**FARMING FUTURES: PREDICTIVE AGRICULTURE SOLUTIONS**

## A PROJECT REPORT

***Submitted by,***

|  |  |
| --- | --- |
| **20201COM0059** | **ABHIRAJ R** |
| **20201COM0035** | **DEEKSHITH B M** |
| **20201COM0031** | **ATHARSH** |
| **20201COM0033** | **ABHIRAM** |

### *Under the guidance of,*

**MR. HASEEB KHAN**

**Assistant Professor**

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

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER ENGINEERING**

**At**



**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**

**PRESIDENCY UNIVERSITY**

**BENGALURU**

**JANUARY 2024**

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE**

This is to certify that the Project report **“FARMING FUTURES: PREDICTIVE AGRICULTURE SOLUTIONS”** being submitted by “Abhiraj R, Deekshith B M, Atharsh, Abhiram” bearing roll number(s) “20201COM0059, 20201COM0035, 20201COM0031, 20201COM0033” in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

|  |  |
| --- | --- |
| **Mr. HASEEB KHAN**  Assistant Professor  School of CSE&IS  Presidency University | **Dr. GOPAL K. SHYAM**  Assistant Professor & HOD  School of CSE&IS  Presidency University |

|  |  |  |
| --- | --- | --- |
| **Dr. C. KALAIARASAN**  Associate Dean  School of CSE&IS  Presidency University | **Dr. SHAKKEERA L**  Associate Dean  School of CSE&IS  Presidency University | **Dr. SAMEERUDDIN KHAN** 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 **FARMING FUTURES: PREDICTIVE AGRICULTURE SOLUTIONS** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Engineering**, is a record of our own investigations carried under the guidance of **Mr. HASEEB KHAN**, Assistant 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.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Roll Number** | **Student Name** | **Signature** |
| 1 | 20201COM0059 | ABHIRAJ R |  |
| 2 | 20201COM0035 | DEEKSHITH |  |
| 3 | 20201COM0031 | ATHARSH |  |
| 4 | 20201COM0033 | ABHIRAM |  |

**ABSTRACT**

This smart farming project represents a groundbreaking initiative to transform traditional agricultural methods through the integration of cutting-edge technologies, specifically focusing on precise crop prediction. Utilizing sophisticated machine learning algorithms, the system analyzes extensive datasets comprising weather patterns, soil quality, and historical crop yields. The central objective is to equip farmers with powerful data-driven tools that empower them to make informed decisions, guiding the selection of crops best suited to their unique environmental circumstances. The project spans various stages, encompassing data collection facilitated by sensors and satellite technology, algorithm development for accurate predictions, and the creation of a user-friendly interface for farmers. Continuous testing, feedback loops, and updates ensure the ongoing accuracy 70% and relevance of the system. By merging technology with agricultural expertise, this project envisions a future where smart farming significantly contributes to heightened productivity, enhanced resource efficiency, and the promotion of sustainable agricultural practices.

**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**, Dean, School of Computer Science and Engineering and School of Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans  **Dr. Kalaiarasan C and Dr. Shakkeera L,** School of Computer Science and Engineering and School of Information Science, Presidency University and **Dr. Gopal K. Shyam,** Head of the Department, School of Computer Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Mr. Haseeb Khan**, School of Computer Science and Engineering, Presidency University for her 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 University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Mrutyunjaya MS** and also the department Project Coordinators are **Dr. Sasidhar Babu S, Dr. Sudha P, Ms. Yogeetha B. R.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**Abhiraj R**

**Deekshith B M**

**ATHARSH J R**

**ABHIRAM**

**LIST OF TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Table Name** | **Table Caption** | **Page No.** |
| 1 | Table 1.1 | Gant Chart Table | 31 |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Figure Name** | **Caption** | **Page No.** |
| 1 | Figure 1.1 | Use case diagram | 16 |
| 2 | Figure 1.2 | State diagram | 18 |
| 3  4  5  6  7  8  9 | Figure 1.3  Figure 2.1  Figure 2.2  Figure 2.3  Figure 3.1  Figure 3.2  Figure 3.3 | System Architecture  Random forest classifier  Gantt chart  Homepage  Demo 01  Demo 02  Demo 03 | 21  27  31  49  50  51  52 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **DECLARATION**  **ABSTRACT**  **ACKNOWLEDGMENT** | **i**  **ii**  **iii** |
| **1** | **INTRODUCTION** | 1 |
|  | 1.1 BACKGROUND | 1 |
|  | 1.2 PROBLEM STATEMENT | 3 |
|  | 1.3 SCOPE | 3 |
| **2** | **LITERATURE SURVEY** | 4 |
| **3** | **RESEARCH GAPS OF EXISTING SYATEMS** | 11 |
|  | 3.1 EXISTING METHODS | 11 |
|  | 3.2 LIMITATION OF EXISTING SYTEM | 11 |
|  | 3.3 OBJECTIVES | 11 |
| **4** | **PROPOSED METHODOLOGY** | 12 |
|  | 4.1 ADVANTAGES  4.2 DISADVANTAGES |  |
| **5** | **HARDWARE AND SOFTWARE REQUIREMENTS** | 15 |
| **6** | **SYSTEM DESIGN & IMPLEMENTATION** | 16 |
|  | 6.1 USE CASE DIAGRAM | 16 |
|  | 6.2 STATE DIAGRAM | 18 |
|  | 6.3 SYSTEM ARCHITECTURE DIAGRAM | 21 |
|  | 6.4 ABOUT PYTHON AND MACHINE LEARNING | 23 |
|  | 6.5 ALGORITHM DETAILS | 26 |
|  | 6.6 SOURCE CODE DETAILS | 28 |
| **7** | **IMPLEMENTATION FOR EXECUTION OF PROJECT** | 31 |
| **8** | **OUTCOMES** | 32 |
| **9** | **RESULTS AND DISCUSSIONS** | 34 |
|  | 9.1 RESULTS | 34 |
|  | 9.1.1 VARIOUS LEVEL OF TESTING | 34 |
|  | 9.1.2 WHITE BOX TESTING | 34 |
|  | 9.1.3 BLACK BOX TESTING  9.1.4 UNIT TESTING  9.1.5 FUNCTIONAL TESTING  9.1.6. PERFORMANCE TESTING  9.1.7 INTEGRATION TESTING  9.1.8 VALIDATION TESTING  9.1.9 SYSTEM TESTING  9.1.10 OUTPUT TESTING  9.1.11 USER ACCEPTANCE TESTING | 34  36  36  37  37  38  38  39  39 |
| **10** | **CONCLUSION** | 40 |
| **11** | **REFERENCE** | 41 |
| **12** | **APENDIX** | 43 |
|  | 1. PSEUDOCODE | 43 |
|  | 1. SCREENSHOTS | 50 |
|  |  |  |
| **13** | **PLAGIARISM REPORT** | 55 |

**CHAPTER-1**

**INTRODUCTION**

* 1. **BACKGROUND**

Agriculture serves as the backbone of the Indian economy, requiring consistent support and improvement. Innovations in equipping agricultural machinery have played a crucial role in making farmers' work more efficient and manageable. The introduction of advanced machinery and technologies has paved the way for easier and more streamlined farming and cultivation processes.

In alignment with these advancements, we are developing a web application that integrates data analysis to optimize crop selection based on soil pH levels. This application serves as a valuable tool for farmers, guiding them in determining the most suitable crops for their specific lands. The focus is not only on enhancing agricultural productivity but also on financial analysis, aiding farmers in making informed decisions about the crops that will thrive in their soil conditions. This initiative strives to bridge the gap between technological innovation and traditional farming wisdom, ultimately contributing to a more sustainable and efficient agricultural future.

The essence of our initiative lies in leveraging machine learning algorithms to usher in a new era of precision farming. By meticulously analyzing diverse datasets encompassing critical factors such as weather patterns, soil quality, and historical crop yields, our system empowers farmers with data-driven decision-making tools. The overarching objective is to guide farmers towards selecting crops that align seamlessly with the unique environmental conditions of their lands, thus optimizing productivity.

Traditionally, farmers relied on their experience and knowledge passed down through generations. However, with new technologies, especially machine learning, we can take farming to the next level. Our aim is to help farmers make better decisions about what crops to plant by analyzing lots of data, including information about the weather, the quality of the soil, and how well crops have done in the past. This data is then fed into machine learning algorithms, which are like smart programs that can learn and make predictions.

The result is a tool that farmers can use to choose the best crops for their specific land, increasing their chances of a successful harvest.

Crops play a pivotal role in providing sustenance and fibre for the global population. The World Resource Institute is actively addressing the challenge of sustainably feeding ten billion people by 2050. Enhancing the yield of high-quality crops holds paramount importance in achieving this goal. The selection of crops profoundly influences both crop yields and financial returns. The unpredictability of successful crop cultivation, compounded by climate change and environmental variables, poses a considerable challenge for farmers.

This project leverages machine learning to offer crop recommendations to farmers. It involves the collection and pre-processing of datasets, followed by the training and testing of models utilizing key features such as soil content and type, soil pH value, temperature, humidity, and rainfall. Feature engineering concepts are explored to assess whether combining different features enhances model performance, with the newly created features being integrated into the dataset. Addressing the specific challenges that agriculture faces, we delve into the intricacies of machine learning in this context. Additionally, the project proposes innovative ideas for readers to explore.

The fusion of machine learning techniques with agricultural data equips farmers with the capacity to make informed decisions. From crop selection and irrigation management to pest control and yield prediction, this integration enables farmers to enhance efficiency, optimize resources, and adopt sustainable practices. The ultimate outcome is improved productivity and profitability within the agricultural sector.

**1.2 PROBLEM STATEMENT**

Farming in India faces problems like unpredictable weather and different soil types, making it tough for farmers to decide which crops to grow. Many farmers still use traditional methods, and they might not have access to modern tools that can help them make better choices. The challenge is to bring new technology to farming in a way that's easy for farmers to use. We need a solution that uses smart tools, like machine learning, to look at things like weather and soil conditions. This way, farmers can make smarter decisions about which crops will do well on their land. Another issue is that many farmers find it hard to understand and use these new technologies. We want to create a solution that is simple and user-friendly, ensuring that even farmers who are not about tech can benefit from it.

**1.3 SCOPE**

Our project is to revolutionize agriculture in India through the integration of user-friendly smart farming technologies. Our focus is on creating a comprehensive system that guides farmers in making optimal crop choices based on essential factors such as weather conditions and soil quality. By leveraging machine learning algorithms, we aim to provide a tool that simplifies decision-making for farmers, enhancing crop yields and overall productivity.

Furthermore, we recognize the importance of financial considerations in farming. Therefore, our project incorporates a financial analysis component, helping farmers make informed decisions not only about crop selection but also about the economic aspects of their agricultural practices.

**CHAPTER-2**

**LITERATURE SURVEY**

1. **Crop Prediction using Machine Learning**

**Journal Details:** Published in 2020 by M. Kalimuthu, P. Vaishnavi, M. Kishore

**Abstract:** Agriculture is crucial for India's economy, but climate change is affecting food production. This research aims to help novice farmers predict suitable crops using machine learning, specifically the Naive Bayes algorithm. The study collects seed data with factors like temperature and humidity, essential for successful crop growth. A website is being developed, allowing users to input parameters for location-based crop predictions. The goal is to support farmers in making informed decisions and enhance crop yields despite unpredictable climatic conditions.

**Advantages:**

* **Informed Decision-Making:** The use of machine learning, particularly the Naive Bayes algorithm, allows farmers to make more informed decisions about which crops to plant based on relevant parameters like temperature and humidity.
* **Increased Efficiency:** By leveraging technology for crop prediction, farmers can optimize their resource usage, leading to increased efficiency in terms of time, effort, and resources.
* **Improved Yield:** The research aims to enhance crop yields by guiding farmers to choose crops that are more likely to thrive in their specific environmental conditions, ultimately contributing to better productivity.
* **Accessibility through Website:** Developing a website for the application makes the tool accessible to a broader audience. Farmers can easily access and use the prediction tool from their computers or mobile devices.

**Limitations:**

* **Dependency on Historical Data:** The effectiveness of machine learning algorithms, including Naive Bayes, relies on historical data. If the training data is not representative or lacks diversity, the predictions may be less accurate.
* **Data Accuracy:** The success of the prediction model heavily depends on the accuracy of the input data. If the collected data is not precise or regularly updated, the predictions may not be reliable.
* **Technological Accessibility:** The effectiveness of the tool assumes that farmers have access to the necessary technology (such as smartphones or computers) and are comfortable using the website or app. In regions with limited technological infrastructure, this could be a limitation.
* **Algorithmic Complexity:** While Naive Bayes is a relatively simple algorithm, it may not capture all the nuances of complex interactions affecting crop growth. More sophisticated algorithms might be needed for a more comprehensive prediction model.

1. **Methodology for Optimal Crop Selection to Maximize Yield Rates Through Machine Learning Techniques.**

**Journal Details:** Published in 2015 by [M.P. Singh](https://ieeexplore.ieee.org/author/37086303945), [Prabhat Kumar](https://ieeexplore.ieee.org/author/38245554800), [J.P. Singh](https://ieeexplore.ieee.org/author/38187001400)

**Abstract:** An agro-based nation's economic growth and food security are significantly influenced by its agricultural planning. The choice of crop or crops is a crucial factor in agricultural planning. It is influenced by a number of factors, including market pricing, government policy, and production rate. Using statistical approaches or machine learning techniques, numerous studies investigated crop yield rate prediction, weather prediction, soil classification, and crop classification for agricultural planning. Crop selection becomes a conundrum if there are multiple ways to plant a crop at once utilising a limited amount of land. In order to address the crop selection problem, maximise crop net yield rate throughout the growing season, and ultimately attain the highest possible level of national economic growth, this study presented the Crop Selection Method (CSM).

