AGRI WISE-BRIDGING TRADITION WITH TECHNOLOGY

An Industrial Oriented Mini Project (CS653PC)

Submitted

in partial fulfilment of the requirements for completion of

Bachelor of Technology VI Semester

in

Computer Science and Engineering

by

B. Meenakshi 22261A05D4

and

G. Tejasvi Reddy 22261A05E5

Under the guidance of

Dr. K Sreekala

Assistant Professor

and

Y. Pavan Narasimha Rao

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY(A)

Gandipet, Hyderabad-500075, Telangana (India)

2024-2025

MAHATMA GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)
Gandipet, Hyderabad-500 075, Telangana (India)

CERTIFICATE

This is to certify that the an Industry Oriented Mini Project (CS653PC) entitled **“Agri Wise-Bridging Tradition with Technology ”** is being submitted by **B. Meenakshi** bearing **roll no. 22261A05D4** and **G. Tejasvi Reddy** bearing **roll no. 22261A05E5** in partial fulfillment for completion of **Bachelor of Technology VI Semester** in **Computer Science and Engineering , Mahatma Gandhi Institute of Technology** is a record of bona-fide work carried out under the guidance and supervision.

Project Guide
Dr. K Sreekala
Asst. Professor, Dept of CSE

Head of the Department
Dr. C.R.K Reddy
Professor, Dept of CSE

Project Guide
Y. Pavan Narasimha Rao
Asst. Professor, Dept of CSE

EXTERNAL EXAMINER

DECLARATION

This is to certify that the work reported in this Industry Oriented Mini Project (CS653PC) titled **“Agri Wise-Bridging tradition with technology”** is a record of work done by us in the Department of Computer Science and Engineering, Mahatma Gandhi Institute and Technology, Hyderabad. No part of the work is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred to in the text. The report is based on the work done entirely by us and not copied from any other source.

B. Meenakshi
(22261A05D4)

G. Tejasvi Reddy
(22261A05E5)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible because success is the abstract of hard work and perseverance, but steadfast of all is encouraging guidance. So, we acknowledge all those whose guidance and encouragement served as a beacon light and crowned my efforts with success.

We would like to express my sincere thanks to **Prof. G. Chandra Mohan Reddy, Principal, MGIT**, for providing the working facilities in the college.

We wish to express my sincere thanks and gratitude to **Dr. C. R. K. Reddy, Professor and HOD, Department of CSE, MGIT** for all the timely support and valuable suggestions during the period of project.

We are extremely thankful to **Dr. K. Sreekala, Assistant Professor & Mr. Y. Pavan Narismha Rao, Assistant Professor, Department of CSE**, **MGIT**, Industry Oriented Mini Project Guides for their encouragement and support throughout the project.

We are extremely thankful to **Mr. N. Rama Krishna ,Assistant Professor & Mr. G. Nagi Reddy, Assistant Professor, Department of CSE**, **MGIT**, Industry Oriented Mini Project Coordinators for their encouragement and support throughout the project.

Finally, we would also like to thank all the faculty and staff of the CSE Department who helped us directly or indirectly for completing this project.

B. Meenakshi
(22261A05D4)

G. Tejasvi Reddy
(22261A05E5)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ML	: Machine Learning
DL	: Deep Learning
CSV	: Comma-Separated Values
CNN	: Convolutional Neural Network
KNN	: K-Nearest Neighbour
UI	: User Interface
AI	: Artificial Intelligence
VT	: Vision Transformer
VS	: Visual Studio Code
CPU	: Central Processing Unit

ABSTRACT

The Argo Wise application is a comprehensive desktop-based software tool designed to assist farmers with intelligent agricultural decision-making. Developed using Python's Tkinter GUI framework and integrated with machine learning, the system provides personalized guidance across several critical areas such as crop selection, fertilizer recommendation, and pesticide management. The primary objective of the platform is to enhance farming efficiency and sustainability by combining data-driven insights with an intuitive user interface. A standout feature of the application is its crop recommendation system, which uses an NPK dataset alongside real-time inputs like temperature, humidity, pH, and rainfall to suggest the most suitable crops for cultivation. This functionality is powered by a trained machine learning model, enabling accurate predictions based on environmental and soil conditions. Additionally, the platform offers fertilizer and pesticide guidance tailored to specific crops, complete with visual references to improve usability. To further empower users, a chatbot powered by the Gemini AI model is included, enabling farmers to receive instant answers to agricultural queries. With a strong emphasis on ease of use, the software caters even to users with minimal technical expertise. Through image-based input selections, real-time crop insights, and AI-powered conversation support, the Agro Wise application bridges the gap between traditional farming practices and modern technological advancements. Ultimately, it aims to equip farmers with the knowledge and tools necessary to optimize crop yields, promote environmental stewardship, and contribute to a more sustainable agricultural ecosystem.

Keywords: Crop Recommendation System, Environmental parameters, Data driven decision making, Smart farming, Sustainable agriculture

1. INTRODUCTION

The Argo Wise application is an innovative desktop-based solution developed to support and modernize agricultural practices by providing farmers with intelligent, data-driven recommendations. Designed with an intuitive graphical user interface using Python's Tkinter library and enhanced by machine learning algorithms, the application assists users in making informed decisions regarding crop selection, fertilizer usage, and pest management. By incorporating key environmental inputs such as soil type, temperature, humidity, and rainfall, the system generates tailored suggestions that enhance crop productivity and sustainability. Additionally, the integration of a conversational AI chatbot powered by Gemini allows users to access real-time agricultural assistance, making the platform not only interactive but also educational. Through this blend of technology and user-friendly design, Agro Wise aims to bridge the gap between conventional farming techniques and emerging digital innovations, ultimately contributing to the growth and resilience of the agricultural sector.

1.1 Problem Statement

Despite rapid technological advancements, a significant number of farmers still rely on traditional methods and lack access to real-time, personalized agricultural support. This gap leads to challenges such as improper crop selection, inefficient use of fertilizers and pesticides, and poor market awareness, ultimately affecting productivity and sustainability. Moreover, most existing platforms demand a certain level of technical expertise or internet dependency, making them inaccessible to many rural users. There is a pressing need for an intuitive, intelligent, and offline-accessible solution that simplifies agricultural decision-making while integrating modern technologies like machine learning and AI for accurate and timely guidance.

1.2 Existing Applications

The Farmers' Portal on the National Portal of India provides essential resources to support farmers in various aspects of agriculture. It offers detailed guidance on crop selection, helping farmers choose the best crops based on climate, soil, and water availability. For fertilizer selection, the portal provides tailored advice to enhance soil health and crop yields, along with information on government subsidies. Pesticide selection is also covered, offering guidelines for effective and environmentally safe pest control.

Disadvantages:

- **Technical knowledge required and connectivity concerns** [2]:
  Most online platforms require stable internet access and basic digital literacy, which may not be available in many rural areas. This restricts accessibility and limits the utility of the tools for many farmers.

- **Limited interactivity and personalization**:
  Existing portals provide static information with minimal user engagement or customization based on individual farm conditions, which reduces the effectiveness of recommendations.

- **Competition with cash crops** [1]:
  General guidance on cover crops may conflict with farmers' focus on high-value cash crops, creating confusion in resource allocation and cropping patterns.

- **Dependency on government support** [4]:
  Many advisory platforms rely heavily on government updates and policies. Delays or inconsistencies in content can result in outdated or inaccurate information for users.

- **Knowledge and skill requirements** [3]:
  Farmers may struggle with interpreting scientific data, pesticide quantities, and integrated pest management techniques due to lack of training or education in these areas.

1.3 Proposed Application

The AgriWise application stands out by providing a comprehensive, intelligent desktop-based platform that addresses the shortcomings of existing solutions. It features crop prediction through machine learning, tailored fertilizer and pesticide suggestions, and an AI-powered chatbot for real-time guidance—all integrated into a simple, image-driven interface that operates offline.

Advantages:

- **AI-Based Crop Recommendation System**:
  Uses environmental inputs and NPK values to suggest suitable crops through a machine learning model, ensuring precise and data-backed recommendations.

- **Interactive and Visual Interface**:
  Provides an intuitive image-based UI that enhances usability, especially for farmers with limited technical knowledge.

- **Offline Accessibility**:
  Being a desktop application, it reduces dependence on continuous internet access, making it ideal for rural and low-connectivity regions.

- **Real-Time AI Chatbot Support**:
  Integrated Gemini chatbot enables users to ask farming-related questions and receive dynamic responses in natural language.

- **Tailored Fertilizer and Pesticide Guidance**:
  Offers specific recommendations and visual references for each crop, improving resource management and reducing environmental impact.

- **Empowerment Through Simplicity**:
  Designed with minimal learning curve, the application empowers farmers to independently make critical agricultural decisions.

