Dr. D.Y. Patil School of MCA

Charoli (BK), PUNE- 412105

SAVITRIBAI PHULE PUNE UNIVERSITY

MASTER OF COMPUTER APPLICATION

Project Report on

Smart Agricultural Website

BY

Student Name: Pawar Indrajeet Vasant

Roll No: 25320

Under The Guidance of

Prof. Supriya Ghodekar.

MCA-I(Sem-I)

2025-26

Date: -----/ ----- /2025

Certificate

This is to certify that **Pawar Indrajeet Vasant**, Roll No. **25264** a student at **Dr D Y Patil School of MCA**, has successfully completed the project entitled **KrishiDhan - Smart Agricultural Website** in partial fulfilment of the requirements for the **MCA**-Mini Project (Semester I) during the academic year **2025–2026**.

Prof. Supriya Ghodekar	**Prof. Sapna Chavan**	**Dr E B Khedkar**
Project Guide	**HOD – MCA**	**Director**

ACKNOWLEDGEMENT

I acknowledge to all those who have been so helpful in my academic project work. Nevertheless, I have tried through this report to express my deepest gratitude to all those who have given their precious time, skill, knowledge, valuable advice and guidance and facilities.

At the very outset I take opportunity to express my deepest gratitude and thanks to **Director** of our College **Dr. E. B. Khedkar** and **Prof. Sapna Chavan**–HOD and other teaching and non-teaching staff for their useful guidance during the completion of this project report.

I am highly obliged to **Guide Name** - **Prof. Supriya Ghodekar.** for his valuable guidance and encouragement given to complete the project. His guidance has certainly helped me to simplify the difficulties and finalize the system effectively.

Last but not the least I thank to my almighty God, Parents and friends for their constant support to me in all aspects during work.

Thank You,
Pawar Indrajeet Vasant
MCA – I (Sem-I)
Dr D Y Patil School of MCA

INDEX

Chapter 1: Introduction

1.1 Abstract

Agriculture acts as the primary source of livelihood for a majority of the Indian population; however, the sector faces significant modernization challenges. Farmers often struggle with delayed disease diagnosis, lack of transparency in market pricing, and fragmented supply chains for agricultural inputs.

KrishiDhan is a comprehensive, web-based "Smart Farming" platform designed to bridge this digital divide by integrating advanced technologies into a unified solution. The system is developed using **Python (Flask)** for the backend and utilizes **Artificial Intelligence** and **Data Analytics** to empower farmers.

The core functionalities of the system include an AI-powered **"Plant Doctor"** that uses image processing (Deep Learning) to detect crop diseases instantly from user-uploaded photos. Additionally, the platform features a **Data Visualization Module** that renders real-time Mandi market trends and weather updates to aid in decision-making. To support the economic aspect of farming, KrishiDhan includes a robust **E-commerce Marketplace** that connects farmers directly with suppliers, eliminating middlemen. By offering these services through an accessible, multilingual interface with **Voice Assistant** capabilities, KrishiDhan aims to enhance productivity, ensure food security, and maximize profitability for the Indian farming community.

1.2 Existing System and Need for the System

Existing System

The traditional system described above suffers from several critical drawbacks:

- **Time Delay:** The turnaround time for identifying plant diseases via physical labs is too long. By the time a diagnosis is received, the infection may have already spread.
- **Information Asymmetry:** Farmers lack access to real-time market data, putting them at a disadvantage during price negotiations.
- **Geographical Barriers:** Experts and quality agricultural inputs are not easily accessible to farmers in remote or rural locations.
- **Lack of Personalization:** Broadcast media (TV/Radio) provides generic advice that may not be specific to the farmer’s soil type, location, or specific crop variety.

Need for the Proposed System (KrishiDhan)

To overcome these limitations, there is a critical need for an integrated, web-based "Smart Farming" platform. **KrishiDhan** is proposed to fill this gap by leveraging modern computing technologies such as Artificial Intelligence, Data Analytics, and Web Development.

The specific needs addressed by the system include:

1. **Automation of Diagnostics:** There is a need for an automated "Plant Doctor" that uses Computer Vision (AI) to provide instant disease diagnosis and treatment recommendations from a simple photo upload.
2. **Centralization of Services:** Farmers need a "One-Stop Solution" that combines diagnostics, weather updates, and an e-commerce marketplace, reducing the need to visit multiple physical locations.
3. **Data-Driven Decision Making:** The system is needed to democratize access to data analytics. By visualizing market trends (using Python libraries), the system helps farmers decide the optimal time to sell their produce.
4. **Accessibility and Inclusion:** To ensure the system is usable by farmers with limited digital literacy, there is a need for inclusive features such as a Voice Assistant and multi-lingual support, which are absent in most existing corporate agricultural tools.

1.3 Scope of the System

The scope of the **KrishiDhan** project is defined by its functional boundaries, which encompass the development of a web-based integrated platform for smart farming. The system is designed to provide end-to-end digital services to farmers, ranging from crop diagnostics to market transactions.

The specific functional scope is divided into the following five key modules:

1. User Administration Module

This module handles the security and access control of the system.

- **Role-Based Access Control (RBAC):** The system distinguishes between two primary user roles:
  - **Farmers:** Have access to the front-end features (Shop, Plant Doctor, Weather).

- **Administrators:** Have access to the back-end dashboard to manage products, view user statistics, and maintain system health.
- **Authentication:** Implementation of secure registration and login mechanisms using password hashing (SHA-256) to protect user data.

2. AI Diagnostic Module (The "Plant Doctor")

This module defines the scope of the Artificial Intelligence integration.

- **Image Acquisition:** The system allows users to capture or upload high-resolution images of plant leaves.
- **Disease Detection:** Utilization of a **Convolutional Neural Network (CNN)** model to analyze the uploaded images and classify them into specific disease categories (e.g., Bacterial Blight, Rust, or Healthy).
- **Remedy Recommendation:** Upon diagnosis, the system automatically suggests appropriate treatments or fertilizers available in the integrated shop.

3. E-Commerce Module

This module covers the scope of the online marketplace features.

- **Product Cataloging:** A dynamic inventory system where products (seeds, fertilizers, tools) are categorized and displayed with prices and stock levels.
- **Shopping Cart Management:** Functionality for users to add multiple items to a digital cart, update quantities, and view real-time total costs.
- **Order Placement:** A streamlined checkout process that generates an order summary and updates the database inventory (simulating the purchase process).

4. Information and Analytics Module

This module focuses on data presentation and visualization.

- **Weather Integration:** Integration with third-party APIs (e.g., OpenWeatherMap) to fetch and display real-time weather conditions and forecasts based on the user's location.
- **Market Trend Analysis:** The scope includes a dedicated analytics section where historical Mandi price data is processed using **Python (Matplotlib/Seaborn)** to generate visual graphs. This helps farmers understand price fluctuations over time.

5. Accessibility Module

To ensure the system is usable by the target demographic (rural farmers), the scope includes specific accessibility features:

- **Voice Command Integration:** A voice-enabled assistant that allows users to navigate the website and query information using speech input, reducing the reliance on typing.
- **Localization (Multi-Language Support):** The system includes a translation layer (via Google Translate API) to render text in local regional languages, making the content intelligible to non-English speakers.

1.4 Operating Environment – Hardware and Software

To ensure the effective development and deployment of the KrishiDhan Smart Farming platform, specific hardware and software configurations are required. As the system involves AI and image processing, the requirements are slightly higher than a standard text-based website.

Hardware Requirements

- **Processor:** Intel Core i5 or higher (Recommended for training/running AI models)
- **RAM:** Minimum 8 GB (Required for efficient processing of TensorFlow/Keras libraries)
- **Hard Disk:** Minimum 256 GB SSD (For faster data retrieval and dataset storage)
- **Internet Connection:** High-speed connectivity (Required for API fetching and cloud database operations)
- **Input Device:** Webcam or Smartphone Camera (For testing the "Plant Doctor" image upload feature)

Software Requirements

- **Operating System:** Windows 10/11 (64-bit) or Linux (Ubuntu 20.04+)
- **Web Browser:** Google Chrome, Mozilla Firefox, or Microsoft Edge (Latest versions)
- **Code Editor / IDE:** Visual Studio Code (VS Code) or PyCharm
- **Programming Language:** Python 3.8 or higher
- **Web Server Framework:** Flask (Python)
- **Database Management System:** MySQL Server
- **AI/ML Libraries:** TensorFlow, Keras, NumPy, Pandas
- **Version Control:** Git

1.4.1 Brief Description of Technology Used

The system follows a standard **Client-Server Architecture** designed for scalability and modularity.

- **Frontend (Client-Side):**
  - **HTML5 & CSS3:** Used to build the structural layout and styling of the web pages, ensuring a clean and accessible user interface.
  - **JavaScript:** Implements client-side logic, such as dynamic cart updates, form validation, and handling the Voice Assistant integration.
  - **Bootstrap:** Utilized to ensure the website is "Mobile-Responsive," allowing farmers to access it easily on smartphones.
- **Backend (Server-Side):**
  - **Python:** The core programming language used for the server logic due to its extensive support for data science and AI libraries.
  - **Flask:** A lightweight micro-framework used to handle HTTP requests, routing (URLs), and rendering templates. It acts as the bridge between the user interface and the database/AI models.
- **Database:**
  - **MySQL:** A Relational Database Management System (RDBMS) used to store structured data such as user profiles, product inventory, order history, and market pricing data.

- **Artificial Intelligence & Data Science:**
  - **TensorFlow & Keras:** Used to build and run the **Convolutional Neural Network (CNN)** model that powers the "Plant Doctor" for disease detection.
  - **Matplotlib & Seaborn:** Python libraries used to process historical market data and generate visual graphs (line charts/bar graphs) for the "Market Trends" feature.

1.5 Operating Systems and Database Used

1.5.1 Operating Systems Used

- **Windows 10/11 Operating System** Used for the initial development, coding, and testing phases due to its user-friendly interface and robust support for IDEs like Visual Studio Code.
- **Linux (Ubuntu) Operating System** Recommended for the final deployment and production server environment. It is chosen for its superior stability, enhanced security features, and seamless optimization with Python web servers (such as Gunicorn and Nginx).

