

```
recom.py
```

```
import streamlit as st
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
import plotly.express as px
from PIL import Image
import requests
import time

# Page config
st.set_page_config(page_title="🌱 Smart Crop Recommendation System", layout="wide",
page_icon="🌿")

# Custom CSS with larger fonts and background
st.markdown("""
<style>
/* Main Theme Colors */
:root {
    --primary-green: #2E7D32;
    --secondary-green: #4CAF50;
    --light-green: #A5D6A7;
    --dark-green: #1B5E20;
    --accent-gold: #FFB300;
}
/* Background Image */
.stApp {
    background: linear-gradient(rgba(255, 255, 255, 0.95), rgba(255, 255, 255, 0.98)),
        url('https://images.unsplash.com/photo-1500382017468-9049fed747ef?ixlib=rb-4.0.3&auto=format&fit=crop&w=2000&q=80');
```

```
background-size: cover;  
background-attachment: fixed;  
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
min-height: 100vh;  
}  
  
/* Header Styling with LARGER FONTS */  
.main-header {  
background: linear-gradient(135deg, rgba(46, 125, 50, 0.95), rgba(76, 175, 80, 0.95));  
color: white;  
padding: 2.5rem 2rem;  
border-radius: 20px;  
margin-bottom: 2rem;  
text-align: center;  
box-shadow: 0 10px 25px rgba(46, 125, 50, 0.3);  
backdrop-filter: blur(10px);  
border: 2px solid rgba(255, 255, 255, 0.2);  
}  
.main-title {  
font-size: 3.5rem !important;  
font-weight: 800 !important;  
margin-bottom: 0.5rem !important;  
text-shadow: 2px 2px 6px rgba(0,0,0,0.3);  
}  
.main-subtitle {  
font-size: 1.6rem !important;  
opacity: 0.95;  
max-width: 800px;  
margin: 0 auto;  
line-height: 1.5;  
}
```

```
/* Card Styling */

.metric-card {

background: rgba(255, 255, 255, 0.98);
border-radius: 20px;
padding: 2rem;
margin: 1rem 0;
box-shadow: 0 8px 20px rgba(0,0,0,0.1);
border: 2px solid rgba(46, 125, 50, 0.2);
transition: all 0.3s ease;

}

.metric-card:hover {

transform: translateY(-5px);
box-shadow: 0 12px 25px rgba(46, 125, 50, 0.2);

}

/* Input Container */

.input-container {

background: rgba(255, 255, 255, 0.98);
border-radius: 20px;
padding: 2.5rem;
margin: 2rem 0;
box-shadow: 0 10px 25px rgba(0,0,0,0.1);
border: 2px solid rgba(46, 125, 50, 0.2);

}

.input-title {

font-size: 1.8rem !important;
color: var(--primary-green) !important;
margin-bottom: 1.5rem !important;
font-weight: 700 !important;
```

```
text-align: center;  
}  
  
/* Button Styling */  
.stButton > button {  
background: linear-gradient(135deg, var(--primary-green), var(--secondary-green));  
color: white;  
border: none;  
padding: 1rem 2.5rem;  
border-radius: 40px;  
font-weight: 700 !important;  
font-size: 1.3rem !important;  
transition: all 0.3s ease;  
box-shadow: 0 5px 15px rgba(46, 125, 50, 0.3);  
}  
.stButton > button:hover {  
transform: translateY(-3px);  
box-shadow: 0 8px 20px rgba(46, 125, 50, 0.4);  
}  
  
/* Result Cards */  
.result-card {  
background: linear-gradient(135deg, rgba(232, 245, 232, 0.98), rgba(200, 230, 201, 0.98));  
border-radius: 20px;  
padding: 2rem;  
margin: 1.5rem 0;  
box-shadow: 0 10px 25px rgba(46, 125, 50, 0.15);  
border-left: 6px solid var(--primary-green);  
}  
  
/* Crop Badge */
```

```
.crop-badge {  
    display: inline-block;  
    background: linear-gradient(135deg, var(--accent-gold), #FFCA28);  
    color: var(--dark-green);  
    padding: 1rem 2rem;  
    border-radius: 40px;  
    font-weight: 800 !important;  
    font-size: 1.5rem !important;  
    box-shadow: 0 5px 15px rgba(255, 179, 0, 0.3);  
    margin: 0.5rem;  
    border: 2px solid white;  
    text-transform: uppercase;  
}  
  
/* Labels */
```

```
label {  
    font-size: 1.1rem !important;  
    font-weight: 600 !important;  
    color: var(--dark-green) !important;  
    margin-bottom: 0.5rem !important;  
}
```

```
/* Footer */  
.footer {  
    background: linear-gradient(135deg, rgba(46, 125, 50, 0.95), rgba(27, 94, 32, 0.95));  
    color: white;  
    padding: 2rem;  
    border-radius: 20px;  
    text-align: center;  
    margin-top: 2rem;  
    box-shadow: 0 8px 20px rgba(46, 125, 50, 0.2);
```

```
}

.footer h3 {
    font-size: 1.8rem !important;
    margin-bottom: 0.8rem !important;
}

.footer p {
    font-size: 1.1rem !important;
    opacity: 0.9;
    margin-bottom: 0.3rem !important;
}

</style>
""", unsafe_allow_html=True)
```

```
# Create simplified dataset with only required parameters
np.random.seed(42)
n_samples = 300

data = {
    'N': np.random.uniform(0, 200, n_samples).round(1),
    'P': np.random.uniform(0, 150, n_samples).round(1),
    'K': np.random.uniform(0, 250, n_samples).round(1),
    'rainfall': np.random.uniform(100, 400, n_samples).round(1),
    'humidity': np.random.uniform(20, 95, n_samples).round(1),
    'temperature': np.random.uniform(10, 40, n_samples).round(1), # Added temperature
}
```

```
df = pd.DataFrame(data)

# Assign crops based on simplified conditions
```

```
def assign_crop_simple(row):
    n, p, k = row['N'], row['P'], row['K']
    rainfall = row['rainfall']
    humidity = row['humidity']
    temperature = row['temperature']

    # Added temperature

    # Rice - needs high rainfall, humidity, and warm temperature
    if rainfall > 300 and humidity > 80 and n > 100 and temperature > 20:
        return 'rice'

    # Wheat - moderate conditions, cooler temperature
    elif 200 < rainfall < 300 and 50 < humidity < 80 and p > 40 and 15 < temperature < 25:
        return 'wheat'

    # Maize - good rainfall, potassium, and warm temperature
    elif rainfall > 250 and humidity > 60 and k > 100 and temperature > 18:
        return 'maize'

    # Cotton - lower humidity needs, warm temperature
    elif rainfall < 250 and humidity < 60 and n > 80 and temperature > 22:
        return 'cotton'

    # Sugarcane - high rainfall, warm temperature
    elif rainfall > 350 and humidity > 75 and temperature > 20:
        return 'sugarcane'

    # Fruits based on conditions - warm temperatures
    elif humidity > 70 and rainfall > 200 and temperature > 18:
        return np.random.choice(['banana', 'papaya', 'pineapple', 'mango', 'coconut'])
```

```
# Vegetables - moderate conditions, moderate temperature
elif 150 < rainfall < 300 and 50 < humidity < 80 and 15 < temperature < 30:
    return np.random.choice(['tomato', 'potato', 'cucumber', 'onion'])

# Legumes - lower nitrogen needs, moderate temperature
elif n < 80 and p > 30 and 15 < temperature < 30:
    return np.random.choice(['mungbean', 'chickpea', 'kidneybeans', 'lentil'])

# Coffee - specific humidity, rainfall, and temperature
elif 150 < rainfall < 300 and 70 < humidity < 90 and 18 < temperature < 25:
    return 'coffee'

# Jute - high rainfall, warm temperature
elif rainfall > 300 and temperature > 22:
    return 'jute'

# Watermelon - hot and humid
elif humidity > 70 and rainfall > 200 and temperature > 25:
    return 'watermelon'

else:
    # Default based on primary nutrients and temperature
    if temperature > 25:
        if n > p and n > k:
            return 'cotton'
        elif p > n and p > k:
            return 'potato'
        else:
            return 'banana'
    else:
        if n > p and n > k:
```

```

    return 'wheat'

elif p > n and p > k:
    return 'potato'

else:
    return 'mungbean'

df['label'] = df.apply(assign_crop_simple, axis=1)

# Train model with only required features

feature_cols = ['N', 'P', 'K', 'rainfall', 'humidity', 'temperature'] # Added temperature

X = df[feature_cols]
y = df['label']

model = RandomForestClassifier(
    n_estimators=200,
    random_state=42,
    max_depth=10,
    min_samples_split=5,
    min_samples_leaf=2
)
model.fit(X, y)

# Header
st.markdown("""
<div class="main-header">
    <h1 class="main-title">🌿 CROP RECOMMENDATION SYSTEM</h1>
    <p class="main-subtitle">AI-powered recommendations based on Soil Nutrients, Climate & Weather Conditions</p>
</div>
""", unsafe_allow_html=True)

```

```

# Input section

st.markdown('<div class="input-container">', unsafe_allow_html=True)

st.markdown('<h1 class="input-title">  ENTER PARAMETERS</h1>', unsafe_allow_html=True)

# Create columns for input

col1, col2 = st.columns(2)

with col1:

    st.markdown('###  SOIL NUTRIENTS')

    N = st.slider("**NITROGEN (N) in kg/ha**", 0.0, 200.0, 80.0, 1.0,
                  help="Essential for leaf growth and plant development")

    P = st.slider("**PHOSPHORUS (P) in kg/ha**", 0.0, 150.0, 45.0, 1.0,
                  help="Important for root development and flowering")

    K = st.slider("**POTASSIUM (K) in kg/ha**", 0.0, 250.0, 120.0, 1.0,
                  help="Essential for fruit quality and disease resistance")

with col2:

    st.markdown('###  CLIMATE FACTORS')

    rainfall = st.slider("**ANNUAL RAINFALL in mm**", 100.0, 400.0, 250.0, 10.0,
                          help="Total annual rainfall in millimeters")

    humidity = st.slider("**AVERAGE HUMIDITY in %**", 20.0, 95.0, 65.0, 1.0,
                          help="Average relative humidity percentage")

    temperature = st.slider("**AVERAGE TEMPERATURE in °C**", 10.0, 40.0, 25.0, 0.5, # Added temperature
                           slider
                           help="Average annual temperature in Celsius")

st.markdown('</div>', unsafe_allow_html=True)

# Center the button

col_btn1, col_btn2, col_btn3 = st.columns([1, 2, 1])

```

```
with col_btn2:
    analyze_clicked = st.button("🚀 GET CROP RECOMMENDATIONS", type="primary",
use_container_width=True)

if analyze_clicked:
    # Prepare input data
    input_data = np.array([[

        N, P, K, rainfall, humidity, temperature # Added temperature
    ]])

    # Get predictions
    prediction = model.predict(input_data)[0]
    probabilities = model.predict_proba(input_data)[0]

    # Top recommendations
    prob_df = pd.DataFrame({
        'Crop': model.classes_,
        'Probability': probabilities
    }).sort_values('Probability', ascending=False).head(8)

    # Display results
    st.markdown('<div class="result-card">', unsafe_allow_html=True)

    # Top recommendation - centered without image
    st.markdown(f"""

<div style='text-align: center;'>

    <h1 style='color: var(--primary-green); margin-bottom: 1rem; font-size: 2.2rem;'>🚀 BEST CROP FOR
YOUR SOIL</h1>

    <div class="crop-badge">{prediction.title()}</div>
    <p style='font-size: 1.4rem; color: var(--dark-green); margin-top: 1rem;'>

        Confidence: <strong style='font-size: 1.8rem;'>{prob_df[prob_df["Crop"] ==
prediction]["Probability"].iloc[0]:.1%}</strong>
    </p>
</div>

```

```

</p>
</div>
""", unsafe_allow_html=True)

st.markdown('</div>', unsafe_allow_html=True)

# Footer
st.markdown("""
<div class="footer">
    <h3>  Smart Agriculture System</h3>
    <p>25+ Crops • 6 Key Parameters • AI-Powered Recommendations</p> <!-- Updated to 6 parameters -->
    <p style='font-size: 0.9rem; margin-top: 1rem; opacity: 0.8;'>
         Based on N, P, K nutrients, rainfall, humidity, and temperature data <!-- Added temperature -->
    </p>
</div>
""", unsafe_allow_html=True)

```

CROP DISEASE DETECTION SYSTEM -

Final Year Academic Project

```

import streamlit as st
from PIL import Image
import numpy as np
import random
import time
import pytsxs3

# =====

```

```
# PAGE CONFIGURATION

# =====

st.set_page_config(
    page_title="Crop Disease Detection System",
    page_icon="🌿",
    layout="wide",
    initial_sidebar_state="expanded"
)

# =====

# TEXT-TO-SPEECH FUNCTION (SIMPLIFIED)
# =====

def speak_text(text):
    """Convert text to speech and play it"""

    try:
        # Try to import pyttsx3
        import pyttsx3

        # Initialize the engine
        engine = pyttsx3.init()

        # Set properties
        engine.setProperty('rate', 150) # Speed
        engine.setProperty('volume', 0.9) # Volume

        # Get available voices
        voices = engine.getProperty('voices')

        # Try to use a female voice if available
        for voice in voices:
            if 'female' in voice.name.lower():
                engine.setProperty('voice', voice.name)
                engine.say(text)
                engine.runAndWait()
            else:
                print(f"Voice '{voice.name}' is not female")
    except ImportError:
        print("pyttsx3 module not found. Please install it using 'pip install pyttsx3'")
```

```
    engine.setProperty('voice', voice.id)

    break

# Speak the text

engine.say(text)

engine.runAndWait()

return True

except Exception as e:

    # If pyttsx3 fails, show a message

    st.warning(f"Voice feature unavailable. Please install pyttsx3 with: pip install pyttsx3")

    return False

# =====

# CUSTOM CSS STYLING

# =====

st.markdown(""""

<style>

/* Main styles */

.stApp {

background: linear-gradient(135deg, #f9fff9 0%, #e8f5e9 100%);

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

}

/* Header styling */

.main-header {

background: linear-gradient(135deg, #2E7D32, #1B5E20);

color: white;

padding: 2.5rem;

border-radius: 15px;

margin-bottom: 2rem;
```

```
text-align: center;  
box-shadow: 0 4px 12px rgba(46, 125, 50, 0.2);  
}
```

```
.main-title {  
    font-size: 3rem;  
    font-weight: 700;  
    margin-bottom: 0.5rem;  
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.2);  
}
```

```
.main-subtitle {  
    font-size: 1.3rem;  
    opacity: 0.95;  
    margin: 0 auto;  
    max-width: 800px;  
}
```

```
/* Card styling */  
.card {  
    background: white;  
    border-radius: 15px;  
    padding: 2rem;  
    margin: 1.5rem 0;  
    box-shadow: 0 4px 12px rgba(0,0,0,0.1);  
    border: 1px solid rgba(46, 125, 50, 0.1);  
}
```

```
.card-title {  
    color: #2E7D32;  
    font-size: 1.8rem;
```

```
font-weight: 600;  
margin-bottom: 1rem;  
border-bottom: 2px solid #A5D6A7;  
padding-bottom: 0.5rem;  
}
```

```
/* Result card */  
.result-card {  
background: white;  
border-radius: 15px;  
padding: 2.5rem;  
margin: 2rem auto;  
max-width: 900px;  
box-shadow: 0 10px 30px rgba(0,0,0,0.1);  
text-align: center;  
animation: fadeIn 0.5s ease-out;  
}
```

```
@keyframes fadeIn {  
from { opacity: 0; transform: translateY(20px); }  
to { opacity: 1; transform: translateY(0); }  
}
```

```
/* Disease name */  
.disease-name {  
color: #2E7D32;  
font-size: 2.5rem;  
font-weight: 700;  
margin: 1rem 0;  
text-transform: uppercase;  
letter-spacing: 1px;
```

