FARM ASSISTANT USING ML AND DL

A Report submitted to the Rajiv Gandhi University of Knowledge and Technologies in partial fulfillment of the degree of

Bachelor of technology in Computer Science and Engineering

Submitted by

P Bhanuchandar(S180668)

Ch Geetha(S180402)

P Ch K NagaVinay(S180580)

3rd year BTech 2nd Semester

Under the supervision of

Sri. S Satheesh Kumar

Asst. Professor-Department of CSE

RGUKT-SRIKAKULAM

Department of Computer Science and Engineering Rajiv Gandhi University of Knowledge and Technologies, Srikakulam S.M. Puram (V), Etcherla (M), Srikakulam (Dt) – 532410

CERTIFICATE

This is to certify that the report entitled "**FARM ASSISTANT USING ML AND DL**" was submitted by Bhanuchandar Pinninti, bearing ID. No. S180668, Geetha Chikati, bearing ID. No. S180402 and Ch K NagaVinay Perisetti bearing ID. NO. S180580 in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science is a bonafide work carried out by them under my supervision and guidance.

The report has not been submitted previously in part or in full to this or any other University or Institution to award any degree or diploma.

Satheesh Kumar Sankarapu,

Project Guide,

Department of CSE,

RGUKT,SRIKAKULAM.

Sesha Kumar Nalluri,

Head of the Department,

Department of CSE,

RGUKT,SRIKAKULAM.

DECLARATION

We **Bhanuchandar Pinninti, Geetha Cheekati and Ch K Naga Vinay** hereby declare that this report entitled **“FARM ASSISTANT USING ML AND DL”** submitted by us under the guidance and supervision of **Satheesh Kumar Sankarapu** is a bonafide work.

We also declare that it has not been submitted previously in part or in full to this University or other University or Institution to award any degree or diploma.

(Bhanuchandar Pinninti)

ID NO . S180303

(Geetha Chikati)

ID NO . S180402

(Ch K NagaVinay Perisetti)

ID NO . S180572

ACKNOWLEDGEMENTS

We would like to express my sincere gratitude and appreciation to my project Guide **Sankarapu Satheesh Kumar**, for the support and guidance provided throughout the development and completion of our project titled **“FARM ASSISTANT”** and for valuable suggestions and keen interest throughout the progress of my course of research.

We are grateful to **Sesha Kumar Nalluri**, HOD CSE, for providing excellent computing facilities and a congenial atmosphere for progressing with my project.

At the outset,we would like to thank **Rajiv Gandhi University of Knowledge and Technologies, Srikakulam** for providing all the necessary resources for the successful completion of our course work. At last, but not least we thank our classmates and other students for their physical and moral support.

With Sincere Regards

Bhanuchandar Pinninti,

Geetha Chikati

Ch K Naga Vinay Perisetti.

ABSTRACT

Title: FARM ASSISTANT

The Farm Assistant is a project that aimed at developing a comprehensive precision agriculture system that incorporates Machine Learning (ML) and Convolutional Neural Networks (CNN) algorithms to provide accurate crop and pest recommendations, Leveraging the power of ML, the system analyses various data inputs such as soil quality, climate conditions, historical crop performance and suggest the best crop. Additionally, the CNN-based pest recommendation module utilizes image recognition techniques to identify and classify pests, enabling targeted pesticide recommendations. By integrating these two modules, Farm Assistant aims to optimize farming practices, minimize pesticide usage, and improve crop yields, ultimately promoting sustainable and efficient agricultural operations.

Keywords: Precision Agriculture, Machine Learning, Convolutional Neural Networks, recommendations, sustainability.

Contents

CHAPTER 1

INTRODUCTION

1.1 Introduction

Agriculture plays a vital role in the Indian economy, serving as a significant source of income for its population. Despite the relentless efforts of Indian farmers, their productivity is consistently threatened by uncontrollable natural factors. While it is impossible to change these natural forces, maximizing field output remains the best approach. Soil degradation stands out as a prominent challenge, but it can be mitigated by cultivating crops that are best suited for the specific land conditions. Another pressing issue is pest infestation, which necessitates the use of appropriate pesticides to protect crops effectively. By addressing these concerns, farmers can significantly benefit. The Indian government conducts various soil tests to assess soil quality, but unfortunately, farmers often lack knowledge about how to interpret and apply the results of these tests effectively.

1.2 Motivation

The motivation behind the Farm Assistant project is to support and empower Indian farmers who rely on agriculture as their main source of livelihood. Indian farmers face numerous challenges, including soil degradation, crop selection, pest management, and pesticide selection. The project aims to address these challenges by leveraging machine learning and data analysis techniques to provide valuable recommendations.

1.3 Problem Statement

Indian farmers face various challenges in maximizing their agricultural productivity due to factors such as soil degradation, crop selection, pest management, and pesticide selection. Although several crop recommendation systems exist, they often vary in their approach and lack comprehensive integration of relevant parameters. Additionally, there is a notable gap in the availability of a reliable and user-friendly pesticide recommendation system that considers pest identification and suggests suitable pesticides readily available in India

1.4 Objectives

Farm Assistant aims to address the mentioned problems through the following objectives:

1.4.1 Implementing Precision Agriculture:

By utilizing research data on soil characteristics, soil types, and crop yield, Farm Assistant seeks to adopt modern farming techniques. The platform provides farmers with site-specific recommendations for the right crop selection, reducing the likelihood of choosing unsuitable crops and ultimately increasing productivity.

1.4.2. Developing an Ensemble Model Recommendation System:

Farm Assistant proposes a recommendation system based on an ensemble model with a majority voting technique. This approach ensures high accuracy and efficiency in suggesting crops based on specific parameters of the farming site. By leveraging multiple models, the platform enhances the reliability of its recommendations.

1.4.3. Pest Recognition and Pesticide Recommendation:

Farm Assistant goes beyond pest recognition by recommending appropriate pesticides available in India. The platform adheres to ISO standards, ensuring that the suggested pesticides meet quality and safety requirements. By providing farmers with reliable pest management solutions, Farm Assistant helps protect crops effectively.

1.4.4. Designing a Web Application:

To achieve the above objectives, Farm Assistant is developing a user-friendly web application. This application serves as the interface through which farmers can access and benefit from the precision agriculture, crop recommendation, and pesticide recommendation services provided by the platform.

1.5 Goal

- User Friendly
- Simple fast
- Low cost and effective
- Recommendations

1.6 Scope

The scope of the Farm Assistant project includes:

1. Providing accurate crop recommendations based on site-specific parameters to improve productivity.

2. Developing a comprehensive pest recognition system with ISO-compliant pesticide recommendations.

3. Integrating and analyzing data from various sources to generate precise recommendations.

4. Designing a user-friendly web application as the primary interface for farmers.

5. Promoting adoption and awareness of the platform among Indian farmers.

6. Potential for future scalability or adaptation to address agricultural challenges in other regions.

1.7 Applications

The Farm Assistant project has applications in:

1. Crop optimization and precision agriculture.

2. Pest management and pesticide recommendations.

3. Decision support system for farmers.

4. Accessibility and knowledge sharing.

5. Promotion of agricultural sustainability.

1.8 Limitations

1. Constraints:

1.1.

Farm Assistant supports 22 crops: apple, banana, blackgram, chickpea, coconut, coffee, cotton, grapes, jute, kidney beans, lentil, maize, mango, mothbeans, mungbean, muskmelon, orange, papaya, pigeon peas, pomegranate, rice, watermelon. Hence the user will get results which best suit the land but only from these 22 crops.

1.2

The system supports 10 pests: aphids, armyworm, beetle, bollworm, earthworm, grasshopper, mites, mosquito, sawfly and stem borer, which is a constraint.

1.3

Farm Assistant offers two options for pest identification: uploading an image of the pest or manually selecting the pest from a list. If the user chooses to upload an image, any picture of a pest that is not among the 10 supported pests will display the closest resemblance from the supported pests as the result. On the other hand, if the user selects the pest manually, they can only choose from the pre-defined list of 10 pests.

1.4

When the user chooses to upload a picture for pest identification, it is essential that the picture clearly shows the pest. This ensures accurate identification and allows Farm Assistant to provide the most relevant result based on the uploaded image. Clear and visible pictures enable the system to analyze the characteristics of the pest effectively, improving the accuracy of the identification process.

CHAPTER 2

LITERATURE SURVEY

2.1 Collect Information

[1] We have taken the information from the some research papers which are relevant to the project. Following are some of the research papers.

