoon analysis, AI-based soil analysis, fertilizer recommendations, smart crop recommendations, user-friendly mobile application, and sustainability-focused farming strategies.

## Analysis using the machine learning models Long Short-Term Memory, etc., has enabled a better look into historical data in terms of weather and localized real-time delivery for farmers, helping them plan agriculture sowing and irrigation activities better. AI-based soil analysis through input data regarding pH levels and moisture content has also been applied for detailed health analysis of soils. The machine learning algorithms analyze nutrient deficiencies and suggest corrective measures, including optimal fertilizer recommendations tailored to the specific needs of the soil.

## Fertilizer recommendations are set based on a cost-effective and environment-friendly approach for proper utilization of chemical and organic fertilizers. The smart crop recommendations use dynamic models of machine learning in analyzing the conditions of soil, weather, and market trends, ensuring that suggested crops are not only agronomically appropriate but also economically feasible, which results in maximizing the yield and profit.

## A user-friendly mobile application acts as the front end for the farmer, providing support in multiple languages and offline to reach the more remote areas. The app then gives actionable insights to farmers through a very intuitive dashboard that helps farmers make informed decisions. Lastly, AgroDoc stresses sustainability, with balanced use of fertilizers, water saving, and crop rotation. This holistic approach ensures that the platform not only increases productivity and profitability but also preserves environmental resources for future generations.

### 4.1Monsoon Analysis

### Data-Driven Weather Forecasting:

### Monsoon trends are of prime importance in agricultural planning. It affects sowing schedules, irrigation management, and harvesting. AgroDoc uses machine learning models like Long Short-Term Memory (LSTM) networks and time-series algorithms to analyze historical weather data, including rainfall patterns, temperature fluctuations, and humidity levels. The models are trained on large meteorological datasets that enable the system to predict regional monsoon patterns accurately. This will mean ensuring timely, precision weather information regarding a farming area from credible real-time data such as satellite imaging and APIs, such as in the weather condition.

### Actionable Insights and Real-Time Alerts

### This monsoon analysis module is able to give actionable insights in preparation for weather risks in case of delayed rains, floods, or droughts. Alerts are sent immediately, informing the users of a change in the weather, enabling them to take appropriate precautions. It thus reduces the uncertainty of weather dependence in monsoon farming and further reduces losses on crops. Through localized and accurate weather forecasting, AgroDoc enables farmers to optimize their agriculture practices with greater productivity and less risk.

### 4.2 AI-Based Soil Analysis

### AgroDoc uses AI-based soil analysis to give farmers an in-depth analysis of their soil health. It processes information inputted by farmers, like pH levels, moisture content, and nutrient availability, through a machine learning model such as support vector machines (SVMs) and decision trees. The model is able to recognize nutrient deficiencies, soil texture imbalances, and potential issues with fertility, thereby enabling the provision of recommendations on what amendments should be applied to the soil.

### 4.3 Fertilizer Recommendations

### Based on the results of soil analysis, AgroDoc suggests the best type and amount of fertilizers suited to the particular needs of the land. AI models process nutrient-level data and cross-reference it with crop-specific requirements to ensure balanced fertilization. This minimizes the overuse of chemical fertilizers, which can harm the environment and reduce soil fertility over time.

### The system also provides alternative organic fertilizers and encourages eco-friendly farming practices. Recommendations are provided with detailed application schedules to ensure that fertilizers are used efficiently to maximize their benefits. This feature contributes to both economic and environmental sustainability by reducing input costs and enhancing soil health.

### 4.4 Smart Crop Recommendation

### AgroDoc uses algorithms, namely random forests and k-nearest neighbors, to make crop recommendations for a specific area. These algorithms analyze permutations of soil properties, climatic conditions, and historical yields to determine the best crops for maximum profitability and adaptability.

### Agronomic factors aside, the system integrates real-time market trends and price data to recommend crops that are in high demand. This way, farmers can gain better financial returns by aligning crop production with market needs. Crop recommendations are made both economically viable and environmentally sustainable.

### 4.5 User-Friendly Mobile Application

### The AgroDoc mobile application is designed with simplicity and ease of use. It has an interactive dashboard where key insights are presented in visually engaging formats: monsoon forecasts, soil health reports, crop recommendations, etc. Large icons and intuitive navigation cater to farmers with varying levels of technological proficiency.

### To make the app more accessible, it supports multiple regional languages and has offline capabilities for core functionalities. Farmers can input data and access recommendations even in areas with limited internet connectivity. This inclusivity ensures that AgroDoc can reach a diverse user base across rural and remote regions.

### 4.6 Sustainability and Environmental Focus

AgroDoc focuses on sustainable agriculture by promoting balanced fertilizer application, crop rotation, and organic farming. The recommendations of the system are made to reduce environmental impacts such as soil degradation and water pollution due to overuse of chemicals.

The platform guides users on efficient irrigation techniques and the best water-saving strategies, enabling farmers to conserve water. Additionally, it enhances long-term soil health through activities such as cover cropping and minimal tillage. AgroDoc contributes to the preservation of natural resources while ensuring the sustainability of agricultural ecosystems by integrating sustainability into every facet of farming.

This methodology ensures that AgroDoc delivers a holistic and scalable solution, addressing the key challenges in agriculture while promoting sustainable and profitable farming practices. Each component has been designed to work seamlessly within the system, providing actionable insights and fostering technological adoption among farmers.

**CHAPTER-5**

**OBJECTIVES**

1. **Empower Farmers with Predictive Insights:** Provide accurate monsoon forecasts and soil health analyses to enable informed decision-making and optimize farming practices.
2. **Optimize Crop Planning:** Recommend suitable crops based on climatic conditions, soil properties, and market trends to maximize yield and profitability.
3. **Enhance Economic Stability:** Utilize market sentiment analysis to help farmers navigate price volatility and secure better returns for their produce.
4. **Facilitate Post-Harvest Management:** Identify nearby storage facilities to minimize post-harvest losses and ensure crop quality.
5. **Promote Sustainable Practices:** Encourage balanced fertilizer use and sustainable farming methods to maintain soil health and long-term productivity.
6. **Leverage Advanced Technology:** Harness the power of AI and ML to deliver precise, user-friendly solutions accessible to farmers, even in remote areas.
7. **Improve Accessibility and Inclusivity:** Design a mobile application optimized for low-bandwidth environments with multilingual support to reach a diverse user base.
8. **Foster Knowledge Sharing:** Create a platform for farmers to share experiences, access government schemes, and benefit from community-driven learning.

**CHAPTER-6**

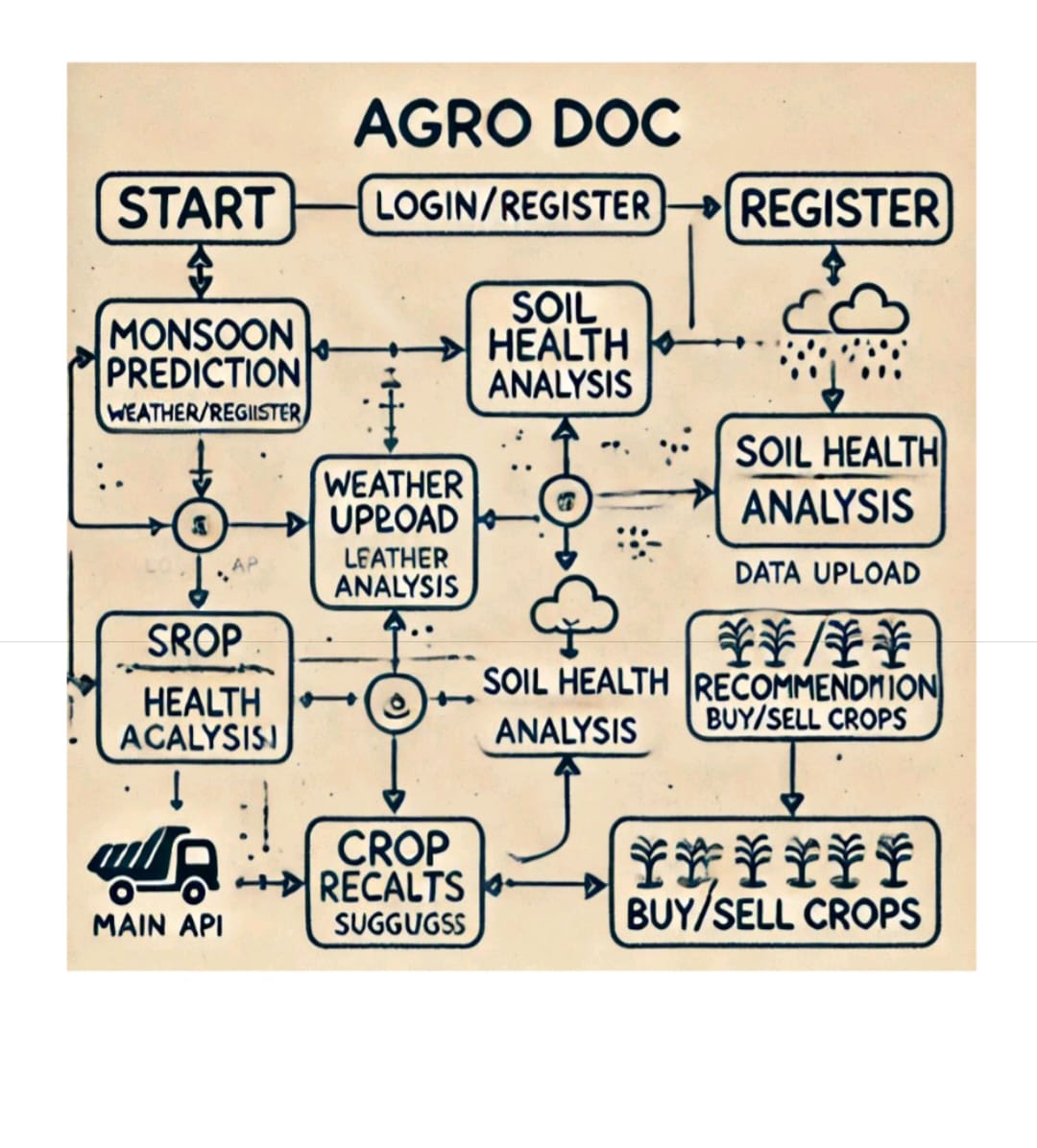
**SYSTEM DESIGN & IMPLEMENTATION**

**System Design:**

The system design and implementation of AgroDoc focus on developing an integrated, scalable, and user-centric platform that leverages AI and ML to address key challenges in agriculture. The approach involves designing a robust architecture, creating a farmer-friendly interface, and ensuring seamless integration of data-driven modules to deliver actionable insights.