}

```
/* Confidence badge */  
.confidence-badge {  
    display: inline-block;  
    background: linear-gradient(135deg, #4CAF50, #2E7D32);  
    color: white;  
    padding: 0.7rem 1.8rem;  
    border-radius: 25px;  
    font-weight: 600;  
    font-size: 1.2rem;  
    margin: 1rem 0;  
    box-shadow: 0 4px 8px rgba(76, 175, 80, 0.3);  
}
```

```
/* Severity badges */
```

```
.severity-high {  
    background: #FFEBEE;  
    color: #C62828;  
    padding: 0.5rem 1rem;  
    border-radius: 8px;  
    font-weight: 600;  
    display: inline-block;  
}
```

```
.severity-moderate {
```

```
    background: #FFF3EO;  
    color: #EF6C00;  
    padding: 0.5rem 1rem;  
    border-radius: 8px;  
    font-weight: 600;
```

```
display: inline-block;  
}  
  
.severity-low {  
background: #E8F5E9;  
color: #1B5E20;  
padding: 0.5rem 1rem;  
border-radius: 8px;  
font-weight: 600;  
display: inline-block;  
}  
  
/* Treatment cards */  
.treatment-card {  
background: #F9FFF9;  
border-radius: 10px;  
padding: 1.5rem;  
margin: 1rem 0;  
border-left: 4px solid #4CAF50;  
}  
  
/* Button styling */  
.stButton > button {  
background: linear-gradient(135deg, #2E7D32, #43A047);  
color: white;  
border: none;  
padding: 1rem 2rem;  
border-radius: 10px;  
font-weight: 600;  
font-size: 1.1rem;  
transition: all 0.3s ease;
```

```
}
```

```
.stButton > button:hover {  
    transform: translateY(-2px);  
    box-shadow: 0 5px 15px rgba(46, 125, 50, 0.3);  
}
```

```
/* Upload container */
```

```
.upload-container {  
    background: linear-gradient(135deg, #E8F5E9, #F1F8E9);  
    border: 3px dashed #2E7D32;  
    border-radius: 15px;  
    padding: 3rem;  
    text-align: center;  
    margin: 2rem 0;  
    cursor: pointer;  
    transition: all 0.3s ease;  
}
```

```
.upload-container:hover {
```

```
    background: linear-gradient(135deg, #F1F8E9, #E8F5E9);  
    border-color: #43A047;  
}
```

```
/* Sidebar styling */
```

```
.sidebar-header {  
    background: linear-gradient(135deg, #2E7D32, #1B5E20);  
    color: white;  
    padding: 1.5rem;  
    border-radius: 10px;  
    margin-bottom: 1.5rem;
```

```
text-align: center;  
}  
  
.sidebar-title {  
    font-size: 1.5rem;  
    font-weight: 600;  
    margin-bottom: 0.5rem;  
}  
</style>  
"""", unsafe_allow_html=True)  
  
# =====  
# ENHANCED DISEASE DATABASE WITH MULTIPLE SOLUTIONS  
# =====  
  
DISEASE_DATABASE = {  
    "Early Blight": {  
        "scientific_name": "Alternaria solani",  
        "description": "A fungal disease characterized by dark concentric rings on leaves, affecting tomatoes and potatoes.",  
        "affected_crops": ["Tomato", "Potato", "Eggplant", "Pepper"],  
        "symptoms": [  
            "Small dark spots on lower leaves",  
            "Concentric rings on lesions",  
            "Yellowing around spots",  
            "Leaf drop in severe cases",  
            "Lesions on stems and fruits"  
        ],  
        "severity": "Moderate-High",  
        "severity_score": 65,  
        "solutions": {  
            "chemical": [  
                "Fungicides (e.g., Propiconazole, Mancozeb)",  
                "Organic options (e.g., Neem oil, Garrett Juice)"  
            ]  
        }  
    }  
}
```

"Apply Chlorothalonil (Bravo) every 7-10 days",
"Use Mancozeb (Dithane) as preventative spray",
"Apply Copper-based fungicides (Kocide)"
,
"organic": [
"Spray neem oil solution every 5-7 days",
"Apply baking soda solution (1 tbsp per gallon of water)",
"Use garlic extract spray"
,
"cultural": [
"Practice 3-year crop rotation",
"Remove and destroy infected plant debris",
"Ensure proper spacing (60-90 cm between plants)",
"Water at soil level, avoid overhead irrigation",
"Use mulch to prevent soil splash"
,
"biological": [
"Apply Trichoderma harzianum bio-fungicide",
"Use Bacillus subtilis products",
"Introduce beneficial microbes to soil"
,
}
,"prevention": [
"Use disease-resistant varieties",
"Plant in well-drained soil",
"Avoid working with wet plants",
"Sanitize garden tools regularly",
"Monitor plants weekly for early signs"
,
"emoji": "🍅"
,

"Late Blight": {
 "scientific_name": "Phytophthora infestans",
 "description": "Devastating disease that spreads rapidly in cool, wet conditions, famous for causing the Irish Potato Famine.",
 "affected_crops": ["Potato", "Tomato"],
 "symptoms": [
 "Water-soaked lesions on leaves",
 "White mold growth on underside of leaves",
 "Rapid spreading in wet conditions",
 "Dark lesions on stems",
 "Rotting of tubers and fruits"
],
 "severity": "Very High",
 "severity_score": 90,
 "solutions": {
 "chemical": [
 "Apply Metalaxyl (Ridomil) as soil drench",
 "Use Chlorothalonil (Bravo) every 5-7 days during wet weather",
 "Apply Famoxadone + Cymoxanil (Tanos)"
],
 "organic": [
 "Apply copper fungicides every 7-10 days",
 "Use hydrogen peroxide spray (3% solution)",
 "Apply compost tea to boost plant immunity"
],
 "cultural": [
 "Destroy infected plants immediately",
 "Use certified disease-free seeds",
 "Avoid overhead irrigation",
 "Improve air circulation",
 "Harvest before rainy season"
]
 }
}

],
"biological": [
 "Apply Streptomyces lydicus products",
 "Use Bacillus amyloliquefaciens",
 "Apply chitosan-based products"
]
},
"prevention": [
 "Plant resistant varieties",
 "Monitor weather forecasts",
 "Use protective fungicides before rainy periods",
 "Avoid planting near infected fields",
 "Practice strict sanitation"
],
"emoji": "🥔",
},
"Powdery Mildew": {
 "scientific_name": "Erysiphe spp.",
 "description": "White powdery fungal growth on leaves and stems, common in dry conditions with high humidity.",
 "affected_crops": ["Cucumber", "Squash", "Grapes", "Wheat", "Mango", "Rose"],
 "symptoms": [
 "White powdery spots on leaves",
 "Leaves turning yellow then brown",
 "Stunted plant growth",
 "Distorted leaves",
 "Reduced fruit production"
],
 "severity": "Moderate",
 "severity_score": 50,
 "solutions": {

"chemical": [
 "Apply Sulfur dust or spray",
 "Use Myclobutanil (Systhane) every 10-14 days",
 "Apply Triflumizole (Procure)"
],
"organic": [
 "Spray milk solution (1 part milk to 9 parts water)",
 "Apply baking soda spray (1 tsp per liter)",
 "Use neem oil every 5-7 days",
 "Apply potassium bicarbonate solution"
],
"cultural": [
 "Ensure good air circulation",
 "Avoid overhead watering",
 "Plant resistant varieties",
 "Remove infected leaves promptly",
 "Space plants properly"
],
"biological": [
 "Apply Ampelomyces quisqualis",
 "Use Bacillus pumilus products",
 "Apply horticultural oil sprays"
]
},
"prevention": [
 "Water early in the day",
 "Maintain proper plant spacing",
 "Avoid excess nitrogen fertilizer",
 "Keep garden clean of debris",
 "Monitor humidity levels"
],

"emoji": "🌿",

},

"Bacterial Leaf Spot": {

 "scientific_name": "Xanthomonas spp.",

 "description": "Bacterial disease causing angular water-soaked spots that turn brown with yellow halos.",

 "affected_crops": ["Tomato", "Pepper", "Cabbage", "Rice", "Mango", "Citrus"],

 "symptoms": [

 "Small water-soaked spots",

 "Spots turning brown or black",

 "Yellow halos around spots",

 "Leaf drop in severe cases",

 "Fruit lesions and spots"

],

 "severity": "Moderate",

 "severity_score": 55,

 "solutions": {

 "chemical": [

 "Apply Copper-based bactericides (Kocide 3000)",

 "Use Streptomycin for severe cases",

 "Apply Oxytetracycline products"

],

 "organic": [

 "Apply copper soap sprays",

 "Use hydrogen peroxide (3%) solution",

 "Apply garlic-chili spray",

 "Use vinegar solution (1:3 vinegar:water)"

],

 "cultural": [

 "Use disease-free seeds and transplants",

 "Avoid working with wet plants",

```
"Practice 2-3 year crop rotation",
"Remove weed hosts",
"Disinfect tools regularly"
],
"biological": [
    "Apply Bacillus subtilis products",
    "Use Pseudomonas fluorescens",
    "Apply beneficial microbes"
]
},
"prevention": [
    "Purchase certified disease-free seeds",
    "Avoid overhead irrigation",
    "Remove infected plants immediately",
    "Control insect vectors",
    "Improve soil drainage"
],
"emoji": "⭐"
}
```

```
# =====
# INITIALIZE SESSION STATE
# =====

if 'current_page' not in st.session_state:
    st.session_state.current_page = 'Detection' # Default to detection page

if 'last_analysis' not in st.session_state:
    st.session_state.last_analysis = None

# =====
# SIDEBAR - SIMPLE NAVIGATION
```

```
# =====
```

```
with st.sidebar:
```

```
    st.markdown(""""  
        <div class="sidebar-header">  
            <div class="sidebar-title">  Navigation</div>  
            <div style="font-size: 0.9rem; opacity: 0.9;">  
                Crop Disease Detection System  
            </div>  
        </div>  
    """", unsafe_allow_html=True)
```

```
st.markdown("---")
```

```
# Voice Assistant Section
```

```
st.markdown("###  Voice Assistant")
```

```
if st.button("  Test Voice Assistant", use_container_width=True):
```

```
    test_message = "Welcome to the Crop Disease Detection System. I am your voice assistant. I will help you understand the disease detection results."
```

```
    if speak_text(test_message):
```

```
        st.success("Voice assistant is working!")
```

```
    else:
```

```
        st.warning("Please install pyttsx3: pip install pyttsx3")
```

```
st.markdown("---")
```

```
# Quick Tips
```

```
st.markdown("###  Quick Tips")
```

```
st.info(""""
```

```
1. Upload clear leaf images
```

```
2. Select correct crop type
```

```
3. Review all solution options  
4. Use voice assistant for explanations  
""")
```

```
st.markdown("---")  
  
# Disease Statistics  
st.markdown("### 🌱 Disease Database")  
st.metric("Total Diseases", len(DISEASE_DATABASE))
```

```
# List available diseases  
st.markdown("**Available Diseases:**")  
for disease in DISEASE_DATABASE.keys():  
    st.markdown(f"- {DISEASE_DATABASE[disease]['emoji']} {disease}")
```

```
# ======  
# MAIN HEADER  
# ======  
st.markdown("""  


<h1 class="main-title">🌱 AI-Powered Crop Disease Detection</h1>  
    <p class="main-subtitle">  
        Upload leaf images to detect diseases and get multiple treatment solutions instantly  
    </p>  
</div>  
""", unsafe_allow_html=True)


```

```
# ======  
# MAIN CONTENT - DISEASE DETECTION ONLY  
# ======  
st.markdown("""
```

```
<div class="card">

  <h2 class="card-title">🔍 Upload Leaf Image for Disease Detection</h2>

  <p style="font-size: 1.1rem; color: #555; line-height: 1.6;">

    Our advanced AI system analyzes plant leaf images to detect diseases and provides
    multiple treatment solutions including chemical, organic, cultural, and biological methods.

  </p>

</div>

"""", unsafe_allow_html=True)
```

```
# Upload Section
st.markdown(""""

<div class="upload-container">

  <div style="font-size: 4rem;">📤 </div>

  <h3 style="color: #2E7D32; margin-bottom: 1rem;">

    Drag & Drop or Click to Upload Image

  </h3>

  <p style="color: #666; margin-bottom: 1.5rem;">

    Supported formats: JPG, JPEG, PNG | Max size: 10MB

  </p>

</div>

"""", unsafe_allow_html=True)
```

```
# File Uploader
uploaded_file = st.file_uploader(
  "Choose a leaf image...", 
  type=["jpg", "jpeg", "png"],
  label_visibility="collapsed"
)
```

```
# Process Uploaded Image
if uploaded_file is not None:
```

```
try:  
    # Display uploaded image  
    image = Image.open(uploaded_file)  
  
    st.markdown("### 📸 Uploaded Image Preview")  
    col1, col2, col3 = st.columns([1, 2, 1])  
    with col2:  
        st.image(image, caption="Leaf Image for Analysis", use_column_width=True)  
  
    # Crop Selection  
    st.markdown("### 🌾 Select Crop Type")  
    crop_type = st.selectbox(  
        "Choose the crop type for better accuracy:",  
        ["Auto Detect", "Tomato", "Potato", "Rice", "Banana", "Cucumber", "Wheat", "Grapes", "Other"],  
        index=0  
    )  
  
    # Analyze Button  
    st.markdown("<br>", unsafe_allow_html=True)  
    col1, col2, col3 = st.columns([1, 2, 1])  
    with col2:  
        if st.button("🔍 Analyze Image with AI", type="primary", use_container_width=True):  
            with st.spinner("🕒 AI is analyzing the image... Please wait."):br/>                # Show progress  
                progress_bar = st.progress(0)  
                for i in range(100):  
                    time.sleep(0.02)  
                    progress_bar.progress(i + 1)  
  
                # Get random disease prediction (for demo)  
                disease_name = random.choice(list(DISEASE_DATABASE.keys()))
```

```
disease_info = DISEASE_DATABASE[disease_name]
confidence = random.randint(88, 97)

# Store analysis in session state
st.session_state.last_analysis = {
    "disease_name": disease_name,
    "disease_info": disease_info,
    "confidence": confidence
}

# Complete progress
progress_bar.progress(100)
time.sleep(0.5)

# DISPLAY RESULTS
st.markdown('<div class="result-card">', unsafe_allow_html=True)

# Title
st.markdown('<h2 style="color: #2E7D32; margin-bottom: 1.5rem;">✓ Analysis  
Complete!</h2>', unsafe_allow_html=True)

# Emoji
st.markdown(f'<div style="font-size: 4rem; margin: 1rem 0;">{disease_info["emoji"]}</div>',
unsafe_allow_html=True)

# Disease Name
st.markdown(f'<h1 class="disease-name">{disease_name}</h1>', unsafe_allow_html=True)

# Confidence Badge
st.markdown(f'<div class="confidence-badge">{confidence}% Confidence Level</div>',
unsafe_allow_html=True)
```

```
# Severity

if "High" in disease_info["severity"]:
    severity_class = "severity-high"
elif "Moderate" in disease_info["severity"]:
    severity_class = "severity-moderate"
else:
    severity_class = "severity-low"

st.markdown(f'<div style="margin: 1.5rem 0;"><span class="{severity_class}">Severity: {disease_info["severity"]}</span></div>', unsafe_allow_html=True)

st.markdown('</div>', unsafe_allow_html=True)

# Disease Information

st.markdown("### 📄 Disease Information")

col1, col2 = st.columns(2)

with col1:
    st.info(f"""
        **Scientific Name:** {disease_info['scientific_name']}
        **Description:** {disease_info['description']}
        **Affected Crops:** {', '.join(disease_info['affected_crops'])}
    """)

with col2:
    st.info("***Main Symptoms:**")
    for symptom in disease_info['symptoms'][:4]:
        st.write(f"• {symptom}")
```

```
# MULTIPLE SOLUTIONS SECTION
```

```
st.markdown("### 💡 Multiple Treatment Solutions")
```

```
# Chemical Solutions
```

```
with st.expander("🧪 Chemical Solutions", expanded=True):
```

```
    st.success("**Recommended chemical treatments:**")
```

```
    for i, solution in enumerate(disease_info['solutions']['chemical'], 1):
```

```
        st.write(f"{i}. {solution}")
```

```
# Organic Solutions
```

```
with st.expander("🌿 Organic Solutions", expanded=True):
```

```
    st.success("**Natural and organic treatments:**")
```

```
    for i, solution in enumerate(disease_info['solutions']['organic'], 1):
```

```
        st.write(f"{i}. {solution}")
```

```
# Cultural Solutions
```

```
with st.expander("🌱 Cultural Practices", expanded=True):
```

```
    st.success("**Management and cultural practices:**")
```

```
    for i, solution in enumerate(disease_info['solutions']['cultural'], 1):
```

```
        st.write(f"{i}. {solution}")
```

```
# Biological Solutions
```

```
with st.expander("🦠 Biological Control", expanded=True):
```

```
    st.success("**Biological control methods:**")
```

```
    for i, solution in enumerate(disease_info['solutions']['biological'], 1):
```

```
        st.write(f"{i}. {solution}")
```

```
# Prevention Tips
```

```
st.markdown("### 💙 Prevention Tips")
```

```
prevention_cols = st.columns(2)
```

```

for i, tip in enumerate(disease_info['prevention']):
    with prevention_cols[i % 2]:
        st.info(f"✓ {tip}")

except Exception as e:
    st.error(f"Error processing image: {str(e)}")
    st.info("Please upload a valid image file.")

# Show analysis results if available
if st.session_state.last_analysis:
    disease_name = st.session_state.last_analysis["disease_name"]
    disease_info = st.session_state.last_analysis["disease_info"]
    confidence = st.session_state.last_analysis["confidence"]

# VOICE ASSISTANT BUTTONS (AFTER RESULTS ARE DISPLAYED)
st.markdown("---")
st.markdown("### 🎤 Voice Assistant")
col1, col2, col3 = st.columns(3)

with col1:
    if st.button("🎤 Hear Diagnosis", key="hear_diagnosis", use_container_width=True):
        voice_text = f"Disease detected: {disease_name}. Confidence level: {confidence} percent. Severity: {disease_info['severity']}. This disease affects: {' , '.join(disease_info['affected_crops'])}.""
        if speak_text(voice_text):
            st.success("Playing diagnosis...")

with col2:
    if st.button("💊 Hear Treatments", key="hear_treatments", use_container_width=True):
        if disease_info['solutions']['chemical']:
            chemical_treatment = disease_info['solutions']['chemical'][0]
        else:
            pass