1.4 Requirements specification

1. Software Requirements:

- **Libraries** : Tkinter GUI Library of Python
- **Packages** : Pillow, joblib, google.generativeai
- **Coding Language** : Python (latest version with IDLE)
- **ML Integration** : Scikit-learn model (crop recommendation)

2. Hardware Requirements:

- **RAM and Processor** : Minimum 4 GB RAM and Dual Core Processor
- **Operating System** : Windows 10 (64-bit) or later

2. LITERATURE SURVEY

1. “**Machine Learning for Crop Recommendation and Yield Prediction**” by N. Patel, R. Shah, P. Dabhi.

   This study proposes a data-driven system for recommending crops and predicting yield using machine learning models such as Decision Trees and Random Forests. It focuses on optimizing crop selection based on soil parameters (NPK values), temperature, rainfall, and pH. The system was trained on agricultural datasets and evaluated using accuracy and F1 score, demonstrating Random Forest's superior performance. The paper emphasizes the role of supervised learning models in improving agricultural decision-making and productivity.

2. “**Crop Selection Method to Maximize Yield Using Machine Learning Technique”** by S. Parikh, J. Mistry.

   This research introduces a machine learning-based method to recommend the most suitable crop by analyzing environmental variables like temperature, rainfall, and soil pH. The authors apply the Naive Bayes classification technique on real-world agricultural datasets to predict crops with higher accuracy. The model aims to assist farmers in selecting crops that are most likely to thrive in given conditions, thereby enhancing yield and minimizing resource waste.

3. “**Smart Farming Based on Cloud Computing and IoT**” by A. Kamilaris, A. Kartakoullis, F. Prenafeta-Boldú.

   This paper presents a comprehensive overview of how cloud computing and IoT technologies are transforming modern agriculture. It emphasizes the integration of sensor-based data collection and cloud-based analytics to deliver real-time recommendations on irrigation, fertilization, and pest control. The review highlights successful case studies and implementation challenges while advocating for smart farming infrastructure in developing regions.

4. “**Precision Agriculture and the Future of Farming in Europe**” by European Parliament Research Service.

   This policy document explores the future of precision agriculture in Europe and its implications for food security, rural development, and environmental sustainability. The paper discusses the adoption of GPS, remote sensing, drones, and robotics in farming and their impact on improving resource efficiency and yield. It also considers regulatory and ethical challenges in deploying technology-driven agriculture at scale.

5. “**Use of Data Mining Techniques in Agriculture**” by J. Han, M. Kamber.

   This foundational text on data mining outlines how techniques such as classification, clustering, and association rules can be used in agricultural data analysis. Applications discussed include soil fertility classification, crop yield forecasting, and pest/disease prediction. The book introduces concepts like decision trees and rule-based learning in the context of real-world agricultural problems.

6. “**Crop Disease Prediction Using Deep Learning**” by S. Sladojevic, M. Arsenovic, et al.

   A CNN-based deep learning model is proposed for classifying plant diseases from leaf images. This system significantly enhances early disease detection accuracy, making it a valuable tool in precision agriculture. The paper also discusses the challenges of deploying deep learning in rural farming setups due to limited computational resources.

7. “**Recommendation System for Farmers Using Machine Learning”** by S. Deshmukh, K. Pawar, et al.

   This study outlines a crop and fertilizer recommendation system based on location-specific soil and weather data using supervised learning models. The approach is practical and cost-effective, especially for farmers with limited access to expert agronomists. It highlights the scalability of ML in local and regional farming contexts.

8. “**Smart Farming: IoT-Based Smart Sensors Agriculture Stick**” by S. Nandurkar, V. Thool, et al.

   This project describes a smart sensor stick for real-time monitoring of soil moisture, humidity, and temperature. It helps farmers make data-driven decisions about irrigation and crop health. The device integrates with a mobile app to provide actionable insights, supporting smart farming practices in resource-limited areas.

9. “**Big Data in Smart Farming**” by S. Wolfert, L. Ge, et al.

   The paper explores the use of big data analytics to optimize agricultural processes. It reviews frameworks and architectures that collect and analyze vast amounts of sensor and environmental data to deliver personalized recommendations on irrigation, crop choice, and harvest timing. It emphasizes interoperability and data sharing among agricultural stakeholders.

10. “**Soil Parameter Estimation and Crop Recommendation Using Machine Learning**” by V. S. Dinesh, S. Murthy.

    The paper applies supervised ML techniques to estimate soil fertility and recommend appropriate crops. It uses a dataset with features such as NPK, pH, and temperature, comparing the performance of Decision Trees and SVMs. The results show high prediction accuracy and potential for use in smart advisory systems.

11. “**Weather-Based Crop Prediction Using ML**” by A. Ghosh, T. Banerjee.

    This research incorporates meteorological data into an SVM model to predict optimal crops and harvest times. The authors highlight the seasonal impact on yield and propose a dynamic recommendation model that adjusts to changing weather patterns, improving agricultural resilience.

12. “**Fuzzy Logic Based Expert System for Crop Recommendation**” by S. Ramesh, K. Sakthivel.

A fuzzy logic approach is used to handle uncertainty and imprecision in agricultural inputs. The expert system simulates human decision-making, offering crop recommendations based on soil quality, climate, and water availability. It is particularly useful in low-tech environments where precise sensor data may be lacking.

13. “**A Survey on Applications of ML in Agriculture**” by R. Kaur, M. Kaur.

This review summarizes machine learning applications across various agricultural domains including weed detection, plant disease classification, soil type prediction, and yield forecasting. It highlights common algorithms such as SVM, KNN, and Decision Trees, along with their performance metrics and challenges.

14. “**Soil Data Analysis Using ML Algorithms**” by M. Patel, P. Pandya.

The paper compares the effectiveness of different ML algorithms (SVM, Random Forest, KNN) for soil classification and crop suggestion. It uses real soil data from agricultural regions and emphasizes the importance of feature selection in improving model accuracy.

15. “**Smart Crop Suggestion Using Decision Tree and IoT**” by M. Kulkarni, A. Waghmare.

This work integrates IoT sensors with a Decision Tree algorithm to provide real-time crop suggestions. By continuously monitoring soil and environmental conditions, the system dynamically updates its recommendations. The paper demonstrates a prototype system and its effectiveness in test environments.

16. “**Advances in Precision Agriculture Technologies for Sustainable Crop Production**” by Vikram Singh et al.

This paper explores the latest advancements in precision agriculture, highlighting their potential to enhance sustainability, increase productivity, and reduce environmental impact. By leveraging cutting-edge technologies such as GPS mapping, remote sensing, and data analytics, precision agriculture enables farmers to make data-driven decisions, optimize resource use, and improve crop yields. Through an exploration of these innovations, the authors showcase how precision agriculture is transforming farming practices and paving the way for a more resilient and efficient agricultural future.

17. “**Cover Crops Effect on Soil Quality and Soil Health”** by M. R. Rahman et al.

Focusing on growing concerns over agricultural sustainability, this paper explores the profound impact of cover crops on soil quality and health. It highlights their potential to mitigate erosion, suppress weeds, improve water retention, and enhance nutrient cycling. Through empirical research and expert insights, the study delves into the mechanisms behind cover crops' influence on soil properties and their broader implications for climate resilience and sustainable agriculture.

18. “**Technology Diffusion, Government Policy and Agricultural Sustainable Development**” by Zhang Ju-Yong et al.

This study examines the pivotal role that technology diffusion and supportive government policies play in transforming agricultural practices. It explores how policies can facilitate or hinder the adoption of cutting-edge technologies, emphasizing the interplay between governance and innovation. Through case studies and expert insights, the authors highlight strategies that drive successful technology adoption and sustainable agricultural development.

19.“**Optimization Chemical Control in Integrated Pest Management**” by Hong Jun Guan et al.

This paper discusses the optimization of chemical control within Integrated Pest Management (IPM) systems. It promotes a balanced approach that combines chemical, biological, cultural, and mechanical pest control methods. The goal is to enhance pest management efficacy while minimizing environmental and health impacts. The study presents innovative approaches and technologies for precise, sustainable chemical usage in agriculture.

20.“**Crop Recommendation Using Hybrid ML Techniques**” by S. Aravindan, A. M. Jyothi.

This work evaluates hybrid machine learning models for crop prediction based on soil fertility, environmental factors, and rainfall. It combines ensemble techniques and feature selection to improve accuracy and reliability. The authors demonstrate how hybrid models can outperform single algorithms in diverse agricultural datasets, making them well-suited for real-world deployment.

