SMART FARMING : CROP RECOMMENDATION USING DEEP LEARNING TECHNIQUES

A PROJECT REPORT

Submitted by,

20201CSE0216	**Shravana S.S**
20201CSE0247	**Poojarani**
20201CSE0211	**Rashi**
20201CSE0248	**Chethan A**
20201CSE9004	**Pourmamilla Mohith Gowda**

Under the guidance of,

Ms. RANJITHA P
Assistant Professor

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & 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 **"SMART FARMING USING DEEP LEARNING TECHNIQUES"** being submitted by "Shravana S S, Poojarani, Rashi, Chethan A, Pourmamilla Mohith Gowda" bearing roll number(s) "20201CSE0216, 20201CSE0247, 20201CSE011, 20201CSE0248, 20201CSE9004" 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.

Ms.Ranjitha P
Assistant Professor
School of CSE&I
Presidency University

Dr. PALLAVI R
Associate 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 **SMART FARMING: CROP PREDICTION USING DEEP LEARNING TECHNIQUES** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Ms. Ranjitha P**, Assistant Professor, **School of Computer Science and 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	20201CSE0216	Shravana S S	
2	20201CSE0247	Poojarani	
3	20201CSE0211	Rashi	
4	20201CSE0248	Chethan A	
5	20201CSE9004	Pourmamilla Mohith Gowd	

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 primary goal is to provide farmers with robust data-driven decision-making tools, assisting them in selecting optimal crops tailored to their specific environmental conditions. 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 **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering & 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 Engineering & Information Science, Presidency University and **Dr. Pallavi R** Head of the Department, School of Computer Science and Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Ms.Ranjitha P**, School of Computer Science Engineering & Information Science, 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 **Mr. Zia Ur Rahman, Mr. Peniel John Whistely** and also the department Project Coordinators are **Dr.Sanjeev P Kaulgud**

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

Shravana S S

Poojarani

Rashi

Chethan A

Pourmamila Mohith Gowd

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

1.1. 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 are a huge wellspring of food and fiber for the total populace. World Asset Establishment is attempting to tackle the issue of how to take care of ten billion individuals reasonably by 2050. In this way, expanding excellent harvest yield is vital. The selection of harvests to plant can essentially influence crop yields and productivity. Environmental change and other natural elements make it intense to foresee which yield will succeed, given the area.

In this task, we use AI to prescribe yields to ranchers. To begin with, gather the dataset and preprocess it. Then, we train and test models utilizing elements, for example, soil content and type, soil pH esteem, temperature, moistness, and precipitation. We likewise endeavored highlight designing ideas to confirm in the event that the model performs better involving a blend of various elements and use it as another component in the equivalent dataset. Agribusiness has general difficulties, and with regards to AI, consequently, we feature these difficulties completely. Ultimately, we present a few invigorating thoughts for the perusers to wander into.

By consolidating AI procedures with horticultural information, ranchers gain the capacity to pursue information driven choices, going from crop determination and water system the board to bug control and yield expectation. This combination enables ranchers to accomplish higher effectiveness, asset streamlining, and maintainable practices, eventually prompting further developed efficiency and benefit in the agrarian area.

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. An 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.

2. Crop Selection Method to maximize crop yield rate using machine learning technique

Journal Details: Published in 2015 by M.P. Singh, Prabhat Kumar, J.P. Singh

Abstract: Farming arranging assumes a huge part in the monetary development and food security of agro-based countries. Selection of crop(s) is a significant issue for horticulture arranging. It relies upon different boundaries, for example, creation rate, market cost, and government strategies. Numerous analysts concentrated on the expectation of yield pace of harvest, forecast of climate, soil arrangement and harvest grouping for horticulture arranging utilizing insights strategies or AI procedures. On the off chance that there is more than each choice to establish a yield in turn utilizing restricted land asset, then, at that point, determination of harvest is a riddle. This paper proposed a technique named Harvest Choice Strategy (CSM) to tackle crop determination issue, and expand net yield pace of yield over season and hence accomplishes greatest financial development of the country. The proposed technique might further develop net yield pace of harvests.

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.

3. **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 setup costs, including hardware, software, and training, can be a barrier for small-scale farmers or those in resource-constrained regions.

- **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.

-

4. **Soil Based Prediction for Crop Yield using Predictive Analytics**
Journal Details: Published in 2021 by M. Chandraprabha, Rajesh Kumar Dhanaraj
Abstract: : Soil is the principal part and assumes a critical part in farming. In light of the supplements and pH worth of the dirt, crop not entirely set in stone. Ranchers are as yet utilizing conventional way to deal with examination the dirt quality. The procedures like Information Mining, Man-made brainpower, AI, Profound learning and Prescient Examination are the arising advances in exploration to work on the farming field. Prescient investigation is a method of AI that predicts the future results and examination depends on the verifiable or past information. In farming, prescient examination assists with anticipating or distinguish the dirt supplements level expected for the harvests like Paddy, Raagi, Cumbu and so on.,. In this paper, the dirt based dataset is gathered from TNAU site and it has 32 regions of Tamilnadu. The calculations, for example, Innocent bayes, Bayes Net, and IbK have been conveyed to anticipate the yield assortment reasonable for the dirt in view of the absolute creation and region planted locale wise. Likewise, its exactness levels are thought about. The exactness is resolved utilizing genuine positive worth, misleading positive worth, accuracy, review, f-measure and MCC.

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.

5. **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 SYSTEM

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 METHODOLOGY

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 itscomponents. 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

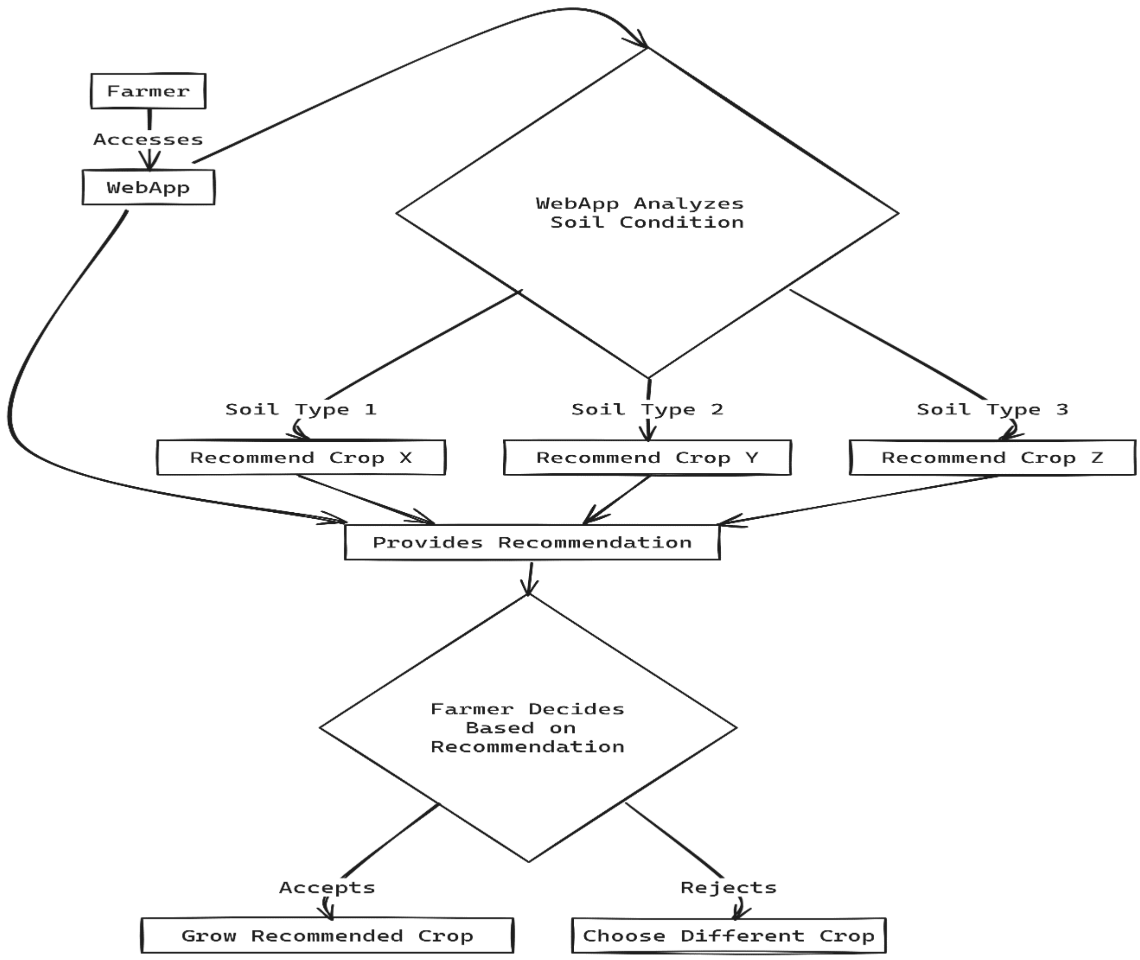


Fig 1.1:Use case diagram

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

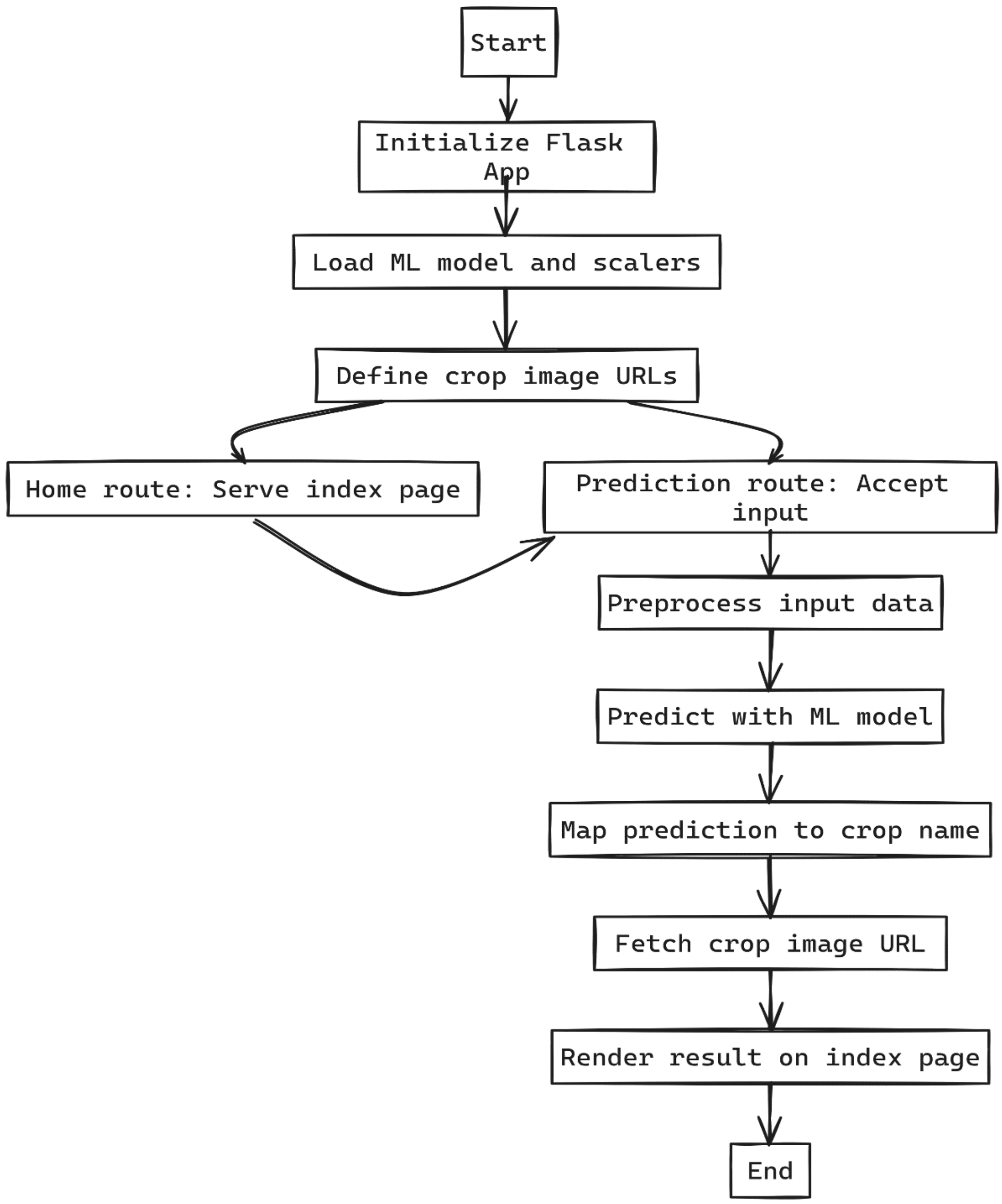


Fig 1.2: State diagram

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 actual crop 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

