**OS Hackathon Challenge - Pre-Qualification Assignment**

**Group Members:**

|  |  |  |
| --- | --- | --- |
| **Name** | **Mobile No.** | **Email** |
| Mohanish Khambadkar | 9403904623 | [mohanish.khambadkar@mitaoe.ac.in](mailto:mohanish.khambadkar@mitaoe.ac.in) |
| Shruti Dhumal | 9322370319 | [shruti.dhumal@mitaoe.ac.in](mailto:shruti.dhumal@mitaoe.ac.in) |
| Shrutika Jadhav | 8788721430 | [shrutika.jadhav@mitaoe.ac.in](mailto:shrutika.jadhav@mitaoe.ac.in) |

**Team Name**: Mohanish

**College Name :** MIT Academy of Engineering, Alandi, Pune.

1. **Github Code Repository**

**Link to Repository:**

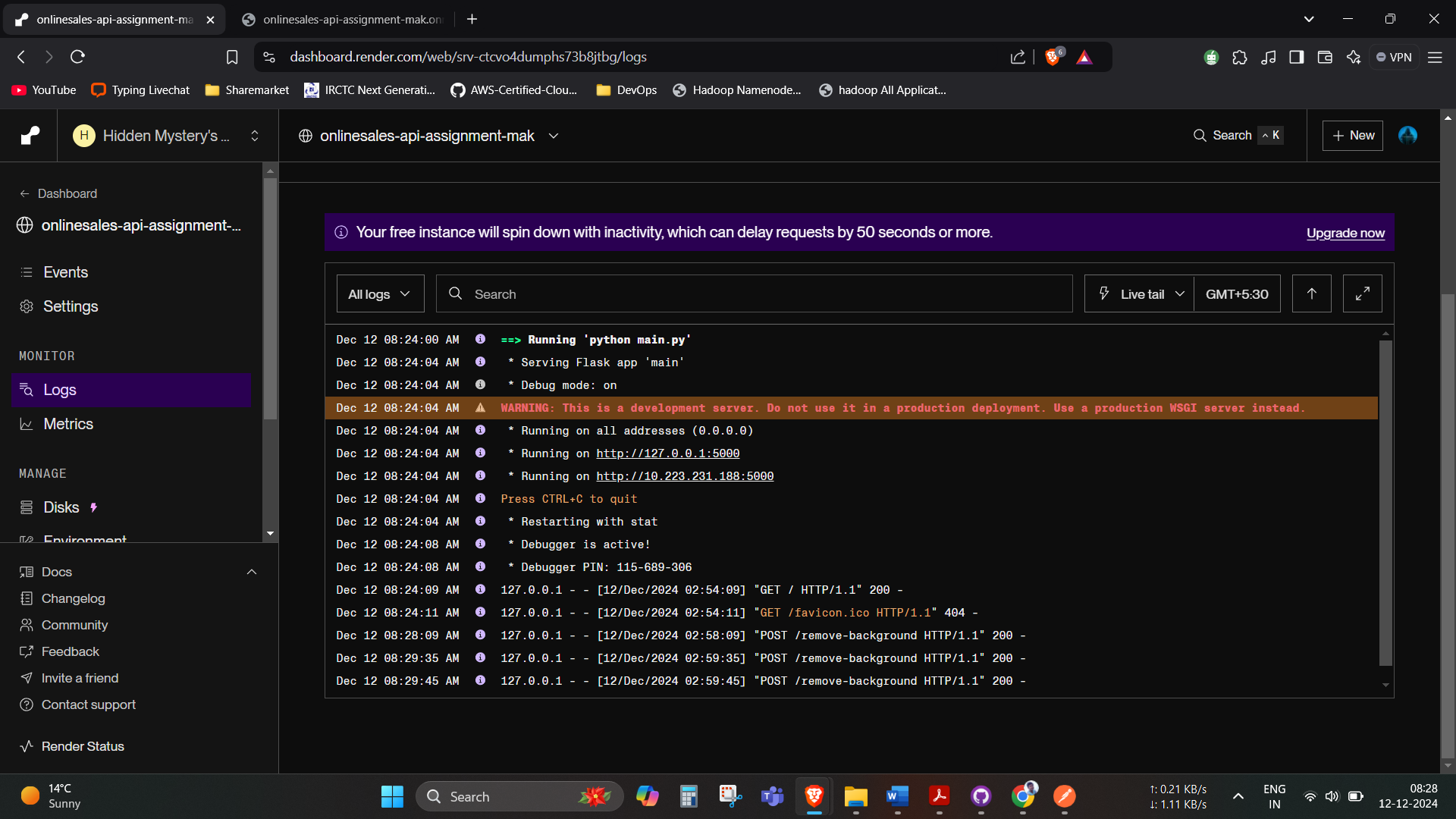
<https://github.com/Mohanish-5744/OS-Hackathon-Challenge---Pre-Qualification-Assignment-Mohanish.git>