Table 2.1: Literature survey table

S.No	Authors	Title	Year	Journal/Conference	Methodology / Concept Used / Summary	Merits	Demerits
1	N. Patel, R. Shah, P. Dabhi	Machine Learning for Crop Recommendation and Yield Prediction	2018	WiSPNET	Random Forest and Decision Tree algorithms used on agricultural datasets to predict suitable crops.	High accuracy and interpretability.	Limited to static environmental variables.
2	S. Parikh, J. Mistry	Crop Selection Method to Maximize Yield Using Machine Learning Technique	2017	IJCA	Naive Bayes classification used for predicting suitable crops.	Easy implementation and good accuracy.	Limited scope of parameters.
3	A. Kamilaris, A. Kartakoullis, F. Prenafeta-Boldú	Smart Farming Based on Cloud Computing and IoT	2017	Computers and Electronics in Agriculture	Reviews IoT and cloud computing applications in farming.	Real-time data integration and decision making.	Infrastructure dependent.
4	European Parliament Research Service	Precision Agriculture and the Future of Farming in Europe	2016	Policy Report	Overview of technologies in precision agriculture.	Supports sustainable farming practices.	Ethical and regulatory challenges.
5	J. Han, M. Kamber	Use of Data Mining Techniques in Agriculture	2011	Textbook	Data mining techniques applied in agriculture.	Foundational guide with multiple techniques.	Theoretical without implementation specifics.

6	S. Sladojevic, M. Arsenovic, et al.	Crop Disease Prediction Using Deep Learning	2016	CEAgri	CNN-based plant disease classification.	Early disease detection, high precision.	High computational requirement.
7	S. Deshmukh, K. Pawar, et al.	Recommendation System for Farmers Using ML	2018	IJRCCE	Supervised learning-based crop and fertilizer suggestions.	Local data integration.	Regional limitations.
8	S. Nandurkar, V. Thool, et al.	Smart Farming: IoT-Based Smart Sensors Agriculture Stick	2016	IEEE ICCIC	IoT sensor-based real-time monitoring stick.	User-friendly and mobile integrated.	Hardware maintenance issues.
9	S. Wolfert, L. Ge, et al.	Big Data in Smart Farming	2017	Agricultural Systems	Review of big data architectures in agriculture.	Comprehensive and scalable solutions.	Data interoperability challenges.
10	V. S. Dinesh, S. Murthy	Soil Parameter Estimation and Crop Recommendation Using ML	2018	IRJET	Decision Tree and SVM used on soil data.	High prediction accuracy.	Limited real-world testing.
11	A. Ghosh, T. Banerjee	Weather-Based Crop Prediction Using ML	2017	IEEE ICCCNT	SVM applied with weather forecasting.	Seasonal adaptation.	Prediction errors due to climate variability.
12	S. Ramesh, K. Sakthivel	Fuzzy Logic Based Expert System for Crop Recommendation	2015	Journal of AI	Fuzzy rule-based crop decision system.	Handles imprecise input well.	Needs manual rule tuning.
13	R. Kaur, M. Kaur	A Survey on Applications of ML in Agriculture	2019	IJRCS	Review of ML in agriculture: KNN, SVM, Decision Tree.	Broad overview of tools and uses.	Lacks deep experimental insights.

14	M. Patel, P. Pandya	Soil Data Analysis Using ML Algorithms	2018	IJRASET	Compares SVM, KNN, and Random Forest for soil data.	Empirical accuracy comparison.	Lacks advanced validation.
15	M. Kulkarni, A. Waghmare	Smart Crop Suggestion Using Decision Tree and IoT	2020	IJSR	Integrates IoT sensors and Decision Trees.	Dynamic recommendation.	Prototype stage only.
16	Vikram Singh et al.	Advances in Precision Agriculture Technologies for Sustainable Crop Production	2021	Elsevier	Reviews GPS, remote sensing, and analytics in farming.	Promotes sustainable precision farming.	Tech accessibility in rural areas.
17	M. R. Rahman et al.	Cover Crops Effect on Soil Quality and Soil Health	2020	Soil & Tillage Research	Empirical study of cover crops' impact on soil.	Improved soil health, sustainability.	Implementation varies regionally.
18	Zhang Ju-Yong et al.	Technology Diffusion, Government Policy and Agricultural Sustainable Development	2019	Agri Policy Journal	Case study-based policy-tech impact assessment.	Policy recommendations and insights.	Context-sensitive applicability.
19	Hong Jun Guan et al.	Optimization Chemical Control in Integrated Pest Management	2018	IPM Journal	IPM techniques combining chemical and biological control.	Reduced pesticide use.	Needs multidisciplinary coordination.
20	S. Aravindan, A. M. Jyothi	Crop Recommendation Using Hybrid ML Techniques	2021	AI in Agriculture	Uses ensemble learning and feature selection.	Higher prediction accuracy.	Complex model integration.

3. METHODOLOGY

In the development of the AgriWise application, a desktop-based approach using Python's Tkinter library was adopted to create an intuitive and accessible tool for farmers. The methodology focused on integrating machine learning for intelligent recommendations alongside static information and external resources.

3.1 Technologies Used:

The AgriWise application primarily leveraged Python and its specialized libraries for GUI development, machine learning model integration, and external API communication:

- **Python**: As the core programming language, Python provided the robust environment for developing the application's logic, integrating various functionalities, and handling data.
- **Tkinter**: This standard Python library was utilized for building the graphical user interface (GUI) of the application, providing a user-friendly and interactive experience for farmers.
- **Pillow (PIL)**: Integrated with Tkinter, Pillow was essential for handling and displaying various image assets within the application, enhancing visual appeal and information delivery.
- **Joblib**: This library was crucial for efficiently loading the pre-trained machine learning model (crop recommendation model), enabling the application to perform rapid predictions for crop suggestions.
- **Google Generative AI (Gemini API)**: The Gemini API was integrated to power the "Agro ChatBot" feature, providing a conversational interface for users to ask farming-related questions and receive AI-generated responses.
- **Webbrowser**: This Python module facilitated seamless navigation to external web resources, such as live crop market rates and agricultural technique videos, directly from the application.

3.2 Development Process:

The development process for AgriWise followed a structured approach, emphasizing user interaction, data integration, and the delivery of practical agricultural insights:

1. **Requirements Gathering**: The initial phase involved understanding the key information and tools farmers require, including crop suitability, fertilizer guidance, pest management.
2. **System Design**: A desktop application architecture was designed, outlining the Tkinter-based GUI layout, the integration points for the machine learning model, the Gemini API for the chatbot, and external web links. Emphasis was placed on clear navigation and visual presentation.
3. **Implementation**: The application's various modules were developed in Python. This included crafting the Tkinter GUI elements, implementing the logic for user input collection, integrating the pre-trained ML model for predictions, setting up the API calls for the chatbot, and embedding external web browser functionalities.
4. **Testing**: Rigorous testing was conducted to ensure the application's reliability and usability. This involved:
   - **GUI Functionality Testing**: Validating that all buttons, input fields, and navigation elements worked as intended across different sections.
   - **ML Model Integration Testing**: Verifying that the crop suggestion model loaded correctly, accepted inputs, and produced accurate and relevant predictions within expected ranges.
   - **Chatbot Functionality Testing**: Assessing the Gemini API integration, ensuring the chatbot could receive queries and provide coherent and helpful responses.
   - **External Link Verification**: Confirming that all embedded web links opened correctly in the default browser.
   - **Image Loading and Display**: Ensuring all image assets were correctly loaded and displayed within the Tkinter windows.
5. **Deployment**: The AgriWise application is designed as a standalone Python executable (though not explicitly packaged in the provided code, this is the typical deployment for Tkinter apps), allowing it to run on local machines without complex server setups.

