D:\Python>python app.py

\* Serving Flask app 'app'

\* Debug mode: on

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on http://127.0.0.1:5000

Press CTRL+C to quit

\* Restarting with stat

\* Debugger is active!

\* Debugger PIN: 911-824-061

Defaulted code

from flask import Flask, request, jsonify

import joblib

import numpy as np

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

# Load trained model

model = joblib.load("crop\_model.pkl")

@app.route('/')

def index():

    return "Crop Advisory System is Running!"

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

def recommend():

    data = request.get\_json()

    # Extract input values in correct order

    features = [

        data["N"], data["P"], data["K"],

        data["temperature"], data["humidity"],

        data["ph"], data["rainfall"]

    ]

    prediction = model.predict([features])[0]

    return jsonify({"recommended\_crop": prediction})

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

    app.run(debug=True)

D:\Python>python app.py

\* Serving Flask app 'app'

\* Debug mode: on

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on http://127.0.0.1:5000

Press CTRL+C to quit

\* Restarting with stat

\* Debugger is active!

\* Debugger PIN: 911-824-061

That error means the requests library isn’t installed in your Python environment yet.

No worries — you just need to install it!

**How to install requests:**

Open your terminal or command prompt and run:

pip install requests

If you have multiple versions of Python installed, you might want to be more specific:

python -m pip install requests

or

python3 -m pip install requests

**After installation:**

Run your test script again:

python test\_recommend.py

Let me know if you get any other errors or if you want help with the next steps!

<!DOCTYPE html>

<html>

<head>

    <title>Rice Crop Advisory System</title>

    <style>

        body { font-family: Arial, sans-serif; background: #f0f4f8; }

        .container { max-width: 600px; margin: 40px auto; background: #fff; padding: 30px; border-radius: 8px; box-shadow: 0 2px 8px #ccc; }

        h2, h3 { text-align: center; color: #2c5282; }

        label { display: block; margin-top: 15px; font-weight: bold; }

        input { width: 100%; padding: 8px; margin-top: 5px; border-radius: 4px; border: 1px solid #ccc; }

        button { margin-top: 20px; width: 100%; padding: 10px; background: #2c5282; color: #fff; border: none; border-radius: 4px; font-size: 16px; cursor: pointer; }

        button:hover { background: #2a4365; }

        #result { margin-top: 20px; }

        .error { color: #e53e3e; padding: 10px; border-radius: 4px; background: #fff5f5; }

        .advice { background: #f0fff4; padding: 15px; border-radius: 4px; margin-top: 10px; }

        .soil-analysis { background: #ebf8ff; padding: 15px; border-radius: 4px; margin: 10px 0; }

        ul { padding-left: 20px; }

        li { margin: 5px 0; }

    </style>

</head>

<body>

    <div class="container">

        <h2>Rice Crop Advisory System</h2>

        <form id="advisoryForm">

            <label>Nitrogen (N) in kg/ha:</label>

            <input type="number" name="N" required placeholder="Enter value (e.g., 90)">

            <label>Phosphorus (P) in kg/ha:</label>

            <input type="number" name="P" required placeholder="Enter value (e.g., 42)">

            <label>Potassium (K) in kg/ha:</label>

            <input type="number" name="K" required placeholder="Enter value (e.g., 43)">

            <label>Temperature (°C):</label>

            <input type="number" step="any" name="temperature" required placeholder="Enter value (e.g., 25)">

            <label>Humidity (%):</label>

            <input type="number" step="any" name="humidity" required placeholder="Enter value (e.g., 82)">

            <label>Soil pH:</label>

            <input type="number" step="any" name="ph" required placeholder="Enter value (e.g., 6.5)">

            <label>Rainfall (mm):</label>

            <input type="number" step="any" name="rainfall" required placeholder="Enter value (e.g., 200)">

            <button type="submit">Get Recommendation</button>

        </form>

        <div id="result"></div>

    </div>

    <script>

        document.getElementById('advisoryForm').onsubmit = async function(e) {

            e.preventDefault();

            const formData = new FormData(e.target);

            const data = Object.fromEntries(formData.entries());

            for (let key in data) {

                data[key] = Number(data[key]);

            }

            const response = await fetch('/recommend', {

                method: 'POST',

                headers: {'Content-Type': 'application/json'},

                body: JSON.stringify(data)

            });

            const result = await response.json();

            document.getElementById('result').innerHTML =

                result.error ? `<div class="error">${result.error}</div>` :