1.5.2 Database Used

- **PostgreSQL** PostgreSQL is a powerful, open-source Object-Relational Database Management System (ORDBMS).

Rationale: It was selected for **KrishiDhan** because:

1. **Robustness**: It is highly reliable and ACID compliant, ensuring that critical transactional data (like User Orders and Payments) is never lost or corrupted.
2. **Advanced Capabilities:** Unlike standard SQL, PostgreSQL supports advanced data types (like JSONB) and complex queries, which are beneficial for handling the analytical data required for the "Market Trends" and "Crop Recommendations" features.
3. **Python Integration:** It integrates seamlessly with Python (via the psycopg2 adapter) and handles concurrent users efficiently.

Chapter 2: Proposed System

2.1 Proposed System Overview

KrishiDhan is designed as a modular, scalable, and user-centric "Smart Farming" ecosystem. Unlike traditional agricultural portals that only provide static information, KrishiDhan is an **action-oriented platform** that integrates Artificial Intelligence (AI) and Data Science to solve real-world problems on the field.

The system is structured around four pillars that handle the complete lifecycle of a farming season:

1. The Diagnostic Pillar (AI Plant Doctor)

At the heart of KrishiDhan is the **Plant Doctor** module. This feature uses a **Convolutional Neural Network (CNN)**—a type of Deep Learning model—to identify crop diseases. Farmers simply take a photo of an infected leaf or stem and upload it. The system then:

- Analyzes patterns, colors, and textures to identify the specific pathogen (fungal, bacterial, or viral).
- Provides an instant diagnosis with a confidence score.
- Offers a curated list of chemical or organic treatments available in the e-commerce store.

2. The Analytical Pillar (Market & Weather Intelligence)

This module transforms raw data into **actionable intelligence**. Using Python's **Pandas** and **Matplotlib** libraries, the system processes historical and live Mandi (market) price data.

- **Visual Trends:** Instead of complex tables, farmers see intuitive line graphs showing price fluctuations over weeks or months, helping them decide whether to sell immediately or store their harvest.
- **Precision Weather:** Integration with OpenWeatherMap API provides micro-local forecasts, allowing farmers to time their irrigation and pesticide spraying with precision.

3. The Transactional Pillar (E-Commerce Marketplace)

To eliminate exploitative middlemen, KrishiDhan hosts a direct **Agri-Shop**.

- **Supply Chain Transparency:** Farmers can buy seeds, tools, and fertilizers directly at transparent prices.
- **Dynamic Inventory:** The system manages real-time stock levels and provides a seamless "Cart-to-Checkout" experience, specifically tailored for low-bandwidth rural internet connections.

4. The Accessibility Pillar (Multilingual Voice Assistant)

Recognizing that digital literacy varies, KrishiDhan includes a **Voice-First interface**.

- **NLP & Speech-to-Text:** Using Natural Language Processing, farmers can navigate the site by speaking in their native language (e.g., "Show me the price of Wheat today").
- **Localization:** The entire UI can be toggled between English and regional languages, ensuring that technology serves the farmer, rather than becoming a barrier.

2.2 Feasibility Study

A feasibility study is a crucial phase in the software development life cycle (SDLC) used to evaluate the practicality of the proposed project. For **KrishiDhan**, the study was conducted across three primary dimensions: Technical, Economic, and Operational.

2.2.1 Technical Feasibility

The technical feasibility assesses whether the current technical resources are sufficient to build and maintain the system.

- **Technology Stack:** The project leverages **Python**, which is the industry standard for AI and Data Science. The **Flask** framework provides a lightweight yet powerful backend that is easy to scale.
- **Database Management:** By using **PostgreSQL**, the system ensures high data integrity and the ability to handle complex relational queries for market analytics.
- **AI Integration:** The use of **TensorFlow** and **Keras** allows for the implementation of state-of-the-art Convolutional Neural Networks (CNN) for image recognition, which is essential for the "Plant Doctor" feature.

- **Scalability:** The system is built on a modular architecture, meaning new features (like IoT sensor integration) can be added in the future without a complete system overhaul.
- **Hardware Availability:** The development does not require specialized hardware; standard high-performance PCs are sufficient for training the models.

2.2.2 Economic Feasibility

This study evaluates the cost-effectiveness of the project to ensure that the benefits outweigh the investment.

- **Zero Licensing Fees:** By choosing an open-source stack (Python, Flask, PostgreSQL, and Linux), the project avoids expensive software licensing costs.
- **Low Infrastructure Costs:** The website can be hosted on affordable cloud platforms (like AWS or Heroku). Since it is a web application, there is no cost associated with distributing the software to users.
- **Value Addition:** The system provides high economic value to farmers by reducing crop loss through early disease detection and increasing profit margins by eliminating middlemen in the e-commerce module.
- **Development Efficiency:** Using pre-trained models for AI (Transfer Learning) significantly reduces the time and cost required for research and development.

2.2.3 Operational Feasibility

Operational feasibility measures how well the proposed system will be accepted by the users and integrated into their daily activities.

- **Accessibility Features:** The inclusion of a **Voice Assistant** and **Multi-language support** directly addresses the potential barrier of low digital literacy among the farming community.
- **Simplified Workflow:** The "Plant Doctor" transforms a complex scientific process into a simple "Upload and Result" workflow that any user with a smartphone can navigate.
- **User Motivation:** Farmers are highly motivated to use the system because it solves their two biggest pain points: crop health and market pricing.
- **Administrative Ease:** The backend dashboard allows administrators to manage inventory and view market trends with minimal technical training, ensuring the system can be maintained by non-developers.

2.3 Objectives of the Proposed System

Here is the detailed content for **Section 2.3: Objectives of the Proposed System**. I have expanded the points you provided into a formal professional format, emphasizing the measurable goals and the specific technological impact of the **KrishiDhan** project.

2.3 Objectives of the Proposed System

The primary objective of the **KrishiDhan** platform is to create a technology-driven agricultural ecosystem that empowers farmers with data, diagnostics, and direct market access. The specific objectives defined for this system are as follows:

1. AI-Driven Disease Diagnosis and Management

- **High Accuracy Detection:** To implement a Convolutional Neural Network (CNN) model capable of detecting crop diseases with a target accuracy of **over 85%**.
- **Instant Response:** To provide immediate diagnosis upon image upload, significantly reducing the turnaround time compared to manual or laboratory-based inspection.
- **Remedy Integration:** To automatically link diagnosed diseases with specific treatment recommendations and direct product links within the e-commerce shop.

2. Elimination of Market Intermediaries

- **Direct Marketplace:** To establish a transparent B2C (Business-to-Consumer/Farmer) e-commerce platform that allows farmers to purchase high-quality seeds, fertilizers, and tools directly from verified suppliers.
- **Price Transparency:** To reduce the dependency on middlemen and brokers who often inflate costs, thereby improving the farmer's profit margins and purchasing power.
- **Inventory Management:** To provide a real-time stock management system for agricultural inputs to ensure availability during peak sowing seasons.

3. Data-Driven Agricultural Intelligence

- **Graphical Market Analytics:** To utilize Python-based data science libraries (**Pandas and Matplotlib**) to transform raw Mandi price data into intuitive, easy-to-understand visual graphs.
- **Strategic Decision Support:** To help farmers identify historical price trends, enabling them to make informed decisions on the best timing for harvesting and selling their produce.
- **Localized Weather Alerts:** To provide real-time, location-specific weather forecasting to assist in planning irrigation and pesticide applications.

4. Inclusive Accessibility and User Experience

- **Vernacular Interface:** To overcome language barriers by providing multi-lingual support (Localization), ensuring that technology is accessible to farmers in their native languages.
- **Voice-Enabled Navigation:** To implement a **Voice Assistant** that allows users with low digital literacy to interact with the website using simple voice commands rather than complex menu navigation.
- **Mobile-First Design:** To ensure the web application is fully responsive and optimized for low-bandwidth mobile networks common in rural areas.

5. Security and Data Integrity

- **Role-Based Access Control:** To implement a secure authentication system that provides customized dashboards for both farmers (seekers) and administrators (managers).
- **Transaction Reliability:** To utilize **PostgreSQL** to maintain strict data integrity for all e-commerce transactions, orders, and user diagnostic history.

2.4 Users of the System

The **KrishiDhan** platform is designed with a role-based access control (RBAC) architecture to ensure that each user interacts with a personalized interface suited to their needs. The system primarily supports two categories of users:

1. Farmer (The Primary End-User)

The Farmer role is the heart of the application, intended for users who require on-ground agricultural assistance. The interface for this user is optimized for simplicity, featuring larger icons and voice-enabled navigation.

Key Responsibilities and Actions:

- **Profile Management:** Register and maintain a personal profile, including location data to receive localized weather and market alerts.
- **Diagnostic Tool (Plant Doctor):** Capture or upload images of distressed crops to receive AI-generated disease diagnosis and treatment suggestions.
- **Market Intelligence:** Access and interpret graphical data representing live and historical Mandi prices for various crops to strategically plan sales.
- **Direct Procurement:** Browse the integrated e-commerce shop, manage a shopping cart, and place orders for seeds, fertilizers, and tools.
- **Voice-Guided Navigation:** Use the multi-lingual voice assistant to query prices, weather updates, or specific crop information without the need for manual typing.
- **Real-time Planning:** Monitor hyper-local weather forecasts to determine optimal times for irrigation or pesticide application.

2. Administrator (The System Manager)

The Administrator role is responsible for the operational management of the platform. The Admin has access to a secure backend dashboard that oversees the entire ecosystem.

Key Responsibilities and Actions:

- **Inventory Management:** Add new agricultural products to the shop, update prices, manage stock levels, and categorize items based on crop types.
- **Data Oversight:** Monitor and update the market pricing database (PostgreSQL) to ensure the analytics module displays accurate and current information.
- **User Management:** Oversee farmer registrations, manage account security, and handle any user-related support queries or complaints.
- **Content Curation:** Update the "Agro-Advisory" section with the latest government schemes, seasonal tips, and expert recommendations.
- **System Monitoring:** Track platform performance, analyze sales reports from the e-commerce module, and monitor the accuracy of the AI diagnostic model.

