Crop recommendation module:

import numpy as np

import pandas as pd

crop = pd.read\_csv("Crop\_recommendation.csv")

crop.head()

crop['label'].value\_counts()

crop\_dict = {

    'rice': 1,

    'maize': 2,

    'jute': 3,

    'cotton': 4,

    'coconut': 5,

    'papaya': 6,

    'orange': 7,

    'apple': 8,

    'muskmelon': 9,

    'watermelon': 10,

    'grapes': 11,

    'mango': 12,

    'banana': 13,

    'pomegranate': 14,

    'lentil': 15,

    'blackgram': 16,

    'mungbean': 17,

    'mothbeans': 18,

    'pigeonpeas': 19,

    'kidneybeans': 20,

    'chickpea': 21,

    'coffee': 22

}

crop['crop\_num']=  crop['label'].map(crop\_dict)

crop.drop(['label'],axis=1,inplace=True)

crop.head()

X = crop.drop(['crop\_num'],axis=1)

y = crop['crop\_num']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

from sklearn.preprocessing import MinMaxScaler

ms = MinMaxScaler()

X\_train = ms.fit\_transform(X\_train)

X\_test = ms.transform(X\_test)

from sklearn.linear\_model import LogisticRegression

from sklearn.naive\_bayes import GaussianNB

from sklearn.svm import SVC

from sklearn.neighbors import KNeighborsClassifier

from sklearn.tree import DecisionTreeClassifier

from sklearn.tree import ExtraTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import BaggingClassifier

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.ensemble import AdaBoostClassifier

from sklearn.metrics import accuracy\_score,confusion\_matrix

# create instances of all models

models = {

    'Logistic Regression': LogisticRegression(),

    'Naive Bayes': GaussianNB(),

    'Support Vector Machine': SVC(),

    'K-Nearest Neighbors': KNeighborsClassifier(),

    'Decision Tree': DecisionTreeClassifier(),

    'Random Forest': RandomForestClassifier(),

    'Bagging': BaggingClassifier(),

    'AdaBoost': AdaBoostClassifier(),

    'Gradient Boosting': GradientBoostingClassifier(),

    'Extra Trees': ExtraTreeClassifier(),

}

for name, model in models.items():

    model.fit(X\_train,y\_train)

    ypred = model.predict(X\_test)

    print(f"{name}  with accuracy : {accuracy\_score(y\_test,ypred)}")

    print("Confusion matrix : ",confusion\_matrix(y\_test,ypred))

    # finally selected randomforest model

rfc = RandomForestClassifier()

rfc.fit(X\_train,y\_train)

ypred = rfc.predict(X\_test)

accuracy\_score(y\_test,ypred)

def recommendation(N,P,k,temperature,humidity,ph,rainfal):

    features = np.array([[N,P,k,temperature,humidity,ph,rainfal]])

    transformed\_features = ms.fit\_transform(features)

    prediction = rfc.predict(transformed\_features)

    print(prediction)

    return prediction[0]

# new inputs

N = 40

P = 50

k = 50

temperature = 40.0

humidity = 20

ph = 100

rainfall = 100

predict = recommendation(N,P,k,temperature,humidity,ph,rainfall)

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

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

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

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

if predict in crop\_dict:

    crop = crop\_dict[predict]

    print("{} is a best crop to be cultivated in this environment  ".format(crop))

else:

    print("Sorry we  are not able to recommend a proper or preferable  crop for this environment")

import pickle

pickle.dump(rfc,open('model.pkl','wb'))

pickle.dump(ms,open('minmaxscaler.pkl','wb'))

**Fertilizer recommendation module:**

**from** sklearn.preprocessing **import** LabelEncoder

encode\_soil **=** LabelEncoder()

data**.**Soil\_Type **=** encode\_soil**.**fit\_transform(data**.**Soil\_Type)

Soil\_Type **=** pd**.**DataFrame(zip(encode\_soil**.**classes\_,encode\_soil**.**transform(encode\_soil**.**classes\_)),columns**=**['Original','Encoded'])

Soil\_Type **=** Soil\_Type**.**set\_index('Original')

Soil\_Type

encode\_crop **=** LabelEncoder()

data**.**Crop\_Type **=** encode\_crop**.**fit\_transform(data**.**Crop\_Type)

Crop\_Type **=** pd**.**DataFrame(zip(encode\_crop**.**classes\_,encode\_crop**.**transform(encode\_crop**.**classes\_)),columns**=**['Original','Encoded'])

Crop\_Type **=** Crop\_Type**.**set\_index('Original')

Crop\_Type

encode\_ferti **=** LabelEncoder()

data**.**Fertilizer **=** encode\_ferti**.**fit\_transform(data**.**Fertilizer)

Fertilizer **=** pd**.**DataFrame(zip(encode\_ferti**.**classes\_,encode\_ferti**.**transform(encode\_ferti**.**classes\_)),columns**=**['Original','Encoded'])

Fertilizer **=** Fertilizer**.**set\_index('Original')

Fertilizer

**from** sklearn.model\_selection **import** train\_test\_split

x\_train, x\_test, y\_train, y\_test **=** train\_test\_split(data**.**drop('Fertilizer',axis**=**1),data**.**Fertilizer,test\_size**=**0.2,random\_state**=**1)

print('Shape of Splitting :')

print('x\_train = {}, y\_train = {}, x\_test = {}, y\_test = {}'**.**format(x\_train**.**shape,y\_train**.**shape,x\_test**.**shape,y\_test**.**shape))

**from** sklearn.ensemble **import** RandomForestClassifier

RF **=** RandomForestClassifier(n\_estimators**=**20, random\_state**=**0)

RF**.**fit(x\_train,y\_train)

predicted\_values **=** RF**.**predict(x\_test)

x **=** metrics**.**accuracy\_score(y\_test, predicted\_values)

acc**.**append(x)

predicted\_values **=** RF**.**predict(x\_train)

y **=** metrics**.**accuracy\_score(y\_train, predicted\_values)

acc1**.**append(y)

model**.**append('RF')

print("RF's Accuracy is: ", x,y)

print(classification\_report(y\_test,predicted\_values))

Weed Detection Module:

import os

# Operating system interfaces

import tensorflow as tf

# TensorFlow deep learning framework

import matplotlib.pyplot as plt

# Plotting library

import matplotlib.image as mpimg

# Image loading and manipulation library

from tensorflow import keras

from keras.models import Sequential, Model

# Sequential and Functional API for building models

from keras.optimizers import Adam

# Adam optimizer for model training

from keras.callbacks import EarlyStopping

# Early stopping callback for model training

from keras.regularizers import l1, l2

# L1 and L2 regularization for model regularization

from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Data augmentation and preprocessing for images

from keras.layers import Dense, Flatten, Dropout, GlobalAveragePooling2D, AveragePooling2D, MaxPooling2D, BatchNormalization

# Various types of layers for building neural networks

from keras.applications import DenseNet121, EfficientNetB4, Xception, VGG16, VGG19

# Pre-trained models for transfer learning

directory = '/content/drive/MyDrive/weed\_detection\_final/train'

datagen = ImageDataGenerator(

rescale=1./255,

rotation\_range=20,

# Random rotation up to 20 degrees

width\_shift\_range=0.2,

# Random horizontal shift

height\_shift\_range=0.2,

# Random vertical shift

shear\_range=0.2,

# Shear transformation

zoom\_range=0.2,

# Random zoom

horizontal\_flip=True,

# Random horizontal flip

fill\_mode='nearest'

# Fill mode for handling empty pixels after transformations

)

# Create a flow\_from\_directory generator

train\_data = datagen.flow\_from\_directory(

directory,

classes=['.'], # Specify the current directory

class\_mode='categorical',

target\_size=(256, 256),

batch\_size=32,

)

directory = '/content/drive/MyDrive/weed\_detection\_final/test'

# Load the validation data with moderate data augmentation

datagen = ImageDataGenerator(

rescale=1./255,

rotation\_range=20,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True,

fill\_mode='nearest'

)

# Create a flow\_from\_directory generator

test\_data = datagen.flow\_from\_directory(

directory,

classes=['.'], # Specify the current directory

class\_mode='categorical',

target\_size=(256, 256),

batch\_size=32,

)

from google.colab import drive

drive.mount('/content/drive')

conv\_base = DenseNet121(

weights='imagenet',

include\_top = False,

input\_shape=(256,256,3),

pooling='avg'

)

conv\_base.trainable = False

conv\_base.summary()

model = Sequential()

model.add(conv\_base)

model.add(BatchNormalization())

model.add(Dense(256, activation='relu'))

model.add(Dropout(0.35))

model.add(BatchNormalization())

model.add(Dense(120, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

model.summary()

model.compile(optimizer=Adam(learning\_rate=0.0001), loss='binary\_crossentropy', metrics=['accuracy'])

early\_stopping = keras.callbacks.EarlyStopping(

monitor="val\_loss",

min\_delta=0,

patience=5,

verbose=0,

mode="auto",

baseline=None, # Set to the value of val\_loss at the desired epoch

restore\_best\_weights=False,

)

history = model.fit(train\_data, epochs=4, validation\_data=test\_data, callbacks=[early\_stopping])

model.save("densenet.h5")

Chatbot Module:

import requests

import PyPDF2

from itertools import chain

from langchain\_community.vectorstores import Chroma

from langchain.text\_splitter import RecursiveCharacterTextSplitter

from langchain\_community.embeddings.sentence\_transformer import SentenceTransformerEmbeddings

# Function to fetch content from a website

def fetch\_website\_content(url):

    response = requests.get(url)

    return response.text

# Function to extract text from a PDF file

def extract\_pdf\_text(pdf\_file):

    with open(pdf\_file, "rb") as file:

        pdf\_reader = PyPDF2.PdfReader(file)

        text = ""

        for page\_num in range(len(pdf\_reader.pages)):

            text += pdf\_reader.pages[page\_num].extract\_text()

    return text

# Split the combined content into smaller chunks

def split\_text(text, chunk\_size=600, chunk\_overlap=150):

    text\_splitter = RecursiveCharacterTextSplitter(chunk\_size=chunk\_size, chunk\_overlap=chunk\_overlap)

    chunks = text\_splitter.split\_text(text)

    return chunks

# Initialize embeddings and vector store

def initialize\_vector\_store(contents):

    embedding\_function = SentenceTransformerEmbeddings(model\_name="all-MiniLM-L6-v2")

    web\_chunks = list(chain.from\_iterable(split\_text(content) for content in contents))

    db = Chroma.from\_texts(web\_chunks, embedding\_function)

    return db

from langchain\_together import Together

from langchain.chains import RetrievalQA

from langchain.prompts import PromptTemplate

# Initialize the language model with validated parameters

llm = Together(

    model="meta-llama/Llama-3.3-70B-Instruct-Turbo-Free",

    together\_api\_key="5c7cd22cdd7a944ae6de96df70a74203d87e3c776d4f7525029cef58aa25b16a",

    temperature=0.1,

    max\_tokens=150,

    # Remove the 'stop' parameter as it's not supported in this version

    top\_k=1

)

def setup\_retrieval\_qa(db):

    retriever = db.as\_retriever(search\_kwargs={"k": 3})

    # Improved prompt with built-in stopping mechanism

    prompt\_template = """[INST] <<SYS>>You are SasyaVaani. Answer in 1-2 sentences.