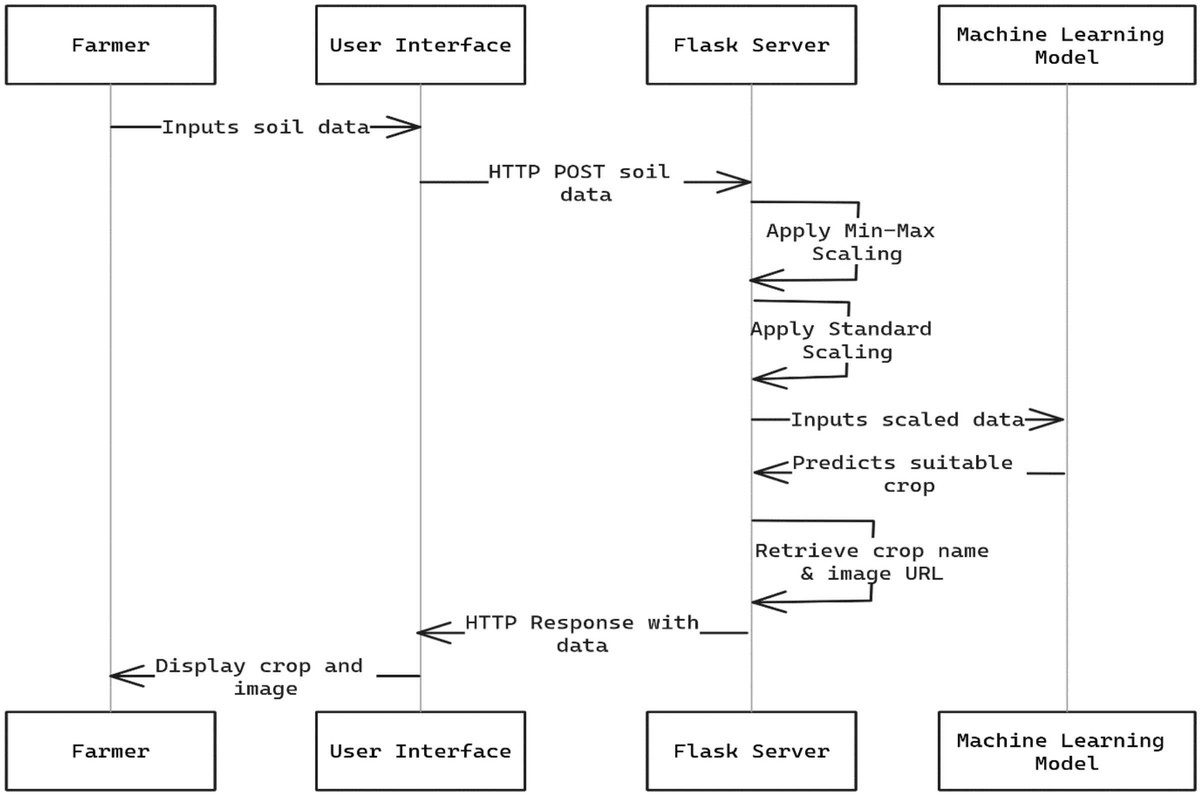


Fig 1.3:System Archietecture diagram

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

a. Python:

Python is a deciphered, object-arranged, undeniable level, progressively semantic programming language. It is especially attractive for Quick Application Improvement as well concerning utilization as a prearranging or stick language to integrate existing parts because of its undeniable level implicit information structures, dynamic composing, and dynamic restricting. Python's clear punctuation underlines comprehensibility and simplifies it to realize, which brings down the expense of program upkeep. Python's help for modules and bundles advances the measured quality and reuse of code in programs. Both source code and paired adaptations of the complete standard library and the python mediator are accessible. The alter test-investigate cycle is phenomenally fast since there is no accumulation stage. Python programs are easy to investigate since a division disappointment is never brought about by a bug or wrong information. All things being equal, on the off chance that the translator recognizes an error, an exemption is raised by it. The mediator delivers a stack follow on the off chance that the program doesn't get the special case. Setting breakpoints, assessing erratic articulations, investigating neighborhood and worldwide factors, venturing through the code each line in turn, and different highlights are conceivable with a source level debugger. Python's capacity to do thoughtfulness is shown by the debugger, which is created in python. Then again, adding a couple of print explanations to the source code is habitually the quickset method for troubleshooting a program because of the short alter test-investigate cycle. It is accessible with the expectation of complimentary circulation on every famous stage.

b. Machine Learning:

An area of fake intelligence(AI) called AI is dedicated to making programming programs that, without being unequivocally coded, gain from information over the long run and become more precise.

A calculation is a progression of measurable handling processes in information science. In AI, calculations are 'prepared' to filter through huge volumes of information looking for examples and qualities to determine end and gauges in view of new information. As it investigations more information, a more intelligent calculation will give ends and forecasts that are more precise.

AI models are surrounding us now. At the point when we address computerized colleagues, they answer by playing music and doing web look. sites give proposals to products, movies, and music in light of what we've recently bought, saw, or paid attention to. Our floors are vacuumed by robots while we invest our energy doing more prominent things. Undesirable messages are kept from entering our inboxes by spam indicators. Frameworks for breaking down clinical pictures help clinicians in finding cancers they have missed. Furthermore, the primary independent vehicles are currently going on open streets.

We might expect more. AI will increment efficiency in both our own and proficient lives as large information keeps on developing, PCs get all the more impressive and available, and information researchers keep on making more powerful calculations.

The working of AI

Building an AI application (or model) includes four basic stages. Information Researchers who intently team up with the business leaders for whom the model is being produced often carry out the tasks.

Stage 1: Select and set up a preparation informational collection

An informational collection utilized for preparing is an example of the information that will be ingested by the AI model to resolve the main thing. In specific conditions, the preparation information is "labeled" to cause to notice the attributes and classifications the model should perceive. Since a portion of the information is unlabeled, the model should separate those qualities and make its own groupings.

Stage 2: Pick a calculation to run on the preparation informational index

A progression of measurable handling stages make up a calculation. The sort of calculation utilized depends on the amount and sort of information (named or unlabeled) in the preparation informational index, as well as the job that needs to be done.

Coming up next are instances of AI calculations regularly utilized with named information:

- Relapse calculations
- Choice trees
- Occasion based calculations

Calculations for use with unlabeled information incorporate the accompanying:

- Grouping calculations
- Affiliation calculations
- Brain Organizations

Stage 3: Preparing the calculation to make the model

Yet again iteratively running factors through the calculation, contrasting the result and the normal outcomes, changing loads and inclinations inside the calculation that could create a more exact outcome, and afterward running the factors through the calculation until the calculation reliably returns the right response.

Stage 4: Utilizing and working on the model

The model must then be utilized with new information, and in a perfect world it will do as such after some time while likewise turning out to be more precise and effective. The issue that must be tackled will figure out where the new information starts from. An AI model used to work a robot hoover. Cleaner, for example, may ingest information from certifiable cooperations with moving furnishings or new items in the room, yet an AI model used to recognize spam will retain 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. The algorithm builds multiple decision trees during training and merges their 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 requires minimal 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.

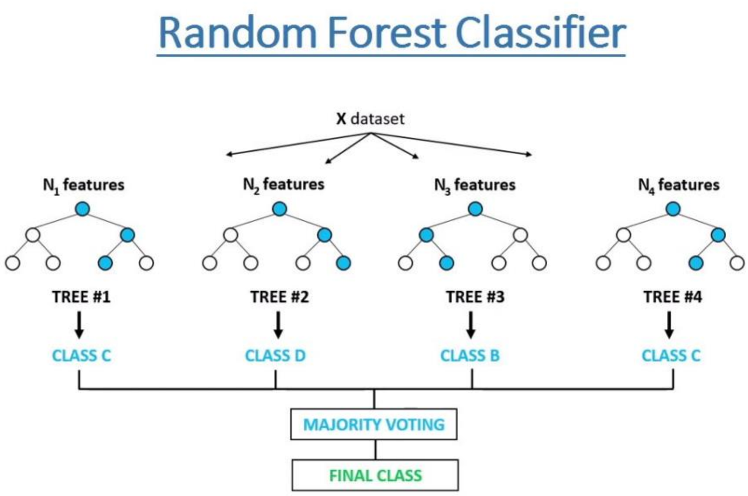


Fig 2.1: Random forest classifier

We can run arbitrary woodland relapses in different projects like SAS, R, and python. In an irregular woods relapse, each tree creates a particular forecast. The mean expectation of the singular trees is the result of the relapse. This is in opposition to irregular woodland arrangement, whose not set in stone by the method of the choice trees' class. Albeit irregular woodland relapse and straight relapse follow similar idea, they vary regarding capabilities. The capability of straight relapse is y=bx + c, where y is the reliant variable, x is the free factor, b is the assessment boundary, and c is a steady. The capability of a complicated irregular backwoods relapse is like 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 used for modeling the relationship between a dependent variable and one or more independent 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 and business 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 :** NumPy is a powerful library in Python for numerical computing. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. NumPy is the fundamental package for scientific computing in Python and is widely used in various domains such as machine learning, data analysis, signal processing, and more.

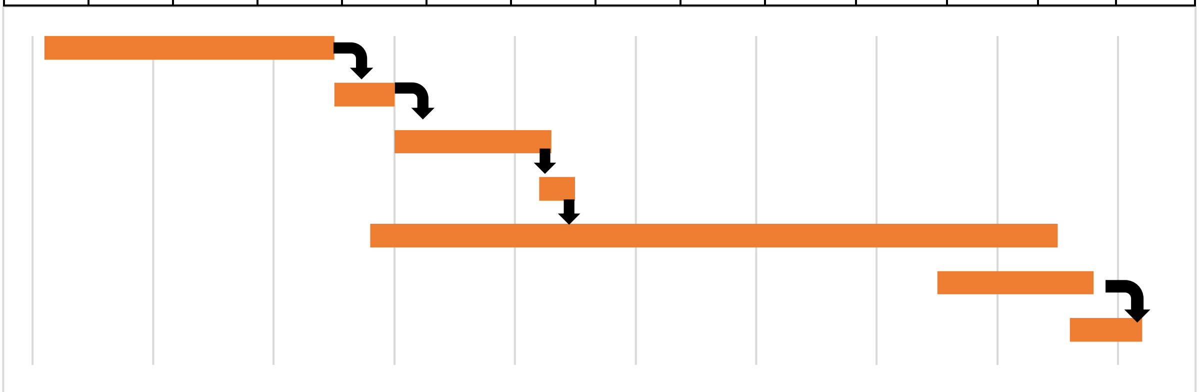
- NumPy, a fundamental library for numerical computing in Python, introduces key features that empower efficient handling of large-scale numerical data. At its core is the powerful numpy.ndarray, facilitating the representation of multi-dimensional arrays and matrices. With an extensive array of mathematical 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

(GANTT CHART)

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



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

Fig 2.2: Gantt chart

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

The software, which has been developed, has to be tested to prove its validity. Testing is considered to be the least creative phase of the whole cycle of system design. In the real sense it is the phase, which helps to bring out the creativity of the other phases makes it shine.