4. DESIGN

4.1 System Design

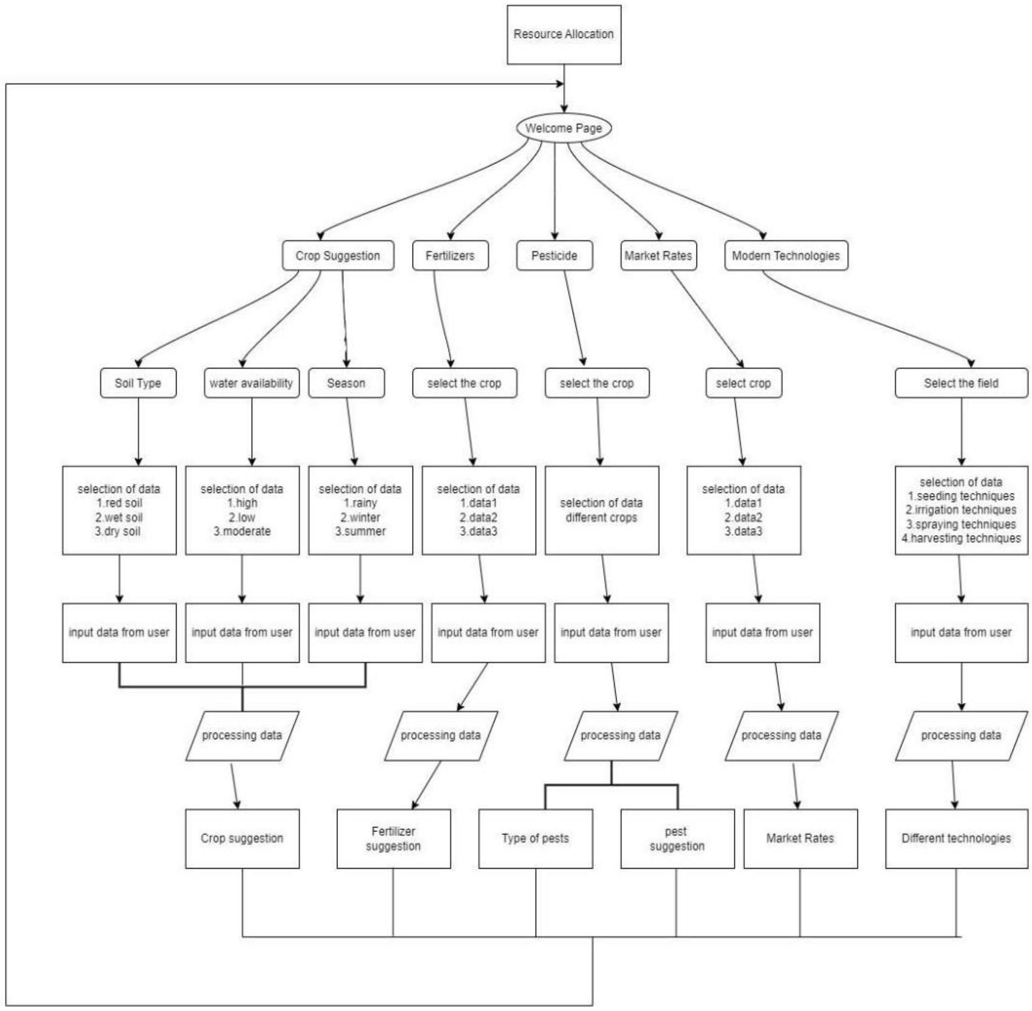

Figure 4.1 System Architecture

The Agri Wise desktop application architecture is designed to provide farmers with intelligent, data-driven guidance through a visually interactive interface. The flow begins with user interaction via the dashboard, allowing selection between key agricultural services: Crop Suggestion, Fertilizer Guidance, Pesticide Management, Market Insights, and an AI Chatbot. Once a selection is made, the system gathers relevant user inputs such as soil type, water availability, temperature, humidity, and pH levels. The architecture integrates a machine learning model for crop prediction, processing NPK values and environmental conditions to recommend the most suitable crops. Each selected crop links to its respective fertilizer and pesticide information, displayed with supportive visual elements. An embedded AI chatbot, powered by the Gemini model, provides real-time conversational assistance. The system is structured to operate offline, with all media, logic, and predictions embedded in the local environment.

Architecture Components:

- **User Dashboard**: Central control panel from which the user selects desired features.
- **Crop Suggestion (ML Integrated)**: Uses NPK values, weather data, and user input to suggest suitable crops.
- **Fertilizer Guidance**: Displays fertilizers best suited to the selected crop with images and details.
- **Pesticide Management**: Provides major pests and matching pesticides for different crops.
- **Water Availability**: Incorporated into crop prediction logic to ensure sustainability and irrigation planning.
- **AI Chatbot (Gemini)**: Offers personalized answers to farming-related queries in real-time.
- **Data Processing**: Handles environmental inputs and feeds into the ML model for accurate prediction and suggestion logic.

4.2 UML Diagram

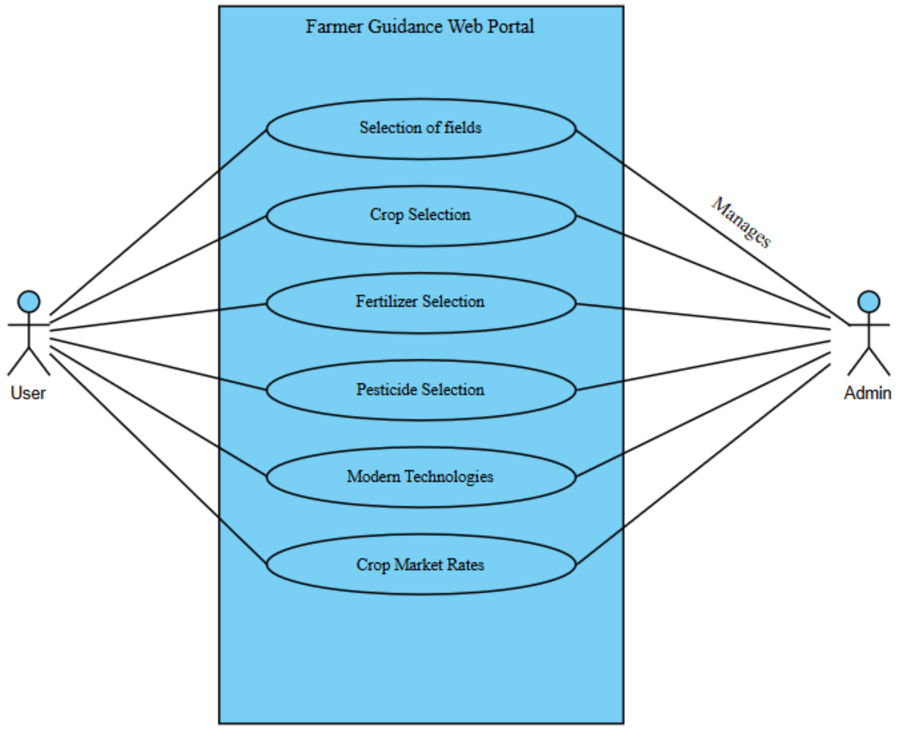

Figure 4.2 Use Case Diagram

Figure 4.2 shows the use case diagram, shows the user will be able to select different fields i.e. crop selection, fertilizer selection, pesticide selection, modern technologies and crop markets, according to the selection of the user the admin will manage the data and show suggestions according to the user input.

4.3 Sequence Diagram

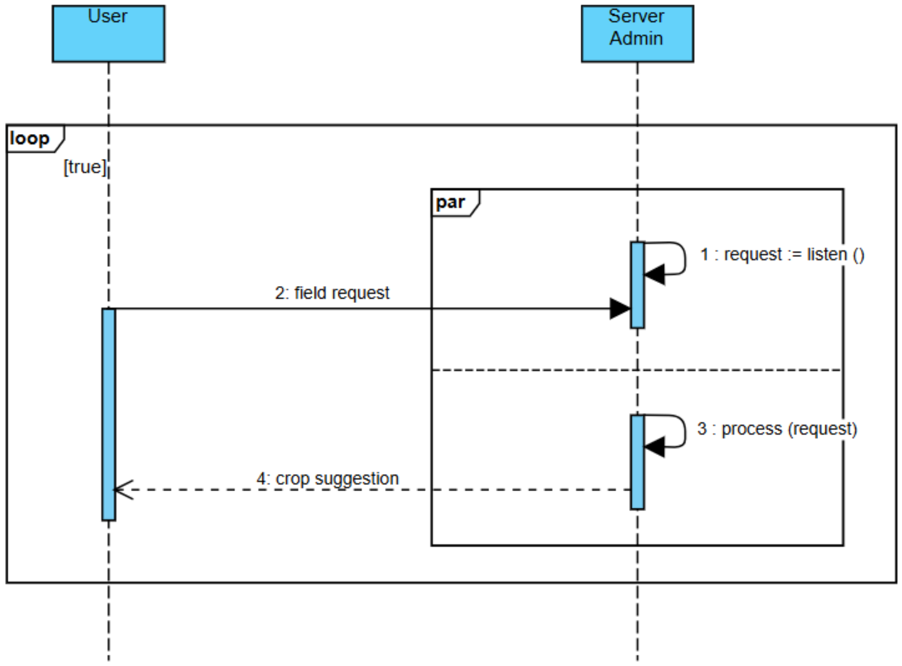

Figure 4.3 Sequence Diagram

Figure 4.3 shows the sequence diagram that the user needs to access the field and make a request to the server admin and according to the field the result or crop suggestion given to the user. This process will run in a loop according to the user.

5. IMPLEMENTATION

The implementation of the AgriWise application involved translating the system design into functional Python code, integrating various components to deliver the desired agricultural functionalities.

