Fall 2023-ITIS-6177 System Integration

Azure Computer Vision API

By

Pritesh Rajendra Ambavane (801296733)

In this project, we are developing an OCR (Optical Character Recognition) Application leveraging Computer Vision API. OCR technology is aimed at transforming images containing printed or handwritten text into text that is readable by machines. It employs sophisticated algorithms within the field of computer vision to detect and decipher text in images. This guide will detail the implementation of the OCR functionality, which is programmed in Node.js using the Express.js framework, in conjunction with the Computer Vision API.Prerequisites:

- Node.js installed on your computer
- Microsoft Azure account with access to the Computer Vision API
- I have deployed the code on Vulture

Working of the Application:

1. **Upload image**:-
    - It is a Web-Application in which the user has to provide with the image **url.**
2. **Text Recognition**:-
    - The server processes the received image by employing Microsoft Cognitive Services Vision API, specifically for text detection. This API generates a response that includes an operation ID unique to the text recognition task.
    - The Computer Vision API's Read API is utilized for the Read Operation. This involves a POST request that returns a response header containing the Operation ID.
    - The Computer Vision API's Get Read Result function utilizes the provided Operation ID to access and extract the text identified within the image.
3. **Error Handling**:-
    - The server is equipped to manage any potential errors arising in the course of text recognition. In case of an error, it conveys the error message directly to the user interface.
    - Error Code 400 indicates either a malformed request in JSON format or an unrecognizable binary file/image. This error also occurs when the image format is not among the supported ones, which include JPEG, PNG, BMP, PDF, and TIFF.
    - Error Code 415 is triggered when there is a mismatch between the 'Content-Type' specified and the actual content in the POST request, indicating an unsupported media type.
    - Error Code 404 signifies that the Operation ID provided is either incorrect or has expired.

API Endpoints

Create a **.env** and add your Azure Subscription key there:

“API_KEY=<your-subscription-key>”

Testing the Application on Postman and UI:-

Post Method :- POST "/":

This endpoint is responsible for processing the form submission from the main page. It receives the image as a parameter in the request body. Once the server receives the image, it examines it and provides the operation ID necessary for initiating the text recognition process.

Request URL: ‘https://eastus.api.cognitive.microsoft.com/vision/v3.2/read/analyze’

API:- http://64.176.215.214:3000/

Headers:- {

Ocp-Apim-Subscription-Key : “your-subscription-key”