```

```
chemical_treatment = "Consult agricultural expert"

if disease_info['solutions']['organic']:
    organic_treatment = disease_info['solutions']['organic'][0]
else:
    organic_treatment = "Use neem oil spray"

voice_text = f"Recommended treatments. Chemical treatment: {chemical_treatment}. Organic option: {organic_treatment}."

if speak_text(voice_text):
    st.success("Playing treatments...")

with col3:
    if st.button(" 🎧 Hear Prevention", key="hear_prevention", use_container_width=True):
        if disease_info['prevention']:
            prevention1 = disease_info['prevention'][0]
            if len(disease_info['prevention']) > 1:
                prevention2 = disease_info['prevention'][1]
                voice_text = f"Prevention tips: {prevention1}. Also remember to: {prevention2}."
            else:
                voice_text = f"Prevention tip: {prevention1}."
        else:
            voice_text = "Practice crop rotation and monitor plants regularly."
    if speak_text(voice_text):
        st.success("Playing prevention tips...")

# Download Report and Analyze Another
st.markdown("---")
col1, col2 = st.columns(2)
```

```
with col1:
```

```
    if st.button("📥 Download Full Report", use_container_width=True):  
        st.success("Report generated! (Demo feature - would create PDF in full implementation)")
```

```
with col2:
```

```
    if st.button("🔄 Analyze Another Image", type="primary", use_container_width=True):  
        st.session_state.last_analysis = None  
        st.rerun()
```

```
else:
```

```
    # Show sample images when no file is uploaded
```

```
    st.markdown("""  
        <div class="card">  
            <h3 style="color: #2E7D32; margin-bottom: 1rem;">📸 Sample Images for Testing</h3>  
            <p style="color: #666; margin-bottom: 1.5rem;">  
                Upload images like these for best results:  
            </p>  
        </div>  
    """", unsafe_allow_html=True)
```

```
# Create columns for sample images using Streamlit's native columns
```

```
col1, col2, col3 = st.columns(3)
```

```
with col1:
```

```
    st.markdown('<div style="text-align: center; font-size: 3rem;">🍅 </div>', unsafe_allow_html=True)  
    st.markdown("**Tomato Leaf**")  
    st.caption("With disease symptoms")
```

```
with col2:
```

```
    st.markdown('<div style="text-align: center; font-size: 3rem;">🥔 </div>', unsafe_allow_html=True)  
    st.markdown("**Potato Leaf**")  
    st.caption("Clear close-up image")
```

with col3:

```
st.markdown('<div style="text-align: center; font-size: 3rem;"> 🌾 </div>', unsafe_allow_html=True)
st.markdown("**Rice Plant**")
st.caption("Showing leaf lesions")
```

Image Guidelines

```
st.markdown("""
<div class="card">
  <h3 style="color: #2E7D32; margin-bottom: 1rem;"> 📄 Image Guidelines</h3>
</div>
""", unsafe_allow_html=True)
```

Guidelines in columns

```
gcol1, gcol2, gcol3 = st.columns(3)
```

with gcol1:

```
st.markdown('<div style="text-align: center; font-size: 2.5rem;"> ☀️ </div>', unsafe_allow_html=True)
st.markdown("**Good Lighting**")
st.caption("Take photos in natural light")
```

with gcol2:

```
st.markdown('<div style="text-align: center; font-size: 2.5rem;"> 🍃 </div>', unsafe_allow_html=True)
st.markdown("**Focus on Leaves**")
st.caption("Capture clear leaf details")
```

with gcol3:

```
st.markdown('<div style="text-align: center; font-size: 2.5rem;"> ⚡ </div>', unsafe_allow_html=True)
st.markdown("**Quick Results**")
st.caption("Get diagnosis within seconds")
```

```

# =====
# FOOTER
# =====

st.markdown("---")

st.markdown('<div style="text-align: center; color: #666; padding: 2rem;">', unsafe_allow_html=True)

st.markdown('<p style="font-size: 1.1rem; font-weight: 600; color: #2E7D32;"> 🌱 Crop Disease Detection System</p>', unsafe_allow_html=True)

st.markdown('<p>Final Year Project | Computer Science Department</p>', unsafe_allow_html=True)

st.markdown('<p style="font-size: 0.9rem; margin-top: 0.5rem;">© 2024 [Your University Name]. Voice-enabled AI detection system.</p>', unsafe_allow_html=True)

st.markdown('</div>', unsafe_allow_html=True)

```

weather forecast using API

```

<!DOCTYPE html>

<html lang="en">

    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Weather Forecast</title>
        <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
              integrity="sha512-  

iecdLmaskI7CVkqkXNQ/ZH/XLlvWZOJyj7Yy7tcenmpD1ypASozpmT/E0iPtmFIB46ZmdtAc9eNBvH0H/ZpiBw=="
              ">

            crossorigin="anonymous" referrerpolicy="no-referrer" />
        <link rel="stylesheet" href="style.css">
    </head>

    <body>

```

```
<header class="header">  
  <h1 class="header-title">WEATHER FORECAST</h1>  
</header>  
  
<div class="container">  
  <div class="header-sec">  
    <div class="search-box">  
      <input type="text" placeholder="Enter your location" class="input-box">  
  
      <button class="fa-solid fa-magnifying-glass" id="searchBtn"></button>  
    </div>  
  </div>  
  
<div class="location-not-found">  
  <h1>Sorry, Location not found!!!</h1>  
    
</div>  
  
<div class="weather-body">  
    
  
  <div class="weather-box">  
    <p class="temperature">0 <sup>°C</sup></p>  
    <p class="description">light rain</p>  
  </div>  
  
<div class="weather-details">  
  <div class="humidity">  
    <i class="fa-sharp fa-solid fa-droplet"></i>  
    <div class="text">
```

```
<span id="humidity">45%</span>
<p>Humidity</p>
</div>
</div>

<div class="wind">
  <i class="fa-solid fa-wind"></i>
  <div class="text">
    <span id="wind-speed">12Km/H</span>
    <p>Wind Speed</p>
  </div>
</div>
</div>
</div>
</div>

<script src="script.js"></script>
<script src="https://kit.fontawesome.com/595a890311.js" crossorigin="anonymous"></script>
</body>

</html>
```

Style.css

```
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  border: none;
```

```
outline: none;  
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
}
```

```
body{  
min-height: 100vh;  
display: flex;  
justify-content: center;  
align-items: center;
```

```
/* BRIGHT WEATHER BACKGROUND IMAGE + LIGHT OVERLAY */  
background:  
linear-gradient(135deg, rgba(0, 0, 0, 0.15), rgba(0, 0, 0, 0.35)),  
url('https://images.unsplash.com/photo-1501973801540-  
537f08ccae7b?auto=format&fit=crop&w=1600&q=80')  
center/cover no-repeat fixed;
```

```
padding: 20px;  
}
```

```
.header {  
position: fixed;  
top: 0;  
left: 0;  
width: 100%;  
/* CHANGED: nicer blue gradient instead of red */  
background: linear-gradient(135deg, #2193f3, #0052cc);  
padding: 20px;  
text-align: center;  
box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2);  
z-index: 1000;
```

```
}

.header-title {
    color: #fff;
    font-size: 28px;
    margin: 0;
    font-weight: 700;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
}


```

```
.container{
    width: 420px;
    height: min-content;
    background: linear-gradient(135deg, #ffffff, #f8f9fa);
    border-radius: 20px;
    padding: 30px;
    box-shadow: 0 15px 30px rgba(0, 0, 0, 0.35);
    margin-top: 80px;
    margin-bottom: 80px;
}


```

```
.search-box{
    width: 100%;
    height: min-content;
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 20px;
}
```

```
.search-box input{  
    width: 84%;  
    font-size: 18px;  
    text-transform: capitalize;  
    color: #2d3436;  
    background-color: #f1f3f4;  
    padding: 15px 20px;  
    border-radius: 15px;  
    transition: all 0.3s ease;  
    box-shadow: inset 0 2px 5px rgba(0, 0, 0, 0.1);  
}  
  
}
```

```
.search-box input:focus{  
    background-color: #e8f4f8;  
    box-shadow: inset 0 2px 10px rgba(74, 144, 226, 0.2);  
}  
  
}
```

```
.search-box input::placeholder{  
    color: #636e72;  
}  
  
}
```

```
.search-box button{  
    width: 50px;  
    height: 50px;  
    background: linear-gradient(135deg, #74b9ff, #0984e3);  
    border-radius: 50%;  
    cursor: pointer;  
    font-size: 20px;  
    color: white;  
    transition: all 0.3s ease;  
    box-shadow: 0 4px 15px rgba(116, 185, 255, 0.4);  
}
```