    End your response with '[END]' when finished.

    Context: {context}

    Question: {question}

    <</SYS>>[/INST]"""

    PROMPT = PromptTemplate(

        template=prompt\_template,

        input\_variables=["context", "question"]

    )

    return RetrievalQA.from\_chain\_type(

        llm=llm,

        chain\_type='stuff',

        retriever=retriever,

        input\_key='query',

        return\_source\_documents=True,

        chain\_type\_kwargs={"prompt": PROMPT}

    )

**App.py**

from flask import Flask, render\_template, request, session, redirect, url\_for, jsonify

import mysql.connector

from flask\_bcrypt import Bcrypt

import pickle

import numpy as np

import pandas as pd

from tensorflow.keras.models import load\_model

import cv2

import os

from chat1 import fetch\_website\_content, extract\_pdf\_text, initialize\_vector\_store

from chat2 import setup\_retrieval\_qa

# Initialize Flask App

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

app.secret\_key = "your\_secret\_key"

bcrypt = Bcrypt(app)

# Database Connection

db = mysql.connector.connect(

    host="localhost",

    user="root",

    password="root",

    database="smart\_farming",

    charset="utf8"

)

cursor = db.cursor()

# Load ML Models

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

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

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

# Load Weed Detection Model

weed\_model = load\_model('densenet.h5')

# Load Crop NPK Data

crop\_npk\_data = pd.read\_csv("Crop\_NPK.csv")

# Initialize Chatbot Components

def initialize\_components():

    urls = ["https://mospi.gov.in/4-agricultural-statistics"]

    pdf\_files = ["Data/Farming Schemes.pdf", "Data/farmerbook.pdf"]

    website\_contents = [fetch\_website\_content(url) for url in urls]

    pdf\_texts = [extract\_pdf\_text(pdf\_file) for pdf\_file in pdf\_files]

    all\_contents = website\_contents + pdf\_texts

    vector\_db = initialize\_vector\_store(all\_contents)

    return setup\_retrieval\_qa(vector\_db)

qa\_chain = initialize\_components()

# Home Page

@app.route("/")

def index():

    return render\_template("index.html")

# User Registration

@app.route("/register", methods=["GET", "POST"])

def register():

    if request.method == "POST":

        username = request.form.get("username")

        email = request.form.get("email")

        phone = request.form.get("phone")

        password = request.form.get("password")

        state = request.form.get("state")

        district = request.form.get("district")

        cursor.execute("SELECT \* FROM users WHERE email=%s OR phone=%s", (email, phone))

        user = cursor.fetchone()

        if user:

            return "User already exists!"

        hashed\_password = bcrypt.generate\_password\_hash(password).decode('utf-8')

        cursor.execute(

            "INSERT INTO users (username, email, phone, password, state, district) VALUES (%s, %s, %s, %s, %s, %s)",

            (username, email, phone, hashed\_password, state, district)

        )

        db.commit()

        return redirect(url\_for("login"))

    return render\_template("register.html")

# User Login

@app.route("/login", methods=["GET", "POST"])

def login():

    if request.method == "POST":

        email\_or\_phone = request.form.get("email\_or\_phone")

        password = request.form.get("password")

        cursor.execute("SELECT \* FROM users WHERE email=%s OR phone=%s", (email\_or\_phone, email\_or\_phone))

        user = cursor.fetchone()

        if user and bcrypt.check\_password\_hash(user[4], password):

            session["user\_id"] = user[0]

            session["username"] = user[1]  # Store username in session

            return redirect(url\_for("dashboard"))

        return "Invalid credentials. Please try again."

    return render\_template("login.html")

# Dashboard

@app.route("/dashboard")

def dashboard():

    if "user\_id" not in session:

        return redirect(url\_for("login"))

    return render\_template("dashboard.html", username=session.get("username"))

# Crop Recommendation Route

@app.route("/crop\_recommendation", methods=["GET", "POST"])

def crop\_recommendation():

    if "user\_id" not in session:

        return redirect(url\_for("login"))

    result = None

    if request.method == "POST":

        N = float(request.form['Nitrogen'])

        P = float(request.form['Phosporus'])

        K = float(request.form['Potassium'])

        temp = float(request.form['Temperature'])

        humidity = float(request.form['Humidity'])

        ph = float(request.form['Ph'])

        rainfall = float(request.form['Rainfall'])

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

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

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

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

        prediction = model.predict(final\_features)

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

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

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

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

        if prediction[0] in crop\_dict:

            crop = crop\_dict[prediction[0]]

            result = f"{crop} is the best crop to be cultivated right there"

        else:

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

    return render\_template("crop\_recommendation.html", result=result)

# Fertilizer Recommendation Route

@app.route("/fertilizer\_recommendation", methods=["GET", "POST"])

def fertilizer\_recommendation():

    if "user\_id" not in session:

        return redirect(url\_for("login"))

    result = None

    if request.method == "POST":

        crop = request.form['crop']

        current\_n = float(request.form['Nitrogen'])

        current\_p = float(request.form['Phosporus'])

        current\_k = float(request.form['Potassium'])

        # Get fertilizer recommendation

        crop\_data = crop\_npk\_data[crop\_npk\_data['Crop'] == crop]

        if crop\_data.empty:

            result = "Crop not found in the dataset."

        else:

            recommended\_n = crop\_data['N'].values[0]

            recommended\_p = crop\_data['P'].values[0]

            recommended\_k = crop\_data['K'].values[0]

            required\_n = max(0, recommended\_n - current\_n)

            required\_p = max(0, recommended\_p - current\_p)

            required\_k = max(0, recommended\_k - current\_k)

            result = f"Fertilizer Recommendation for {crop}:<br>"

            result += f"Nitrogen (N): {required\_n} kg/ha<br>"

            result += f"Phosphorus (P): {required\_p} kg/ha<br>"

            result += f"Potassium (K): {required\_k} kg/ha"

    # Pass the unique list of crops to the template

    crops = crop\_npk\_data['Crop'].unique()

    return render\_template("fertilizer\_recommendation.html", result=result, crops=crops)

# Weed Detection Route

@app.route("/weed\_detection", methods=["GET", "POST"])

def weed\_detection():

    if "user\_id" not in session:

        return redirect(url\_for("login"))

    result = None

    if request.method == "POST":

        if 'file' not in request.files:

            return "No file uploaded", 400

        file = request.files['file']

        if file.filename == '':

            return "No file selected", 400

        # Save the uploaded file temporarily

        upload\_folder = os.path.join('static', 'uploads')

        os.makedirs(upload\_folder, exist\_ok=True)

        file\_path = os.path.join(upload\_folder, file.filename)

        file.save(file\_path)

        # Preprocess the image

        image = cv2.imread(file\_path)

        image = cv2.resize(image, (256, 256))  # Resize to match model input size (256x256)

        image = image / 255.0  # Normalize pixel values

        image = np.expand\_dims(image, axis=0)  # Add batch dimension

        # Make a prediction

        prediction = weed\_model.predict(image)

        class\_label = 'Weed' if prediction[0][0] > 0.5 else 'Crop'

        # Return the result

        result = f"The image contains: {class\_label}"

    return render\_template("weed\_detection.html", result=result)

# Chatbot Route

@app.route("/chatbot")

def chatbot():

    if "user\_id" not in session:

        return redirect(url\_for("login"))

    return render\_template("chatbot.html")

# Chatbot API Endpoint

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

def ask():

    if "user\_id" not in session:

        return jsonify({"answer": "Please login to use the chatbot."})

    query = request.form['messageText'].strip().lower()

    # Hardcoded responses

    if query in ["who developed you", "who created you", "who made you"]:

        return jsonify({"answer": "I was developed by Krishna Kanisetti."})

    try:

        response = qa\_chain({"query": query})

        # Clean the response by removing everything after [END]

        answer = response['result'].split('[END]')[0].strip()

        return jsonify({"answer": answer})

    except Exception as e:

        return jsonify({"answer": "Sorry, I encountered an error. Please try again."})

# Logout Route

@app.route("/logout")

def logout():

    session.clear()

    return redirect(url\_for("index"))

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

    app.run(debug=True)

**Web Pages:**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Smart Farming Platform</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

        body {

            font-family: 'Arial', sans-serif;

            background-color: #f8f9fa; /\* Light gray background \*/

            margin: 0;

            padding: 0;

        }

        header {

            background-color: #00A86B; /\* Jade green for the header \*/

            color: #fff;

            padding: 100px 20px;

            text-align: center;

            animation: fadeIn 1.5s ease-in-out;

        }

        header h1 {

            font-size: 3.5rem;

            margin: 0;

            animation: slideInFromTop 1s ease-in-out;

        }

        nav {

            background-color: #007B5E; /\* Darker green for the nav \*/

            padding: 15px 20px;

            text-align: center;

        }

        nav a {

            color: #fff;

            text-decoration: none;

            margin: 0 20px;

            font-size: 1.2rem;

            font-weight: bold;

            transition: color 0.3s ease;

        }

        nav a:hover {

            color: #E8F5E9; /\* Very light green on hover \*/

        }

        section {

            padding: 60px 20px;

            text-align: center;

        }

        section h2 {

            color: #00A86B; /\* Jade green for headings \*/

            font-size: 2.8rem;

            margin-bottom: 20px;

            animation: fadeIn 2s ease-in-out;

        }

        section p {

            color: #333;

            font-size: 1.2rem;

            max-width: 800px;

            margin: 0 auto;

            animation: fadeIn 2.5s ease-in-out;

        }

        .features {

            display: flex;

            justify-content: space-around;

            flex-wrap: wrap;

            margin-top: 40px;

        }

        .feature-card {

            background-color: #E8F5E9; /\* Very light green for cards \*/

            border: 1px solid #00A86B; /\* Jade green border \*/

            border-radius: 10px;

            padding: 30px;

            width: 30%;

            margin: 15px;

            transition: transform 0.3s ease, box-shadow 0.3s ease;

            animation: fadeIn 3s ease-in-out;

        }

        .feature-card:hover {

            transform: scale(1.05);

            box-shadow: 0 0 20px rgba(0, 168, 107, 0.3); /\* Glow effect on hover \*/

        }

        .feature-card h3 {

            color: #007B5E; /\* Darker green for feature titles \*/

            font-size: 2rem;

            margin-bottom: 15px;

        }

        .feature-card p {

            color: #333;

            font-size: 1.1rem;

        }

        .testimonials {

            background-color: #fff;

            padding: 60px 20px;

            margin-top: 40px;

        }

        .testimonial-card {

            background-color: #E8F5E9;

            border: 1px solid #00A86B;

            border-radius: 10px;

            padding: 25px;

            max-width: 600px;

            margin: 20px auto;

            animation: fadeIn 3.5s ease-in-out;

        }

        .testimonial-card p {

            font-style: italic;

            color: #333;

            font-size: 1.1rem;

        }

        .testimonial-card h4 {

            color: #007B5E;

            margin-top: 15px;

            font-size: 1.2rem;

        }

        .cta {

            background-color: #00A86B;

            color: #fff;

            padding: 80px 20px;

            text-align: center;

            animation: fadeIn 4s ease-in-out;

        }

        .cta h2 {

            font-size: 2.8rem;

            margin-bottom: 30px;