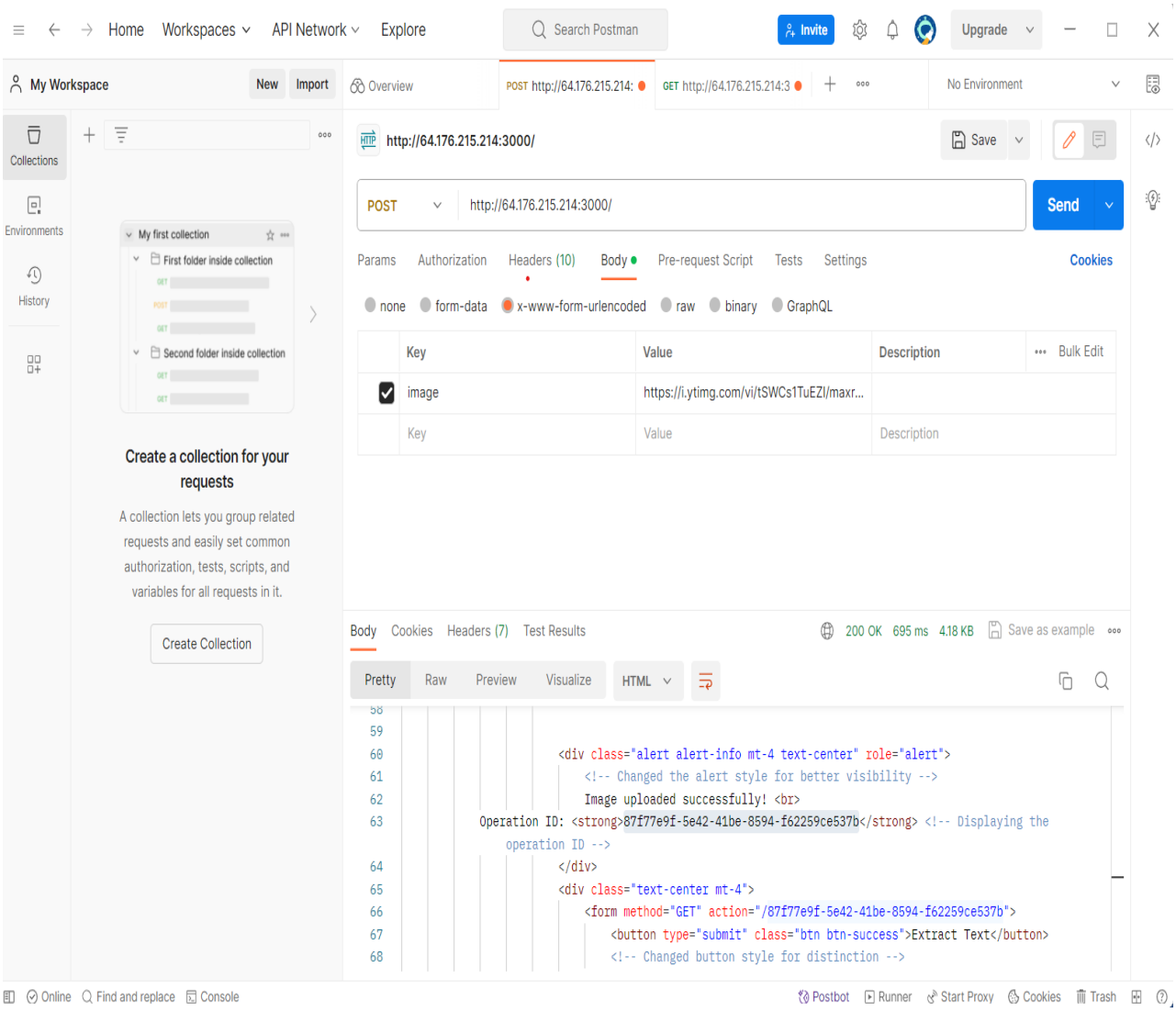
Content-Type: “Application/json”

}

Body:- {

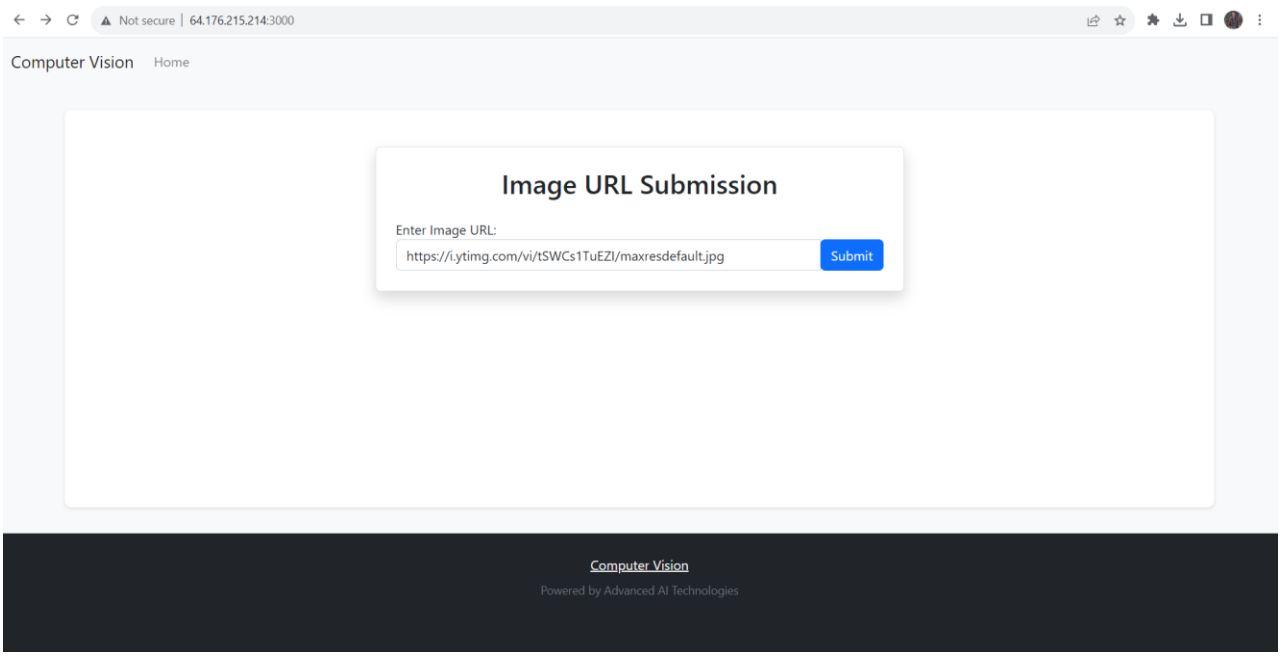
“image”:” https://i.ytimg.com/vi/tSWCs1TuEZI/maxresdefault.jpg”

