**OCR Flask App - Project Description**

* Project Title: OCR Flask App - Extract Text from Images Using OCR Technology
* Overview:

This project is a web-based Optical Character Recognition (OCR) application built using Python’s Flask framework. It allows users to upload images containing text and extracts the text content from these images using Tesseract OCR engine.

* Purpose:

The app serves as a simple and accessible tool for converting printed or handwritten text in images into editable and searchable digital text. This can be useful for digitizing documents, automating data entry, accessibility purposes, and more.

* Key Features:

1. Upload image files (JPEG, PNG, etc.) via a web interface
2. Convert uploaded images to grayscale for better OCR accuracy
3. Use the Tesseract OCR engine to extract text from images
4. Display extracted text on the webpage in a read-only text area
5. Simple and clean user interface built with basic HTML and CSS (no external UI frameworks)

* Technology Stack:

1. Backend: Python, Flask
2. OCR Engine: Tesseract OCR via pytesseract library
3. Image Processing: PIL (Pillow), OpenCV (cv2)
4. Frontend: HTML, CSS, Jinja2 templating (Flask)
5. Deployment: Can be deployed on any platform supporting Flask apps (Heroku, Railway, local server, etc.)

* How it Works:
* User uploads an image file via the web form.
* The Flask backend receives the image file and converts it to a grayscale image using OpenCV for better OCR results.
* The grayscale image is passed to the Tesseract OCR engine through pytesseract.
* Extracted text is returned and displayed on the web page for the user to copy or use.
* Project Structure:
* app.py – Main Flask application handling routes and OCR processing
* templates/index.html – Frontend HTML template for the user interface
* requirements.txt – List of Python dependencies to install
* ocr.py – Optional standalone script to perform OCR on a local image file
* Other files/folders – Virtual environment, git files, etc.
* Usage Instructions:
* Clone the repository.
* Create and activate a Python virtual environment.
* Install dependencies: pip install -r requirements.txt
* Run the Flask app: python app.py
* Open http://127.0.0.1:5000 in your web browser.
* Upload an image and get the extracted text instantly.
* Future Improvements:
* Add support for multiple image upload and batch OCR processing.
* Improve UI with better design or frameworks like Bootstrap or Tailwind CSS (optional).
* Add language selection and OCR configurations.
* Save extracted text files or export options.
* Deploy on cloud platforms with persistent storage.