- https://ieeexplore.ieee.org/document/9418351
- https://www.frontiersin.org/articles/10.3389/fpls.2022.839572/full

2.2 Study

Key features:

The key features of the Farm Assistant project are:

1. Accurate crop and pesticide recommendations based on site-specific parameters.

2. Pest identification through image upload or manual selection.

3. User-friendly web application interface.

4. Integration and analysis of data from various sources.

5. Compliance with ISO standards for pesticide recommendations.

6. Focus on adoption and awareness among farmers.

7. Scalability and adaptability to address agricultural challenges in different regions.

2.3 Summary

In our application, with all these features we build an user friendly exchange portal between students, establishes communication and provide student necessary items at low cost. It provides sustainability, reusability, secure exchange.

CHAPTER 3

ANALYSIS OF EXISTING AND PROPOSED SYSTEMS

3.1 Existing System

There are so many existing models that recommends crops and pesticides that uses low accuracy algorithms.

3.2 Disadvantages

Although several crop recommendation systems exist, they often vary in their approach and lack comprehensive integration of relevant parameters. Additionally, there is a notable gap in the availability of a reliable and user-friendly pesticide recommendation system that considers pest identification and suggests suitable pesticides readily available in India.

3.3 Proposed System

"Farm Assistant" has two different modules. Methodology for all the modules will be discussed one by one.

3.3.1 Crop Recommendation

This module can be implemented in four steps as discussed below

Step 1: Data Acquisition

Dataset can be acquired from kaggle.

Step 2: Values Input

Users are expected to input the site specific parameters like: N, P, K (all of them in %),

temperature (in °C), relative humidity (in %), rainfall (in mm) and pH.

Step 3: ML Model Training and creating .pkl file

Recommendation system is based on the ensemble model with majority voting technique. The

constituent models are:

1. SVM
2. Random Forest
3. Naive Bayes
4. KNN

After the model is trained, a .pkl file is created.

Step 4: Crop Recommendation.pkl file is loaded to recommend the crop based on input.

3.3.2 Pesticide Recommendation

This module can be implemented in four steps as discussed below

Step 1: Data Acquisition

Dataset will be collected from internet sources.

Step 2: Data Cleaning and Data Augmentation

The data collected internet to be cleaned manually to get rid of non-useful content e.g: In case of scraping images of pest named “beetle” there are also few images of “car called

beetle”. Later on, the dataset needs to be augmented so as to increase variability.

Step 3: DL Model Creation

This involves model configuration, training configuration and model evaluation. Later on, .h5

file will be created to store the model.

Step 4: Pest Identification and corresponding Pesticide Recommendation

.h5 model will be loaded to identify the pest, later on based on the result, corresponding pesticide

will be recommended the pesticides.

3.4 Advantages

1. Better Crop Selection: Farmers can choose the most suitable crops for their land, leading to increased productivity and reduced risk of crop failure.

2. Improved Pest Management: The project helps farmers effectively deal with pests by identifying them and suggesting the right pesticides to use, minimizing crop damage and yield losses.

3. Time and Cost Savings: Farmers save time and money by receiving quick and accurate recommendations, eliminating the need for extensive research and trial and error.

4. Informed Decision-Making: Farm Assistant provides reliable information and insights, helping farmers make informed decisions about their farming practices.

5. Easy to Use: The user-friendly web application makes it simple for farmers to access and navigate the platform, regardless of their technical expertise.

6. Adoption of Best Practices: Farmers are encouraged to follow precision agriculture techniques and use recommended pesticides that meet quality and safety standards.

Overall, the Farm Assistant project benefits farmers by improving crop selection, pest management, saving time and costs, aiding decision-making, being user-friendly, promoting best practices, and sharing valuable knowledge.

3.5 System Requirements

3.5.1 Software Requirements:

- HTML
- CSS
- JAVASCRIPT
- PYTHON
- FLASK

3.5.2 Hardware Requirements:

- Ram : 4 GB above
- Windows: 10
- Hard Disk : 500 above

CHAPTER 4

SYSTEM DESIGN

4.1 DESIGN OF THE SYSTEM

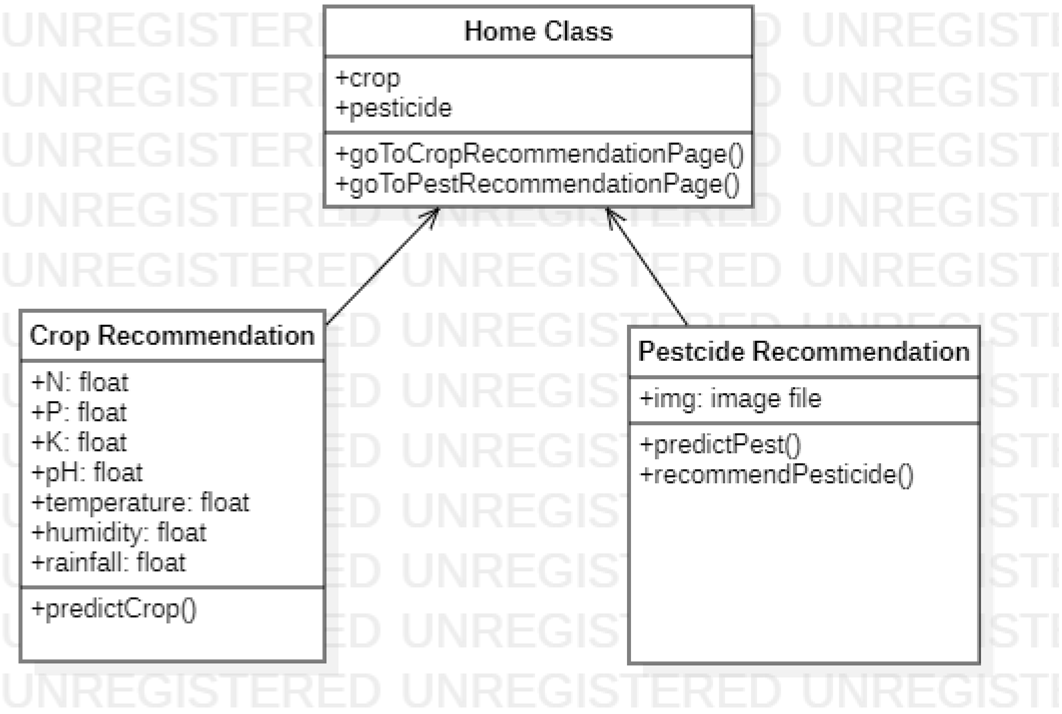
Unified Modelling Language (UML) was created in 1995 by using merging diagramming conventions used by three application development methodologies: OMT by James Rumbaugh, Objector y by Invar Jacobson and the Brooch procedure by using Grady Brooch. Previous to this time, these three amigos, together with a few dozen other practitioners had promoted competing methodologies for systematic program development, each and every with its possess system of diagramming conventions. One purpose of UML was once to slash the proliferation of diagramming techniques by way of standardizing on a original modelling language, as a result facilitating verbal exchange between builders.

It performed that goal in 1997 when the (international) Object administration team (OMG) adopted it as a commonplace. Some critics don’t forget that UML is a bloated diagramminglanguage written by means of a committee. That said, I do not forget it to be the nice mannerto be had today for documenting object-oriented program progress. It has been and is fittingmore and more utilized in industry and academia. Rational Rose is a PC Aided program Engineering (CASE) software developed by way of the Rational organization underneath the course of Brooch, Jacobson and Rumbaugh to support application progress using UML.

Rational Rose is always complex due to its mission of wholly supporting UML. Furthermore, Rational Rose has countless language extensions to Ada, C++, VB, Java, J2EE, and many others. Rational Rose supports ahead and reverse engineering to and from these langue ages.