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, some of the time called glass-box, is an experiment plan technique that utilizes the control construction of the procedural plan to infer experiments. Utilizing White Box testing strategies, we can infer experiments that

• Ensure that all free ways inside a module have been practiced no less than once

• Practice all consistent choices on their valid and misleading sides.

• Execute all circles at their limits and inside their functional limits.

• Practice inside information designs to guarantee their legitimacy.

9.1.3. BLACK BOX TESTING

Discovery Testing will be trying the product with next to no information on the inward functions, construction or language of the module being tried. Black box tests, as most different sorts of tests, should be composed from a conclusive source record, for example, detail or necessities report, for example, particular or prerequisites report. It is a trying wherein the product under test is dealt with, as a black box. You can't "see" into it. The test gives inputs and answers yields disregarding the way in which the product works.

In this testing by knowing the inside activity of an item, test can be directed to guarantee that "all cog wheels network", that is the inside activity performs as per detail and all interior parts have been enough worked out. It on a very basic level spotlights on the practical prerequisites of the product.

9.1.4. UNIT TESTING

Unit testing is a technique by which individual units of source code, sets of at least one PC program modules along with related control information, use methodology, and working systems are tried to decide whether they are good for use. Naturally, one can see a unit as the littlest testable piece of an application. In procedural programming, a unit could be a whole module, yet it is all the more generally a singular capability or methodology. In object situated programming, a unit is in many cases a whole connection point, like a class, however could be a singular technique. Unit tests are short code sections made by software engineers or once in a while by white box analyzers during the improvement cycle.

Unit testing is programming check and approval strategy in which the singular units of source code are tried fit for use. A unit is the littlest testable piece of an application. In this testing, each class is tried to work acceptably.

Unit testing includes the plan of experiments that approve that the inside program rationale is working appropriately, and that program inputs produce legitimate results. All choice branches and inward code stream ought to be approved. It is the trying of individual programming units of the application .it is finished after the consummation of a singular unit before joining.

9.1.5. FUNCTIONAL TESTING

Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (not like in white-box testing). Functional Testing usually describes what the system does. Functional testing differs from system testing in that functional testing "verifies a program by checking it against ... design document(s) or specification(s)", while system testing "validate a program by checking it against the published user or system requirements" (Kane, Falk, Nguyen 1999, p. 52). Functional testing typically involves five steps. The identification of functions that the software is expected to perform.

1. The creation of input data based on the function's specifications
2. The determination of output based on the function's specifications
3. The execution of the test case
4. The comparison of actual and expected outputs.

9.1.6. PERFORMANCE TESTING

Overall testing performed to decide how a framework acts regarding responsiveness and strength under a specific responsibility. It can likewise explore, measure, approve or confirm other quality credits of the framework, like versatility, dependability and asset use.

Execution testing is a subset of operational efficiency, an arising software engineering practice which endeavors to incorporate execution into the execution, plan and design of a framework.

9.1.7. INTEGRATION TESTING

Joining testing is a deliberate method for developing the program structure while simultaneously leading tests to uncover blunders related with. Individual modules, which are exceptionally inclined to interact mistakes, ought not be accepted to work in a split second when assembled. The issue obviously, is "assembling them"- interacting. There might be the possibilities of information lost across on another's sub capabilities, when joined may not deliver the ideal significant capability; exclusively satisfactory impression might be amplified to inadmissible levels; worldwide information designs can introduce issues.

Joining testing is the stage in programming testing wherein individual programming modules are consolidated and tried collectively. Combination testing takes as info modules have been unit tried, bunches them in bigger totals, applies tests characterized in a mix test plan to those totals, and conveys as its result the coordinated framework prepared. Every one of the blunders found in the framework are amended for the following stage.

The reason for mix testing is to confirm useful, execution, and dependability necessities put on significant plan things. These "plan things", for example collections (or gatherings of units), are practiced through their points of interaction utilizing discovery testing, achievement and mistake cases being mimicked by means of suitable boundary and information inputs. Mimicked 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 gatherings collaborate correctlyfor model across methodology calls or cycle enactments, and this is finished subsequent to testing individual modules, for example unit testing.

9.1.8. VALIDATION TESTING

Confirmation and Approval are free strategies that are utilized together for really looking 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 unbiased outsider.

It is now and again said that approval can be communicated by the question "Would you say you are building the correct thing?" and check by "Would you say you are building it right?". Practically speaking, the use of these terms changes. At times they are even utilized conversely.

9.1.9. SYSTEM TESTING

Framework testing of programming or equipment is trying directed on a total, coordinated framework to assess the framework's consistence with its predefined necessities. Framework testing falls inside the extent of black box testing, and thusly, ought to require no information on the internal plan of the code or rationale. Generally speaking, framework testing takes, as its feedback, all of the "incorporated" programming parts that have passed mix testing and furthermore the product framework itself coordinated with any material equipment system(s). The motivation behind joining testing is to identify any irregularities between the product units that are coordinated together (called collections) or between any of the gatherings and the equipment. Framework testing is a more restricted kind of testing; it looks to recognize surrenders both inside the "between gatherings" and furthermore inside the framework in general.

Framework testing is performed on the whole framework with regards to a Utilitarian Prerequisite Specification(s) (FRS) as well as a Framework Necessity Detail (SRS). Framework testing tests the plan, yet additionally the way of behaving and, surprisingly, the trusted assumptions for the client. It is additionally expected to test up to and past the limits characterized in the product/equipment necessities 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 on the off chance that it doesn't deliver the necessary result created or considered in to two different ways. One is on screen and another is printed design. The result comes as the predefined necessities by the client. Consequently yield testing brings about no remedy 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.

Readiness 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) **,**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.

[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).

[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.

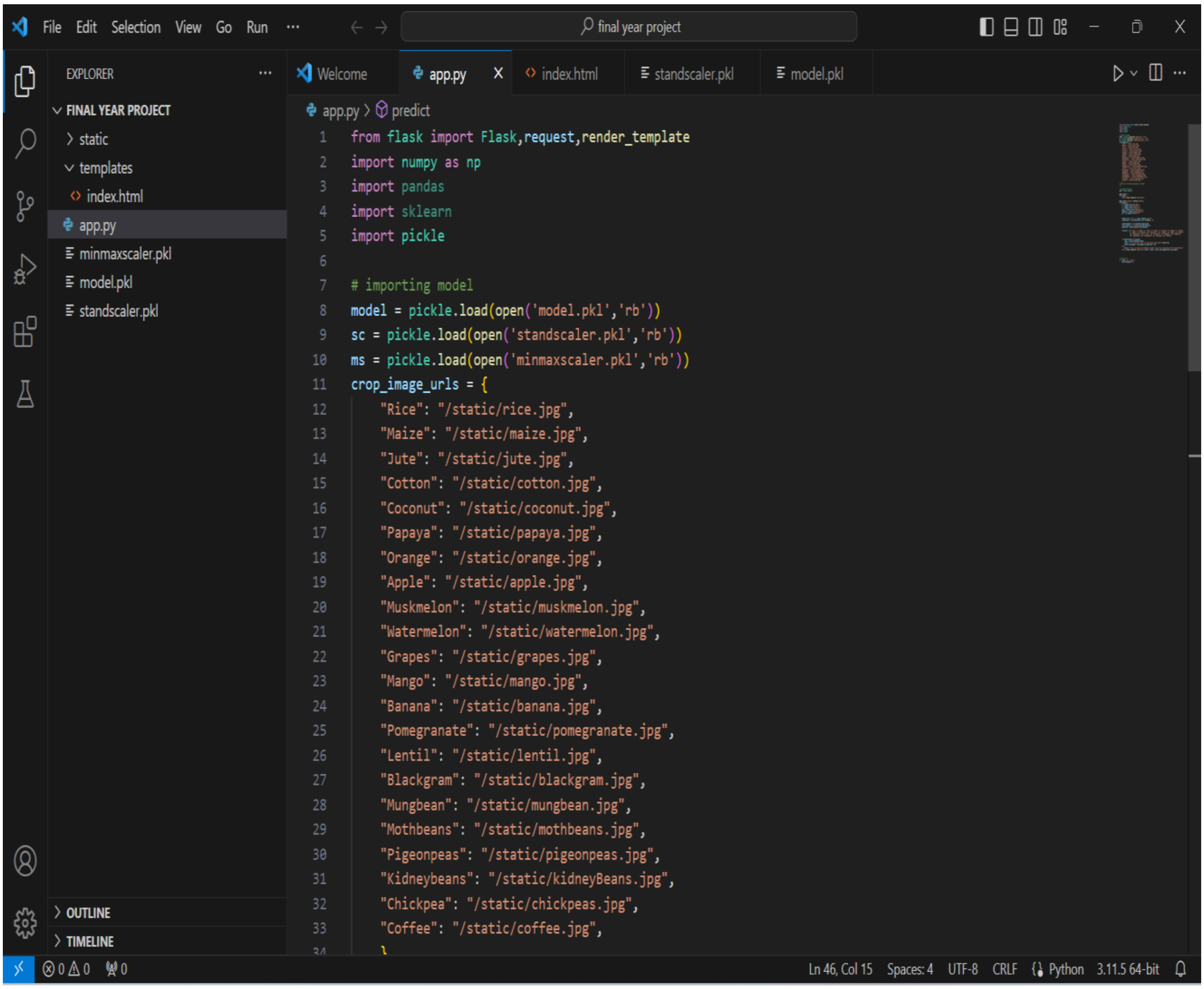
[9] H.S.Abdullahi, F.mahieddine and R.E.Sheriff,2015 Technology Impact on Agricultural Productivity: A Review of Precision Agriculture Using Unmanned Aerial Vehicles | SpringerLink.

[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. Journal of Bionic Engineering 7 (3), 276-283.