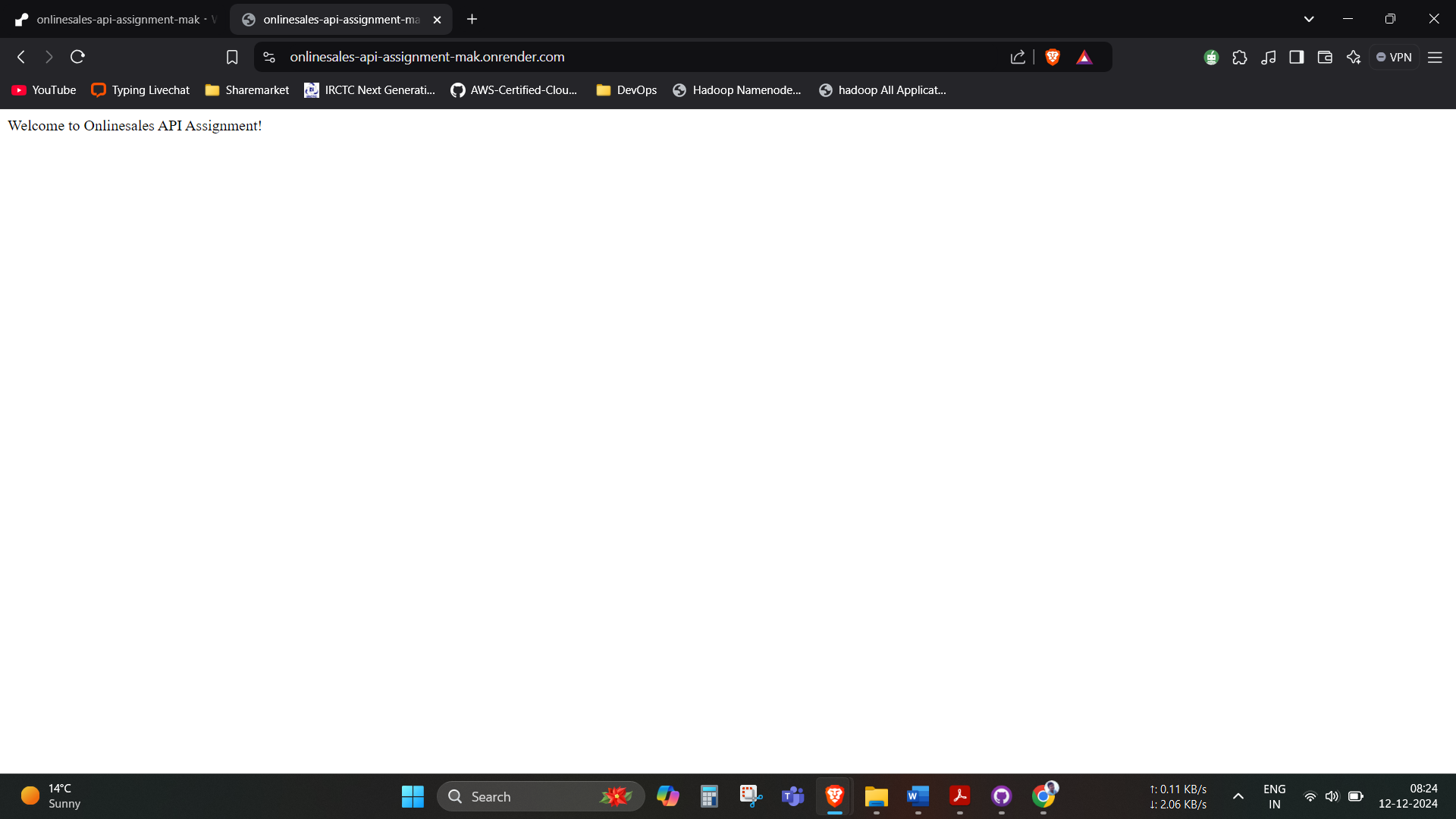
* Also shared the repository link with [hackathon@onlinesales.ai](mailto:hackathon@onlinesales.ai) and also grant acces for the same.

1. **API Hosting**

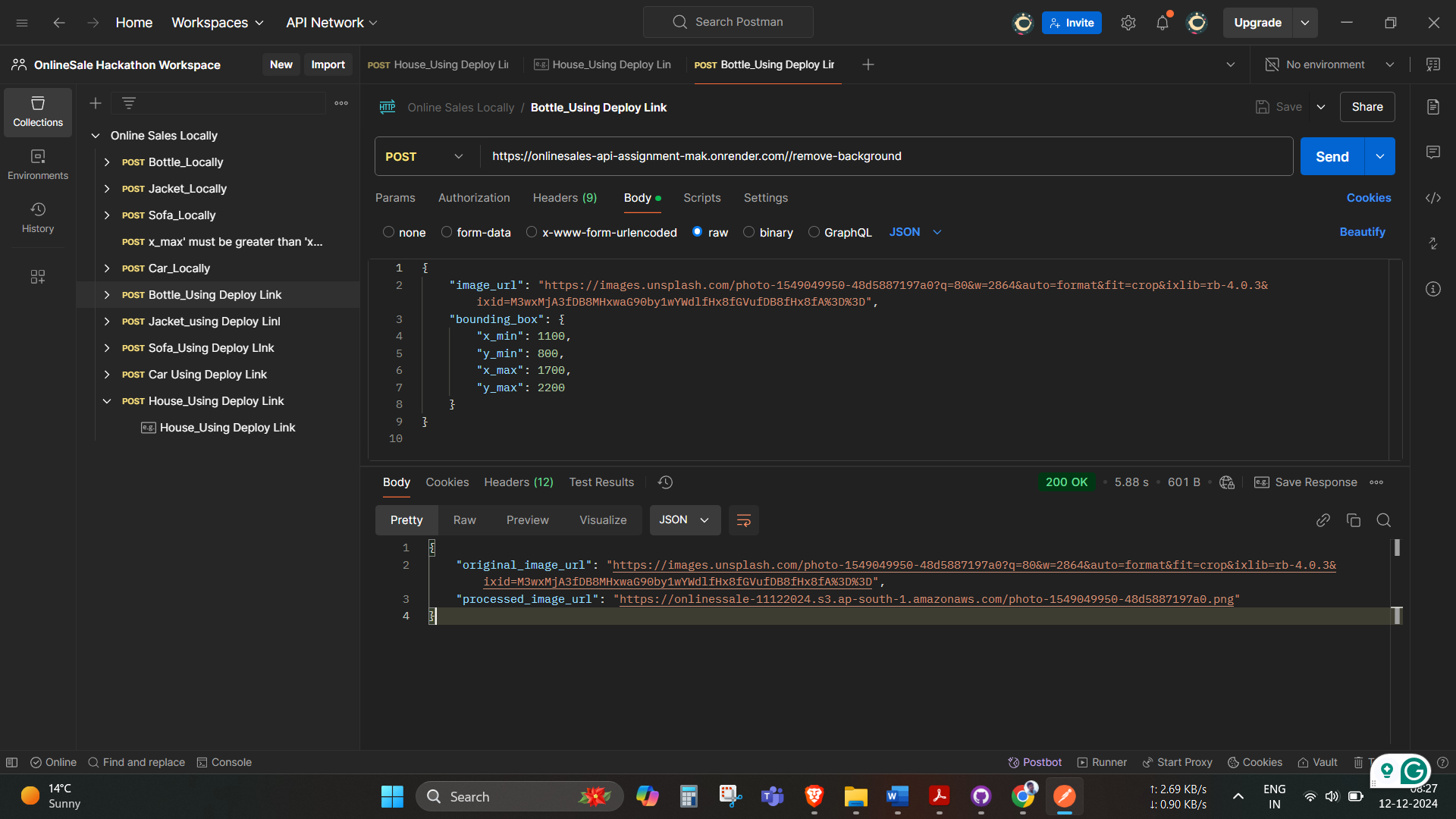
**API is deployed on the Render Website Link for the same is given below:**

<https://onlinesales-api-assignment-mak.onrender.com/>

It will take nearly 30-50 seconds to start the service because of free tier instance kindly consider this issue.



