# IMAGE CONVERTER USING FLASK

**Line-by-Line Explanation and Requirements:**

**1. Installation Requirements**

Before running this code, install the following packages:

* Flask: pip3 install flask
* OpenCV: pip3 install opencv-python

Additionally, ensure Python 3.x is installed.

**2. Code Breakdown**

**Import Statements**

python

Copy code

from flask import Flask, render\_template, request, flash

from werkzeug.utils import secure\_filename

import cv2

import os

* **Flask**: Used for creating a web application.
* **render\_template**: To render HTML templates (index.html, about.html).
* **request**: To handle HTTP requests (GET/POST).
* **flash**: For displaying messages to the user.
* **secure\_filename**: Sanitizes file names for security.
* **OpenCV (cv2)**: Processes images for operations like grayscale conversion.
* **os**: Manages file paths and directories.

**Configuration**

python

Copy code

UPLOAD\_FOLDER = 'uploads'

PROCESSED\_FOLDER = 'static'

ALLOWED\_EXTENSIONS = {'png', 'webp', 'jpg', 'jpeg', 'gif'}

* **UPLOAD\_FOLDER**: Directory where uploaded images will be saved.
* **PROCESSED\_FOLDER**: Directory for storing processed images.
* **ALLOWED\_EXTENSIONS**: File types allowed for upload.

**Flask Application Setup**

python

Copy code

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

app.secret\_key = 'super secret key'

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

app.config['PROCESSED\_FOLDER'] = PROCESSED\_FOLDER

* **Flask Instance**: app initializes the web application.
* **Secret Key**: Enables secure sessions and message flashing.
* **Config**: Sets upload and processing folder paths.

**Ensure Folders Exist**

python

Copy code

os.makedirs(UPLOAD\_FOLDER, exist\_ok=True)

os.makedirs(PROCESSED\_FOLDER, exist\_ok=True)

* Ensures that required directories (uploads and static) exist, creating them if necessary.

**Allowed File Check**

python

Copy code

def allowed\_file(filename):

return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

* Checks if the uploaded file has an extension listed in ALLOWED\_EXTENSIONS.

**Image Processing Function**

python

Copy code

def processImage(filename, operation):

img\_path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

img = cv2.imread(img\_path)

new\_filename = None

match operation:

case "cgray":

img\_processed = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

new\_filename = os.path.join(app.config['PROCESSED\_FOLDER'], f"{filename.rsplit('.', 1)[0]}\_gray.jpg")

cv2.imwrite(new\_filename, img\_processed)

* **Function Purpose**: Handles image processing based on the operation type.
* **Supported Operations**:
  + cgray: Converts the image to grayscale.
  + cwebp, cjpg, cpng: Converts to specific formats (WEBP, JPG, PNG).
* **Match-Case**: Determines the operation type and applies corresponding OpenCV transformations.
* **File Writing**: Saves processed images in the static directory.

**Route: Home Page**

python

Copy code

@app.route("/")

def home():

return render\_template("index.html")

* **Purpose**: Displays the homepage with the image upload form.

**Route: About Page**

python

Copy code

@app.route("/about")

def about():

return render\_template("about.html")

* **Purpose**: Displays the "About" page with project information.

**Route: Edit Image**

python

Copy code

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

def edit():

if request.method == "POST":

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

if 'file' not in request.files:

flash('No file part in the request.')

return render\_template("index.html")

* **Purpose**: Handles file uploads and image processing.
* **Key Steps**:
  1. **Check for File**: Ensures a file is uploaded.
  2. **Validate File Type**: Ensures the file extension is allowed.
  3. **Save File**: Stores the file in the uploads directory.
  4. **Process Image**: Calls processImage() for the selected operation.
  5. **Flash Message**: Displays a success or error message.

**Running the App**

python

Copy code

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

app.run(debug=True, port=5001)

* **Debug Mode**: Provides detailed error logs for development.
* **Port**: Runs the application on http://127.0.0.1:5001.

**3. Features**

* **Image Upload**: Supports common formats (PNG, JPG, GIF, WEBP).
* **Operations**:
  1. Grayscale conversion.
  2. Format conversions (to WEBP, JPG, PNG).
* **Dynamic Flash Messages**: Informs users about processing status.
* **Simple Routing**: Two main pages: Home and About.

**4. Requirements**

1. **Python**: Version 3.x.
2. **Flask**: For backend development.
3. **OpenCV**: For image processing.
4. **HTML Templates**: index.html and about.html for UI.
5. **Directories**:
   * uploads: To store uploaded images.
   * static: To save processed images.
6. **Bootstrap (Optional)**: For styling the UI.

**5. Deployment Steps**

1. **Install Required Libraries**:

bash

Copy code

pip install flask opencv-python

1. **Run the Application**:

bash

Copy code

python app.py

1. **Access the Application**:
   * Open a browser and navigate to http://127.0.0.1:5001.

**6. Future Improvements**

* **Error Handling**: Add detailed error messages for file upload and processing failures.
* **Multiple File Uploads**: Support batch processing.
* **Preview Feature**: Allow users to preview processed images before saving.
* **Styling**: Improve UI with advanced CSS or Bootstrap features

**FRONTEND -index.html**.:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Image Converter</title>

    <!-- Bootstrap CSS -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom Styles -->

    <style>

        body {

            background-color: #f8f9fa;

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

        }

        .navbar {

            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

        }

        .container {

            margin-top: 50px;

        }

        .card {

            border: none;

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

        }

        .btn-custom {

            background-color: #007bff;

            color: #fff;

            border-radius: 20px;

            transition: background-color 0.3s ease;

        }

        .btn-custom:hover {

            background-color: #0056b3;

        }

        .logo-container {

            text-align: center;

            margin-bottom: 20px;