}

```
.search-box button:hover{  
    transform: scale(1.1);  
    box-shadow: 0 6px 20px rgba(116, 185, 255, 0.6);  
    background: linear-gradient(135deg, #0984e3, #74b9ff);  
}
```

```
.weather-body{  
    justify-content: center;  
    align-items: center;  
    flex-direction: column;  
    margin-block: 25px;  
    display: none;  
    animation: fadeIn 0.5s ease-in;  
}
```

```
/* HIDE BROKEN WEATHER IMAGE INSIDE CARD */
```

```
.weather-body img{  
    display: none;  
}
```

```
@keyframes fadeIn {  
    from { opacity: 0; transform: translateY(20px); }  
    to { opacity: 1; transform: translateY(0); }  
}
```

```
.weather-box{  
    margin-block: 25px;  
    text-align: center;  
}
```

```
.weather-box .temperature{  
    font-size: 48px;  
    font-weight: 800;  
    position: relative;  
    color: #e17055;  
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);  
}  
  
}
```

```
.weather-box .temperature sup{  
    font-size: 24px;  
    position: absolute;  
    font-weight: 600;  
    color: #e17055;  
}  
  
}
```

```
.weather-box .description{  
    font-size: 22px;  
    font-weight: 700;  
    text-transform: capitalize;  
    color: #2d3436;  
    margin-top: 10px;  
    padding: 8px 20px;  
    background: linear-gradient(135deg, #dfe6e9, #b2bec3);  
    border-radius: 25px;  
    display: inline-block;  
}  
  
}
```

```
.weather-details{  
    width: 100%;  
    display: flex;
```

```
justify-content: space-between;  
margin-top: 35px;  
padding: 20px;  
background: linear-gradient(135deg, #f8f9fa, #e9ecef);  
border-radius: 15px;  
box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);  
}
```

```
.humidity, .wind{  
display: flex;  
align-items: center;  
flex-direction: column;  
text-align: center;  
flex: 1;  
}
```

```
.humidity{  
margin-left: 0;  
}
```

```
.wind{  
margin-right: 0;  
}
```

```
.weather-details i{  
font-size: 42px;  
margin-bottom: 10px;  
color: #0984e3;  
}
```

```
.weather-details .text{
```

```
margin-left: 0;  
font-size: 16px;  
color: #636e72;  
}  
  
}
```

```
.text span{  
font-size: 24px;  
font-weight: 700;  
color: #2d3436;  
display: block;  
margin-bottom: 5px;  
}
```

```
.location-not-found{  
margin-top: 30px;  
display: none;  
align-items: center;  
justify-content: center;  
flex-direction: column;  
text-align: center;  
animation: shake 0.5s ease-in;  
}
```

```
@keyframes shake {  
0%, 100% { transform: translateX(0); }  
25% { transform: translateX(-5px); }  
75% { transform: translateX(5px); }  
}
```

```
.location-not-found h1{  
font-size: 22px;
```

```
color: #e17055;  
margin-block-end: 20px;  
font-weight: 600;  
}
```

```
.location-not-found img{  
width: 70%;  
max-width: 200px;  
opacity: 0.8;  
}
```

```
.footer {  
background: linear-gradient(135deg, #2d3436, #636e72);  
padding: 15px;  
text-align: center;  
position: fixed;  
bottom: 0;  
left: 0;  
right: 0;  
box-shadow: 0 -4px 15px rgba(0, 0, 0, 0.2);  
}
```

```
.footer-text {  
color: #fff;  
font-size: 14px;  
margin: 0;  
font-weight: 500;  
}
```

```
/* Loading animation */  
.loading {
```

```
display: none;  
text-align: center;  
margin: 20px 0;  
}
```

```
.loading-spinner {  
width: 40px;  
height: 40px;  
border: 4px solid #f3f3f3;  
border-top: 4px solid #74b9ff;  
border-radius: 50%;  
animation: spin 1s linear infinite;  
margin: 0 auto;  
}
```

```
@keyframes spin {  
0% { transform: rotate(0deg); }  
100% { transform: rotate(360deg); }  
}
```

```
/* Weather condition specific backgrounds – all use same nice image */
```

```
body.clear-sky,  
body.clouds,  
body.rain,  
body.snow,  
body.thunderstorm,  
body.mist,  
body.haze,  
body.fog {  
background:  
linear-gradient(135deg, rgba(0, 0, 0, 0.15), rgba(0, 0, 0, 0.35)),
```

```
url('https://images.unsplash.com/photo-1501973801540-  
537f08ccae7b?auto=format&fit=crop&w=1600&q=80')  
center/cover no-repeat fixed;  
}
```

```
/* Additional weather icons styling */
```

```
.weather-icon {  
    font-size: 80px;  
    margin: 20px 0;  
    display: none;  
}
```

```
.weather-icon.sunny { color: #fdcb6e; }
```

```
.weather-icon.cloudy { color: #b2bec3; }
```

```
.weather-icon.rainy { color: #74b9ff; }
```

```
.weather-icon.stormy { color: #a29bfe; }
```

```
.weather-icon.snowy { color: #dfe6e9; }
```

```
/* Temperature color variations based on temp */
```

```
.temperature.cold {  
    color: #0984e3;  
}
```

```
.temperature.moderate {  
    color: #00b894;  
}
```

```
.temperature.warm {  
    color: #e17055;  
}
```

```
.temperature.hot {  
    color: #d63031;  
}  
  
@media only screen and (min-width: 320px) and (max-width: 480px) {  
    body {  
        padding: 10px;  
        background:  
            linear-gradient(135deg, rgba(0, 0, 0, 0.15), rgba(0, 0, 0, 0.35)),  
            url('https://images.unsplash.com/photo-1501973801540-  
537f08ccae7b?auto=format&fit=crop&w=1600&q=80')  
            center/cover no-repeat fixed;  
    }  
  
.container {  
    width: 95%;  
    padding: 20px;  
    margin-top: 70px;  
    margin-bottom: 70px;  
}  
  
.header-title {  
    font-size: 22px;  
    padding: 15px;  
}  
  
.search-box input {  
    font-size: 16px;  
    padding: 12px 16px;  
}
```

```
.search-box button {
```

```
    width: 45px;
```

```
    height: 45px;
```

```
    font-size: 18px;
```

```
}
```

```
.weather-box .temperature {
```

```
    font-size: 36px;
```

```
}
```

```
.weather-box .temperature sup {
```

```
    font-size: 18px;
```

```
}
```

```
.weather-box .description {
```

```
    font-size: 18px;
```

```
    padding: 6px 15px;
```

```
}
```

```
.weather-details {
```

```
    padding: 15px;
```

```
    margin-top: 25px;
```

```
}
```

```
.weather-details i {
```

```
    font-size: 36px;
```

```
}
```

```
.weather-details .text {
```

```
    font-size: 14px;
```

```
}
```

```
.text span {  
    font-size: 20px;  
}  
  
.footer {  
    padding: 12px;  
}  
  
.footer-text {  
    font-size: 12px;  
}  
  
.location-not-found h1 {  
    font-size: 18px;  
}  
  
.location-not-found img {  
    width: 60%;  
}  
  
/* Tablet styles */  
@media only screen and (min-width: 481px) and (max-width: 768px) {  
    .container {  
        width: 450px;  
    }  
}  
  
/* Large screen styles */  
@media only screen and (min-width: 1200px) {
```

```
.container {  
    width: 450px;  
}  
  
}  
  
Script.js  
  
const inputBox = document.querySelector('.input-box');  
const searchBtn = document.getElementById('searchBtn');  
const weather_img = document.querySelector('.weather-img');  
const temperature = document.querySelector('.temperature');  
const description = document.querySelector('.description');  
const humidity = document.getElementById('humidity');  
const wind_speed = document.getElementById('wind-speed');  
const location_not_found = document.querySelector('.location-not-found');  
const weather_body = document.querySelector('.weather-body');  
const loading = document.createElement('div');  
  
// Create loading element  
loading.className = 'loading';  
loading.innerHTML = '<div class="loading-spinner"></div><p>Searching for weather...</p>';  
document.querySelector('.container').insertBefore(loading, weather_body);  
  
async function checkWeather(city){  
    // Show loading  
    loading.style.display = "block";  
    location_not_found.style.display = "none";  
    weather_body.style.display = "none";  
  
    const api_key = "4c4286de4f6a3794841e570fd8bc4a0b";  
    const url = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${api_key}`;
```

```
try {  
  
    const response = await fetch(url);  
  
    const weather_data = await response.json();  
  
  
    if(weather_data.cod === '404') {  
  
        location_not_found.style.display = "flex";  
  
        weather_body.style.display = "none";  
  
        loading.style.display = "none";  
  
        changeBackground('error');  
  
        return;  
  
    }  
  
  
    console.log("Weather data loaded successfully");  
  
    loading.style.display = "none";  
  
    location_not_found.style.display = "none";  
  
    weather_body.style.display = "flex";  
  
  
    const tempCelsius = Math.round(weather_data.main.temp - 273.15);  
  
    temperature.innerHTML = `${tempCelsius}°C</sup>`;  
  
    description.innerHTML = `${weather_data.weather[0].description}`;  
  
  
    humidity.innerHTML = `${weather_data.main.humidity}%`;  
  
    wind_speed.innerHTML = `${weather_data.wind.speed} Km/H`;  
  
  
    // Update temperature color based on temperature  
  
    updateTemperatureColor(tempCelsius);  
  
  
    // Change background based on weather condition  
  
    changeBackground(weather_data.weather[0].main.toLowerCase());  
  
  
    // Update weather image
```

```
updateWeatherImage(weather_data.weather[0].main, weather_data.weather[0].description);

console.log(weather_data);

} catch (error) {
    console.error("Error fetching weather data:", error);
    loading.style.display = "none";
    location_not_found.style.display = "flex";
    weather_body.style.display = "none";
    changeBackground('error');
}

}

function updateWeatherImage(weatherMain, weatherDescription) {
    const weatherCondition = weatherMain.toLowerCase();

    switch(weatherCondition){
        case 'clouds':
            if (weatherDescription.toLowerCase().includes('few') ||
weatherDescription.toLowerCase().includes('scattered')) {
                weather_img.src = "img/partly-cloudy.png";
            } else {
                weather_img.src = "img/cloud.png";
            }
            break;
        case 'clear':
            weather_img.src = "img/clear-sky.png";
            break;
        case 'rain':
            if (weatherDescription.toLowerCase().includes('light') || 
weatherDescription.toLowerCase().includes('drizzle')) {
                weather_img.src = "img/light-rain.png";
            }
    }
}
```

```
    } else {
        weather_img.src = "img/rain.png";
    }
    break;
}

case 'haze':
case 'mist':
case 'fog':
    weather_img.src = "img/haze.png";
    break;

case 'thunderstorm':
    weather_img.src = "img/thunderstorm.png";
    break;

case 'snow':
    weather_img.src = "img/snow.png";
    break;

case 'drizzle':
    weather_img.src = "img/drizzle.png";
    break;

default:
    weather_img.src = "img/cloud.png";
}

}
```

```
function changeBackground(weatherCondition) {
    // Remove all weather background classes
    document.body.className = "";

    // Add the appropriate background class
    switch(weatherCondition) {
        case 'clear':
            document.body.classList.add('clear-sky');
```

```
break;

case 'clouds':
    document.body.classList.add('clouds');
    break;

case 'rain':
case 'drizzle':
    document.body.classList.add('rain');
    break;

case 'snow':
    document.body.classList.add('snow');
    break;

case 'thunderstorm':
    document.body.classList.add('thunderstorm');
    break;

case 'mist':
case 'haze':
case 'fog':
    document.body.classList.add('mist');
    break;

case 'error':
    // Keep default background for errors
    break;

default:
    // Keep default background for unknown conditions
    break;
}

}

function updateTemperatureColor(temp) {
    // Remove all temperature classes
    temperature.classList.remove('cold', 'moderate', 'warm', 'hot');
```

```
// Add appropriate temperature class
if (temp < 10) {
    temperature.classList.add('cold');
} else if (temp >= 10 && temp < 20) {
    temperature.classList.add('moderate');
} else if (temp >= 20 && temp < 30) {
    temperature.classList.add('warm');
} else {
    temperature.classList.add('hot');
}

}

// Event listeners
searchBtn.addEventListener('click', ()=>{
    if(inputBox.value.trim() !== "") {
        checkWeather(inputBox.value);
    }
});

inputBox.addEventListener('keypress', (e)=>{
    if(e.key === 'Enter' && inputBox.value.trim() !== "") {
        checkWeather(inputBox.value);
    }
});

// Add some sample cities for quick testing
const sampleCities = ['London', 'New York', 'Tokyo', 'Paris', 'Sydney', 'Colombo', 'Mumbai'];

// Function to display sample cities (optional feature)
function showSampleCities() {
```

```

        console.log('Try these cities:', sampleCities.join(', '));
    }

// Initialize
showSampleCities();

// Add click animation to search button
searchBtn.addEventListener('mousedown', () => {
    searchBtn.style.transform = 'scale(0.95)';
});

searchBtn.addEventListener('mouseup', () => {
    searchBtn.style.transform = 'scale(1)';
});

searchBtn.addEventListener('mouseleave', () => {
    searchBtn.style.transform = 'scale(1)';
});

// Add input validation
inputBox.addEventListener('input', (e) => {
    // Remove any numbers from input (cities don't usually have numbers)
    e.target.value = e.target.value.replace(/\d/g, '');
});

MAIN FINAL CODE

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>KrishiSiri -Smart Agriculture Assistant</title>

```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<style>
  * {
    margin:0;
    padding:0;
    box-sizing:border-box;
    font-family:"Poppins", sans-serif;
  }

html, body {
  width:100%;
  overflow-x:hidden;
  background:#f8f9fa;
  color:#333;
  scroll-behavior:smooth;
}

a {
  text-decoration:none;
  transition:0.3s;
}

a:hover {
  color:#ffd60a;
}

img {
  max-width:100%;
  display:block;
```

}

/* Google Translate Widget Styling */

#google_translate_element {

position: fixed;

top: 80px;

right: 20px;

z-index: 1000;

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

padding: 8px;

border-radius: 8px;

box-shadow: 0 4px 12px rgba(0,0,0,0.2);

}

.goog-te-gadget-simple {

border: none !important;

background: transparent !important;

font-family: "Poppins", sans-serif !important;

}

.goog-te-gadget-simple .goog-te-menu-value span {

color: #2d6a4f !important;

}

.goog-te-gadget-simple .goog-te-menu-value span:first-child {

display: none;

}

.goog-te-gadget-simple .goog-te-menu-value span:last-child::before {

content: "🌐";

}

```
/* Hide Google Translate top bar */
.goog-te-banner-frame {
    display: none !important;
}

body {
    top: 0px !important;
}

/* AI Voice Control Button */
.voice-control {
    position: fixed;
    top: 130px;
    right: 20px;
    z-index: 1001;
}

.voice-btn {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    border: none;
    border-radius: 50px;
    padding: 12px 20px;
    font-weight: 600;
    font-size: 14px;
    cursor: pointer;
    box-shadow: 0 4px 15px rgba(0,0,0,0.2);
    transition: all 0.3s ease;
    display: flex;
    align-items: center;
}
```

gap: 8px;
}

.voice-btn:hover {
transform: translateY(-2px);
box-shadow: 0 6px 20px rgba(0,0,0,0.25);
}

.voice-btn.active {
background: linear-gradient(135deg, #ff6b6b 0%, #ee5a24 100%);
animation: pulse 1.5s infinite;
}

.voice-btn i {
font-size: 16px;
}

@keyframes pulse {
0% { box-shadow: 0 0 0 rgba(255,107,107,0.7); }
70% { box-shadow: 0 0 0 10px rgba(255,107,107,0); }
100% { box-shadow: 0 0 0 rgba(255,107,107,0); }
}

/* AI Chatbot Styles */
.chatbot-container {
position: fixed;
bottom: 20px;
right: 20px;
width: 380px;
height: 600px;
background: white;

```
border-radius: 15px;  
box-shadow: 0 10px 30px rgba(0,0,0,0.2);  
z-index: 1002;  
display: none;  
flex-direction: column;  
overflow: hidden;  
border: 2px solid #2d6a4f;  
}
```

```
.chatbot-container.active {  
display: flex;  
}
```

```
.chatbot-header {  
background: linear-gradient(135deg, #2d6a4f, #40916c);  
color: white;  
padding: 15px 20px;  
display: flex;  
justify-content: space-between;  
align-items: center;  
}
```

```
.chatbot-header h3 {  
margin: 0;  
font-size: 1.2rem;  
display: flex;  
align-items: center;  
gap: 10px;  
}
```

```
.chatbot-close {
```

```
background: none;  
border: none;  
color: white;  
font-size: 1.2rem;  
cursor: pointer;  
padding: 5px;  
}
```

```
.chatbot-messages {  
flex: 1;  
padding: 20px;  
overflow-y: auto;  
background: #f8f9fa;  
}
```

```
.message {  
margin-bottom: 15px;  
display: flex;  
align-items: flex-start;  
gap: 10px;  
}
```

```
.message.user {  
flex-direction: row-reverse;  
}
```

```
.message-avatar {  
width: 35px;  
height: 35px;  
border-radius: 50%;  
display: flex;
```

```
  align-items: center;  
  justify-content: center;  
  font-size: 0.9rem;  
  flex-shrink: 0;  
}  
  
}
```

```
.message.bot .message-avatar {  
  background: #2d6a4f;  
  color: white;  
}  
  
}
```

```
.message.user .message-avatar {  
  background: #667eea;  
  color: white;  
}  
  
}
```

```
.message-content {  
  max-width: 70%;  
  padding: 12px 15px;  
  border-radius: 18px;  
  font-size: 0.9rem;  
  line-height: 1.4;  
}  
  
}
```

```
.message.bot .message-content {  
  background: white;  
  border: 1px solid #e9ecef;  
  color: #333;  
}  
  
}
```

```
.message.user .message-content {
```

```
background: #667eea;  
color: white;  
}
```

```
.chatbot-input {  
padding: 15px 20px;  
background: white;  
border-top: 1px solid #e9ecef;  
display: flex;  
gap: 10px;  
align-items: center;  
}
```

```
.chatbot-input input {  
flex: 1;  
padding: 12px 15px;  
border: 1px solid #ddd;  
border-radius: 25px;  
outline: none;  
font-size: 0.9rem;  
}
```

```
.chatbot-input input:focus {  
border-color: #2d6a4f;  
}
```

```
.chatbot-input button {  
background: #2d6a4f;  
color: white;  
border: none;  
border-radius: 50%;
```

```
width: 45px;  
height: 45px;  
cursor: pointer;  
display: flex;  
align-items: center;  
justify-content: center;  
transition: all 0.3s ease;  
}
```

```
.chatbot-input button:hover {  
background: #40916c;  
transform: scale(1.05);  
}
```