4.1.1 Class Diagram:

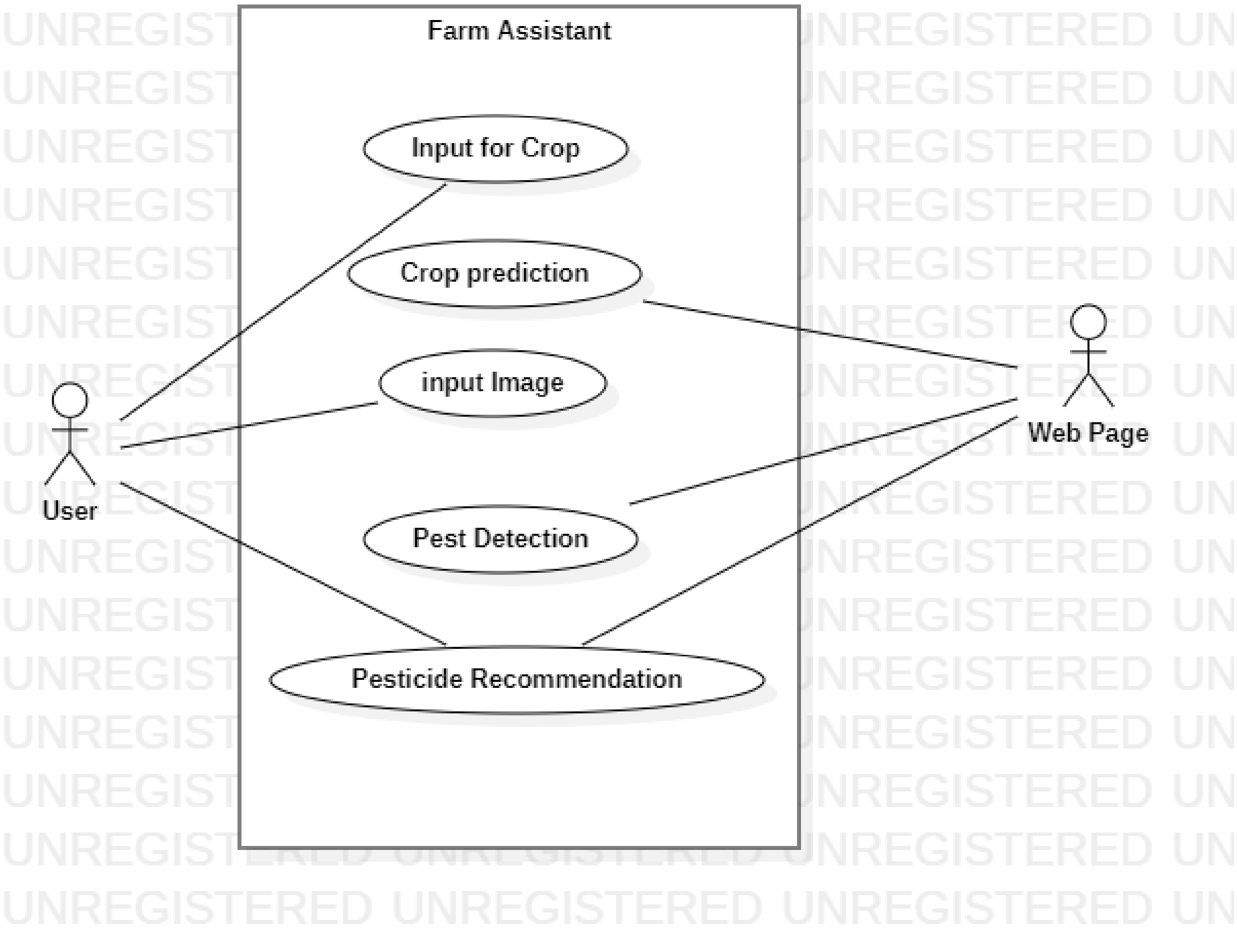
Class diagram in the Unified Modelling Language (UML), is a kind of static structure diagram that describes the constitution of a process through showing the system's classes, their attributes, and the relationships between the class. The motive of a class diagram is to depict the classes within a model. In an object-oriented software, classes have attributes (member variables), operations (member capabilities) and relation.



4.1.2 UseCase Diagram:

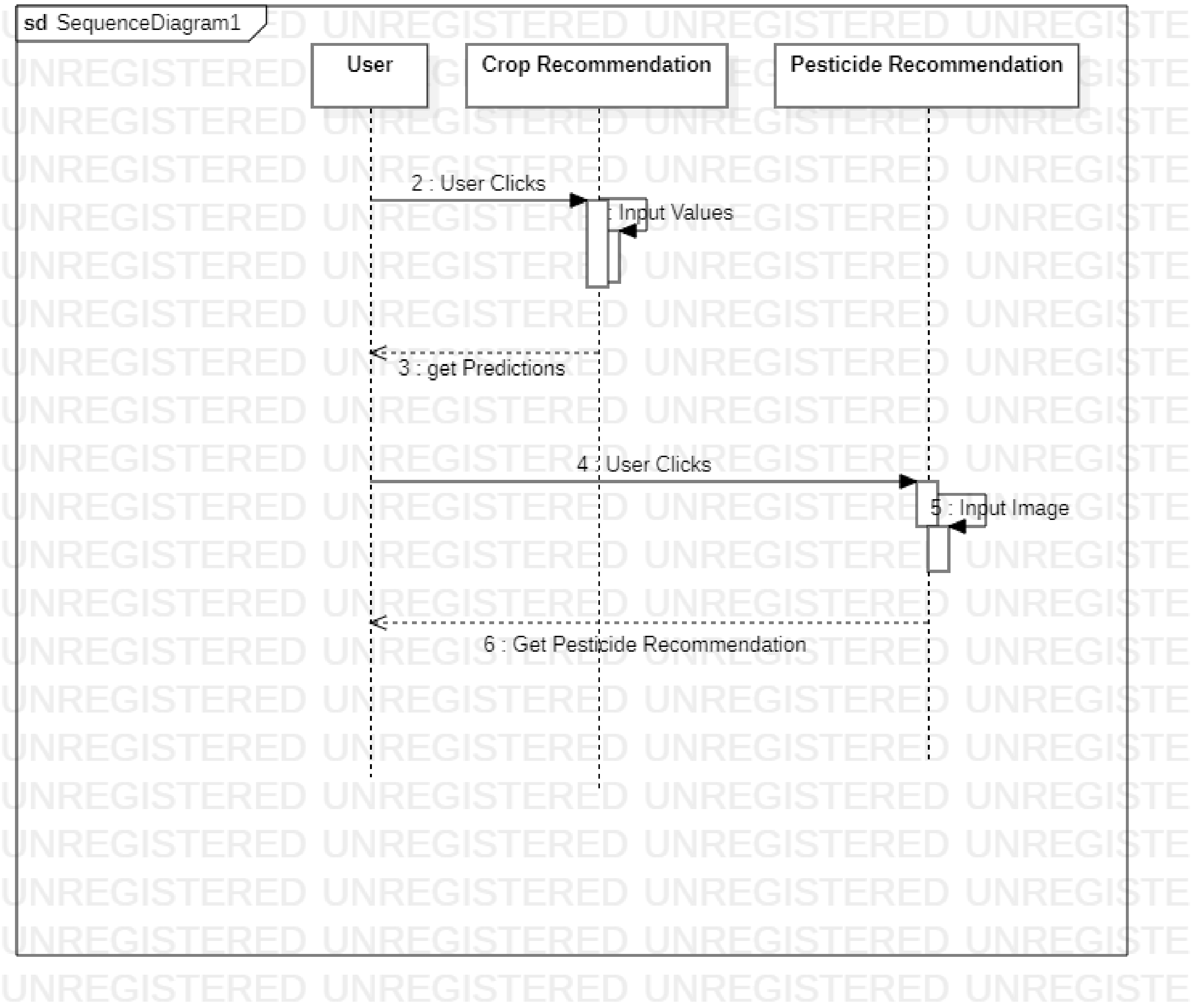
It is a visually representation what happens when actor interacts with system. A use case diagram captures the functional aspects of a system.

The system is shown as a rectangle with name of the system inside ,the actor are shown as stick figures, the use case are shown as solid bordered ovals labeled with name of the use case and relationships are lines or arrows between actor and use cases. Symbols used in Usecase are as follows-



4.1.3 Sequence Diagram:

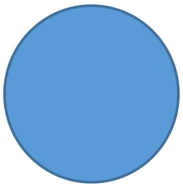
A sequence diagram in Unified Modelling Language (UML) is one variety of interaction diagram that suggests how methods operate with one other and in what order. It is a constructof a Message Sequence Chart. Sequence diagrams are quite often referred to as event-hint diagrams, event situations, and timing diagrams. A sequence diagram suggests, as parallel vertical traces (lifelines), special systems or objects that are residing at the same time, and, as horizontal arrows, the messages exchanged between them, within the order the place they occur.



