# VisionText AI - Project Documentation

## 1. Introduction

VisionText AI is a lightweight Flask web application designed to extract text from images using Optical Character Recognition (OCR), process it, and allow users to query ChatGPT based on the extracted content. It features image and camera upload, OCR preprocessing with OpenCV, and a chat assistant UI with a dark mode toggle and animated elements.

## 2. Key Features

- Upload or capture images directly from camera
- Extract text using Tesseract OCR with preprocessing (grayscale + adaptive thresholding)
- Interact with ChatGPT via clickable buttons that open a new tab with the extracted text as a prompt
- Cute UI with dark mode, chat bubble assistant, and comic-style fonts
- Download, copy, or fix OCR output easily

## 3. Project Structure

- app.py → Flask backend logic
- templates/index.html → Frontend UI (HTML + JS + CSS)
- static/uploads/ → Temporary folder to store uploaded images
- requirements.txt → Dependencies

## 4. Backend Workflow

1. User uploads or captures an image.
2. Image is saved temporarily in 'static/uploads'.
3. OpenCV preprocesses the image (grayscale + adaptive threshold).
4. pytesseract extracts text using OCR.
5. Extracted text is rendered back to the UI and available for ChatGPT interactions.

## 5. Frontend Workflow

- User interface is styled with Comic Neue and Caveat fonts.
- Toggle button switches between light and dark modes.
- OCR result area displays the extracted text in a styled textarea.
- Buttons allow copy, download, ChatGPT interaction, and OCR fixing.
- Floating assistant provides a friendly chat bubble interaction.

## 6. Dependencies

- Flask
- pytesseract
- pillow
- opencv-python
- openai>=1.0.0 (optional for advanced integrations)