

```
import os
import logging
from flask import Flask, render_template, request, redirect, url_for, flash, jsonify, session
from werkzeug.utils import secure_filename
import uuid
from document_processor import DocumentProcessor

# Configure logging
logging.basicConfig(level=logging.DEBUG)
logger = logging.getLogger(__name__)

# Initialize Flask app
app = Flask(__name__)
app.secret_key = os.environ.get("SESSION_SECRET", "default_secret_key_for_development")
app.config['MAX_CONTENT_LENGTH'] = 16 * 1024 * 1024 # 16MB max upload size
app.config['UPLOAD_FOLDER'] = '/tmp/solar_docs'

# Create upload folder if it doesn't exist
os.makedirs(app.config['UPLOAD_FOLDER'], exist_ok=True)

# Initialize document processor
document_processor = DocumentProcessor()

# Allowed file extensions
ALLOWED_EXTENSIONS = {'pdf', 'png', 'jpg', 'jpeg', 'tiff', 'tif'}

def allowed_file(filename):
    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/upload', methods=['POST'])
def upload_file():
    if 'document' not in request.files:
        flash('No file part', 'danger')
        return redirect(url_for('index'))

    file = request.files['document']

    if file.filename == '':
        flash('No file selected', 'danger')
        return redirect(url_for('index'))

    if not allowed_file(file.filename):
        flash('Invalid file type. Please upload PDF, PNG, JPG, JPEG, or TIFF files.', 'danger')
        return redirect(url_for('index'))

    try:
        # Create a unique filename
        filename = secure_filename(file.filename)
        unique_filename = f"{uuid.uuid4()}_{filename}"
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], unique_filename)

        # Save the file
        file.save(file_path)
        logger.debug(f"File saved to {file_path}")

        # Process the document
        result = document_processor.process_document(file_path)

        # Store result in session for display
        session['processing_result'] = result

        # Clean up - remove the file after processing
        os.remove(file_path)
```

```
        return redirect(url_for('results'))

    except Exception as e:
        logger.error(f"Error processing document: {str(e)}", exc_info=True)
        flash(f'Error processing document: {str(e)}', 'danger')
        return redirect(url_for('index'))

@app.route('/results')
def results():
    result = session.get('processing_result')
    if not result:
        flash('No processing results found', 'warning')
        return redirect(url_for('index'))

    return render_template('results.html', result=result)

@app.route('/api/process', methods=['POST'])
def api_process():
    if 'document' not in request.files:
        return jsonify({'error': 'No file part'}), 400

    file = request.files['document']

    if file.filename == '':
        return jsonify({'error': 'No file selected'}), 400

    if not allowed_file(file.filename):
        return jsonify({'error': 'Invalid file type'}), 400

    try:
        # Create a unique filename
        filename = secure_filename(file.filename)
        unique_filename = f"{uuid.uuid4()}_{filename}"
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], unique_filename)

        # Save the file
        file.save(file_path)

        # Process the document
        result = document_processor.process_document(file_path)

        # Clean up - remove the file after processing
        os.remove(file_path)

        return jsonify(result)

    except Exception as e:
        logger.error(f"API Error: {str(e)}", exc_info=True)
        return jsonify({'error': str(e)}), 500

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```