Chapter 3: Analysis and Design

3.1 System Requirements

System requirements define the specific services the KrishiDhan platform must provide and the constraints under which it must operate to ensure a seamless experience for the farming community.

3.1.1 Functional Requirements

Functional requirements define the core internal processes and "what" the system does to meet user needs.

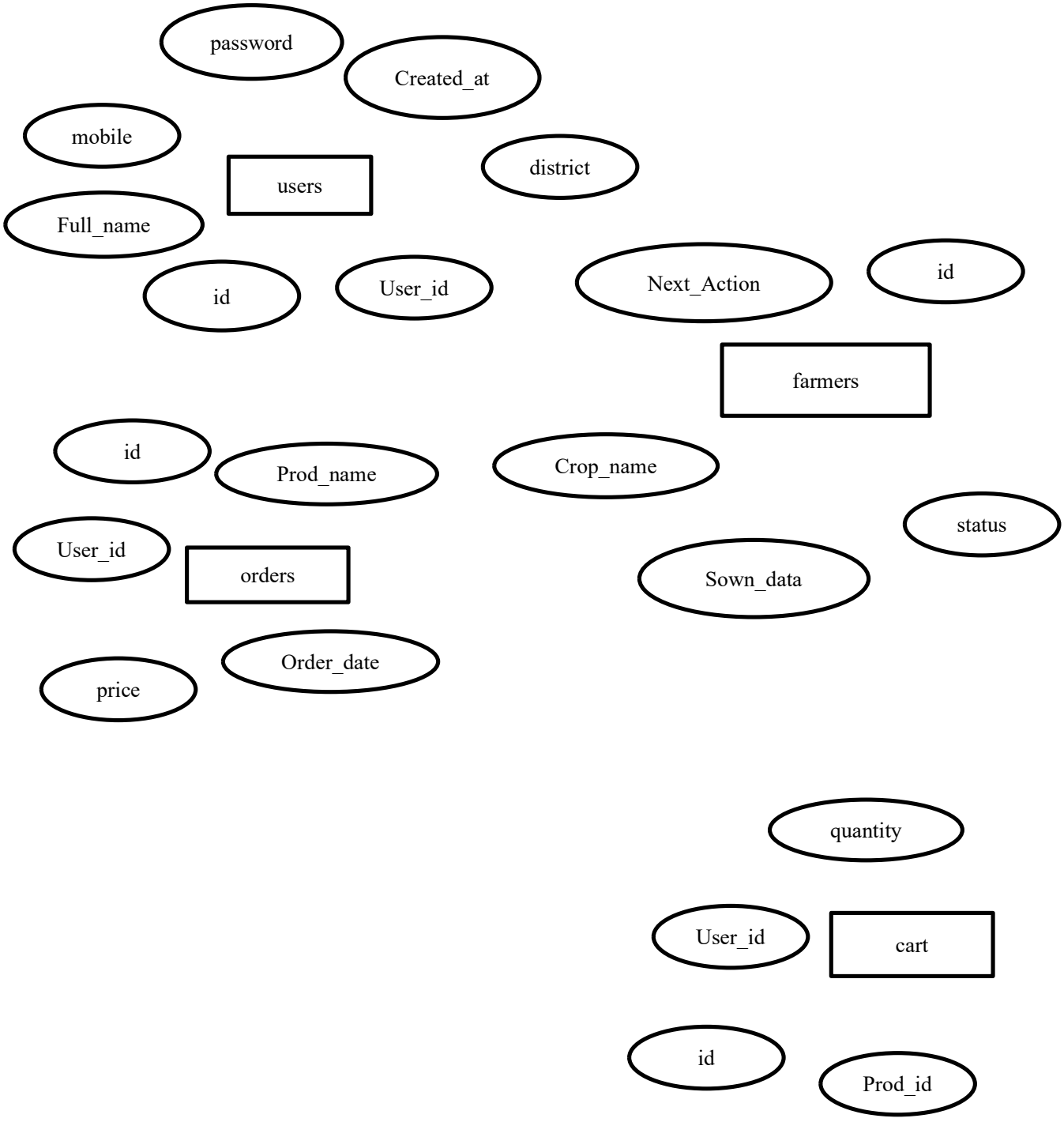
- **User Authentication & Authorization:**
  - The system must allow new users to register via a secure sign-up form.
  - The system must provide a login gateway that distinguishes between **Farmer** and **Admin** roles, redirecting them to their respective dashboards.
- **AI Plant Doctor (Diagnostic Tool):**
  - The system must provide an interface for users to upload or capture images of infected plant leaves.
  - The backend must process the image using a **CNN (Convolutional Neural Network)** and return a specific disease name and treatment plan.
- **Mandi Market Intelligence:**
  - The system must fetch historical and live market data.
  - The platform must utilize **Matplotlib** to render interactive graphs showing price fluctuations for different crops.
- **E-Commerce Management:**
  - The platform must allow farmers to browse an updated catalog of seeds, fertilizers, and tools.
  - The system must support "Add to Cart," "Remove from Cart," and "Checkout" functionalities.
  - The database must automatically update stock quantities after a successful order is placed.
- **Accessibility Features:**
  - The system must integrate a **Voice Assistant** to allow navigation via speech-to-text triggers.
  - The interface must allow users to toggle between English and regional languages.

3.1.2 Non-Functional Requirements

Non-functional requirements describe the quality attributes and performance constraints of the platform.

- **Usability**:
  - The user interface (UI) must be intuitive, using large icons and high-contrast text to accommodate users with varying digital literacy.
  - The website must be fully responsive (Mobile-First) to work on basic smartphones used in rural areas.
- **Performance & Latency**:
  - The AI diagnosis result must be generated and displayed within 5 seconds of the image upload.
  - Data visualization graphs must load within 2 seconds on a standard 3G/4G connection.
- **Security:**
  - User passwords must be encrypted using PBKDF2 with SHA-256 hashing.
  - PostgreSQL database connections must be secured to prevent SQL injection and unauthorized data access.
- **Reliability:**
  - The system should maintain 99.9% uptime to ensure farmers can access weather and market data at any time.
  - The database must ensure transactional integrity (ACID properties) so that order data is never lost during high traffic.
- **Scalability:**
  - The backend architecture (Flask) and database (PostgreSQL) must be capable of handling an increase in concurrent users and an expanding product inventory without significant performance degradation.

3.2Entity Relationship Diagram (ERD)



3.3 Table Structure:

KrishiDhan – Database Schema Documentation

1. Table: users

Description: Stores information for registered farmers.

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key	Unique ID for the user
full_name	String	NOT NULL	User's full name
mobile	String	UNIQUE, NOT NULL	User's mobile number (used for login)
password	String	NOT NULL	Hashed password for security
district	String	—	User's district
state	String	—	User's state
created_at	Timestamp	—	Date and time the account was created

2. Table: admins

Description: Stores credentials for platform administrators.

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key	Unique ID for the admin
username	String	UNIQUE, NOT NULL	Admin login username
password	String	NOT NULL	Hashed password
full_name	String	—	Admin's full display name

3. Table: products

Description: Inventory for the Agri Shop (Seeds, Tools, Fertilizers).

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key	Unique product ID
name	String	NOT NULL	Name of the product
category	String	—	Seeds / Fertilizer / Tools
price	Decimal	NOT NULL	Cost per unit
image_url	String	—	Product image URL
description	Text	—	Product details
stock	Integer	NOT NULL	Available quantity
image	String	Legacy	Older column (optional/unused)

4. Table: cart

Description: Stores items currently added to a user's shopping cart.

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key	Unique cart entry ID
user_id	Integer	Foreign Key	References users(id)
product_id	Integer	Foreign Key	References products(id)
quantity	Integer	NOT NULL	Quantity selected

---

5. Table: orders

Description: Maintains history of completed purchases.

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key	Unique order ID
user_id	Integer	Foreign Key	References users(id)
product_name	String	NOT NULL	Product name at purchase time
price	Decimal	NOT NULL	Price paid
order_date	Timestamp	—	Date of order

---

6. Table: farmer_crops

Description: Tracks crops currently grown by a farmer ("My Farm").

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key	Unique record ID

Column Name	Data Type	Constraints	Description
user_id	Integer	Foreign Key	References users(id)
crop_name	String	NOT NULL	Name of the crop
sown_date	Date	—	Date of sowing
status	String	—	Current crop stage
next_action	String	—	Next recommended action

7. Table: crops

Description: General crop knowledge base for the Crop Info section.