APPENDIX-A

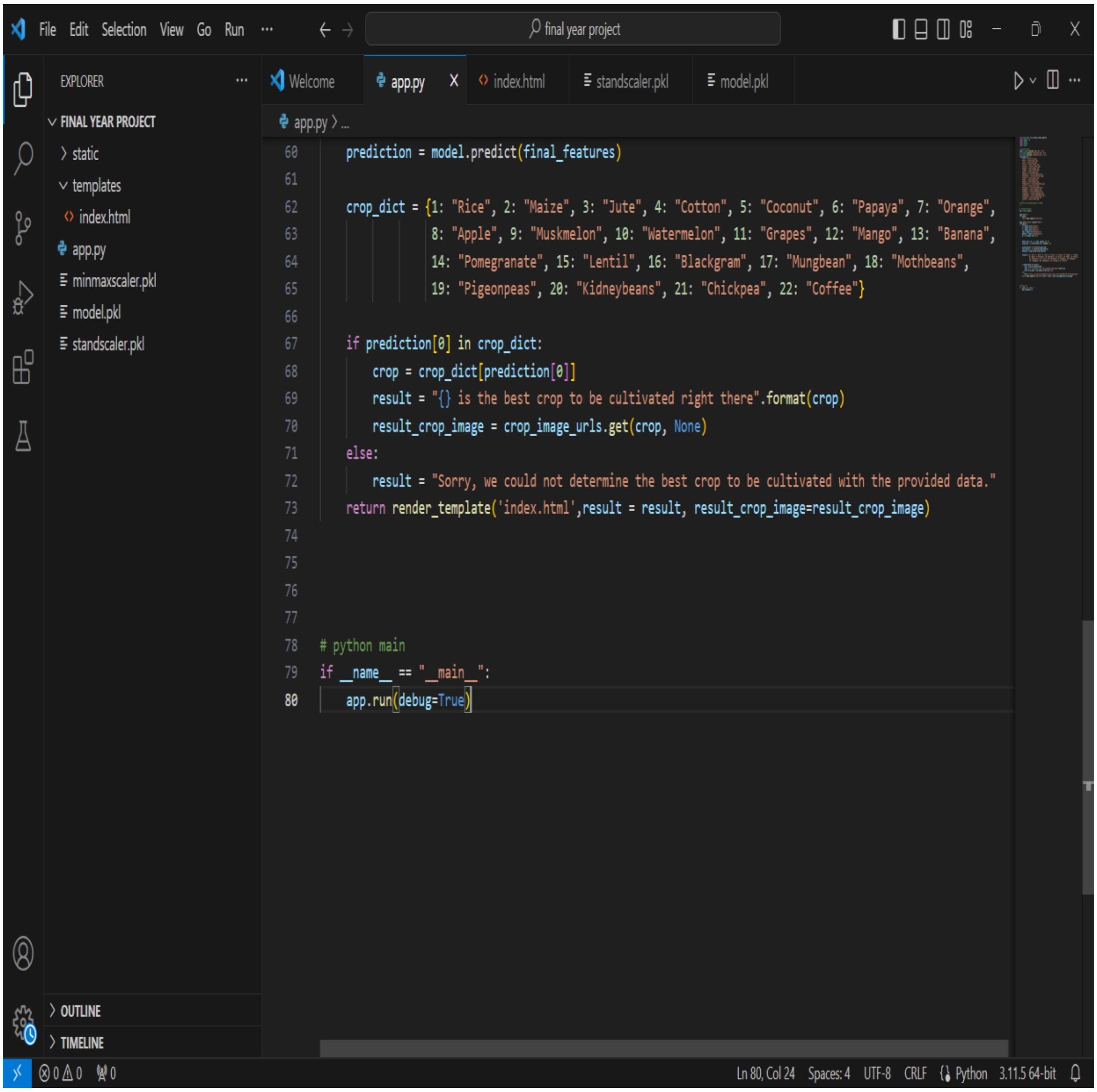
PSUEDOCODE

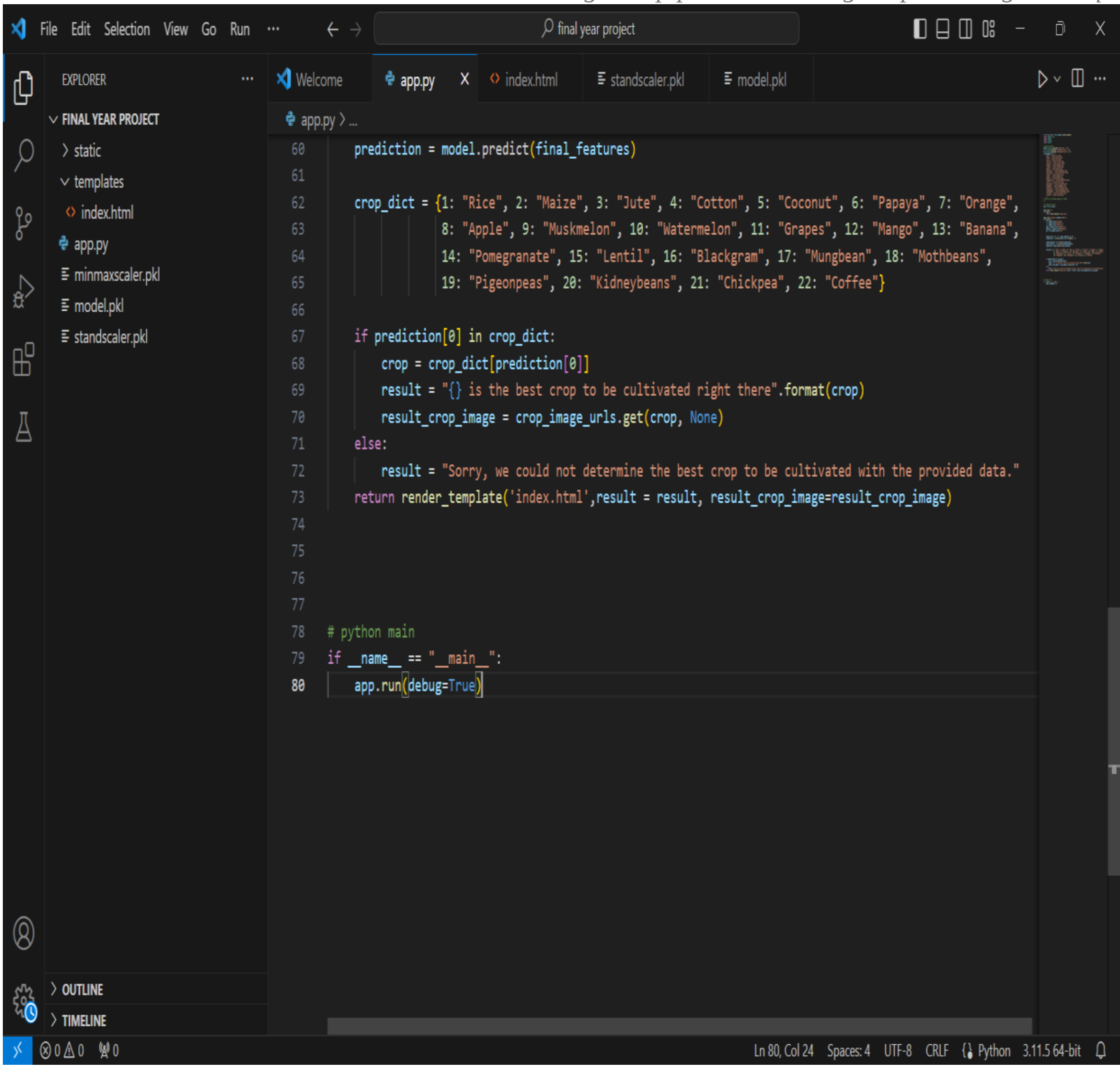
app.py(file)



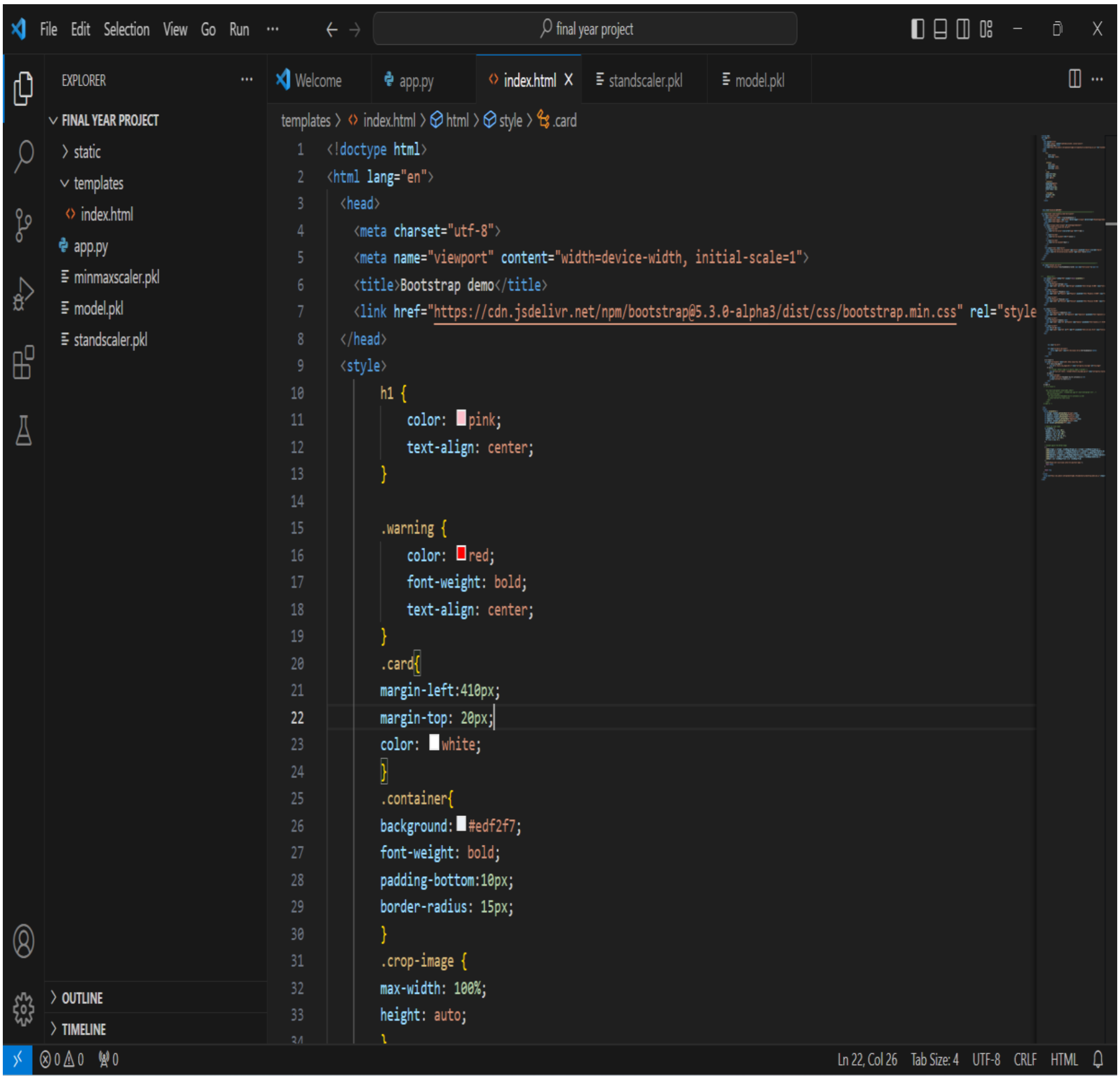
```python
from flask import Flask,request,render_template
import numpy as np
import pandas
import sklearn
import pickle

# 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",
}
```