5.1 Frontend Components

- Tkinter GUI: Tkinter was extensively used to construct the graphical user interface. This included creating the main window (top) and various Toplevel windows for different functionalities (e.g., crop, fertilizer, pesticide, cropmarket, modern, chatbot). Widgets such as Label, Button, Entry, Radiobutton, and Text were employed for displaying information, taking user input, and enabling navigation.
- Pillow (PIL) for Image Display: The PIL (Pillow) library was crucial for loading and displaying images (Image.open, ImageTk.PhotoImage) within the Tkinter labels and buttons, providing visual context for soil types, water availability, crop-specific fertilizers/pesticides, and general welcome/logo images.

5.2 Backend Components

- Python Core Logic: The application's core logic, written in Python, manages user interactions, processes inputs, and orchestrates calls to various modules. This includes functions for handling button clicks, managing window transitions, and validating user input.
- Machine Learning Model Integration (joblib): The crop_recommendation_model.pkl file, containing the pre-trained machine learning model, is loaded using joblib.load(). User inputs for soil type, water availability, temperature, humidity, and pH are processed and mapped to numerical features (e.g., soil_npk_map, water_rainfall_map) before being fed into the loaded model for prediction.
- Google Generative AI Integration: The google.generative ai library is used to connect to the Gemini API. A GenerativeModel is initialized, and a chat session is started to maintain conversational context. User questions are sent to the API using chat send message(), and the AI's responses are retrieved and displayed in the chatbot interface.
- Web Browser Integration: The web browser module is used to open external URLs when users click on "Crop Market" or "Modern Techniques" video links, directing them to relevant online resources.

5.3 Key Functionalities

1. Crop Suggestion:
   - Users select soil type and water availability via radio buttons, and input temperature, humidity, and pH.
   - The selected soil type maps to NPK values, and water availability maps to rainfall, using predefined dictionaries (soil npk map, water rainfall map).
   - These inputs, along with temperature, humidity, and pH, form the feature vector for the loaded machine learning model.
   - The model. Predict() method is called to get the recommended crop.

```python
# Load ML model
model = joblib.load("crop_recommendation_model.pkl")
# Soil and water level mappings
soil_npk_map = {
    "Red Soil": [60, 30, 35],
    "Black Soil": [70, 50, 40],
    "Alluvial Soil": [90, 42, 43],
    "Clayey Soil": [65, 40, 38]
}
water_rainfall_map = {
    "High": 250,
    "Medium": 150,
    "Low": 50
}
# ... (inside open_prediction function) ...
try:
    temp = float(temp_entry.get())
    hum = float(hum_entry.get())
    ph = float(ph_entry.get())
    # Check if temperature and humidity are within valid range
    if not (TEMP_MIN <= temp <= TEMP_MAX):
        raise ValueError("Temperature out of range.")
    if not (HUM_MIN <= hum <= HUM_MAX):
        raise ValueError("Humidity out of range.")
    npk = selected_npk
    rainfall = selected_rain[0] if selected_rain else 0
    sample = [[npk[0], npk[1], npk[2], temp, hum, ph, rainfall]]
    prediction = model.predict(sample)[0]
    result_window = Toplevel()
    result_window.geometry("1500x1000")
    result_window.title("Prediction Result")
    Label(result_window, text=f"Recommended Crop: {prediction.upper()}",
font=("Arial", 20, "bold"), fg="green").pack(pady=80)
```

Figure 5.1: Code logic for crop suggestion prediction

2. Fertilizer Information:
   - Users select a crop from a series of buttons.
   - Upon selection, the application displays static images of relevant fertilizers and their details for the chosen crop. This is achieved by dynamically updating Label widgets with new Photo Image objects.
3. Pesticide Information:
   - Similar to the fertilizer section, users select a crop.
   - The application then displays static images of major pests affecting that crop and details about recommended pesticides, again by updating Label widgets with pre-loaded images.
4. Crop Market Rates:
   - Buttons are provided to directly open external agricultural market websites (https://www.oneindia.com/vegetables-price.html, https://agmarknet.gov.in/)
   - in the user's default web browser using the web browser. Open() function.
5. Modern Techniques:
   - This section presents information on modern agricultural techniques (seeding, irrigation, spraying, harvesting).
   - Each technique has a dedicated button that, when clicked, displays static text descriptions and images.
   - Additionally, a "Watch on YouTube" button is provided for each technique, which opens a placeholder YouTube link (https://youtu.be/WygacroTB8g?si=iAX6hfVyZ1yoRIw_ etc.) in the web browser.
6. Agro ChatBot:
   - An interactive chatbot interface is provided where users can type questions related to farming.
   - The google generative Ai library is used to send the user's query to the Gemini API (chat send message()).
   - The AI's response is then displayed in a text widget, providing a conversational experience.

```
# ===== CONFIGURE API KEY =====
genai.configure(api_key="AIzaSyB3j6x74WE1pqjOVR2zLtn6Od30_MRmxp0")

# ===== GUI APP FUNCTION =====
def chatbot():
    # ... (Tkinter setup) ...

    # Initialize a **chat session**
    model = genai.GenerativeModel("gemini-1.5-pro-latest")  # Using a generally
available model
    chat = model.start_chat(history=[])

    def ask_gemini():
        question = entry.get()
        if not question.strip():
            return
        text_area.insert(END, f"You: {question}\n")
try:
            response = chat.send_message(question)
            answer = response.text.strip()
            text_area.insert(END, f"AgroBot: {answer}\n\n")
        except Exception as e:
            text_area.insert(END, f"Error: {str(e)}\n\n")

        entry.delete(0, END)
    Button(top, text="Ask", font=("Arial", 14),
bg="lightblue",              command=ask_gemini).pack(pady=5)
    top.mainloop()
```
Figure 5.2: Code logic for Chatbot Interaction

The AgriWise application, developed as a Python Tkinter desktop application, requires a Python environment with the necessary libraries (Pillow, joblib, google-generativeai) installed. It also relies on locally stored image assets for its visual components. For optimal performance, a system with sufficient RAM and processing power for running the Python interpreter and handling image rendering is recommended.

6. TESTING AND RESULTS

1. Home page

Figure 6.1: Home Page

Figure 6.1 shows the opening page of my application which includes the input field of levels, as select the field of different suggestions. When we enter valid input in the given field we need to click on the field.

2. Field selection and crop suggestion

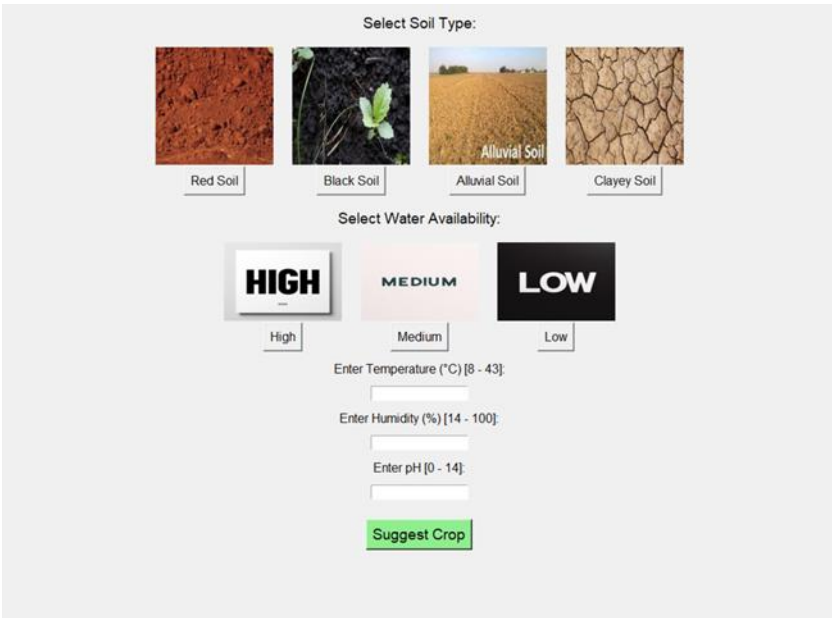

Figure 6.2: Field selection and crop suggestion

The Figure 6.2 shows a GUI-based crop suggestion system where users select soil type, water availability, and enter environmental parameters (temperature, humidity, and pH). A "Suggest Crop" button uses these inputs to predict the most suitable crop using a machine learning model.