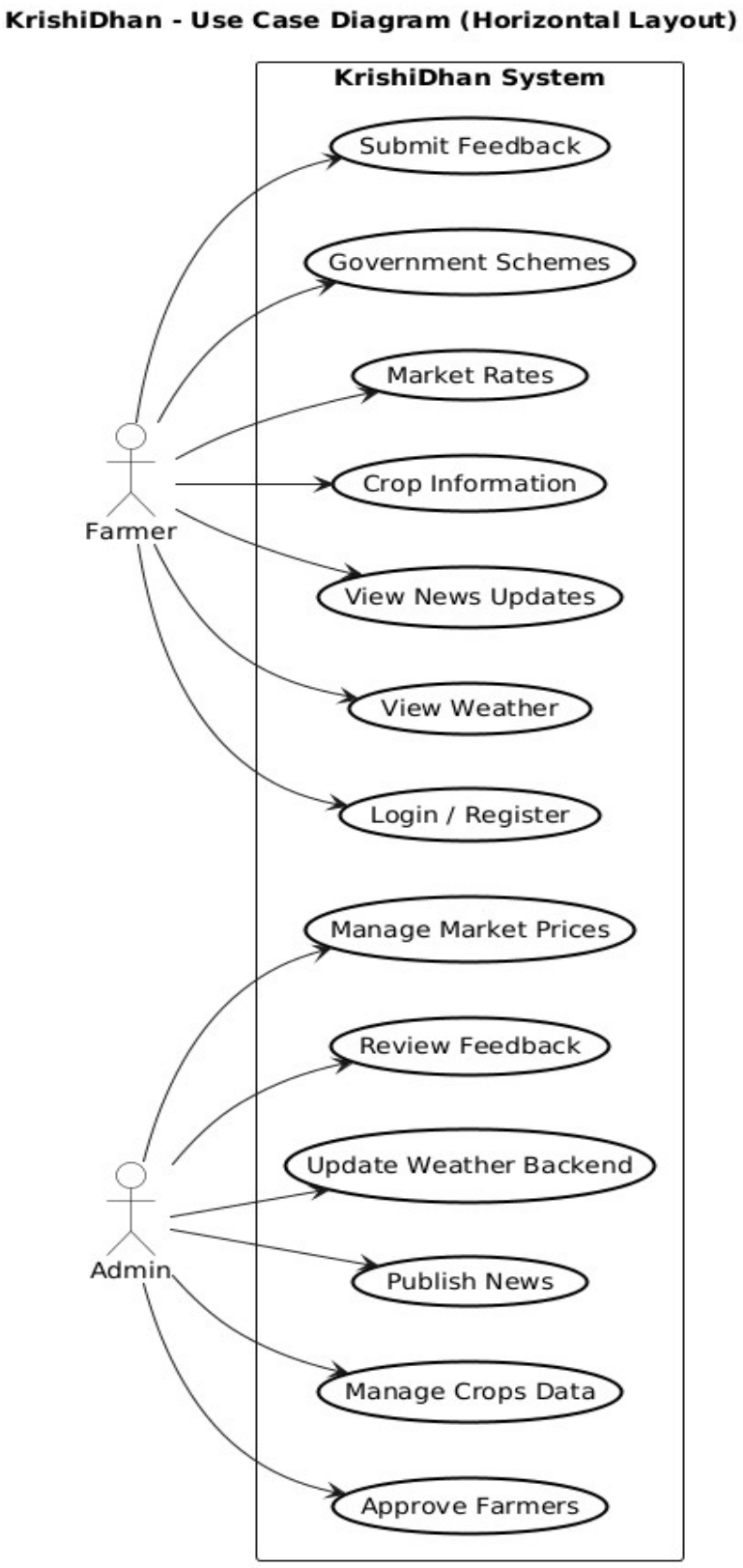
Column Name	Data Type	Constraints	Description
id	Integer	Primary Key	Unique crop ID
name	String	NOT NULL	Common crop name
category	String	—	Cereal, Pulse, Vegetable
image_url	String	—	Crop image
description	Text	—	Detailed crop information
soil_type	String	—	Suitable soil type
season	String	—	Rabi / Kharif / Zaid
growth_stage	String	—	Growth stage details
created_at	Timestamp	—	Record creation date

8. Table: market_prices

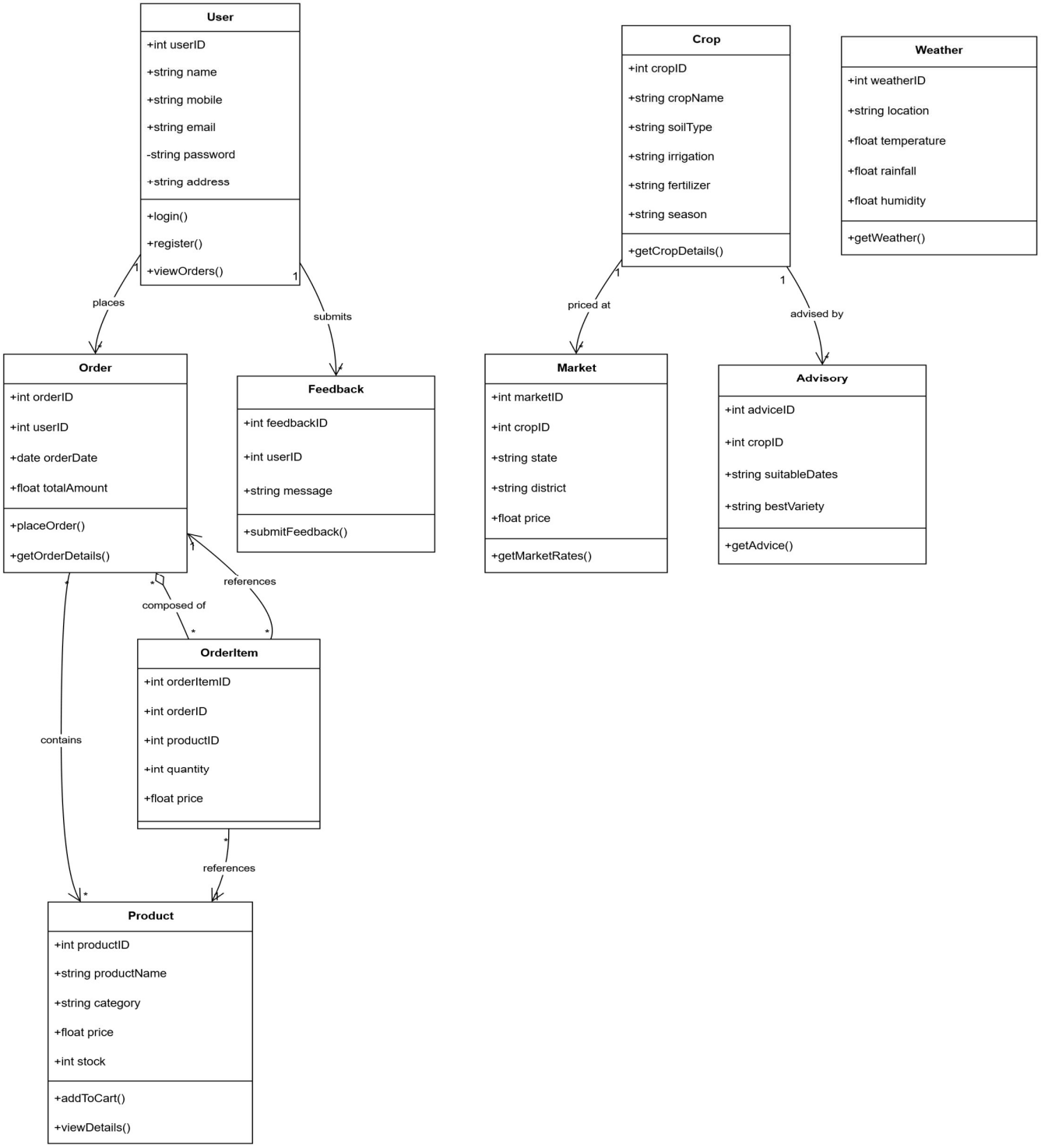
Description: Stores mandi market prices for analytics and trends.

Column Name	Data Type	Constraints	Description
id	Integer	Primary Key	Unique price record ID
crop_id	Integer	Foreign Key	References crops(id)
state	String	—	State name
district	String	—	District name
price_per_quintal	Decimal	NOT NULL	Market price
date	Date	NOT NULL	Date of price record