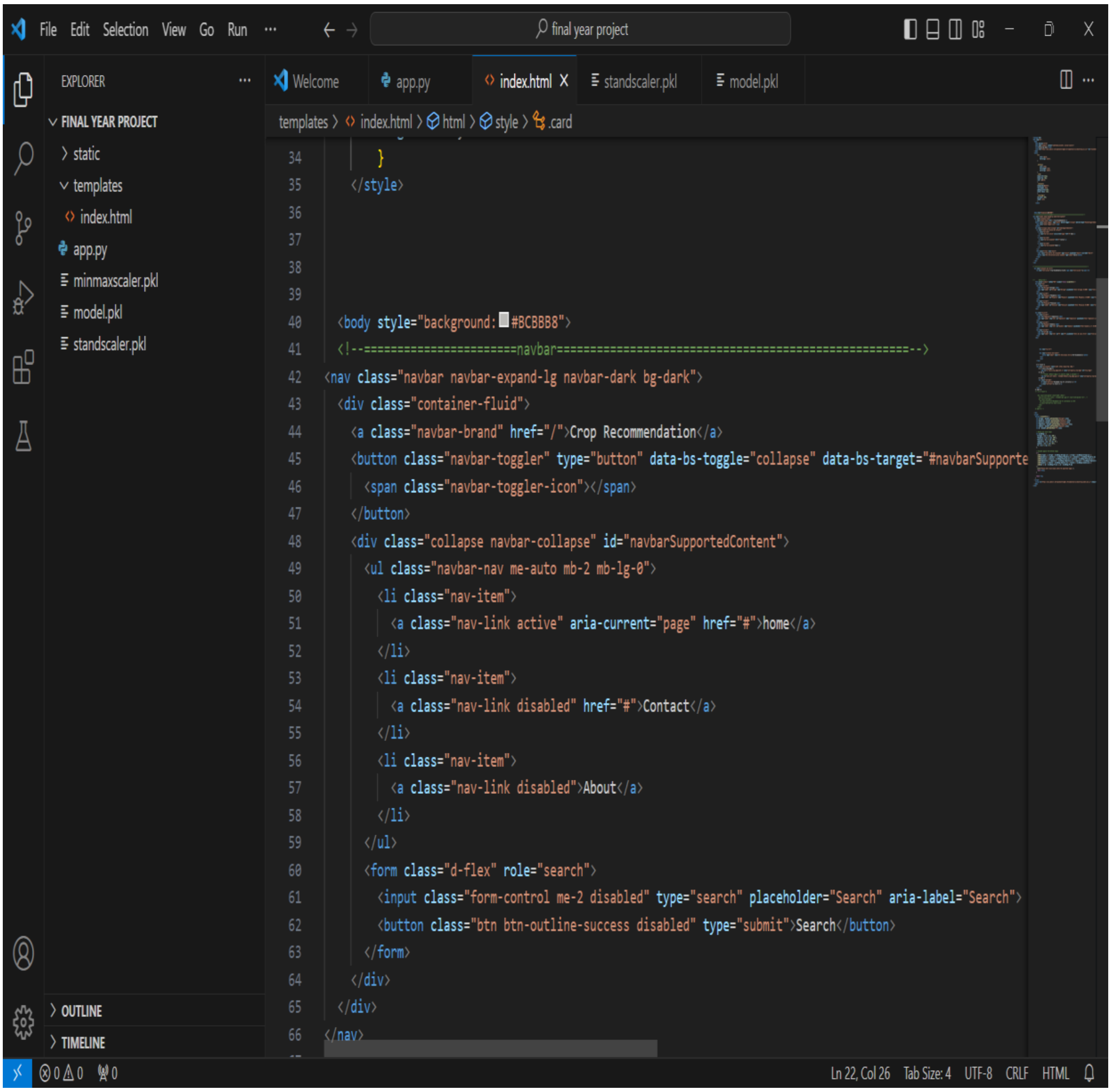
fig


File Edit Selection View Go Run ···
final year project
EXPLORER
FINAL YEAR PROJECT
static
templates
index.html
app.py
minmaxscaler.pkl
model.pkl
standscaler.pkl
Welcome
app.py
index.html
standscaler.pkl
model.pkl
app.py › ...
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)
OUTLINE
TIMELINE
Ln 80, Col 24 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit


File Edit Selection View Go Run ...
final year project
EXPLORER
FINAL YEAR PROJECT
static
templates
index.html
app.py
minmaxscaler.pkl
model.pkl
standscaler.pkl
Welcome
app.py
index.html
standscaler.pkl
model.pkl
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)
OUTLINE
TIMELINE
Ln 80, Col 24 Spaces: 4 UTF-8 CRLF Python 3.11.5 64-bit

- **index.html**


File Edit Selection View Go Run ...
final year project
EXPLORER
FINAL YEAR PROJECT
static
templates
index.html
app.py
minmaxscaler.pkl
model.pkl
standscaler.pkl
OUTLINE
TIMELINE
Welcome
app.py
index.html
standscaler.pkl
model.pkl
templates > index.html > html > style > .card
<!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="style
</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;
}
Ln 22, Col 26 Tab Size: 4 UTF-8 CRLF HTML

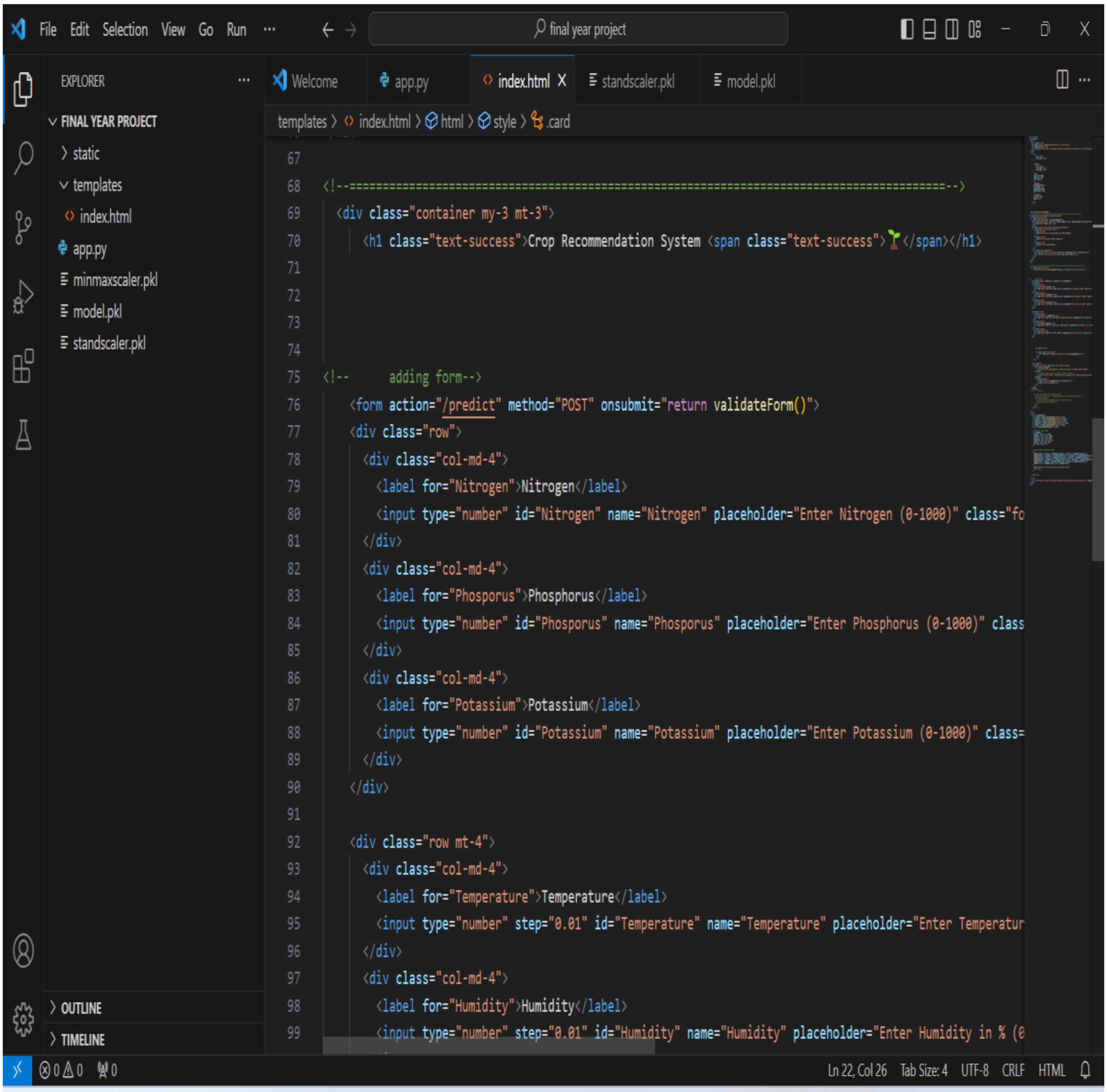

File Edit Selection View Go Run ···
final year project
EXPLORER
FINAL YEAR PROJECT
static
templates
index.html
app.py
minmaxscaler.pkl
model.pkl
standscaler.pkl
OUTLINE
TIMELINE
Welcome
app.py
index.html
standscaler.pkl
model.pkl
templates › index.html › html › style › .card
}
</style>
<body style="background: #BCBBB8">
<!--=====================navbar==-->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<div class="container-fluid">
Crop Recommendation
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupporte