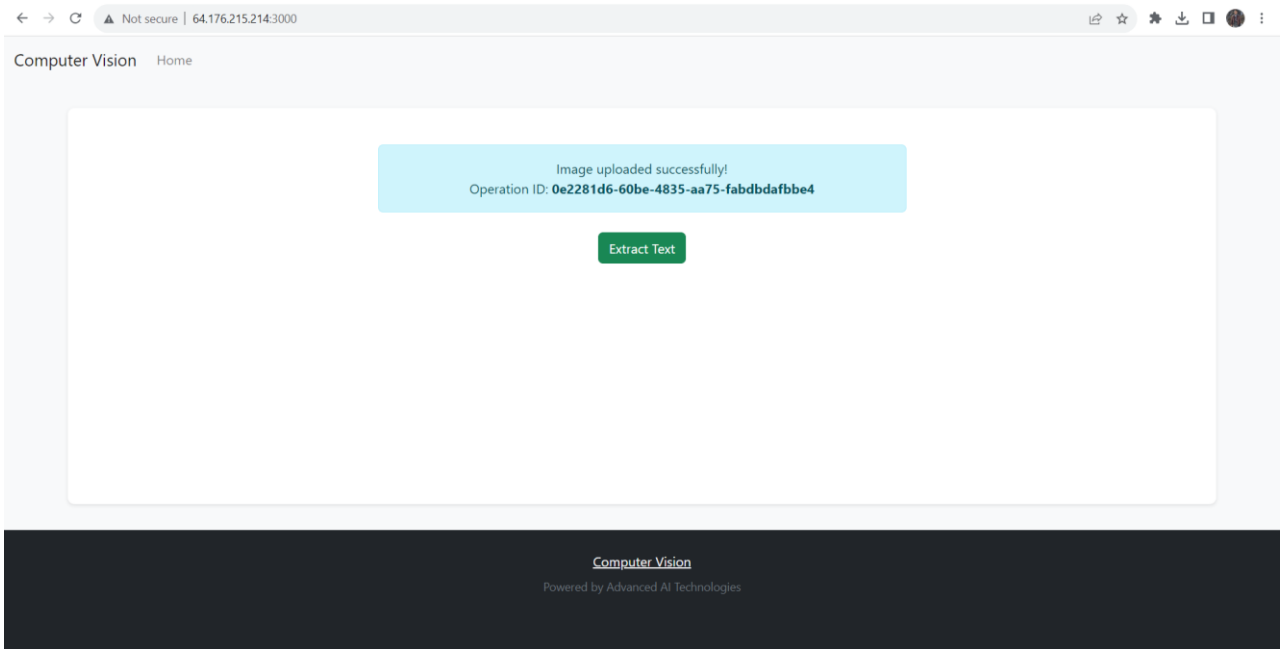
}


Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Overview
POST http://64.176.215.214:
GET http://64.176.215.214:3
No Environment
Collections
Environments
History
My first collection
First folder inside collection
Second folder inside collection
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
http://64.176.215.214:3000/
Save
POST
http://64.176.215.214:3000/
Send
Params
Authorization
Headers (10)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
Key
Value
Description
Bulk Edit
image
https://i.ytimg.com/vi/tSWCs1TuEZI/maxr...
Key
Value
Description
Body
Cookies
Headers (7)
Test Results
200 OK
695 ms
4.18 KB
Save as example
Pretty
Raw
Preview
Visualize
HTML