#### **Architecture:**

.



**6.1 System Architecture**

The AgroDoc system is structured into three main layers:

**6.1.1 User Interface (UI) Layer**

Mobile Application:

A farmer-friendly mobile app serves as the primary interaction point for users.

**Features:**

• Input forms for soil data, crop preferences, and location details.

Dashboards displaying key metrics such as monsoon forecasts, soil health status, crop recommendations, and market insights.

Alerts and notifications for real-time updates on weather, pests, and market fluctuations

**6.1.2** **Application Layer**

The application layer of AgroDoc constitutes the core as it houses AI and ML models that power most of its functionality. These are monsoon prediction, soil health analysis, fertilizer recommendation, smart crop selection, and market price analysis. The layer uses the advanced algorithms and processes input from users and other APIs to deliver actionable insights pertinent to the farmers' needs. The system ensures that all inputs are integrated, which include soil parameters, weather forecasts, and market trends, with the help of sophisticated data processing pipelines.

This layer relies extensively on decision-making algorithms that determine outputs to be personalized and region-specific. Based on climatic conditions, composition of soil, and economic parameters, these algorithms will give correct and context-sensitive recommendations. Thus, this further enhances the utility of the system by allowing the farmers to enhance their agricultural activities and make well-informed decisions. The design of the application layer is also modular, with new functionalities getting integrated efficiently without compromising the scalabilities and adaptation towards changing agriculture needs.

**6.1.3 Data Layer**

The data layer is a critical component of AgroDoc, designed to store and manage vast amounts of agricultural data efficiently. A centralized Database Management System (DBMS) serves as the foundation, housing user data, historical weather trends, soil health records, and market information. This centralized structure ensures seamless access and retrieval of data, enabling quick processing and analysis for various system functionalities. By maintaining a well-organized and secure database, the data layer supports the system's ability to deliver accurate and personalized recommendations.

To enhance data accuracy and relevance, the system integrates reliable external APIs for weather, soil, and market trends. These APIs provide real-time updates, ensuring that the insights delivered to farmers reflect current conditions. This integration allows the platform to combine locally sourced data with global datasets, improving its predictive capabilities and enabling informed decision-making. The data layer’s robust architecture ensures scalability and adaptability, making it capable of handling the increasing data demands as AgroDoc expands its user base

**6.2. User Interface Design**

A key focus of the system design is to ensure that the mobile application is intuitive and accessible to farmers with varying levels of technological proficiency

**6.2.1 Features of the UI**

The UI of AgroDoc is intuitive and user-friendly, so it is easy to use for farmers with different levels of technological proficiency. A key feature is the simplified data input system, which allows farmers to easily enter soil and crop information through straightforward forms. The interactive dashboard presents vital information such as monsoon trends, soil health status, crop recommendations, and alerts in a visually engaging and easy-to-understand format.

To cater to a diverse farming population, the app provides multi-language support to overcome language barriers by offering content in multiple regional languages. Besides, offline functionality ensures that critical features remain available even in places with limited or no internet connectivity, making it highly reliable for remote regions. Enhance further the usability of real-time notifications by alerting farmers to the most critical events, including sudden weather changes, sudden appearances of pests, or upsetting market prices.

**6.2.2 Accessibility Features**

Optimized for low-bandwidth environments to ensure usability in remote rural areas.

Designed with large icons and simple navigation for ease of use.

farmers reflect current conditions. This integration allows the platform to combine locally sourced data with global datasets, improving its predictive capabilities and enabling informed decision-making. The data layer’s robust architecture ensures scalability and adaptability, making it capable of handling the increasing data demands as AgroDoc expands its user base

**6.3. Core Functional Modules**

**6.3.1 Monsoon Prediction**

AI/ML Models:

Predicts regional weather patterns based on historical and real-time data.

Insights:

Provides actionable information for crop planning, irrigation, and harvest scheduling.

**6.3.2 Soil Health Analysis**

Input Parameters:

Includes soil type, pH, moisture, and nutrient levels.

Al-Based Analysis:

Evaluates soil health and identifies nutrient deficiencies.

Fertilizer Recommendations:

Suggests optimal fertilizers and their application schedules.

**6.3.3 Smart Crop Recommendations**

Integrates soil health, climatic conditions, and market demand to recommend profitable crops.

**6.3.4 Market Price Analysis**

.

Analyzes market trends to help farmers secure better prices for their produce.

**6.3.5 Post-Harvest Management**

Recommends nearby storage facilities and offers guidance on crop handling and preservation.

**6.4. Implementation Plan**

The implementation plan is divided into multiple phases to ensure systematic development and deployment:

**6.4.1 Phase 1: Requirement Gathering & Planning**

Activities:

• Identify farmer needs through surveys and stakeholder consultations.

Define the scope of the project and key deliverables.

Output:

A comprehensive project plan detailing timelines, resources, and expected outcomes.

**6.4.2 Phase 2: System Design & Prototyping**

Activities:

Design the system architecture and database schema.

Create UI/UX prototypes for the mobile application.

• Develop proof-of-concept models for core Al functionalities.

Output:

A functional prototype demonstrating key features.

**6.4.3 Phase 3: Development & Integration**

Activities:

• Develop Al models for weather predictions, soil analysis, and crop recommendations.

Integrate external APIs for real-time weather and market data.

Build the mobile application and backend systems.

Output:

A fully developed platform ready for testing.

4.4 Phase 4: Testing & Optimization

Activities:

Conduct user testing to identify usability issues.

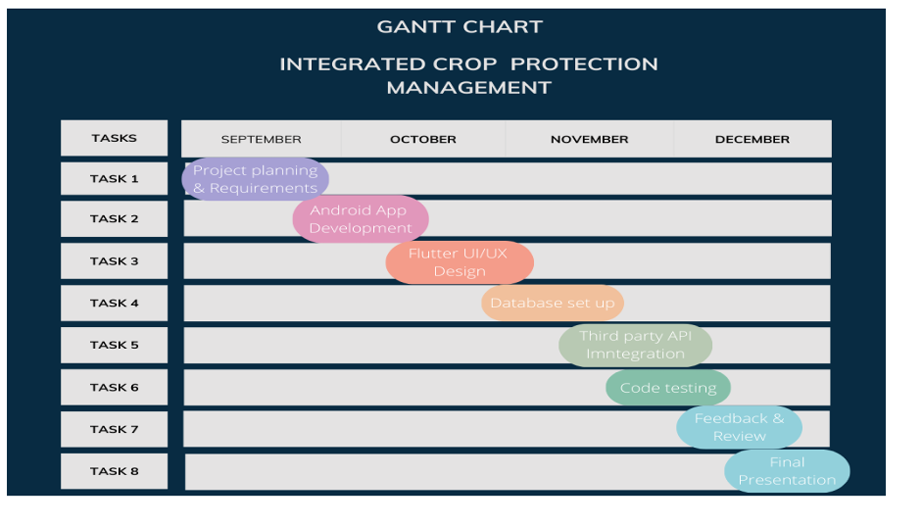
• Optimize Al models for accuracy and reliability.

Test the application for scalability, performance, and security.

**CHAPTER-7**

**TIMELINE FOR EXECUTION OF PROJECT**

**(GANTT CHART)**

****

**Project Timeline Overview (Gantt Chart)**

The execution timeline for the Integrated Crop Protection Management project is structured over four months, from September to December, with a detailed Gantt chart outlining eight distinct tasks. Each task is sequenced to ensure smooth progression and alignment with the project's objectives. Below is an explanation of the tasks and their timeline:

**1. Project Planning & Requirements (Task 1)**

Duration: September

The first step involves comprehensive planning and gathering requirements for the project. This includes defining the scope, identifying key features, and consulting stakeholders such as farmers, agricultural experts, and technical developers. This phase sets the foundation for the subsequent tasks.

**2. Android App Development (Task 2)**

Duration: October

The primary development phase begins with building the Android application. The focus is on creating a scalable and robust backend and integrating essential features such as data input forms, dashboards, and real-time alerts.

**3. Flutter UI/UX Design (Task 3)**

Duration: October-November

Simultaneously, the user interface and user experience (UI/UX) design phase is initiated.

Flutter is used to develop a cross-platform design that is visually appealing, intuitive, and accessible to users. Key design considerations include multi-language support and ease of navigation.