3. Fertilizer suggestion

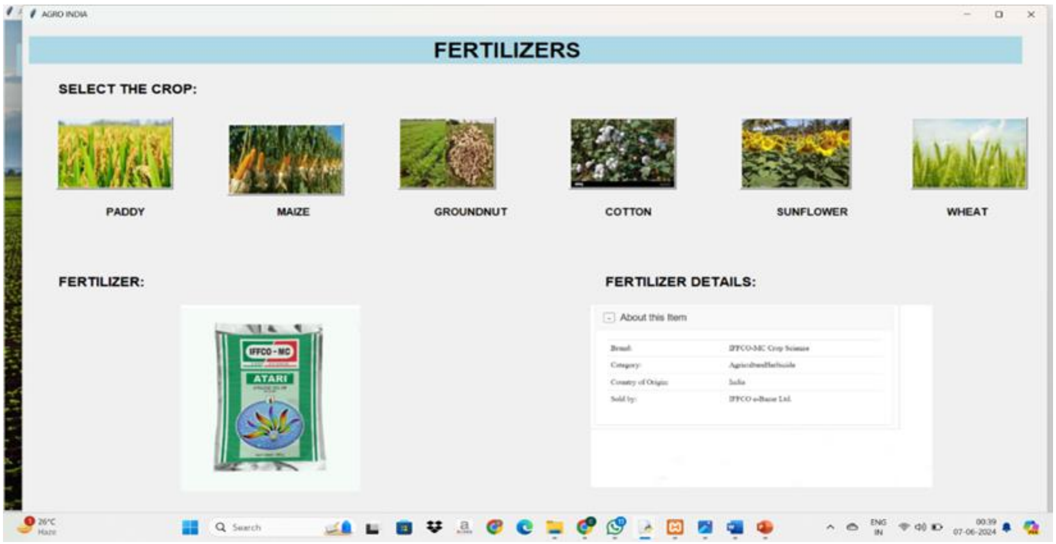

Figure 6.3: Fertilizer suggestion

The Figure 6.3 displays the Fertilizer Recommendation interface where users select a crop to view a corresponding fertilizer image and its detailed product information. It supports visual identification and provides essential specifications to guide the user in agricultural input decisions.

4. Pesticide suggestion

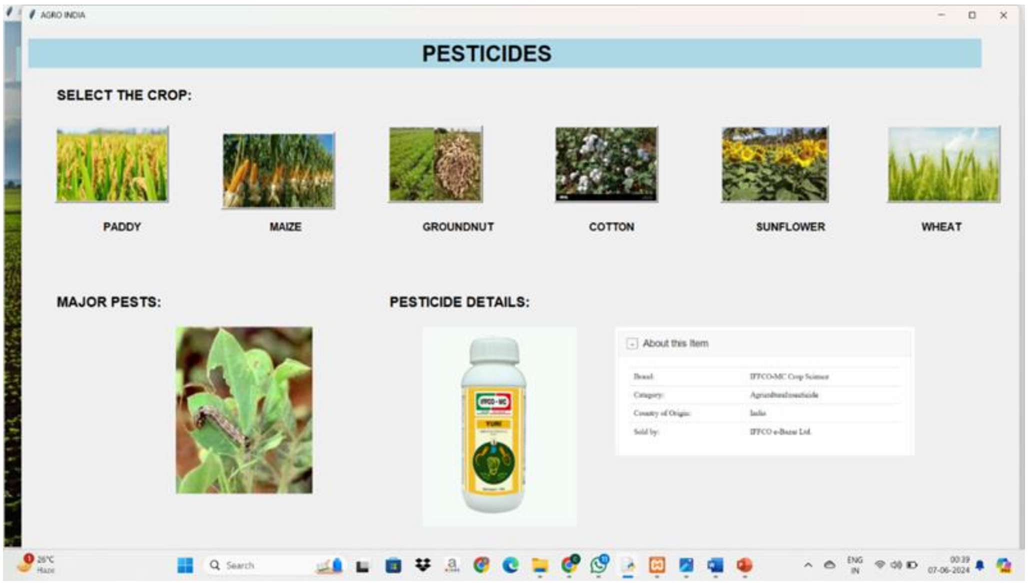

Figure 6.4: Pesticide suggestion

Figure 6.4 shows us the different types of crops and the user needs to select a crop and according to the field selection the fertilizer suggestion is given to us.

5. Crop market rates

Figure 6.5: Crop market rates

This Figure 6.5 displays the "Market Rates" section of an agricultural application. Users can click buttons to view either live market prices or Agmarknet prices for agricultural commodities.

6. Live market prices

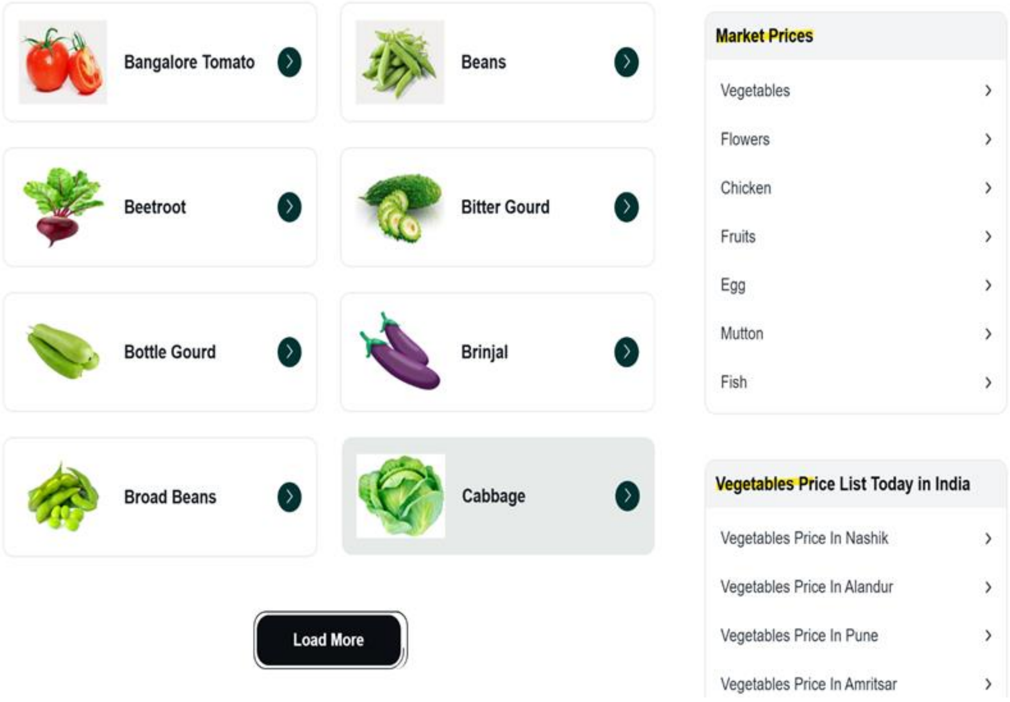

Figure 6.6: Live market prices

This Figure 6.6 displays a categorized list of vegetables with clickable options to view individual market prices. It also offers quick links to check vegetable prices in various Indian cities.

7. Agmarknet

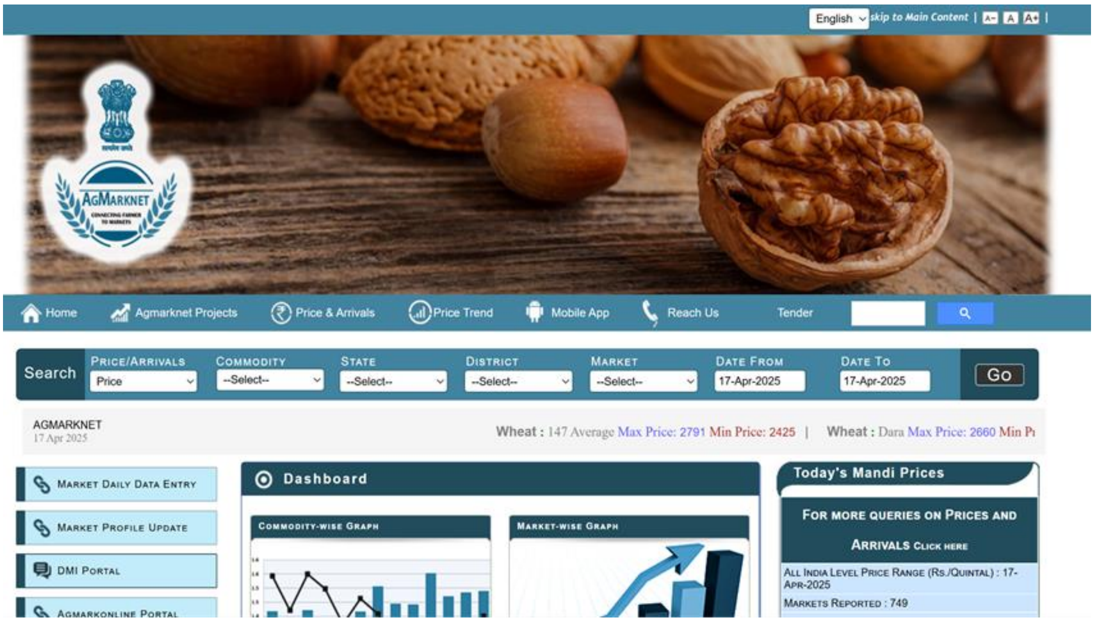

Figure 6.7: Agmarknet

This Figure 6.7 is the Agmarknet portal, providing agricultural market data including prices and arrivals for various commodities. Users can filter by state, district, market, and date to access detailed mandi price trends.