                `<div>

                    <h3>Soil Analysis Results</h3>

                    <div class="soil-analysis">

                        ${Object.values(result.soil\_analysis).join('<br>')}

                    </div>

                    <h3>Recommendations</h3>

                    <div class="advice">

                        ${result.fertilizer\_advice ? `

                            <strong>${result.fertilizer\_advice.general\_advice}</strong>

                            <ul>

                                ${result.fertilizer\_advice.npk\_recommendation.map(r => `<li>${r}</li>`).join('')}

                            </ul>

                            <ul>

                                ${result.fertilizer\_advice.application\_schedule.map(s => `<li>${s}</li>`).join('')}

                            </ul>

                            <ul>

                                ${result.fertilizer\_advice.important\_notes.map(n => `<li>${n}</li>`).join('')}

                            </ul>

                        ` : 'No specific fertilizer advice available for this crop.'}

                    </div>

                </div>`;

        };

    </script>

</body>

</html>

This below code was so far so good

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>🌾 Crop Advisory System</title>

    <link rel="icon" href="/static/images/leaf\_icon.png" type="image/png">

    <style>

        /\* General Body Styling with Background Image \*/

        body {

            font-family: 'Segoe UI', Roboto, 'Helvetica Neue', Arial, sans-serif;

            background-image: url('/static/images/background.jpg');

            background-size: cover;

            background-position: center;

            background-repeat: no-repeat;

            display: flex;

            justify-content: center;

            align-items: center;

            min-height: 100vh;

            margin: 0;

            padding: 20px;

            box-sizing: border-box;

        }

        /\* Optional overlay to improve text readability over the background \*/

        body::before {

            content: "";

            position: fixed;

            top: 0;

            left: 0;

            width: 100%;

            height: 100%;

            background-color: rgba(224, 242, 247, 0.7); /\* Light blue with some transparency \*/

            z-index: -1;

        }

        /\* Container for the form and results \*/

        .container {

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

            border-radius: 12px;

            box-shadow: 0 8px 25px rgba(0, 0, 0, 0.1);

            padding: 40px;

            width: 100%;

            max-width: 650px;

            animation: fadeIn 0.8s ease-out;

            backdrop-filter: blur(10px); /\* Optional: Adds a blur effect to the background behind the container \*/

        }

        /\* Animations \*/

        @keyframes fadeIn {

            from { opacity: 0; transform: translateY(20px); }

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

        }

        /\* Headings \*/

        h2, h3 {

            text-align: center;

            color: #2a6f7b;

            margin-bottom: 25px;

            font-weight: 600;

            letter-spacing: 0.5px;

        }

        /\* --- MINIMIZED HEADER IMAGE STYLING --- \*/

        .header-image {

            text-align: center;

            margin-bottom: 25px; /\* Space below the image \*/

            /\* Added padding/margin if needed to separate from top of container \*/

        }

        .header-image img {

            max-width: 50%;   /\* Reduced from 100% to 50% of parent container width \*/

            height: auto;

            border-radius: 8px;

            box-shadow: 0 4px 10px rgba(0,0,0,0.1);

            /\* Optional: Add a small border for more definition \*/

            /\* border: 1px solid #e0e0e0; \*/

        }

        /\* Form Layout \*/

        #advisoryForm {

            display: grid;

            grid-template-columns: 1fr 1fr;

            gap: 20px;

        }

        /\* Label and Input Styling \*/

        .form-group {

            display: flex;

            flex-direction: column;

        }

        label {

            font-weight: 600;

            color: #4a5568;

            margin-bottom: 8px;

            font-size: 0.95em;

        }

        input[type="number"],

        input[type="text"] {

            width: calc(100% - 20px);

            padding: 12px 10px;

            border: 1px solid #cbd5e0;

            border-radius: 6px;

            font-size: 1em;

            color: #2d3748;

            box-shadow: inset 0 1px 3px rgba(0,0,0,0.05);

            transition: border-color 0.2s ease, box-shadow 0.2s ease;

        }

        input[type="number"]:focus,

        input[type="text"]:focus {

            border-color: #4299e1;

            box-shadow: 0 0 0 3px rgba(66, 153, 225, 0.25);

            outline: none;

        }

        /\* Make submit button span both columns \*/

        button[type="submit"] {

            grid-column: 1 / -1;

            padding: 15px 25px;

            background-color: #2a6f7b;

            color: #ffffff;

            border: none;

            border-radius: 8px;

            font-size: 1.1em;

            font-weight: 700;

            cursor: pointer;

            transition: background-color 0.3s ease, transform 0.2s ease;

            box-shadow: 0 4px 10px rgba(42, 111, 123, 0.3);