        }

        .cta button {

            background-color: #007B5E;

            color: #fff;

            border: none;

            padding: 15px 40px;

            font-size: 1.3rem;

            cursor: pointer;

            border-radius: 5px;

            transition: background-color 0.3s ease, transform 0.3s ease;

        }

        .cta button:hover {

            background-color: #005f4b; /\* Darker green on hover \*/

            transform: translateY(-5px);

        }

        footer {

            background-color: #007B5E;

            color: #fff;

            text-align: center;

            padding: 20px;

            margin-top: 40px;

        }

        /\* Keyframes for animations \*/

        @keyframes fadeIn {

            from { opacity: 0; }

            to { opacity: 1; }

        }

        @keyframes slideInFromTop {

            from { transform: translateY(-50px); opacity: 0; }

            to { transform: translateY(0); opacity: 1; }

        }

    </style>

</head>

<body>

    <header>

        <h1>Welcome to the Smart Farming Platform</h1>

    </header>

    <nav class="navbar navbar-expand-lg navbar-dark" style="background-color: #007B5E;">

        <div class="container-fluid">

            <a class="navbar-brand" href="/">Smart Farming Platform</a>

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

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

            </button>

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

                <ul class="navbar-nav ms-auto">

                    <li class="nav-item">

                        <a class="nav-link" href="/">Home</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/about">About Us</a>

                    </li>

                    <li class="nav-item dropdown">

                        <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

                            Features

                        </a>

                        <ul class="dropdown-menu" aria-labelledby="navbarDropdown">

                            <li><a class="dropdown-item" href="/crop\_recommendation">Crop Recommendation</a></li>

                            <li><a class="dropdown-item" href="/fertilizer\_recommendation">Fertilizer Recommendation</a></li>

                            <li><a class="dropdown-item" href="/irrigation\_planning">Irrigation Planning</a></li>

                        </ul>

                    </li>

                    <li class="nav-item">

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

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/register">Register</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/login">Login</a>

                    </li>

                </ul>

            </div>

        </div>

    </nav>

    <section>

        <h2>Enhancing Agriculture with Technology</h2>

        <p>Get AI-powered recommendations for crop selection, irrigation, and yield optimization. Join us to revolutionize farming with cutting-edge technology.</p>

        <div class="features">

            <div class="feature-card">

                <h3>Crop Recommendations</h3>

                <p>Get the best crop suggestions based on soil and weather conditions to maximize your yield.</p>

            </div>

            <div class="feature-card">

                <h3>Fertilizer Guidance</h3>

                <p>Optimize soil health with personalized fertilizer recommendations tailored to your crops.</p>

            </div>

            <div class="feature-card">

                <h3>Irrigation Planning</h3>

                <p>Plan your irrigation efficiently with AI-driven insights to save water and improve crop health.</p>

            </div>

        </div>

    </section>

    <section class="testimonials">

        <h2>What Our Users Say</h2>

        <div class="testimonial-card">

            <p>"This platform has transformed the way I farm. The recommendations are accurate and easy to follow!"</p>

            <h4>- John Doe, Farmer</h4>

        </div>

        <div class="testimonial-card">

            <p>"The fertilizer guidance helped me reduce costs and improve my crop yield significantly."</p>

            <h4>- Jane Smith, Agripreneur</h4>

        </div>

    </section>

    <section class="cta">

        <h2>Ready to Transform Your Farming?</h2>

        <button onclick="window.location.href='{{ url\_for('register') }}'">Get Started Now</button>

    </section>

    <footer>

        <p>&copy; 2023 Smart Farming Platform. All rights reserved.</p>

    </footer>

</body>

</html>

**Register.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Register</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

        body {

            font-family: 'Arial', sans-serif;

            margin: 0;

            padding: 0;

            display: flex;

            flex-direction: column;

            height: 100vh;

            overflow: hidden;

        }

        .navbar {

            background-color: #007B5E; /\* Darker green for the navbar \*/

            padding: 10px 20px;

            position: fixed; /\* Fix the navbar at the top \*/

            width: 100%;

            z-index: 1000; /\* Ensure navbar is above other content \*/

        }

        .navbar-brand, .nav-link {

            color: #fff !important; /\* White text for navbar \*/

        }

        .nav-link:hover {

            color: #E8F5E9 !important; /\* Very light green on hover \*/

        }

        .split-screen {

            display: flex;

            flex: 1;

            overflow: hidden;

            padding-top: 70px; /\* Add padding to avoid overlap with navbar \*/

        }

        .left-side {

            flex: 1;

            background: url('https://images.unsplash.com/photo-1464226184884-fa280b87c399?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=1170&q=80') no-repeat center center/cover;

            display: flex;

            justify-content: center;

            align-items: center;

            color: #fff;

            text-align: center;

            animation: fadeIn 1.5s ease-in-out;

        }

        .left-side h1 {

            font-size: 2.5rem;

            background: rgba(0, 0, 0, 0.5);

            padding: 20px;

            border-radius: 10px;

        }

        .right-side {

            flex: 1;

            display: flex;

            justify-content: center;

            align-items: center;

            background-color: #f8f9fa; /\* Light gray background \*/

            padding: 20px;

            animation: slideInFromRight 1s ease-in-out;

            overflow: auto; /\* Allow scrolling if content overflows \*/

        }

        .register-form {

            width: 100%;

            max-width: 400px;

            padding: 20px;

            background-color: #fff;

            border-radius: 10px;

            box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);

        }

        .register-form h2 {

            color: #00A86B; /\* Jade green for the heading \*/

            font-size: 2rem;

            margin-bottom: 15px;

            text-align: center;

        }

        label {

            color: #007B5E; /\* Darker green for labels \*/

            font-weight: bold;

            margin-top: 5px;

            display: block;

        }

        input, select {

            width: 100%;

            padding: 8px;

            margin: 5px 0 10px 0; /\* Reduced margin \*/

            border: 1px solid #00A86B; /\* Jade green border \*/

            border-radius: 5px;

            font-size: 0.9rem;

            transition: border-color 0.3s ease, box-shadow 0.3s ease;

        }

        input:focus, select:focus {

            border-color: #007B5E; /\* Darker green on focus \*/

            box-shadow: 0 0 8px rgba(0, 168, 107, 0.3); /\* Glow effect on focus \*/

        }

        button {

            background-color: #00A86B; /\* Jade green for the button \*/

            color: #fff;

            border: none;

            padding: 10px 20px;

            font-size: 1rem;

            cursor: pointer;

            border-radius: 5px;

            width: 100%;

            margin-top: 10px;

            transition: background-color 0.3s ease, transform 0.3s ease;

        }

        button:hover {

            background-color: #007B5E; /\* Darker green on hover \*/

            transform: translateY(-2px);

        }

        button:active {

            transform: translateY(0);

        }

        /\* Keyframes for animations \*/

        @keyframes fadeIn {

            from { opacity: 0; }

            to { opacity: 1; }

        }

        @keyframes slideInFromRight {

            from { transform: translateX(100px); opacity: 0; }

            to { transform: translateX(0); opacity: 1; }

        }

        /\* Responsive Design \*/

        @media (max-width: 768px) {

            .split-screen {

                flex-direction: column;

                padding-top: 60px; /\* Adjust padding for mobile \*/

            }

            .left-side, .right-side {

                flex: 1;

            }

            .left-side h1 {

                font-size: 2rem;

            }

            .register-form {

                max-width: 90%;

            }

        }

    </style>

    <script>

        // Fetch states & districts dynamically

        document.addEventListener("DOMContentLoaded", function () {

            fetch("{{ url\_for('static', filename='states.json') }}")

            .then(response => response.json())

            .then(data => {

                let stateDropdown = document.getElementById("state");

                let districtDropdown = document.getElementById("district");

                stateDropdown.innerHTML = '<option value="">Select State</option>';

                Object.keys(data).forEach(state => {

                    let option = document.createElement("option");

                    option.value = state;

                    option.textContent = state;

                    stateDropdown.appendChild(option);

                });

                stateDropdown.addEventListener("change", function () {

                    let selectedState = this.value;

                    districtDropdown.innerHTML = '<option value="">Select District</option>';

                    if (selectedState in data) {

                        data[selectedState].forEach(district => {

                            let option = document.createElement("option");

                            option.value = district;

                            option.textContent = district;

                            districtDropdown.appendChild(option);

                        });

                    }

                });

            });

        });

    </script>

</head>

<body>

    <!-- Navbar -->

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

        <div class="container-fluid">

            <a class="navbar-brand" href="/">Smart Farming Platform</a>

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

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

            </button>

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

                <ul class="navbar-nav ms-auto">

                    <li class="nav-item">

                        <a class="nav-link" href="/">Home</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/about">About Us</a>

                    </li>

                    <li class="nav-item">

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

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/register">Register</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/login">Login</a>

                    </li>

                </ul>

            </div>

        </div>

    </nav>

    <!-- Split Screen -->

    <div class="split-screen">

        <!-- Left Side: Visual Background -->

        <div class="left-side">

            <h1>Join the Future of Farming</h1>

        </div>

        <!-- Right Side: Registration Form -->

        <div class="right-side">

            <div class="register-form">

                <h2>Register</h2>

                <form action="{{ url\_for('register') }}" method="POST">

                    <label for="username">Username:</label>

                    <input type="text" id="username" name="username" required>

                    <label for="email">Email:</label>

                    <input type="email" id="email" name="email" required>

                    <label for="phone">Phone Number:</label>

                    <input type="text" id="phone" name="phone" required>

                    <label for="password">Password:</label>

                    <input type="password" id="password" name="password" required>

                    <label for="state">State:</label>

                    <select id="state" name="state" required>

                        <option value="">Select State</option>

                    </select>

                    <label for="district">District:</label>

                    <select id="district" name="district" required>

                        <option value="">Select District</option>

                    </select>

                    <button type="submit" id="register-btn">Register</button>

                </form>

            </div>

        </div>

    </div>

</body>

</html>

**Login.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

        body {

            font-family: 'Arial', sans-serif;

            margin: 0;

            padding: 0;

            display: flex;

            flex-direction: column;

            height: 100vh;

            overflow: hidden;

        }

        .navbar {

            background-color: #007B5E; /\* Darker green for the navbar \*/

            padding: 10px 20px;

            position: fixed; /\* Fix the navbar at the top \*/

            width: 100%;

            z-index: 1000; /\* Ensure navbar is above other content \*/

        }

        .navbar-brand, .nav-link {

            color: #fff !important; /\* White text for navbar \*/

        }

        .nav-link:hover {

            color: #E8F5E9 !important; /\* Very light green on hover \*/

        }

        .split-screen {

            display: flex;

            flex: 1;

            overflow: hidden;

            padding-top: 70px; /\* Add padding to avoid overlap with navbar \*/

        }

        .left-side {

            flex: 1;

            background: url('https://images.unsplash.com/photo-1464226184884-fa280b87c399?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=1170&q=80') no-repeat center center/cover;

            display: flex;

            justify-content: center;

            align-items: center;

            color: #fff;

            text-align: center;

            animation: fadeIn 1.5s ease-in-out;

        }

        .left-side h1 {

            font-size: 2.5rem;

            background: rgba(0, 0, 0, 0.5);

            padding: 20px;

            border-radius: 10px;