**Advantages:**

* **Optimized Yield Rate:** CSM aims to maximize crop yield over the season, ensuring improved agricultural productivity for enhanced food security and economic growth.
* **Consideration of Multiple Parameters:** The method's holistic approach incorporates various factors like production rate, market prices, and government policies, facilitating well-informed decision-making in crop selection.**Integration of Machine Learning:** CSM utilizes machine learning techniques, offering potential for more accurate predictions and insights compared to traditional statistical methods, thus enhancing precision and overall efficiency.
* **Adaptability to Dynamic Conditions:** CSM considers dynamic factors such as soil classification, and crop types, making it adaptable to changing environmental conditions, crucial for sustainable agriculture planning.
* **Economic Growth:** Maximizing net yield directly contributes to economic growth by increasing agricultural output, positively impacting the nation's economy.

**Limitations:**

* **Data Dependency:** CSM effectiveness heavily relies on the quality and availability of data, and inaccurate or insufficient data may lead to suboptimal results.
* **Complexity:** Implementation may require advanced technical expertise, posing a potential adoption barrier in regions with limited technological resources.
* **Sensitivity to Model Accuracy:** Inaccurate machine learning models may lead to flawed crop selection decisions, impacting the overall success of the method.
* **Ethical Considerations:** The method should consider ethical aspects, including the potential impact on small-scale farmers, biodiversity, and environmental sustainability to avoid unintended negative consequences.

1. **Analysis and Prediction of Soil Nutrients pH, N, P, K for Crop Using Machine Learning Classifier**

**Journal Details:** Published in 2020 by Disha S. Wankhede

**Abstract:** This paper explores the crucial role of agriculture in the development of regions

such as India, emphasizing the potential of computer engineering in aiding farmers' decision-

making processes for improved crop yield and soil quality. The focus is on the application

of machine learning classifiers to address specific challenges related to soil nutrients and

plant diseases. Various classifiers, including Naive Bayes, Random Forest, and SVM, are

surveyed for their effectiveness in predicting crop outcomes based on soil conditions. The

study primarily concentrates on essential nutrients like sulfur, iron, boron, zinc, copper, pH,

nitrogen, phosphorus, magnesium, potassium, with particular emphasis on pH, N, P, and K.

The findings suggest that the Naive Bayes classifier performs well in comparison to other

machine learning classifiers, especially when dealing with large datasets.

**Advantages:**

* **Personalization:** Machine learning models can be tailored to specific regions, crops, and soil types. This allows for personalized recommendations and predictions based on the unique characteristics of each farming area.
* **Cost-Effectiveness:** While initial implementation may require an investment, the long-term benefits can outweigh the costs. Improved resource allocation, reduced wastage, and increased yields contribute to the cost-effectiveness of using machine learning in agriculture.
* **Data-Driven Decision Making:** Machine learning empowers farmers with data-driven insights. This enables informed decision-making regarding crop selection, fertilizer application, and other farming practices, leading to optimized outcomes.

**Limitations:**

* **Overfitting:** Overfitting occurs when a model is too complex and learns noise in the data rather than the underlying patterns. This can lead to poor generalization to new, unseen data.
* **Interpretability:** Some machine learning models, especially complex ones like Random Forests or Support Vector Machines, can be difficult to interpret. Understanding the decision-making process of these models may pose challenges for farmers.
* **Cost of Implementation:** Initial investment expenses, encompassing hardware, software, and training, may pose a challenge for small-scale farmers or those operating in regions with limited resources.
* **Security Concerns:** As data plays a crucial role in machine learning, there are concerns about the security and privacy of agricultural data. Protecting sensitive information from unauthorized access is essential.

1. **Soil Based Prediction for Crop Yield using Predictive Analytics**

**Journal Details:** Published in 2021 by M. Chandraprabha, Rajesh Kumar Dhanaraj **Abstract:** The primary element and key player in agriculture is soil. Crop yielding is calculated using the soil's pH and nutrients. Farmers continue to analyse the quality of their soil using a conventional method. Emerging technologies being researched to advance the agricultural field include data mining, artificial intelligence, machine learning, deep learning, and predictive analytics. Predictive analysis is a machine learning technique that makes predictions about the future using data from the past or historical periods. Predictive analytics is used in agriculture to forecast or identify the amount of soil nutrients needed for crops like paddy, ragi, cumbu, etc. The soil-based dataset used in this study was gathered from the TNAU website and includes 32 districts of Tamilia Using district-wise total production and sown area, algorithms like Naïve Bayes, Bayes Net, and IbK have been used to determine which crop variety is best for the soil. Its accuracy levels are also contrasted. True positive value, false positive value, precision, recall, f-measure, and MCC are used to calculate accuracy.

**Advantages:**

* **Increased accuracy and efficiency:** Compared to traditional soil analysis methods, which can be time-consuming and expensive, predictive models can quickly and accurately analyze large datasets of soil data to identify nutrient levels and recommend suitable crops. This can lead to improved resource allocation and better decision-making for farmers.
* **Improved crop yields:** By understanding the specific needs of the soil and selecting the most suitable crops, farmers can potentially increase their yield and profitability. Predictive models can also help to optimize fertilization and irrigation practices, reducing waste and optimizing resource use.
* **Adaptability to changing conditions:** Predictive models can incorporate historical and real-time data on weather, pests, and diseases to provide dynamic recommendations that adapt to changing conditions. This can help farmers to mitigate risks and make better decisions in the face of challenges.
* **Sustainability:** By promoting efficient resource use and reducing waste, predictive analytics can contribute to more sustainable agricultural practices. This can benefit both the environment and the long-term profitability of farms.

**Limitations:**

* **Data quality:** The accuracy of the predictions is highly dependent on the quality and quantity of data used to train the model. Poor quality data can lead to inaccurate recommendations and negative consequences for farmers.
* **Model complexity:** Implementing and maintaining complex predictive models can be challenging for small-scale farmers who may lack the necessary resources or expertise. Additionally, overfitting the model to specific data can lead to poor performance when applied to different conditions.
* **Access to technology:** Many farmers, particularly in developing countries, may not have access to the technology or infrastructure required to utilize predictive analytics tools effectively. Closing the digital divide is crucial for ensuring equitable access to these benefits.
* **Limited scope:** Current models primarily focus on soil nutrients and basic crop selection. Further research is needed to incorporate a wider range of factors, such as pest and disease risk, market demand, and economic considerations, to provide more comprehensive recommendations for farmers.

1. **Temperature and Crop Development**

**Journal Details:** Published in 1991 by J. T. Ritchie, D. S. Nesmith

**Abstract:** This chapter delves into the significance of utilizing temperature as a predictive factor for both growth and developmental processes in plants. It examines the value of employing a temperature summation system for predicting plant development while highlighting potential sources of uncertainties in such systems. The historical context of temperature records, primarily derived from liquid-in-glass thermometers in standard white, louvered instrument shelters, is explored.The focus of thermal time-based development prediction has been on crucial events in the phasic development of determinant crops, such as the timing of anthesis and physiological maturity. Germination is frequently employed as a metric for evaluating temperature response functions in plant development. The abstract also touches upon the influence of photoperiod on the prediction of crop development and the intricate relationships between temperature and plant growth. Additionally, the chapter delves into the impact of vernalization on leaf numbers, emphasizing its role akin to photoperiod in influencing plant developmental patterns.

**Advantages:**

* **Simplicity and accessibility:** Measuring temperature is relatively straightforward and inexpensive compared to other plant development indicators. This makes it accessible to a wider range of farmers and researchers.
* **Quantifiable data:** Temperature can be easily monitored and recorded, providing numerical data suitable for modeling and analysis. This allows for precise predictions and comparisons.
* **Strong correlation with specific phases:** Temperature has a strong and well-established link to certain key developmental stages in plants, particularly in determinant crops. This makes it a reliable predictor for events like flowering and maturity.
* **Thermal time summation:** The concept of thermal time summation provides a convenient way to accumulate temperature units and track progress towards critical developmental stages.

**Limitations:**

* **Incomplete picture:** Temperature only reflects one aspect of the environment. Other factors like photoperiod, light intensity, rainfall, and nutrient availability also play significant roles in plant development and can lead to inaccuracies in predictions solely based on temperature.
* **Variability between species and cultivars:** Different plant species and cultivars have varying temperature requirements and responses. A single model based on temperature may not be accurate for all types of plants.
* **Microclimatic variations:** Standard temperature records taken in instrument shelters may not accurately represent the microclimates experienced by individual plants, leading to discrepancies in predictions.
* **Complex interactions:** Interactions between temperature and other environmental factors can be complex and difficult to model accurately. Oversimplifying these interactions can lead to unreliable predictions.

**CHAPTER-3**

**RESEARCH GAPS OF EXISTING METHODS**

**3.1. Existing Methods**

In smart farming, various advanced methods are employed to revolutionize traditional agricultural practices. Remote sensing technology, utilizing satellites and drones, plays a pivotal role in collecting crucial data on crops, soil conditions, and weather patterns. On-field sensor networks provide real-time information about soil moisture, temperature, and humidity, aiding farmers in making informed decisions about irrigation and fertilization. Internet of Things (IoT) devices connect different elements on the farm, facilitating seamless data exchange and monitoring. Machine learning algorithms analyze extensive datasets, offering predictive analytics for crop selection, disease prediction, and optimal harvesting times. Autonomous farming machinery guided by AI algorithms performs tasks such as planting and harvesting, reducing dependence on manual labor.

**3.2. LIMITATION OF EXISTING SYSTEMS**

• Simplistic Model The machine learning model used for crop prediction might be simplistic.

• Input validation the code assumes that the user inputs are valid and within the expected range.

• Limited crop dataset the crop dictionary used for mapping predictions may not cover all possible crops.

**3.3. OBJECTIVES**

* + **Crop prediction**: Develop a system that accurately predicts the most suitable crop for cultivation based on input parameters such as Nitrogen, Phosphorus, Potassium, Temperature, Humidity, and Soil ph.
  + **User-friendly interface:** Create an intuitive and user-friendly web interface that allows users, especially farmers, to easily input relevant data for crop prediction. The interface should be accessible and straightforward to use.
  + **Scalability:** The system should be able to manage multiple requests concurrently without compromising performance.

**CHAPTER-4**

**PROPOSED MOTHODOLOGY**

Our project, "Smart Farming", to enhance the functionality and robustness of the provided smart farming prediction application, several improvements can be proposed. To make the smart farming prediction better and safer, we can do a few things. First, we need to ensure that the code is written in a way that protects it from common online threats. This involves checking and cleaning up the information that users input to avoid potential security issues. We also need to handle errors more effectively to provide users with helpful messages when something unexpected happens. Next, we should organize the software's building blocks better and manage the tools it depends on more efficiently. This makes it easier to recreate the application exactly as it is, and it ensures that it can handle more users without breaking.

For the long term, we should set up a way to keep the application's brain (the machine learning model) up to date. This is important because as time goes on, the model might need improvements to stay accurate.

The part of the application that users interact with (the user interface) can be made friendlier by adding clear instructions and helpful hints. We can also make the application look good and work well on different devices, like phones and tablets. Additionally, we should keep an eye on how the application is doing by creating records of its activities. This helps us understand what's happening and fix any problems that might come up. By doing these things, we can make the smart farming application more secure, scalable, easy to use, and ready for the future.

**4.1. ADVANTAGES OF PROPOSED METHOD**

1. **Enhanced Security**: By ensuring the code is resilient to common online threats and carefully handling user input, the application becomes more resistant to potential security risks, protecting user data and the system.

2. **Error Resilience:** Implementing effective error handling mechanisms ensures that users receive informative messages when unexpected issues occur, providing a smoother and more reliable experience.

**3. Better Code Organization:** Organizing the application's building blocks and managing dependencies more efficiently not only simplifies the development process but also makes it easier to maintain and scale the application.

**4.** **Scalability**: Efficiently managing tools and resources allows the application to handle increased user loads without compromising performance, making it more scalable and adaptable to growing user bases.

**5.** **Model Maintenance**: Establishing a mechanism to update the machine learning model ensures that the application remains accurate over time. Regular updates can incorporate improvements and adjustments based on evolving data patterns.

**6. Performance Monitoring**: Keeping records of the application's activities enables the development team to monitor its performance, identify trends, and address any issues promptly, ensuring a stable and reliable service.

**7. Future-Ready**: By implementing these improvements, the application becomes more adaptable and ready for future advancements, ensuring its longevity and relevance in the ever-changing landscape of technology.

**4.2. DISADVANTAGES OF PROPOSED METHODS**

**1. Limited Input Validation:** The application lacks comprehensive input validation, which might result in unexpected behavior if users provide invalid or malicious input. Proper input validation is crucial to enhance security.

**2. Static Crop Image URLs:** The mapping of crop names to image URLs is static and may not be easily extensible. Adding new crops or modifying existing mappings would require changes in the code, making it less scalable.

**3. Limited Model Explanation:** The application doesn't provide explanations or insights into why a particular crop is recommended. Adding model explanations or insights could enhance user trust and understanding.

**4. Potential Model Staleness:** If the underlying machine learning model is not periodically updated, it might become stale and less accurate over time as agricultural conditions change. Implementing a strategy for model updates is essential for maintaining prediction accuracy.