        }

        button[type="submit"]:hover {

            background-color: #1a5666;

            transform: translateY(-2px);

        }

        button[type="submit"]:active {

            transform: translateY(0);

            box-shadow: 0 2px 5px rgba(42, 111, 123, 0.3);

        }

        /\* Result and feedback sections \*/

        #result {

            margin-top: 30px;

            padding-top: 20px;

            border-top: 1px solid #e2e8f0;

        }

        .error {

            color: #e53e3e;

            background-color: rgba(255, 245, 245, 0.85);

            padding: 15px;

            border-radius: 8px;

            text-align: center;

            font-weight: 600;

        }

        .advice, .soil-analysis {

            background-color: rgba(230, 255, 250, 0.85);

            padding: 20px;

            border-radius: 8px;

            margin-top: 15px;

            line-height: 1.6;

            font-size: 0.95em;

            border: 1px solid rgba(160, 174, 192, 0.6);

        }

        .soil-analysis {

            margin: 15px 0;

            background-color: rgba(235, 248, 255, 0.85);

            border: 1px solid rgba(160, 174, 192, 0.6);

        }

        .soil-analysis strong {

            color: #2a6f7b;

        }

        ul {

            list-style-type: disc;

            padding-left: 25px;

            margin-top: 10px;

        }

        li {

            margin: 8px 0;

            color: #4a5568;

        }

        /\* Responsive adjustments for smaller screens \*/

        @media (max-width: 768px) {

            #advisoryForm {

                grid-template-columns: 1fr;

            }

            .container {

                padding: 25px;

                margin: 20px;

            }

            .header-image img {

                max-width: 70%; /\* Adjust for smaller screens if needed \*/

            }

        }

    </style>

</head>

<body>

    <div class="container">

        <div class="header-image">

            <img src="/static/images/header.jpg" alt="Crop Advisory Header">

        </div>

        <h2>🌾 Rice Crop Advisory System</h2>

        <form id="advisoryForm">

            <div class="form-group">

                <label for="N">Nitrogen (N) in kg/ha:</label>

                <input type="number" id="N" name="N" required placeholder="e.g., 90">

            </div>

            <div class="form-group">

                <label for="P">Phosphorus (P) in kg/ha:</label>

                <input type="number" id="P" name="P" required placeholder="e.g., 42">

            </div>

            <div class="form-group">

                <label for="K">Potassium (K) in kg/ha:</label>

                <input type="number" id="K" name="K" required placeholder="e.g., 43">

            </div>

            <div class="form-group">

                <label for="temperature">Temperature (°C):</label>

                <input type="number" step="any" id="temperature" name="temperature" required placeholder="e.g., 25.5">

            </div>

            <div class="form-group">

                <label for="humidity">Humidity (%):</label>

                <input type="number" step="any" id="humidity" name="humidity" required placeholder="e.g., 82.0">

            </div>

            <div class="form-group">

                <label for="ph">Soil pH:</label>

                <input type="number" step="any" id="ph" name="ph" required placeholder="e.g., 6.5">

            </div>

            <div class="form-group">

                <label for="rainfall">Rainfall (mm):</label>

                <input type="number" step="any" id="rainfall" name="rainfall" required placeholder="e.g., 200.0">

            </div>

            <button type="submit">🌱 Get Recommendation</button>

        </form>

        <div id="result"></div>

    </div>

    <script>

        document.getElementById('advisoryForm').onsubmit = async function(e) {

            e.preventDefault();

            const form = e.target;

            const formData = new FormData(form);

            const data = Object.fromEntries(formData.entries());

            for (let key in data) {

                data[key] = Number(data[key]);

            }

            const resultDiv = document.getElementById('result');

            resultDiv.innerHTML = '<div style="text-align: center; color: #4a5568;">Fetching recommendation...</div>';

            try {

                const response = await fetch('/recommend', {

                    method: 'POST',

                    headers: {'Content-Type': 'application/json'},

                    body: JSON.stringify(data)

                });

                if (!response.ok) {

                    const errorData = await response.json();

                    resultDiv.innerHTML = `<div class="error">Error: ${errorData.error || 'Something went wrong on the server.'}</div>`;

                    return;

                }

                const result = await response.json();

                if (result.error) {

                    resultDiv.innerHTML = `<div class="error">${result.error}</div>`;

                } else {

                    const recommendedCrop = result.recommended\_crop || 'N/A';

                    const soilAnalysisHtml = result.soil\_analysis ?