8. Modern technology

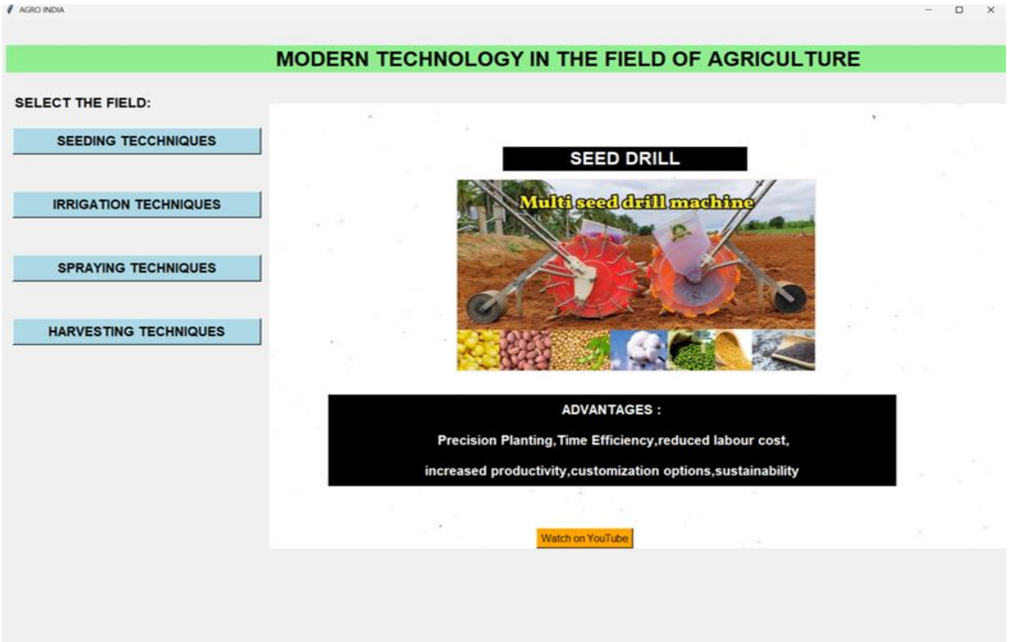

Figure 6.8: Modern technology

This Figure 6.8 highlights modern agricultural technologies, specifically the "Seed Drill" used for precision seeding. It showcases advantages like time efficiency, reduced labor cost, increased productivity, and sustainability.

9. AI based chatbot

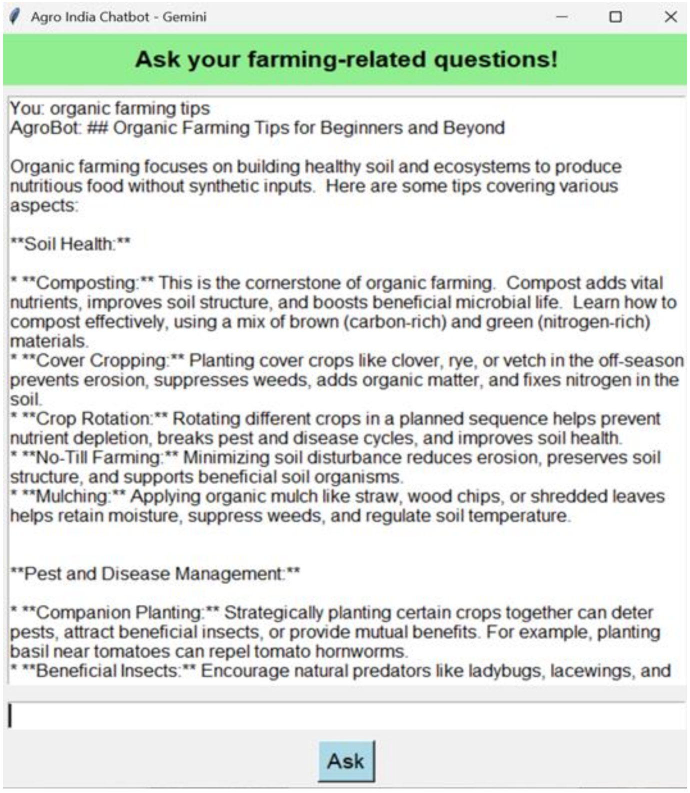

Figure 6.9: AI Chatbot

This Figure 6.9 provides organic farming tips, focusing on soil health and pest management techniques. It includes practices like composting, crop rotation, and companion planting to promote sustainable agriculture.

7.CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

Our farmer guidance web portal revolutionizes agricultural support by offering comprehensive tools like Crop Selection, Fertilizer Selection, and Pesticide Selection, empowering farmers with precise, tailored decisions. With real-time Crop Market Guidance and detailed Modern Machines Information, farmers can optimize profitability and efficiency. By promoting the adoption of modern technologies and sustainable practices, our user-friendly platform enhances farming productivity and sustainability. In summary, our web portal equips farmers with essential resources and knowledge, fostering a resilient and prosperous agricultural future.

7.2 FUTURE SCOPE

The future scope of our farmer guidance web portal includes integrating AI and IoT for precise, real-time decision-making, and blockchain for supply chain transparency. Expanding educational resources, developing a mobile app, and providing global market access can further enhance its utility. Strengthening community features and partnering with agricultural institutions, government agencies, and NGOs will foster collaboration and support. Additionally, incorporating sustainability metrics will help farmers improve their environmental impact. These advancements will ensure our platform remains a vital, innovative tool for sustainable and profitable farming.

REFERENCES

[1] Patel, N., Shah, R., & Dabhi, P. (2018). "Machine Learning for Crop Recommendation and Yield Prediction." In IEEE WiSPNET.

[2] Parikh, S., & Mistry, J. (2017). "Crop Selection Method to Maximize Yield Using Machine Learning Technique." International Journal of Computer Applications.

[3] Kamilaris, A., Kartakoullis, A., & Prenafeta-Boldú, F. (2017). "Smart Farming Based on Cloud Computing and IoT." Computers and Electronics in Agriculture.

[4] European Parliament Research Service. (2016). "Precision Agriculture and the Future of Farming in Europe." Policy Document.

[5] Han, J., & Kamber, M. (2011). "Use of Data Mining Techniques in Agriculture." In Data Mining: Concepts and Techniques.

[6] Sladojevic, S., Arsenovic, M., et al. (2016). "Crop Disease Prediction Using Deep Learning." Computers and Electronics in Agriculture.

[7] Deshmukh, S., Pawar, K., et al. (2018). "Recommendation System for Farmers Using Machine Learning." International Journal of Innovative Research in Computer and Communication Engineering.

[8] Nandurkar, S., Thool, V., et al. (2016). "Smart Farming: IoT-Based Smart Sensors Agriculture Stick." In IEEE ICCIC.

[9] Wolfert, S., Ge, L., et al. (2017). "Big Data in Smart Farming." Agricultural Systems.

[10] Dinesh, V. S., & Murthy, S. (2018). "Soil Parameter Estimation and Crop Recommendation Using Machine Learning." International Research Journal of Engineering and Technology (IRJET).

[11] Ghosh, A., & Banerjee, T. (2017). "Weather-Based Crop Prediction Using Machine Learning." IEEE International Conference on Computing, Communication and Networking Technologies (ICCCNT).

[12] Ramesh, S., & Sakthivel, K. (2015). "Fuzzy Logic Based Expert System for Crop Recommendation." Journal of Artificial Intelligence.

[13] Kaur, R., & Kaur, M. (2019). "A Survey on Applications of Machine Learning in Agriculture." International Journal of Advanced Research in Computer Science.

[14] Patel, M., & Pandya, P. (2018). "Soil Data Analysis Using Machine Learning Algorithms." International Journal for Research in Applied Science and Engineering Technology (IJRASET).

[15] Kulkarni, M., & Waghmare, A. (2020). "Smart Crop Suggestion Using Decision Tree and IoT." International Journal of Science and Research (IJSR).

[16] Singh, V., et al. (2021). "Advances in Precision Agriculture Technologies for Sustainable Crop Production." Elsevier.

[17] Rahman, M. R., et al. (2020). "Cover Crops Effect on Soil Quality and Soil Health." Soil & Tillage Research.

[18] Zhang, J. Y., et al. (2019). "Technology Diffusion, Government Policy and Agricultural Sustainable Development." Agri Policy Journal.

[19] Guan, H. J., et al. (2018). "Optimization Chemical Control in Integrated Pest Management." IPM Journal.