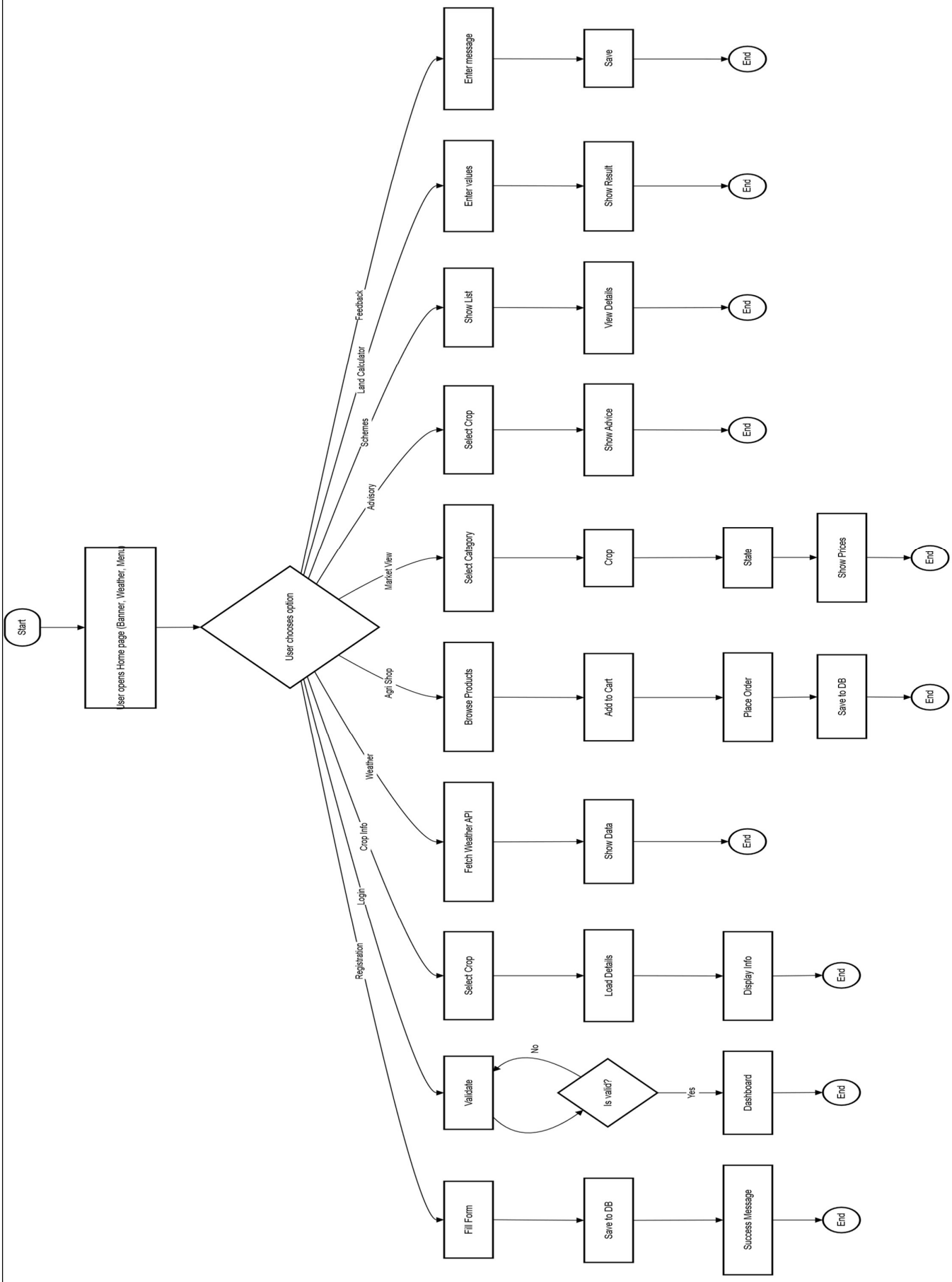
3.4 Use Case Diagrams



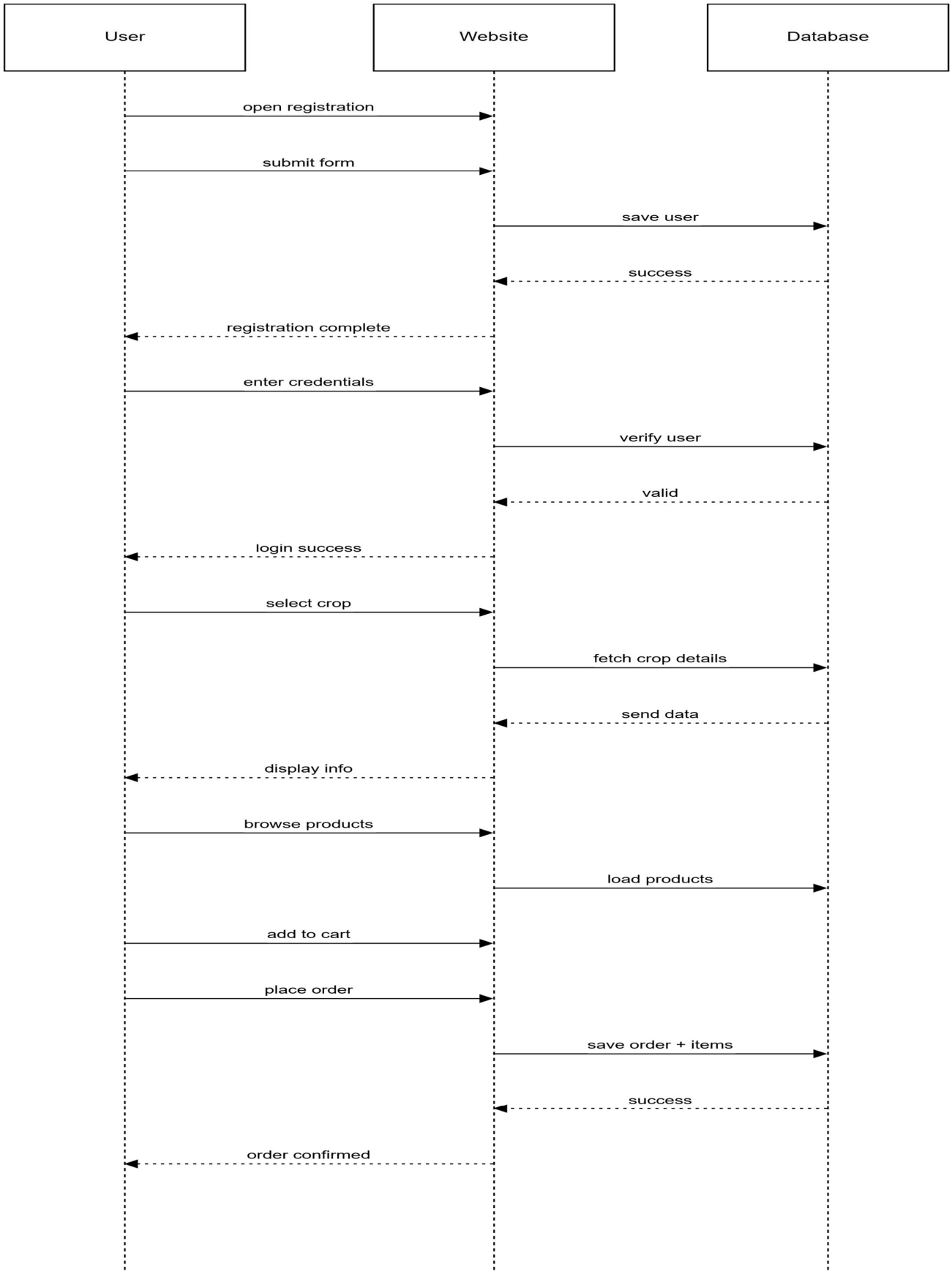
3.5 Class Diagram:



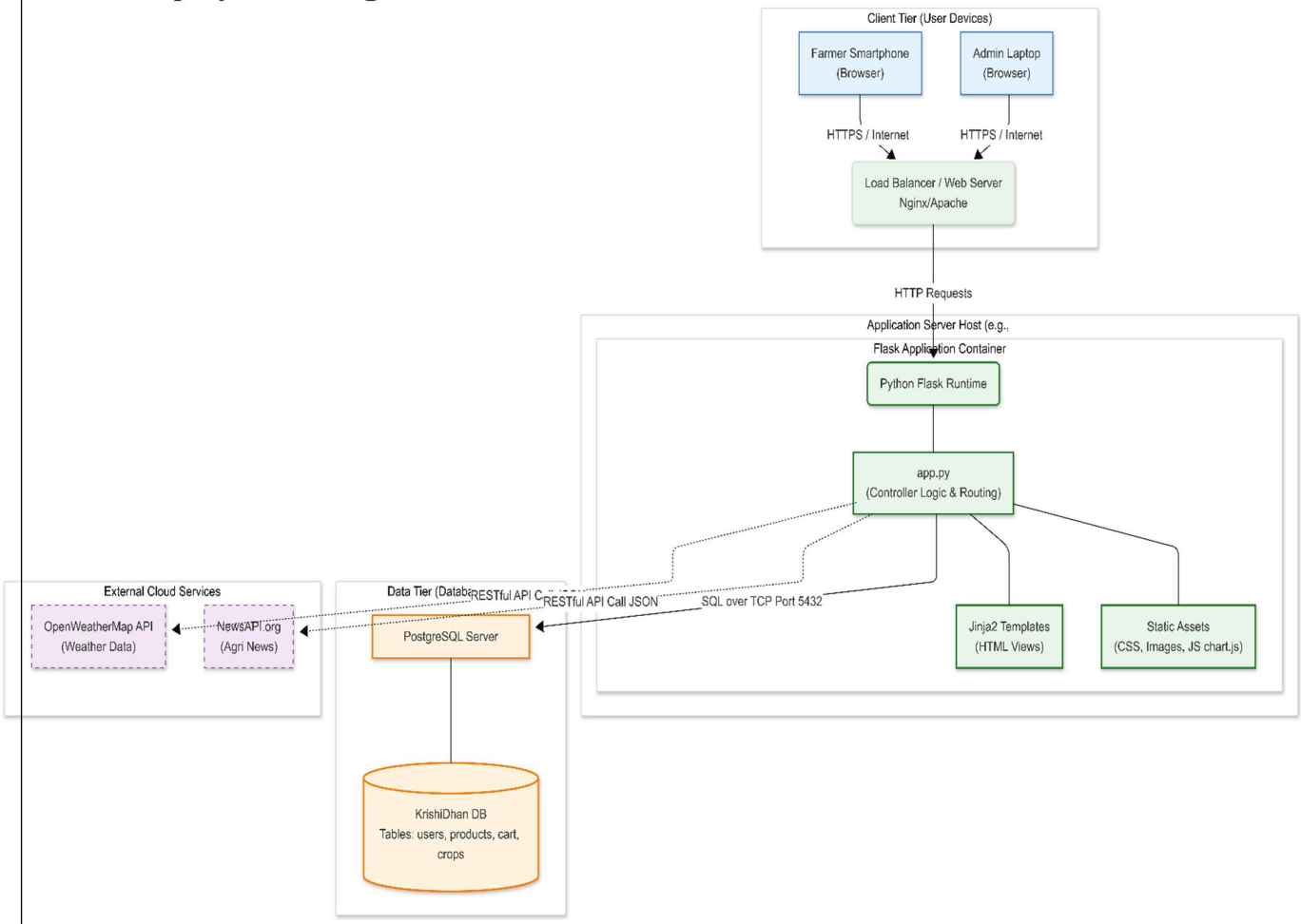
3.6 Activity Diagram :