**5. Single-Threaded Server:** The development server provided by Flask (`app.run(debug=True)`) is single-threaded and not suitable for production use. In a production environment, a more robust server (e.g., Gunicorn) should be used to handle concurrent requests efficiently.

**6. Lack of Unit Tests:** The application lacks unit tests to verify the correctness of its components. Implementing a testing strategy would help identify and fix issues more reliably.

**CHAPTER-5**

**HARDWARE AND SOFTWARE REQUIREMENTS**

#### **5.1. HARDWARE REQUIREMENTS**

* PROCESSOR : INTEL 13
* RAM : 8 GB DD RAM
* HARD DISK : 250 GB
* KEYBOARD
* MOUSE
* PC (PERSONAL COMPUTERS)

#### **5.2. SOFTWARE REQUIREMENTS**

* OPERATING SYSTEM : WINDOWS 10 AND ABOVE
* IDE : VISUAL STUDIO CODE,GOOGLE
* FRONT-END : HTML,CSS,JS
* SERVER-SIDE PROGRAMMING : PYTHON
* OTHER LIBRARIES : PANDAS,FLASK,SCIKIT-LEARN ,

NUMPY.

**CHAPTER-6**

**SYSTEM DESIGN & IMPLEMENTATION**

**6.1. USE CASE DIAGRAM**

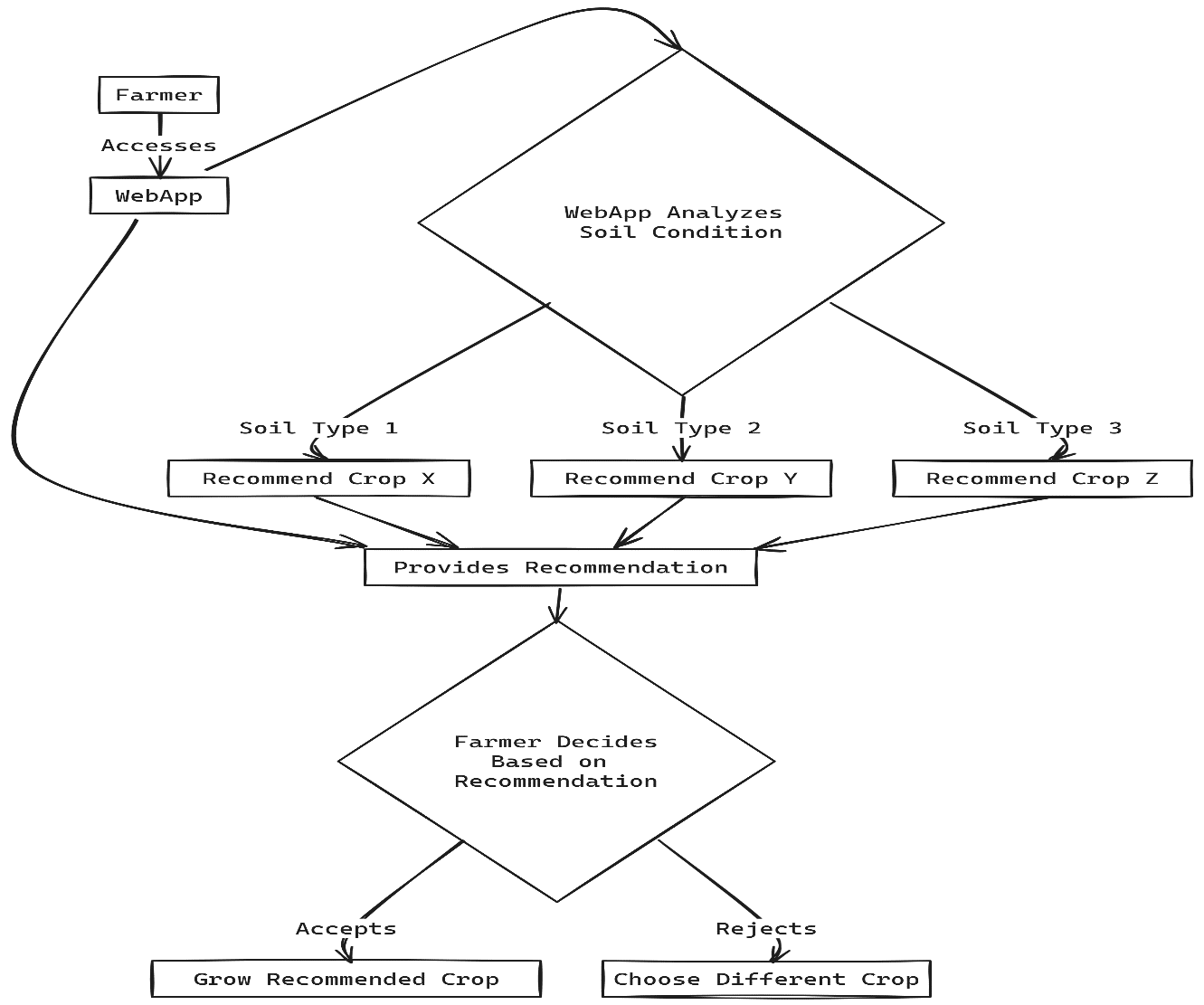
****

Fig 1.1: Use case diagram

The web application is designed to guide farmers through a data-driven decision-making process for selecting the most suitable crops to cultivate based on soil conditions. This process is depicted as a straight forward flowchart that delineates the interaction between the farmer and the web application.When a farmer accesses the web application, they enter a user-friendly interface that has been developed to simplify the complexity of agronomic data analysis. The primary function of the application is to analyze the soil conditions, which is a critical factor in determining crop suitability. Soil condition analysis might involve assessing various parameters such as soil type, pH level, nutrient content, moisture level, and more. Upon the completion of soil analysis, the application processes this data through its built-in algorithms, which likely incorporate machine learning models trained on agronomic data. These models can compare the farmer's soil conditions with a dataset that includes optimal growing conditions for a wide variety of crops. Depending on the soil analysis results, the web application suggests the best crop choices tailored to the specific soil type identified. For instance, if the soil is identified as Soil Type 1, the application might recommend Crop X, which is known to thrive in such soil conditions. Similarly, for Soil Type 2 and Soil Type 3, the application would recommend Crop Y and Crop Z, respectively.

Once the application provides its recommendations, the farmer is faced with a decision point. They must decide whether to accept the recommendation or to reject it in favor of a different crop that they might prefer for economic, cultural, or personal reasons. The application's recommendation is based purely on agronomic suitability, but the farmer's decision might also consider market prices, personal experience, or the demand for certain crops . If the farmer accepts the recommendation, they proceed to grow the recommended crop, leveraging the analytical insights provided by the application to potentially increase yield, reduce risk, and improve overall farming efficiency. On the other hand, if the farmer rejects the recommendation, they may choose a different crop based on other factors. In this scenario, the application could still be useful by providing the farmer with information on how to best manage the chosen crop under the given soil conditions. This process encapsulates a data-driven approach to agriculture, wherein farmers are empowered with scientific analysis and machine learning predictions to make informed decisions about their crops. Such an approach can lead to more sustainable farming practices, as crops are chosen based on their compatibility with the environment rather than trial and error or guesswork.

The web application serves as an advisory tool, providing recommendations rather than mandates. Farmers retain the ultimate decision-making power, reflecting the reality that while data can provide powerful insights, the practicalities of farming often require human judgment and expertise. The use of such a web application signifies a shift toward precision agriculture, where technology and data are harnessed to optimize farming practices, thereby enhancing productivity and sustainability in the agricultural sector. Hence the fig1.1 shows that the given interface is user friendly.

**6.2. STATE DIAGRAM**

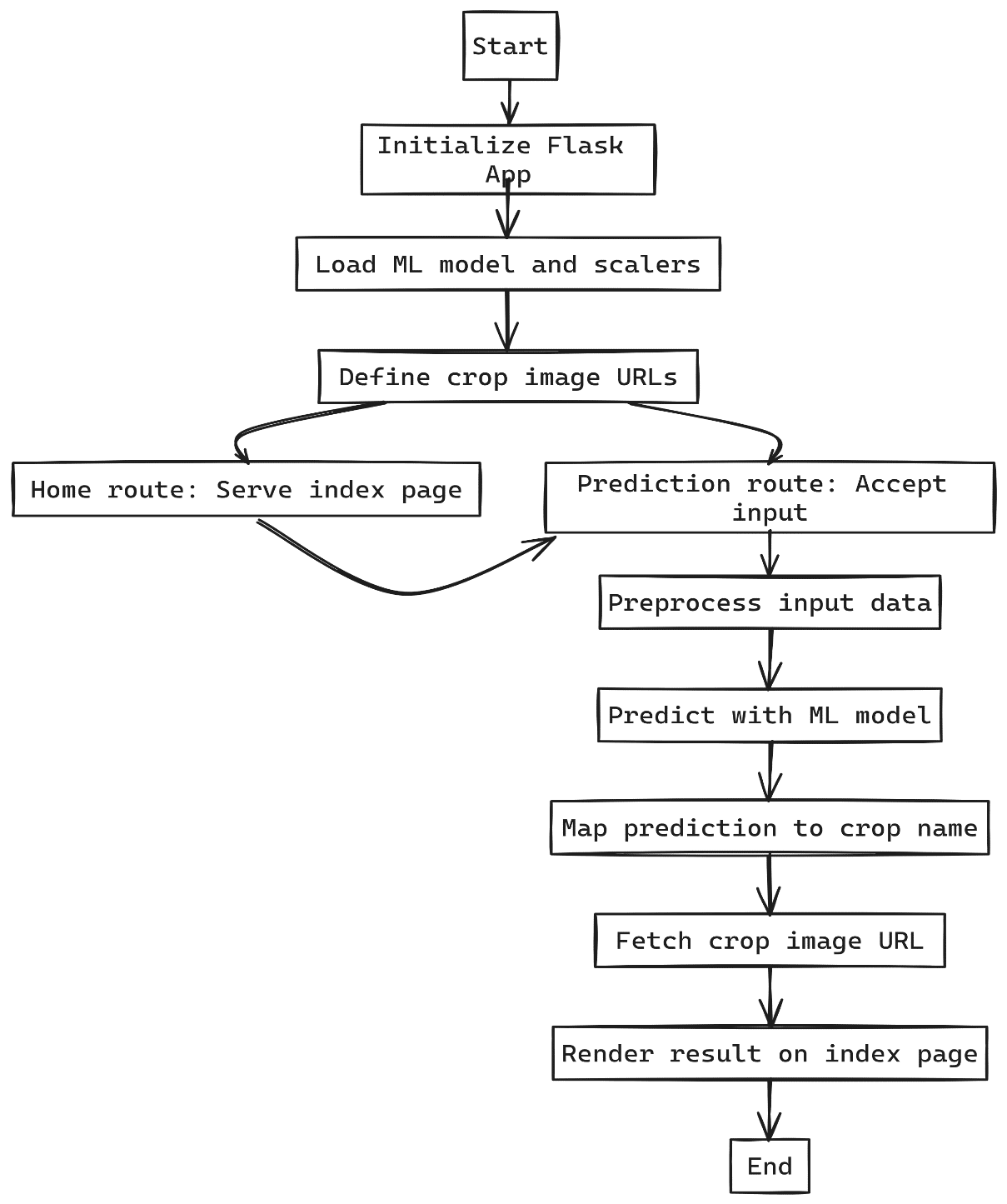
****

Fig 1.2: State diagram

The flowchart provided illustrates the operational framework of a Flask-based web application the fig1.2 is designed to offer agricultural crop recommendations to farmers. This application employs a machine learning (ML) model to analyze soil data and suggest the most suitable crop, enhancing the decision-making process for agricultural activities.

* **Initialization: The application begins with the initiation of the Flask app,** lightweight yet powerful web application framework in Python that is widely used for building web applications quickly and with minimal setup. Flask's simplicity and flexibility make it an ideal choice for small-scale web applications, such as this agricultural recommendation system.
* **Machine Learning Model and Scalers**: The next step involves loading the pre-trained ML model and associated scalers. The ML model is a predictive algorithm that has been trained on historical agricultural data to identify patterns and correlations between soil characteristics and crop viability. Scalers are used to normalize the input data to match the scale of the data used during the model's training phase. This ensures that the model's predictions are as accurate as possible.
* **Crop Image URLs**: The system defines a dictionary mapping various crop names to their corresponding image URLs. This mapping is essential for the UI to display visual information about the recommended crop, which can help farmers make more informed decisions.
* **Home Route**: The home route is responsible for serving the index page to the user. This page acts as the main interface for the farmer to input their soil data, such as nitrogen, phosphorus, potassium levels, temperature, humidity, and pH.
* **Prediction Route**: When the farmer submits their data, the prediction route is triggered. This route handles the incoming POST request containing the soil data.
* **Preprocess Input Data:** The data preprocessing phase involves applying Min-Max Scaling and Standard Scaling to normalize the data, preparing it for analysis. This step is crucial as it adjusts the input data to the scale the ML model expects, based on its training.
* **Predict with ML Model:** The preprocessed data is fed into the ML model, which then predicts the most suitable crop for the given soil conditions. The prediction is made based on the model's learned patterns from the training dataset.
* **Map Prediction to Crop Name:** Once a prediction is made, the application maps the numerical or categorical output of the ML model to an actualcrop name using a predefined mapping.
* **Fetch Crop Image URL:** With the crop name determined, the application fetches the corresponding image URL from the dictionary defined earlier.
* **Render Result on Index Page:** Finally, the application renders the recommendation result back on the index page. The result includes the name of the suggested crop and its image, providing a visually informative response to the farmer.