</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item">
home

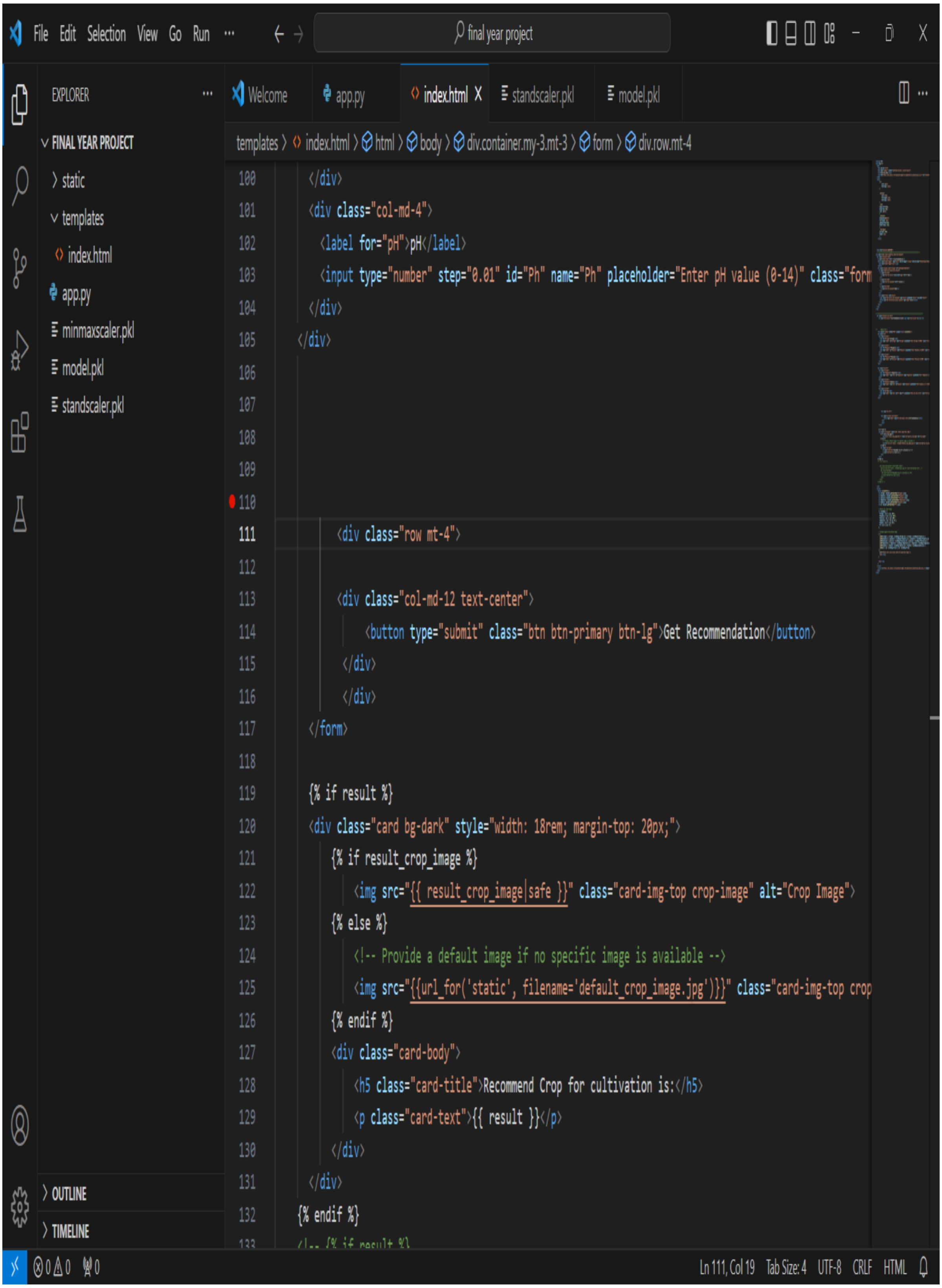
<li class="nav-item">
Contact

<li class="nav-item">
About

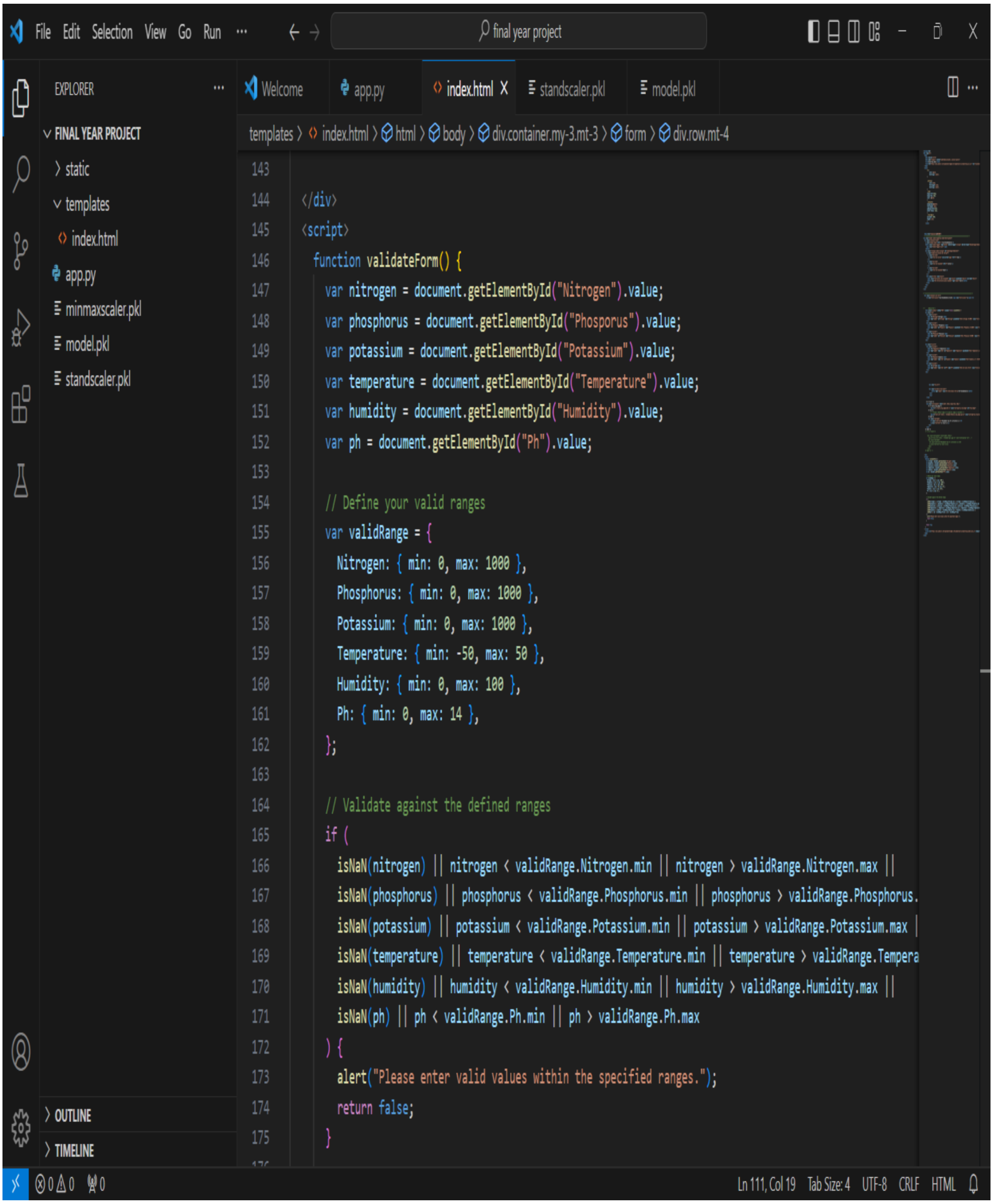
<form class="d-flex" role="search">
<input class="form-control me-2 disabled" type="search" placeholder="Search" aria-label="Search">
<button class="btn btn-outline-success disabled" type="submit">Search</button>
</form>
</div>
</div>
</nav>
Ln 22, Col 26 Tab Size: 4 UTF-8 CRLF HTML



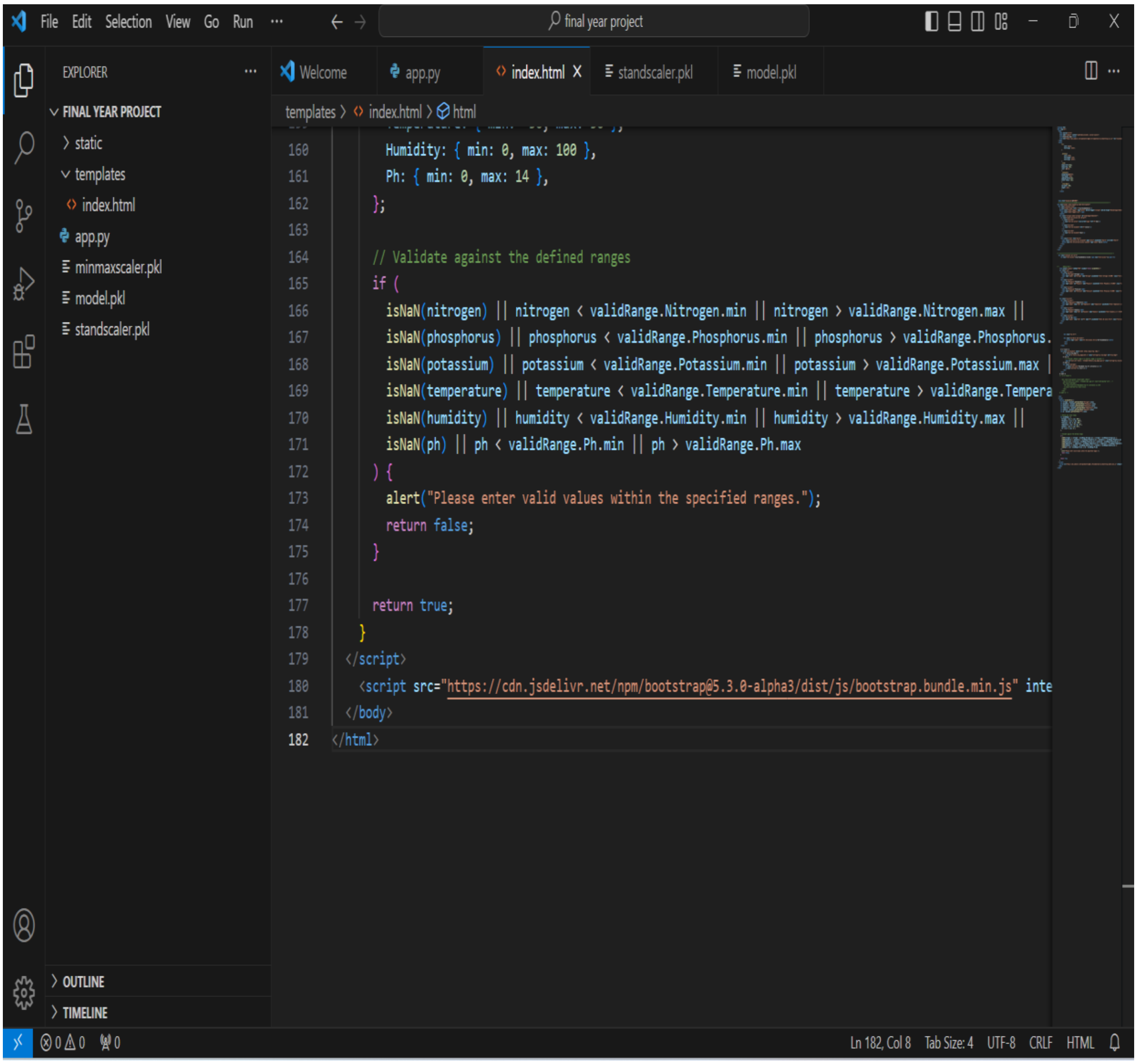
File Edit Selection View Go Run ···
final year project
EXPLORER
FINAL YEAR PROJECT
static
templates
index.html
app.py
minmaxscaler.pkl
model.pkl
standscaler.pkl
OUTLINE
TIMELINE
Welcome
app.py
index.html
standscaler.pkl
model.pkl
templates › index.html › html › style › .card
<!--===-->
<div class="container my-3 mt-3">
<h1 class="text-success">Crop Recommendation System </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="fo
</div>
<div class="col-md-4">
<label for="Phosporus">Phosphorus</label>
<input type="number" id="Phosporus" name="Phosporus" placeholder="Enter Phosphorus (0-1000)" class
</div>
<div class="col-md-4">
<label for="Potassium">Potassium</label>
<input type="number" id="Potassium" name="Potassium" placeholder="Enter Potassium (0-1000)" class=
</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 Temperatur
</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
Ln 22, Col 26 Tab Size: 4 UTF-8 CRLF HTML



File Edit Selection View Go Run
final year project
EXPLORER
FINAL YEAR PROJECT
static
templates
index.html
app.py
minmaxscaler.pkl
model.pkl
standscaler.pkl
Welcome
app.py
index.html
standscaler.pkl
model.pkl
templates > index.html > html > body > div.container.my-3.mt-3 > form > div.row.mt-4
</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
</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
{% endif %}
<div class="card-body">
<h5 class="card-title">Recommend Crop for cultivation is:</h5>
<p class="card-text">{{ result }}</p>
</div>
</div>
{% endif %}
OUTLINE
TIMELINE
Ln 111, Col 19 Tab Size: 4 UTF-8 CRLF HTML



File Edit Selection View Go Run ···
final year project
EXPLORER
FINAL YEAR PROJECT
static
templates
index.html
app.py
minmaxscaler.pkl
model.pkl
standscaler.pkl
OUTLINE
TIMELINE
Welcome
app.py
index.html
standscaler.pkl
model.pkl
templates › index.html › html › body › div.container.my-3.mt-3 › form › div.row.mt-4
</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.
isNaN(potassium) || potassium < validRange.Potassium.min || potassium > validRange.Potassium.max |
isNaN(temperature) || temperature < validRange.Temperature.min || temperature > validRange.Tempera
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;
}
Ln 111, Col 19 Tab Size: 4 UTF-8 CRLF HTML