**Testing link of API through Postman:**

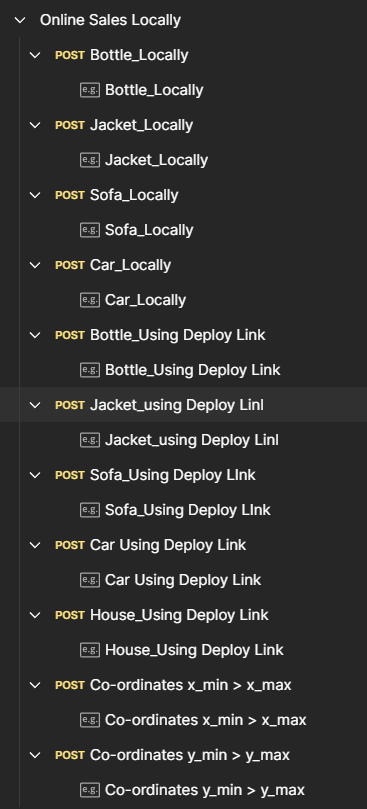
[https://onlinesales-api-assignment-mak.onrender.com//remove-background](https://onlinesales-api-assignment-mak.onrender.com/remove-background)

1. **Postman Collection:**

There is a collection name called Onilne Sales Hackathon API for testing API and created workspace and given access to email [hackathon@onlinesales.ai](mailto:hackathon@onlinesales.ai) below is the testing images. The collection is also uploaded on the GitHub repository link for the same is given below:

<https://github.com/Mohanish-5744/OS-Hackathon-Challenge---Pre-Qualification-Assignment-Mohanish/blob/main/Online%20Sales%20Hackathon%20API.postman_collection.json>

We have tested it first locally and then by deployable link. We have tested for 4 images that you gave and and additional House images for checking API. We also tested for if co-ordinates xmin> xmax error will occur and ymin> ymax error will occur.



1. **Documentation**

**How to set up and run the code locally.**

1. Clone the repository in your local system.
2. Install requirements.txt file given in repository in the local machine using below command make sure python is install on your computer.

Command -> pip install -r requirements.txt

1. Now open terminal and put below command to run api you will get below output in the terminal

Command-> python main.py

You will get below output

PS C:\Users\HP\Desktop\Onlinesales\OS-Hackathon-Challenge---Pre-Qualification-Assignment-Mohanish> python main.py

\* Serving Flask app 'main'

\* Debug mode: on

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

\* Running on all addresses (0.0.0.0)

\* Running on <http://127.0.0.1:5000>

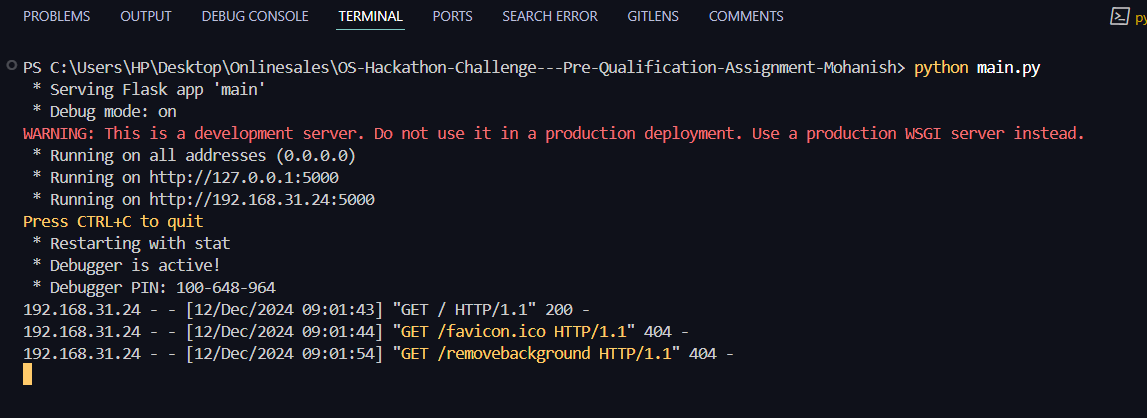
\* Running on <http://192.168.31.24:5000>

Press CTRL+C to quit

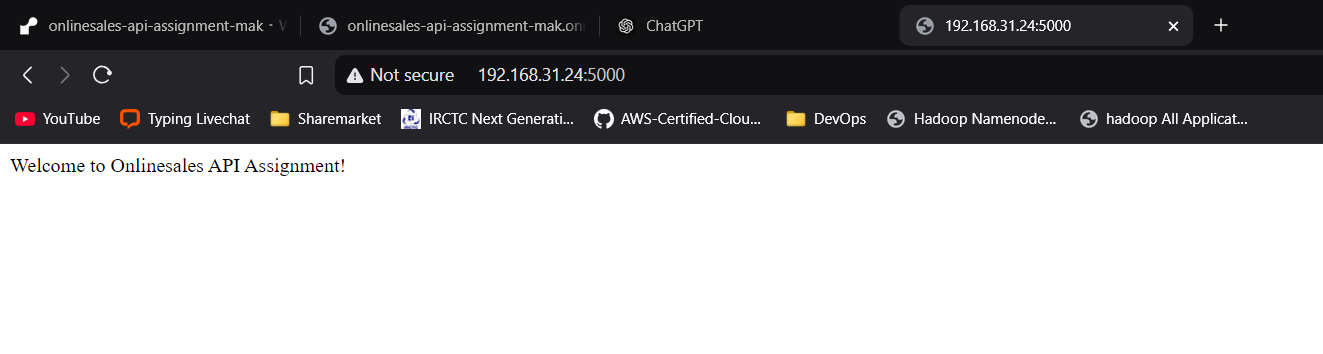
\* Restarting with stat

\* Debugger is active!