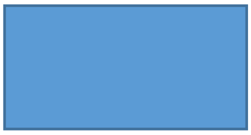
4.1.4 DFD Diagrams:

A data flow diagram or bubble chart (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFD's can also be used for the visualization of data processing (structured design).

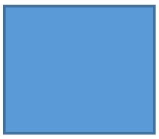
A DFD shows what kinds of information will be input to and output from the system, wherethe data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart). The primitive symbols used for constructing DFD's are: Symbols used in DFD.



A circle represents a process.

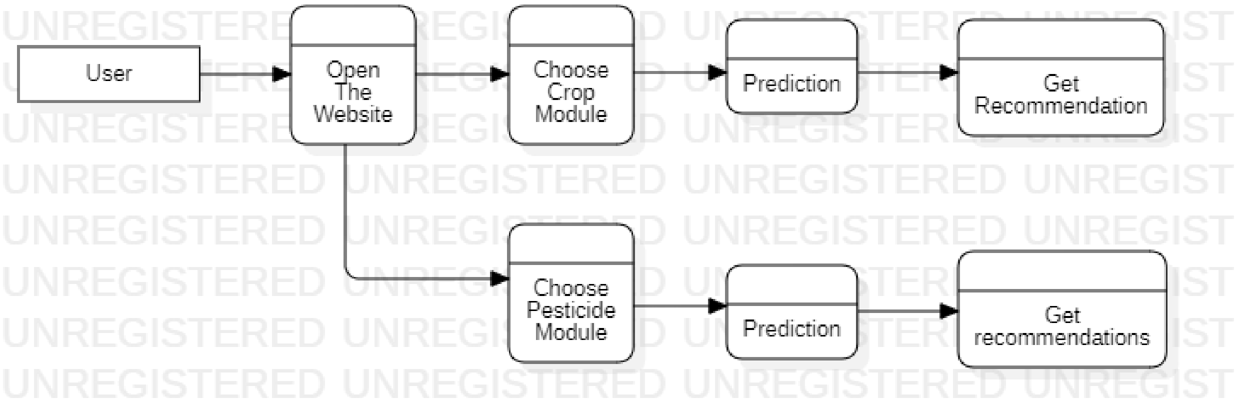


A rectangle represents external entities.



A square defines a source or destination of the system data.

An arrow identifies data flow.


User
Open The Website
Choose Crop Module
Prediction
Get Recommendation
Choose Pesticide Module
Prediction
Get recommendations

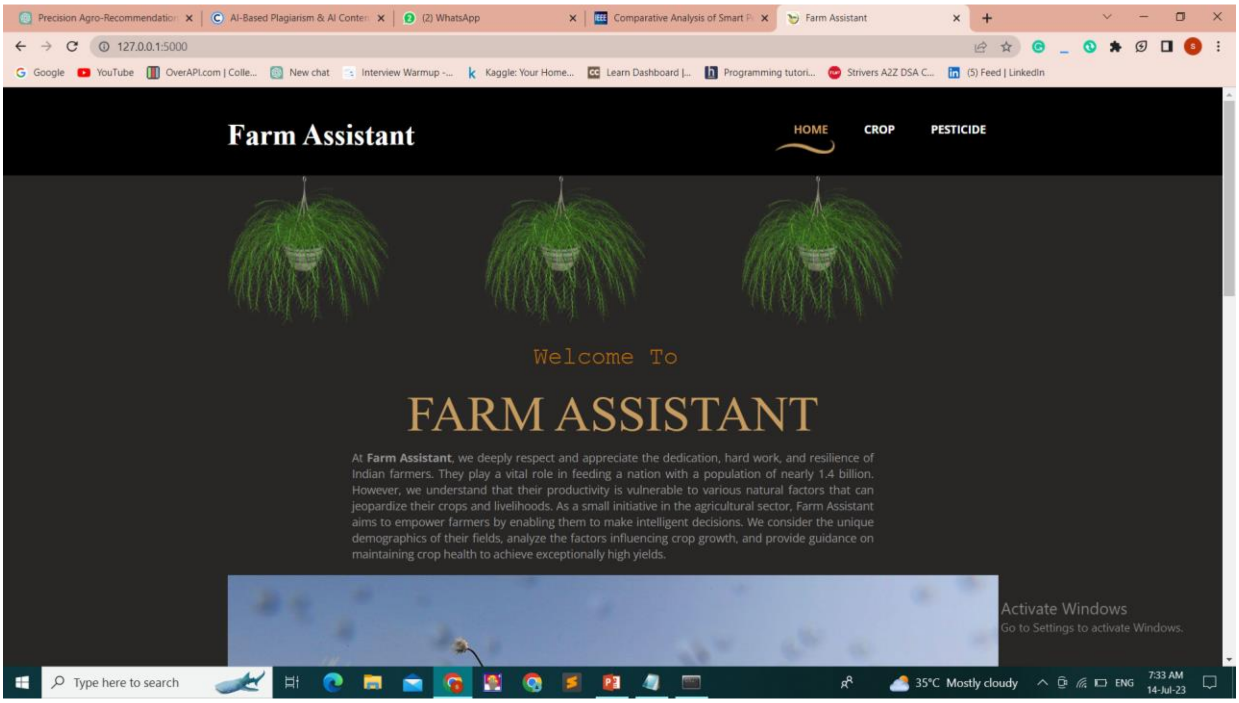
CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 Farm Assistant Web Application

It is done by using hypertext markup language, CSS, Javascript and python framework Flask.

5.1.1 Home Page

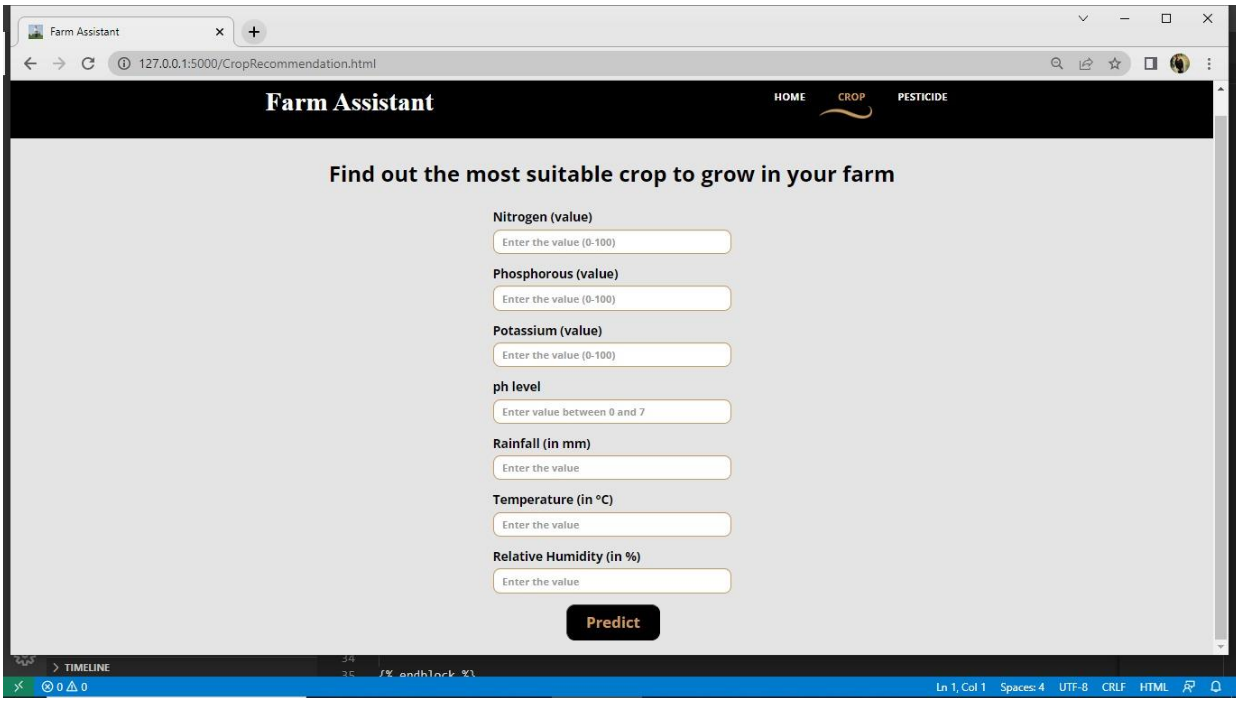


Welcome To

FARM ASSISTANT

At **Farm Assistant**, we deeply respect and appreciate the dedication, hard work, and resilience of Indian farmers. They play a vital role in feeding a nation with a population of nearly 1.4 billion. However, we understand that their productivity is vulnerable to various natural factors that can jeopardize their crops and livelihoods. As a small initiative in the agricultural sector, Farm Assistant aims to empower farmers by enabling them to make intelligent decisions. We consider the unique demographics of their fields, analyze the factors influencing crop growth, and provide guidance on maintaining crop health to achieve exceptionally high yields.