3.7 Sequence Diagram:

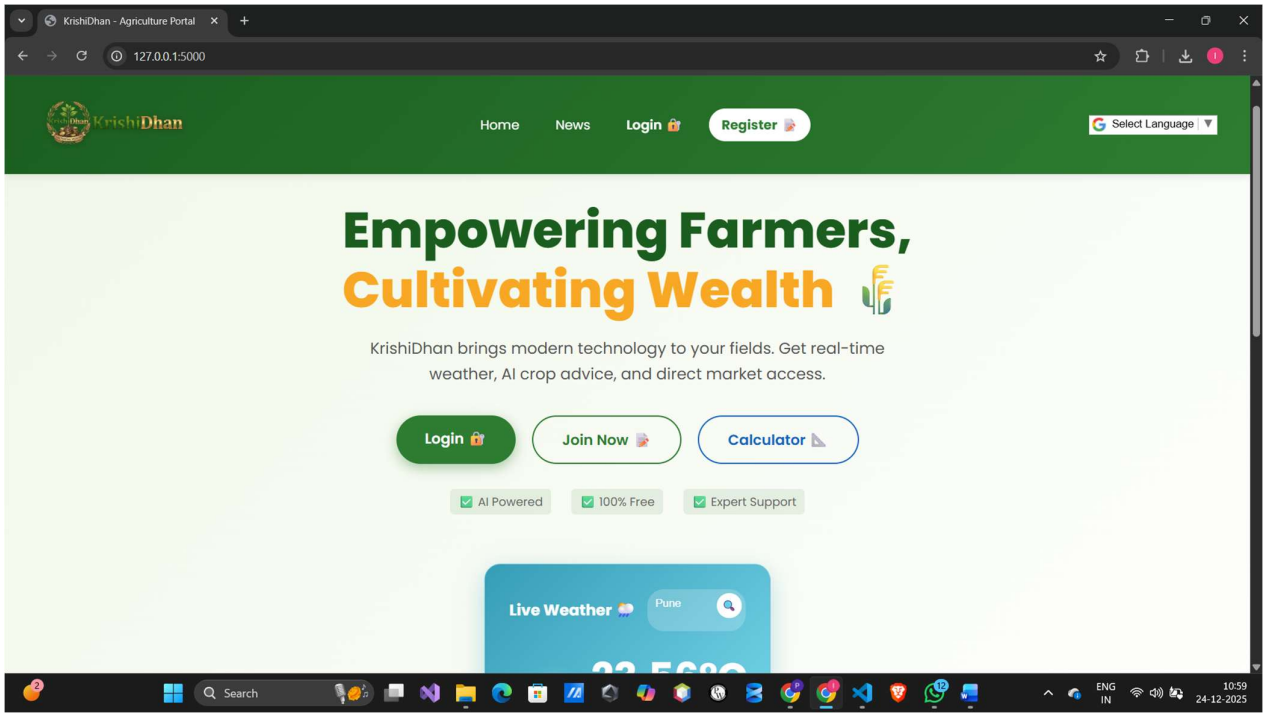


3.8 Deployment Diagram

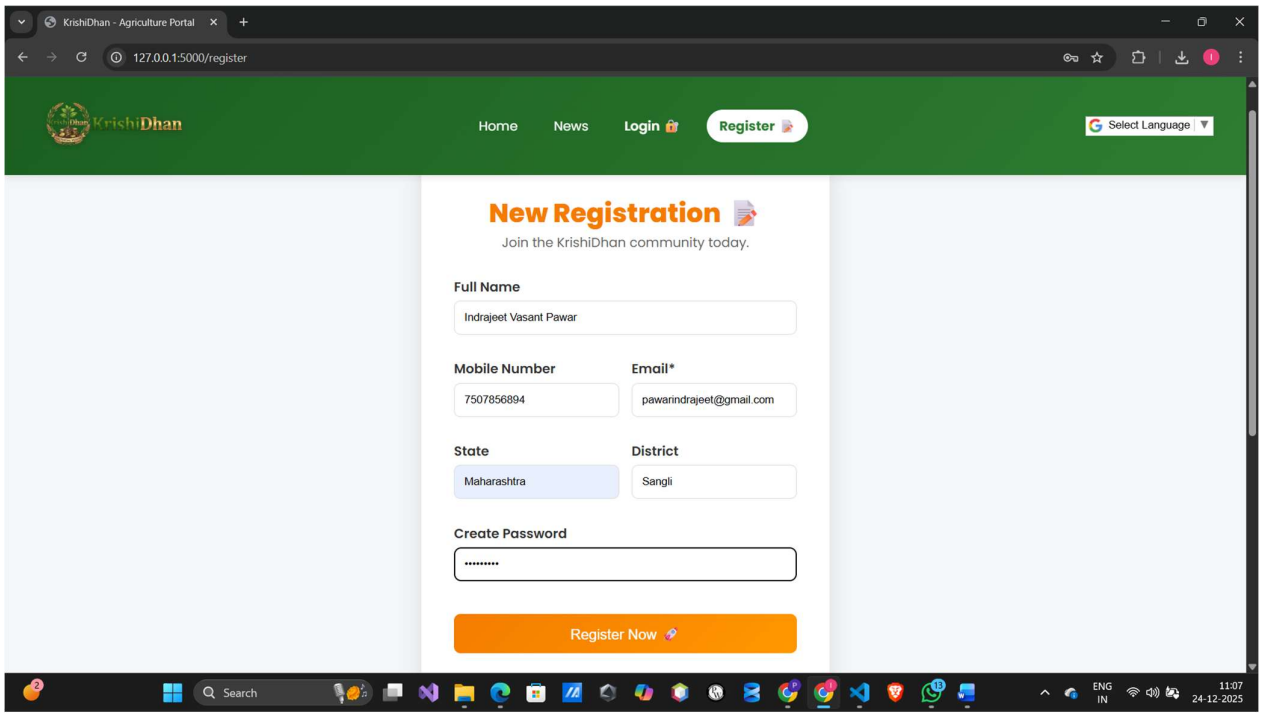


3.9 Sample Input and Output Screens.

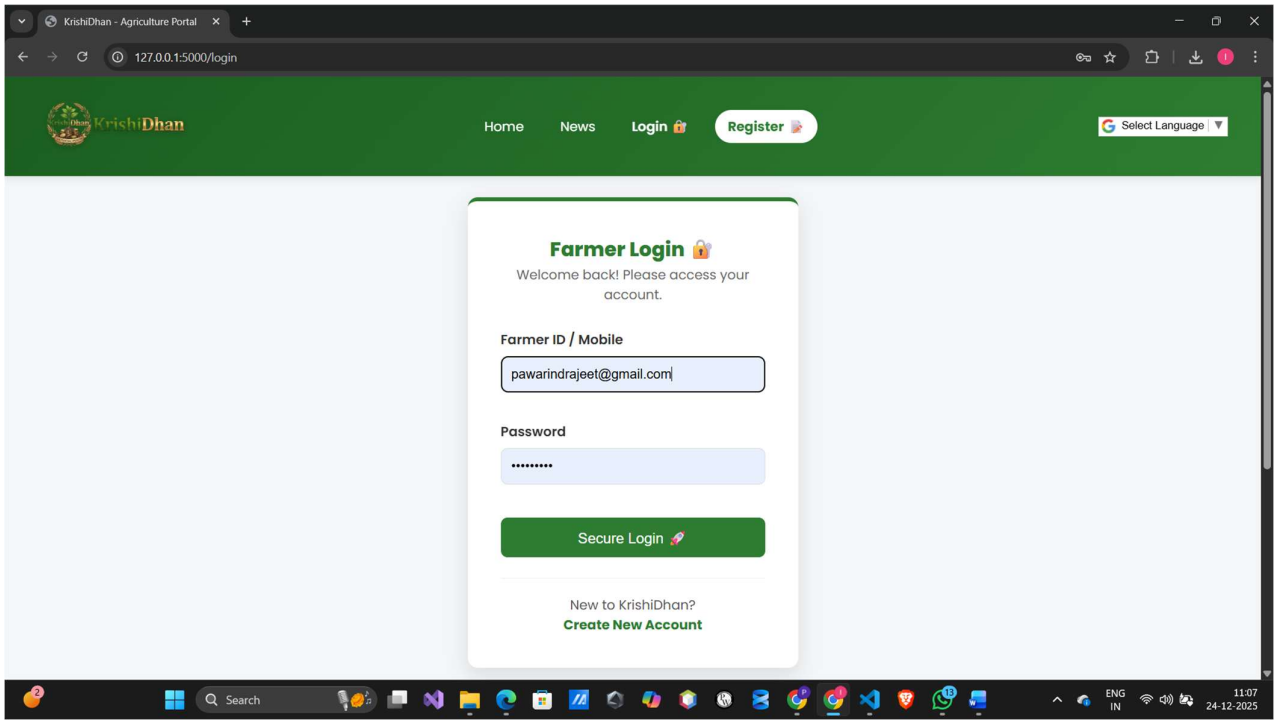
1.Home Page :



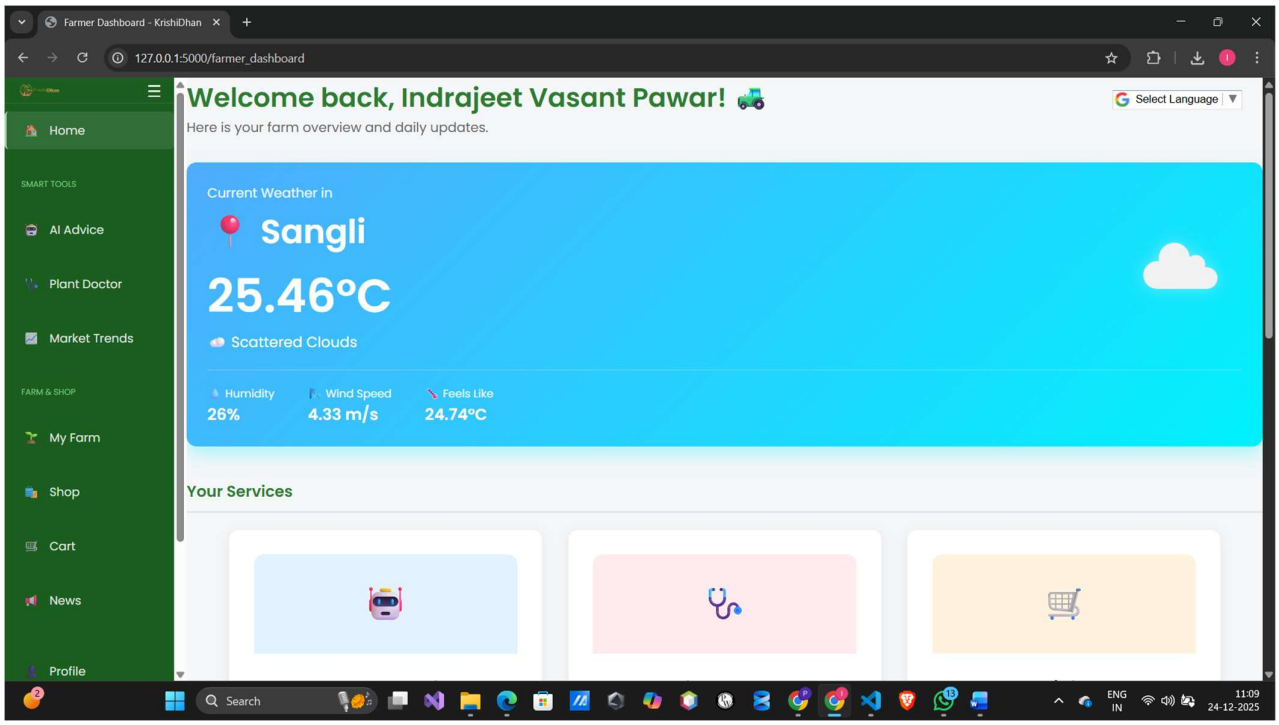
2.Registration Page:



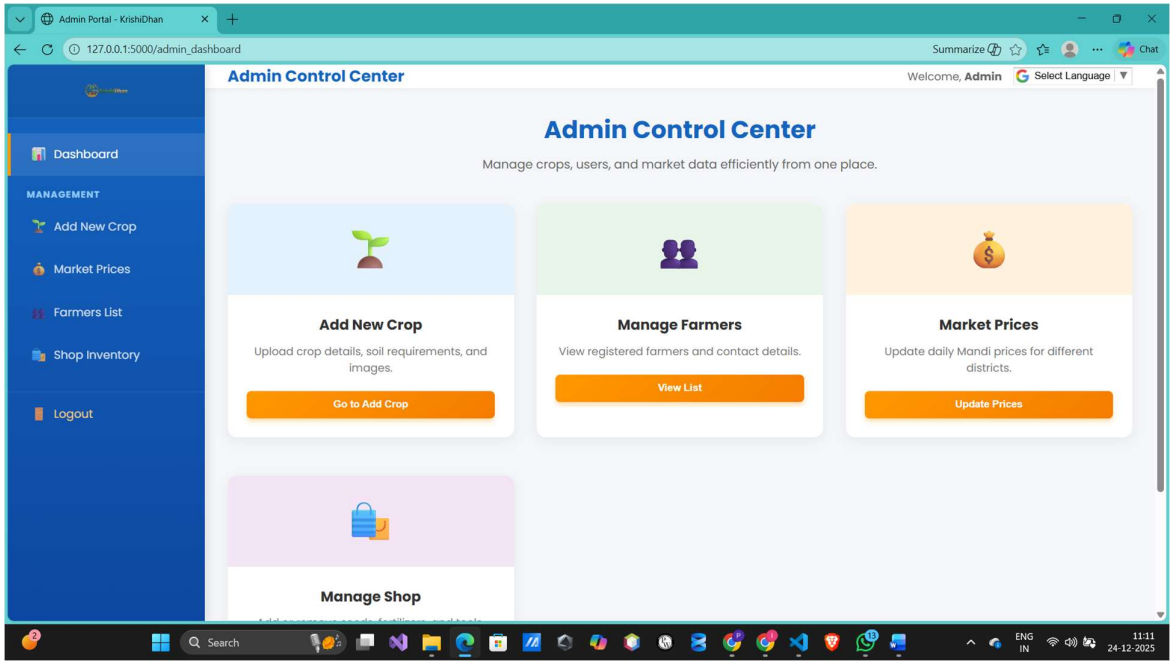
3.Login Page:



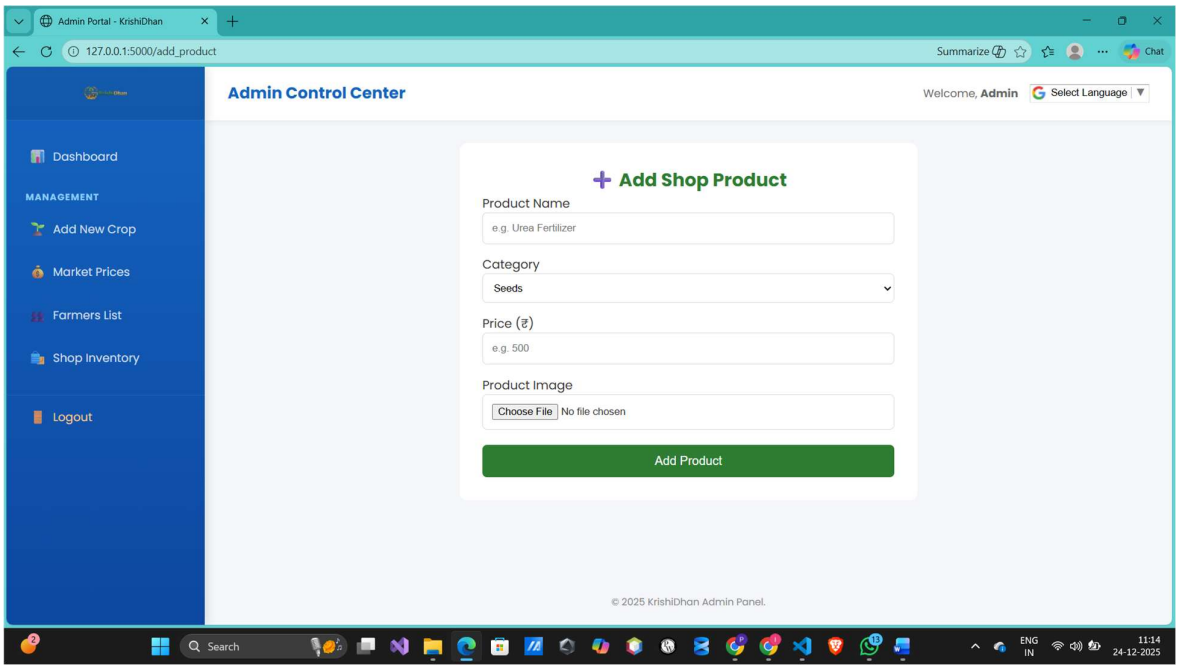
4.Farmer Dashboard Page:



4.Admin Dashboard Page:



5.Shop Inventory Page:



Chapter 4: Coding

Algorithm 1: User Registration

1. Start
2. User enters name, email, mobile, password, and role (Farmer/Admin).
3. Validate input fields (ensure email format is correct and mobile is 100 digits).
4. Check if email or mobile already exists in the PostgreSQL database.
5. If exists, show error message: "User already registered."
6. Else, encrypt the password using SHA-256 hashing.
7. Store user data in the users table in PostgreSQL.
8. Display registration success message.
9. End

Algorithm 2: User Login

1. Start
2. User enters username/email and password.
3. Fetch the hashed password from PostgreSQL for the given user.
4. Verify the entered password against the stored hash.
5. If invalid, display error: "Invalid Credentials."
6. If valid, check user role (Farmer or Admin).
7. Create a session and redirect to Farmer Dashboard or Admin Panel.
8. End

Algorithm 3: Plant Disease Diagnosis (Plant Doctor)

1. Start
2. Farmer logs in and navigates to the "Plant Doctor" section.
3. Farmer uploads or captures an image of a crop leaf.
4. Pre-process the image (Resize to 224x224 and normalize).
5. Load the CNN (Convolutional Neural Network) model.
6. Pass the image through the model to predict the disease class.
7. Fetch the treatment/remedy for the predicted disease from the database.
8. Display the diagnosis result and recommended fertilizers.
9. End

Algorithm 4: Purchase Agri-Products (E-Shop)

1. Start
2. Farmer logs in and browses the "Agri-Shop."
3. Select desired seeds, fertilizers, or tools.
4. Click "Add to Cart."
5. System checks product availability in the products table.
6. If stock is available, update the temporary cart session.
7. Proceed to checkout and confirm the order.
8. Deduct quantity from the stock_qty in PostgreSQL.
9. Display order confirmation and generate receipt.
10. End

Algorithm 5: Market Trend Visualization (Mandi Analytics)

1. Start
2. Farmer selects a specific crop (e.g., Wheat, Rice).
3. System queries the market_data table for historical price records.
4. Pass the data to the Pandas library for cleaning and grouping.
5. Generate a line/bar graph using Matplotlib.
6. Convert the graph into a base64 image string.
7. Render the graph on the "Market Trends" web page.
8. End

Algorithm 6: Voice Assistant Navigation

1. Start
2. Farmer clicks the "Voice" icon to activate the microphone.
3. Capture audio input and convert Speech-to-Text.
4. Match keywords (e.g., "Shop", "Doctor", "Weather").
5. If match found, trigger the Flask route for that module.
6. Else, use Text-to-Speech to ask the user to "Please repeat."
7. End

4.2 Code Snippets

Here are the critical code implementations for the KrishiDhan backend using Python and Flask.

4.2.1 AI Model Prediction (The Plant Doctor)

This snippet shows how the Flask route handles an image upload and returns a prediction.

Python

```
import tensorflow as tf
from flask import request, render_template
import numpy as np

# Load the pre-trained model
model = tf.keras.models.load_model('models/plant_disease_model.h5')

@app.route('/predict', methods=['POST'])
def predict_disease():
    if 'file' not in request.files:
        return "No file uploaded"

    file = request.files['file']
    # Convert file to image array and normalize
    img = tf.keras.preprocessing.image.load_img(file, target_size=(224, 224))
    img_array = tf.keras.preprocessing.image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0) / 255.0

    # Predict
    predictions = model.predict(img_array)
    result_index = np.argmax(predictions)

    # Map index to class name
    labels = ['Healthy', 'Leaf Blight', 'Rust', 'Yellow Mosaic']
    diagnosis = labels[result_index]

    return render_template('doctor_result.html', diagnosis=diagnosis)
```

4.2.2 Generating Market Trend Graphs (Data Science)

This snippet processes market data from the database and converts it into a visual format for the frontend.