        }

        .right-side {

            flex: 1;

            display: flex;

            justify-content: center;

            align-items: center;

            background-color: #f8f9fa; /\* Light gray background \*/

            padding: 20px;

            animation: slideInFromRight 1s ease-in-out;

            overflow: auto; /\* Allow scrolling if content overflows \*/

        }

        .login-container {

            width: 100%;

            max-width: 400px;

            padding: 20px;

            background-color: #fff;

            border-radius: 10px;

            box-shadow: 0 0 20px rgba(0, 0, 0, 0.1);

        }

        .login-container h2 {

            color: #00A86B; /\* Jade green for the heading \*/

            font-size: 2rem;

            margin-bottom: 15px;

            text-align: center;

        }

        label {

            color: #007B5E; /\* Darker green for labels \*/

            font-weight: bold;

            margin-top: 5px;

            display: block;

        }

        input {

            width: 100%;

            padding: 8px;

            margin: 5px 0 10px 0; /\* Reduced margin \*/

            border: 1px solid #00A86B; /\* Jade green border \*/

            border-radius: 5px;

            font-size: 0.9rem;

            transition: border-color 0.3s ease, box-shadow 0.3s ease;

        }

        input:focus {

            border-color: #007B5E; /\* Darker green on focus \*/

            box-shadow: 0 0 8px rgba(0, 168, 107, 0.3); /\* Glow effect on focus \*/

        }

        .password-container {

            position: relative;

        }

        #togglePassword {

            position: absolute;

            right: 10px;

            top: 50%;

            transform: translateY(-50%);

            cursor: pointer;

            color: #007B5E; /\* Darker green for the eye icon \*/

        }

        button {

            background-color: #00A86B; /\* Jade green for the button \*/

            color: #fff;

            border: none;

            padding: 10px 20px;

            font-size: 1rem;

            cursor: pointer;

            border-radius: 5px;

            width: 100%;

            margin-top: 10px;

            transition: background-color 0.3s ease, transform 0.3s ease;

        }

        button:hover {

            background-color: #007B5E; /\* Darker green on hover \*/

            transform: translateY(-2px);

        }

        button:active {

            transform: translateY(0);

        }

        .login-container p {

            text-align: center;

            margin-top: 15px;

        }

        .login-container a {

            color: #00A86B; /\* Jade green for links \*/

            text-decoration: none;

            font-weight: bold;

        }

        .login-container a:hover {

            color: #007B5E; /\* Darker green on hover \*/

        }

        /\* Keyframes for animations \*/

        @keyframes fadeIn {

            from { opacity: 0; }

            to { opacity: 1; }

        }

        @keyframes slideInFromRight {

            from { transform: translateX(100px); opacity: 0; }

            to { transform: translateX(0); opacity: 1; }

        }

        /\* Responsive Design \*/

        @media (max-width: 768px) {

            .split-screen {

                flex-direction: column;

                padding-top: 60px; /\* Adjust padding for mobile \*/

            }

            .left-side, .right-side {

                flex: 1;

            }

            .left-side h1 {

                font-size: 2rem;

            }

            .login-container {

                max-width: 90%;

            }

        }

    </style>

    <script>

        // Toggle password visibility

        document.addEventListener("DOMContentLoaded", function () {

            const togglePassword = document.getElementById("togglePassword");

            const passwordInput = document.getElementById("password");

            togglePassword.addEventListener("click", function () {

                if (passwordInput.type === "password") {

                    passwordInput.type = "text";

                    togglePassword.textContent = "🙈"; // Hide icon

                } else {

                    passwordInput.type = "password";

                    togglePassword.textContent = "👁️"; // Show icon

                }

            });

        });

        // Validate login form

        function validateLogin() {

            const emailOrPhone = document.getElementById("email\_or\_phone").value;

            const password = document.getElementById("password").value;

            if (!emailOrPhone || !password) {

                alert("Please fill in all fields.");

                return false;

            }

            return true;

        }

    </script>

</head>

<body>

    <!-- Navbar -->

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

        <div class="container-fluid">

            <a class="navbar-brand" href="/">Smart Farming Platform</a>

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

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

            </button>

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

                <ul class="navbar-nav ms-auto">

                    <li class="nav-item">

                        <a class="nav-link" href="/">Home</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/about">About Us</a>

                    </li>

                    <li class="nav-item">

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

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/register">Register</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/login">Login</a>

                    </li>

                </ul>

            </div>

        </div>

    </nav>

    <!-- Split Screen -->

    <div class="split-screen">

        <!-- Left Side: Visual Background -->

        <div class="left-side">

            <h1>Welcome Back!</h1>

        </div>

        <!-- Right Side: Login Form -->

        <div class="right-side">

            <div class="login-container">

                <h2>Login</h2>

                <form action="{{ url\_for('login') }}" method="POST" onsubmit="return validateLogin()">

                    <label>Email or Phone:</label>

                    <input type="text" name="email\_or\_phone" id="email\_or\_phone" required>

                    <label>Password:</label>

                    <div class="password-container">

                        <input type="password" name="password" id="password" required>

                        <span id="togglePassword">👁️</span> <!-- Eye Icon -->

                    </div>

                    <button type="submit">Login</button>

                </form>

                <p>Don't have an account? <a href="{{ url\_for('register') }}">Register Here</a></p>

            </div>

        </div>

    </div>

</body>

</html>

**Crop Recommendation.html**

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

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

    <title>Crop Recommendation</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

      body {

        background-color: #f8f9fa; /\* Light gray background \*/

      }

      .navbar {

        background-color: #00A86B !important; /\* Jade green for the navbar \*/

      }

      .navbar-brand, .nav-link {

        color: #fff !important; /\* White text for navbar \*/

      }

      .container {

        background-color: #fff; /\* White background for the container \*/

        padding: 20px;

        border-radius: 10px;

        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        animation: fadeIn 1s ease-in-out; /\* Fade-in animation for the container \*/

      }

      h1 {

        color: #00A86B; /\* Jade green for the heading \*/

        font-size: 2.5rem;

        margin-bottom: 20px;

        animation: slideInFromTop 1s ease-in-out; /\* Slide-in animation for the heading \*/

      }

      .form-control {

        border: 1px solid #00A86B; /\* Jade green border for inputs \*/

        transition: border-color 0.3s ease, box-shadow 0.3s ease; /\* Smooth transition for focus \*/

      }

      .form-control:focus {

        border-color: #007B5E; /\* Darker green on focus \*/

        box-shadow: 0 0 8px rgba(0, 168, 107, 0.5); /\* Glow effect on focus \*/

      }

      .btn-primary {

        background-color: #00A86B; /\* Jade green for the button \*/

        border: none;

        font-size: 1.2rem;

        padding: 10px 20px;

        transition: background-color 0.3s ease, transform 0.3s ease; /\* Smooth hover and click effects \*/

      }

      .btn-primary:hover {

        background-color: #007B5E; /\* Darker green on hover \*/

        transform: translateY(-2px); /\* Slight lift on hover \*/

      }

      .btn-primary:active {

        transform: translateY(0); /\* Reset on click \*/

      }

      .card {

        background-color: #E8F5E9; /\* Very light green for the card \*/

        border: 1px solid #00A86B; /\* Jade green border \*/

        border-radius: 10px;

        margin-top: 20px;

        animation: fadeIn 1.5s ease-in-out; /\* Fade-in animation for the card \*/

      }

      .card-title {

        color: #00A86B; /\* Jade green for the card title \*/

        font-size: 1.8rem;

      }

      .card-text {

        color: #333; /\* Dark gray for the card text \*/

        font-size: 1.5rem;

        font-weight: bold;

      }

      label {

        color: #007B5E; /\* Darker green for labels \*/

        font-weight: bold;

      }

      /\* Keyframes for animations \*/

      @keyframes fadeIn {

        from {

          opacity: 0;

        }

        to {

          opacity: 1;

        }

      }

      @keyframes slideInFromTop {

        from {

          transform: translateY(-50px);

          opacity: 0;

        }

        to {

          transform: translateY(0);

          opacity: 1;

        }

      }

    </style>

  </head>

  <body>

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

      <div class="container-fluid">

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

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

          <ul class="navbar-nav ms-auto">

            <li class="nav-item">

              <a class="nav-link" href="/dashboard">Dashboard</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="/logout">Logout</a>

            </li>

          </ul>

        </div>

      </div>

    </nav>

    <div class="container mt-5">

      <h1 class="text-center">Crop Recommendation</h1>

      <form action="/crop\_recommendation" method="POST">

        <div class="row mt-4">

          <div class="col-md-4">

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

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

          </div>

          <div class="col-md-4">

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

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

          </div>

          <div class="col-md-4">

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

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

          </div>

        </div>

        <div class="row mt-4">

          <div class="col-md-4">

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

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

          </div>

          <div class="col-md-4">

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

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

          </div>

          <div class="col-md-4">

            <label for="pH">pH</label>

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

          </div>

        </div>

        <div class="row mt-4">

          <div class="col-md-4">

            <label for="Rainfall">Rainfall</label>

            <input type="number" step="0.01" id="Rainfall" name="Rainfall" placeholder="Enter Rainfall in mm" class="form-control" required>

          </div>

        </div>

        <div class="row mt-4">

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

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

          </div>

        </div>

      </form>

      {% if result %}

      <div class="card mt-4">

        <div class="card-body">

          <h5 class="card-title">Recommended Crop:</h5>

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

        </div>

      </div>

      {% endif %}

    </div>

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

  </body>

</html>

**Fertilizer Recommendation Module:**

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

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

    <title>Fertilizer Recommendation</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

      body {

        background-color: #f8f9fa; /\* Light gray background \*/

      }

      .navbar {

        background-color: #00A86B !important; /\* Jade green for the navbar \*/

      }

      .navbar-brand, .nav-link {

        color: #fff !important; /\* White text for navbar \*/

      }

      .container {

        background-color: #fff; /\* White background for the container \*/

        padding: 20px;

        border-radius: 10px;

        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        animation: fadeIn 1s ease-in-out; /\* Fade-in animation for the container \*/

      }

      h1 {

        color: #00A86B; /\* Jade green for the heading \*/

        font-size: 2.5rem;

        margin-bottom: 20px;

        animation: slideInFromTop 1s ease-in-out; /\* Slide-in animation for the heading \*/

      }

      .form-control {

        border: 1px solid #00A86B; /\* Jade green border for inputs \*/

        transition: border-color 0.3s ease, box-shadow 0.3s ease; /\* Smooth transition for focus \*/

      }

      .form-control:focus {

        border-color: #007B5E; /\* Darker green on focus \*/

        box-shadow: 0 0 8px rgba(0, 168, 107, 0.5); /\* Glow effect on focus \*/

      }

      .btn-primary {

        background-color: #00A86B; /\* Jade green for the button \*/

        border: none;

        font-size: 1.2rem;

        padding: 10px 20px;

        transition: background-color 0.3s ease, transform 0.3s ease; /\* Smooth hover and click effects \*/