5.1.2 Crop Recommendation Page:



Find out the most suitable crop to grow in your farm

Nitrogen (value)
Enter the value (0-100)

Phosphorous (value)
Enter the value (0-100)

Potassium (value)
Enter the value (0-100)

ph level
Enter value between 0 and 7

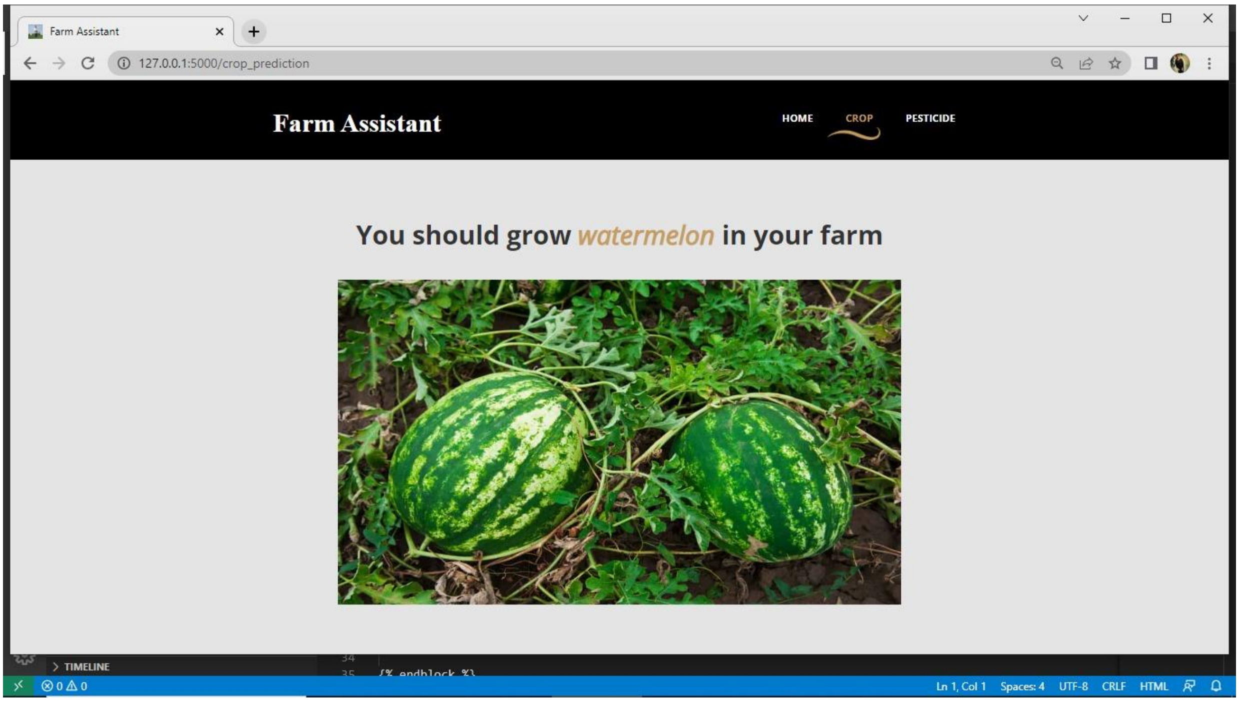
Rainfall (in mm)
Enter the value

Temperature (in °C)
Enter the value

Relative Humidity (in %)
Enter the value

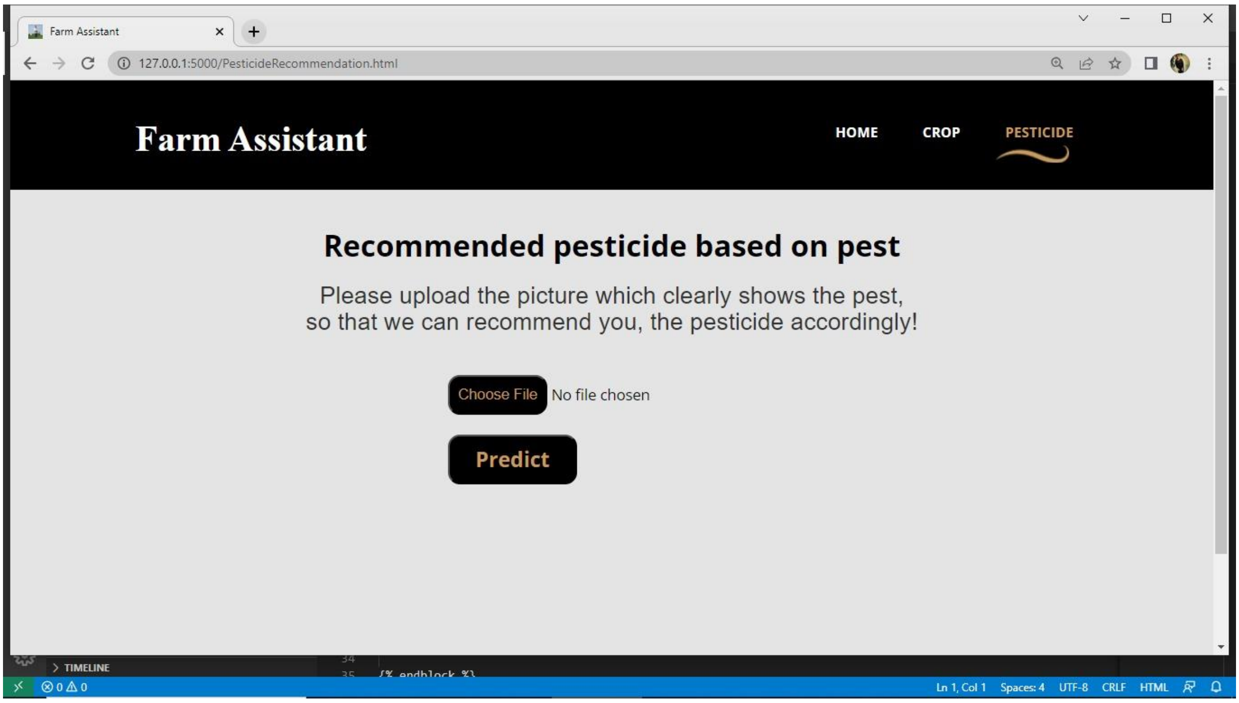
Predict

5.1.3 Crop Result



You should grow *watermelon* in your farm

5.1.4 Pest Detection



Farm Assistant

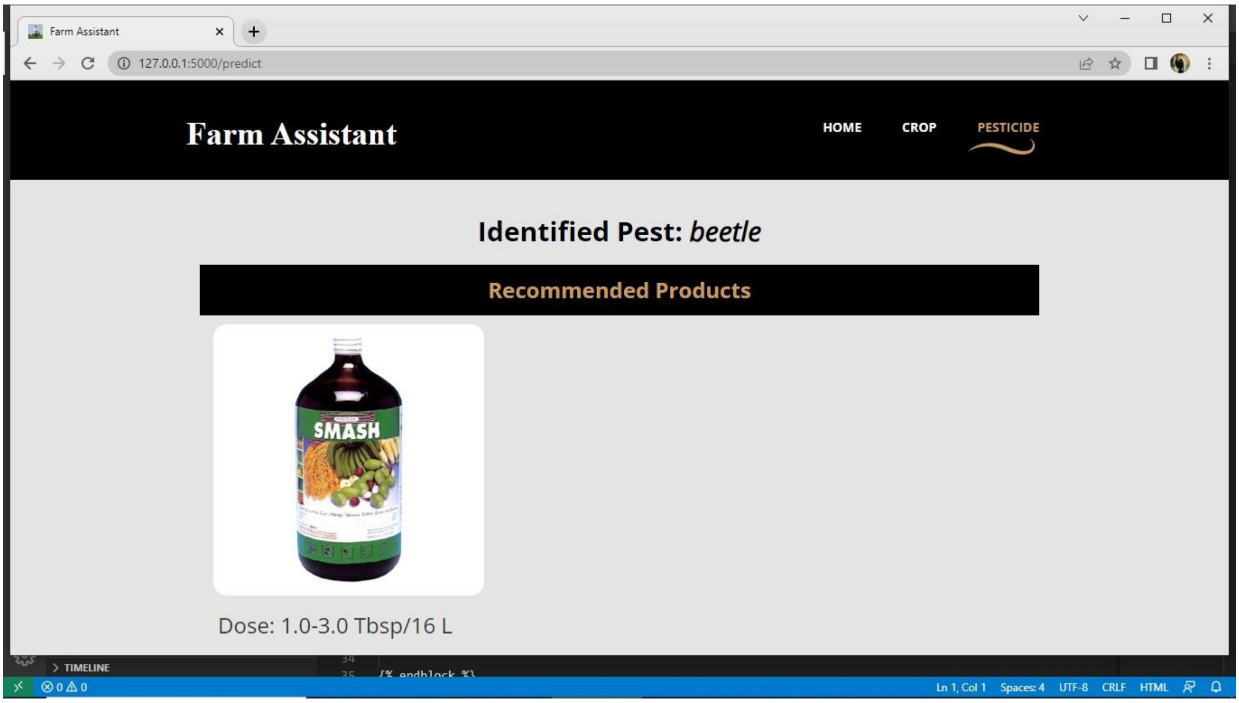
Recommended pesticide based on pest

Please upload the picture which clearly shows the pest,
so that we can recommend you, the pesticide accordingly!

Choose File No file chosen

Predict

5.1.5 Pest recommendation-result



Farm Assistant

Identified Pest: *beetle*

Recommended Products

Dose: 1.0-3.0 Tbsp/16 L

CHAPTER 5

SOURCE CODE

6.1 Home Page:

```
{% extends "layout.html" %}

{% block nav %}
<li><a href="index.html" class="active">Home</a></li>
<li><a href="CropRecommendation.html">Crop</a></li>
<li><a href="PesticideRecommendation.html">Pesticide</a></li>
{% endblock %}

{% block body %}
<section class="tm-welcome-section">
    <div class="container tm-position-relative">
      <div class="tm-lights-container">
        <img src="{{ url_for('static', filename='img/light.png') }}" alt="Light" class="light light-
1" height="188px" width="200px">
        <img src="{{ url_for('static', filename='img/light.png') }}" alt="Light" class="light light-
2" height="188px" width="200px">
        <img src="{{ url_for('static', filename='img/light.png') }}" alt="Light" class="light light-
3" height="188px" width="200px">
        <br>
      </div>
      <div class="row tm-welcome-content">
        <br><br>
        <h2 style="font-family: 'Courier New', Courier, monospace;color: #b36c09;">
          &nbsp;Welcome To&nbsp;&nbsp;</h2>
        <h2 class="gold-text tm-welcome-header-2">Farm Assistant</h2>
        <p class="gray-text tm-welcome-description">At <b>Farm Assistant</b>, we deeply
respect and appreciate the dedication, hard work, and resilience of Indian farmers. They play a
vital role in feeding a nation with a population of nearly 1.4 billion. However, we understand
that their productivity is vulnerable to various natural factors that can jeopardize their crops and
livelihoods. As a small initiative in the agricultural sector, Farm Assistant aims to empower
farmers by enabling them to make intelligent decisions. We consider the unique demographics
of their fields, analyze the factors influencing crop growth, and provide guidance on
maintaining crop health to achieve exceptionally high yields.</p>
        <br><br><br>
      </div>
      <img src="{{ url_for('static', filename='img/home.jpg') }}" alt="Table Set" class="tm-
table-set img-responsive" width="100%">
    </div>
  </section>
  <div class="tm-main-section light-gray-bg">
```

```
    <div class="container" id="main">
      <section class="tm-section tm-section-margin-bottom-0 row">
        <div class="col-lg-12 tm-section-header-container">
          <h2 style="font-family: 'Courier New', Courier, monospace;">Our Services</h2>
          <div class="tm-hr-container">
            <hr class="tm-hr">
          </div>
        </div>
        <div class="col-lg-12 tm-popular-items-container">
          <div class="tm-popular-item">
            <img src="{{ url_for('static', filename='img/popular-2.jpg') }}" alt="Popular" class="tm-popular-item-img" width="286px" height="166px">
            <div class="tm-popular-item-description">
              <h3 style="font-family: 'Courier New', Courier, monospace;font-weight: bold;">Crop Recommendation</h3>
              <hr class="tm-popular-item-hr">
              <p>