**4. Database Setup (Task 4)**

Duration: November

The next phase involves configuring a database to store essential data such as soil health metrics, weather trends, and market prices. This includes integrating reliable data sources and ensuring that the database is optimized for scalability and performance.

**5. Third-Party API Integration (Task 5)**

Duration: November

External APIs for weather predictions, soil analysis, and market trends are integrated into the system. This step ensures real-time data retrieval and enhances the app's functionality, enabling accurate predictions and recommendations.

**6. Code Testing (Task 6)**

Duration: November December

Once development and integration are complete, rigorous testing is conducted to identify and resolve bugs, optimize performance, and ensure the app's reliability. This phase also involves validating the outputs from AI/ML models.

**7. Feedback & Review (Task 7)**

Duration: December

Stakeholders, including farmers and technical experts, are invited to review the application and provide feedback. The review focuses on the app's usability, accuracy of recommendations, and overall performance.

**8. Final Presentation (Task 8)**

Duration: December

The project concludes with a final presentation, where the system is demonstrated to stakeholders. The presentation highlights the app's features, functionalities, and benefits, along with its impact on modern agricultural practices.

**CHAPTER-8**

**OUTCOMES**

The development and deployment of the Integrated Crop Protection Management system 18

yield transformative outcomes for the agricultural sector. By leveraging advanced technologies such as AI, ML, and a robust mobile application, the project addresses critical challenges faced by farmers and delivers measurable benefits. The outcomes of this project can be categorized into technological, economic, environmental, and social impacts, ensuring holistic growth and sustainability.

**8.1. Technological Outcomes**

**8.1.1 AI-Powered Monsoon Predictions**

Farmers receive highly accurate and localized monsoon forecasts, enabling better planning for sowing, irrigation, and harvesting.

Reduces dependency on unpredictable weather patterns, mitigating risks associated with delayed rains or droughts.

**8.1.2 Data-Driven Soil Health Analysis**

The system provides detailed soil health assessments, identifying deficiencies in nutrients and suggesting corrective actions.

Recommendations for optimal fertilizer usage minimize overuse, improving both yield and soil sustainability.

**8.1.3 Smart Crop Recommendations**

Al models analyze regional soil and climate conditions to suggest crops that promise maximum profitability and adaptability.

Ensures crop diversification, reducing the risks associated with monoculture farming.

**8.1.4 Market Insight Tools**

Farmers gain access to real-time market price trends, empowering them to make informed decisions regarding the sale of their produce.

Reduces exploitation by middlemen and enhances market competitiveness for farmers.

**8.2. Economic Outcomes**

**8.2.1 Increased Productivity**

By implementing data-driven recommendations for soil health, crop selection, and irrigation, farmers achieve higher yields.

Efficient resource utilization, such as water and fertilizers, reduces input costs and improves overall profitability.

**8.2.2 Better Market Access**

The integration of market trends into the app enables farmers to identify the best times and locations for selling crops, maximizing revenue.

Access to nearby storage solutions reduces post-harvest losses, further boosting income.

**8.2.3 Cost Optimization**

Al-powered analysis ensures judicious use of fertilizers and pesticides, leading to cost savings.

Reduced dependence on costly traditional weather forecasting services or external consultants.

**8.3. Environmental Outcomes**

**8.3.1 Sustainable Farming Practices**

Optimal fertilizer recommendations prevent overuse, protecting soil fertility and reducing water pollution caused by runoff.

The system promotes crop rotation and diversification, which maintain ecological balance.

**8.3.2 Efficient Resource Utilization**

The app aids in precise irrigation planning based on weather predictions, conserving water resources.

Encourages minimal use of chemical inputs, reducing the carbon footprint of farming activities.

**8.3.3 Soil Preservation**

Regular monitoring of soil health ensures timely interventions, preventing long-term degradation and ensuring sustainable farming for future generations.

**8.4. Social Outcomes**

**8.4.1 Empowerment of Farmers**

By providing farmers with actionable insights and advanced tools, the app bridges the gap between modern technology and traditional farming.

The system's multilingual support ensures inclusivity, catering to diverse linguistic and literacy levels among farmers.

**8.4.2 Improved Livelihoods**

Increased agricultural productivity and profitability directly translate into better living standards for farming communities.

Reduced dependence on external loans due to increased income stability.

**8.4.3 Community Knowledge Sharing**

Farmers become more knowledgeable about sustainable practices and can share their experiences and insights within their communities, fostering collective growth.

**8.5. Educational and Policy Impacts**

**8.5.1 Farmer Education**

The platform indirectly educates farmers on best agricultural practices, including soil conservation, pest management, and sustainable cropping.

Promotes awareness about climate change and its impact on farming, preparing communities for future challenges.

**8.5.2 Policy Formulation**

Data collected by the system can serve as a valuable resource for policymakers to design farmer-centric schemes and subsidies.

Insights from the platform may inform national strategies to enhance agricultural output and mitigate climate risks.

Sustainable Farming Practices

Optimal fertilizer recommendations prevent overuse, protecting soil fertility and reducing water pollution caused by runoff.

The system promotes crop rotation and diversification, which maintain ecological balance

**CHAPTER-9**

**RESULTS AND DISCUSSIONS**

### AgroDoc offers a streamlined, intelligent platform to farmers for acquiring critical agricultural insights. Its core functionalities are built with a user-centric and systematic approach to solving critical farming challenges. Upon opening the application, users are walked through a smooth workflow to get to the most relevant features, making data-driven decisions for their farming activities.

### The interaction starts when the user opens the AgroDoc application, by which the key features thereof gain access. Three main functionalities have been offered in the main menu: Monsoon Analysis, Soil Health Prediction, and Crop Recommendation. These features, with one visit to the app, provide the farmers with tailored insights into the particular agricultural needs of the users. Users can easily navigate to a desired feature based on their immediate priorities.

### The Monsoon Analysis module makes use of past and real-time weather data for accurate predictions about the upcoming monsoon conditions. The system aggregates data from authentic meteorological sources and applies sophisticated machine learning algorithms to analyze trends and patterns in weather. Based on AI-driven models, the app generates highly accurate monsoon forecasts that are essential for irrigation schedule planning and sowing time. Once the analysis is conducted, the forecast is displayed to the user in a readable manner so that a farmer can make some informed decisions, thereby mitigating risks from changing weather patterns.

### The Soil Health Prediction module makes an overall analysis of the condition of the soil. Users are allowed to enter manually or collect soil test data with the help of connected devices directly from the device, including parameters such as pH levels, moisture content, and nutrient availability. The application processes this data using AI models to identify deficiencies and imbalances in the soil. Based on the analysis, the system recommends specific fertilizers or soil amendments to enhance soil fertility and optimize crop growth. These actionable insights are presented in a clear and concise manner, enabling farmers to implement corrective measures efficiently and sustainably.

### The Crop Recommendation module integrates multiple data points to suggest the most suitable crops for cultivation. It begins by gathering climate data specific to the farmer's region, which includes temperature, humidity, and rainfall patterns. Additionally, real-time market trends and demand for various crops are incorporated into the analysis. The AI algorithms then analyze these inputs to present the crops having the highest profitable potential and fit to the local environmental and economic conditions. Therefore, the result crops are made easily understandable in a way which helps farmers integrate their production approach according to agro and the market perspective of crop selection.

### Once the user has explored the chosen feature, the system consolidates all outputs to ensure the delivery of comprehensive insights tailored to the user's needs. The results from different modules are interconnected, which enables farmers to make holistic decisions for their agricultural practices. For example, a farmer who has used the monsoon analysis feature can directly apply its insights to crop recommendations and irrigation management. This integrated approach minimizes uncertainty, optimizes resource use, and maximizes productivity.

### AgroDoc simplifies complex agriculture processes and brings a data-driven culture of farming together by integrating all these features into a single platform. With real-time delivery of precise, localized, and actionable insights, the application equips farmers to overcome problems, create yield improvements, and enhance profitability. Its intuitive design and user-friendly interface also make it accessible to farmers of all levels of technological proficiency, thereby ensuring a wide reach and impact. AgroDoc is the epitome of the transformative power of AI and technology in modern agriculture, given its streamlined workflow and advanced functionalities.

**CHAPTER-10**

**CONCLUSION**

Agriculture has always been the backbone of the Indian economy, employing over half of the population. However, despite its critical role, farmers often grapple with significant challenges such as unpredictable weather conditions, deteriorating soil health, and volatile market dynamics. These issues result in inconsistent yields and financial instability, further exacerbating the struggles of rural communities. The AgroDoc application address these challenges through a robust and innovative platform that integrates cutting-edge technologies like Artificial Intelligence (AI) and Machine Learning (ML) into the agricultural domain. The application aims to equip farmers with actionable insights that enhance productivity, optimize costs, and foster sustainability.

One of the core strengths of AgroDoc lies in its ability to deliver precise and actionable data through its key features-monsoon analysis, soil health prediction, and crop recommendations. The monsoon analysis feature provides farmers with reliable weather forecasts, enabling them to plan agricultural activities like sowing, irrigation, and harvesting with precision. By reducing the unpredictability of weather conditions, this feature significantly lowers the risks associated with climate variability. Similarly, the soil health prediction module allows farmers to monitor and assess the quality of their soil. By analyzing soil nutrient levels and structural health, the application recommends appropriate fertilizers and amendments, ensuring that crops receive the necessary nourishment without overuse of resources. This fosters a balance between maximizing yields and preserving soil fertility for long-term agricultural sustainability. Additionally, the crop recommendation feature leverages market trends, climatic conditions, and soil analysis to suggest the most suitable crops for cultivation. This not only ensures higher profitability for farmers but also minimizes losses associated with planting unsuitable crops.