```
.chatbot-tools {  
display: flex;  
gap: 10px;  
padding: 10px 20px;  
background: #f8f9fa;  
border-top: 1px solid #e9ecef;  
}
```

```
.tool-btn {  
background: white;  
border: 1px solid #ddd;  
border-radius: 20px;  
padding: 8px 15px;  
font-size: 0.8rem;  
cursor: pointer;  
display: flex;  
align-items: center;
```

```
gap: 5px;  
transition: all 0.3s ease;  
}  
  
}
```

```
.tool-btn:hover {  
background: #2d6a4f;  
color: white;  
border-color: #2d6a4f;  
}
```

```
.tool-btn.active {  
background: #2d6a4f;  
color: white;  
border-color: #2d6a4f;  
}
```

```
.uploaded-image {  
max-width: 200px;  
border-radius: 10px;  
margin-top: 5px;  
border: 2px solid #e9ecef;  
}
```

```
.chatbot-toggle {  
position: fixed;  
bottom: 20px;  
right: 20px;  
width: 60px;  
height: 60px;  
background: linear-gradient(135deg, #2d6a4f, #40916c);  
color: white;
```

```
border: none;  
border-radius: 50%;  
cursor: pointer;  
display: flex;  
align-items: center;  
justify-content: center;  
font-size: 1.5rem;  
box-shadow: 0 4px 15px rgba(0,0,0,0.2);  
z-index: 1001;  
transition: all 0.3s ease;  
}
```

```
.chatbot-toggle:hover {  
    transform: scale(1.1);  
    box-shadow: 0 6px 20px rgba(0,0,0,0.3);  
}
```

```
.file-input {  
    display: none;  
}
```

```
/* Navbar */  
.header {  
    position:fixed;  
    top:0;  
    width:100%;  
    background-color:rgba(45,106,79,0.95);  
    color:white;  
    display:flex;  
    align-items:center;  
    justify-content:space-between;
```

```
padding:12px 5%;  
z-index:1000;  
backdrop-filter:blur(6px);  
box-shadow:0 2px 10px rgba(0,0,0,0.2);  
}
```

```
.logo {  
display: flex;  
align-items: center;  
}
```

```
.logo h1 {  
font-size:1.6rem;  
color:#f4fce3;  
display:flex;  
align-items:center;  
}
```

```
.logo img {  
height:45px;  
margin-right:10px;  
border-radius:50%;  
}
```

```
.navbar {  
display: flex;  
align-items: center;  
}
```

```
.navbar a {  
color:#fff;
```

```
margin:0 10px;  
font-weight:500;  
font-size:0.95rem;  
position: relative;  
}
```

```
.navbar a::after {  
content: " ";  
position: absolute;  
bottom: -5px;  
left: 0;  
width: 0;  
height: 2px;  
background: #ffd60a;  
transition: width 0.3s ease;  
}
```

```
.navbar a:hover::after {  
width: 100%;  
}
```

```
.navbar .btn {  
background-color:#52b788;  
padding:6px 12px;  
border-radius:6px;  
font-weight:600;  
font-size:0.9rem;  
}
```

```
.navbar .btn:hover {  
background-color:#40916c;
```

```
}
```

```
#menu-btn {  
    display: none;  
    font-size: 1.5rem;  
    cursor: pointer;  
}
```

```
/* Home - Updated Background Image */
```

```
.home {  
    display:flex;  
    align-items:center;  
    justify-content:center;  
    min-height:100vh;  
    padding:100px 5% 80px;  
    background:linear-gradient(rgba(5, 125, 49, 0.1), rgba(29, 198, 10, 0.029)),  
        url('https://img.freepik.com/premium-photo/beautiful-sunset-green-field-with-tree-foreground_924629-12969.jpg') center/cover no-repeat;  
    color:#fff;  
    text-shadow:1px 1px 4px rgba(0,0,0,0.6);  
    flex-direction:column;  
    text-align:center;  
    position: relative;  
    background-size: cover;  
    background-position: center;  
}
```

```
.home::before {  
    content: " ";  
    position: absolute;  
    top: 0;
```

```
left: 0;  
width: 100%;  
height: 100%;  
background: linear-gradient(45deg, rgba(45,106,79,0.3), rgba(168,218,220,0.3));  
z-index: -1;  
}
```

```
.home h1 {  
font-size:2.8rem;  
margin-bottom:10px;  
color:#f1faee;  
text-shadow: 2px 2px 6px rgba(0,0,0,0.7);  
}
```

```
.home p {  
font-size:1.2rem;  
margin-bottom:20px;  
color:#f8f9fa;  
}
```

```
.home-btn {  
background-color:#2d6a4f;  
color:#fff;  
padding:16px 32px;  
border-radius:8px;  
font-weight:600;  
font-size:1.1rem;  
transition:0.3s;  
box-shadow:0 5px 15px rgba(0,0,0,0.3);  
display: inline-block;  
margin: 8px;
```

```
position: relative;  
overflow: hidden;  
border: none;  
cursor: pointer;  
min-width: 220px;  
}
```

```
.home-btn::before {  
content: " ";  
position: absolute;  
top: 0;  
left: -100%;  
width: 100%;  
height: 100%;  
background: linear-gradient(90deg, transparent, rgba(255,255,255,0.2), transparent);  
transition: 0.5s;  
}
```

```
.home-btn:hover::before {  
left: 100%;  
}
```

```
.home-btn:hover {  
background-color:#74c69d;  
transform:translateY(-3px);  
}
```

```
/* Farming Words Section - Improved Card Sizes */  
.farming-words {  
display: flex;  
justify-content: center;
```

```
gap: 30px;  
margin: 40px 0;  
flex-wrap: wrap;  
}  
  
.word-card {  
background: rgba(255, 255, 255, 0.9);  
border-radius: 12px;  
padding: 30px 20px;  
text-align: center;  
min-width: 200px;  
box-shadow: 0 5px 15px rgba(0,0,0,0.1);  
transition: transform 0.3s;  
position: relative;  
overflow: hidden;  
flex: 1;  
max-width: 250px;  
}  
  
.word-card::before {  
content: " ";  
position: absolute;  
top: 0;  
left: 0;  
width: 100%;  
height: 5px;  
background: linear-gradient(90deg, #52b788, #2d6a4f);  
}  
  
.word-card:hover {  
transform: translateY(-5px);
```

```
}
```

```
.word-card i {  
    font-size: 2.5rem;  
    color: #2d6a4f;  
    margin-bottom: 15px;  
}
```

```
.word-card h4 {  
    margin-bottom: 15px;  
    color: #2d6a4f;  
    font-size: 1.3rem;  
}
```

```
.word-card p {  
    font-weight: bold;  
    font-size: 1.2rem;  
    color: #1b4332;  
    line-height: 1.4;  
}
```

```
/* Features - New Background */
```

```
.features {  
    padding: 80px 5%;  
    text-align: center;  
    background: linear-gradient(rgba(51, 156, 16, 0.452), rgba(0, 0, 0, 0.6)),  
        url('https://thumbs.dreamstime.com/z/different-types-crops-grown-adjacent-rows-farmer-practicing-crop-rotation-sustainable-farm-where-different-types-324576474.jpg?w=992') center/cover no-repeat;  
    color: #fff;  
    position: relative;  
    overflow: hidden;
```

}

```
.features h1 {  
color:#ffd60a;  
font-size:2rem;  
margin-bottom:10px;  
}
```

```
.features p {  
margin-bottom:30px;  
font-size:1.1rem;  
}
```

```
.buttonn {  
display:inline-block;  
margin:10px;  
padding:12px 20px;  
font-weight:600;  
font-size:1rem;  
border-radius:12px;  
background:linear-gradient(145deg,#52b788,#2d6a4f);  
color:white;  
box-shadow:0 8px 15px rgba(0,0,0,0.3);  
transition: all 0.3s ease;  
text-align:center;  
position: relative;  
overflow: hidden;  
border: none;  
cursor: pointer;  
}
```

```
.buttonn::before {  
    content: " ";  
    position: absolute;  
    top: 0;  
    left: -100%;  
    width: 100%;  
    height: 100%;  
    background: linear-gradient(90deg, transparent, rgba(255,255,255,0.2), transparent);  
    transition: 0.5s;  
}  
  
/*
```

```
.buttonn:hover::before {  
    left: 100%;  
}  
  
.buttonn:hover {  
    transform:translateY(-6px) scale(1.05);  
    box-shadow:0 15px 25px rgba(0,0,0,0.4);  
    background:linear-gradient(145deg,#74c69d,#40916c);  
}  
/*
```

```
/* Crop Calendar - Redesigned */  
.crop-calendar {  
    background: rgba(255, 255, 255, 0.9);  
    border-radius: 15px;  
    padding: 25px;  
    margin: 30px auto;  
    max-width: 900px;  
    box-shadow: 0 10px 30px rgba(0,0,0,0.2);  
}
```

```
.calendar-container {
```

```
    margin-top: 20px;
```

```
}
```

```
#cropSelect {
```

```
    padding: 12px;
```

```
    border-radius: 8px;
```

```
    border: 2px solid #2d6a4f;
```

```
    margin-bottom: 20px;
```

```
    background: white;
```

```
    color: #2d6a4f;
```

```
    font-weight: 600;
```

```
    width: 100%;
```

```
    max-width: 300px;
```

```
    font-size: 1rem;
```

```
}
```

```
.calendar-view {
```

```
    display: grid;
```

```
    grid-template-columns: repeat(6, 1fr);
```

```
    gap: 8px;
```

```
}
```

```
.calendar-month {
```

```
    background: #f8f9fa;
```

```
    border-radius: 8px;
```

```
    padding: 12px 8px;
```

```
    text-align: center;
```

```
    transition: all 0.3s;
```

```
    font-size: 0.9rem;
```

```
    border: 1px solid #e9ecef;
```

}

```
.calendar-month.active {  
background: #ffd60a;  
color: #1b4332;  
font-weight: bold;  
transform: scale(1.05);  
box-shadow: 0 4px 8px rgba(0,0,0,0.2);  
}
```

```
.calendar-month.sowing {  
background: #a7c957;  
color: white;  
}
```

```
.calendar-month.harvesting {  
background: #e76f51;  
color: white;  
}
```

/* Weather Forecast Section - New Background */

```
.weather-forecast {  
padding: 80px 5%;  
background: linear-gradient(rgba(5, 1, 1, 0.093), rgba(0,0,0,0.7)),  
url('https://tse4.mm.bing.net/th/id/OIP.tA6naA3hS2LUvi0lvbS8PgHaEH?rs=1&pid=ImgDetMain&o=7&rm=3') center/cover no-repeat;  
color: white;  
text-align: center;  
}
```

```
.weather-cards {
```

```
display: flex;
justify-content: center;
gap: 20px;
flex-wrap: wrap;
margin-top: 30px;
}

.weather-card {
background: rgba(255, 255, 255, 0.9);
border-radius: 15px;
padding: 20px;
width: 150px;
text-align: center;
color: #2d6a4f;
box-shadow: 0 8px 20px rgba(0,0,0,0.2);
transition: transform 0.3s;
}

.weather-card:hover {
transform: translateY(-10px);
}

.weather-card i {
font-size: 2.5rem;
margin-bottom: 15px;
}

/* Advantages - New Background */
.advantages {
background: linear-gradient(rgba(0, 0, 0, 0.264), rgba(0,0,0,0.7)),
```

```
url('https://thumbs.dreamstime.com/z/different-types-crops-grown-adjacent-rows-farmer-practicing-crop-rotation-sustainable-farm-where-different-types-324576474.jpg?w=992') center/cover no-repeat;  
color:#fff;  
padding:80px 5%;  
text-align:center;  
position: relative;  
}
```

```
.advantages h1 { color:#ffd60a; font-size:2rem; margin-bottom:40px; }
```

```
.advantages .row {  
display: flex;  
flex-wrap: wrap;  
justify-content: center;  
gap: 30px;  
}
```

```
.adv-card {  
flex: 1 1 250px;  
max-width: 300px;  
background: linear-gradient(145deg,#74c69d,#52b788);  
border-radius: 20px;  
padding: 25px;  
color: white;  
box-shadow: 0 15px 30px rgba(0,0,0,0.2);  
transition: 0.4s;  
text-align: center;  
position: relative;  
overflow: hidden;  
}
```

```
.adv-card::before {  
    content: " ";  
    position: absolute;  
    top: -50%;  
    left: -50%;  
    width: 200%;  
    height: 200%;  
    background: radial-gradient(circle, rgba(255,255,255,0.1) 0%, rgba(255,255,255,0) 70%);  
    transform: rotate(30deg);  
    transition: all 0.5s ease;  
    opacity: 0;  
}  
  
}
```

```
.adv-card:hover::before {  
    opacity: 1;  
    transform: rotate(30deg) translate(10%, 10%);  
}  
  
}
```

```
.adv-card:hover {  
    transform: translateY(-15px) rotateX(3deg) rotateY(3deg) scale(1.05);  
    box-shadow: 0 25px 40px rgba(0,0,0,0.3);  
}  
  
}
```

```
.adv-card i { font-size: 2rem; margin-bottom: 15px; }  
.adv-card h3 { margin-bottom: 12px; font-size: 1.2rem; font-weight: 600; }  
.adv-card p { font-size: 1rem; line-height: 1.4; }
```

```
/* About / Team - Updated to show images clearly */
```

```
.about {  
    background: linear-gradient(135deg, #aaafb6ed, #34495e);  
    color:rgb(217, 224, 228);
```

```
padding:80px 5%;  
text-align:center;  
position: relative;  
overflow: hidden;  
}  
  
.about::before {  
content: " ";  
position: absolute;  
top: 0;  
left: 0;  
width: 100%;  
height: 100%;  
background: url('https://www.discoverengineering.org/wp-content/uploads/2024/11/guides-to-master-engineering-exams.jpeg') center/cover no-repeat;  
opacity: 0.1;  
z-index: 0;  
}  
  
.about > * {  
position: relative;  
z-index: 1;  
}  
  
.about h1 { margin-bottom:30px; font-size:2rem; color:#ffd60a; }  
.about .box-container { display:flex; flex-wrap:wrap; justify-content:center; gap:20px; }  
.about .box {  
background: linear-gradient(145deg, #ffffff, #f0f0f0);  
color:#1b4332;  
border-radius:12px;  
padding:20px;
```

```
width:250px;  
text-align:center;  
transition:0.4s;  
box-shadow:0 10px 20px rgb(236, 234, 244);  
position: relative;  
overflow: hidden;  
}  
  
}
```

```
.about .box::before {  
content: "";  
position: absolute;  
top: 0;  
left: 0;  
width: 100%;  
height: 5px;  
background: linear-gradient(90deg, #52b788, #2d6a4f);  
}
```

```
.about .box:hover { transform:translateY(-10px) scale(1.05); box-shadow:0 20px 30px rgba(0,0,0,0.3); }  
.about .box img {  
border-radius:50%;  
height:120px;  
width:120px;  
object-fit:cover;  
margin-bottom:15px;  
transition:0.3s;  
border: 3px solid #52b788;  
box-shadow: 0 4px 8px rgba(0,0,0,0.2);  
}  
.about .box img:hover { transform:scale(1.1); }  
.about .box h3 { margin-bottom:8px; }
```

```
.about .box p { font-size:0.9rem; color:#2d6a4f; }

/* Footer */

.footer {
    background: linear-gradient(135deg, #2a2a2a, #1a1a1a);
    color:rgb(230, 222, 222);
    padding:40px 5% 20px;
    text-align:center;
    position: relative;
}