                Make an informed decision about which crop to cultivate by considering the ideal conditions. Ensure your crop selection is perfectly suited to the prevailing conditions before embarking on cultivation.</p>
              <div class="order-now-container">
                <a href="CropRecommendation.html" class="order-now-link tm-handwriting-font">Proceed Now</a>
              </div>
            </div>
          </div>
          <div class="tm-popular-item">
            <img src="{{ url_for('static', filename='img/popular-1.jpg') }}" alt="Popular" class="tm-popular-item-img" width="286px" height="166px">
            <div class="tm-popular-item-description">
              <h3 style="font-family: 'Courier New', Courier, monospace;font-weight: bold;">Pesticide Recommendation</h3>
              <hr class="tm-popular-item-hr">
              <p>Protect your crops from potential damage caused by pests by using Irrigreat. Discover the ideal pesticide to control pests and mitigate the threat they pose.</p>
              <div class="order-now-container">
                <a href="PesticideRecommendation.html" class="order-now-link tm-handwriting-font">Proceed Now</a>
              </div>
            </div>
          </div>
        </div>
      </section>
    </div>
  </div>

<footer>
    <div class="tm-black-bg">
```

```
      <div class="container">
        <div class="row tm-copyright">
          <p class="col-lg-12 small copyright-text text-center">Farmer Digital Assistance </p>
        </div>
      </div>
    </div>
  </footer>

{% endblock %}
```

6.2 Layout Page

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Farm Assistant</title>

  <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,300,400italic,600,700'
rel='stylesheet'
    type='text/css'>
   <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Recursive:wght@500&display=swap"
rel="stylesheet">
  <link href='http://fonts.googleapis.com/css?family=Damion' rel='stylesheet' type='text/css'>
  <link href="{{ url_for('static', filename='css/bootstrap.min.css') }}" rel="stylesheet">
  <link href="{{ url_for('static', filename='css/font-awesome.min.css') }}" rel="stylesheet">
  <link href="{{ url_for('static', filename='css/templatemo-style.css') }}" rel="stylesheet">
  <link rel="shortcut icon" href="{{ url_for('static', filename='img/favicon.ico') }}"
type="image/x-icon" />
  <script type="text/JavaScript" src="{{ url_for('static', filename='js/cities.js') }}"></script>
  <style>
    html body {
      background-color: #e4e4e4;
  }
  </style>
</head>

<body>
  <!-- Preloader -->
  <div id="loader-wrapper">
    <div id="loader"></div>
    <div class="loader-section section-left"></div>
    <div class="loader-section section-right"></div>
  </div>
  <!-- End Preloader -->
  <div class="tm-top-header">
    <div class="container">
      <div class="row">
        <div class="tm-top-header-inner">
          <div class="tm-logo-container">

            <h1 style="font-family: 'Times New Roman', Times, serif;font-weight: bold;">Farm
Assistant</h1>
          </div>
          <div class="mobile-menu-icon">
            <i class="fa fa-bars"></i>
          </div>
          <nav class="tm-nav">
            <ul>
              {% block nav %}
              {% endblock %}
            </ul>
          </nav>
        </div>
      </div>
    </div>
  </div>
  {% block body %} {% endblock %}
  <!-- JS -->
  <script type="text/javascript" src="{{ url_for('static', filename='js/jquery-1.11.2.min.js')
}}"></script> <!-- jQuery -->
  <script type="text/javascript" src="{{ url_for('static', filename='js/templatemo-script.js')
}}"></script> <!-- Templatemo Script -->

</body>

</html>
```

6.3 Crop Recommendation Page

```
{% extends "layout.html" %}

{% block nav %}
<li><a href="index.html">Home</a></li>
<li><a href="CropRecommendation.html" class="active">Crop</a></li>
<li><a href="PesticideRecommendation.html">Pesticide</a></li>
```