      }

      .btn-primary:hover {

        background-color: #007B5E; /\* Darker green on hover \*/

        transform: translateY(-2px); /\* Slight lift on hover \*/

      }

      .btn-primary:active {

        transform: translateY(0); /\* Reset on click \*/

      }

      .card {

        background-color: #E8F5E9; /\* Very light green for the card \*/

        border: 1px solid #00A86B; /\* Jade green border \*/

        border-radius: 10px;

        margin-top: 20px;

        animation: fadeIn 1.5s ease-in-out; /\* Fade-in animation for the card \*/

      }

      .card-title {

        color: #00A86B; /\* Jade green for the card title \*/

        font-size: 1.8rem;

      }

      .card-text {

        color: #333; /\* Dark gray for the card text \*/

        font-size: 1.5rem;

        font-weight: bold;

      }

      label {

        color: #007B5E; /\* Darker green for labels \*/

        font-weight: bold;

      }

      /\* Keyframes for animations \*/

      @keyframes fadeIn {

        from {

          opacity: 0;

        }

        to {

          opacity: 1;

        }

      }

      @keyframes slideInFromTop {

        from {

          transform: translateY(-50px);

          opacity: 0;

        }

        to {

          transform: translateY(0);

          opacity: 1;

        }

      }

    </style>

  </head>

  <body>

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

      <div class="container-fluid">

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

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

          <ul class="navbar-nav ms-auto">

            <li class="nav-item">

              <a class="nav-link" href="/dashboard">Dashboard</a>

            </li>

            <li class="nav-item">

              <a class="nav-link" href="/logout">Logout</a>

            </li>

          </ul>

        </div>

      </div>

    </nav>

    <div class="container mt-5">

      <h1 class="text-center">Fertilizer Recommendation</h1>

      <form action="/fertilizer\_recommendation" method="POST">

        <div class="row mt-4">

          <div class="col-md-4">

            <label for="crop">Crop</label>

            <select id="crop" name="crop" class="form-control" required>

              <option value="">Select Crop</option>

              {% for crop in crops %}

              <option value="{{ crop }}">{{ crop }}</option>

              {% endfor %}

            </select>

          </div>

          <div class="col-md-4">

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

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

          </div>

          <div class="col-md-4">

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

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

          </div>

        </div>

        <div class="row mt-4">

          <div class="col-md-4">

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

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

          </div>

        </div>

        <div class="row mt-4">

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

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

          </div>

        </div>

      </form>

      {% if result %}

      <div class="card mt-4">

        <div class="card-body">

          <h5 class="card-title">Fertilizer Recommendation:</h5>

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

        </div>

      </div>

      {% endif %}

    </div>

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

  </body>

</html>

**Weed Detection Module:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Weed Detection</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

        body {

            background-color: #f8f9fa; /\* Light gray background \*/

            font-family: 'Arial', sans-serif;

        }

        .navbar {

            background-color: #00A86B !important; /\* Jade green for the navbar \*/

        }

        .navbar-brand, .nav-link {

            color: #fff !important; /\* White text for navbar \*/

        }

        .container {

            background-color: #fff; /\* White background for the container \*/

            padding: 20px;

            border-radius: 10px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

            margin-top: 20px;

            animation: fadeIn 1s ease-in-out; /\* Fade-in animation for the container \*/

        }

        h1 {

            color: #00A86B; /\* Jade green for the heading \*/

            font-size: 2.5rem;

            margin-bottom: 20px;

            animation: slideInFromTop 1s ease-in-out; /\* Slide-in animation for the heading \*/

        }

        .upload-form {

            margin: 20px 0;

        }

        .upload-form input[type="file"] {

            display: none; /\* Hide the default file input \*/

        }

        .upload-form label {

            background-color: #00A86B; /\* Jade green for the upload button \*/

            color: #fff;

            padding: 10px 20px;

            border-radius: 5px;

            cursor: pointer;

            transition: background-color 0.3s;

        }

        .upload-form label:hover {

            background-color: #008c5a; /\* Darker green on hover \*/

        }

        .result {

            margin-top: 20px;

            padding: 20px;

            background-color: #e8f5e9; /\* Light green background for the result \*/

            border-radius: 5px;

            border: 1px solid #00A86B; /\* Jade green border \*/

        }

        .result h2 {

            color: #00A86B; /\* Jade green for the result heading \*/

            font-size: 1.8rem;

        }

        .result p {

            color: #333; /\* Dark gray for the result text \*/

            font-size: 1.2rem;

        }

        /\* Keyframes for animations \*/

        @keyframes fadeIn {

            from {

                opacity: 0;

            }

            to {

                opacity: 1;

            }

        }

        @keyframes slideInFromTop {

            from {

                transform: translateY(-50px);

                opacity: 0;

            }

            to {

                transform: translateY(0);

                opacity: 1;

            }

        }

    </style>

</head>

<body>

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

        <div class="container-fluid">

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

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

                <ul class="navbar-nav ms-auto">

                    <li class="nav-item">

                        <a class="nav-link" href="/dashboard">Back to Dashboard</a>

                    </li>

                    <li class="nav-item">

                        <a class="nav-link" href="/logout">Logout</a>

                    </li>

                </ul>

            </div>

        </div>

    </nav>

    <div class="container">

        <h1 class="text-center">Weed Detection</h1>

        <form action="/weed\_detection" method="post" enctype="multipart/form-data" class="upload-form text-center">

            <label for="file-upload" class="btn btn-primary">

                Choose Image

            </label>

            <input id="file-upload" type="file" name="file" accept="image/\*" required>

            <button type="submit" class="btn btn-success">Upload and Detect</button>

        </form>

        {% if result %}

            <div class="result text-center">

                <h2>Prediction Result:</h2>

                <p>{{ result }}</p>

            </div>

        {% endif %}

    </div>

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

</body>

</html>

**Chatbot Module:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>AgriGenius</title>

    <link href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&display=swap" rel="stylesheet">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">

    <style>

    body {

        font-family: 'Roboto', sans-serif;

        background-color: #f0f4f8;

        background: linear-gradient(to bottom, #87CEEB, #E0F6FF);

        display: flex;

        justify-content: center;

        align-items: center;

        height: 100vh;

        margin: 0;

    }

    /\* Rest of the CSS from the provided stylesheet goes here \*/

    .chat-container {

        width: 470px;

        height: 650px;

        background-color: rgba(255, 255, 255, 0.9);

        border-radius: 20px;

        box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1),

                    0 4px 8px rgba(0, 0, 0, 0.3),

                    0 10px 20px rgba(0, 0, 0, 0.5);

        display: flex;

        flex-direction: column;

        overflow: hidden;

        position: relative;

    }

    /\* Include all CSS styles from the previous stylesheet \*/

    .chat-header {

        background-color: #4fd1c5;

        color: white;

        padding: 15px 20px;

        text-align: left;

    }

    .chat-header h1 {

        margin: 0;

        font-size: 24px;

        font-weight: 500;

    }

    .chat-messages {

        flex: 1;

        overflow-y: auto;

        padding: 20px;

        display: flex;

        flex-direction: column;

    }

    .message {

        max-width: calc(100% - 50px);

        margin-bottom: 15px;

        padding: 10px 15px;

        border-radius: 20px;

        line-height: 1.4;

        font-size: 14px;

    }

    .bot-message {

        background-color: #e6f6f4;

        align-self: flex-start;

        color: #333;

        margin-right: auto;

    }

    .user-message {

        background-color: #4fd1c5;

        color: white;

        align-self: flex-end;

        margin-left: 10px;

    }

    .chat-input {

        display: flex;

        padding: 20px;

        background-color: #f9f9f9;

        border-top: 1px solid #eee;

    }

    #chatbot-form-btn, #chatbot-form-btn-voice {

        background-color: #4fd1c5;

        color: white;

        border: none;

        border-radius: 50%;

        width: 40px;

        height: 40px;

        margin-left: 10px;

        cursor: pointer;

        display: flex;

        align-items: center;

        justify-content: center;

        transition: background-color 0.3s ease;

    }

    #chatbot-form-btn:hover, #chatbot-form-btn-voice:hover {

        background-color: #3cb3a7;

    }

    .typing-indicator {

        display: flex;

        align-items: center;

        margin-bottom: 15px;

    }

    .typing-indicator span {

        height: 8px;

        width: 8px;

        background-color: #4fd1c5;

        border-radius: 50%;

        display: inline-block;

        margin-right: 5px;

        animation: typing 1s infinite ease-in-out;

    }

    .typing-indicator span:nth-child(2) {

        animation-delay: 0.2s;

    }

    .typing-indicator span:nth-child(3) {

        animation-delay: 0.4s;

    }

    @keyframes typing {

        0% { transform: translateY(0); }

        50% { transform: translateY(-5px); }

        100% { transform: translateY(0); }

    }

    .chat-footer {

        padding: 10px 20px;

        background-color: #f9f9f9;

        border-top: 1px solid #eee;

        display: flex;

        justify-content: space-between;

        align-items: center;

    }

    #chatbot-form-btn-clear {

        background-color: #f44336;

        color: white;

        border: none;

        padding: 5px 10px;

        border-radius: 5px;

        cursor: pointer;

        transition: background-color 0.3s ease;

    }

    #chatbot-form-btn-clear:hover {

        background-color: #d32f2f;

    }

    .voice-reading {

        display: flex;

        align-items: center;

    }

    .voice-reading input[type="checkbox"] {

        margin-right: 5px;

    }

    #chatbot-form {

        display: flex;

        width: 100%;

    }

    #messageText {

        flex-grow: 1;

        padding: 10px 15px;

        border: 1.6px solid #ccc;

        border-radius: 20px;

        background-color: white;

        font-size: 14px;

        transition: border-color 0.3s ease;

    }

    #messageText:focus {

        outline: none;

        border-color: #4fd1c5;

        box-shadow: 0 0 0 3px rgba(79, 209, 197, 0.3);

    }

    .logo-title {

        display: flex;

        align-items: center;

        justify-content: flex-start;

    }

    .logo-circle {

        width: 43px;

        height: 43px;

        border-radius: 50%;

        overflow: hidden;

        margin-right: 10px;

        background-color: white;

        display: flex;

        align-items: center;

        justify-content: center;

        flex-shrink: 0;

    }

    .logo-image {

        width: 90%;

        height: 90%;

        object-fit: contain;

    }

    .chat-container::before {

        content: "";

        position: absolute;

        top: 0;

        left: 0;

        right: 0;

        bottom: 0;

        background-color: rgba(255, 255, 255, 0.5);

        z-index: -1;

    }

    .message-container {

        display: flex;

        align-items: flex-start;

        margin-bottom: 15px;

        max-width: 80%;

    }

    .bot-container {

        align-self: flex-start;

    }

    .user-container {

        align-self: flex-end;

        justify-content: flex-end;

        margin-left: auto;

        flex-direction: row-reverse;

    }

    .bot-logo, .user-image {

        width: 30px;

        height: 30px;

        border-radius: 50%;

        overflow: hidden;

        flex-shrink: 0;

    }

    .bot-logo {

        margin-right: 10px;

    }

    .user-image {

        margin-left: 10px;

    }

    .bot-logo img, .user-image img {

        width: 100%;

        height: 100%;

        object-fit: cover;

    }

    </style>

</head>

<body>

    <div class="chat-container">

        <div class="chat-header">

            <div class="logo-title">

                <div class="logo-circle">

                    <img src="https://cdn-icons-png.flaticon.com/512/2903/2903226.png" alt="SasyaVani" class="logo-image">