.footer::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: url('https://static.theprint.in/wp-content/uploads/2021/04/20200628060L-1.jpg')
    center/cover no-repeat;
    opacity: 0.05;
    z-index: 0;
}

.footer > * {
    position: relative;
    z-index: 1;
}

.footer .credit { border-top:1px solid #2d6a4f; margin-top:20px; padding-top:10px; font-size:0.9rem; }
.footer span { color:#ffd60a; font-weight:600; }
```

```
@keyframes fadeInUp {  
from {  
    opacity: 0;  
    transform: translateY(30px);  
}  
to {  
    opacity: 1;  
    transform: translateY(0);  
}  
}  
  
.animate-fadeInUp {  
    animation: fadeInUp 0.8s ease forwards;  
}  
  
.delay-1 { animation-delay: 0.2s; }  
.delay-2 { animation-delay: 0.4s; }  
.delay-3 { animation-delay: 0.6s; }  
  
.loader {  
    display: none;  
    width: 48px;  
    height: 48px;  
    border: 5px solid #FFF;  
    border-bottom-color: transparent;  
    border-radius: 50%;  
    animation: rotation 1s linear infinite;  
    margin: 20px auto;
```

}

```
@keyframes rotation {  
0% { transform: rotate(0deg); }  
100% { transform: rotate(360deg); }  
}
```

```
@media (max-width: 1024px) {  
.home h1 {  
font-size: 2.4rem;  
}
```

```
.calendar-view {  
grid-template-columns: repeat(4, 1fr);  
}
```

```
.voice-btn {  
padding: 10px 16px;  
font-size: 13px;  
}
```

```
.chatbot-container {  
width: 350px;  
height: 500px;  
}  
}
```

```
@media (max-width: 768px){  
.navbar {
```

```
display: none;  
position: absolute;  
top: 100%;  
left: 0;  
width: 100%;  
background: rgba(45,106,79,0.95);  
flex-direction: column;  
padding: 20px;  
box-shadow: 0 5px 15px rgba(0,0,0,0.2);  
}
```

```
.navbar.active {  
display: flex;  
}
```

```
.navbar a {  
margin: 10px 0;  
font-size: 1.1rem;  
}
```

```
#menu-btn {  
display: block;  
font-size: 1.5rem;  
}
```

```
.home h1 {  
font-size: 2rem;  
}  

```

```
.home p {  
font-size: 1rem;
```

}

.features h1, .advantages h1, .about h1 {

font-size: 1.5rem;

}

.adv-card {

flex: 1 1 90%;

max-width: 90%;

}

.about .box {

width: 90%;

}

#google_translate_element {

top: 70px;

right: 10px;

}

.calendar-view {

grid-template-columns: repeat(3, 1fr);

}

.home-btn {

padding: 14px 28px;

font-size: 1rem;

min-width: 200px;

}

.farming-words {

```
flex-direction: column;  
align-items: center;  
}
```

```
.word-card {  
width: 100%;  
max-width: 250px;  
}
```

```
.voice-control {  
top: 130px;  
right: 15px;  
}
```

```
.voice-btn {  
padding: 8px 14px;  
font-size: 12px;  
}
```

```
.chatbot-container {  
width: 320px;  
height: 450px;  
bottom: 10px;  
right: 10px;  
}
```

```
.chatbot-toggle {  
bottom: 10px;  
right: 10px;  
width: 50px;  
height: 50px;
```

```
font-size: 1.3rem;
```

```
}
```

```
}
```

```
@media (max-width: 480px) {
```

```
.home h1 {
```

```
    font-size: 1.8rem;
```

```
}
```

```
.home p {
```

```
    font-size: 0.9rem;
```

```
}
```

```
.calendar-view {
```

```
    grid-template-columns: repeat(2, 1fr);
```

```
}
```

```
.weather-card {
```

```
    width: 100%;
```

```
    max-width: 300px;
```

```
}
```

```
.farming-words {
```

```
    flex-direction: column;
```

```
    align-items: center;
```

```
}
```

```
.word-card {
```

```
    width: 100%;
```

```
    max-width: 250px;
```

```
}
```

```
.home-btn {  
padding: 12px 24px;  
font-size: 0.95rem;  
min-width: 180px;  
}  
  
 .voice-btn {  
padding: 6px 12px;  
font-size: 11px;  
}
```

```
.chatbot-container {  
width: 300px;  
height: 400px;  
}
```

```
.message-content {  
max-width: 85%;  
}  
}  
</style>  
</head>  
<body>
```

```
<button class="chatbot-toggle" id="chatbotToggle">  
<i class="fas fa-robot"></i>  
</button>
```

```
<div class="chatbot-container" id="chatbotContainer">

<div class="chatbot-header">
  <h3><i class="fas fa-robot"></i> KrishiSiri AI Assistant</h3>
  <button class="chatbot-close" id="chatbotClose">
    <i class="fas fa-times"></i>
  </button>
</div>

<div class="chatbot-messages" id="chatbotMessages">
  <div class="message bot">
    <div class="message-avatar">
      <i class="fas fa-robot"></i>
    </div>
    <div class="message-content">
      Hello! I am KrishiSiri AI assistant. Ask me about your farming problems, crop advice, disease diagnosis, or weather information. You can upload images or talk directly!
    </div>
  </div>
</div>

<div class="chatbot-tools">
  <button class="tool-btn" id="voiceChatBtn">
    <i class="fas fa-microphone"></i> Voice Chat
  </button>
  <button class="tool-btn" id="uploadImageBtn">
    <i class="fas fa-image"></i> Upload Image
  </button>
  <button class="tool-btn" id="clearChatBtn">
    <i class="fas fa-trash"></i> Clear Chat
  </button>
</div>
```

```
</button>

<button class="tool-btn" id="quickQuestionsBtn">
  <i class="fas fa-question-circle"></i> Quick Questions
</button>

</div>

<div class="chatbot-input">
  <input type="file" id="imageUpload" class="file-input" accept="image/*">
  <input type="text" id="chatInput" placeholder="Type your question here...">
  <button id="voiceInputBtn" class="chatbot-input-btn">
    <i class="fas fa-microphone"></i>
  </button>
  <button id="imageInputBtn" class="chatbot-input-btn">
    <i class="fas fa-image"></i>
  </button>
  <button id="sendMessageBtn">
    <i class="fas fa-paper-plane"></i>
  </button>
</div>
</div>

<div class="voice-control">
  <button class="voice-btn" id="voiceToggle">
    <i class="fas fa-volume-up"></i>
    <span id="voiceStatus">AI Voice: OFF</span>
  </button>
</div>
```

```
<div id="google_translate_element"></div>

<header class="header">
  <div class="logo">
    <div style="height:45px; width:45px; background:#52b788; border-radius:50%; display:flex; align-items:center; justify-content:center; margin-right:10px; color:white;">
      <i class="fas fa-seedling"></i>
    </div>
    <h1>KrishiSiri</h1>
  </div>
  <nav class="navbar" id="navbar">
    <a href="#home">Home</a>
    <a href="#features">Features</a>
    <a href="#advantages">Advantages</a>
    <a href="#about">About Us</a>
    <a href="#footer">Contact</a>
    <a href="explore/index.html" class="btn">Explore</a>
  </nav>
  <div class="fas fa-bars" id="menu-btn" onclick="toggleNavbar()"></div>
</header>

<section class="home" id="home">
  <div class="content">
    <h1 id="homeTitle" class="animate-fadeInUp">Your Smart Agriculture Assistant</h1>
    <p id="homeText" class="animate-fadeInUp delay-1">A new beginning for the green revolution with KrishiSiri!</p>
  </div>
</section>
```

```
<div class="farming-words animate-fadeInUp delay-2">

<div class="word-card">
  <i class="fas fa-seedling"></i>
  <h4>Crop Planning</h4>
  <p>Right crop at<br>the right time</p>
</div>

<div class="word-card">
  <i class="fas fa-tint"></i>
  <h4>Irrigation</h4>
  <p>Efficient use<br>of water</p>
</div>

<div class="word-card">
  <i class="fas fa-leaf"></i>
  <h4>Organic Farming</h4>
  <p>Environment-friendly<br>farming practice</p>
</div>
</div>

<div class="animate-fadeInUp delay-3">
  <a href="explore/index.html" class="home-btn" target="_blank">Start Exploring</a>
</div>
</div>
</section>

<section class="features" id="features">
  <h1 id="featuresTitle" class="animate-fadeInUp">Features</h1>
  <p id="featuresText" class="animate-fadeInUp delay-1">AI and data analytics based agricultural tools.</p>
```

```
<div class="animate-fadeInUp delay-2">  
  <a href="https://meghana-1422-krishi-siri-weather-forecastimgrecom-hnstwg.streamlit.app/"  
  class="buttonn">Crop Recommendation</a>  
  <a href="https://atbyqzxbhxdinn9ba4sk8s.streamlit.app/" class="buttonn">Plant Disease Detection</a>  
  <a href="weather-forecast/index.html" class="buttonn">Weather Forecast</a>  
  <a href="guide/index.html" class="buttonn">Smart Farming Guide</a>  
</div>
```

```
<div class="crop-calendar animate-fadeInUp delay-3">  
  <h3>Crop Calendar</h3>  
  <p>Know the best crop timing for your region</p>  
  <div class="calendar-container">  
    <select id="cropSelect" onchange="updateCalendar()">  
      <option value="rice">Rice</option>  
      <option value="ragi">Ragi (Finger Millet)</option>  
      <option value="jowar">Jowar (Sorghum)</option>  
      <option value="maize">Maize</option>  
      <option value="sugarcane">Sugarcane</option>  
      <option value="cotton">Cotton</option>  
      <option value="groundnut">Groundnut</option>  
      <option value="sunflower">Sunflower</option>  
      <option value="chilli">Chilli</option>  
      <option value="turmeric">Turmeric</option>  
      <option value="onion">Onion</option>  
      <option value="tomato">Tomato</option>  
    </select>  
    <div id="calendarView" class="calendar-view">
```

```
</div>
</div>
</div>
</section>

<section class="weather-forecast" id="weather">
  <h1 class="animate-fadeInUp">5-Day Weather Forecast</h1>
  <p class="animate-fadeInUp delay-1">Know your region's weather in advance</p>

  <div class="weather-cards">
    <div class="weather-card animate-fadeInUp delay-1">
      <i class="fas fa-sun" style="color: #ffd60a;"></i>
      <h4>Today</h4>
      <p>28°C</p>
      <p>Morning Chill</p>
    </div>
    <div class="weather-card animate-fadeInUp delay-2">
      <i class="fas fa-cloud-sun" style="color: #74c69d;"></i>
      <h4>Tomorrow</h4>
      <p>30°C</p>
      <p>Partly Cloudy</p>
    </div>
    <div class="weather-card animate-fadeInUp delay-3">
      <i class="fas fa-cloud-rain" style="color: #457b9d;"></i>
      <h4>Day 3</h4>
      <p>26°C</p>
      <p>Chance of Rain</p>
    </div>
    <div class="weather-card animate-fadeInUp delay-1">
```

```
<i class="fas fa-cloud" style="color: #a8dadc;"></i>
<h4>Day 4</h4>
<p>29°C</p>
<p>Cloudy</p>
</div>

<div class="weather-card animate-fadeInUp delay-2">
<i class="fas fa-sun" style="color: #ffd60a;"></i>
<h4>Day 5</h4>
<p>31°C</p>
<p>Sunny</p>
</div>
</div>
</section>
```

```
<section class="advantages" id="advantages">
<h1 class="animate-fadeInUp">KrishiSiri Advantages</h1>
<div class="row">
<div class="adv-card animate-fadeInUp delay-1">
<i class="fas fa-cloud-sun"></i>
<h3>Accurate Forecast</h3>
<p>Timely weather and disease information.</p>
</div>
<div class="adv-card animate-fadeInUp delay-2">
<i class="fas fa-seedling"></i>
<h3>Increased Yield</h3>
<p>Improve crop results with AI recommendations.</p>
</div>
<div class="adv-card animate-fadeInUp delay-3">
<i class="fas fa-leaf"></i>
```

```
<h3>Eco-Friendly</h3>
<p>Reduce chemical usage and promote crop rotation.</p>
</div>
</div>
</section>

<section class="about" id="about">
<h1 class="animate-fadeInUp">About Us</h1>
<div class="box-container">
<div class="box animate-fadeInUp delay-1">

<h3>Pallavi Vinayak Belgumkar</h3>
<p>B.E Computer Science</p>
<p>9448068525</p>
<p>pbelgumkar@gmail.com</p>
<p>github.com/pallavi</p>
</div>
<div class="box animate-fadeInUp delay-2">

<h3>Meghana V Swamy</h3>
<p>B.E Computer Science</p>
<p>9035580041</p>
<p>meghanaswamy13@gmail.com</p>
<p>https://github.com/meghana-arch</p>
</div>
<div class="box animate-fadeInUp delay-3">

```

```
<h3>Triveni Kammar</h3>
<p>B.E Computer Science</p>
<p>9148311073</p>
<p>trivenikammar65@gmail.com</p>
<p>https://github.com/Triveni-Kammar/</p>
</div>
</div>
</section>
```

```
<footer class="footer" id="footer">
<p>📞 Contact: Pallavi | Meghana | Triveni</p>
<div class="credit">© 2025 <span>KrishiSiri</span> | All Rights Reserved</div>
</footer>
```

```
<div class="loader" id="pageLoader"></div>
```

```
<script>
// Chatbot Questions Database with Correct Answers
const chatbotQuestions = [
    "What is the best crop for red soil?", 
    "How to control leaf spot disease?", 
    "What is the ideal rainfall for rice cultivation?", 
    "How to improve soil fertility naturally?", 
    "What are the symptoms of nitrogen deficiency?", 
    "How much water does wheat need?", 
    "What is crop rotation?", 
    "How to prevent pest attacks organically?",
```

"When to harvest tomatoes?",
"What is drip irrigation?",
"How to test soil pH?",
"What are cover crops?",
"How to store grains properly?",
"What is precision farming?",
"How to control weeds without chemicals?"
];

// Enhanced AI Responses Database with Correct Answers