AgroDoc also contributes significantly to the broader goals of environmental and economic sustainability. By promoting efficient resource management, such as the optimal use of water, fertilizers, and pesticides, the application minimizes environmental degradation. The reduction in chemical usage not only benefits the ecosystem but also reduces production costs for farmers, thereby increasing their profit margins. Furthermore, the emphasis on soil conservation ensures that agricultural lands remain fertile and productive for generations to come. The climate adaptation capabilities of the app empower farmers to make informed decisions amidst changing environmental conditions, reducing their vulnerability to climate-related risks.

The economic benefits of AgroDoc extend beyond the individual farmer to the larger agricultural community. By offering real-time insights and modern solutions, the application bridges the gap between traditional farming practices and modern technological advancements. This digital transformation creates opportunities for farmers to access previously unavailable tools, enhancing their ability to compete in an increasingly globalized market. Additionally, the increased productivity and profitability supported by AgroDoc contribute to improved living standards and financial stability for rural households. This has a cascading effect on the socio-economic development of the farming community, fostering a culture of innovation and resilience in agriculture.

In conclusion, AgroDoc stands as a transformative solution for modern agriculture, addressing long-standing challenges with innovative technology. By integrating data-driven insights with user-friendly features, the application empowers farmers to make better decisions, optimize their resources, and improve their livelihoods. AgroDoc not only enhances agricultural productivity and profitability but also promotes sustainable and eco-friendly farming practices. Its ability to combine environmental conservation with economic empowerment ensures that it serves as a catalyst for long-term growth and resilience in the agricultural sector. As AgroDoc continues to eelve, it holds the potential to revolutionize farming practices, contributing to a more secure and sustainable future for farmers and the global food supply chain.

**REFERENCES**

1. https://www.researchgate.net/publication/372664253\_Sustainable\_C rop Protection via Robotics and Artificial Intelligence Solutions/ link/6580e9dc2468df72d3b70cbd/download
2. https://www.ir.jmets.com/uploadedfiles/paper//issue 1 january 2024 /48795/final/fin\_irjmets1705938824.pdf
3. <https://ieeexplore.ieee.org/document/8409427>
4. [https://www.sciencedirect.com/science/article/abs/pii/S016788099700 0261](https://www.sciencedirect.com/science/article/abs/pii/S016788099700%200261)
5. https://www.taylorfrancis.com/chapters/edit/10.1201/9781351072717
6. -2/integrated-crop-management-systems-pest-control-kamal-el-zik- raymond-frisbie
7. IRJMETS - Low cost journal with DOI IRs. 599 publication fees In 4 hr fast paper publicationl Engineering, Scientific journal
8. <https://www.researchgate.net/publication/372664253_Sustainable_Crop_Protection_via_Robotics_and_Artificial_Intelligence_Solutions?__cf_chl_tk=xuLOJviiWP3NuVaaAGVyIWh7.WR8Wxdg4qKcq_4pbhM-1736536055-1.0.1.1-itH9UkiexE6umsymqgwghM4MZbF3f_AwOL_YXziguMU>
9. <https://www.irjmets.com/>
10. <https://www.sciencedirect.com/science/article/abs/pii/S0167880997000261>
11. <https://www.springer.com/in>
12. <https://www.elsevier.com/>
13. <https://www.wiley.com/en-in>
14. <https://www.sciencedirect.com/science/article/pii/S0261219421002001>
15. <https://www.researchgate.net/publication/337581992_The_preoperative_use_of_stair_climbing_test_to_predict_postoperative_complications_in_thoracic_surgery_A_meta_analysis>
16. <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-7652.2008.00340.x>

**APPENDIX-A**

**PSUEDOCODE**

**APP.TSX:**

import React, { useEffect, useRef, useState } from 'react';

import { SafeAreaView, StyleSheet, BackHandler, ToastAndroid, PermissionsAndroid, Platform } from 'react-native';

import { WebView } from 'react-native-webview';

import AsyncStorage from '@react-native-async-storage/async-storage';

import RNFetchBlob from 'rn-fetch-blob';

//import messaging from '@react-native-firebase/messaging';

const App = () => {

  const webViewRef = useRef(null);

  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const [showLoader, setShowLoader] = useState(true); // State to handle showing the loader

  let backPressCount = 0;

  useEffect(() => {

    // Check if the user is logged in

    const checkLoginSession = async () => {

      const session = await AsyncStorage.getItem('userSession');

      if (session) {

        setIsLoggedIn(true);

      }

    };

    checkLoginSession();

    // Show loader for 3 seconds, then load the actual URL

    setTimeout(() => {

      setShowLoader(false); // Hide loader after 3-5 seconds

    }, 3000); // Adjust this value to change the delay (3000ms = 3 seconds)

    const onBackPress = () => {

      if (webViewRef.current) {

        webViewRef.current.goBack(); // Navigate back in WebView history

        if (backPressCount === 0) {

          ToastAndroid.show('Press back again to exit', ToastAndroid.SHORT);

          backPressCount += 1;

          setTimeout(() => {

            backPressCount = 0;

          }, 2000);

          return true; // Prevent the default behavior of closing the app

        } else {

          return false; // Allow the app to close

        }

      }

    };

    const backHandler = BackHandler.addEventListener('hardwareBackPress', onBackPress);

    return () => backHandler.remove();

  }, []);

  // Handle messages from WebView

  const onWebViewMessage = async (event) => {

    const message = event.nativeEvent.data;

    if (message === 'LOGIN\_SUCCESS') {

      // Store session in AsyncStorage when user logs in

      await AsyncStorage.setItem('userSession', 'loggedIn');

      setIsLoggedIn(true);

    }

  };

  // Request storage permission (for file downloads)

  const requestStoragePermission = async () => {

    if (Platform.OS === 'android' && Platform.Version < 30) { // Android 10 and below

      try {

        const granted = await PermissionsAndroid.request(

          PermissionsAndroid.PERMISSIONS.WRITE\_EXTERNAL\_STORAGE,

          {

            title: 'Storage Permission',

            message: 'This app needs access to your storage to download files.',

            buttonNeutral: 'Ask Me Later',

            buttonNegative: 'Cancel',

            buttonPositive: 'OK',

          }

        );

        if (granted !== PermissionsAndroid.RESULTS.GRANTED) {

          console.log('Storage permission denied');

        }

      } catch (err) {

        console.warn(err);

      }

    }

  };

  // Handle file download through WebView

  const handleFileDownload = (url) => {

    const { config, fs } = RNFetchBlob;

    const downloads = fs.dirs.DownloadDir;

    // Request permission first (only for Android 10 and below)

    requestStoragePermission();

    config({

      fileCache: true,

      addAndroidDownloads: {

        useDownloadManager: true,

        notification: true,

        path: `${downloads}/downloadedFile\_${Date.now()}.pdf`, // Save with the correct extension

        description: 'Downloading file...',

        mime: 'application/pdf',

        mediaScannable: true,

      },

    })

      .fetch('GET', url)

      .then((res) => {

        console.log('File downloaded successfully:', res.path());

        ToastAndroid.show('File downloaded successfully', ToastAndroid.LONG);

      })

      .catch((err) => {

        console.error('File download error:', err);

        ToastAndroid.show('Failed to download file', ToastAndroid.LONG);

      });

  };

/\*//Added

  async function requestUserPermission() {

    const authStatus = await messaging().requestPermission();

    const enabled =

      authStatus === messaging.AuthorizationStatus.AUTHORIZED ||

      authStatus === messaging.AuthorizationStatus.PROVISIONAL;

    if (enabled) {

      console.log('Authorization status:', authStatus);

    }

  }

  const App = () => {

    useEffect(() => {

      const unsubscribe = messaging().onMessage(async (remoteMessage) => {

        Alert.alert('A new FCM message arrived!', JSON.stringify(remoteMessage));

      });

      return unsubscribe;

    }, []);

    messaging().setBackgroundMessageHandler(async (remoteMessage) => {

      console.log('Message handled in the background!', remoteMessage);

    });

    useEffect(() => {

      messaging().onNotificationOpenedApp(remoteMessage => {

        console.log('Notification caused app to open from background state:', remoteMessage.notification);

        // Navigate to a specific screen if needed

      });

      // Check if the app was opened by tapping on a notification (cold start)

      messaging().getInitialNotification()

        .then(remoteMessage => {

          if (remoteMessage) {

            console.log('Notification caused app to open from quit state:', remoteMessage.notification);

            // Navigate to a specific screen if needed

          }

        });

    }, []);

//Added\*/

  return (

    <SafeAreaView style={styles.container}>

      <WebView

        ref={webViewRef}

        source={{ uri: showLoader ? 'https://projectforengineers.com/agrodoc/dashboard.php' : 'https://projectforengineers.com/agrodoc/dashboard.php' }} // Show loader first, then switch to dashboard

        javaScriptEnabled={true}

        domStorageEnabled={true}

        allowFileAccess={true} // Required for file access

        onShouldStartLoadWithRequest={(request) => {

          const { url } = request;

          if (url.endsWith('.pdf')) {

            handleFileDownload(url); // Handle PDF download manually

            return false; // Prevent WebView from loading the PDF URL

          }

          return true;

        }}

        onMessage={onWebViewMessage} // Capture messages from the WebView

      />

    </SafeAreaView>

  );

};

const styles = StyleSheet.create({

  container: {

    flex: 1,

  },

});

export default App;

**Monsoon Analysis:**

<?php

session\_start();