                </div>

                <h1>SasyaVani</h1>

            </div>

        </div>

        <div class="chat-messages" id="chatPanel">

            <!-- Messages will be appended here -->

        </div>

        <div class="chat-input">

            <form id="chatbot-form">

                <input type="text" id="messageText" name="messageText" placeholder="Type your message...">

                <button id="chatbot-form-btn" type="submit">

                    <i class="fas fa-paper-plane"></i>

                </button>

                <button id="chatbot-form-btn-voice" type="button">

                    <i class="fas fa-microphone"></i>

                </button>

            </form>

        </div>

        <div class="chat-footer">

            <button id="chatbot-form-btn-clear">Clear Chat</button>

            <div class="voice-reading">

                <input type="checkbox" id="voiceReadingCheckbox">

                <label for="voiceReadingCheckbox">Enable voice reading</label>

            </div>

        </div>

    </div>

    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>

    <script>

    $(function() {

        var synth = window.speechSynthesis;

        var msg = new SpeechSynthesisUtterance();

        var voices = synth.getVoices();

        msg.voice = voices[0];

        msg.rate = 1;

        msg.pitch = 1;

        function appendMessage(message, isUser) {

            var messageClass = isUser ? 'user-message' : 'bot-message';

            var logoHTML = isUser ? '' : '<div class="bot-logo"><img src="https://cdn-icons-png.flaticon.com/512/4712/4712027.png" alt="SasyaVani"></div>';

            var userImageHTML = isUser ? '<div class="user-image"><img src="https://cdn-icons-png.flaticon.com/512/456/456212.png" alt="User"></div>' : '';

            var messageElement = $('<div class="message-container ' + (isUser ? 'user-container' : 'bot-container') + '">' +

                                logoHTML +

                                '<div class="message ' + messageClass + '"></div>' +

                                userImageHTML +

                               '</div>');

            $('.chat-messages').append(messageElement);

            if (isUser) {

                messageElement.find('.message').text(message);

            } else {

                typeMessage(message, messageElement.find('.message'));

            }

            $('.chat-messages').scrollTop($('.chat-messages')[0].scrollHeight);

        }

        function typeMessage(message, element, speed = 15) {

            let i = 0;

            element.html('');

            const typingInterval = setInterval(() => {

                if (i < message.length) {

                    element.html(element.html() + message.charAt(i));

                    i++;

                } else {

                    clearInterval(typingInterval);

                }

                $('.chat-messages').scrollTop($('.chat-messages')[0].scrollHeight);

            }, speed);

        }

        function showTypingIndicator() {

            var typingIndicator = $('<div class="typing-indicator bot-message"><span></span><span></span><span></span></div>');

            $('.chat-messages').append(typingIndicator);

            $('.chat-messages').scrollTop($('.chat-messages')[0].scrollHeight);

        }

        function removeTypingIndicator() {

            $('.typing-indicator').remove();

        }

        $('#chatbot-form-btn').click(function(e) {

            e.preventDefault();

            sendMessage();

        });

        $('#messageText').keypress(function(e) {

            if (e.which == 13) {

                e.preventDefault();

                sendMessage();

            }

        });

        var isProcessing = false;

        function disableInput() {

            $('#messageText').prop('disabled', true);

            $('#chatbot-form-btn').prop('disabled', true);

            $('#chatbot-form-btn-voice').prop('disabled', true);

        }

        function enableInput() {

            $('#messageText').prop('disabled', false);

            $('#chatbot-form-btn').prop('disabled', false);

            $('#chatbot-form-btn-voice').prop('disabled', false);

        }

        function sendMessage() {

            var message = $('#messageText').val().trim();

            if (message && !isProcessing) {

                isProcessing = true;

                disableInput();

                appendMessage(message, true);

                $('#messageText').val('');

                showTypingIndicator();

                $.ajax({

                    type: "POST",

                    url: "/ask",

                    data: { messageText: message },

                    success: function(response) {

                        removeTypingIndicator();

                        if (response.error) {

                            appendMessage("Error: " + response.error, false);

                        } else {

                            var answer = response.answer;

                            appendMessage(answer, false);

                            if ($('#voiceReadingCheckbox').is(':checked')) {

                                msg.text = answer;

                                synth.speak(msg);

                            }

                        }

                        isProcessing = false;

                        enableInput();

                    },

                    error: function(jqXHR, textStatus, errorThrown) {

                        removeTypingIndicator();

                        console.log(errorThrown);

                        appendMessage("Sorry, there was an error processing your request. Please try again later.", false);

                        isProcessing = false;

                        enableInput();

                    }

                });

            }

        }

        var welcomeMessage = "🌱🌾 Welcome to SasyaVani !! 🌾🌱 Hi there! I'm SasyaVani, your virtual assistant for Agriculture Queries. How can I assist you today?";

        $('#chatbot-form-btn-clear').click(function(e) {

            e.preventDefault();

            $('.chat-messages').empty();

            appendMessage(welcomeMessage, false);

        });

        $('#chatbot-form-btn-voice').click(function(e) {

            e.preventDefault();

            if ('webkitSpeechRecognition' in window && !isProcessing) {

                var recognition = new webkitSpeechRecognition();

                recognition.lang = 'en-US';

                recognition.interimResults = false;

                recognition.maxAlternatives = 1;

                recognition.start();

                recognition.onresult = function(event) {

                    var speechResult = event.results[0][0].transcript;

                    $('#messageText').val(speechResult);

                    sendMessage();

                };

                recognition.onerror = function(event) {

                    console.error('Speech recognition error:', event.error);

                };

            } else {

                console.log('Web Speech API is not supported in this browser or processing is in progress');

            }

        });

        $('#voiceReadingCheckbox').change(function() {

            if (!$(this).is(':checked')) {

                synth.cancel();

            }

        });

        setTimeout(function() {

            appendMessage(welcomeMessage, false);

        }, 500);

    });

    </script>

</body>

</html>

**DashBoard.html**

<!doctype html>

<html lang="en">

  <head>

    <meta charset="utf-8">

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

    <title>Dashboard</title>

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet">

    <style>

      body {

        background-color: #f8f9fa; /\* Light gray background \*/

      }

      .navbar {

        background-color: #00A86B !important; /\* Jade green for the navbar \*/

      }

      .navbar-brand, .nav-link {

        color: #fff !important; /\* White text for navbar \*/

      }

      .container {

        background-color: #fff; /\* White background for the container \*/

        padding: 20px;

        border-radius: 10px;

        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

        animation: fadeIn 1s ease-in-out; /\* Fade-in animation for the container \*/

      }

      h1 {

        color: #00A86B; /\* Jade green for the heading \*/

        font-size: 2.5rem;

        margin-bottom: 20px;

        animation: slideInFromTop 1s ease-in-out; /\* Slide-in animation for the heading \*/

      }

      .card {

        margin: 20px;

        cursor: pointer;

        transition: transform 0.2s, box-shadow 0.2s;

        background-color: #E8F5E9; /\* Very light green for the card \*/

        border: 1px solid #00A86B; /\* Jade green border \*/

        border-radius: 10px;

      }

      .card:hover {

        transform: scale(1.05);

        box-shadow: 0 0 15px rgba(0, 168, 107, 0.3); /\* Glow effect on hover \*/

      }

      .card-title {

        color: #00A86B; /\* Jade green for the card title \*/

        font-size: 1.8rem;

      }

      .card-text {

        color: #333; /\* Dark gray for the card text \*/

        font-size: 1.2rem;

      }

      /\* Keyframes for animations \*/

      @keyframes fadeIn {

        from {

          opacity: 0;

        }

        to {

          opacity: 1;

        }

      }

      @keyframes slideInFromTop {

        from {

          transform: translateY(-50px);

          opacity: 0;

        }

        to {

          transform: translateY(0);

          opacity: 1;

        }

      }

    </style>

  </head>

  <body>

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

      <div class="container-fluid">

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

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

          <ul class="navbar-nav ms-auto">

            <li class="nav-item">

              <a class="nav-link" href="/logout">Logout</a>

            </li>

          </ul>

        </div>

      </div>

    </nav>

    <div class="container mt-5">

      <h1 class="text-center">Welcome, {{ username }}!</h1>

      <div class="row mt-5">

        <div class="col-md-3">

          <div class="card text-center" onclick="window.location.href='/crop\_recommendation'">

            <div class="card-body">

              <h5 class="card-title">Crop Recommendation</h5>

              <p class="card-text">Get the best crop recommendation based on soil and weather conditions.</p>

            </div>

          </div>

        </div>

        <div class="col-md-3">

          <div class="card text-center" onclick="window.location.href='/fertilizer\_recommendation'">

            <div class="card-body">

              <h5 class="card-title">Fertilizer Recommendation</h5>

              <p class="card-text">Get fertilizer recommendations for your crop based on soil nutrient levels.</p>

            </div>

          </div>

        </div>

        <div class="col-md-3">

          <div class="card text-center" onclick="window.location.href='/weed\_detection'">

            <div class="card-body">

              <h5 class="card-title">Weed Detection</h5>

              <p class="card-text">Detect weeds in your field using image analysis.</p>

            </div>

          </div>

        </div>

        <div class="col-md-3">

          <div class="card text-center" onclick="window.location.href='/chatbot'">

            <div class="card-body">

              <h5 class="card-title">AgriGenius Chatbot</h5>

              <p class="card-text">Get instant agricultural advice and support from our AI assistant.</p>

            </div>

          </div>

        </div>

      </div>

    </div>

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

  </body>

</html>

**Styles.css**

/\* ========== Global Styling ========== \*/

body {

    font-family: Arial, sans-serif;

    background-color: #f4f4f4;

    margin: 0;

    padding: 0;

    text-align: center;

}

/\* Header \*/

header {

    background-color: #28a745;

    color: white;

    padding: 15px;

    font-size: 24px;

}

/\* Navigation Bar \*/

nav {

    background-color: #333;

    padding: 10px;

}

nav a {

    color: white;

    text-decoration: none;

    padding: 10px 20px;

    margin: 5px;

    display: inline-block;

    font-size: 18px;

    transition: background 0.3s ease-in-out;

}

nav a:hover {

    background-color: #575757;

    border-radius: 5px;

}

/\* ========== Form Styling (Common for Login & Registration) ========== \*/

form {

    background: white;

    padding: 25px;

    border-radius: 10px;

    box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.1);

    width: 100%;

    max-width: 420px;

    margin: 20px auto;

    text-align: left;

    box-sizing: border-box;

}

label {

    display: block;

    font-weight: bold;

    margin: 12px 0 5px;

}

input, select {

    width: 100%;

    padding: 12px;

    margin-bottom: 12px;

    border: 1px solid #ccc;

    border-radius: 5px;

    font-size: 16px;

    transition: border-color 0.3s;

    box-sizing: border-box;

}

input:focus, select:focus {

    border-color: #28a745;

    outline: none;

}

/\* Buttons \*/

button {

    background-color: #28a745;

    color: white;

    border: none;

    padding: 12px;

    cursor: pointer;

    font-size: 16px;

    border-radius: 5px;

    width: 100%;

    margin-top: 12px;

    transition: background 0.3s ease-in-out;

    box-sizing: border-box;

}

button:hover {

    background-color: #218838;