\* Debugger PIN: 100-648-964



Chek <http://127.0.0.1:5000> You will get below message



1. Open Postman and put <http://127.0.0.1:5000/remove-background> on Post request and add JSON file below

{

    "image\_url": "https://images.unsplash.com/photo-1549049950-48d5887197a0?q=80&w=2864&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",

  "bounding\_box": {

      "x\_min": 1100,

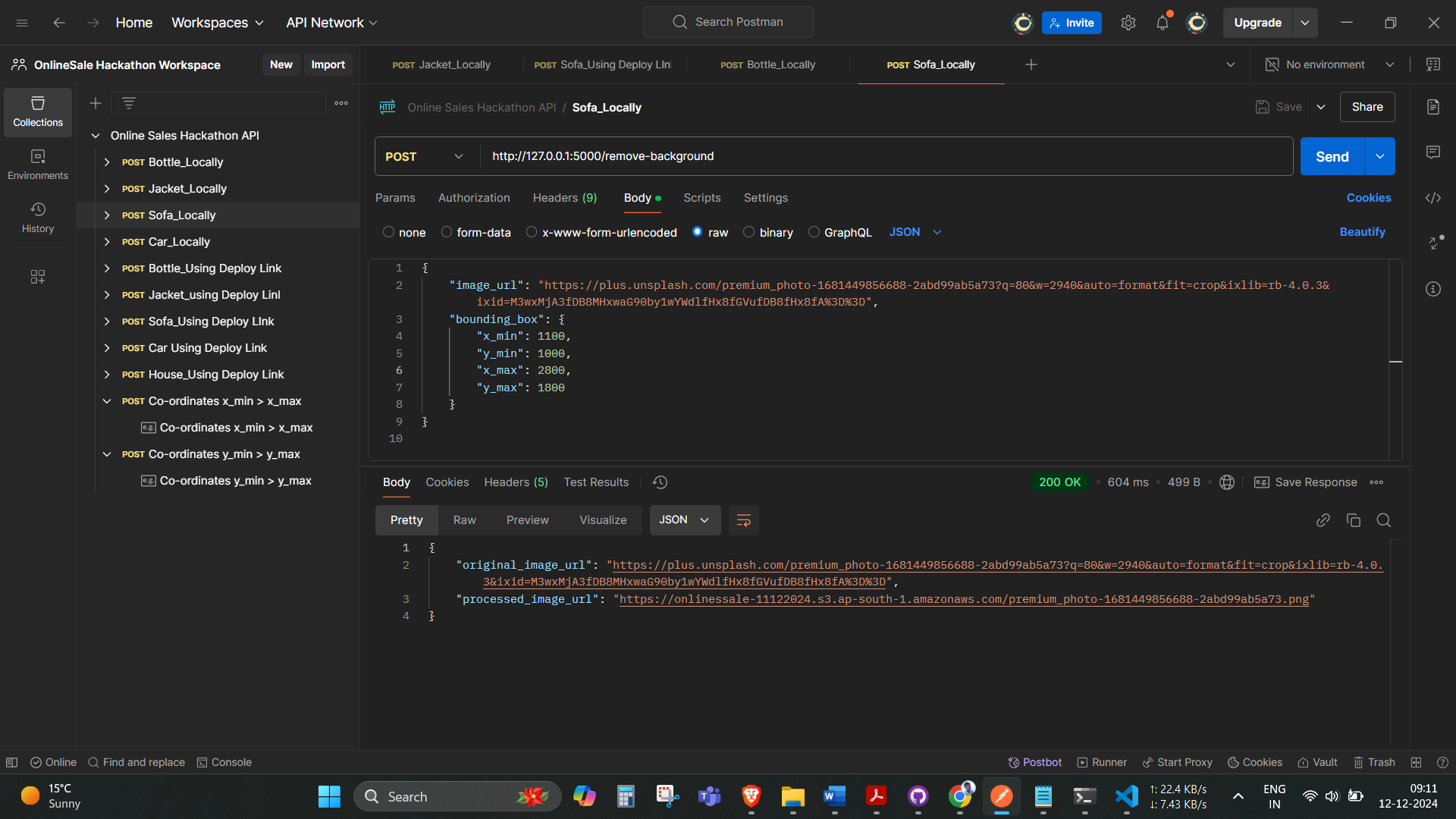
      "y\_min": 800,

    "x\_max": 1700,

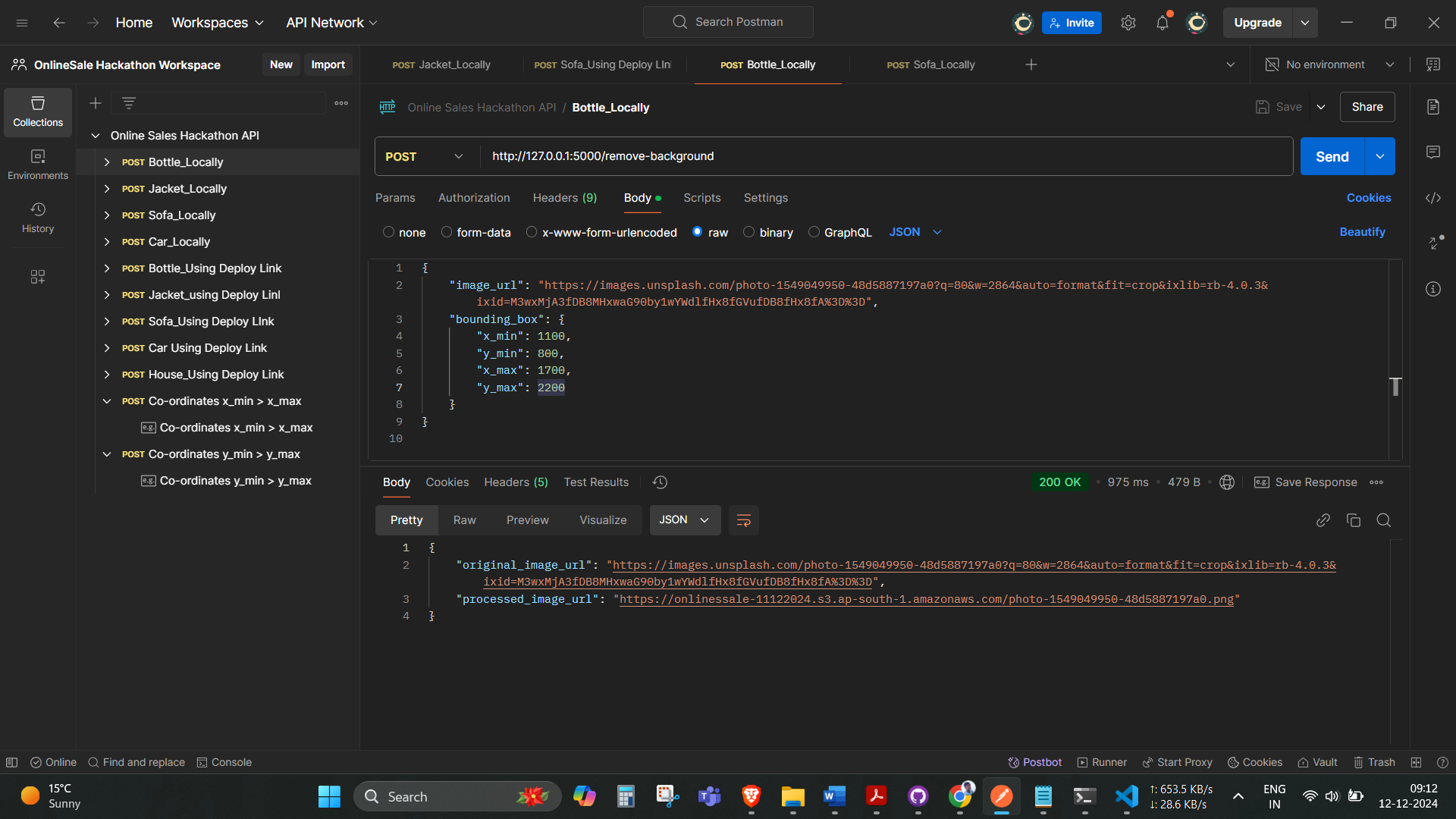
      "y\_max": 2200

    }

}

Sofa Testing

Bottle Testing

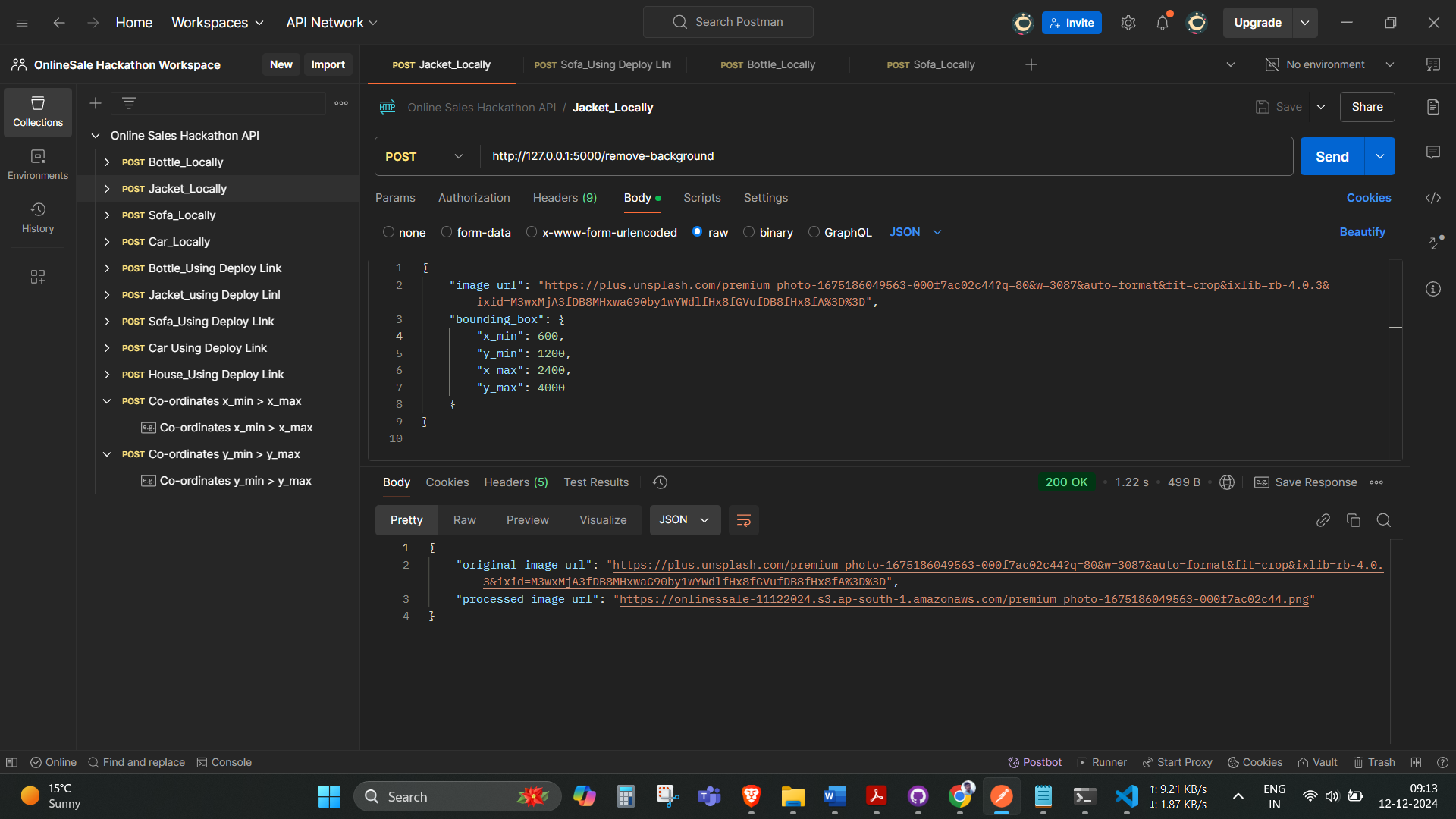


{

"original\_image\_url": "https://images.unsplash.com/photo-1549049950-48d5887197a0?q=80&w=2864&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",

"processed\_image\_url": "https://onlinessale-11122024.s3.ap-south-1.amazonaws.com/photo-1549049950-48d5887197a0.png"

}

Jacket Testing

j

{

{

"original\_image\_url": "https://plus.unsplash.com/premium\_photo-1675186049563-000f7ac02c44?q=80&w=3087&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",

"processed\_image\_url": "https://onlinessale-11122024.s3.ap-south-1.amazonaws.com/premium\_photo-1675186049563-000f7ac02c44.png"