                        `<div class="soil-analysis">

                            <h4>Soil Characteristics:</h4>

                            <ul>

                                ${Object.keys(result.soil\_analysis).map(key => `<li><strong>${key.replace('\_', ' ')}:</strong> ${result.soil\_analysis[key]}</li>`).join('')}

                            </ul>

                        </div>` : '';

                    const fertilizerAdviceHtml = result.fertilizer\_advice ?

                        `<div class="advice">

                            <h4>Fertilizer Advisory:</h4>

                            <p><strong>${result.fertilizer\_advice.general\_advice || 'General advice missing.'}</strong></p>

                            ${result.fertilizer\_advice.npk\_recommendation ? `<h5>N-P-K Recommendation:</h5><ul>${result.fertilizer\_advice.npk\_recommendation.map(r => `<li>${r}</li>`).join('')}</ul>` : ''}

                            ${result.fertilizer\_advice.application\_schedule ? `<h5>Application Schedule:</h5><ul>${result.fertilizer\_advice.application\_schedule.map(s => `<li>${s}</li>`).join('')}</ul>` : ''}

                            ${result.fertilizer\_advice.important\_notes ? `<h5>Important Notes:</h5><ul>${result.fertilizer\_advice.important\_notes.map(n => `<li>${n}</li>`).join('')}</ul>` : ''}

                        </div>` : '';

                    resultDiv.innerHTML = `

                        <div style="text-align: center; margin-bottom: 25px;">

                            <h3>Recommended Crop: <span style="color: #007bff; font-weight: 700;">${recommendedCrop}</span></h3>

                        </div>

                        ${soilAnalysisHtml}

                        ${fertilizerAdviceHtml}

                        <div class="advice" style="margin-top: 25px;">

                            <h4>General Crop Care Tip:</h4>

                            <p>Ensure proper drainage and monitor for early signs of pests. Regular soil testing helps maintain optimal conditions.</p>

                        </div>

                    `;

                }

            } catch (error) {

                console.error('Fetch error:', error);

                resultDiv.innerHTML = `<div class="error">Could not connect to the server or parse response. Please try again.</div>`;

            }

        };

    </script>

</body>

</html>

This one below is nice

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>🌾 Crop Advisory System</title>

    <link rel="icon" href="/static/images/leaf\_icon.png" type="image/png">

    <style>

        /\* General Body Styling with Background Image \*/

        body {

            font-family: 'Segoe UI', Roboto, 'Helvetica Neue', Arial, sans-serif;

            background-image: url('/static/images/background.jpg');

            background-size: cover;

            background-position: center;

            background-repeat: no-repeat;

            display: flex;

            justify-content: center;

            align-items: center;

            min-height: 100vh;

            margin: 0;

            padding: 20px;

            box-sizing: border-box;

        }

        /\* Optional overlay to improve text readability over the background \*/

        body::before {

            content: "";

            position: fixed;

            top: 0;

            left: 0;

            width: 100%;

            height: 100%;

            background-color: rgba(224, 242, 247, 0.7); /\* Light blue with some transparency \*/

            z-index: -1;

        }

        /\* Container for the form and results \*/

        .container {

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

            border-radius: 12px;

            box-shadow: 0 8px 25px rgba(0, 0, 0, 0.1);

            padding: 40px;

            width: 100%;

            max-width: 650px;

            animation: fadeIn 0.8s ease-out;

            backdrop-filter: blur(10px); /\* Optional: Adds a blur effect to the background behind the container \*/

            overflow: hidden; /\* Ensures image doesn't break rounded corners \*/

        }

        /\* Animations \*/

        @keyframes fadeIn {

            from { opacity: 0; transform: translateY(20px); }

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

        }

        /\* Headings \*/

        h2, h3 {

            text-align: center;

            color: #2a6f7b;

            margin-bottom: 25px;

            font-weight: 600;

            letter-spacing: 0.5px;