}

button:disabled {

    background-color: #ccc;

    cursor: not-allowed;

}

/\* ========== Login Page Styling ========== \*/

.login-container {

    background: white;

    padding: 25px;

    border-radius: 10px;

    box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.1);

    width: 100%;

    max-width: 350px;

    margin: auto;

    margin-top: 50px;

    text-align: left;

    box-sizing: border-box;

}

h2 {

    font-size: 24px;

    margin-bottom: 20px;

}

/\* Password Field with Toggle \*/

.password-container {

    position: relative;

}

#togglePassword {

    position: absolute;

    right: 12px;

    top: 50%;

    transform: translateY(-50%);

    cursor: pointer;

    font-size: 18px;

    color: #777;

}

/\* Responsive Design \*/

@media (max-width: 500px) {

    form {

        width: 92%;

    }

    button {

        font-size: 14px;

        padding: 10px;

    }

    nav a {

        font-size: 16px;

        padding: 8px;

    }

}

**Script.js**

document.addEventListener("DOMContentLoaded", function () {

    const togglePassword = document.getElementById("togglePassword");

    const passwordField = document.getElementById("password");

    // Set initial state of the password visibility icon based on password type

    togglePassword.textContent = passwordField.type === "password" ? "👁️" : "🙈";

    // Toggle password visibility

    togglePassword.addEventListener("click", function () {

        if (passwordField.type === "password") {

            passwordField.type = "text";

            togglePassword.textContent = "🙈"; // Change icon to hide

        } else {

            passwordField.type = "password";

            togglePassword.textContent = "👁️"; // Change icon to show

        }

    });

});

// Validate login form before submission

function validateLogin() {

    const emailOrPhone = document.getElementById("email\_or\_phone").value;

    const password = document.getElementById("password").value;

    if (emailOrPhone.trim() === "" || password.trim() === "") {

        alert("Please fill in all fields!");

        return false; // Prevent form submission

    }

    return true;

}

**States.js**

document.addEventListener("DOMContentLoaded", function () {

    const stateSelect = document.getElementById("state");

    const districtSelect = document.getElementById("district");

    // Show loading message

    stateSelect.innerHTML = "<option>Loading...</option>";

    // Fetch JSON file containing states and districts

    fetch("/static/states\_districts.json")

        .then(response => response.json())

        .then(data => {

            stateSelect.innerHTML = "<option value=''>Select State</option>"; // Reset state dropdown

            // Populate states dynamically

            Object.keys(data).forEach(state => {

                let option = document.createElement("option");

                option.value = state;

                option.textContent = state;

                stateSelect.appendChild(option);

            });

            // Update districts when a state is selected

            stateSelect.addEventListener("change", function () {

                let selectedState = stateSelect.value;

                districtSelect.innerHTML = "<option value=''>Select District</option>"; // Reset district dropdown

                if (selectedState && data[selectedState]) {

                    data[selectedState].forEach(district => {

                        let option = document.createElement("option");

                        option.value = district;

                        option.textContent = district;

                        districtSelect.appendChild(option);

                    });

                }

            });

        })

        .catch(error => {

            console.error("Error loading states:", error);

            stateSelect.innerHTML = "<option value=''>Failed to load states</option>";

        });

});

**states\_districts.json:**

{

  "Andhra Pradesh"

         :[

            "Anantapur",

            "Chittoor",

            "East Godavari",

            "Guntur",

            "Krishna",

            "Kurnool",

            "Nellore",

            "Prakasam",

            "Srikakulam",

            "Visakhapatnam",

            "Vizianagaram",

            "West Godavari",

            "YSR Kadapa"

         ],

         "Arunachal Pradesh":[

            "Tawang",

            "West Kameng",

            "East Kameng",

            "Papum Pare",

            "Kurung Kumey",

            "Kra Daadi",

            "Lower Subansiri",

            "Upper Subansiri",

            "West Siang",

            "East Siang",

            "Siang",

            "Upper Siang",

            "Lower Siang",

            "Lower Dibang Valley",

            "Dibang Valley",

            "Anjaw",

            "Lohit",

            "Namsai",

            "Changlang",

            "Tirap",

            "Longding"

         ],

      "Assam":[

            "Baksa",

            "Barpeta",

            "Biswanath",

            "Bongaigaon",

            "Cachar",

            "Charaideo",

            "Chirang",

            "Darrang",

            "Dhemaji",

            "Dhubri",

            "Dibrugarh",

            "Goalpara",

            "Golaghat",

            "Hailakandi",

            "Hojai",

            "Jorhat",

            "Kamrup Metropolitan",

            "Kamrup",

            "Karbi Anglong",

            "Karimganj",

            "Kokrajhar",

            "Lakhimpur",

            "Majuli",

            "Morigaon",

            "Nagaon",

            "Nalbari",

            "Dima Hasao",

            "Sivasagar",

            "Sonitpur",

            "South Salmara-Mankachar",

            "Tinsukia",

            "Udalguri",

            "West Karbi Anglong"

         ],

      "Bihar":[

            "Araria",

            "Arwal",

            "Aurangabad",

            "Banka",

            "Begusarai",

            "Bhagalpur",

            "Bhojpur",

            "Buxar",

            "Darbhanga",

            "East Champaran (Motihari)",

            "Gaya",

            "Gopalganj",

            "Jamui",

            "Jehanabad",

            "Kaimur (Bhabua)",

            "Katihar",

            "Khagaria",

            "Kishanganj",

            "Lakhisarai",

            "Madhepura",

            "Madhubani",

            "Munger (Monghyr)",

            "Muzaffarpur",

            "Nalanda",

            "Nawada",

            "Patna",

            "Purnia (Purnea)",

            "Rohtas",

            "Saharsa",

            "Samastipur",

            "Saran",

            "Sheikhpura",

            "Sheohar",

            "Sitamarhi",

            "Siwan",

            "Supaul",

            "Vaishali",

            "West Champaran"

         ]

      ,

      "Chandigarh (UT)":[

            "Chandigarh"

         ]

      ,

      "Chhattisgarh"

         :[

            "Balod",

            "Baloda Bazar",

            "Balrampur",

            "Bastar",

            "Bemetara",

            "Bijapur",

            "Bilaspur",

            "Dantewada (South Bastar)",

            "Dhamtari",

            "Durg",

            "Gariyaband",

            "Janjgir-Champa",

            "Jashpur",

            "Kabirdham (Kawardha)",

            "Kanker (North Bastar)",

            "Kondagaon",

            "Korba",

            "Korea (Koriya)",

            "Mahasamund",

            "Mungeli",

            "Narayanpur",

            "Raigarh",

            "Raipur",

            "Rajnandgaon",

            "Sukma",

            "Surajpur  ",

            "Surguja"

         ],

         "Dadra and Nagar Haveli (UT)":[

            "Dadra & Nagar Haveli"

         ]

      ,

      "Daman and Diu (UT)":[

            "Daman",

            "Diu"

         ]

      ,

     "Delhi (NCT)":[

            "Central Delhi",

            "East Delhi",

            "New Delhi",

            "North Delhi",

            "North East  Delhi",

            "North West  Delhi",

            "Shahdara",

            "South Delhi",

            "South East Delhi",

            "South West  Delhi",

            "West Delhi"

         ]

      ,

      "Goa":[

            "North Goa",

            "South Goa"

         ]

      ,

         "Gujarat":[

            "Ahmedabad",

            "Amreli",

            "Anand",

            "Aravalli",

            "Banaskantha (Palanpur)",

            "Bharuch",

            "Bhavnagar",

            "Botad",

            "Chhota Udepur",

            "Dahod",

            "Dangs (Ahwa)",

            "Devbhoomi Dwarka",

            "Gandhinagar",

            "Gir Somnath",

            "Jamnagar",

            "Junagadh",

            "Kachchh",

            "Kheda (Nadiad)",

            "Mahisagar",

            "Mehsana",

            "Morbi",

            "Narmada (Rajpipla)",

            "Navsari",

            "Panchmahal (Godhra)",

            "Patan",

            "Porbandar",

            "Rajkot",

            "Sabarkantha (Himmatnagar)",

            "Surat",

            "Surendranagar",

            "Tapi (Vyara)",

            "Vadodara",

            "Valsad"

         ]

      ,

      "Haryana":[

            "Ambala",

            "Bhiwani",

            "Charkhi Dadri",

            "Faridabad",

            "Fatehabad",

            "Gurgaon",

            "Hisar",

            "Jhajjar",

            "Jind",

            "Kaithal",

            "Karnal",

            "Kurukshetra",

            "Mahendragarh",

            "Mewat",

            "Palwal",

            "Panchkula",

            "Panipat",

            "Rewari",

            "Rohtak",

            "Sirsa",

            "Sonipat",

            "Yamunanagar"

         ]

      ,

      "Himachal Pradesh":[

            "Bilaspur",

            "Chamba",

            "Hamirpur",

            "Kangra",

            "Kinnaur",

            "Kullu",

            "Lahaul &amp; Spiti",

            "Mandi",

            "Shimla",

            "Sirmaur (Sirmour)",

            "Solan",

            "Una"

         ]

      ,

      "Jammu and Kashmir":[

            "Anantnag",

            "Bandipore",

            "Baramulla",

            "Budgam",

            "Doda",

            "Ganderbal",

            "Jammu",

            "Kargil",

            "Kathua",

            "Kishtwar",

            "Kulgam",

            "Kupwara",

            "Leh",

            "Poonch",

            "Pulwama",

            "Rajouri",

            "Ramban",

            "Reasi",

            "Samba",

            "Shopian",

            "Srinagar",

            "Udhampur"

         ]

      ,

       "Jharkhand":[

            "Bokaro",

            "Chatra",

            "Deoghar",

            "Dhanbad",

            "Dumka",

            "East Singhbhum",

            "Garhwa",

            "Giridih",

            "Godda",

            "Gumla",

            "Hazaribag",

            "Jamtara",

            "Khunti",

            "Koderma",

            "Latehar",

            "Lohardaga",

            "Pakur",

            "Palamu",

            "Ramgarh",

            "Ranchi",

            "Sahibganj",

            "Seraikela-Kharsawan",

            "Simdega",

            "West Singhbhum"

         ]

      ,

      "Karnataka":[

            "Bagalkot",

            "Ballari (Bellary)",

            "Belagavi (Belgaum)",

            "Bengaluru (Bangalore) Rural",

            "Bengaluru (Bangalore) Urban",

            "Bidar",

            "Chamarajanagar",

            "Chikballapur",

            "Chikkamagaluru (Chikmagalur)",

            "Chitradurga",

            "Dakshina Kannada",

            "Davangere",

            "Dharwad",

            "Gadag",

            "Hassan",

            "Haveri",

            "Kalaburagi (Gulbarga)",

            "Kodagu",

            "Kolar",

            "Koppal",

            "Mandya",

            "Mysuru (Mysore)",

            "Raichur",

            "Ramanagara",

            "Shivamogga (Shimoga)",

            "Tumakuru (Tumkur)",

            "Udupi",

            "Uttara Kannada (Karwar)",

            "Vijayapura (Bijapur)",

            "Yadgir"

         ]

