TECHNICAL DOCUMENTATION

By: Hrid Chakraborty

Date: 17th February 2024

Project Name:

HTML Table Extraction from Checklist image using AI (Backend)

Context:

Companies have their checklists in image form within maintenance logbooks. Working with data for huge checklists become cumbersome.

Objective:

Creation of API that takes a base64 image of the checklist and an optional user prompt as input and returns a json response containing the stringified html table as output.

Main Tools Used:

1. [Python](https://www.python.org/) programming language

2. [Flask](https://flask.palletsprojects.com/en/stable/) framework for creating the backend API server

3. [Pandas](https://pandas.pydata.org/) library for table manipulation

4. [Gemini](https://ai.google.dev/gemini-api/docs) 2.0 Flash API for finding the relevant column indices based on user input

5. [Postman](https://www.postman.com/) for testing the API endpoint

API Usage:

API Endpoint: http://<host\_ip>/GenerateTable

Content-Type: application/json

Method: POST  
API Body:  
{

“userPrompt”: <optional: used for specifying a different tag suffix>  
 “base64Image”: <base64 PNG image>  
}

(NOTE: Be careful when copy-pasting the body from this document. The quotation marks are rich text and can cause issues.)

Config File Format:

Type: json

Body:

{  
 “API\_KEY”: <GEMINI\_API\_KEY>  
}

(NOTE: Be careful when copy-pasting the body from this document. The quotation marks are rich text and can cause issues.)

Project Usage Guide (Requires Python 3.10):

1. Create a virtual environment named “myenv” in the project folder using “python -m venv myenv”

2. Activate the virtual environment using “./myenv/Scripts/activate”

3. Run “pip install -r requirements.txt” to install all the necessary packages and dependencies

4. Create a config.json file, if it does not exist. Follow the format given in the documentation.

5. Start the server by running “python main.py”. Make sure that the venv is activated while doing so.

6. The endpoint is localhost:5001/GenerateTable. Test it with Postman.

The code will be available in the Azure Repo “inSisAskIT” under “Generate Table” folder