```
{% endblock %}

{% block body %}
<h2 style="text-align: center; margin: 0px; color: black">
  <br>
  <b>Find out the most suitable crop to grow in your farm</b>
</h2>
<br />
<div class="light-gray-bg">
<div
  style="
    width: 350px;
    height: 50rem;
    margin: 0px auto;
    color: black;
    border-radius: 25px;
    padding: 10px 10px;
  ">
  <form method="POST" action="/crop_prediction">
    <div class="form-group">
      <label for="Nitrogen" style="font-size: 17px"><b>Nitrogen (ratio)</b></label>
      <input
        type="number"
        class="form-control"
        id="Nitrogen"
        max = "100"
        min = "0"
        name="nitrogen"
        placeholder="Enter the value between 0 and 100"
        style="font-weight: bold"
        required
      />
    </div>
    <div class="form-group">
      <label for="Phosphorous" style="font-size: 17px"
        ><b>Phosphorous (ratio)</b></label>
      <input
        type="number"
        class="form-control"
        id="Phosphorous"
        max = "100"
        min = "0"
        name="phosphorous"
        placeholder="Enter the value (example:50)"
        style="font-weight: bold"
        required
      />
    </div>

    <div class="form-group">
      <label for="Potassium" style="font-size: 17px"><b>Potassium (ppm)</b></label>
      <input
        type="number"
        class="form-control"
        id="Potassium"
        name="potassium"
        max = "100"
        min = "0"
        placeholder="Enter the value (example:50)"
        style="font-weight: bold"
        required
      />
    </div>
    <div class="form-group">
      <label for="ph" style="font-size: 17px"><b>ph level</b></label>
      <input
        type="number"
        class="form-control"
        id="ph"
        name="ph"
        max = "14"
        min = "0"
        step=".01"
        placeholder="Enter value between 0 and 14 "
        style="font-weight: bold"
        required
      />
    </div>
    <div class="form-group">
      <label for="Rainfall" style="font-size: 17px"><b>Rainfall (in mm)&nbsp;</b></label>
      <input
        type="number"
        class="form-control"
        id="Rainfall"
        name="rainfall"
        min = "0"
        step="0.01"
        placeholder="Enter the value"
        style="font-weight: bold"
        required/>
    </div>

    <div class="form-group">
      <label for="Temperature" style="font-size: 17px"><b>Temperature (in °C)</b></label>
      <input
        type="number"
        class="form-control"
        id="Temperature"
        max = "100"
        min = "0"
        step="0.01"
        name="temperature"
        placeholder="Enter the value"
        style="font-weight: bold"
        required/>
    </div>

    <div class="form-group">
      <label for="Humidity" style="font-size: 17px"><b>Relative Humidity (in %)</b></label>
      <input
```

```
        type="number"
        class="form-control"
        id="Humidity"
        max = "100"
        min = "0"
        step="0.01"
        name="humidity"
        placeholder="Enter the value"
        style="font-weight: bold"
        required/>
    </div>


    <div class="d-flex justify-content-center">
      &emsp;&emsp;&emsp;&emsp;&emsp;&emsp;&emsp;
      <button type="submit"
        style="color: #c79c60; font-weight: bold;
         background-color: black;
         width: 130px; height:50px; border-radius:12px; font-size: 21px;">
        Predict
      </button>
      <br><br>
    </div>
  </form>
</div>
  </div>

{% endblock %}
```

6.4 Pesticide Recommendation

```
{% extends "layout.html" %}

{% block nav %}
<li><a href="index.html">Home</a></li>
<li><a href="CropRecommendation.html">Crop</a></li>
<li><a href="PesticideRecommendation.html" class="active">Pesticide</a></li>
{% endblock %}

{% block body %}
<br /><br />
<h2 style="text-align: center; margin: 0px; color: black">
  <b>Recommended pesticide based on pest</b>
</h2>
<h3 align="center">Please upload the picture which clearly shows the pest,<br>
            so that we can recommend you, the pesticide accordingly!</h3>
<br />

<div
  style="
```

```
    width: 350px;
    height: 40rem;
    margin: 0px auto;
    color: black;
    border-radius: 25px;
    padding: 10px 10px;
  "
>
    <section>
            <form action="/predict" method="post" enctype="multipart/form-data"
onsubmit="showloading()">
                <input type="file" name="image" class="upload" style="font-size: 15px;">
                <br>
                <input type="submit" value="Predict" style="color: #c79c60; font-weight: bold;
                background-color: black;
                width: 130px; height:50px; border-radius:12px; font-size: 21px;">
            </form>
    </section>
</div>
{% endblock %}
```

6.5 Flask App Page

```
from flask import Flask, render_template, request
import os
import numpy as np
from keras.preprocessing import image
from keras.models import load_model
import pickle

classifier = load_model('Trained_model.h5')
classifier._make_predict_function()

crop_recommendation_model_path = 'Crop_Recommendation.pkl'
crop_recommendation_model = pickle.load(open(crop_recommendation_model_path, 'rb'))

app = Flask(__name__)


def pred_pest(pest):
    try:
        test_image = image.load_img(pest, target_size=(64, 64))
        test_image = image.img_to_array(test_image)
        test_image = np.expand_dims(test_image, axis=0)
        result = classifier.predict_classes(test_image)
        return result
    except:
        return 'None'

@app.route("/")
@app.route("/index.html")
def index():
    return render_template("index.html")

@app.route("/CropRecommendation.html")
def crop():
    return render_template("CropRecommendation.html")

@app.route("/PesticideRecommendation.html")
def pesticide():
    return render_template("PesticideRecommendation.html")


@app.route("/predict", methods=['GET', 'POST'])
def predict():
    if request.method == 'POST':
        file = request.files['image']
        filename = file.filename

        file_path = os.path.join('static/user uploaded', filename)
        file.save(file_path)

        pred = pred_pest(pest=file_path)
        if pred == 'None':
            return render_template('unaptfile.html')
        if pred[0] == 0:
            pest_identified = 'aphids'
        elif pred[0] == 1:
            pest_identified = 'armyworm'
        elif pred[0] == 2:
            pest_identified = 'beetle'
        elif pred[0] == 3:
            pest_identified = 'bollworm'
        elif pred[0] == 4:
            pest_identified = 'earthworm'
        elif pred[0] == 5:
            pest_identified = 'grasshopper'
        elif pred[0] == 6:
            pest_identified = 'mites'
        elif pred[0] == 7:
            pest_identified = 'mosquito'
        elif pred[0] == 8:
            pest_identified = 'sawfly'
        elif pred[0] == 9:
            pest_identified = 'stem borer'


        return render_template(pest_identified + ".html",pred=pest_identified)

@ app.route('/crop_prediction', methods=['POST'])
def crop_prediction():
    if request.method == 'POST':
        N = int(request.form['nitrogen'])
        P = int(request.form['phosphorous'])
        K = int(request.form['potassium'])
        ph = float(request.form['ph'])
        rainfall = float(request.form['rainfall'])
        temperature = float(request.form['temperature'])
        humidity = float(request.form['humidity'])
        data = np.array([[N, P, K, temperature, humidity, ph, rainfall]])
        my_prediction = crop_recommendation_model.predict(data)


        final_prediction = my_prediction[0]
        return render_template('crop-result.html', prediction=final_prediction,
pred='img/crop/'+final_prediction+'.jpg')

if __name__ == '__main__':
    app.run(debug=True)
```

6.6 Crop Recommendation.py

```
from sklearn.model_selection import train_test_split
import pandas as pd
from sklearn.svm import SVC
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn import model_selection

crop = pd.read_csv('Data/crop_recommendation.csv')
X = crop.iloc[:,:-1].values
Y = crop.iloc[:,-1].values

X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.15)

models = []
models.append(('SVC', SVC(gamma ='auto', probability = True)))
models.append(('svm1', SVC(probability=True, kernel='poly', degree=1)))
models.append(('svm2', SVC(probability=True, kernel='poly', degree=2)))
models.append(('svm3', SVC(probability=True, kernel='poly', degree=3)))
models.append(('svm4', SVC(probability=True, kernel='poly', degree=4)))
models.append(('svm5', SVC(probability=True, kernel='poly', degree=5)))
models.append(('rf',RandomForestClassifier(n_estimators = 21)))
models.append(('gnb',GaussianNB()))
models.append(('knn1', KNeighborsClassifier(n_neighbors=1)))
models.append(('knn3', KNeighborsClassifier(n_neighbors=3)))
models.append(('knn5', KNeighborsClassifier(n_neighbors=5)))
models.append(('knn7', KNeighborsClassifier(n_neighbors=7)))
models.append(('knn9', KNeighborsClassifier(n_neighbors=9)))

vot_soft = VotingClassifier(estimators=models, voting='soft')
vot_soft.fit(X_train, y_train)
y_pred = vot_soft.predict(X_test)

scores = model_selection.cross_val_score(vot_soft, X_test, y_test,cv=5,scoring='accuracy')
print("Accuracy: ",scores.mean())

score = accuracy_score(y_test, y_pred)
print("Voting Score % d" % score)

import pickle
pkl_filename = 'Crop_Recommendation.pkl'
Model_pkl = open(pkl_filename, 'wb')
pickle.dump(vot_soft, Model_pkl)
Model_pkl.close()
```

6.7 Pesticide Recommendation.py

