Codes till now (working)

Backend  
from flask import Flask, request, jsonify

import pytesseract

from PIL import Image

import os

from flask\_cors import CORS  *# Import and enable CORS*

pytesseract.pytesseract.tesseract\_cmd = r'C:\\Program Files\\Tesseract-OCR\\tesseract.exe'

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

CORS(app)  *# Enable CORS*

@app.route('/api/image-to-text', methods=['POST'])

def image\_to\_text():

    if 'image' not in request.files:

        return jsonify({'error': 'No file uploaded'}), 400

    file = request.files['image']

    file\_path = os.path.join('temp', file.filename)

    os.makedirs('temp', exist\_ok=True)

    file.save(file\_path)

    try:

        image = Image.open(file\_path)

        text = pytesseract.image\_to\_string(image)

        os.remove(file\_path)

        return jsonify({'text': text}), 200

    except Exception as e:

        return jsonify({'error': str(e)}), 500

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

frontend

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Image to Text Converter</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            text-align: center;

            margin: 0;

            padding: 0;

            background-color: #f4f4f4;

        }

        header {

            background: #007BFF;

            color: white;

            padding: 1em 0;

        }

        .container {

            margin: 20px auto;

            max-width: 600px;

            padding: 20px;

            background: white;

            border-radius: 8px;

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

        }

        .drop-area {

            margin: 10px 0;

            padding: 20px;

            border: 2px dashed #007BFF;

            border-radius: 8px;

            background-color: #f9f9f9;

            color: #333;

            cursor: pointer;

        }

        .drop-area.dragover {

            border-color: #0056b3;

            background-color: #e0f7ff;

        }

        .drop-area span {

            color: #007BFF;

            text-decoration: underline;

            cursor: pointer;

        }

        button {

            padding: 10px 20px;

            background: #007BFF;

            color: white;

            border: none;

            border-radius: 5px;

            cursor: pointer;

        }

        button:hover {

            background: #0056b3;

        }

        #result {

            margin-top: 20px;

            white-space: pre-wrap;

            word-wrap: break-word;

            color: #333;

        }

        #preview {

            margin-top: 20px;

            max-width: 100%;

            max-height: 300px;

            border: 1px solid #ccc;

            border-radius: 8px;

        }

        #copyButton {

            margin-top: 10px;

            display: none;

            background-color: #28a745;

        }

        #copyButton:hover {

            background-color: #218838;

        }

    </style>

</head>

<body>

    <header>

        <h1>Image to Text Converter</h1>

    </header>

    <div class="container">

        <form id="uploadForm">

            <div id="dropArea" class="drop-area">

                <p>Drag & Drop your image here, or <span>browse</span>.</p>

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

            </div>

            <button type="submit">Convert to Text</button>

        </form>

        <img id="preview" src="" alt="Image Preview" style="display: none;">

        <div id="result"></div>

        <button id="copyButton">Copy Text</button>

    </div>

    <script>

        const dropArea = document.getElementById('dropArea');

        const fileInput = document.getElementById('image');

        const preview = document.getElementById('preview');

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

        const copyButton = document.getElementById('copyButton');

*// Highlight drop area when file is dragged over*

        ['dragenter', 'dragover'].forEach(eventName => {

            dropArea.addEventListener(eventName, (e) => {

                e.preventDefault();

                e.stopPropagation();

                dropArea.classList.add('dragover');

            });

        });

*// Remove highlight when file is dragged out*

        ['dragleave', 'drop'].forEach(eventName => {

            dropArea.addEventListener(eventName, (e) => {

                e.preventDefault();

                e.stopPropagation();

                dropArea.classList.remove('dragover');

            });

        });

*// Handle file drop*

        dropArea.addEventListener('drop', (e) => {

            e.preventDefault();

            e.stopPropagation();

            const files = e.dataTransfer.files;

            if (files.length > 0) {

                fileInput.files = files; *// Set the dropped file to the input*

                showPreview(files[0]);  *// Show image preview*

            }

        });

*// Allow click to trigger file input*

        dropArea.addEventListener('click', () => {

            fileInput.click();

        });

*// Show preview when file is selected*

        fileInput.addEventListener('change', () => {

            if (fileInput.files.length > 0) {

                showPreview(fileInput.files[0]);

            }

        });

*// Function to show image preview*

        function showPreview(file) {

            const reader = new FileReader();

            reader.onload = (e) => {

                preview.src = e.target.result;

                preview.style.display = 'block';

            };

            reader.readAsDataURL(file);

        }

        document.getElementById('uploadForm').addEventListener('submit', async (e) => {

            e.preventDefault();

            const formData = new FormData();

            if (fileInput.files.length === 0) {

                alert('Please select an image file.');

                return;

            }

            formData.append('image', fileInput.files[0]);

            try {

                const response = await fetch('http://127.0.0.1:5000/api/image-to-text', {

                    method: 'POST',

                    body: formData

                });

                const result = await response.json();

                if (response.ok) {

                    resultDiv.textContent = result.text;

                    copyButton.style.display = 'block'; *// Show the copy button*

                } else {

                    resultDiv.textContent = `Error: ${result.error}`;

                    copyButton.style.display = 'none'; *// Hide the copy button*

                }

            } catch (err) {

                resultDiv.textContent = `Error: ${err.message}`;

                copyButton.style.display = 'none'; *// Hide the copy button*

            }

        });

*// Copy to Clipboard Functionality*

        copyButton.addEventListener('click', () => {

            const text = resultDiv.textContent;

            if (text) {

                navigator.clipboard.writeText(text).then(() => {

                    alert('Text copied to clipboard!');

                }).catch(err => {

                    alert('Failed to copy text: ' + err);

                });

            }

        });

    </script>

</body>

</html>