// Redirect to login if not authenticated

if (!isset($\_SESSION['user\_id'])) {

    header("Location: index.php");

    exit;

}

$user\_name = $\_SESSION['user\_name'];

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>User Dashboard</title>

    <link rel="stylesheet" href="css/dashboard.css">

</head>

<body>

    <?php include 'navbar.php';?>

    <div class="banner">

        <div class="banner-text">

            <h1>Welcome to Agrodoc</h1>

            <p>Your ultimate farming companion. Get personalized recommendations for crops, fertilizers, and more to maximize your yield and ensure sustainable farming practices.</p>

        </div>

    </div>

    <div class="container">

        <!-- Weather Report Section -->

        <div class="weather-app">

            <h2><i class="fa-solid fa-cloud-sun"></i> Real-Time <br> Weather Report</h2>

            <div class="search-box">

                <input type="text" id="city-input" placeholder="Enter city name" />

                <button id="search-button">Get Weather</button>

            </div>

            <div id="weather-result" class="weather-container">

                <!-- Weather details will be displayed here -->

            </div>

        </div>

        <!-- Predicted Monsoon Section -->

        <div class="monsoon-prediction">

            <h2><i class="fa-solid fa-calendar-day"></i> Predicted Monsoon</h2>

            <div id="monsoon-result" class="monsoon-container">

                <!-- Monsoon prediction will be displayed here -->

            </div>

        </div>

        <div class="soil">

            <a href="soil\_prediction.php">

                <img src="img/soil.png" alt="">

                Click for Soil Prediction

            </a>

        </div>

        <div class="button-group">

            <div class="button-card">

                <a href="view\_crops.php">

                    <img src="img/buys.jpeg" alt="">

                    Buy Crops

                </a>

            </div>

            <div class="button-card">

                <a href="add\_products.php">

                    <img src="img/buy.png" alt="buy">

                    Sell Crops

                </a>

            </div>

            <div class="button-card">

                <a href="see\_all\_crops.php">

                    <img src="img/crops.png" alt="">

                    View Crops

                </a>

            </div>

            <div class="button-card">

                <a href="see\_enquiries.php">

                    <img src="img/enquery.jpg" alt="">

                    View Enquiries

                </a>

            </div>

        </div>

    </div>

    <script>

        // WeatherStack API Key

        const apiKey = 'd26460d581467e08cb37567bcb89bfe6'; // Replace with your WeatherStack API key

        const searchButton = document.getElementById('search-button');

        const cityInput = document.getElementById('city-input');

        const weatherResult = document.getElementById('weather-result');

        const monsoonResult = document.getElementById('monsoon-result');

        // Get Current Location and fetch weather

        function getCurrentLocationWeather() {

            if (navigator.geolocation) {

                navigator.geolocation.getCurrentPosition((position) => {

                    const latitude = position.coords.latitude;

                    const longitude = position.coords.longitude;

                    getWeatherByCoordinates(latitude, longitude);

                }, (error) => {

                    weatherResult.innerHTML = "Unable to retrieve location. Please enter a city.";

                });

            } else {

                weatherResult.innerHTML = "Geolocation is not supported by this browser.";

            }

        }

        // Get weather based on coordinates

        async function getWeatherByCoordinates(lat, lon) {

            const apiUrl = `http://api.weatherstack.com/current?access\_key=${apiKey}&query=${lat},${lon}`;

            try {

                const response = await fetch(apiUrl);

                const data = await response.json();

                if (data.error) {

                    weatherResult.innerHTML = "Unable to retrieve weather data.";

                    return;

                }

                displayWeather(data);

            } catch (error) {

                weatherResult.innerHTML = "Error fetching weather data. Please try again later.";

            }

        }

        // Search Button Event Listener

        searchButton.addEventListener('click', () => {

            const cityName = cityInput.value;

            if (cityName === "") {

                weatherResult.innerHTML = "Please enter a city name.";

                return;

            }

            getWeather(cityName);

        });

        // Get weather based on city name

        async function getWeather(city) {

            const apiUrl = `http://api.weatherstack.com/current?access\_key=${apiKey}&query=${city}`;

            try {

                const response = await fetch(apiUrl);

                const data = await response.json();

                if (data.error) {

                    weatherResult.innerHTML = "City not found. Please try again.";

                    return;

                }

                displayWeather(data);

            } catch (error) {

                weatherResult.innerHTML = "Error fetching weather data. Please try again later.";

            }

        }

        // Display weather information

        function displayWeather(data) {

            const { location, current } = data;

            const city = location.name;

            const country = location.country;

            const temperature = current.temperature;

            const weatherDescription = current.weather\_descriptions[0];

            const humidity = current.humidity;

            const windSpeed = current.wind\_speed;

            weatherResult.innerHTML = `

                <h2> <i class="fa-solid fa-location-dot"></i> ${city},<br> ${country}</h2>

                <div class='weather-data'>

                <p><strong><i class="fa-solid fa-temperature-low"></i> Temp:</strong><br> ${temperature}°C</p>

                <p><strong><i class="fa-solid fa-cloud"></i> Desc:</strong><br> ${weatherDescription}</p>

                <p><strong><i class="fa-solid fa-droplet"></i> Hum:</strong><br> ${humidity}%</p>

                <p><strong><i class="fa-solid fa-cloud-bolt"></i> Wind Speed:</strong><br> ${windSpeed} km/h</p>

                </div>

            `;

        }

        // Automatically fetch weather for current location on page load

        getCurrentLocationWeather();

        // Predict Monsoon Season based on IST Date

        function predictMonsoon() {

    // Get current system time

    const now = new Date();

    const istOffset = 5.5 \* 60 \* 60 \* 1000; // IST offset in milliseconds

    const istTime = new Date(now.getTime() + istOffset);

    // Get IST date and month

    const month = istTime.getUTCMonth() + 1; // Months are zero-based

    const day = istTime.getUTCDate();

    let season;

    // Determine the season based on the month

    if (month >= 6 && month <= 9) {

        season = "South-West Monsoon";

    } else if (month >= 10 && month <= 11) {

        season = "North-East Monsoon";

    } else if ((month === 12 || month <= 2) || (month === 3 && day <= 15)) {

        season = "Winter Season";

    } else if (month >= 3 && month <= 5) {

        season = "Summer Season";

    } else {

        season = "Dry Season";

    }

    // Format date and time in IST

    const date = istTime.toLocaleDateString("en-IN", { day: "2-digit", month: "long", year: "numeric" });

    const time = istTime.toLocaleTimeString("en-IN");

    // Display the result

    const monsoonResult = document.getElementById('monsoon-result');

    monsoonResult.innerHTML = `

        <p><strong>Date:</strong> ${date}</p>

        <!--<p><strong>Time:</strong> ${time}</p>-->

        <p><strong>Predicted Monsoon:</strong> ${season}</p>

    `;

}

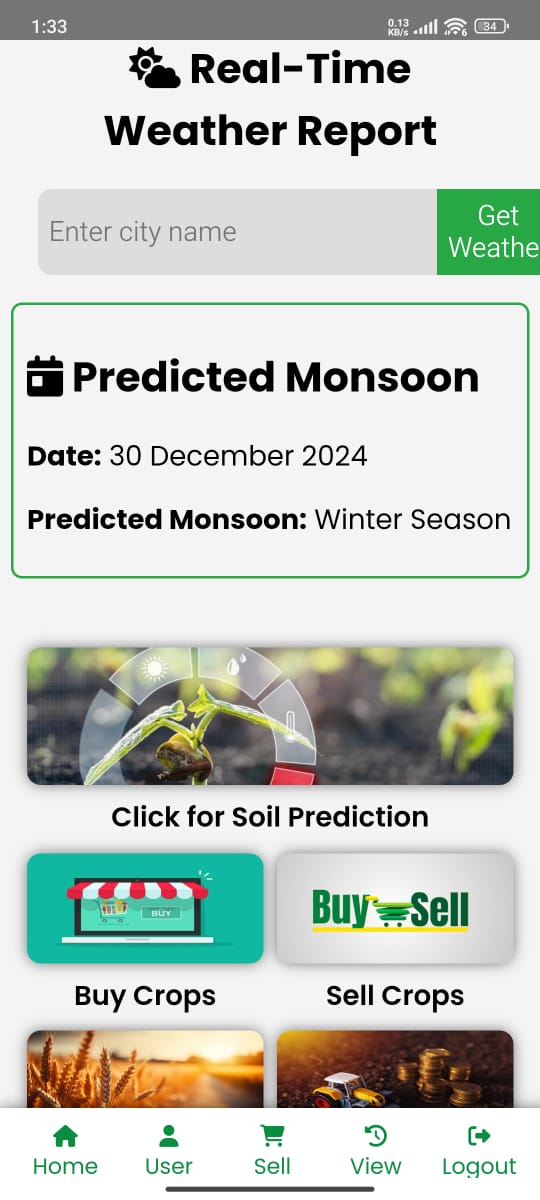
// Automatically display predicted monsoon on page load

predictMonsoon();

    </script>

</body>

</html>

****

**Fertilizer Recommendation AND Crop Recommendation**

<?php

$api\_key = "sk-I3wWMcy3BpSrLfPi1JmLUwsxiIksylbFZvsTMT3BlbkFJ-DsOq0g39xqp4PzdBae8o9ioZTEsa-l\_cd3CZ5amcA"; // Replace with your OpenAI API key