```
# Part 1 - Building the CNN
#importing the Keras libraries and packages
from keras.models import Sequential
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense, Dropout
from keras import optimizers

# Initialing the CNN
classifier_ = Sequential()

# Step 1 - Convolution Layer
classifier_.add(Convolution2D(32, 3,  3, input_shape = (64, 64, 3), activation = 'relu'))

#step 2 - Pooling
classifier_.add(MaxPooling2D(pool_size =(2,2)))

# Adding second convolution layer
```

```
classifier_.add(Convolution2D(32, 3,  3, activation = 'relu'))
classifier_.add(MaxPooling2D(pool_size =(2,2)))

#Adding 3rd Convolution Layer
classifier_.add(Convolution2D(64, 3,  3, activation = 'relu'))
classifier_.add(MaxPooling2D(pool_size =(2,2)))

#Step 3 - Flattening
classifier_.add(Flatten())

#Step 4 - Full Connection
classifier_.add(Dense(256, activation = 'relu'))
classifier_.add(Dropout(0.5))
classifier_.add(Dense(10, activation = 'softmax'))

#Compiling The CNN
classifier_.compile(
            optimizer = 'adam',
            loss = 'categorical_crossentropy',
            metrics = ['accuracy'])

#Part 2 Fittting the CNN to the image
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(
        rescale=1./255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1./255)

training_set = train_datagen.flow_from_directory(
        'Data/train',
        target_size=(64, 64),
        batch_size=32,
        class_mode='categorical')

test_set = test_datagen.flow_from_directory(
        'Data/test',
        target_size=(64, 64),
        batch_size=32,
        class_mode='categorical')

model = classifier_.fit_generator(
        training_set,
        steps_per_epoch=100,
        epochs=100,
        validation_data = test_set,
        validation_steps = 6500
      )

#Saving the model
import h5py
classifier_.save('Trained_Model.h5')
```

CHAPTER 7

SYSTEM TESTING

Introduction

The cause of testing is to detect mistakes. Making an attempt out is the technique of looking for to realize each viable fault or weakness in a piece product. It presentsa method to determine the performance of add-ons, sub-assemblies, assemblies and/or a completed product. It is the method of excising program with the intent of constructing certain that the application procedure meets its necessities and client expectations and does no longer fail in an unacceptable process. There are rather plenty of forms of scan. Each experiment sort addresses a special trying out requirement.

TYPES OF TESTS:

Unit Testing:

In the Unit Testing each module is considered independently. It focuses on each unit of software as implemented in the source code. It is white box testing. Unit checking out involves the design of scan circumstances that validate that the Internal application good judgment is functioning safely, and that program inputs produce legitimate outputs. All decision branches and interior code float must be validated. It's the checking out of character application items of the application It is achieved after the completion of an person unit earlier than integration.

It is a structural checking out, that relies on competencies of its construction and is invasive. Unit exams participate in common exams at component level and scan a distinct business approach, utility, and/or process configuration. Unit testing is mainly performed by the developers.

Integration Testing:

Integration testing aims at constructing the program structure while at the same constructing tests to uncover errors associated with interfacing the modules.Modules are integrated by using the top down approach.Integration Testing are designed to scan built-in program accessories to determine within the occasion that they evidently run as one software.

Trying out is occasion driven and is more concerned with the fundamental final result of screens or fields.Integration assessments reveal that despite the fact that the accessories had been formy part pleasure, as proven through effectively unit checking out, the combo of accessories is correct and regular. Integration checking out is chiefly aimed at exposing the issues that come up from the performance of different components.

Functional Testing:

Functional Testing checks provide systematic demonstrations that capabilitiesestablished are to be had as particular by means of the business and technical specifications, method documentation, and consumer manuals. Functional testing is working on below mentioned data:

Legitimate input : identified lessons of legitimate input ought to be accredited.Invalid enter : recognized lessons of unacceptable effort must be rejected.

Capabilities : recognized features ought to be exercised.

Output : recognized courses of software outputs have got to be exercised.Systems/Procedures : performance of the system here was invoked.

Individual and team work of useful checks is fascinated by specifications, key capabilities, or special scan instances. Moreover, systematic insurance plan concerning establish business method flows; data fields, predefined processes, andsuccessive strategies have to be regarded for trying out. Before useful trying out iswhole, extra checks are recognized and the strong price of present checks be strong-minded.

System Testing:

System Testing is executing programs to check logical changes made in it with intention of finding errors. A system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements. Scheme difficult ensure so as to the whole included agenda process meets principles.It exams a pattern to make sure identified and predictable outcome. An illustration of procedure testing is the configuration oriented approach integration scan. Systemtesting is based on approach descriptions and flows, emphasizing pre-driven system links and integration aspects.

White Box Testing:

This testing is a trying out wherein where the application tester has competencies of the interior workings, constitution and software language, or at least its cause. It's rationale. It's used to test areas that can't be reached from a black box stage. In this technique, the close examination of the logical parts through the software are tested by cases that exercise species set of conditions or loops. All logical parts of the software checked once. Errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops.

Black Box Testing:

This is testing the software with none advantage of the inside workings, establishment or words of the unit life form veteran. Black field checks, as most other sorts of functionalities. This method enables the software engineer to device sets of input techniques that fully exercise all functional requirements for a program. Black box testing tests the input, the output and the external data. It checks whether the input data is correct and whether we are getting the desired output.

LEVELS OF TESTING

Unit Testing Strategy:

Unit checking out is most commonly performed as a part of a mixed code and unitexperiment part of the software life cycle, though it be not exceptional for codingand unit checking out to be performed as two targeted phases.

Test strategy and approach:

Field testing out can be carried out manually and sensible assessments shall be writtenin element.

Test objectives:

- Each field must be work correctly.
- Each page must be activated through the specified link.
- Features to be tested Verify that the entries are of the correct format No duplicateentries should be allowed.

Integration Testing Strategy:

Software integration testing is the incremental integration checking out of two otherwise further included software gears on top of a solo stage to fabricate failureinduced with the aid of interface defects. The project of the mixing scan is to checkthat components or program applications, e.g. Components in a program approach or œ one step up œ software purposes at the company degree œ interact without error.

Test Results:

All of the scan circumstances recounted above passed efficiently. No defects encountered.

Acceptance Testing

User Acceptance testing trying out is a crucial section of any mission and requires enormous participation by the tip user.It additionally ensures that the procedure meets the functional specifications.

Test Results: The entire test cases recounted above passed effectually. No defects Encountered.

CHAPTER 8

CONCLUSION

The conclusion for the project of a farm assistant that provides crop and pesticide recommendations is that it has the potential to significantly improve farming practices and increase agricultural productivity. By leveraging advanced technologies such as machine learning, data analysis, and deep learning, the farm assistant can provide farmers with more accurate, tailored, and timely recommendations.

CHAPTER 9

FUTURE ENHANCEMENTS

Here are some future enhancements for the Project Farm Assistant that provides crop recommendation and pesticide recommendation:

1. Real-time weather data integration. This would allow the system to recommend crops and pesticides that are best suited for the current weather conditions.
2. Integration with soil data. This would allow the system to recommend crops and pesticides that are best suited for the specific soil conditions of a farm.
3. Machine learning algorithms. These algorithms could be used to improve the accuracy of the crop and pesticide recommendations.
4. A mobile app. This would make the system more accessible to farmers who are not always near a computer.
5. A chat bot. This would allow farmers to get help and advice from the system 24/7.

CHAPTER 10

REFERECES

1. https://keras.io/

2. www.google.com

3. https://ieeexplore.ieee.org/document/9418351

4. https://www.frontiersin.org/articles/10.3389/fpls.2022.839572/full