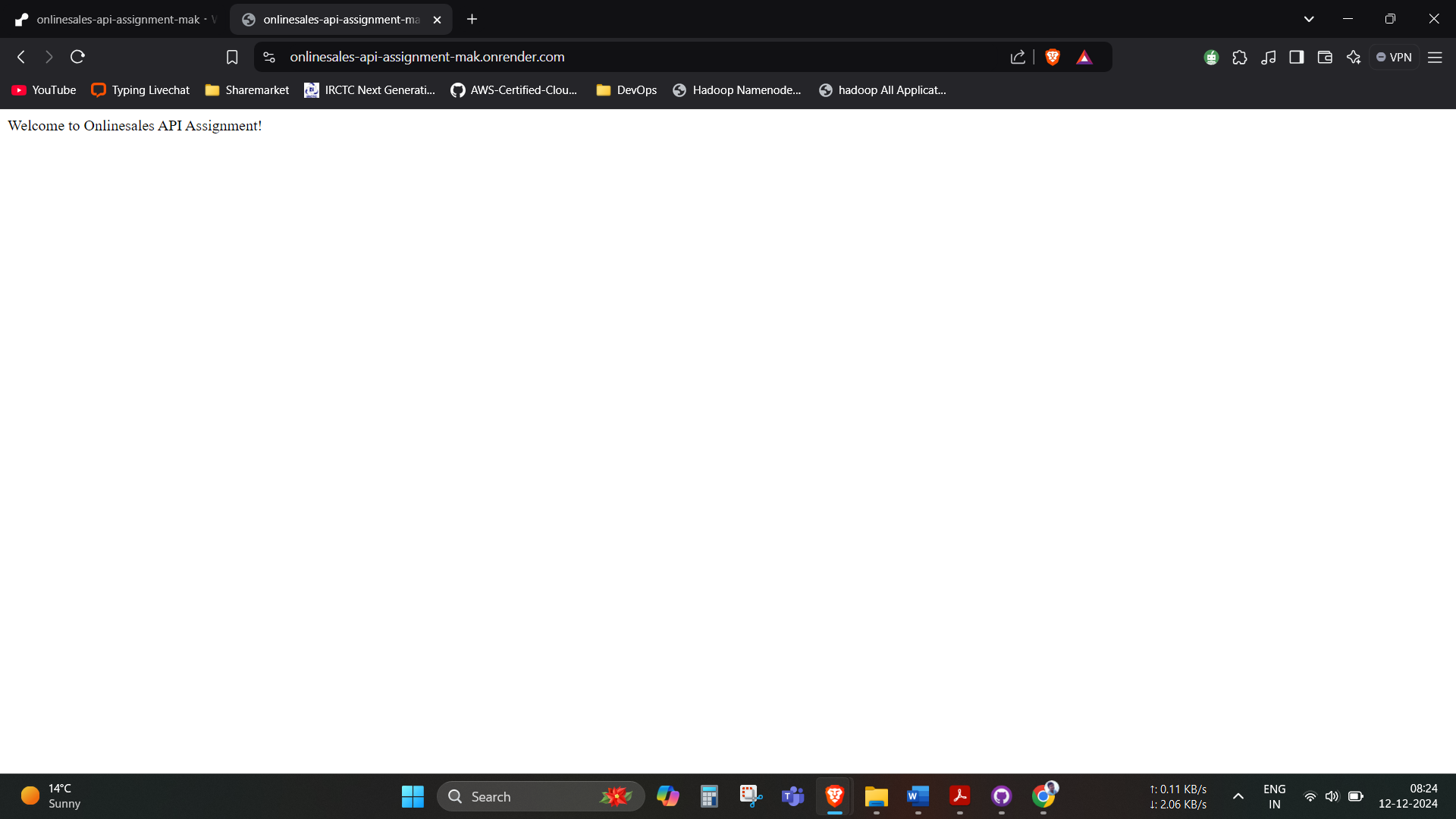
}

**Instructions for API usage (including input/output contracts).**

1. For Testing API is working or not use below link:

<https://onlinesales-api-assignment-mak.onrender.com/>

It will take nearly 30-50 seconds to start the service because of free tier instance kindly consider this issue. It will give below output



C

1. Open Postman, create new collection and test using put below link with POST request and Json file given below according to assignment.

<https://onlinesales-api-assignment-mak.onrender.com//remove-background>

JSON File:

{

    "image\_url": "https://plus.unsplash.com/premium\_photo-1681449856688-2abd99ab5a73?q=80&w=2940&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",