// Fetch form data

$data = filter\_input\_array(INPUT\_POST, [

    'ph' => FILTER\_VALIDATE\_FLOAT,

    'nitrogen' => FILTER\_VALIDATE\_INT,

    'phosphorus' => FILTER\_VALIDATE\_INT,

    'potassium' => FILTER\_VALIDATE\_INT,

    'calcium' => FILTER\_VALIDATE\_INT,

    'sulfur' => FILTER\_VALIDATE\_INT,

    'magnesium' => FILTER\_VALIDATE\_INT,

    'moisture' => FILTER\_VALIDATE\_INT,

    'zinc' => FILTER\_VALIDATE\_INT,

    'copper' => FILTER\_VALIDATE\_INT,

    'iron' => FILTER\_VALIDATE\_INT,

    'texture' => FILTER\_SANITIZE\_STRING,

    'organic\_matter' => FILTER\_SANITIZE\_STRING,

    'climate' => FILTER\_SANITIZE\_STRING,

    'age' => FILTER\_VALIDATE\_INT,

]);

// Ensure all fields are provided

foreach ($data as $key => $value) {

    if ($value === null || $value === false) {

        die("<h1>Error: All fields are required.</h1>");

    }

}

// Prepare OpenAI API request

$prompt = "Analyze the soil data provided below and give the following details:\n\n"

    . "1. Soil Type: Name the soil type.\n"

    . "2. Soil Color: Provide the typical color of the soil type.\n"

    . "3. Crop Recommendations: List suitable crops for this soil.\n"

    . "4. Fertilizer Recommendations: Organic and inorganic fertilizers.\n"

    . "5. States in India: Where this soil type is commonly found.\n\n"

    . json\_encode($data, JSON\_PRETTY\_PRINT);

// Function to call OpenAI API

function call\_openai\_api($api\_key, $model, $messages)

{

    $ch = curl\_init("https://api.openai.com/v1/chat/completions");

    curl\_setopt($ch, CURLOPT\_HTTPHEADER, [

        "Content-Type: application/json",

        "Authorization: Bearer $api\_key",

    ]);

    curl\_setopt($ch, CURLOPT\_POST, true);

    curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, true);

    curl\_setopt($ch, CURLOPT\_POSTFIELDS, json\_encode([

        "model" => $model,

        "messages" => $messages,

    ]));

    $response = curl\_exec($ch);

    curl\_close($ch);

    return $response ? json\_decode($response, true) : null;

}

// Call OpenAI API

$response = call\_openai\_api($api\_key, "gpt-4", [

    ["role" => "system", "content" => "You are an expert soil analyst and agricultural advisor."],

    ["role" => "user", "content" => $prompt],

]);

// Extract response

if ($response && isset($response['choices'][0]['message']['content'])) {

    $result = $response['choices'][0]['message']['content'];

} else {

    file\_put\_contents("api\_error\_log.txt", "API failed: " . json\_encode($response) . PHP\_EOL, FILE\_APPEND);

    die("<h1>Error: Unable to fetch soil analysis. Please try again later.</h1>");

}

// Parse response for soil type and color

preg\_match("/(?:soil type|type of soil|classification)[:\\s]\*(.+?)(?=\\.|\\n|$)/i", $result, $soil\_type\_match);

$soil\_name = $soil\_type\_match[1] ?? 'Unknown';

preg\_match("/(?:soil color|color of soil|typical color)[:\\s]\*(.+?)(?=\\.|\\n|$)/i", $result, $soil\_color\_match);

$soil\_color = $soil\_color\_match[1] ?? 'Unknown';

// Parse other response sections

preg\_match("/crops[:\\s]\*(.\*?)(?=fertilizer|states)/is", $result, $crops\_match);

$crops = $crops\_match[1] ?? 'Crop recommendations not available.';

preg\_match("/fertilizer recommendations[:\\s]\*(.\*?)(?=states)/is", $result, $fertilizer\_match);

$fertilizers = $fertilizer\_match[1] ?? 'Fertilizer recommendations not available.';

preg\_match("/states in India[:\\s]\*(.\*)/is", $result, $states\_match);

$states = $states\_match[1] ?? 'State information not available.';

// Display results

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Soil Analysis Result</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 20px auto;

            max-width: 800px;

            padding: 20px;

            border: 1px solid #ddd;

            border-radius: 10px;

            box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

        }

        .container{

            padding: 20px;

            background-color: white;

            margin: 80px auto;

            border-radius: 8px;

        }

        .container h1{

            background: #06c854;

            color: white;

            padding: 2px;

            font-size: 1.6rem;

            text-align: center;

        }

        .result {

            padding: 10px;

            border: 1px solid #4CAF50;

            border-radius: 5px;

            background-color: #f9f9f9;

        }

        .section {

            margin-bottom: 15px;

        }

        .btn{

            text-align: center;

            padding: 10px;

            font-weight: 600;

            border: 2px solid #4CAF50;

            border-radius: 5px;

            color: #4CAF50;

        }

    </style>

</head>

<body>

    <?php include 'navbar.php';?>

    <div class="container">

        <h1>Soil Analysis Result</h1>

        <div class="result">

            <div class="section">

                <p><strong>Soil Name:</strong> <?= htmlspecialchars($soil\_name); ?></p>

                <p><strong>Soil Color:</strong> <?= htmlspecialchars($soil\_color); ?></p>

            </div>

            <div class="section">

                <p><strong>Crop Recommendations:</strong></p>

                <p><?= nl2br(htmlspecialchars($crops)); ?></p>

            </div>

            <div class="section">

                <p><strong>Fertilizer Recommendations:</strong></p>

                <p><?= nl2br(htmlspecialchars($fertilizers)); ?></p>

            </div>

            <div class="section">

                <p><strong>States in India:</strong></p>

                <p><?= nl2br(htmlspecialchars($states)); ?></p>

            </div>

        </div><br>

        <a href="soil\_prediction.php" class='btn'>Go Back</a>

    </div>

</body>

</html>

<?php

session\_start();

// Redirect to login if not authenticated

if (!isset($\_SESSION['user\_id'])) {

    header("Location: login.php");

    exit;

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Soil Classification Form</title>

    <link rel="stylesheet" href="css/1index.css">

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 20px;

            max-width: 800px;

        }

        .container{

            padding: 20px;

            background-color: white;

            margin: 80px auto;

            border-radius: 8px;

        }

        .container h1{

            background: #06c854;

            color: white;

            padding: 2px;

            font-size: 1.6rem;

            text-align: center;

        }

        form {

            display: flex;

            flex-direction: column;

            gap: 10px;

        }

        label {

            font-weight: bold;

        }

        input, select {

            padding: 10px;

            font-size: 16px;

            border: 1px solid #ccc;

            border-radius: 5px;

        }

        button {

            padding: 10px;

            background-color: #4CAF50;

            color: white;

            font-size: 16px;

            border: none;

            border-radius: 5px;

            cursor: pointer;

        }

        button:hover {

            background-color: #45a049;

        }

        .grid{

            display: grid;

            grid-template-columns: repeat(2, 1fr);

            gap: 10px;

        }

        .form-group{

            display: flex;

            flex-direction: column;

            gap: 5px;

        }

        #loader {

            width: 100%;

            height: 100vh;

            display: none; /\* Hidden by default \*/

            position: fixed;

            left: 50%;

            top: 50%;

            transform: translate(-50%, -50%);

            z-index: 1000;

            background-color: rgba(0, 0, 0, 0.8);

            padding: 20px;

            border-radius: 8px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);

        }

        #loader img{

            position: fixed;

            left: 50%;

            top: 50%;

            transform: translate(-50%, -50%);

            width: 50px;

        }

        #loader p{

            color: #fff;

            position: fixed;

            left: 50%;

            top: 55%;

            transform: translate(-50%, -55%);

        }

    </style>

</head>

<body>

    <?php include 'navbar.php';?>

    <div class="container">

        <h1>Soil Prediction Form</h1>

        <form id="soilForm" action="process\_soil.php" method="POST">

            <div class="grid">

                <div class="form-group">

                    <label for="ph">pH Level (1-14):</label>

                    <input type="number" step="0.1" min="1" max="14" id="ph" name="ph" required>

                </div>

                <div class="form-group">

                    <label for="nitrogen">Nitrogen Level (1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="nitrogen" name="nitrogen" required>

                </div>

            </div>

            <div class="grid">

                <div class="form-group">

                    <label for="phosphorus">Phosphorus Level <br>(1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="phosphorus" name="phosphorus" required>

                </div>

                <div class="form-group">

                    <label for="potassium">Potassium Level <br> (1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="potassium" name="potassium" required>

                </div>

            </div>

            <div class="grid">

                <div class="form-group">

                    <label for="calcium">Calcium Level <br>(1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="calcium" name="calcium" required>

                </div>

                <div class="form-group">

                    <label for="sulfur">Sulfur Level <br>(1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="sulfur" name="sulfur" required>

                </div>

            </div>

            <div class="grid">

                <div class="form-group">

                    <label for="magnesium">Magnesium Level (1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="magnesium" name="magnesium" required>

                </div>

                <div class="form-group">

                    <label for="moisture">Moisture Content (1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="moisture" name="moisture" required>

                </div>

            </div>

            <div class="grid">

                <div class="form-group">

                    <label for="zinc">Zinc Level (1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="zinc" name="zinc" required>

                </div>

                <div class="form-group">

                    <label for="copper">Copper Level (1-100):</label>

                    <input type="number" step="0.1" min="1" max="100" id="copper" name="copper" required>

                </div>

            </div>

            <div class="form-group">

                <label for="iron">Iron Level (1-100):</label>

                <input type="number" step="0.1" min="1" max="100" id="iron" name="iron" required>