        }

        /\* --- FULL HORIZONTAL HEADER IMAGE STYLING --- \*/

        .header-image {

            /\* This div itself doesn't need much styling if img takes 100% width \*/

            text-align: center; /\* Ensures the image is centered if it doesn't take full width (e.g., if you give it max-width < 100%) \*/

            margin: -40px -40px 25px -40px; /\* Pulls image to container edges, then adds bottom margin \*/

            border-top-left-radius: 12px;

            border-top-right-radius: 12px;

            overflow: hidden; /\* Ensures image corners match container \*/

            box-shadow: 0 2px 10px rgba(0,0,0,0.1); /\* Subtle shadow for the image block \*/

        }

        .header-image img {

            width: 100%;        /\* Make image take full width of its parent (.header-image div) \*/

            height: 180px;      /\* Set a fixed height for a prominent horizontal band. ADJUST THIS! \*/

            object-fit: cover;  /\* Ensures the image covers the defined height, cropping as needed \*/

            display: block; /\* Removes extra space below image \*/

        }

        /\* Form Layout \*/

        #advisoryForm {

            display: grid;

            grid-template-columns: 1fr 1fr;

            gap: 20px;

        }

        /\* Label and Input Styling \*/

        .form-group {

            display: flex;

            flex-direction: column;

        }

        label {

            font-weight: 600;

            color: #4a5568;

            margin-bottom: 8px;

            font-size: 0.95em;

        }

        input[type="number"],

        input[type="text"] {

            width: calc(100% - 20px);

            padding: 12px 10px;

            border: 1px solid #cbd5e0;

            border-radius: 6px;

            font-size: 1em;

            color: #2d3748;

            box-shadow: inset 0 1px 3px rgba(0,0,0,0.05);

            transition: border-color 0.2s ease, box-shadow 0.2s ease;

        }

        input[type="number"]:focus,

        input[type="text"]:focus {

            border-color: #4299e1;

            box-shadow: 0 0 0 3px rgba(66, 153, 225, 0.25);

            outline: none;

        }

        /\* Make submit button span both columns \*/

        button[type="submit"] {

            grid-column: 1 / -1;

            padding: 15px 25px;

            background-color: #2a6f7b;

            color: #ffffff;

            border: none;

            border-radius: 8px;

            font-size: 1.1em;

            font-weight: 700;

            cursor: pointer;

            transition: background-color 0.3s ease, transform 0.2s ease;

            box-shadow: 0 4px 10px rgba(42, 111, 123, 0.3);

        }

        button[type="submit"]:hover {

            background-color: #1a5666;

            transform: translateY(-2px);

        }

        button[type="submit"]:active {

            transform: translateY(0);

            box-shadow: 0 2px 5px rgba(42, 111, 123, 0.3);

        }

        /\* Result and feedback sections \*/

        #result {

            margin-top: 30px;

            padding-top: 20px;

            border-top: 1px solid #e2e8f0;

        }

        .error {

            color: #e53e3e;

            background-color: rgba(255, 245, 245, 0.85);

            padding: 15px;

            border-radius: 8px;

            text-align: center;

            font-weight: 600;

        }

        .advice, .soil-analysis {

            background-color: rgba(230, 255, 250, 0.85);

            padding: 20px;

            border-radius: 8px;

            margin-top: 15px;

            line-height: 1.6;

            font-size: 0.95em;

            border: 1px solid rgba(160, 174, 192, 0.6);

        }

        .soil-analysis {

            margin: 15px 0;

            background-color: rgba(235, 248, 255, 0.85);

            border: 1px solid rgba(160, 174, 192, 0.6);

        }

        .soil-analysis strong {

            color: #2a6f7b;

        }

        ul {

            list-style-type: disc;

            padding-left: 25px;

            margin-top: 10px;

        }

        li {

            margin: 8px 0;

            color: #4a5568;

        }

        /\* Responsive adjustments for smaller screens \*/

        @media (max-width: 768px) {

            #advisoryForm {

                grid-template-columns: 1fr;

            }

            .container {

                padding: 25px;

                margin: 20px;

            }

            .header-image {

                margin: -25px -25px 20px -25px; /\* Adjust margin for smaller padding \*/

            }

            .header-image img {

                height: 120px; /\* Adjust height for smaller screens \*/

            }

        }

    </style>

</head>

<body>

    <div class="container">

        <div class="header-image">

            <img src="/static/images/header.jpg" alt="Crop Advisory Header">

        </div>

        <h2>🌾 Rice Crop Advisory System</h2>

        <form id="advisoryForm">

            <div class="form-group">

                <label for="N">Nitrogen (N) in kg/ha:</label>

                <input type="number" id="N" name="N" required placeholder="e.g., 90">

            </div>

            <div class="form-group">

                <label for="P">Phosphorus (P) in kg/ha:</label>

                <input type="number" id="P" name="P" required placeholder="e.g., 42">

            </div>

            <div class="form-group">

                <label for="K">Potassium (K) in kg/ha:</label>

                <input type="number" id="K" name="K" required placeholder="e.g., 43">

            </div>

            <div class="form-group">

                <label for="temperature">Temperature (°C):</label>

                <input type="number" step="any" id="temperature" name="temperature" required placeholder="e.g., 25.5">