File Edit Selection View Go Run ···
final year project
EXPLORER
FINAL YEAR PROJECT
static
templates
index.html
app.py
minmaxscaler.pkl
model.pkl
standscaler.pkl
OUTLINE
TIMELINE
Welcome
app.py
index.html
standscaler.pkl
model.pkl
templates > index.html > html
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.
isNaN(potassium) || potassium < validRange.Potassium.min || potassium > validRange.Potassium.max |
isNaN(temperature) || temperature < validRange.Temperature.min || temperature > validRange.Tempera
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" inte
</body>
</html>
Ln 182, Col 8 Tab Size: 4 UTF-8 CRLF HTML

APPENDIX-B

SCREENSHOTS

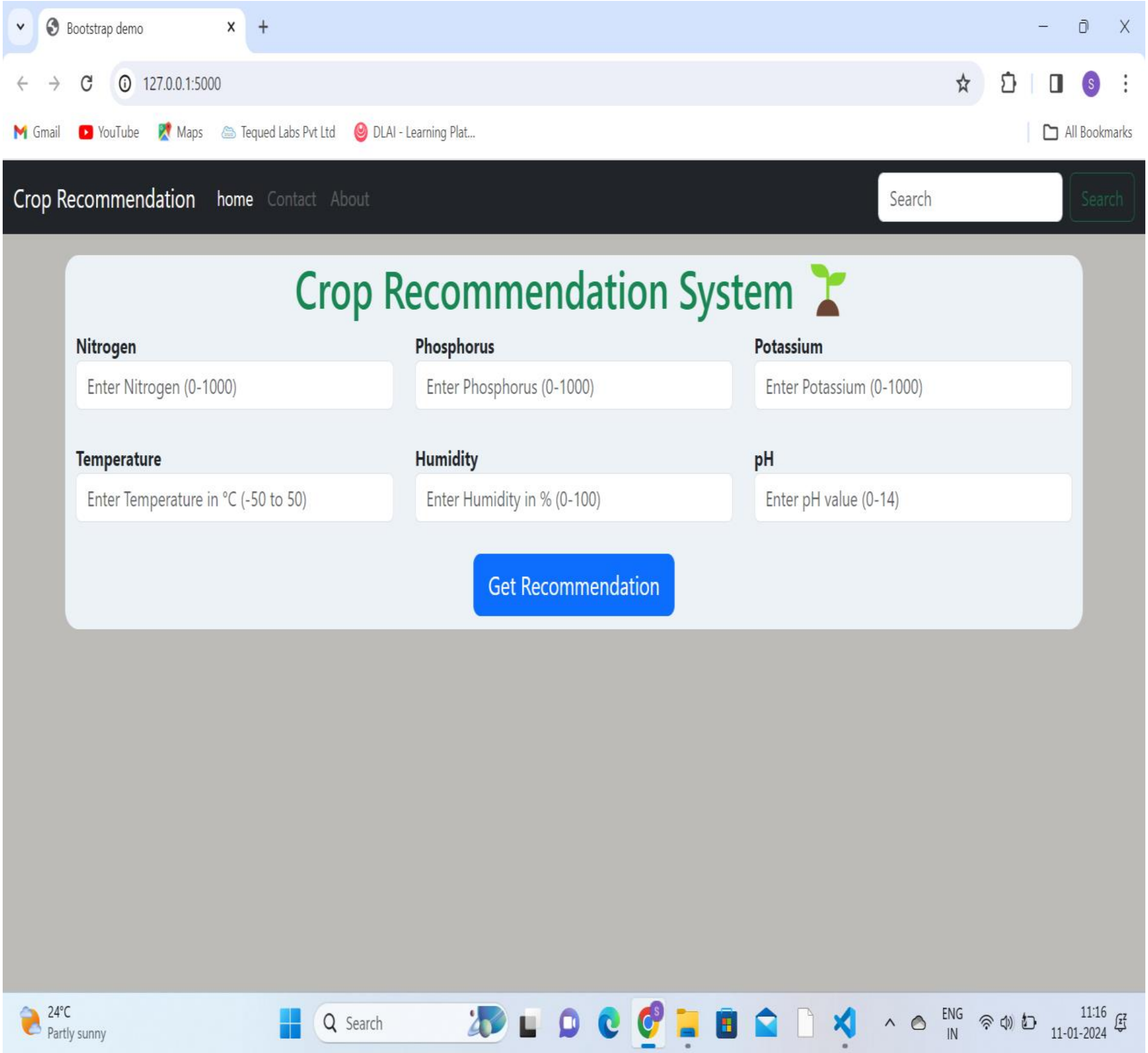


Fig 2.3: Home page

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

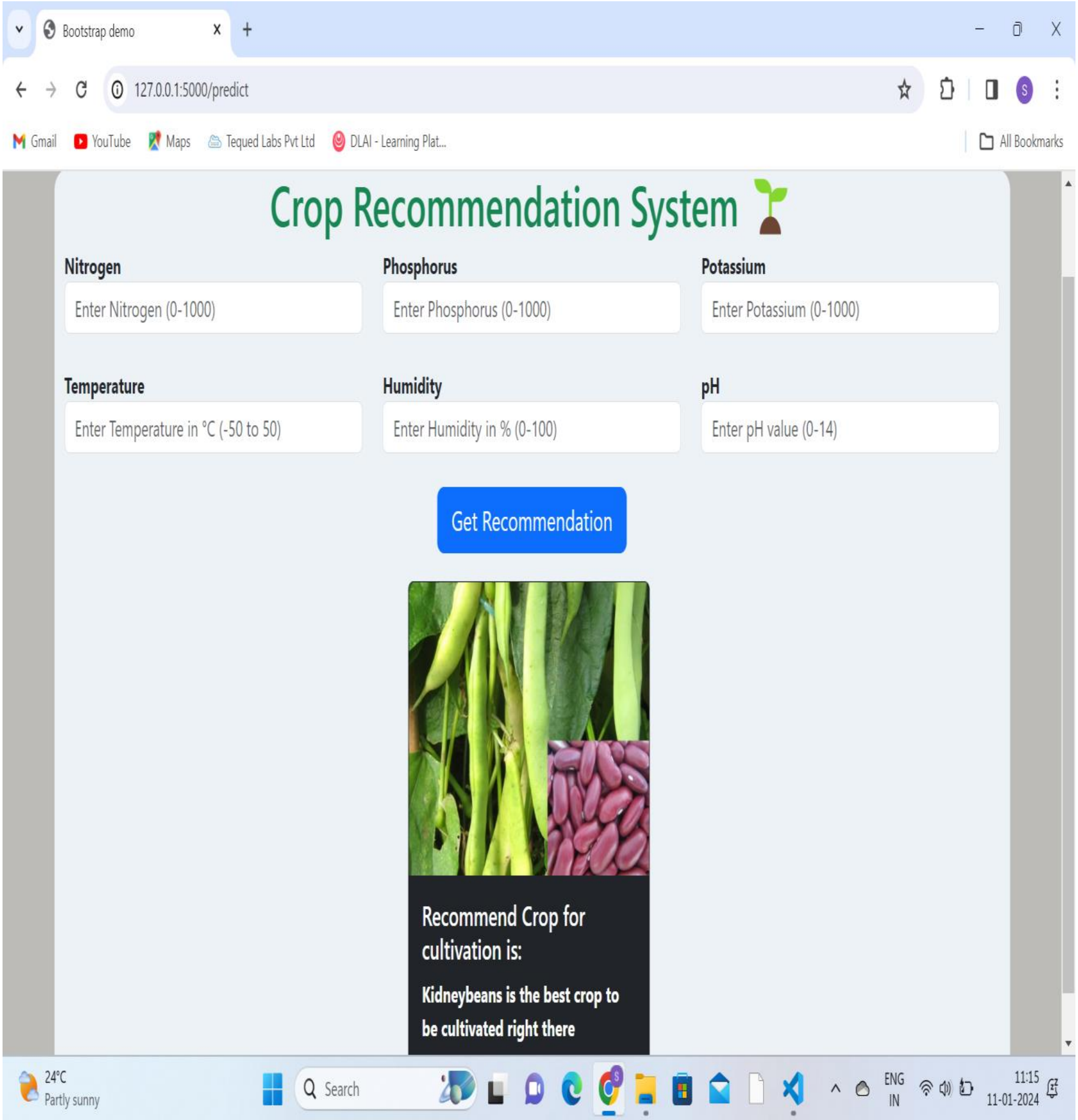


Fig 3.1: Demo 01.

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.

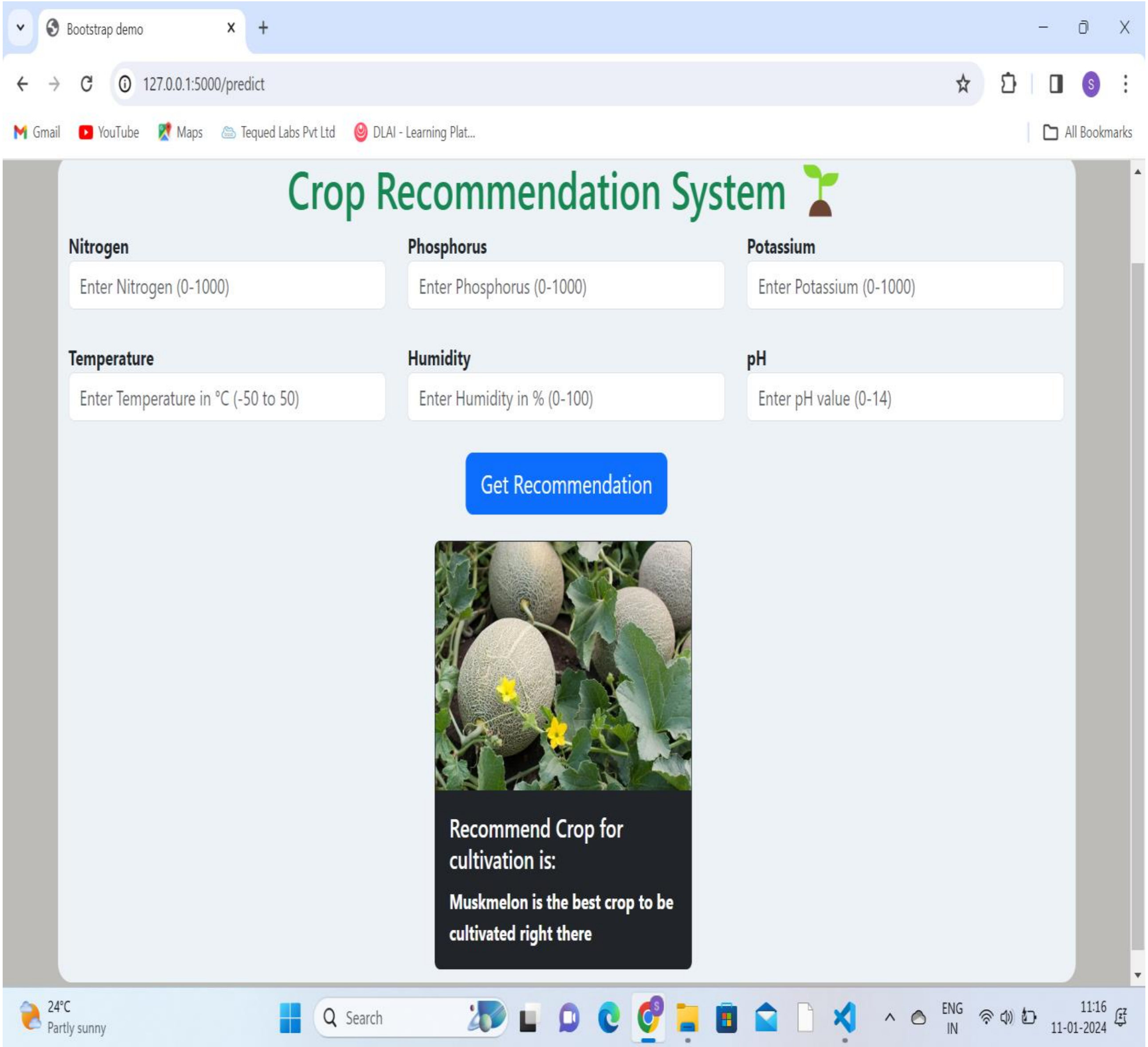


Fig 3.2: Demo 02

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.