<!-- Changed the alert style for better visibility -->
Image uploaded successfully!
Operation ID: 87f77e9f-5e42-41be-8594-f62259ce537b <!-- Displaying the operation ID -->

<form method="GET" action="/87f77e9f-5e42-41be-8594-f62259ce537b">
<button type="submit" class="btn btn-success">Extract Text</button>
<!-- Changed button style for distinction -->
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash


Not secure | 64.176.215.214:3000
Computer Vision
Home
Image URL Submission
Enter Image URL:
https://i.ytimg.com/vi/tSWCs1TuEZI/maxresdefault.jpg
Submit
Computer Vision
Powered by Advanced AI Technologies


Not secure | 64.176.215.214:3000
Computer Vision
Home
Image uploaded successfully!
Operation ID: 0e2281d6-60be-4835-aa75-fabdbdafbbe4
Extract Text
Computer Vision
Powered by Advanced AI Technologies

GET Method:- GET "/:id":

This endpoint manages the monitoring of the text recognition procedure. The server repeatedly queries the Microsoft Cognitive Services Vision API using the operation ID until the recognition task is finalized. Upon completion, the text that has been identified and recognized is then presented on the user interface.

Request URL: 'https://eastus.api.cognitive.microsoft.com/vision/v3.2/read/analyzeResults/{Operation ID}'

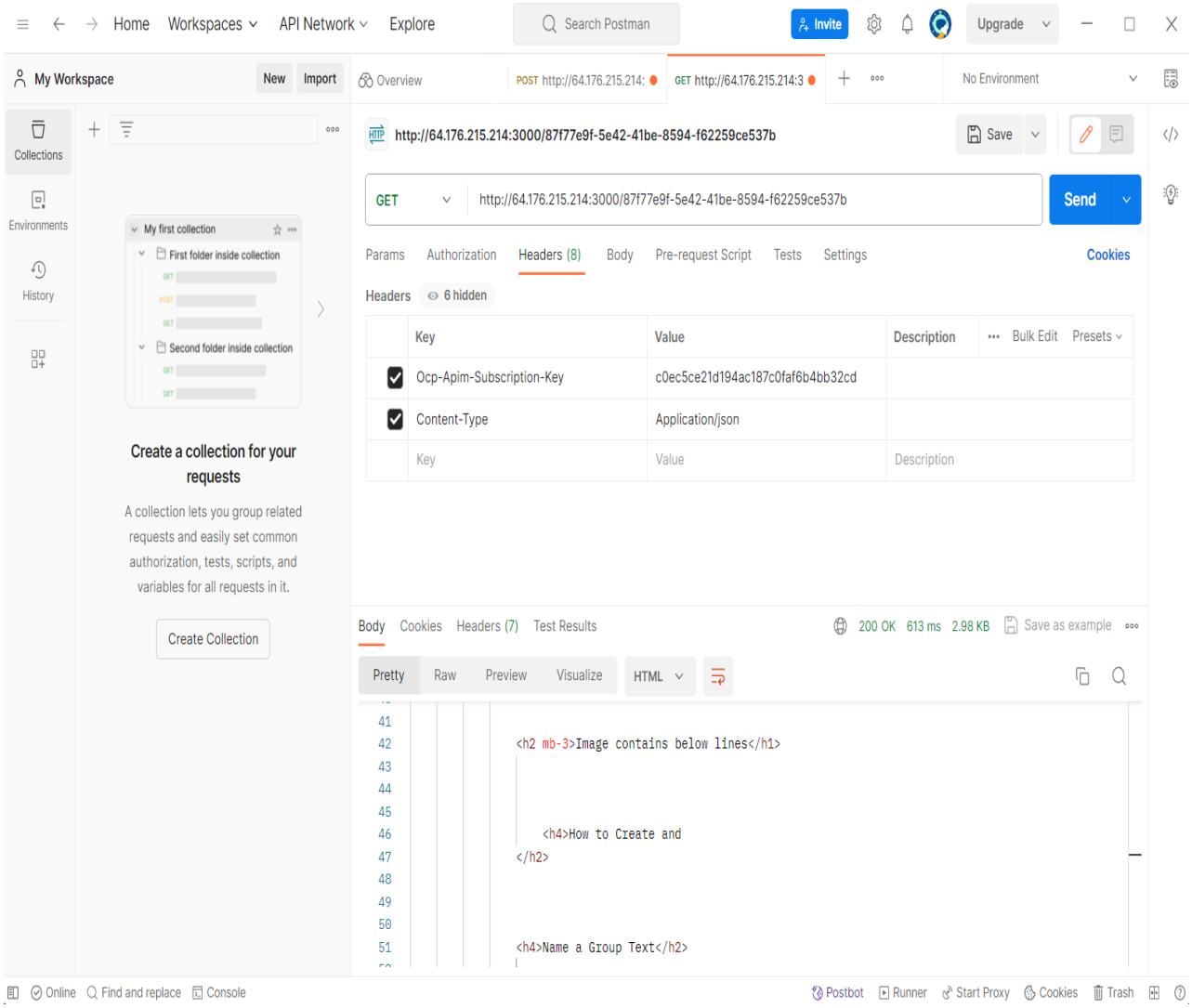
API:- http://64.176.215.214:3000/87f77e9f-5e42-41be-8594-f62259ce537b