            </div>

            <label for="texture">Soil Texture:</label>

            <select id="texture" name="texture" required>

                <option value="">Select Texture</option>

                <option value="Sandy">Sandy</option>

                <option value="Clayey">Clayey</option>

                <option value="Loamy">Loamy</option>

                <option value="Silty">Silty</option>

            </select>

            <label for="organic\_matter">Organic Matter Content (Low, Medium, High):</label>

            <select id="organic\_matter" name="organic\_matter" required>

                <option value="">Select Organic Matter Content</option>

                <option value="Low">Low</option>

                <option value="Medium">Medium</option>

                <option value="High">High</option>

            </select>

            <label for="climate">Local Climate:</label>

            <select id="climate" name="climate" required>

                <option value="">Select Climate</option>

                <option value="Arid">Arid</option>

                <option value="Tropical">Tropical</option>

                <option value="Temperate">Temperate</option>

                <option value="Cold">Cold</option>

            </select>

            <label for="age">Soil Age (in years):</label>

            <input type="number" step="0.1" min="1" max="10000" id="age" name="age" required>

            <button type="submit">Submit</button>

        </form>

    </div>

    <div id="loader">

        <img src="img/loader.gif" alt="">

        <p>AI Processing... Please wait...</p>

    </div>

    <script>

        // JavaScript to handle form submission

        document.getElementById('soilForm').onsubmit = function() {

            // Show the loader

            document.getElementById('loader').style.display = 'block';

        };

    </script>

</body>

</html>

**Login AND Logout**

<?php

session\_start();

require 'config.php';

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

    $email = htmlspecialchars($\_POST['email']);

    $password = $\_POST['password'];

    // Fetch user details from the database

    $stmt = $conn->prepare("SELECT \* FROM users WHERE email = :email");

    $stmt->execute(['email' => $email]);

    $user = $stmt->fetch(PDO::FETCH\_ASSOC);

    if ($user && password\_verify($password, $user['password'])) {

        // Set session variables

        $\_SESSION['user\_id'] = $user['id'];

        $\_SESSION['user\_name'] = $user['name'];

        // Redirect to dashboard

        header("Location: dashboard.php");

        exit;

    } else {

        $error = "Invalid email or password!";

    }

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login</title>

    <link rel="stylesheet" href="css/login.css">

</head>

<body>

<div class="container">

        <h2>Login</h2>

        <?php if (isset($error)) echo "<p class='error'>$error</p>"; ?>

        <form method="POST" action="">

            <div class="form-group">

                <label for="email">Email</label>

                <input type="email" id="email" name="email" required>

            </div>

            <div class="form-group">

                <label for="password">Password</label>

                <input type="password" id="password" name="password" required>

            </div>

            <div class="form-group">

                <button type="submit">Login</button>

            </div>

        </form>

        <div class="register-link">

            <a href="register.php">Dont have account? Register Here</a>

        </div>

    </div>

</body>

</html>

<?php

session\_start();

session\_destroy();

header("Location: index.php");

exit;

**Buy and sell option code**

<?php

session\_start();

include('config.php');

// Redirect to login if not authenticated

if (!isset($\_SESSION['user\_id'])) {

    header("Location: login.php");

    exit;

}

$user\_id = $\_SESSION['user\_id'];

// Initialize search query

$searchTerm = isset($\_POST['search']) ? $\_POST['search'] : '';

// Prepare the base query

$query = "SELECT \* FROM former\_products WHERE user\_id != :user\_id";

// If there's a search term, modify the query to include the LIKE conditions

if ($searchTerm) {

    $query .= " AND (crop\_name LIKE :search OR

                     user\_id IN (SELECT id FROM users WHERE name LIKE :search OR number LIKE :search OR email LIKE :search OR city LIKE :search))";

}

// Prepare and execute the query

$stmt = $conn->prepare($query);

$stmt->bindParam(':user\_id', $user\_id, PDO::PARAM\_INT);

if ($searchTerm) {

    $searchTerm = "%$searchTerm%"; // Add wildcards for LIKE search

    $stmt->bindParam(':search', $searchTerm, PDO::PARAM\_STR);

}

$stmt->execute();

$products = $stmt->fetchAll(PDO::FETCH\_ASSOC);

// Fetch user details for each product

function getUserDetails($user\_id) {

    global $conn;

    $query = "SELECT name, number, email, city FROM users WHERE id = :user\_id";

    $stmt = $conn->prepare($query);

    $stmt->bindParam(':user\_id', $user\_id, PDO::PARAM\_INT);

    $stmt->execute();

    return $stmt->fetch(PDO::FETCH\_ASSOC);

}

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>View Crops</title>

    <style>

        /\* Basic styling for the page \*/

        body {

            font-family: Arial, sans-serif;

            background-color: #f4f4f4;

            margin: 0;

            padding: 0;

        }

        .container {

            padding: 20px;

            background-color: white;

            margin: 50px auto;

            border-radius: 8px;

        }

        .product {

            box-shadow: 0 0 10px rgba(0,0,0,0.3);

            border-bottom: 1px solid #ddd;

            padding: 20px;

        }

        .product h3 {

            margin: 0;

        }

        .product p {

            margin: 5px 0;

        }

        .contact-btn {

            background-color: #06c854;

            color: white;

            padding: 10px 20px;

            text-decoration: none;

            border-radius: 5px;

            margin-top: 10px;

        }

        .contact-btn:hover {

            background-color: #0056b3;

        }

        .search-bar {

            margin-bottom: 20px;

        }

        .search-bar input {

            padding: 10px;

            width: 200px;

            margin-right: 10px;

            border-radius: 5px;

            border: 1px solid #ddd;

        }

        .search-bar button {

            padding: 10px 20px;

            background-color: #06c854;

            color: white;

            border: none;

            border-radius: 5px;

        }

        .search-bar button:hover {

            background-color: #0056b3;

        }

    </style>

</head>

<body>

    <?php include 'navbar.php';?>

    <div class="container">

        <h2>All Crops</h2>

        <!-- Search Form -->

        <form class="search-bar" method="POST" action="">

            <input type="text" name="search" placeholder="Search by name, email, number, crop name, city" value="<?php echo htmlspecialchars($searchTerm); ?>">

            <button type="submit">Search</button>

        </form>

        <!-- Display Products -->

        <?php if ($products): ?>

            <?php foreach ($products as $product): ?>

                <?php

                    $userDetails = getUserDetails($product['user\_id']);

                ?>

                <div class="product">

                    <h3><?php echo htmlspecialchars($product['crop\_name']); ?></h3>

                    <p><strong>Quantity:</strong> <?php echo htmlspecialchars($product['quantity']); ?></p>

                    <p><strong>Selling Price:</strong> ₹<?php echo htmlspecialchars($product['selling\_price']); ?></p>

                    <p><strong>Location:</strong> <?php echo htmlspecialchars($product['location']); ?></p>

                    <p><strong>Posted On:</strong> <?php echo htmlspecialchars($product['created\_at']); ?></p>

                    <p><strong>Seller Name:</strong> <?php echo htmlspecialchars($userDetails['name']); ?></p>

                    <p><strong>Seller Contact:</strong> <?php echo htmlspecialchars($userDetails['number']); ?></p>

                    <p><strong>Seller Email:</strong> <?php echo htmlspecialchars($userDetails['email']); ?></p>

                    <p><strong>Seller City:</strong> <?php echo htmlspecialchars($userDetails['city']); ?></p><br>

                    <a href="contact\_form.php?product\_id=<?php echo $product['id']; ?>" class="contact-btn">Contact Seller</a>