      ,

      "Kerala":[

            "Alappuzha",

            "Ernakulam",

            "Idukki",

            "Kannur",

            "Kasaragod",

            "Kollam",

            "Kottayam",

            "Kozhikode",

            "Malappuram",

            "Palakkad",

            "Pathanamthitta",

            "Thiruvananthapuram",

            "Thrissur",

            "Wayanad"

         ]

      ,

     "Lakshadweep (UT)":[

            "Agatti",

            "Amini",

            "Androth",

            "Bithra",

            "Chethlath",

            "Kavaratti",

            "Kadmath",

            "Kalpeni",

            "Kilthan",

            "Minicoy"

         ]

      ,

      "Madhya Pradesh":[

            "Agar Malwa",

            "Alirajpur",

            "Anuppur",

            "Ashoknagar",

            "Balaghat",

            "Barwani",

            "Betul",

            "Bhind",

            "Bhopal",

            "Burhanpur",

            "Chhatarpur",

            "Chhindwara",

            "Damoh",

            "Datia",

            "Dewas",

            "Dhar",

            "Dindori",

            "Guna",

            "Gwalior",

            "Harda",

            "Hoshangabad",

            "Indore",

            "Jabalpur",

            "Jhabua",

            "Katni",

            "Khandwa",

            "Khargone",

            "Mandla",

            "Mandsaur",

            "Morena",

            "Narsinghpur",

            "Neemuch",

            "Panna",

            "Raisen",

            "Rajgarh",

            "Ratlam",

            "Rewa",

            "Sagar",

            "Satna",

            "Sehore",

            "Seoni",

            "Shahdol",

            "Shajapur",

            "Sheopur",

            "Shivpuri",

            "Sidhi",

            "Singrauli",

            "Tikamgarh",

            "Ujjain",

            "Umaria",

            "Vidisha"

         ]

      ,

     "Maharashtra":[

            "Ahmednagar",

            "Akola",

            "Amravati",

            "Aurangabad",

            "Beed",

            "Bhandara",

            "Buldhana",

            "Chandrapur",

            "Dhule",

            "Gadchiroli",

            "Gondia",

            "Hingoli",

            "Jalgaon",

            "Jalna",

            "Kolhapur",

            "Latur",

            "Mumbai City",

            "Mumbai Suburban",

            "Nagpur",

            "Nanded",

            "Nandurbar",

            "Nashik",

            "Osmanabad",

            "Palghar",

            "Parbhani",

            "Pune",

            "Raigad",

            "Ratnagiri",

            "Sangli",

            "Satara",

            "Sindhudurg",

            "Solapur",

            "Thane",

            "Wardha",

            "Washim",

            "Yavatmal"

         ]

      ,

      "Manipur":[

            "Bishnupur",

            "Chandel",

            "Churachandpur",

            "Imphal East",

            "Imphal West",

            "Jiribam",

            "Kakching",

            "Kamjong",

            "Kangpokpi",

            "Noney",

            "Pherzawl",

            "Senapati",

            "Tamenglong",

            "Tengnoupal",

            "Thoubal",

            "Ukhrul"

         ]

      ,

      "Meghalaya":[

            "East Garo Hills",

            "East Jaintia Hills",

            "East Khasi Hills",

            "North Garo Hills",

            "Ri Bhoi",

            "South Garo Hills",

            "South West Garo Hills ",

            "South West Khasi Hills",

            "West Garo Hills",

            "West Jaintia Hills",

            "West Khasi Hills"

         ]

      ,

     "Mizoram":[

            "Aizawl",

            "Champhai",

            "Kolasib",

            "Lawngtlai",

            "Lunglei",

            "Mamit",

            "Saiha",

            "Serchhip"

         ]

    ,

      "Nagaland":[

            "Dimapur",

            "Kiphire",

            "Kohima",

            "Longleng",

            "Mokokchung",

            "Mon",

            "Peren",

            "Phek",

            "Tuensang",

            "Wokha",

            "Zunheboto"

         ],

         "Odisha":[

            "Angul",

            "Balangir",

            "Balasore",

            "Bargarh",

            "Bhadrak",

            "Boudh",

            "Cuttack",

            "Deogarh",

            "Dhenkanal",

            "Gajapati",

            "Ganjam",

            "Jagatsinghapur",

            "Jajpur",

            "Jharsuguda",

            "Kalahandi",

            "Kandhamal",

            "Kendrapara",

            "Kendujhar (Keonjhar)",

            "Khordha",

            "Koraput",

            "Malkangiri",

            "Mayurbhanj",

            "Nabarangpur",

            "Nayagarh",

            "Nuapada",

            "Puri",

            "Rayagada",

            "Sambalpur",

            "Sonepur",

            "Sundargarh"

         ],

      "Puducherry (UT)":[

            "Karaikal",

            "Mahe",

            "Pondicherry",

            "Yanam"

         ]

      ,

     "Punjab":[

            "Amritsar",

            "Barnala",

            "Bathinda",

            "Faridkot",

            "Fatehgarh Sahib",

            "Fazilka",

            "Ferozepur",

            "Gurdaspur",

            "Hoshiarpur",

            "Jalandhar",

            "Kapurthala",

            "Ludhiana",

            "Mansa",

            "Moga",

            "Muktsar",

            "Nawanshahr (Shahid Bhagat Singh Nagar)",

            "Pathankot",

            "Patiala",

            "Rupnagar",

            "Sahibzada Ajit Singh Nagar (Mohali)",

            "Sangrur",

            "Tarn Taran"

         ]

      ,

      "Rajasthan":[

            "Ajmer",

            "Alwar",

            "Banswara",

            "Baran",

            "Barmer",

            "Bharatpur",

            "Bhilwara",

            "Bikaner",

            "Bundi",

            "Chittorgarh",

            "Churu",

            "Dausa",

            "Dholpur",

            "Dungarpur",

            "Hanumangarh",

            "Jaipur",

            "Jaisalmer",

            "Jalore",

            "Jhalawar",

            "Jhunjhunu",

            "Jodhpur",

            "Karauli",

            "Kota",

            "Nagaur",

            "Pali",

            "Pratapgarh",

            "Rajsamand",

            "Sawai Madhopur",

            "Sikar",

            "Sirohi",

            "Sri Ganganagar",

            "Tonk",

            "Udaipur"

         ]

      ,

     "Sikkim":[

            "East Sikkim",

            "North Sikkim",

            "South Sikkim",

            "West Sikkim"

         ]

      ,

      "Tamil Nadu":[

            "Ariyalur",

            "Chennai",

            "Coimbatore",

            "Cuddalore",

            "Dharmapuri",

            "Dindigul",

            "Erode",

            "Kanchipuram",

            "Kanyakumari",

            "Karur",

            "Krishnagiri",

            "Madurai",

            "Nagapattinam",

            "Namakkal",

            "Nilgiris",

            "Perambalur",

            "Pudukkottai",

            "Ramanathapuram",

            "Salem",

            "Sivaganga",

            "Thanjavur",

            "Theni",

            "Thoothukudi (Tuticorin)",

            "Tiruchirappalli",

            "Tirunelveli",

            "Tiruppur",

            "Tiruvallur",

            "Tiruvannamalai",

            "Tiruvarur",

            "Vellore",

            "Viluppuram",

            "Virudhunagar"

         ]

      ,

      "Telangana":[

            "Adilabad",

            "Bhadradri Kothagudem",

            "Hyderabad",

            "Jagtial",

            "Jangaon",

            "Jayashankar Bhoopalpally",

            "Jogulamba Gadwal",

            "Kamareddy",

            "Karimnagar",

            "Khammam",

            "Komaram Bheem Asifabad",

            "Mahabubabad",

            "Mahabubnagar",

            "Mancherial",

            "Medak",

            "Medchal",

            "Nagarkurnool",

            "Nalgonda",

            "Nirmal",

            "Nizamabad",

            "Peddapalli",

            "Rajanna Sircilla",

            "Rangareddy",

            "Sangareddy",

            "Siddipet",

            "Suryapet",

            "Vikarabad",

            "Wanaparthy",

            "Warangal (Rural)",

            "Warangal (Urban)",

            "Yadadri Bhuvanagiri"

         ]

      ,

      "Tripura"

         :[

            "Dhalai",

            "Gomati",

            "Khowai",

            "North Tripura",

            "Sepahijala",

            "South Tripura",

            "Unakoti",

            "West Tripura"

         ]

      ,

      "Uttarakhand":[

            "Almora",

            "Bageshwar",

            "Chamoli",

            "Champawat",

            "Dehradun",

            "Haridwar",

            "Nainital",

            "Pauri Garhwal",

            "Pithoragarh",

            "Rudraprayag",

            "Tehri Garhwal",

            "Udham Singh Nagar",

            "Uttarkashi"

         ]

      ,

    "Uttar Pradesh":[

            "Agra",

            "Aligarh",

            "Allahabad",

            "Ambedkar Nagar",

            "Amethi (Chatrapati Sahuji Mahraj Nagar)",

            "Amroha (J.P. Nagar)",

            "Auraiya",

            "Azamgarh",

            "Baghpat",

            "Bahraich",

            "Ballia",

            "Balrampur",

            "Banda",

            "Barabanki",

            "Bareilly",

            "Basti",

            "Bhadohi",

            "Bijnor",

            "Budaun",

            "Bulandshahr",

            "Chandauli",

            "Chitrakoot",

            "Deoria",

            "Etah",

            "Etawah",

            "Faizabad",

            "Farrukhabad",

            "Fatehpur",

            "Firozabad",

            "Gautam Buddha Nagar",

            "Ghaziabad",

            "Ghazipur",

            "Gonda",

            "Gorakhpur",

            "Hamirpur",

            "Hapur (Panchsheel Nagar)",

            "Hardoi",

            "Hathras",

            "Jalaun",

            "Jaunpur",

            "Jhansi",

            "Kannauj",

            "Kanpur Dehat",

            "Kanpur Nagar",

            "Kanshiram Nagar (Kasganj)",

            "Kaushambi",

            "Kushinagar (Padrauna)",

            "Lakhimpur - Kheri",

            "Lalitpur",

            "Lucknow",

            "Maharajganj",

            "Mahoba",

            "Mainpuri",

            "Mathura",

            "Mau",

            "Meerut",

            "Mirzapur",

            "Moradabad",

            "Muzaffarnagar",

            "Pilibhit",

            "Pratapgarh",

            "RaeBareli",

            "Rampur",

            "Saharanpur",

            "Sambhal (Bhim Nagar)",

            "Sant Kabir Nagar",

            "Shahjahanpur",

            "Shamali (Prabuddh Nagar)",

            "Shravasti",

            "Siddharth Nagar",

            "Sitapur",

            "Sonbhadra",

            "Sultanpur",

            "Unnao",

            "Varanasi"

         ]

      ,

      "West Bengal":[

            "Alipurduar","Bankura","Birbhum","Burdwan (Bardhaman)","Cooch Behar","Dakshin Dinajpur (South Dinajpur)","Darjeeling","Hooghly","Howrah","Jalpaiguri","Kalimpong","Kolkata","Malda","Murshidabad",

            "Nadia",

            "North 24 Parganas",

            "Paschim Medinipur (West Medinipur)",

            "Purba Medinipur (East Medinipur)",

            "Purulia",

            "South 24 Parganas",

            "Uttar Dinajpur (North Dinajpur)"

         ]

}