In essence, the application encapsulates the entire flow from data input by the farmer to a data-driven recommendation output. It is an exemplar of how modern web technologies and machine learning can be harnessed to provide practical solutions in traditional fields such as agriculture. The integration of scalable web frameworks like Flask with advanced data processing and ML modeling techniques represents a significant step towards agricultural technology (AgTech), aimed at optimizing crop yields, reducing risk, and increasing the efficiency of farming practices. This web application is not just a tool for crop recommendation; it signifies a move towards a more data-informed and technology-driven approach in agriculture, marrying traditional farming knowledge with modern computational power and analytics.

**6.3. SYSTEM ARCHITECTURE DIAGRAM**

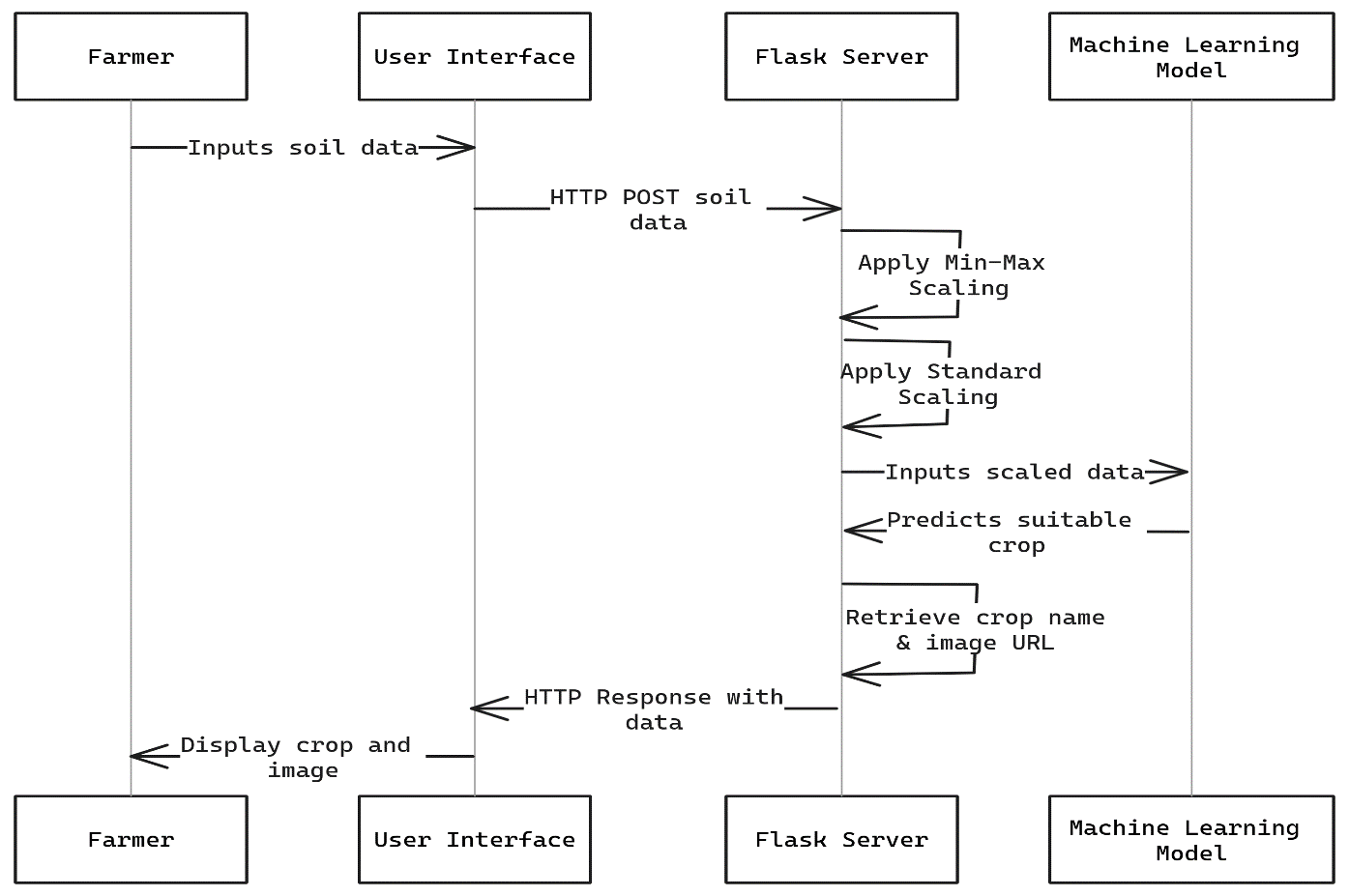
****

Fig 1.3:System Archietecture diagram

It is a web-based agricultural recommendation application designed to assist farmers in making informed decisions about crop selection based on soil data. Here's a detailed examination of the system, broken down into its fundamental components and processes using fig1.3 :

* **Farmer/User Interaction:** The cycle begins with the farmer interfacing with the application through a user interface (UI). This UI is crucial as it must be intuitive enough for farmers with varying levels of technical expertise to input their soil data. The data typically includes soil composition (Nitrogen, Phosphorus, Potassium), temperature, humidity, and pH levels.
* **Data Transmission**: Once the farmer inputs the soil data, it is transmitted to the server using an HTTP POST request. This is a standard method for sending user-generated data to a web server where it can be processed. The application's front-end ensures that the data is properly formatted and validated before sending it to the server.
* **Flask Server Processing:** Upon receiving the soil data, the Flask server performs a series of pre-processing steps. Flask, a lightweight and versatile web server gateway interface (WSGI) web application framework, is particularly well-suited for small to medium web applications, like this agricultural recommendation system.The Flask server leverages its routing capabilities to direct the HTTP POST request to the appropriate function for processing the soil data. The server's functions will apply Min-Max Scaling and Standard Scaling to the data. These scaling methods are crucial for normalizing the data, ensuring that the machine learning model receives it in the correct format and scale, which matches the data used to train the model.
* **Machine Learning Model Prediction:** The scaled data is then fed into a pre-trained machine learning model. This model is the heart of the recommendation system and has likely been trained on a substantial dataset consisting of various soil conditions and the corresponding best-suited crops. It uses the input data to predict the most appropriate crop for the given soil conditions.
* **Output Generation:** Once the machine learning model has made its prediction, the server retrieves the crop name and its associated image URL from a pre-defined set of mappings. This retrieval is instantaneous and is facilitated by the server's access to an internal or external database, or a simple in-memory data structure that holds the mapping between crop names and their image URLs.
* **Response to User Interface:** With the prediction and image URL at hand, the Flask server packages this information into an HTTP response and sends it back to the user interface. This response must be structured in such a way that the UI can interpret and display the information effectively to the farmer.
* **Farmer's Reception of Information:** The farmer receives the crop recommendation and the corresponding image on their UI. The UI's design must ensure that the information is presented in a clear, understandable format. This could include not just the name and image of the crop but also additional information such as the reasons why the crop is recommended, the confidence level of the prediction, and potentially other relevant agronomic data. This system architecture emphasizes a seamless flow of data, from the initial user input to the final delivery of the machine learning model's prediction. Each component plays a critical role in the overall functionality of the application. The user interface must be user-friendly to cater to the needs of diverse users. The HTTP request/response protocol serves as the communication backbone, ensuring the reliable exchange of information between client and server. The Flask server acts as the intermediary, processing data and interfacing with the machine learning model. The machine learning model is the decision-maker, using data to generate valuable insights. The response mechanism is the concluding step that delivers the value back to the user. Overall, the architecture is designed with modularity and scalability in mind, ensuring that each part can be improved or replaced independently as the system evolves. For example, the machine learning model can be updated or retrained without altering the UI or the server infrastructure. The Flask framework supports such scalability, allowing developers to add more complex functionalities or to integrate with other systems, such as weather forecasting services or market price databases, to enrich the recommendations provided to the farmers. The ultimate goal of this system is to enhance the agricultural decision-making process, providing farmers with data-driven insights that were previously inaccessible to them. By doing so, it aims to improve crop yields, reduce the risk of crop failure, and increase the overall efficiency of farming operations.

**6.4. ABOUT PYTHON AND MACHINE LEARNING**

1. **Python:**

Python is a high-level, dynamically semantic, object-oriented, interpreted programming language. Because of its high-level built-in data structures, dynamic typing, and dynamic binding, it is especially useful for Rapid Application Development as well as for use as a scripting or glue language to join existing components. Because of its concise syntax, Python is easier to learn and emphasises readability, which reduces programme maintenance costs. The modularity and reuse of code in programmes are encouraged by Python's support for modules and packages. The Python interpreter and the full standard library are available in both source code and binary formats. There is no compilation step, therefore the edit-test-debug cycle is incredibly fast. Since a segmentation failure in a Python programme is never the result of a bug or inaccurate entry Rather, the interpreter raises an exception if it finds a mistake. In case the programme fails to catch the exception, the interpreter generates a stack trace. A source level debugger can be used to set breakpoints, evaluate arbitrary expressions, analyse local and global variables, walk through the code one line at a time, and perform other functions. The debugger, which is written in Python, demonstrates the language's capacity for introspection. However, because of the fast edit-test-debug cycle, adding a few print statements to the source code is often the quickset approach to debug a programme. It can be distributed for free via all well-known platforms.

1. **Machine Learning:**

An area of artificial intelligence(AI) called machine learning is devoted to creating software programs that, without being explicitly coded, learn from data over time and become more accurate.

An algorithm is a series of statistical processing processes in data science. In machine learning, algorithms are ‘trained’ to sift through vast volumes of data in search of patterns and characteristics in order to derive conclusion and forecasts based on fresh data. As it analyses more data, a smarter algorithm will provide conclusions and predictions that are more accurate.

Machine learning examples are all round us now. When we speak to digital assistants, they respond by playing music and doing web searches. websites provide recommendations for goods, films, and music based on what we’ve previously purchased, viewed, or listened to. Our floors are vacuumed by robots while we spend our time doing greater things. Unwanted emails are prevented from entering our inboxes by spam detectors. Systems for analyzing medical images assist clinicians in finding tumors they have missed. And the first autonomous vehicles are now travelling on public roads.

We may anticipate more. Machine learning will increase productivity in both our personal and professional lives as big data continues to grow, computers get more powerful and accessible, and data scientists continue to create more effective algorithms.

The working of machine learning

Building a machine learning application (or model) involves four simple stages. Data Scientists who closely collaborate with the business executives for whom the model is being produced often carry out the tasks.

**Step 1:** Select and prepare a training data set

A data set used for training is a sample of the data that will be ingested by the machine learning model to address the issue at hand. In certain circumstances, the training data is “tagged” to draw attention to the traits and categories the model will need to recognize. Since some of the data is unlabeled, the model will have to extract those characteristics and make its own classifications.

**Step 2:** Choose an algorithm to run on the training data set

A series of statistical processing stages make up an algorithm. The kind of algorithm used relies on the quantity and kind of data (labeled or unlabeled) in the training data set, as well as the task at hand.

The following are examples of machine learning algorithms frequently used with labeled data:

* **Regression algorithms**
* **Decision trees**
* **Instance-based algorithms**

Algorithms for use with unlabeled data include the following:

* **Clustering algorithms**
* **Association algorithms**
* **Neural Networks**

Step 3: Training the algorithm to create the model

Iteratively running variables through the algorithm, comparing the output with the expected results, adjusting weights and biases within the algorithm that might produce a more accurate result, and then running the variables through the algorithm once more until the algorithm consistently returns the right answer.

Step 4: Using and improving the model

The model must then be used with fresh data, and ideally it will do so over time while also becoming more accurate and efficient. The problem that has to be solved will determine where the fresh data originates from. A machine learning model used to operate a robot hoover. Cleaner, for instance, may ingest data from real-world interactions with moving furniture or new objects in the room, but a machine learning model used to identify spam will absorb messages.

**6.5. ALGORITHM DETAILS**

Random Forest is an ensemble learning algorithm used for both classification and regression tasks. It belongs to the family of tree-based models and is known for its robustness and high performance. Thealgorithm builds multiple decision trees during training andmergestheir predictions to obtain a more accurate and stable result.

**1. Decision Trees:** Random Forest is built upon the foundation of decision trees. A decision tree is a flowchart-like structure where each internal node represents a decision based on a feature, each branch represents the outcome of the decision, and each leaf node represents the final prediction. Decision trees are prone to overfitting, capturing noise in the training data and resulting in poor generalization to new, unseen data.

**2. Ensemble Learning:** The key idea behind Random Forest is to build multiple decision trees and combine their predictions. Each tree is trained on a random subset of the training data (bootstrap sampling), and at each node, a random subset of features is considered for splitting. This randomness introduces diversity among the trees, reducing overfitting and increasing the model's generalization performance.

**3. Bootstrapping:** Random Forest employs a technique called bootstrapping, where random samples are drawn with replacement from the training dataset to create multiple subsets. Each tree is trained on one of these subsets, introducing variability in the training process.

**4. Random Feature Selection:** At each node of the decision tree, only a random subset of features is considered for making the split. This further adds diversity to the trees, preventing them from relying too heavily on a single feature.

**5. Voting or Averaging:** For classification tasks, the final prediction of the Random Forest is determined by a majority vote among the individual trees. For regression tasks, the predictions are averaged. This ensemble approach helps improve the overall accuracy and robustness of the model.

**6. Out-of-Bag Error**: Since each tree is trained on a subset of the data, some data points are not used in the training of certain trees. These "out-of-bag" samples can be used to estimate the model's performance without the need for a separate validation set.

**7. Robustness and Generalization**: Random Forest is known for its robustness against overfitting, noise, and outliers in the data. It often box" and requiresminimal hyperparameter tuning compared to individual decision trees.