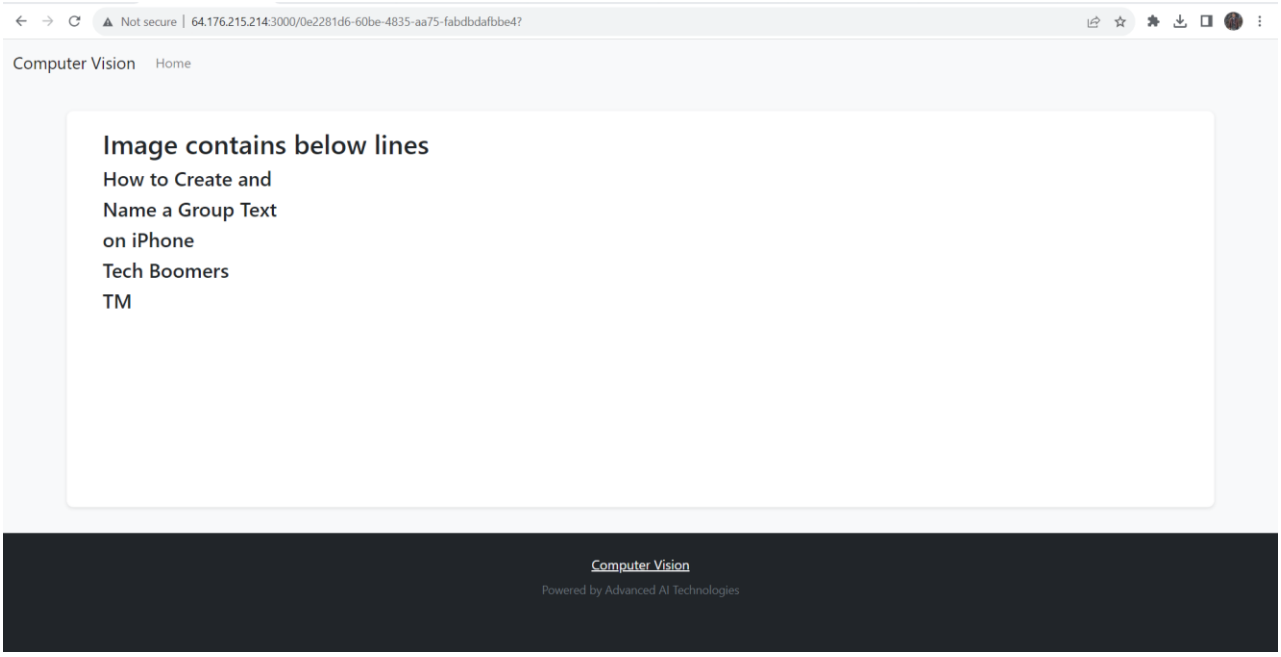
Headers:- {

Ocp-Apim-Subscription-Key : "your-subscription-key"

Content-Type: "Application/json"

}





Future Scopes:-

1) The user interface (UI) has the potential for further enhancements and refinements.
2) The application's scope can be broadened by integrating additional endpoints from the Computer Vision API.
3) Specifically, utilizing the 'Detect Objects' and 'Describe Objects' features could help in assessing whether the images contain any mature or graphic content.