                </div><br>

            <?php endforeach; ?>

        <?php else: ?>

            <p>No products found matching your search criteria.</p>

        <?php endif; ?>

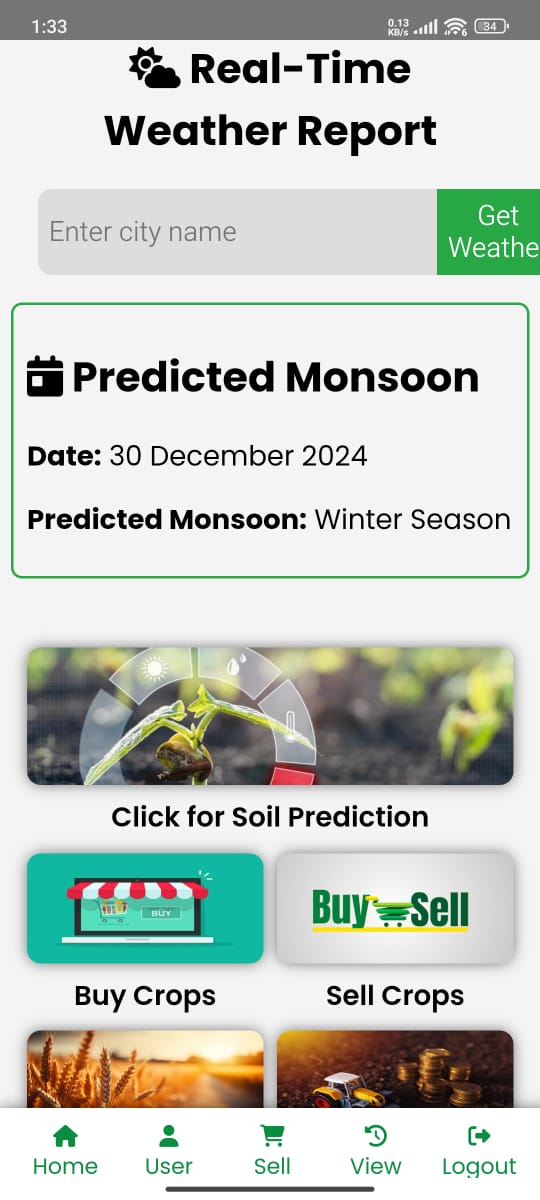
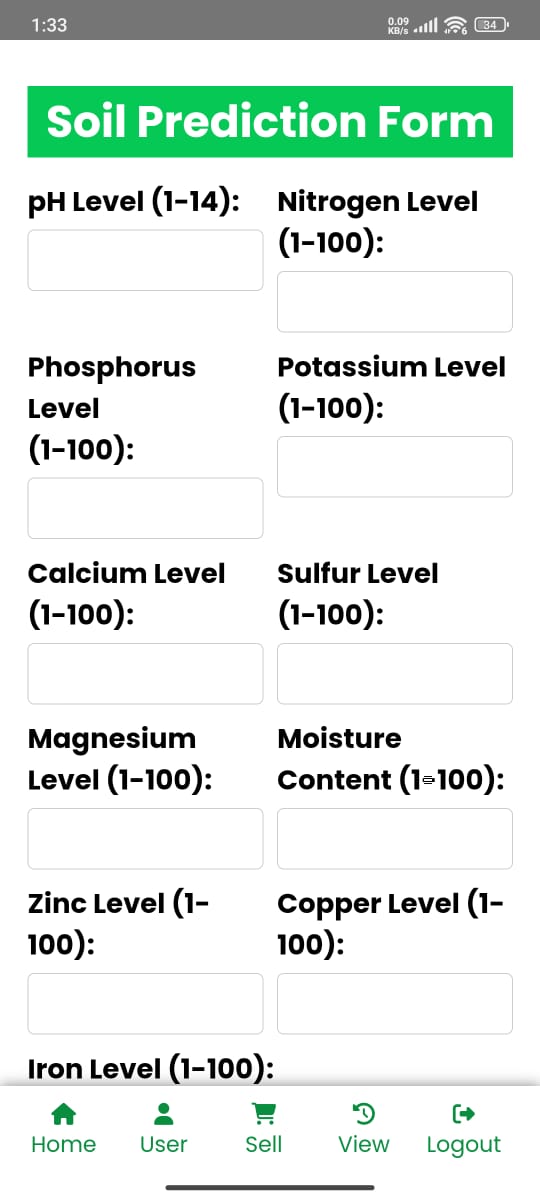
    </div>

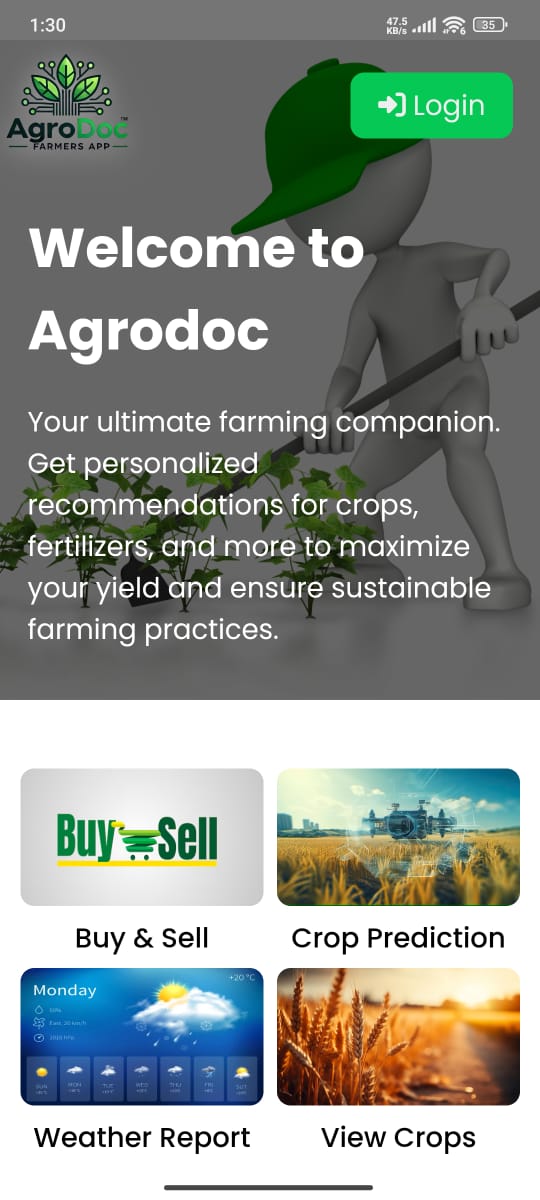
</body>

</html>

**APPENDIX-B**

**SCREENSHOTS**

** **

****

**APPENDIX-C**

**ENCLOSURES**

**Details of mapping the project with the Sustainable Development Goals (SDGs).**

****

1. **Zero Hunger (GOAL 2)**

The project works towards **SDG-2**: Is one of the United Nations' Sustainable Development Goals (SDGs), aiming to end hunger, achieve food security, and improve nutrition while promoting sustainable agriculture. Hunger remains a critical global issue, with millions of people still lacking access to sufficient, safe, and nutritious food. This goal emphasizes the need to address the root causes of hunger, such as poverty, conflict, climate change, and inefficient food systems. It promotes sustainable agricultural practices, investment in rural infrastructure, and equitable access to resources such as land, technology, and markets. Achieving Zero Hunger also involves reducing food waste, ensuring food production systems are resilient, and fostering international cooperation to support developing countries. By ensuring everyone has access to adequate nutrition, Goal 2 seeks to improve livelihoods, health, and overall well-being, contributing to a world free from hunger and malnutrition

**2. Good Health and Well-Being (Goal 3)**

The project aligns with **SDG-3:** is a key United Nations Sustainable Development Goal (SDG) that focuses on ensuring healthy lives and promoting well-being for all people at all ages. It emphasizes reducing maternal and child mortality, combating infectious diseases like HIV/AIDS, malaria, and tuberculosis, and addressing non-communicable diseases such as heart disease, cancer, and diabetes. Goal 3 aims to achieve universal health coverage, including access to quality essential healthcare services, safe and affordable medicines, and vaccines for all. It also prioritizes mental health, reducing substance abuse, and strengthening health systems, especially in low- and middle-income countries. Preventative measures, such as improving sanitation, providing clean water, and promoting healthy lifestyles, play a crucial role. Through collaborative efforts and increased investment in healthcare infrastructure, research, and training, Goal 3 seeks to build resilient healthcare systems and improve global health by 2030, fostering a world where everyone can lead healthier and more productive lives.

**3.Industry, Innovation, and Infrastructure(Goal 9)**

The project aligns with SDG-9: Industry, Innovation, and Infrastructure by leveraging cutting-edge technologies like artificial intelligence (AI) and machine learning (ML) to revolutionize healthcare diagnostics. By integrating these advanced tools, the project fosters innovation in detecting and analyzing blood cancer at its early stages, thus enhancing the efficiency and accuracy of medical diagnosis. Such technological advancements not only improve patient outcomes but also contribute to building robust healthcare infrastructure that is capable of addressing complex medical challenges. Furthermore, the project encourages the adoption of innovative approaches within the healthcare industry, paving the way for sustainable development and modernized healthcare systems that are accessible, efficient, and scalable. This alignment with SDG-9 highlights the importance of technological progress in driving industrial growth and improving public health infrastructure.

**4.Responsible Consumption and Production(GOAL 12)**

is a United Nations Sustainable Development Goal (SDG) aimed at ensuring sustainable consumption and production patterns. This goal focuses on reducing waste, using resources efficiently, and promoting environmentally friendly practices throughout the supply chain. It emphasizes the need for businesses to adopt sustainable practices and integrate sustainability information into their reporting cycles. Goal 12 also calls for reducing food waste at both the production and consumer levels, managing chemicals and waste responsibly, and minimizing their adverse impact on human health and the environment. Additionally, it encourages individuals to make informed choices and adopt sustainable lifestyles. By fostering innovation, improving resource efficiency, and promoting sustainable infrastructure, Goal 12 seeks to decouple economic growth from environmental degradation. Achieving this goal is essential for conserving natural resources, protecting ecosystems, and ensuring a more equitable and sustainable future for all.

**5.Climate Action(Goal 13)**

Is a critical United Nations Sustainable Development Goal (SDG) that focuses on urgent measures to combat climate change and its impacts. Climate change poses a severe threat to the environment, economies, and human well-being, with rising global temperatures, extreme weather events, and sea-level rise affecting millions worldwide. Goal 13 calls for strengthened resilience and adaptive capacity to climate-related hazards, integrating climate change measures into national policies, strategies, and planning. It emphasizes the importance of education, awareness, and capacity-building to empower individuals and communities to take climate action. Additionally, Goal 13 advocates for developed nations to fulfill their commitments to mobilize financial resources and technology transfer to support developing countries in mitigating and adapting to climate change. By promoting international cooperation and sustainable practices, this goal aims to limit global warming, protect ecosystems, and ensure a sustainable future for

**2.Journal publication/Conference Paper Presented Certificates of all students**

****







**3.Include certificate(s) of any Achievement/Award won in any project-related event**

**Link:** [**https://www.doi.org/10.55041/IJSREM40681**](https://www.doi.org/10.55041/IJSREM40681)

[**https://ijsrem.com/download/integrated-crop-protection-management/**](https://ijsrem.com/download/integrated-crop-protection-management/)

**4.Similarity Index / Plagiarism Check report clearly showing the Percentage (14%). No need for a page-wise explanation.**