Random Forests are widely used in various applications, including classification problems such as image recognition, medical diagnosis, and financial fraud detection. Their ability to handle complex datasets and provide reliable predictions makes them a popular choice in machine learning.

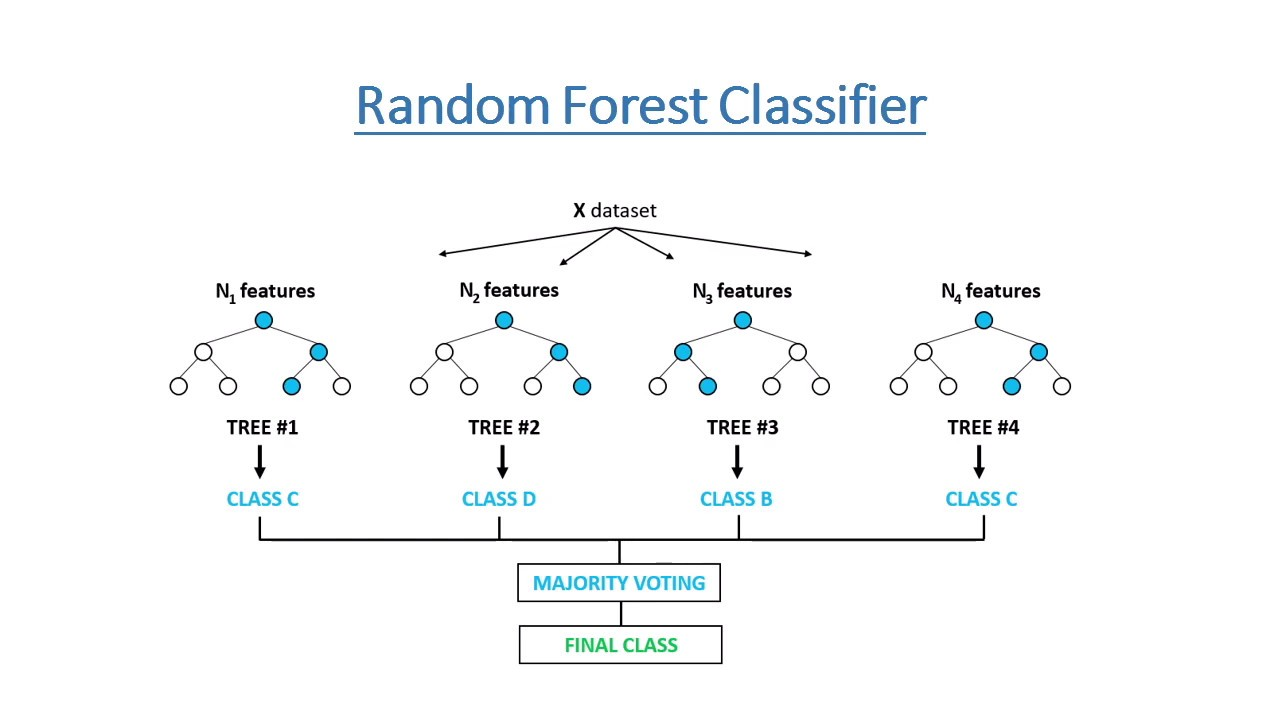
****

Fig 2.1: Random forest classifier

SAS, R, and Python are just a few of the programmes in which we may perform random forest regressions. Every tree in a random forest regression generates a distinct prediction. The result of the regression is the average prediction made by each individual tree. In contrast, the output of a random forest classification is based on the decision trees' class mode. While both linear regression and random forest regression use the same general idea, their functionalities are different. When y is the dependent variable, x is the independent variable, b is the estimation parameter, and c is a constant, the formula for linear regression is y=bx + c. A complicated random forest regression works similarly to a blackbox.

**6.6. SOURCE CODE DETAILS**

* **Flask library**

Python is used to create the Flask web application framework. Flask was developed by Armin Ronacher, who was a part of the Pocco (Python Community) group. The foundation for Flask is built upon the Werkzeug WSGI toolkit and the Jinja2 template engine, both of which are projects under Pocco.

* **Sklearn Library**

Scikit-Learn, often referred to as Sklearn, stands out as a robust and dependable Python machine-learning package renowned for its consistent interface and an array of powerful techniques for statistical modeling and machine learning. Developed primarily in Python and built upon the foundations of NumPy, SciPy, and Matplotlib, this open-source library focuses on modeling data rather than mere data manipulation, offering a versatile suite of features.

**Linear Regression:** A supervised learning algorithm employed to model the correlation between a target variable and one or more predictor variables..

**Decision Tree:** A tree-like model that makes decisions based on features, creating a hierarchical structure for classification or regression.

**Principal Component Analysis (PCA):** Dimensionality reduction method that reduces the number of features while preserving essential information.

**Ensemble Methods (e.g., Random Forest):** Combining predictions from multiple models to improve overall performance.

**Feature Extraction:** Extracting relevant information from data, particularly useful in image and text data, to identify significant patterns.

**Feature Selection:** Choosing the most relevant features in a dataset to enhance model efficiency and accuracy.

**Open Source:** Scikit-Learn is a BSD-licensed open-source library, providing a free and accessible resource for machine learning applications in both research andbusiness settings.

* **Pandas library :** Python's Pandas library serves as a powerful tool for handling and manipulating data collections, offering a comprehensive suite of functionalities for exploration, cleaning, analysis, and manipulation. The brainchild of Wes McKinney in 2008, the name "Pandas" is a portmanteau of "Panel Data" and "Python Data Analysis," reflecting its utility in dealing with structured data.

Facilitating the examination of extensive datasets, Pandas plays a pivotal role in drawing conclusions based on statistical principles. It excels in organizing disorderly datasets, transforming them into coherent and valuable information. In the realm of data science, where the relevance of data is paramount, Pandas provides critical insights. It facilitates the exploration of correlations between different columns, computes average values, identifies maximum and minimum values, and extracts information on data distribution.

Pandas, being a versatile data manipulation tool, is also adept at cleaning datasets. It systematically handles irrelevant or erroneous data, including empty or NULL values, in a process commonly known as data cleaning. This ensures the integrity and accuracy of the data, a crucial step in any data analysis pipeline. By eliminating inconsistencies and inaccuracies, Pandas enhances the reliability of the conclusions drawn from the data.

In essence, Pandas empowers data scientists and analysts to navigate through vast datasets, unravel hidden patterns, and make informed decisions. Its multifaceted capabilities extend beyond data organization to encompass statistical exploration and data cleaning, making it an indispensable asset in the toolkit of anyone engaged in data-driven endeavors.

* **Numpy libraries :** A robust Python package for numerical computation is called NumPy. It provides support for huge, multi-dimensional arrays and matrices, together with a suite of mathematical functions to operate on these arrays. The core Python library for scientific computing, NumPy is widely utilised in many fields,

including signal processing, machine learning, and data analysis.

* NumPy, a fundamental library for numerical computing in Python, introduces key features that empower efficient handling of large-scale numerical data. At its core is the powerful numpy.ndarray, facilitating the representation of multi-dimensional arrays and matrices. With an extensive array of mathematical functions, NumPy supports operations such as basic arithmetic, trigonometry, and exponentiation, all applied element-wise on arrays. A noteworthy feature is broadcasting, enabling seamless operations between arrays of varying shapes. The library boasts a robust random number generation module, vital for simulations and data generation. Additionally, NumPy provides comprehensive support for linear algebra, encompassing operations like matrix multiplication, eigenvalue computation, and singular value decomposition. The library's indexing and slicing capabilities facilitate easy extraction and manipulation of specific elements or subarrays within an array. Integrating seamlessly with other scientific Python libraries, NumPy serves as a foundational tool in diverse fields, including machine learning, data analysis, and signal processing. Its efficiency and versatility make it indispensable for researchers, scientists, and developers engaged in numerical computations and data manipulation.

**CHAPTER-7**

**IMPLEMENTATION FOR EXECUTION OF PROJECT**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| WEEKS | | | | | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

**(GANTT CHART)**

Fig 2.2: Gantt chart

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TASK ID** | **TASK NAME** | **START DATE** | **END DATE** | **DURATION in days** |
| 1 | Research and Analysis | 01/10/2023 | 24/10/2023 | 24 |
| 2 | Requirements Collection | 25/10/2023 | 29/10/2023 | 5 |
| 3 | Design | 30/10/2023 | 11/11/2023 | 13 |
| 4 | Review | 11/11/2023 | 13/11/2023 | 3 |
| 5 | Development | 28/10/2023 | 23/12/2023 | 57 |
| 6 | Testing | 14/12/2023 | 26/12/2023 | 13 |
| 7 | Implementation | 25/12/2023 | 30/12/2023 | 6 |

Table 1.1: Gantt chart table

**CHAPTER-8**

**OUTCOMES**

It appears that you've shared both the Python code (Flask web application) and the corresponding HTML template. Your Flask application seems to be a crop recommendation system, utilizing a machine learning model trained on agricultural data to predict the best crop based on user-input environmental and soil parameters. Below are the outcomes or functionalities you can expect from the combined code:

1. **Web Application Structure:** The Flask application has two routes: `'/'` for the home page and `'/predict'` for handling form submissions. The home page (`'/'`) renders an HTML template (`index.html`) with a form for users to input data related to nitrogen, phosphorus, potassium, temperature, humidity, and pH.
2. **User Input Processing:** When the user submits the form ('/predict' route), the entered values are captured in the Flask predict function. The input values are then used to create a feature vector, which is preprocessed using scalers (standscaler.pkl and minmaxscaler.pkl) loaded from pickled files.
3. **Model Prediction:** The preprocessed feature vector is fed into a machine learning model (‘model.pkl’) loaded from a pickled file. The model predicts the best-suited crop based on the provided input parameters.
4. **Displaying Results:** The predicted crop is translated into human-readable form using a dictionary (‘crop\_dict’).The result, along with an associated crop image URL, is passed to the HTML template for rendering.
5. **HTML Template (‘index.html’):** The HTML template uses Bootstrap for styling and responsiveness. It includes a form for users to input agricultural parameters. Upon form submission, the result of the crop prediction is displayed along with an image representing the recommended crop.
6. **Navigation Bar**: The application includes a Bootstrap navigation bar with links to the home page and disabled links for "Contact" and "About."
7. **Styling:** The application has some basic styling, including background colours, card styling for displaying results, and an image for the crop recommendation.
8. **Default Image Handling:** If a specific image for the recommended crop is not available, a default image (default\_crop\_image.jpg) is displayed.
9. **Dependencies:** The application relies on various Python libraries such as Flask, NumPy, pandas, scikit-learn, and pickle for model and scaler loading.
10. **Run the Application:** The application is set to run in debug mode (app.run(debug=True)), making it convenient for development but should be disabled in production.

**CHAPTER-9**

**RESULTS AND DISCUSSIONS**

**9.1 RESULTS**

**TESTING**

To demonstrate its validity, the built programme must be tested. The least creative stage of the system design cycle is thought to be testing. It truly is the phase that shines because it helps the other phases' creativity to come to life.

**9.1.1. VARIOUS LEVELS OF TESTING**

1. White Box Testing

2. Black Box Testing

3.UnitTesting  
 4. Functional Testing

5. Performance Testing

6. Integration Testing

7. Validation Testing

8. System Testing

9. Output Testing

10. User Acceptance Testing

**9.1.2. WHITE BOX TESTING**

White-box testing, sometimes known as glass-box testing, is a test case design technique that generates test cases from the procedural design's control structure. We can create test cases that:

• Ensure that each independent path within a module has been tested at least once by using White Box testing techniques.

Perform all loops at their boundaries and within their operational constraints. Exercise internal data structures to ensure their validity. Exercise all logical judgements on their true and false sides.

**9.1.3. BLACK BOX TESTING**

BLACK BOX TESTING

Black Box Testing involves assessing software functionality without any understanding of the internal workings, structure, or programming language of the module being tested. Test cases for black box testing are derived from definitive source documents such as specifications or requirements. In this approach, the software under test is treated as a black box, and the testing process is conducted without delving into its internal mechanisms. The test inputs are provided, and the outputs are examined without consideration for the internal operations of the software.

Contrary to black box testing, in which the internal workings are not known, testing can also be performed with knowledge of the internal operations to ensure that all internal components function in accordance with specifications and that the entire system operates seamlessly. This approach focuses primarily on verifying the functional requirements of the software.

**9.1.4. UNIT TESTING**

The process of evaluating individual source code units, sets of one or more computer programme modules along with related control data, usage guidelines, and operating procedures to see if they are suitable for use is known as unit testing. A unit is intuitively thought of as the smallest testable component of a programme. A unit in procedural programming can typically be a single function or method, but it can also be an entire module. A unit in object-oriented programming is typically a whole interface, like a class, though it can also be a single method. Short pieces of code, known as unit tests, are written during the development process by programmers or, on occasion, by white box testers.

Unit testing is a technique for software validation and verification where the Unit testing is the process of determining if individual source code units, sets of one or more computer programme modules, use guidelines, and operational procedures are appropriate for use. It makes sense to think of a unit as the smallest testable part of a programme. In procedural programming, a unit can be a whole module, but it can also be a single function or method. In object-oriented programming, a unit might be a single method, but it is usually an entire interface, such as a class. Unit tests are little pieces of code produced by programmers or, sometimes, white box testers during the development process.Software testing is done using unit testing.