Python

```
import matplotlib.pyplot as plt
import io
import base64
import pandas as pd

@app.route('/mandi-trends/<crop_name>')
def show_trends(crop_name):
    # Fetch data from PostgreSQL using Pandas
    conn = get_db_connection()
    query = f"SELECT date, price FROM market_data WHERE crop_name='{crop_name}'"
    df = pd.read_sql(query, conn)

    # Create the Plot
    plt.figure(figsize=(8, 4))
    plt.plot(df['date'], df['price'], marker='o', color='green')
    plt.title(f'Price Trend for {crop_name}')
    plt.xlabel('Date')
    plt.ylabel('Price per Quintal')

    # Convert plot to base64 string for HTML rendering
    img = io.BytesIO()
    plt.savefig(img, format='png')
    img.seek(0)
    graph_url = base64.b64encode(img.getvalue()).decode()

    return render_template('trends.html', graph_url=graph_url)
```

4.2.3 Secure User Login (Authentication)

Python

```
from werkzeug.security import check_password_hash

@app.route('/login', methods=['POST'])
def login():
    username = request.form['username']
    password = request.form['password']

    conn = get_db_connection()
    cur = conn.cursor()
    cur.execute("SELECT password_hash, role FROM users WHERE username = %s",
(username,))
    user = cur.fetchone()

    if user and check_password_hash(user[0], password):
        session['user'] = username
        session['role'] = user[1]
        return redirect('/dashboard')
    else:
        return "Invalid Credentials"
```

Chapter 5: Testing

5.1 Test Strategy

The KrishiDhan system is tested using a combination of **Manual and Functional Testing** techniques to ensure that the AI and E-commerce modules work seamlessly. The testing focuses on:

- **Functional Correctness:** Verifying that the AI model provides accurate diagnosis and the shop handles orders correctly.
- **Input Validation:** Ensuring images are in the correct format (JPG/PNG) and form fields are sanitized.
- **Role-Based Access:** Confirming that Farmers cannot access Admin inventory management tools.
- **Data Consistency in PostgreSQL:** Ensuring that transactions (like stock deduction) are atomic and consistent.
- **Integration Testing:** Verifying that the Voice Assistant correctly triggers the intended web routes.

5.2 Unit Test Plan

Module	Test Objective
User Registration	To verify that all input fields (Name, Email, Mobile, Password) are correctly validated and data is securely stored in PostgreSQL.
User Login	To verify that the system correctly authenticates users by comparing the entered password with the hashed password stored in the database.
Plant Doctor (AI)	To ensure the CNN model correctly processes uploaded images, performs resizing and normalization, and outputs an accurate disease diagnosis.
Agri-Shop (Cart)	To verify that items are correctly added to the session-based cart and that the total price calculation is accurate.
Mandi Trends (Analytics)	To ensure the system successfully retrieves historical data from the database and renders it into a visual graph using Matplotlib.
Voice Assistant	To verify that the speech-to-text integration correctly identifies keywords and triggers the corresponding

	module navigation.
Database Integration	To ensure that every transaction (such as order placement) correctly updates the stock quantity in the PostgreSQL products table.
Weather API	To verify that the system successfully fetches and displays real-time weather data based on the user's geographical location.

5.3 Acceptance Test Plan

The KrishiDhan Smart Farming Website is considered ready for deployment and officially accepted when it satisfies the following conditions:

- **User Management:** Farmers and Administrators can successfully register, log in, and maintain their respective sessions without data leaks.
- **AI Diagnostic Accuracy:** The "Plant Doctor" module successfully processes uploaded images and returns a valid diagnosis and remedy for at least 85% of known disease samples.
- **Data Visualization:** The "Mandi Trends" module correctly pulls historical data from the **PostgreSQL** database and renders a clear, readable line graph.
- **E-Commerce Workflow:** A farmer can browse the shop, add items to the cart, and complete a checkout process that successfully updates the inventory levels in the backend.
- **Voice Integration:** The system correctly interprets at least three core voice commands ("Open Shop", "Check Weather", "Go to Doctor") and performs the corresponding navigation.
- **Accessibility:** The website interface remains fully functional and readable when toggled between English and the selected regional language.
- **Performance:** All major pages, specifically the AI result page and the Analytics page, load within 5 seconds on a standard 4G mobile network.

5.4 Test Case / Test Script

Test Case ID	Description	Expected Result
TC01	Valid Registration	User record created in PostgreSQL; success message shown.
TC02	Invalid Login	System displays "Invalid Credentials" for wrong password.

TC03	Plant Image Upload	AI model identifies disease (e.g., Rust) and shows remedy.
TC04	Invalid File Upload	System rejects PDF/Doc files for Plant Doctor; shows error.
TC05	Add to Cart	Product added to session; cart count incremented.
TC06	View Mandi Trends	Line graph renders correctly using Matplotlib data.
TC07	Voice Command "Shop"	Browser redirects to the Shop page automatically.
TC08	Admin Product Add	New product appears in the farmer's shop catalog instantly.

5.5 Defect Report / Test Log

Defect ID	Module	Description	Status
D01	Login	System failed to display "Invalid Credentials" when a wrong password was entered.	Fixed
D02	Plant Doctor	AI diagnosis was allowed even when no image was uploaded (Empty submission).	Fixed
D03	E-Shop	The cart total price was not updating correctly after removing an item.	Fixed
D04	Registration	Mobile number field accepted alphabetic characters instead of only digits.	Fixed
D05	Mandi Trends	The price trend graph was not loading for crops with missing historical data.	Fixed
D06	Voice Assist	The voice assistant button was unresponsive on the Safari browser.	Fixed
D07	Database	Order records were being created even if the PostgreSQL stock count was zero.	Fixed

Chapter 6: Limitations of the Proposed System

While **KrishiDhan** provides an advanced platform for digital farming and market analytics, it has certain limitations in its current version:

1. **Dependency on Image Quality:** The accuracy of the "Plant Doctor" (AI Disease Detection) depends heavily on the quality, lighting, and clarity of the uploaded image; blurry or low-resolution photos may result in incorrect diagnoses.
2. **Lack of Real-time Payment Gateway:** The e-commerce module supports order placement and inventory tracking, but it does not currently include an integrated online payment gateway for instant transactions.
3. **No Map-based Mandi Search:** While price trends are provided, the system does not yet offer a GPS-integrated map to show the physical distance to the nearest market (Mandi).
4. **Internet Connectivity Requirements:** As a web-based platform, the system requires a stable internet connection to access AI models and live market data, which may be challenging in remote rural areas with poor connectivity.
5. **Manual Data Entry for Market Prices:** Currently, some market pricing data relies on manual updates or specific API fetches, which may not cover every local village market in real-time.
6. **Limited Offline Support:** The system is available only as a web application and does not offer an "Offline Mode" for farmers to access previously viewed data without an active connection.
7. **Scope of AI Model:** The disease detection model is currently trained on a specific set of common crops (e.g., Tomato, Potato, Rice); it may not identify diseases in rare or exotic plant varieties

Chapter 7: Proposed Enhancements

To make the system more robust, intelligent, and accessible to a wider farming community, the following features can be added in the future:

1. **Online Payment Gateway Integration:** Implementing secure payment gateways like Razorpay or Stripe to allow farmers to pay for seeds and fertilizers directly through the platform.
2. **IoT Sensor Integration:** Connecting the web application with on-field IoT sensors to provide real-time soil moisture, pH levels, and temperature data directly to the farmer's dashboard.
3. **Mobile Application Development:** Developing a native mobile app using **Flutter** or **React Native** to provide features like "Offline Mode" and push notifications for weather alerts.
4. **Google Maps & GPS Integration:** Integrating map-based services to show the exact location of nearby Mandis (markets) and track the delivery of ordered agricultural inputs.
5. **Multilingual SMS/WhatsApp Alerts:** Implementing an automated notification system to send disease diagnosis reports and daily market price alerts via SMS or WhatsApp in regional languages.
6. **Advanced AI Capabilities:** Expanding the "Plant Doctor" dataset to include more crop varieties and implementing "Pest Detection" alongside disease identification.
7. **Smart Crop Recommendation System:** Developing a Machine Learning model that recommends the best crop to plant based on historical weather patterns, soil health, and current market demand.
8. **Expert Consultation Module:** Adding a real-time chat or video call feature connecting farmers with agricultural scientists or government officials for expert advice.

These enhancements would transform KrishiDhan from a basic information portal into a comprehensive, high-tech agricultural management system.

Chapter 8: Conclusion

The **KrishiDhan** Smart Farming platform successfully provides a centralized and technology-driven ecosystem for modernizing traditional agricultural practices. By integrating Artificial Intelligence, data science, and e-commerce into a single web-based portal, the system empowers farmers with critical tools for disease diagnosis, market intelligence, and direct procurement.

The system effectively eliminates the traditional dependency on middlemen by providing transparent Mandi price trends and a direct marketplace for agricultural inputs. By implementing key features such as the **AI Plant Doctor**, role-based authentication, interactive data visualization, and a multilingual voice assistant, the system ensures both technical sophistication and ease of use for users with varying levels of digital literacy.

The use of modern web technologies like **Python/Flask**, a robust **PostgreSQL** database, and deep learning models makes the application scalable, reliable, and efficient. Overall, KrishiDhan fulfills its core objectives of improving crop health management and market transparency, offering a sustainable and reliable digital solution for the farming community.

Chapter 9: Bibliography

The development of the **KrishiDhan** Smart Farming platform was supported by the following technical documentation, books, and online resources:

9.1 Books & Publications

- **Goodfellow, I., Bengio, Y., & Courville, A. (2016):** *Deep Learning*. MIT Press. (Referenced for CNN architecture and image classification logic).
- **Grinberg, M. (2018):** *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media. (Referenced for backend routing and session management).
- **Winand, M. (2012):** *SQL Performance Explained*. (Referenced for PostgreSQL database optimization and indexing).

9.2 Technical Documentation

- **Flask Documentation:** https://flask.palletsprojects.com/ – Official guide for web framework implementation.
- **TensorFlow/Keras Documentation:** https://www.tensorflow.org/api_docs – Used for building and deploying the plant disease detection model.
- **PostgreSQL Official Manual:** https://www.postgresql.org/docs/ – Reference for relational schema design and ACID compliance.
- **Pandas & Matplotlib Documentation:** https://pandas.pydata.org/ – Used for market data analysis and trend visualization.

9.3 Online Resources & Research Papers

- **Kaggle Datasets:** PlantVillage Dataset used for training the CNN disease detection model.
- **W3Schools / MDN Web Docs:** Referenced for HTML5, CSS3, and JavaScript frontend responsive design.
- **OpenWeatherMap API:** Documentation used for implementing real-time weather forecasting.