[20] Aravindan, S., & Jyothi, A. M. (2021). "Crop Recommendation Using Hybrid ML Techniques." AI in Agriculture.

APPENDIX A

Source code snippets

```
from tkinter import*
#import mysql.connector
from PIL import Image,ImageTk
import webbrowser
import joblib
import google.generativeai as genai
#import smtplib
top=Tk()
top.geometry("1500x1000")
top.title("AgriWise")

i0=Image.open(r"C:\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\
    images\welcome.png")
p0=ImageTk.PhotoImage(i0)

l=Label(top,image=p0)
l.place(x=0,y=0)

global soil
soil=IntVar()
water=IntVar()
season=IntVar()
global s
global z
global x
global y
z=0
```

```
#homeapge
l=Label(top,text="AgriWise",font="helvetica 30 bold",bg
    ="lightblue",padx=600).place(x=20,y=40)

i=Image.open(r"C:\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\i
    mages\happy_farmer.jpg")
p=ImageTk.PhotoImage(i)

l=Label(top,image=p)
l.place(x=100,y=110)

ii=Image.open(r"C:\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\
    images\logo.jpg")
pp=ImageTk.PhotoImage(ii)

l=Label(top,image=pp)
l.place(x=570,y=110)

ii1=Image.open(r"C
    :\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\images\drone
    .jpg")
pp1=ImageTk.PhotoImage(ii1)

l=Label(top,image=pp1)
l.place(x=900,y=110)
```

```
#chatbot
# ===== CONFIGURE API KEY =====
genai.configure(api_key="AIzaSyB3j6x74WE1pqjOVR2zLtn6Od30_MRmxp0")
# ===== GUI APP FUNCTION =====
def chatbot():
    top = Toplevel()
    top.geometry("600x700")
    top.title("AgriWise Chatbot - Gemini")
Label(top, text="Ask your farming-related questions!", font
    =("Arial", 16, "bold"), bg="lightgreen", pady=10).pack(fill=X)
    text_area = Text(top, wrap=WORD, font=("Arial", 12))
    text_area.pack(padx=10, pady=10, expand=True, fill=BOTH)
    entry = Entry(top, font=("Arial", 14))
    entry.pack(padx=10, pady=5, fill=X)
    # Initialize a **chat session**
    model = genai.GenerativeModel("gemini-1.5-pro-latest")
    #Using a generally available model
    chat = model.start_chat(history=[])
    def ask_gemini():
        question = entry.get()
        if not question.strip():
            return
        text_area.insert(END, f"You: {question}\n")
        try:
            response = chat.send_message(question)
            answer = response.text.strip()
```

```
            text_area.insert(END, f"AgroBot: {answer}\n\n")
        except Exception as e:
            text_area.insert(END, f"Error: {str(e)}\n\n")

        entry.delete(0, END)
Button(top, text="Ask", font=("Arial", 14), bg="lightblue",
    command=ask_gemini).pack(pady=5)
top.mainloop()
```

```
def fertilizer():
    top4=Toplevel()
    top4.geometry("1500x1000")

    top4.title("AgriWise")
    l=Label(top4,text="          FERTILIZERS                ",font
        ="arial 25 bold",fg="black",bg="lightblue",padx=500).place
        (x=10,y=20)
    l=Label(top4,text="SELECT THE CROP:",font="arial 16 bold",fg
        ="black").place(x=50,y=90)
    l=Label(top4,text="FERTILIZER:",font="arial 16 bold",fg
        ="black").place(x=50,y=400)
    l=Label(top4,text="FERTILIZER DETAILS:",font="arial 16 bold"
        ,fg="black").place(x=850,y=400)
```

```
def paddyfertilizer():

    i34=Image.open(r"C
        :\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\image
        s\whitebg.jpg")
    p34=ImageTk.PhotoImage(i34)

    l=Label(top4,image=p34)
    l.place(x=230,y=450)

    i35=Image.open(r"C
        :\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\image
        s\whitebg.jpg")
    p35=ImageTk.PhotoImage(i35)

    l=Label(top4,image=p35)
    l.place(x=830,y=450)

    i34=Image.open(r"C
        :\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\image
        s\paddy_fertilizer_det.jpg"
```

```
webbrowser.open("https://youtu.be/JwWlnYmHlcI?si=KPh2LtvEhXqj6xVL"
    )  # replace with actual link

        Button(top6, text="Watch on YouTube", font=("Arial", 12),
            bg="orange", command=open_spraying_video).place(x=800,
            y=800)

        top6.mainloop()

    def harvesting():

        i36=Image.open(r"C
            :\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\image
            s\whitebg1.jpg")
        p36=ImageTk.PhotoImage(i36)
        l=Label(top6,image=p36)
        l.place(x=400,y=130)
        l=Label(top6,text="MULTI-USE HARVESTORS",font="arial 20
            bold",padx=10,width=20,bg="black",fg="white").place(x
            =750,y=200)
```

```
        i38=Image.open(r"C
            :\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\image
            s\harvesting_technique.jpg")
        p38=ImageTk.PhotoImage(i38)
        l=Label(top6,image=p38)
        l.place(x=680,y=250)

        l=Label(top6,text="ADVANTAGES :\n\n Labor Savings,Cost
            Efficiency,Versatility,\n\n Enhanced Productivity
            ,Technological Integration,Reduced Human Exposure "
            ,font="helvetica 15 bold",pady=10,width=70,bg="black"
            ,fg="white").place(x=490,y=590)
        def open_harvesting_video():
            webbrowser.open("https://youtu.be/kWd_QnyO3eI?si
                =UNrNf09eGazBsO6g")  # replace with actual link

        Button(top6, text="Watch on YouTube", font=("Arial", 12),
            bg="orange", command=open_harvesting_video).place(x
            =800, y=800)

        top6.mainloop()

    b=Button(top6,text="SEEDING TECCHNIQUES",bg="lightblue",font
```

```
    #modern techniques
def modern():
    top6=Toplevel()
    top6.geometry("1500x1000")
    top6.title("AgriWise")
    l=Label(top6,text="MODERN TECHNOLOGY IN THE FIELD OF
        AGRICULTURE",font="billyargel 24 bold",fg="black",bg
        ="lightgreen",padx=400).place(x=10,y=40)
    l=Label(top6,text="SELECT THE FIELD:",font="billyargel 16
        bold",fg="black").place(x=20,y=116)
```

```
def seding():

    i36=Image.open(r"C
        :\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\image
        s\whitebg1.jpg")
    p36=ImageTk.PhotoImage(i36)
    l=Label(top6,image=p36)
    l.place(x=400,y=130)
    l=Label(top6,text="SEED DRILL",font="arial 20 bold",padx
        =10,width=20,bg="black",fg="white").place(x=750,y=200)

    i38=Image.open(r"C
        :\Users\meenu\OneDrive\Pictures\Desktop\iomp\rtp\image
        s\seeding_technique.jpg")
    p38=ImageTk.PhotoImage(i38)
    l=Label(top6,image=p38)
    l.place(x=680,y=250)
    l=Label(top6,text="ADVANTAGES :\n\n Precision Planting
        ,Time Efficiency,reduced labour cost,\n\n increased
        productivity,customization options,sustainability "
        ,font="helvetica 15 bold",width=70,bg="black",fg
        ="white",pady=10).place(x=490,y=590)
```

APPENDIX B

Software Installation :

1. **Install Python** (if not already installed):
   - Download from https://www.python.org
   - Add Python to PATH during installation.
2. **Install Required Libraries**:
   - Open Command Prompt or Terminal.
   - Run the following commands:

```
pip install tkinter
pip install requests
pip install matplotlib
pip install pandas
pip install beautifulsoup4
pip install selenium
```

3. **Install Web Driver (if using Selenium for scraping)**:
   - Download Chrome Driver that matches your browser.
   - Place it in your project directory or add it to PATH.
4. **Install IDE (Optional but Recommended)**:
   - Use **PyCharm**, **VS Code**, or **Thonny** for writing and running your code efficiently.

Project Execution Steps:

Step 1: **Launch the Main Program**:

- Locate your main Python file (e.g., `main.py` or `app.py`).
- Run it using:

```
python main.py
```

Step 2: **User Interface Interaction**:

- GUI (created using Tkinter) will appear.
- Navigate through buttons like:
- View Live Market Prices
- View Agmarknet Prices
- Select Technology Topics
- Chatbot for Farming Help

Step 3: **Data Retrieval:**

- When clicking buttons, the app may fetch:
- Market prices via APIs or web scraping.
- Data visualization using `matplotlib` or charts.

Step 4: **Chatbot Execution**:

- The chatbot window allows the user to type questions.
- Based on keyword detection or AI integration, it displays help content.