    "bounding\_box": {

        "x\_min": 1100,

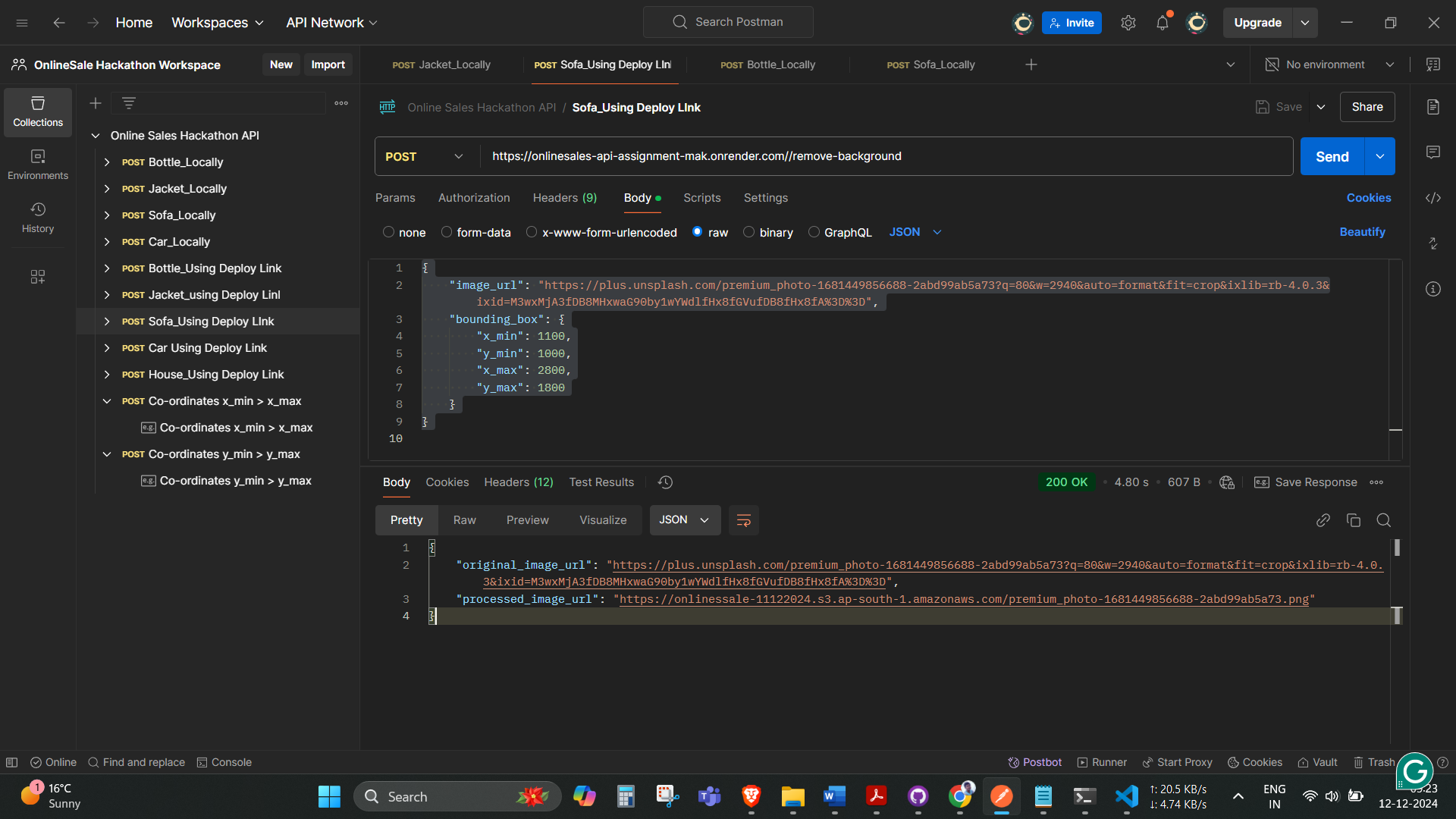
        "y\_min": 1000,

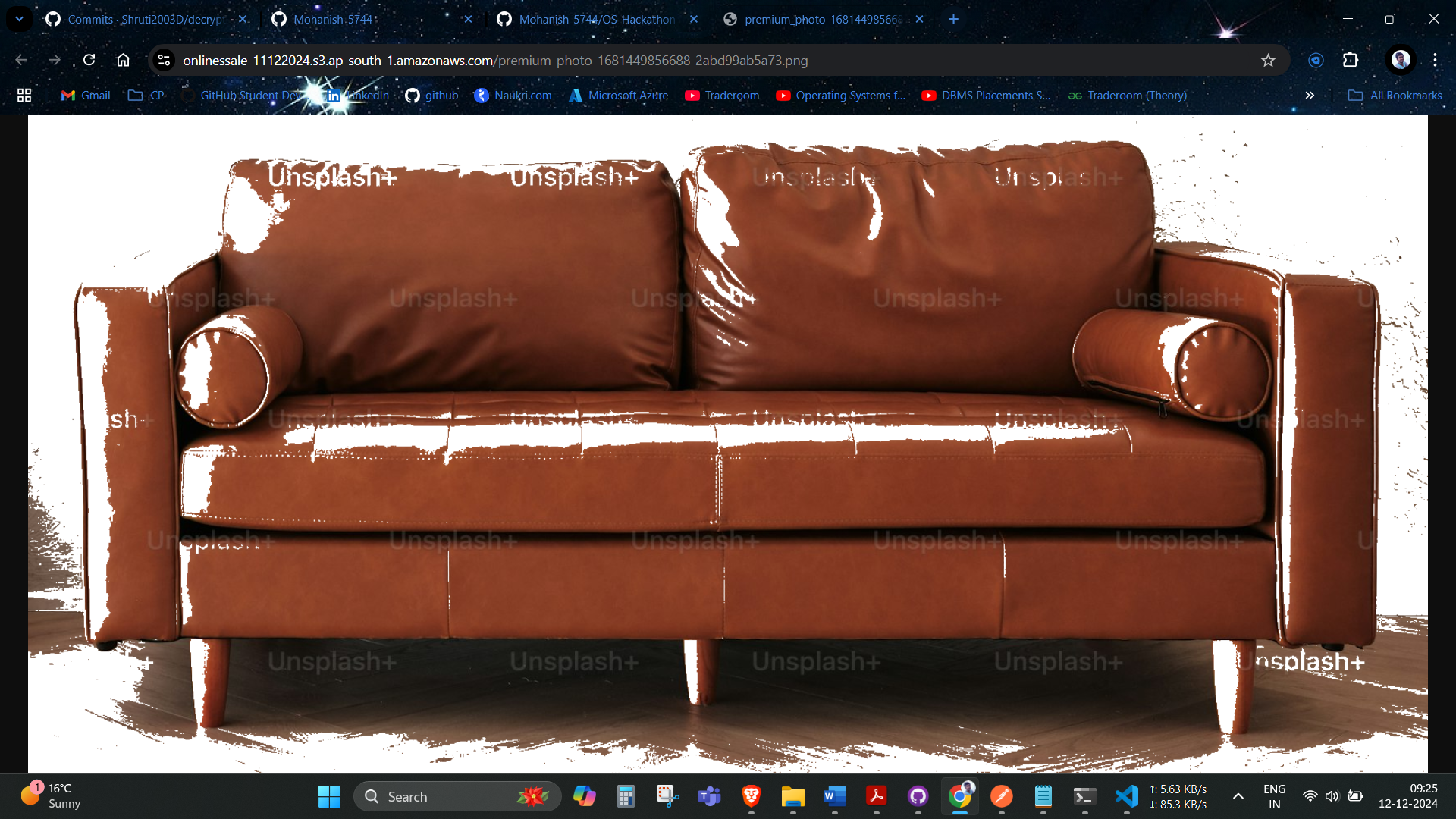
        "x\_max": 2800,

        "y\_max": 1800

    }

}





Processed image of sofa and stored in AWS S3.

We have also chek for xmin>xmax and error is occurred. You can use below JSON file with same API and xmin>xmax co-ordinates given below.

{

    "image\_url": "https://plus.unsplash.com/premium\_photo-1681449856688-2abd99ab5a73?q=80&w=2940&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",