            </div>

            <div class="form-group">

                <label for="humidity">Humidity (%):</label>

                <input type="number" step="any" id="humidity" name="humidity" required placeholder="e.g., 82.0">

            </div>

            <div class="form-group">

                <label for="ph">Soil pH:</label>

                <input type="number" step="any" id="ph" name="ph" required placeholder="e.g., 6.5">

            </div>

            <div class="form-group">

                <label for="rainfall">Rainfall (mm):</label>

                <input type="number" step="any" id="rainfall" name="rainfall" required placeholder="e.g., 200.0">

            </div>

            <button type="submit">🌱 Get Recommendation</button>

        </form>

        <div id="result"></div>

    </div>

    <script>

        document.getElementById('advisoryForm').onsubmit = async function(e) {

            e.preventDefault();

            const form = e.target;

            const formData = new FormData(form);

            const data = Object.fromEntries(formData.entries());

            for (let key in data) {

                data[key] = Number(data[key]);

            }

            const resultDiv = document.getElementById('result');

            resultDiv.innerHTML = '<div style="text-align: center; color: #4a5568;">Fetching recommendation...</div>';

            try {

                const response = await fetch('/recommend', {

                    method: 'POST',

                    headers: {'Content-Type': 'application/json'},

                    body: JSON.stringify(data)

                });

                if (!response.ok) {

                    const errorData = await response.json();

                    resultDiv.innerHTML = `<div class="error">Error: ${errorData.error || 'Something went wrong on the server.'}</div>`;

                    return;

                }

                const result = await response.json();

                if (result.error) {

                    resultDiv.innerHTML = `<div class="error">${result.error}</div>`;

                } else {

                    const recommendedCrop = result.recommended\_crop || 'N/A';

                    const soilAnalysisHtml = result.soil\_analysis ?

                        `<div class="soil-analysis">

                            <h4>Soil Characteristics:</h4>

                            <ul>

                                ${Object.keys(result.soil\_analysis).map(key => `<li><strong>${key.replace('\_', ' ')}:</strong> ${result.soil\_analysis[key]}</li>`).join('')}

                            </ul>

                        </div>` : '';

                    const fertilizerAdviceHtml = result.fertilizer\_advice ?

                        `<div class="advice">

                            <h4>Fertilizer Advisory:</h4>

                            <p><strong>${result.fertilizer\_advice.general\_advice || 'General advice missing.'}</strong></p>

                            ${result.fertilizer\_advice.npk\_recommendation ? `<h5>N-P-K Recommendation:</h5><ul>${result.fertilizer\_advice.npk\_recommendation.map(r => `<li>${r}</li>`).join('')}</ul>` : ''}

                            ${result.fertilizer\_advice.application\_schedule ? `<h5>Application Schedule:</h5><ul>${result.fertilizer\_advice.application\_schedule.map(s => `<li>${s}</li>`).join('')}</ul>` : ''}

                            ${result.fertilizer\_advice.important\_notes ? `<h5>Important Notes:</h5><ul>${result.fertilizer\_advice.important\_notes.map(n => `<li>${n}</li>`).join('')}</ul>` : ''}

                        </div>` : '';

                    resultDiv.innerHTML = `

                        <div style="text-align: center; margin-bottom: 25px;">

                            <h3>Recommended Crop: <span style="color: #007bff; font-weight: 700;">${recommendedCrop}</span></h3>

                        </div>

                        ${soilAnalysisHtml}

                        ${fertilizerAdviceHtml}

                        <div class="advice" style="margin-top: 25px;">

                            <h4>General Crop Care Tip:</h4>

                            <p>Ensure proper drainage and monitor for early signs of pests. Regular soil testing helps maintain optimal conditions.</p>

                        </div>

                    `;

                }

            } catch (error) {

                console.error('Fetch error:', error);

                resultDiv.innerHTML = `<div class="error">Could not connect to the server or parse response. Please try again.</div>`;

            }

        };

    </script>

</body>

</html>