        }

        .logo-container img {

            max-width: 80px;

            height: auto;

        }

        footer {

            margin-top: 50px;

            background-color: #343a40;

            color: white;

            padding: 15px 0;

            text-align: center;

        }

    </style>

</head>

<body>

    <!-- Navigation Bar -->

    <nav class="navbar navbar-expand-lg navbar-light bg-light">

        <div class="container">

            <a class="navbar-brand" href="/">Image Converter</a>

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

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

            </button>

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

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

                    <li class="nav-item">

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

                    </li>

                    <li class="nav-item">

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

                    </li>

                </ul>

            </div>

        </div>

    </nav>

    <!-- Main Content -->

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-md-6">

                <!-- Logo Section -->

                <div class="logo-container">

                    <img src="http://127.0.0.1:5001/static/download.jpg" alt="Image Editor Logo">

                    <h3 class="mt-2">Image Converter</h3>

                </div>

                <!-- Form Section -->

                <div class="card">

                    <div class="card-body">

                        <h3 class="card-title text-center mb-4">Upload and Edit Your Image</h3>

                        <form action="/edit" method="POST" enctype="multipart/form-data">

                            <div class="mb-3">

                                <label for="file" class="form-label">Choose an image</label>

                                <input class="form-control" type="file" id="file" name="file" accept=".jpg,.jpeg,.png,.gif,.webp" required>

                            </div>

                            <div class="mb-3">

                                <label for="operation" class="form-label">Select an operation</label>

                                <select class="form-select" id="operation" name="operation" required>

                                    <option value="cgray">Convert to Grayscale</option>

                                    <option value="cwebp">Convert to WEBP</option>

                                    <option value="cjpg">Convert to JPG</option>

                                    <option value="cpng">Convert to PNG</option>

                                </select>

                            </div>

                            <div class="text-center">

                                <button type="submit" class="btn btn-custom w-100">Process Image</button>

                            </div>

                        </form>

                    </div>

                </div>

                {% with messages = get\_flashed\_messages() %}

                {% if messages %}

                    <div class="mt-4 alert alert-info" role="alert">

                        {% for message in messages %}

                            {{ message|safe }}

                        {% endfor %}

                    </div>

                {% endif %}

                {% endwith %}

            </div>

        </div>

    </div>

    <!-- Footer -->

    <footer>

        <p>&copy; 2024 Image Converter Pranav VP. All rights reserved.</p>

    </footer>

    <!-- Bootstrap JS -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**about.html:**

<!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>About Us - Image Editor</title>

    <!-- Bootstrap CSS -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom Styles -->

    <style>

        body {

            background-color: #f8f9fa;

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

        }

        .navbar {

            box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

        }

        .container {

            margin-top: 50px;

        }

        .card {

            border: none;

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

        }

        .btn-custom {

            background-color: #007bff;

            color: #fff;

            border-radius: 20px;

            transition: background-color 0.3s ease;

        }

        .btn-custom:hover {

            background-color: #0056b3;

        }

        footer {

            margin-top: 50px;

            background-color: #343a40;

            color: white;

            padding: 15px 0;

            text-align: center;

        }

    </style>

</head>

<body>

    <!-- Navigation Bar -->

    <nav class="navbar navbar-expand-lg navbar-light bg-light">

        <div class="container">

            <a class="navbar-brand" href="/">Image Editor</a>

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

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

            </button>

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

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

                    <li class="nav-item">

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

                    </li>

                    <li class="nav-item">

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

                    </li>

                </ul>

            </div>

        </div>

    </nav>

    <!-- About Section -->

    <div class="container">

        <div class="row justify-content-center">

            <div class="col-md-8">

                <div class="card">

                    <div class="card-body">

                        <h3 class="card-title text-center mb-4">About Image Editor</h3>

                        <p class="card-text">

                            Welcome to the Image Converter created by Pranav VP, your one-stop solution for quick and easy image editing! This platform provides you with tools to transform and enhance your images effortlessly.

                        </p>

                        <p class="card-text">

                            Our features include:

                            <ul>

                                <li>Grayscale conversion</li>

                                <li>Format conversion (to PNG, JPG, WEBP)</li>

                                <li>And much more!</li>

                            </ul>

                        </p>

                        <p class="card-text">

                            Built with modern technologies like <strong>Flask</strong> and <strong>OpenCV</strong>, we ensure reliability, speed, and quality for all your image processing needs.

                        </p>

                        <p class="card-text text-center mt-4">

                            <a href="/" class="btn btn-custom">Go Back to Home</a>

                        </p>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <!-- Footer -->

    <footer>

        <p>&copy; 2024 Image Editor. All rights reserved.</p>

    </footer>

    <!-- Bootstrap JS -->

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**For execution:**

Run Command python main.py in terminal.

**Conclusion**

This Flask-based Image Editor application allows users to upload images and perform basic image processing operations, such as converting images to grayscale or different formats (WEBP, JPG, PNG). By leveraging Flask for the web framework and OpenCV for image processing, the application offers a simple yet powerful way for users to manipulate images online.

Key features of the application include:

* File Upload and Validation: Ensures only supported image formats (PNG, JPG, GIF, WEBP) are processed.
* Flexible Image Processing: Supports multiple operations such as grayscale conversion and format changes.
* Flash Messages: Provides real-time feedback to users on the success or failure of their image processing tasks.
* Simple Navigation: Easy-to-use interface with home and about pages.

The application can be extended with additional features, such as:

* Batch processing for multiple files.
* Image previewing before processing.
* Improved error handling to guide users during upload and processing.

In conclusion, this project serves as a strong foundation for building more advanced image editing web applications. With Flask and OpenCV, it can be further enhanced to meet more specific needs or to integrate with additional image editing functionalities.