```
const aiResponses = {  
    // Soil and Crop Questions
```

"red soil": "Red soil is best suited for crops like groundnut, potato, maize, millets, pulses, and oilseeds. It is rich in iron but poor in nitrogen, phosphorus, and organic matter. For optimal yield, add organic compost and balanced fertilizers.",

"best crop for red soil": "In red soil, you can successfully grow: Groundnut, Potato, Maize, Ragi (Finger Millet), Pulses (red gram, green gram), and Oilseeds (sunflower, castor). These crops adapt well to the acidic nature of red soil.",

"soil": "Red soil requires crops that tolerate acidity and low fertility. Recommended crops: Groundnut, Potato, Maize, Millet, Pulses. Add lime to reduce acidity and organic matter to improve fertility.",

// Disease Control

"leaf spot disease": "To control leaf spot disease: 1. Remove infected leaves immediately 2. Avoid overhead watering 3. Apply fungicides like Mancozeb or Chlorothalonil 4. Use neem oil spray for organic control 5. Ensure proper plant spacing for air circulation",

"control leaf spot": "Leaf spot management: Remove affected leaves, improve air circulation, water at soil level, apply copper-based fungicides, and practice crop rotation. For organic control, use baking soda spray (1 tbsp per gallon of water).",

// Rainfall Requirements

"rainfall for rice": "Rice requires 1000-2000 mm of rainfall during its growing season. Ideal distribution: 150-200 mm per month. It needs standing water during initial growth stages (3-5 cm depth).",

"rice cultivation rainfall": "Rice grows best with 1000-1500 mm evenly distributed rainfall. Critical periods: Seedling stage (needs flooded conditions), Tillering stage (needs moist soil), Flowering stage (needs adequate water).",

// Soil Fertility

"improve soil fertility": "Natural ways to improve soil fertility: 1. Add organic compost 2. Use green manure crops 3. Practice crop rotation 4. Apply vermicompost 5. Use biofertilizers 6. Add bone meal or rock phosphate 7. Practice mulching",

"soil fertility naturally": "Enhance soil fertility naturally by: Growing leguminous crops (fix nitrogen), Adding farmyard manure, Using cover crops, Applying neem cake, Practicing zero-tillage, and Maintaining soil pH between 6.0-7.0.",

// Nutrient Deficiency

"nitrogen deficiency": "Symptoms of nitrogen deficiency: Yellowing of older leaves, stunted growth, reduced yield, pale green color throughout plant. Remedy: Apply urea, ammonium sulfate, or organic sources like compost and green manure.",

"nitrogen deficiency symptoms": "Signs: Yellowing starts from older leaves, leaves become pale green to yellow, reduced plant growth, smaller leaves. Treatment: Apply 50-100 kg urea per acre or use leguminous cover crops.",

// Water Requirements

"water for wheat": "Wheat needs 450-650 mm of water throughout its growth cycle. Critical stages: Crown root initiation (20-25 DAS), Tillering (45-50 DAS), Flowering (60-70 DAS). Apply irrigation at these stages for best yield.",

"wheat water needs": "Wheat requires 4-6 irrigations of 5-7 cm depth each. Total water requirement: 400-500 mm. Most critical irrigation: CRI stage (20-25 days after sowing) and flowering stage.",

// Crop Rotation

"crop rotation": "Crop rotation means growing different crops in sequence on the same land. Benefits: Breaks pest cycles, improves soil fertility, reduces erosion, controls weeds. Example: Rice-Wheat-Pulse or Maize-Wheat-Soybean rotation.",

"what is crop rotation": "Crop rotation is alternating different crop species in the same field over time. Example pattern: Legume-Cereal-Root crop. This maintains soil health, controls diseases, and improves yield by 10-25%.",

// Pest Control

"prevent pest attacks organically": "Organic pest control: 1. Use neem oil spray 2. Plant marigolds as companion crops 3. Apply garlic-chili spray 4. Use diatomaceous earth 5. Introduce beneficial insects 6. Practice crop rotation 7. Use yellow sticky traps",

"organic pest control": "Natural pest management: Neem seed kernel extract (5%), Cow urine spray (10%), Trichoderma application, Bird perches in fields, Light traps, and Botanical pesticides like pyrethrum.",

// Harvesting

"harvest tomatoes": "Tomatoes are ready for harvest 60-90 days after transplanting. Signs: Uniform red color, firm texture, easy separation from stem. Harvest in morning, avoid harvesting in rain, and store at 12-15°C.",

"when to harvest tomatoes": "Harvest tomatoes when fully colored but firm. For fresh market: fully ripe. For transport: breaker stage (pink). For seed saving: fully ripe on plant. Best time: Early morning when cool.",

// Irrigation

"drip irrigation": "Drip irrigation delivers water directly to plant roots through emitters. Benefits: Saves 30-70% water, reduces weed growth, prevents leaf diseases, allows fertigation. Components: Pump, filters, mainline, submain, drippers.",

"what is drip irrigation": "Drip irrigation is a micro-irrigation system that applies water slowly to the soil. Advantages: 90% efficiency, reduces waterlogging, saves fertilizer, works on slopes. Suitable for: Fruits, vegetables, plantation crops.",

// Soil Testing

"test soil pH": "Soil pH testing methods: 1. pH meter (most accurate) 2. Litmus paper 3. Chemical test kits 4. Vinegar-baking soda test 5. Professional lab analysis. Ideal pH: 6.0-7.5 for most crops.",

"soil pH test": "To test soil pH: Take soil sample from 15-20 cm depth, mix with distilled water, use pH strip or meter. Acidic soil (pH<6): add lime. Alkaline soil (pH>7.5): add gypsum or sulfur.",

// Cover Crops

"cover crops": "Cover crops protect soil between main crops. Examples: Clover, Rye, Vetch, Mustard, Cowpea. Benefits: Prevents erosion, adds organic matter, fixes nitrogen, suppresses weeds, improves soil structure.",

"what are cover crops": "Cover crops are grown to cover soil rather than for harvest. Types: Legumes (fix nitrogen), Grasses (prevent erosion), Brassicas (biofumigants). They improve soil health and subsequent crop yield.",

// Grain Storage

"store grains properly": "Proper grain storage: 1. Dry to 12-14% moisture 2. Clean thoroughly 3. Use airtight containers 4. Add neem leaves or ash 5. Store in cool, dry place 6. Use PICS bags for hermetic storage 7. Regular inspection",

"grain storage": "Grain storage requirements: Moisture <14%, Temperature <25°C, Clean storage area, Rodent-proof bins, Regular monitoring. Traditional methods: Neem leaves, ash lining, sun drying every 2-3 months.",

// Precision Farming

"precision farming": "Precision farming uses GPS, sensors, drones to optimize inputs. Benefits: 20-30% input savings, higher yields, reduced environmental impact. Tools: Soil sensors, drones, variable rate technology, yield monitors.",

"what is precision farming": "Precision agriculture applies exact inputs (water, fertilizer) where and when needed. Technologies: GIS mapping, remote sensing, auto-steering, yield mapping. Increases efficiency by 25-40%.",

// Weed Control

"control weeds without chemicals": "Non-chemical weed control: 1. Mulching (organic or plastic) 2. Manual weeding 3. Flaming 4. Solarization 5. Crop rotation 6. Cover crops 7. Biological control 8. Stale seedbed technique",

"weed control without chemicals": "Organic weed management: Use cover crops to suppress weeds, practice intercropping, apply corn gluten meal, use vinegar spray (20% acetic acid), flame weeding, and maintain healthy soil.",

// General Responses

"hello": "Hello! I'm KrishiSiri AI assistant. How can I help you with farming today? Ask me about crops, soil, pests, weather, or anything agriculture-related!",

"hi": "Hi there! I'm here to help with all your agricultural queries. What would you like to know about farming?",

"help": "I can help with: Crop selection, Pest control, Soil management, Irrigation, Fertilizer use, Disease diagnosis, Weather information, and Modern farming techniques. Just ask me anything!",

"thanks": "You're welcome! I'm always here to help with your farming success. Feel free to ask more questions anytime.",

"thank": "You're welcome! Happy farming! Let me know if you need more assistance.",

// Weather

"weather": "Current weather in your region: 28°C, moderate humidity. Chance of rain in next 3 days. 5-day forecast: Today - 28°C, Tomorrow - 30°C, Day 3 - 26°C, Day 4 - 29°C, Day 5 - 31°C.",

"rain": "Rainfall is crucial for agriculture. Average annual rainfall for optimal farming is 600-1000mm. Monitor weather forecasts regularly for better planning.",

// Technology

"ai": "AI in agriculture helps with crop monitoring, disease detection, yield prediction, and precision farming. It analyzes satellite images, weather data, and soil sensors for better decisions.",

"technology": "Modern farming technologies include drones, IoT sensors, automated irrigation, AI-based monitoring, and mobile apps for farm management.",

// Market

"price": "Crop prices fluctuate based on season, demand, and quality. Monitor local mandi rates, government MSPs, and market trends. Current average prices: Rice ₹25/kg, Wheat ₹22/kg, Tomato ₹30/kg.",

"market": "Agricultural markets (mandis) connect farmers with buyers. E-NAM (National Agricultural Market) provides online trading platform across India for better price discovery.",

// Government Schemes

"scheme": "Government schemes include: PM-KISAN (₹6000/year income support), PMFBY (crop insurance), Soil Health Card, Paramparagat Krishi Vikas Yojana (organic farming), and Kisan Credit Card.",

"subsidy": "Subsidies available for: Seeds (50%), Drip irrigation (90%), Greenhouses (50%), Tractors (25-40%), and Crop insurance (premium subsidy). Check with local agriculture department.",

};

// Map questions to specific responses

```
const questionResponseMap = {
```

"What is the best crop for red soil?": aiResponses["best crop for red soil"],

"How to control leaf spot disease?": aiResponses["leaf spot disease"],

"What is the ideal rainfall for rice cultivation?": aiResponses["rainfall for rice"],

"How to improve soil fertility naturally?": aiResponses["improve soil fertility"],

"What are the symptoms of nitrogen deficiency?": aiResponses["nitrogen deficiency symptoms"],

"How much water does wheat need?": aiResponses["water for wheat"],

"What is crop rotation?": aiResponses["crop rotation"],

```
"How to prevent pest attacks organically?": aiResponses["prevent pest attacks organically"],  
"When to harvest tomatoes?": aiResponses["harvest tomatoes"],  
"What is drip irrigation?": aiResponses["drip irrigation"],  
"How to test soil pH?": aiResponses["test soil pH"],  
"What are cover crops?": aiResponses["cover crops"],  
"How to store grains properly?": aiResponses["store grains properly"],  
"What is precision farming?": aiResponses["precision farming"],  
"How to control weeds without chemicals?": aiResponses["control weeds without chemicals"]  
};
```

```
let isListening = false;  
let recognition = null;  
let voiceEnabled = false;
```

```
function initChatbot() {  
    const chatbotToggle = document.getElementById('chatbotToggle');  
    const chatbotContainer = document.getElementById('chatbotContainer');  
    const chatbotClose = document.getElementById('chatbotClose');  
    const sendMessageBtn = document.getElementById('sendMessageBtn');  
    const chatInput = document.getElementById('chatInput');  
    const uploadImageBtn = document.getElementById('uploadImageBtn');  
    const imageUpload = document.getElementById('imageUpload');  
    const voiceChatBtn = document.getElementById('voiceChatBtn');  
    const clearChatBtn = document.getElementById('clearChatBtn');  
    const voiceInputBtn = document.getElementById('voiceInputBtn');  
    const imageInputBtn = document.getElementById('imageInputBtn');  
    const quickQuestionsBtn = document.getElementById('quickQuestionsBtn');
```

```
    chatbotToggle.addEventListener('click', () => {  
        chatbotContainer.classList.toggle('active');  
   });
```

```
chatbotClose.addEventListener('click', () => {
  chatbotContainer.classList.remove('active');
});

sendMessageBtn.addEventListener('click', sendMessage);

chatInput.addEventListener('keypress', (e) => {
  if (e.key === 'Enter') {
    sendMessage();
  }
});

uploadImageBtn.addEventListener('click', () => {
  imageUpload.click();
});

imageUpload.addEventListener('change', handleImageUpload);

voiceChatBtn.addEventListener('click', toggleVoiceChat);

voiceInputBtn.addEventListener('click', toggleVoiceInput);

imageInputBtn.addEventListener('click', () => {
  imageUpload.click();
});

clearChatBtn.addEventListener('click', clearChat);

quickQuestionsBtn.addEventListener('click', showQuickQuestions);
```

```
initSpeechRecognition();

}

function showQuickQuestions() {
    const messagesContainer = document.getElementById('chatbotMessages');

    // Add bot message
    const messageDiv = document.createElement('div');
    messageDiv.className = 'message bot';
    messageDiv.innerHTML = `
        <div class="message-avatar">
            <i class="fas fa-robot"></i>
        </div>
        <div class="message-content">
            <strong>Quick Questions You Can Ask:</strong><br><br>
            <div style="display: flex; flex-wrap: wrap; gap: 5px;">
                ${chatbotQuestions.map((q, i) =>
                    `<button class="quick-q-btn" onclick="askQuestion('${q}')"
                        style="background: #2d6a4f; color: white; border: none; padding: 8px 12px; margin: 2px; border-radius: 15px; font-size: 0.75rem; cursor: pointer; transition: all 0.3s;">
                    ${q}
                </button>`)
                .join(''))
            </div>
        </div>
    `;
    messagesContainer.appendChild(messageDiv);
    scrollToBottom();
}

function askQuestion(question) {
```

```
document.getElementById('chatInput').value = question;  
sendMessage();  
}  
  
function initSpeechRecognition() {  
  const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;  
  
  if (SpeechRecognition) {  
    recognition = new SpeechRecognition();  
    recognition.continuous = false;  
    recognition.lang = 'en-US';  
    recognition.interimResults = false;  
    recognition.maxAlternatives = 1;  
  
    recognition.onresult = (event) => {  
      const speechResult = event.results[0][0].transcript;  
      document.getElementById('chatInput').value = speechResult;  
      sendMessage();  
    };  
  
    recognition.onerror = (event) => {  
      console.error('Speech recognition error', event.error);  
      addBotMessage('Sorry, I could not recognize your voice. Please try again.');  
    };  
  
    recognition.onend = () => {  
      isListening = false;  
      updateVoiceButton();  
    };  
  } else {  
    console.warn('Speech recognition not supported in this browser');  
  }
}
```

```
document.getElementById('voiceChatBtn').disabled = true;

document.getElementById('voiceChatBtn').innerHTML = '<i class="fas fa-microphone-slash"></i> Voice
not supported';

document.getElementById('voiceInputBtn').disabled = true;

document.getElementById('voiceInputBtn').innerHTML = '<i class="fas fa-microphone-slash"></i>';

}

}

function toggleVoiceChat() {
    if (!recognition) {
        addBotMessage('Sorry, voice recognition is not supported in this browser.');
        return;
    }

    if (isListening) {
        recognition.stop();
        isListening = false;
        document.getElementById('voiceChatBtn').classList.remove('active');
    } else {
        recognition.start();
        isListening = true;
        document.getElementById('voiceChatBtn').classList.add('active');
        addBotMessage('I am listening... Please speak now.');
    }
}

updateVoiceButton();

}

function toggleVoiceInput() {
    if (!recognition) {
        addBotMessage('Sorry, voice recognition is not supported in this browser.');
    }
}
```

```
return;  
}  
  
if (isListening) {  
    recognition.stop();  
    isListening = false;  
    document.getElementById('voiceInputBtn').classList.remove('active');  
} else {  
    recognition.start();  
    isListening = true;  
    document.getElementById('voiceInputBtn').classList.add('active');  
    addBotMessage('I am listening... Please speak now.');//  
}  
  
updateVoiceButton();  
}  
  
function updateVoiceButton() {  
    const voiceChatBtn = document.getElementById('voiceChatBtn');  
    const voiceInputBtn = document.getElementById('voiceInputBtn');  
  
    if (isListening) {  
        voiceChatBtn.innerHTML = '<i class="fas fa-microphone-slash"></i> Stop';  
        voiceChatBtn.style.background = '#e74c3c';  
        voiceChatBtn.style.color = 'white';  
  
        voiceInputBtn.style.background = '#e74c3c';  
        voiceInputBtn.style.color = 'white';  
    } else {  
        voiceChatBtn.innerHTML = '<i class="fas fa-microphone"></i> Voice Chat';  
        voiceChatBtn.style.background = "";  
    }  
}
```

```
voiceChatBtn.style.color = "";

voicelInputBtn.style.background = "";
voicelInputBtn.style.color = "";