**9.1.5. FUNCTIONAL TESTING**

Functional testing is a sort of black box testing that falls under quality assurance (QA) that builds its test cases from the requirements of the software component that is being tested. Internal programme structure is rarely taken into account while testing functions; instead, input is fed into the function and the output is examined (unlike in white-box testing). Typically, functional testing explains the functionality of the system. Different from system testing, functional testing "validates a programme by checking it against the published user or system requirements" whereas system testing "verifies a programme by checking it against... design document(s) or specification(s)" (Kane, Falk, Nguyen 1999, p. 52). There are usually five processes involved in functional testing. the list of tasks that the software should be able to complete.

1. The generation of input data according to the parameters of the function

2. Determining the output according to the parameters of the function

3. Putting the test case into practice

4. A comparison between the actual and anticipated results.

**9.1.6. PERFORMANCE TESTING**

In a broader sense, testing is carried out to assess the responsiveness and stability of a system under specific workloads. Additionally, it serves to scrutinize, quantify, validate, or confirm various quality attributes of the system, including but not limited to scalability, reliability, and resource utilization.

Performance testing, a component of the evolving field of performance engineering in computer science, is specifically focused on evaluating how a system functions concerning its responsiveness and stability, aiming to ensure optimal performance under diverse conditions and steadiness under specific workload conditions. It can also be used to look into, quantify, confirm, or test other aspects of the system's quality, like resource utilisation, scalability, and reliability. A branch of performance engineering, which is a relatively new field in computer science, is performance testing. Performance engineering aims to include performance into a system's architecture, design, and implementation.

**9.1.7. INTEGRATION TESTING**

Integration testing is a methodical process that builds the program's structure while simultaneously running tests to find related issues. It is not reasonable to expect that individual modules, which are very prone to interface mistakes, will function immediately upon integration. Naturally, "putting them together"—that is, interacting—is the issue. Data loss across one another's subfunctions might occur; when combined, they might not create the intended principal function; an acceptable impression on its own might be amplified to an painful degree; and worldwide information designs could cause issues.

Coordination testing is the stage in programming testing wherein individual programming modules are consolidated and tried collectively. Combination testing takes as information modules have been unit tried, bunches them in bigger totals, applies tests characterized in a coordination test plan to those totals, and conveys as its result the incorporated framework prepared. Every one of the mistakes found in the framework are amended for the following stage.The motivation behind incorporation testing is to confirm practical, execution, and unwavering quality necessities put on significant plan things. These "plan things", for example collections (or gatherings of units), are practiced through their connection points utilizing discovery testing, achievement and mistake cases being recreated by means of fitting boundary and information inputs. Reproduced use of shared information regions and between process correspondence is tried and individual subsystems are practiced through their feedback interface. Experiments are developed to test whether every one of the parts inside arrays associate correctlyfor model across strategy calls or interaction actuations, and this is finished in the wake of testing individual modules, for example unit testing.

**9.1.8. VALIDATION TESTING**

Confirmation and Approval are free methodology that are utilized together for really taking a look at that an item, administration, or framework meets necessities and details and that it full fills its expected reason. These are basic parts of a quality administration framework, for example, ISO 9000. The words "check" and "approval" are some of the time went before with "Free" (or IV&V), showing that the confirmation and approval is to be performed by an impartial outsider.It is now and again said that approval can be communicated by the inquiry "Would you say you are building the proper thing?" and check by "Would you say you are building it right?". By and by, the use of these terms differs. Once in a while they are even utilized conversely.

**9.1.9. SYSTEM TESTING**

Framework testing of programming or equipment is trying directed on a total, incorporated framework to assess the framework's consistence with its predetermined prerequisites. Framework testing falls inside the extent of black box testing, and accordingly, ought to require no information on the inward plan of the code or rationale. When in doubt, framework testing takes, as its feedback, all of the "coordinated" programming parts that have passed combination testing and furthermore the product framework itself incorporated with any relevant equipment system(s). The motivation behind coordination testing is to recognize any irregularities between the product units that are incorporated together (called arrays) or between any of the collections and the equipment. Framework testing is a more restricted kind of testing; it looks to distinguish surrenders both inside the "between gatherings" and furthermore inside the framework overall.

Framework testing is performed on the whole framework with regards to a Useful Necessity Specification(s) (FRS) or potentially a Framework Prerequisite Determination (SRS). Framework testing tests the plan, yet additionally the way of behaving and, surprisingly, the trusted assumptions for the client. It is likewise planned to test up to and past the limits characterized in the product/equipment prerequisites particular.

**9.1.10. OUTPUT TESTING**

Subsequent to playing out the approval testing, following stage is yield trying of the proposed framework since no framework could be valuable in the event that it doesn't deliver the expected result created or considered in to two different ways. One is on screen and another is printed design. The result comes as the predefined prerequisites by the client. Thus yield testing brings about no revision in the framework.

**9.1.11. USER ACCEPTANCE TESTING**

Client acknowledgment of a framework is the variable for the outcome of any framework. The framework viable is tried for the client acknowledgment by continually staying in contact with the imminent framework clients at the hour of creating and making changes any place required.

• Input screen plan.

• Yield screen plan.

• Online message to direct client.

• Arrangement of the specially appointed reports and different results.

Stepping through different sorts of examination information does the above testing. Planning of test information assumes an essential part in the framework testing. In the wake of setting up the test information the framework under review is tried utilizing the test information. While testing the framework by utilizing test information blunders are again revealed and right.

**CHAPTER-10**

**CONCLUSION**

In summary, our project makes farming easier for farmers. By using smart technology, we guide them to choose the right crops, saving resources and improving harvests. We focus on simplicity, empowering even those with little tech knowledge. Our aim is to create a greener, more successful future for farming, where everyone can grow more food while being kind to the environment.

Encouraging sustainable farming methods aligns with our vision of a future where agriculture coexists harmoniously with nature. As we move forward, we remain dedicated to continuous improvement, embracing innovations that further simplify farming, bolster productivity, and ensure a greener, more abundant tomorrow for farmers conclusion, the "Smart Farming" project not only advances the field of precision agriculture but also contributes to the sustainable and efficient future of farming.

The integration of advanced technologies, coupled with a user-centric design and real-world validation, positions the solution as a valuable asset for farmers striving for increased productivity and environmental stewardship. As we move forward, the project's impact will continue to unfold, shaping the trajectory of predictive agriculture and its transformative potential in the farming sector.

As technology continues to evolve, the application serves as a foundation for future innovations in precision agriculture. With a commitment to ongoing testing, refinement, and user feedback, this codebase has the potential to become a cornerstone in the realm of digital solutions for agriculture, contributing to increased efficiency and sustainability in crop cultivation practices.

**REFERENCES**

[1] Ersin Elbasi and Nour Mostafa, “Artificial Intelligence Technology in the Agriculture Sector: A Systematic Literature Review”, The American University of the middle East, [(PDF) Artificial Intelligence Technology in the Agricultural Sector: A Systematic Literature Review (researchgate.net)](https://www.researchgate.net/publication/366612575_Artificial_Intelligence_Technology_in_the_Agricultural_Sector_A_Systematic_Literature_Review) **,**vol.11, Jan 2023.

[2] Tanha Talaviya, Dhara Shah, Nivedita patel, Hiteshri Yagnik and Manan Shah, “Artificial Intelligence in Agriculture”,

<https://doi.org/10.1016/j.aiia.2020.04.002>, vol.4,2020.

[3] Mark Ryan, Gohar Iskhanya and Bedir Tekinerdongan, “An interdisciplinary approach to artificial intelligence”,

<https://doi.org/10.1080/27685241.2023.2168568>, 2023, vol.95,no.1,2168568.

[4] Priya Rani Bhagat, Farheen Naz and Robert Magda, “Artificial Intelligence solutions enabling sustainable agriculture: A bibliometric analysis”, <https://doi.org/10.1371/journal.pone.0268989>, 2022.

[5] Swapnil Raut and Vidya Chitre, “Soil Monitoring and Testing using IotT for Fertility Level and Crop Prediction” , <https://dx.doi.org/10.2139/ssrn.3561682>, 2022.

[6] A.P.Dewangga and H.S. Suhuono, “Internet Of Things in the filed of smart farmings: [Internet of Things in the Field of Smart Farming: Benefits and Challenges | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9307602).

[7] N. Paudel and T.Huang, “Study on real-time detection of pipe bursts with simulation and management System on water distribution network, “in Proc. Int. Conf. Piplines Trenchless Technol. (ICPTT), Sep. 2009, pp.217-223, doi: [Study on Real-Time Detection of Pipe Bursts with Simulation and Management System on Water Distribution Networks | Proceedings | Vol , No (ascelibrary.org)](https://ascelibrary.org/doi/10.1061/41073%28361%2923).

[8] K. Priyadharsini and V. Yazhin, “ A novel methodology for smart soil management and health assessment system to support the contemporary cultivation using AI on mobile app environment,” Int. J. Adv. Sci. Tech-nol., vol. 7, no. 29, p. 7, 2020, doi: [Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides – ScienceDirect](https://www.sciencedirect.com/science/article/pii/S258972172030012X?via%3Dihub).

[9] H.S.Abdullahi, F.mahieddine and R.E.Sheriff,2015 [Technology Impact on Agricultural Productivity: A Review of Precision Agriculture Using Unmanned Aerial Vehicles | SpringerLink](https://link.springer.com/chapter/10.1007/978-3-319-25479-1_29).

[10] Zhu, H., Lan, Y., Wu, W., Hoffmann, W.C., Huang, Y., Xue, X., Firtz, B., 2010. [Development of a PWM Precision Spraying Controller for Unmanned Aerial Vehicles – ScienceDirect](https://www.sciencedirect.com/science/article/pii/S167265291060251X). Journal of Bionic Engineering 7 (3), 276-283.

**APPENDIX-A**

**PSUEDOCODE**

**PSUEDOCODE**

**app.py(file)**

from flask import Flask,request,render\_template

import numpy as np

import pandas

import sklearn

import pickle

# importing model

model = pickle.load(open('model.pkl','rb'))

sc = pickle.load(open('standscaler.pkl','rb'))

ms = pickle.load(open('minmaxscaler.pkl','rb'))

crop\_image\_urls = {

    "Rice": "/static/rice.jpg",

    "Maize": "/static/maize.jpg",

    "Jute": "/static/jute.jpg",

    "Cotton": "/static/cotton.jpg",

    "Coconut": "/static/coconut.jpg",

    "Papaya": "/static/papaya.jpg",

    "Orange": "/static/orange.jpg",

    "Apple": "/static/apple.jpg",

    "Muskmelon": "/static/muskmelon.jpg",

    "Watermelon": "/static/watermelon.jpg",

    "Grapes": "/static/grapes.jpg",

    "Mango": "/static/mango.jpg",

    "Banana": "/static/banana.jpg",

    "Pomegranate": "/static/pomegranate.jpg",

    "Lentil": "/static/lentil.jpg",

    "Blackgram": "/static/blackgram.jpg",

    "Mungbean": "/static/mungbean.jpg",

    "Mothbeans": "/static/mothbeans.jpg",

    "Pigeonpeas": "/static/pigeonpeas.jpg",

    "Kidneybeans": "/static/kidneyBeans.jpg",

    "Chickpea": "/static/chickpeas.jpg",

    "Coffee": "/static/coffee.jpg",

    }

# Add more crop-image mappings as needed

# }

# creating flask app

app = Flask(\_\_name\_\_)

@app.route('/')

def index():

    return render\_template("index.html")

@app.route("/predict",methods=['POST'])

def predict():

    N = request.form['Nitrogen']

    P = request.form['Phosporus']

    K = request.form['Potassium']

    temp = request.form['Temperature']

    humidity = request.form['Humidity']

    ph = int(request.form['Ph'])

    feature\_list = [N, P, K, temp, humidity, ph, 0]

    single\_pred = np.array(feature\_list).reshape(1, -1)

    scaled\_features = ms.transform(single\_pred)

    final\_features = sc.transform(scaled\_features)

    prediction = model.predict(final\_features)

    crop\_dict = {1: "Rice", 2: "Maize", 3: "Jute", 4: "Cotton", 5: "Coconut", 6: "Papaya", 7: "Orange",

                 8: "Apple", 9: "Muskmelon", 10: "Watermelon", 11: "Grapes", 12: "Mango", 13: "Banana",

                 14: "Pomegranate", 15: "Lentil", 16: "Blackgram", 17: "Mungbean", 18: "Mothbeans",

                 19: "Pigeonpeas", 20: "Kidneybeans", 21: "Chickpea", 22: "Coffee"}

    if prediction[0] in crop\_dict:

        crop = crop\_dict[prediction[0]]

        result = "{} is the best crop to be cultivated right there".format(crop)

        result\_crop\_image = crop\_image\_urls.get(crop, None)

    else:

        result = "Sorry, we could not determine the best crop to be cultivated with the provided data."

    return render\_template('index.html',result = result, result\_crop\_image=result\_crop\_image)

# python main

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(debug=True)

* **index.html**

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Bootstrap demo</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ" crossorigin="anonymous">

  </head>

  <style>

        h1 {

            color: pink;

            text-align: center;

        }

        .warning {

            color: red;

            font-weight: bold;

            text-align: center;

        }

        .card{

        margin-left:410px;

        margin-top: 20px;

        color: white;

        }

        .container{

        background:#edf2f7;

        font-weight: bold;

        padding-bottom:10px;

        border-radius: 15px;

        }

        .crop-image {

        max-width: 100%;

        height: auto;

        }

    </style>

  <body style="background:#BCBBB8">

  <!--=======================navbar=====================================================-->