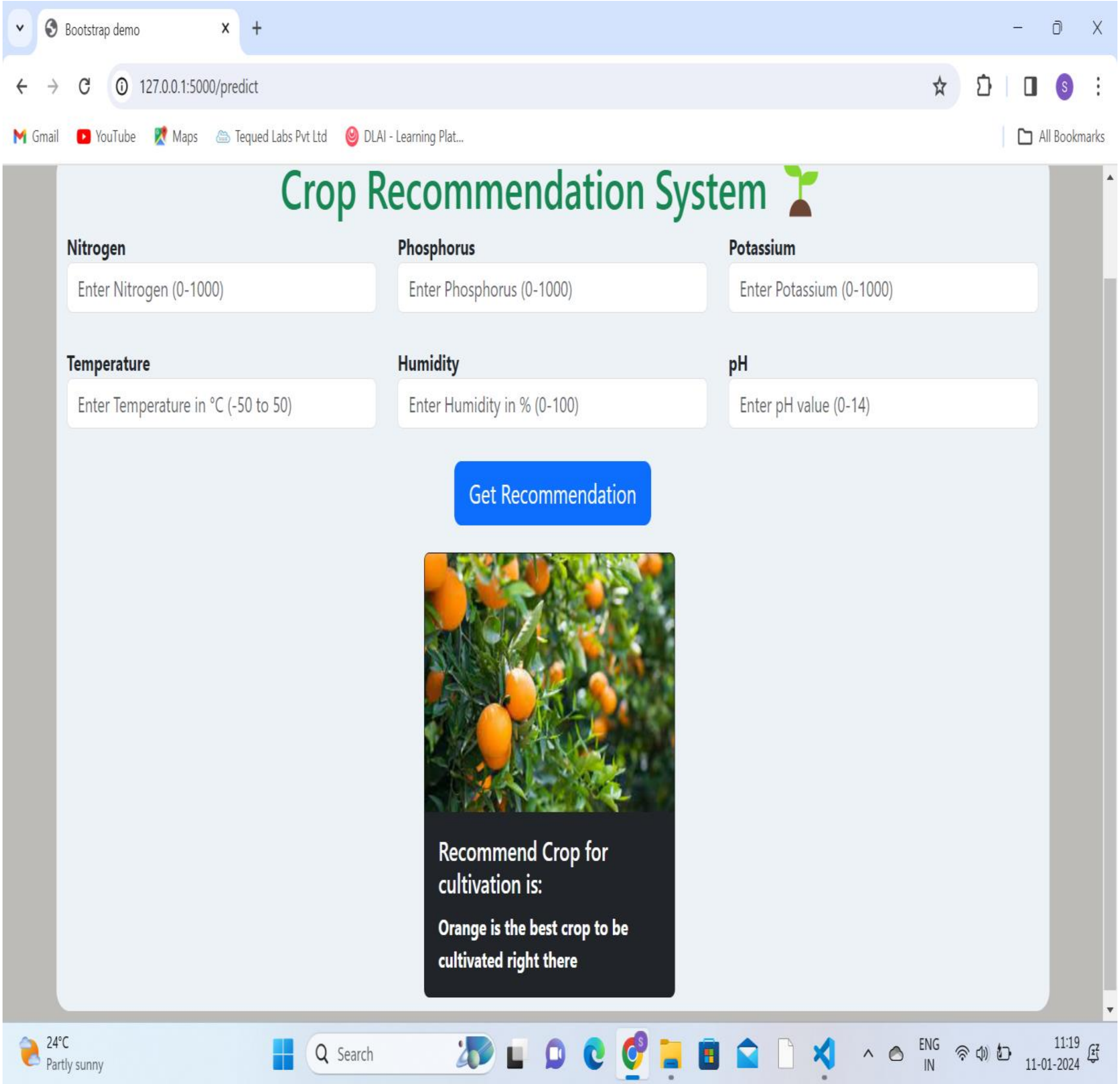


Fig 3.3: Home page output 03

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

Publication:

Dear Author, Congratulation!!!

Your manuscript with Registration ID: **IJCRT_ 249528** has been **accepted** for publication in the **International Journal of Creative Research Thoughts (IJCRT)** | **www.ijcrt.org** | **ISSN: 2320-2882 | International Peer Reviewed & Refereed Journals, Open Access Journal. IJCRT is Peer Review Journal, Refereed Journal, Peer Reviewed Journal Referred Journal and Indexed Journal Open Access Journal Online and Print Journal**

IJCRT Impact Factor: 7.97 | UGC Approved Journal No: 49023 (18)

Check your paper status: **https://ijcrt.org/track.php or http://ijcrt.org/Authorhome/alogin.php**

Online Payment Link for Indian author: http://ijcrt.org/pay_form_2.php

Online Payment Link for Foreign/international author: https://ijcrt.org/pay_form_international.php

Your Paper Review Report :	
Registration ID:	IJCRT – 249528
Title of the Paper:	**SMART FRAMING: CROP RECOMMENDATION USING DEEP LEARNING TECHNIQUES.**
Accepted or Not:	**Accepted**

Criteria	Points out of 100%
Continuity	88%
Text structure	86%
References	95%
Understanding and Illustrations	95%
Explanatory power	89%
Detailing	96%
Relevance and practical advice	89%

Track Your Paper Details:	
Paper Track Link 1:	https://ijcrt.org/track.php
Paper Track Link 2:	http://ijcrt.org/Authorhome/alogin.php
Online Payment link Details:	
Indian author:	http://ijcrt.org/pay_form_2.php
Foreign/international author:	https://ijcrt.org/pay_form_international.php

Overall Assessment (Comments.) **Paper Accepted: YES**

Unique Contents: 94% Paper Accepted

Reviewer comment store in online RMS system. Paper Accepted complete below Step1 and Step2 process and publish your paper within 1 to 2 day.

Publication of Paper	**Within 01-02 Days after Submitting documents. Please complete payment and send payment proof along with below documents.**

Track Your Paper

Use Six Digit Registration ID - XXXXXX

IJCRT_ XXXxxx -Six digit No

submit

Registration ID : IJCRT_249528

Paper Acceptance letter

Title of Paper : SMART FRAMING: CROP RECOMMENDATION USING DEEP LEARNING TECHNIQUES.
Author 1 : Pourmamilla mohith gowd
Author 2 : CHETHAN A
Author 3 : Shravana S S
Author 4 : Rashi
Author 5 : POOJARANI

Dear Author, Congratulation!!!

Your Paper Accepted Complete Payment Process and Follow below Step

Your manuscript has been **accepted for publication** in the International Journal of Creative Research Thoughts (IJCRT) | www.ijcrt.org | ISSN: 2320-2882 | International, Indexed Peer Reviewed & Refereed Journals, Open Access Journal. IJCRT is Peer Review Journal, Refereed Journal, Online and Print Journal, Scholarly open access journals, Impact factor 7.97 (Calculate by google scholar and Semantic Scholar | AI-Powered Research Tool) , Multidisciplinary, Monthly, Indexing in all major database & Metadata, Citation Generator, Digital Object Identifier(DOI)

Please Complete Step 1 and Step 2 Your Payment and Document Process After paper will Publish within 1 to 2 days.

Step 1 : Pay Publication Fees (Click Here to Pay Charges)

Your Paper Acceptance letter | Fast Publication Process

For Indian Author

Indian Author ₹1500 INR online payment Fast,
Secure and best option of payment using this paper publish within 1 day

Indian Author click here for more payment options

For International/foreign Author

International Authors 55 $ (USD) online payment Fast,
Secure and best option of payment using this paper publish within 1 day

International Authors (Foreign author) click here for more payment options

Step 2 : Fill up the following Forms and send to us.

1. Final Formatted Manuscript/Research paper (must be in DOC file format and file name should be your registration ID) (Download)
2. Undertaking by Author Form (scanned copy)(Handwritten signed by author) (Download)
3. Copy right transfer form (scanned copy) (Handwritten signed by author) (Download)
4. Identity Proof of corresponding author only(scanned copy) (election Card/PAN Card/College Identity card/Driving License)
5. Payment Proof (Scan copy/Screen shot of your payment) also mentions Paid amount, Transaction ID, and Date of Payment in the mail.

Send this all required above 5 documents to editor@ijcrt.org (Please mention your Registration ID in the subject of Email)
After sending this all documets to editor@ijcrt.org wait for 2 to 3 days for Status updation and Publication.

Registration ID : IJCRT_249528

Paper Acceptance letter

Title of Paper : SMART FRAMING: CROP RECOMMENDATION USING DEEP LEARNING TECHNIQUES.
Author 1 : Pourmamilla mohith gowd
Author 2 : CHETHAN A
Author 3 : Shravana S S
Author 4 : Rashi
Author 5 : POOJARANI
Reviewer Allotted: No
Final Paper Submitted: No
Documents Submitted: No
Proof of Student/Academician: No
Comment:

Submitted! Please Save This page For Acknowlegment for Submission of paper Save it! Your paper is Successfully submitted 2024-01-12.
Track your paper : https://ijcrt.org/track.php

Submitted! Acknowlegment for Submission of paper. Your paper is Successfully submitted 2024-01-12. wait for Acceptance within 1 to 2 day it will come.

Congratulations...!!
Thank you for submitting paper in IJCRT INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT) . You will be intimated for final selection & acceptance of your paper very soon.

Your paper will undergo the **NORMAL REVIEW PROCESS** of the Journal. The process normally takes 1 to 3 Days to complete depending on the number of rounds the reviews need to take place. Your Conformation Details of paper Submission are sent to your registered mail ID : mohithks007@gmail.com

Your paper has been successfuly submitted to IJCRT. Your Details of paper are set to your instered mail ID. Kindly Check. In case you don't find the mail in INBOX kindly check SPAM folder.

You will be intimated for final selection & acceptance of your paper very soon.

Basic details about submitted paper

Pariculars	Contains
Paper Registration ID	IJCRT_249528
Track Your Paper	Track your paper : https://ijcrt.org/track.php
Paper Title	SMART FRAMING: CROP RECOMMENDATION USING DEEP LEARNING TECHNIQUES.
Corresponding Author's Name	Pourmamilla mohith gowd
Corresponding Author's Mail ID	mohithks007@gmail.com

Click Here to Track your Paper Status

Published paper link:

https://ijcrt.org/

- **Plagiarism Report**

UP-R1

ORIGINALITY REPORT

23%	12%	7%	18%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Lovely Professional University Student Paper	3%
2	pdfcoffee.com Internet Source	2%
3	Submitted to ESC Rennes Student Paper	1%
4	Submitted to Cerritos College Student Paper	1%
5	"Sustainable Development through Machine Learning, AI and IoT", Springer Science and Business Media LLC, 2023 Publication	1%
6	Submitted to Higher Education Commission Pakistan Student Paper	1%
7	Devendra Dahiphale, Pratik Shinde, Koninika Patil, Vijay Dahiphale. "Smart Farming: Crop Recommendation using Machine Learning with Challenges and Future Ideas", Institute	1%

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.