}

}

function handleImageUpload(event) {
  const file = event.target.files[0];
  if (file) {
    if (file.type.startsWith('image/')) {
      const reader = new FileReader();
      reader.onload = (e) => {
        addUserMessage('Image uploaded for analysis', e.target.result);
        setTimeout(() => {
          const responses = [
            "Analyzing image... This appears to be a healthy plant with good leaf color and structure. No signs of disease detected.",
            "Image analysis complete: I detect possible nutrient deficiency. Consider applying balanced fertilizer (NPK 19:19:19).",
            "This image shows signs of fungal infection (yellow spots with brown margins). Remove affected leaves and apply Mancozeb fungicide.",
            "Plant appears healthy but may need more sunlight. Ensure at least 6 hours of direct sunlight daily for optimal growth.",
            "I detect pest damage (chewing marks) in this image. Consider using neem oil spray (5ml per liter) or appropriate organic pesticide."
          ];
          const randomResponse = responses[Math.floor(Math.random() * responses.length)];
          addBotMessage(randomResponse);
        }, 2000);
      };
      reader.readAsDataURL(file);
    }
  }
}
```

```
        } else {
            addBotMessage('Please upload a valid image file (JPG, PNG, etc.).');
        }
    }
```

```
    event.target.value = '';
}
```

```
function sendMessage() {
    const chatInput = document.getElementById('chatInput');
    const message = chatInput.value.trim();
```

```
    if (message) {
        addUserMessage(message);
        chatInput.value = '';
```

```
        setTimeout(() => {
            generateAIResponse(message);
        }, 800);
    }
}
```

```
function addUserMessage(message, imageUrl = null) {
    const messagesContainer = document.getElementById('chatbotMessages');
    const messageDiv = document.createElement('div');
    messageDiv.className = 'message user';

    let content = `
```

```
<div class="message-avatar">
    <i class="fas fa-user"></i>
</div>
```

```
<div class="message-content">
    ${message}
</div>

if (imageUrl) {
    content += ``;
}

content += `</div>`;
messageDiv.innerHTML = content;
messagesContainer.appendChild(messageDiv);
scrollToBottom();
}

function addBotMessage(message) {
    const messagesContainer = document.getElementById('chatbotMessages');
    const messageDiv = document.createElement('div');
    messageDiv.className = 'message bot';
    messageDiv.innerHTML =
        `<div class="message-avatar">
            <i class="fas fa-robot"></i>
        </div>
        <div class="message-content">${message}</div>
    `;
    messagesContainer.appendChild(messageDiv);
    scrollToBottom();

    if (voiceEnabled) {
        speakText(message);
    }
}
```

```
function scrollToBottom() {  
  const messagesContainer = document.getElementById('chatbotMessages');  
  messagesContainer.scrollTop = messagesContainer.scrollHeight;  
}  
  
function clearChat() {  
  const messagesContainer = document.getElementById('chatbotMessages');  
  
  while (messagesContainer.children.length > 1) {  
    messagesContainer.removeChild(messagesContainer.lastChild);  
  }  
  addBotMessage('Chat history cleared. How can I help you today?');  
}  
  
function generateAIResponse(userMessage) {  
  const lowerMessage = userMessage.toLowerCase().trim();  
  let response = '';  
  
  // First check if it's a predefined question  
  if (questionResponseMap[userMessage]) {  
    response = questionResponseMap[userMessage];  
  } else {  
    // Check for keywords in the message  
    let matchedKeyword = '';  
  
    // Check for exact question matches first  
    for (const question of chatbotQuestions) {  
      if (lowerMessage.includes(question.toLowerCase())) {  
        matchedKeyword = question;  
        break;  
      }  
    }  
  }  
  return response;  
}
```

```
}

}

if (matchedKeyword && questionResponseMap[matchedKeyword]) {
  response = questionResponseMap[matchedKeyword];
} else {
  // Check for keywords in aiResponses
  const keywords = Object.keys(aiResponses);
  for (const keyword of keywords) {
    if (lowerMessage.includes(keyword.toLowerCase())) {
      response = aiResponses[keyword];
      break;
    }
  }

  // If still no match, check partial matches
  if (!response) {
    const words = lowerMessage.split(' ');
    for (const word of words) {
      if (word.length > 3) { // Only check meaningful words
        for (const keyword of keywords) {
          if (keyword.toLowerCase().includes(word) || word.includes(keyword.toLowerCase())) {
            response = aiResponses[keyword];
            break;
          }
        }
      }
      if (response) break;
    }
  }
}
```

```

// If still no response, provide a helpful suggestion
if (!response) {

    response = "I understand you're asking about: \"\" + userMessage + "\"". For more specific advice,
    please ask about:\n• Crop selection for your soil type\n• Pest or disease symptoms you're seeing\n•
    Irrigation requirements for specific crops\n• Soil testing and improvement methods\n• Government
    schemes and subsidies\nOr try one of the quick questions from the menu!";

}

}

}

addBotMessage(response);
}

let currentVoiceLanguage = 'en';

let voices = [];

let speechSynthesis = window.speechSynthesis;

const languageMap = {
    'en': { code: 'en-US', name: 'English', voice: 'en-US' },
    'hi': { code: 'hi-IN', name: 'हिन्दी', voice: 'hi-IN' },
    'kn': { code: 'kn-IN', name: 'ಕನ್ನಡ', voice: 'kn-IN' },
    'ta': { code: 'ta-IN', name: 'தமிழ்', voice: 'ta-IN' },
    'te': { code: 'te-IN', name: 'తెలుగు', voice: 'te-IN' },
    'mr': { code: 'mr-IN', name: 'मराठी', voice: 'mr-IN' },
    'bn': { code: 'bn-IN', name: 'বাংলা', voice: 'bn-IN' },
    'gu': { code: 'gu-IN', name: 'ગુજરાતી', voice: 'gu-IN' },
    'pa': { code: 'pa-IN', name: 'ਪੰਜਾਬੀ', voice: 'pa-IN' },
    'ml': { code: 'ml-IN', name: 'മലയാളം', voice: 'ml-IN' }
};

function getGoogleTargetLanguage() {

```

```
try {

    const cookieEntry = document.cookie.split('; ').find(row => row.startsWith('googtrans='));

    if (!cookieEntry) return null;

    const cookieVal = decodeURIComponent(cookieEntry.split('=')[1]);

    const parts = cookieVal.split('/').filter(Boolean);

    if (parts.length === 0) return null;

    return parts[parts.length - 1].toLowerCase();

} catch (e) {

    return null;

}

}

function updateVoiceLanguageFromTranslate() {

    const target = getGoogleTargetLanguage();

    if (target && languageMap[target]) {

        currentVoiceLanguage = target;

    }

}

setInterval(updateVoiceLanguageFromTranslate, 3000);

function loadVoices() {

    voices = speechSynthesis.getVoices();

    if (voices.length === 0) {

        setTimeout(loadVoices, 100);

    }

}

loadVoices();

if (speechSynthesis.onvoiceschanged !== undefined) {
```

```
speechSynthesis.onvoiceschanged = loadVoices;  
}  
  
function toggleVoice() {  
    voiceEnabled = !voiceEnabled;  
    const voiceToggle = document.getElementById('voiceToggle');  
    const voiceStatus = document.getElementById('voiceStatus');  
  
    if (voiceEnabled) {  
        voiceToggle.classList.add('active');  
        voiceStatus.textContent = 'AI Voice: ON';  
        speakText('AI voice assistant enabled. Click on any element to hear its description.');//  
    } else {  
        voiceToggle.classList.remove('active');  
        voiceStatus.textContent = 'AI Voice: OFF';  
        speechSynthesis.cancel();  
    }  
}  
  
function speakText(text, specificLang = null) {  
    if (!voiceEnabled) return;  
  
    if (!specificLang) {  
        updateVoiceLanguageFromTranslate();  
    }  
  
    speechSynthesis.cancel();  
    const utterance = new SpeechSynthesisUtterance(text);  
  
    const langCode = specificLang || currentVoiceLanguage;  
    const langInfo = languageMap[langCode] || languageMap['en'];
```

```
utterance.lang = langInfo.voice;  
utterance.rate = 0.9;  
utterance.pitch = 1.0;  
  
let candidateVoices = voices.filter(voice => {  
  const vLang = (voice.lang || "").toLowerCase();  
  return vLang.startsWith(langCode);  
});  
  
let preferredVoices = candidateVoices.filter(voice =>  
  voice.name.toLowerCase().includes('female') ||  
  voice.name.toLowerCase().includes('google') ||  
  voice.name.toLowerCase().includes('samantha') ||  
  voice.name.toLowerCase().includes('karen')  
);  
  
if (preferredVoices.length > 0) {  
  utterance.voice = preferredVoices[0];  
} else if (candidateVoices.length > 0) {  
  utterance.voice = candidateVoices[0];  
} else if (voices.length > 0) {  
  utterance.voice = voices[0];  
}  
  
speechSynthesis.speak(utterance);  
}  
  
function getElementText(element) {  
  if (element.tagName === 'IMG') {  
    return element.alt || 'Image';  
  }  
}
```

```
} else if (element.tagName === 'A') {
    return element.textContent || 'Link';
} else if (element.tagName === 'BUTTON') {
    return element.textContent || 'Button';
} else if (element.tagName === 'INPUT' || element.tagName === 'SELECT') {
    return element.placeholder || element.name || 'Input field';
} else {
    return element.textContent || element.innerText || 'Element';
}
```

```
function addVoiceListeners() {
    const interactiveElements = document.querySelectorAll(
        'a, button, .home-btn, .buttonn, .word-card, .adv-card, ' +
        '.weather-card, .box, .calendar-month, #cropSelect, #menu-btn'
    );
}
```

```
interactiveElements.forEach(element => {
    element.addEventListener('click', function(e) {
        if (!voiceEnabled) return;
        e.stopPropagation();
        const text = getElementText(this);
        speakText(text);
    });
});
```

```
const headings = document.querySelectorAll('h1, h2, h3, h4');
headings.forEach(heading => {
    heading.addEventListener('click', function() {
        if (!voiceEnabled) return;
        const text = getElementText(this);
```

```
    speakText(text);

  });

};

}

function updateCalendar() {
  const crop = document.getElementById('cropSelect').value;
  const calendarView = document.getElementById('calendarView');

  let sowingMonths = [];
  let harvestingMonths = [];

  if (crop === 'rice') {
    sowingMonths = ['June', 'July'];
    harvestingMonths = ['September', 'October'];
  } else if (crop === 'ragi') {
    sowingMonths = ['June', 'July'];
    harvestingMonths = ['September', 'October'];
  } else if (crop === 'jowar') {
    sowingMonths = ['June', 'July'];
    harvestingMonths = ['September', 'October'];
  } else if (crop === 'maize') {
    sowingMonths = ['June', 'July'];
    harvestingMonths = ['September', 'October'];
  } else if (crop === 'sugarcane') {
    sowingMonths = ['January', 'February'];
    harvestingMonths = ['December', 'January'];
  } else if (crop === 'cotton') {
    sowingMonths = ['June', 'July'];
    harvestingMonths = ['December', 'January'];
  } else if (crop === 'groundnut') {
```

```
sowingMonths = ['June', 'July'];

harvestingMonths = ['September', 'October'];

} else if (crop === 'sunflower') {

sowingMonths = ['June', 'July'];

harvestingMonths = ['September', 'October'];

} else if (crop === 'chilli') {

sowingMonths = ['June', 'July'];

harvestingMonths = ['September', 'October'];

} else if (crop === 'turmeric') {

sowingMonths = ['May', 'June'];

harvestingMonths = ['January', 'February'];

} else if (crop === 'onion') {

sowingMonths = ['October', 'November'];

harvestingMonths = ['February', 'March'];

} else if (crop === 'tomato') {

sowingMonths = ['June', 'July'];

harvestingMonths = ['September', 'October'];

}
```

```
let calendarHTML = "";

const allMonths = ['January', 'February', 'March', 'April', 'May', 'June',
  'July', 'August', 'September', 'October', 'November', 'December'];

allMonths.forEach(month => {

let className = "";

if (sowingMonths.includes(month)) {

className = 'sowing';

} else if (harvestingMonths.includes(month)) {

className = 'harvesting';

}

calendarHTML += `<div class="calendar-month ${className}">${month}</div>`;
```

```
});

calendarView.innerHTML = calendarHTML;

}

function toggleNavbar() {
    const navbar = document.getElementById('navbar');
    navbar.classList.toggle('active');
}

function googleTranslateElementInit() {
    new google.translate.TranslateElement({
        pageLanguage: 'en',
        includedLanguages: 'en,hi,kn,ta,te,mr,ml,gu,bn,pa',
        layout: google.translate.TranslateElement.InlineLayout.SIMPLE,
        autoDisplay: false
    }, 'google_translate_element');
}

function addGoogleTranslateScript() {
    const script = document.createElement('script');
    script.src = '//translate.google.com/translate_a/element.js?cb=googleTranslateElementInit';
    document.head.appendChild(script);
}

function showEmergencyAlert(title, message) {
    let alertDiv = document.getElementById('emergencyAlert');
    if (!alertDiv) {
        alertDiv = document.createElement('div');
        alertDiv.id = 'emergencyAlert';
        alertDiv.style.cssText = `
```

```
position: fixed;  
top: 50%;  
left: 50%;  
transform: translate(-50%, -50%);  
background: #e74c3c;  
color: white;  
padding: 20px;  
border-radius: 10px;  
box-shadow: 0 5px 15px rgba(0,0,0,0.3);  
z-index: 2000;  
display: none;  
text-align: center;  
`;  
alertDiv.innerHTML = `  


### </h3>

  


</p>

Close</button>  
`;  
document.body.appendChild(alertDiv);  
}  
  
document.getElementById('alertTitle').textContent = title;  
document.getElementById('alertMessage').textContent = message;  
alertDiv.style.display = 'block';  
}  
  
function closeEmergencyAlert() {  
const alertDiv = document.getElementById('emergencyAlert');  
if (alertDiv) {  
alertDiv.style.display = 'none';
```

```
}

}

function simulateEmergencyAlert() {
  const alerts = [
    {title: 'Weather Alert', message: 'Low rainfall warning. Manage water usage carefully.'},
    {title: 'Disease Alert', message: 'Plant disease outbreak reported in your region. Check your crops.'},
    {title: 'Price Alert', message: 'Rice prices have dropped. Check market conditions.'}
  ];

  const randomAlert = alerts[Math.floor(Math.random() * alerts.length)];
  showEmergencyAlert(randomAlert.title, randomAlert.message);
}

// Page load event
window.addEventListener('load', function() {
  addGoogleTranslateScript();
  updateCalendar();
  addVoiceListeners();
  document.getElementById('voiceToggle').addEventListener('click', toggleVoice);
  initChatbot();
  document.getElementById('pageLoader').style.display = 'none';

  const observerOptions = {
    threshold: 0.1,
    rootMargin: '0px 0px -50px 0px'
  };

  const observer = new IntersectionObserver((entries) => {
    entries.forEach(entry => {
      if (entry.isIntersecting) {
        ...
      }
    });
  });
});
```

```
entry.target.classList.add('animate-fadeInUp');

}

});

}, observerOptions);

document.querySelectorAll('section').forEach(section => {
  observer.observe(section);
});

});

document.getElementById('pageLoader').style.display = 'block';

document.querySelectorAll('.navbar a').forEach(link => {
  link.addEventListener('click', () => {
    const navbar = document.getElementById('navbar');
    if (navbar.classList.contains('active')) {
      navbar.classList.remove('active');
    }
  });
});

});

</script>

</body>
</html streamlit
```

requirements

pandas

numpy

scikit-learn

plotly

pillow

requests

streamlit

scikit-learn

numpy

pandas

pyttsx3

>