<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

  <div class="container-fluid">

    <a class="navbar-brand" href="/">Crop Recommendation</a>

    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">

      <span class="navbar-toggler-icon"></span>

    </button>

    <div class="collapse navbar-collapse" id="navbarSupportedContent">

      <ul class="navbar-nav me-auto mb-2 mb-lg-0">

        <li class="nav-item">

          <a class="nav-link active" aria-current="page" href="#">home</a>

        </li>

        <li class="nav-item">

          <a class="nav-link disabled" href="#">Contact</a>

        </li>

        <li class="nav-item">

          <a class="nav-link disabled">About</a>

        </li>

      </ul>

      <form class="d-flex" role="search">

        <input class="form-control me-2 disabled" type="search" placeholder="Search" aria-label="Search">

        <button class="btn btn-outline-success disabled" type="submit">Search</button>

      </form>

    </div>

  </div>

</nav>

<!--==========================================================================================-->

  <div class="container my-3 mt-3">

      <h1 class="text-success">Crop Recommendation System <span class="text-success">🌱</span></h1>

<!--      adding form-->

    <form action="/predict" method="POST" onsubmit="return validateForm()">

    <div class="row">

      <div class="col-md-4">

        <label for="Nitrogen">Nitrogen</label>

        <input type="number" id="Nitrogen" name="Nitrogen" placeholder="Enter Nitrogen (0-1000)" class="form-control" required>

      </div>

      <div class="col-md-4">

        <label for="Phosporus">Phosphorus</label>

        <input type="number" id="Phosporus" name="Phosporus" placeholder="Enter Phosphorus (0-1000)" class="form-control" required>

      </div>

      <div class="col-md-4">

        <label for="Potassium">Potassium</label>

        <input type="number" id="Potassium" name="Potassium" placeholder="Enter Potassium (0-1000)" class="form-control" required>

      </div>

    </div>

    <div class="row mt-4">

      <div class="col-md-4">

        <label for="Temperature">Temperature</label>

        <input type="number" step="0.01" id="Temperature" name="Temperature" placeholder="Enter Temperature in °C (-50 to 50)" class="form-control" required>

      </div>

      <div class="col-md-4">

        <label for="Humidity">Humidity</label>

        <input type="number" step="0.01" id="Humidity" name="Humidity" placeholder="Enter Humidity in % (0-100)" class="form-control" required>

      </div>

      <div class="col-md-4">

        <label for="pH">pH</label>

        <input type="number" step="0.01" id="Ph" name="Ph" placeholder="Enter pH value (0-14)" class="form-control" required>

      </div>

    </div>

           <div class="row mt-4">

           <div class="col-md-12 text-center">

                <button type="submit" class="btn btn-primary btn-lg">Get Recommendation</button>

            </div>

            </div>

      </form>

      {% if result %}

      <div class="card bg-dark" style="width: 18rem; margin-top: 20px;">

          {% if result\_crop\_image %}

              <img src="{{ result\_crop\_image|safe }}" class="card-img-top crop-image" alt="Crop Image">

          {% else %}

              <!-- Provide a default image if no specific image is available -->

              <img src="{{url\_for('static', filename='default\_crop\_image.jpg')}}" class="card-img-top crop-image" alt="Default Crop Image">

          {% endif %}

          <div class="card-body">

              <h5 class="card-title">Recommend Crop for cultivation is:</h5>

              <p class="card-text">{{ result }}</p>

          </div>

      </div>

    {% endif %}

    <!-- {% if result %}

        <div class="card bg-dark" style="width: 18rem;">

          <img src="{{url\_for('static', filename='agri.jpg')}}" class="card-img-top" alt="...">

          <div class="card-body">

            <h5 class="card-title">Recommend Crop for cultivation is:</h5>

            <p class="card-text">{{ result }}</p>

          </div>

        </div>

    {% endif %} -->

  </div>

  <script>

    function validateForm() {

      var nitrogen = document.getElementById("Nitrogen").value;

      var phosphorus = document.getElementById("Phosporus").value;

      var potassium = document.getElementById("Potassium").value;

      var temperature = document.getElementById("Temperature").value;

      var humidity = document.getElementById("Humidity").value;

      var ph = document.getElementById("Ph").value;

      // Define your valid ranges

      var validRange = {

        Nitrogen: { min: 0, max: 1000 },

        Phosphorus: { min: 0, max: 1000 },

        Potassium: { min: 0, max: 1000 },

        Temperature: { min: -50, max: 50 },

        Humidity: { min: 0, max: 100 },

        Ph: { min: 0, max: 14 },

      };

      // Validate against the defined ranges

      if (

        isNaN(nitrogen) || nitrogen < validRange.Nitrogen.min || nitrogen > validRange.Nitrogen.max ||

        isNaN(phosphorus) || phosphorus < validRange.Phosphorus.min || phosphorus > validRange.Phosphorus.max ||

        isNaN(potassium) || potassium < validRange.Potassium.min || potassium > validRange.Potassium.max ||

        isNaN(temperature) || temperature < validRange.Temperature.min || temperature > validRange.Temperature.max ||

        isNaN(humidity) || humidity < validRange.Humidity.min || humidity > validRange.Humidity.max ||

        isNaN(ph) || ph < validRange.Ph.min || ph > validRange.Ph.max

      ) {

        alert("Please enter valid values within the specified ranges.");

        return false;

      }

      return true;

    }

  </script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe" crossorigin="anonymous"></script>

  </body>

</html>

**APPENDIX-B**

**SCREENSHOTS**

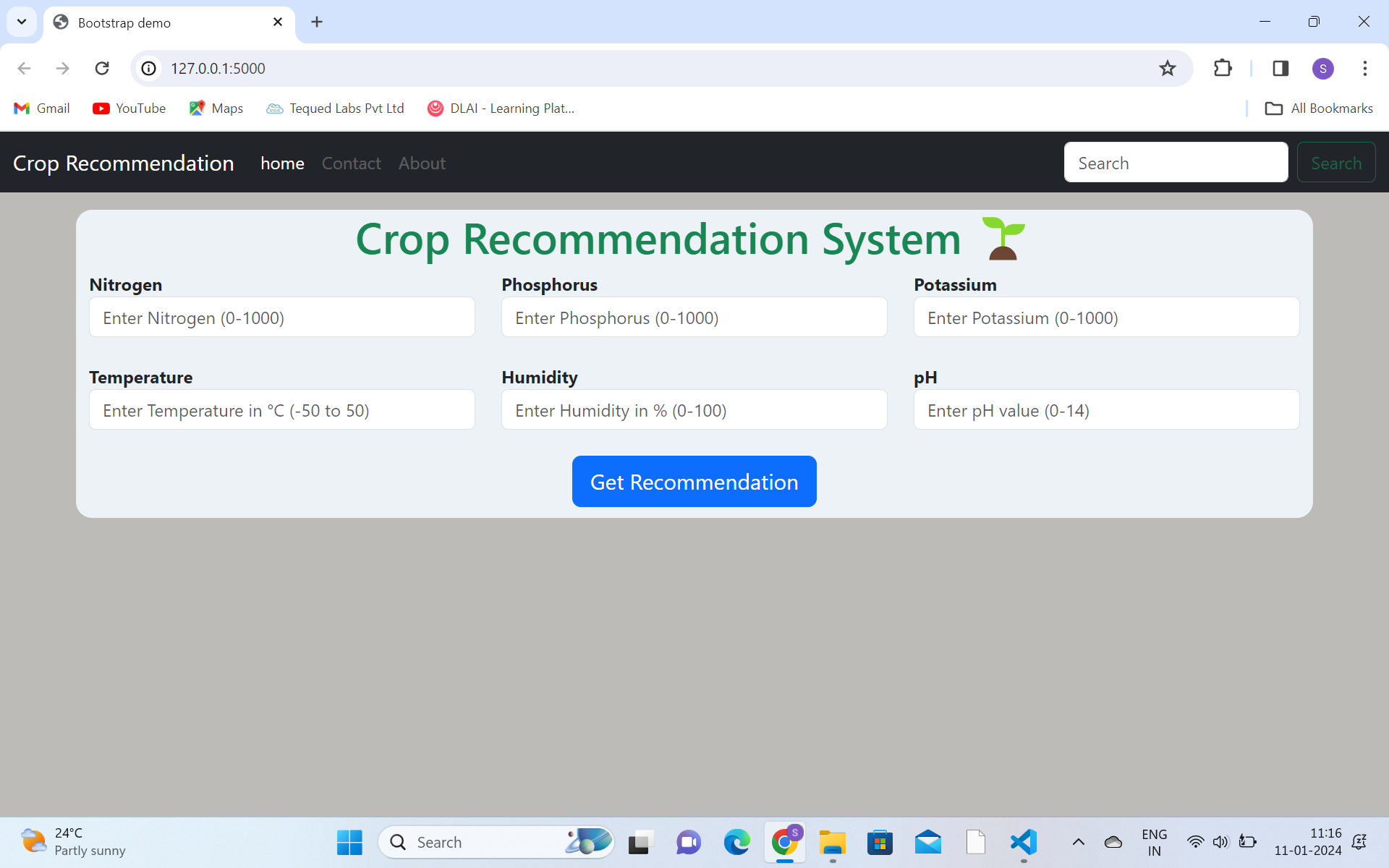


Fig 2.3: Home page

The homepage consists of 6 inputs i.e,nitrogen, phosporous, potassium, temperature, humidity, ph and a submit but called get recommendation button.

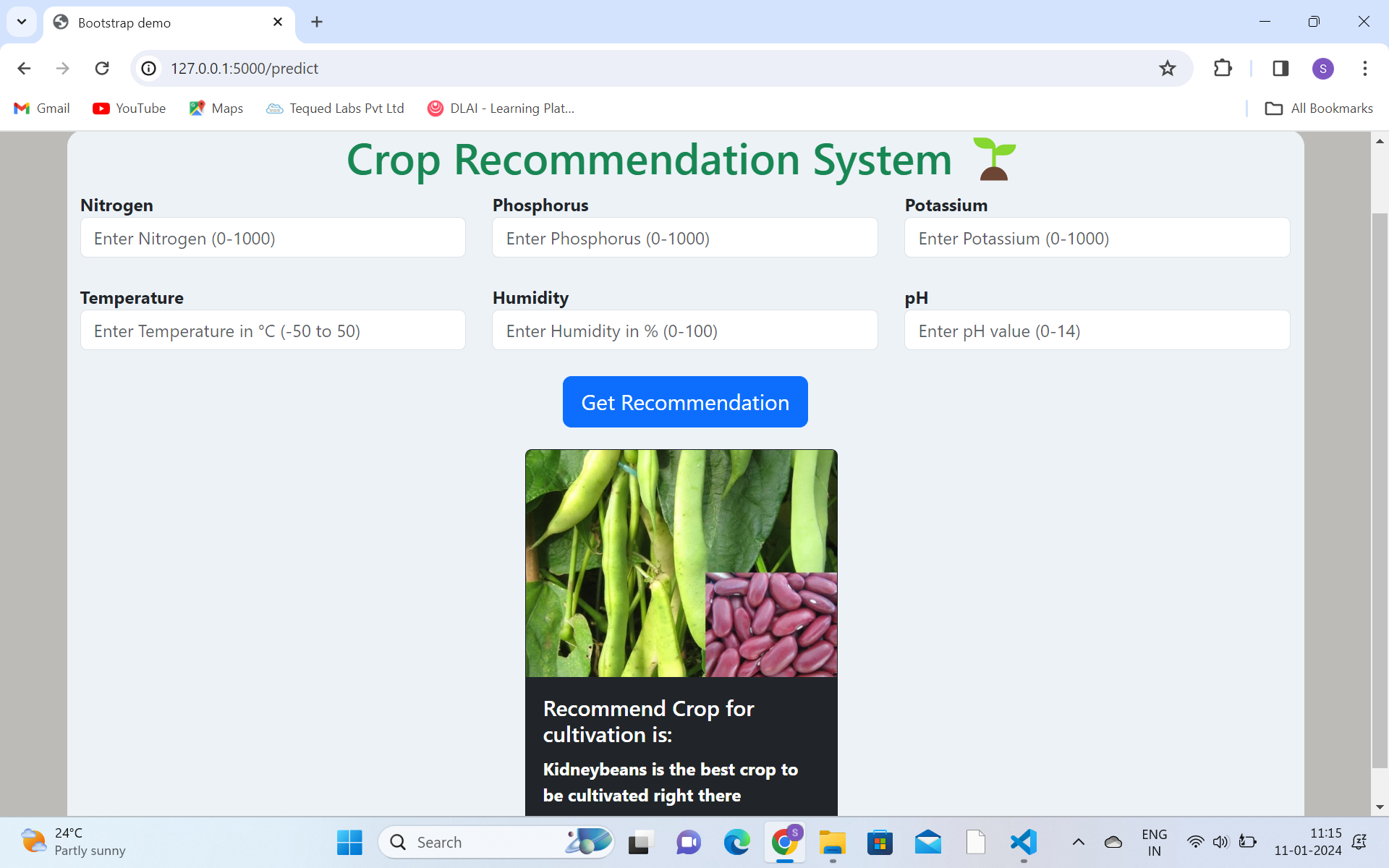


Fig 3.1: Demo 01.

Now in this, when we enter the values of each inputs it gives the crop that is suitable for the land will be dislplayed with the image as shown in fig 3. 1. Demo 01.

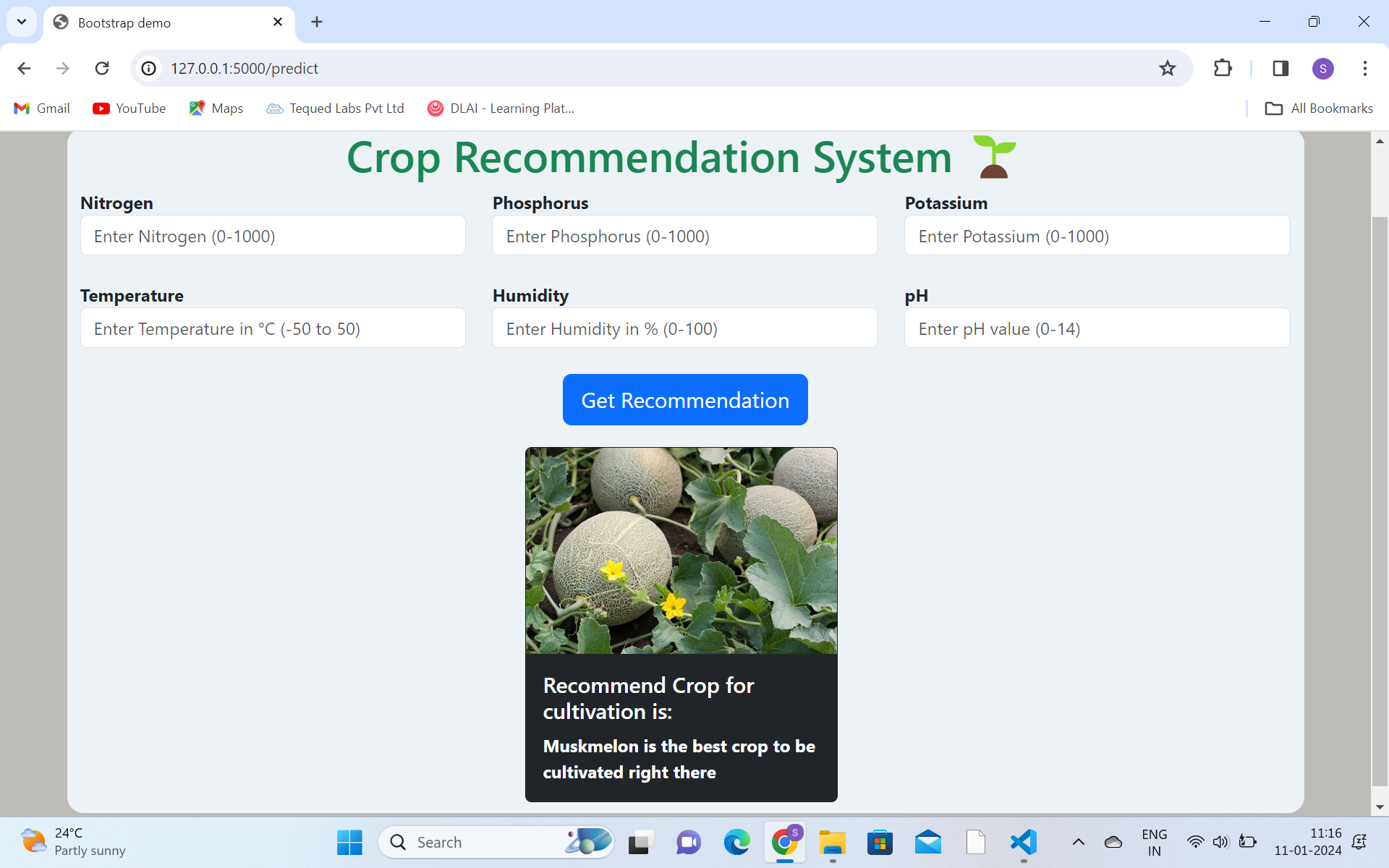


Fig 3.2: Demo 02

Now in this, when we enter the values of each inputs it gives the crop that is suitable for the land will be dislplayed with the image as shown in fig 3. 1. Demo 02.

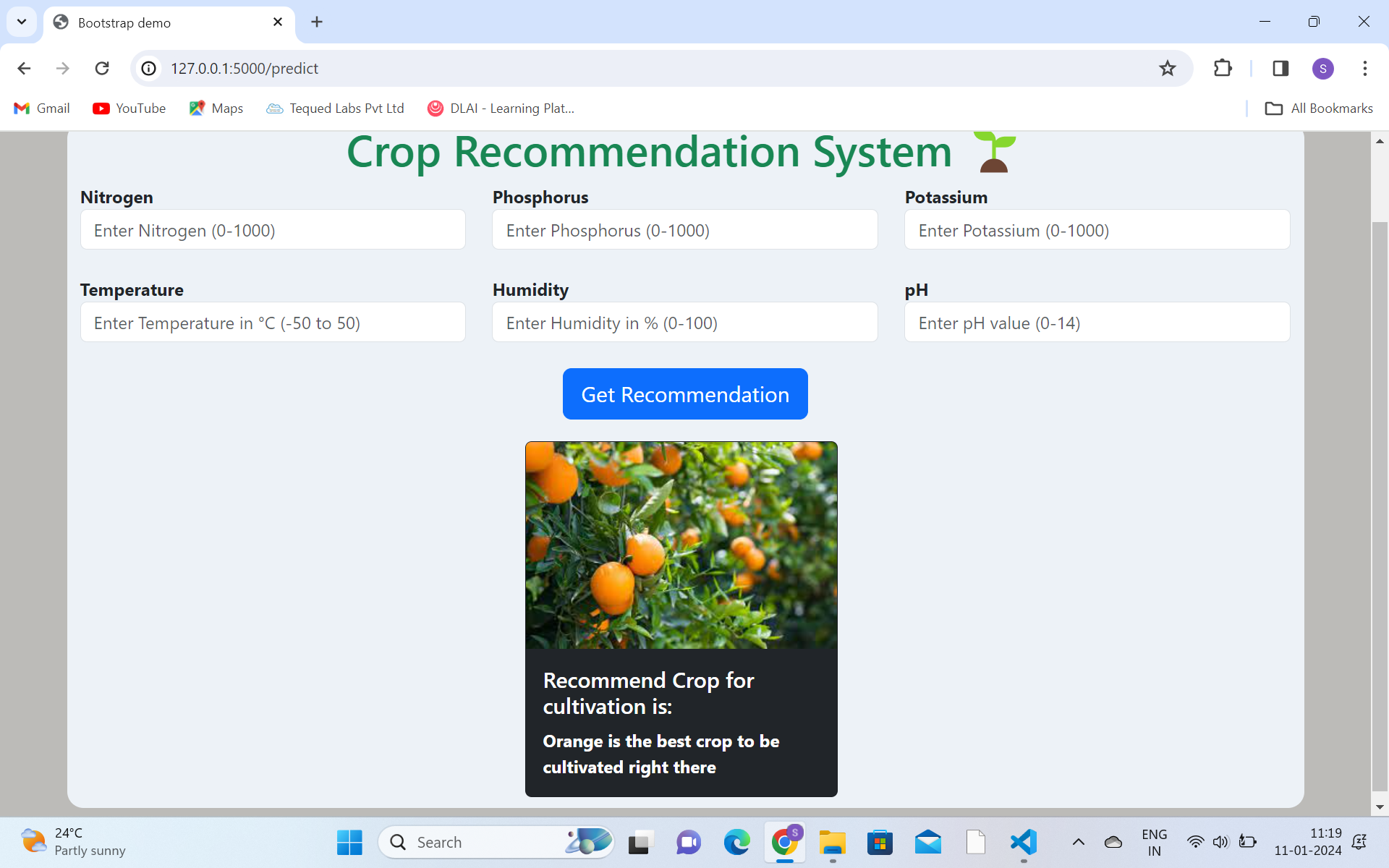


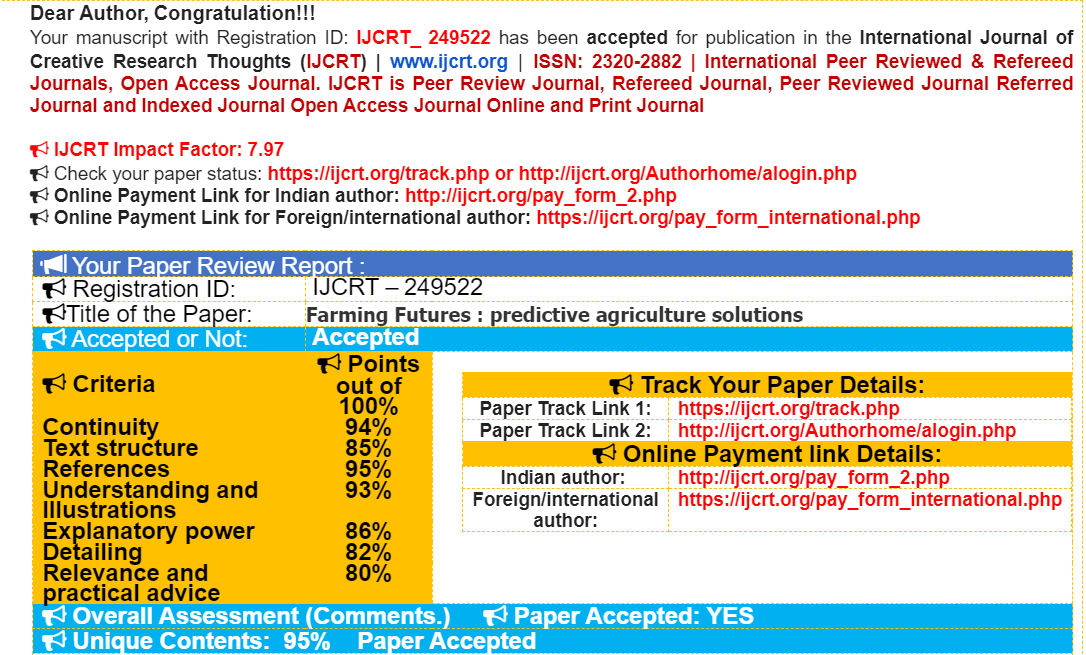
Fig 3.3: Demo 03

Now in this, when we enter the values of each inputs it gives the crop that is suitable for the land will be dislplayed with the image as shown in fig 3. 1. Demo 03.

**APPENDIX-C**

**ENCLOSURES**

* **Journal paper accepted:**

****

* **Plagiarism Report**

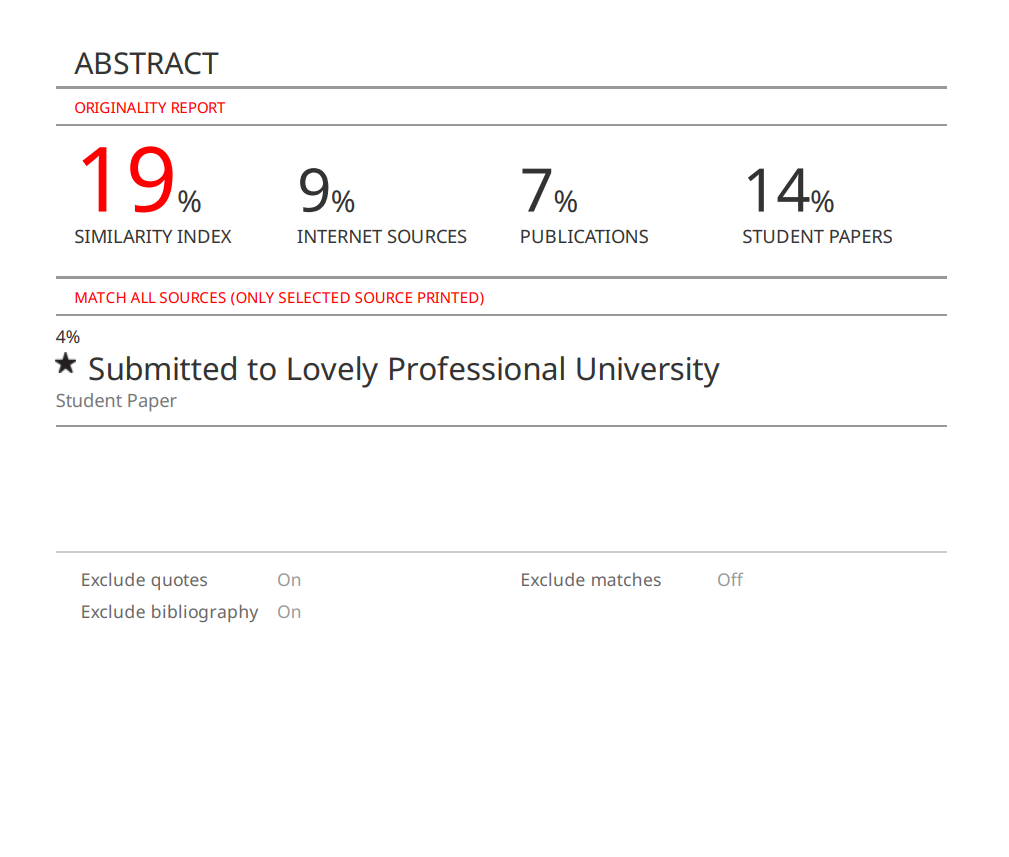


Fig 4.1: Plagiarism Report

**APPENDIX-C**

**SDG MAPPING**



**The Project work carried out here is mapped to SDG-12 Responsible Consumption and Production.**

Using smart computers in farming helps us grow food in a better way. The United Nations, especially the Food and Agriculture Organization (FAO), supports this idea to make farming more efficient and good for the environment. They encourage farmers to use artificial intelligence (AI) to grow crops smarter. AI can help farmers use resources wisely, make less waste, and be better at taking care of the land. This is like a helpful friend for farmers, making sure they grow food in a way that's good for everyone and the planet. It's like a big step toward making sure we have enough food and also take care of our home, Earth.