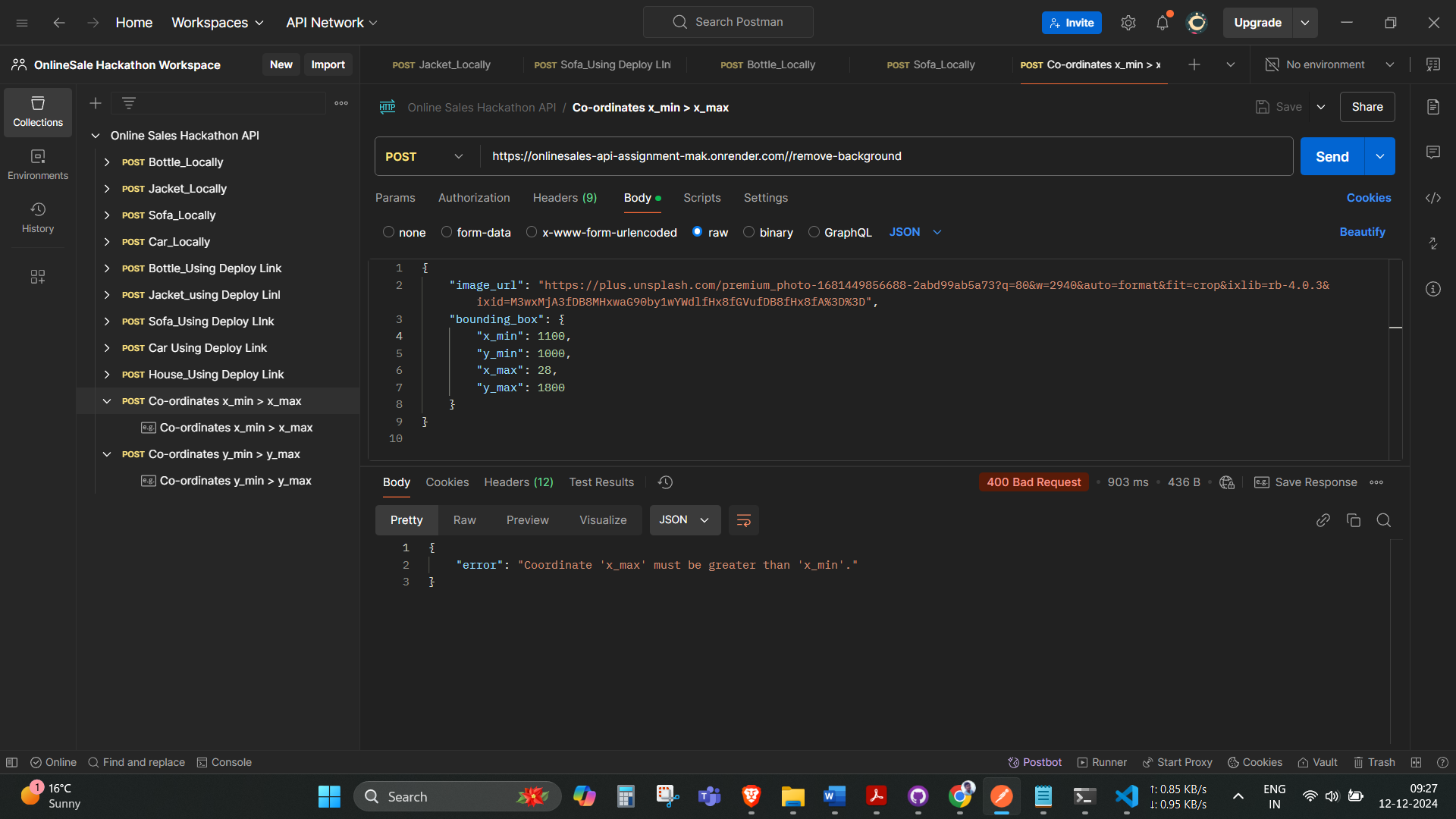
    "bounding\_box": {

        "x\_min": 1100,

        "y\_min": 1000,

        "x\_max": 28,

        "y\_max": 1800

    }

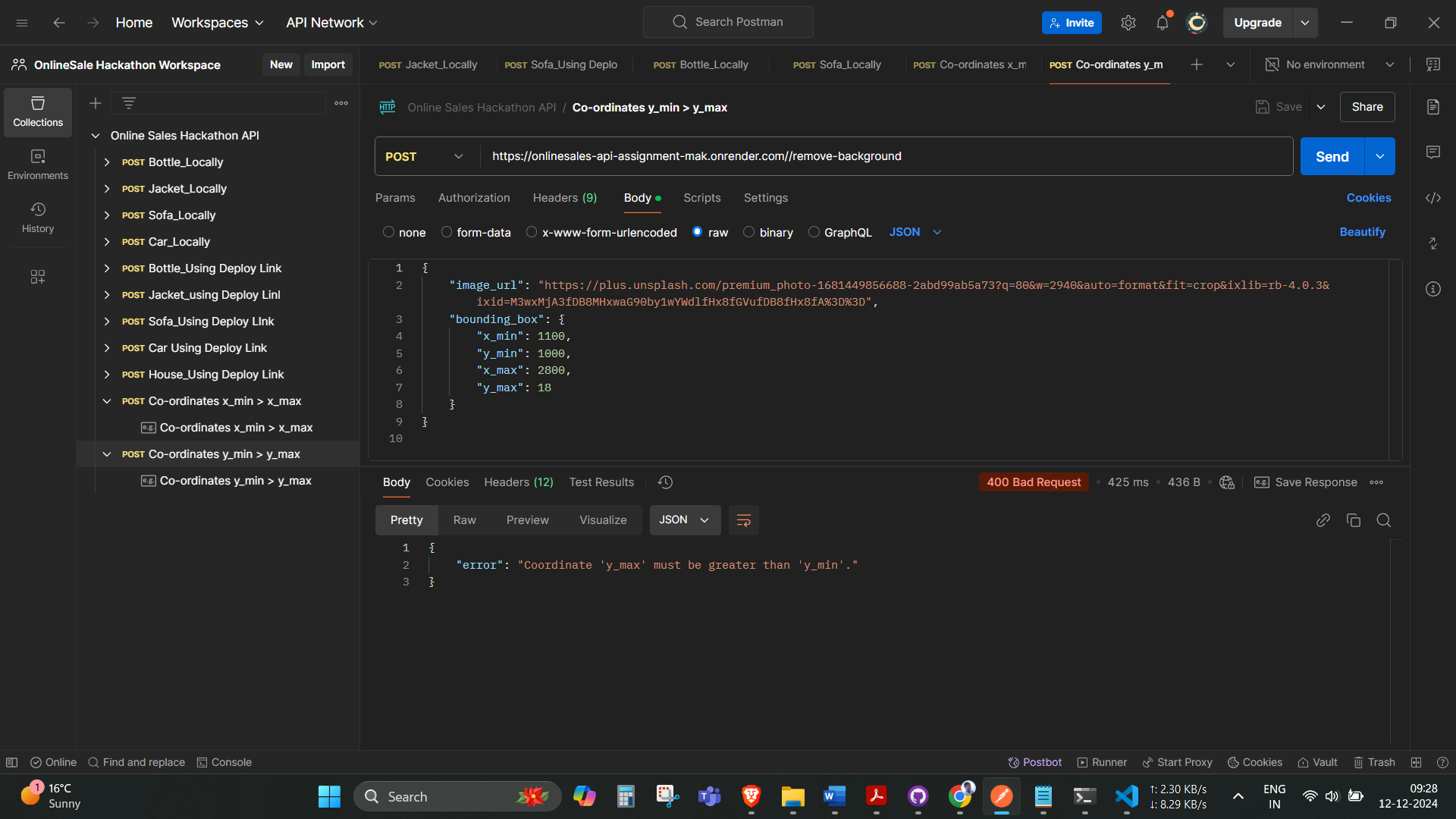
}

And same for y co-ordinates.

{

    "image\_url": "https://plus.unsplash.com/premium\_photo-1681449856688-2abd99ab5a73?q=80&w=2940&auto=format&fit=crop&ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D",

    "bounding\_box": {

        "x\_min": 1100,

        "y\_min": 1000,

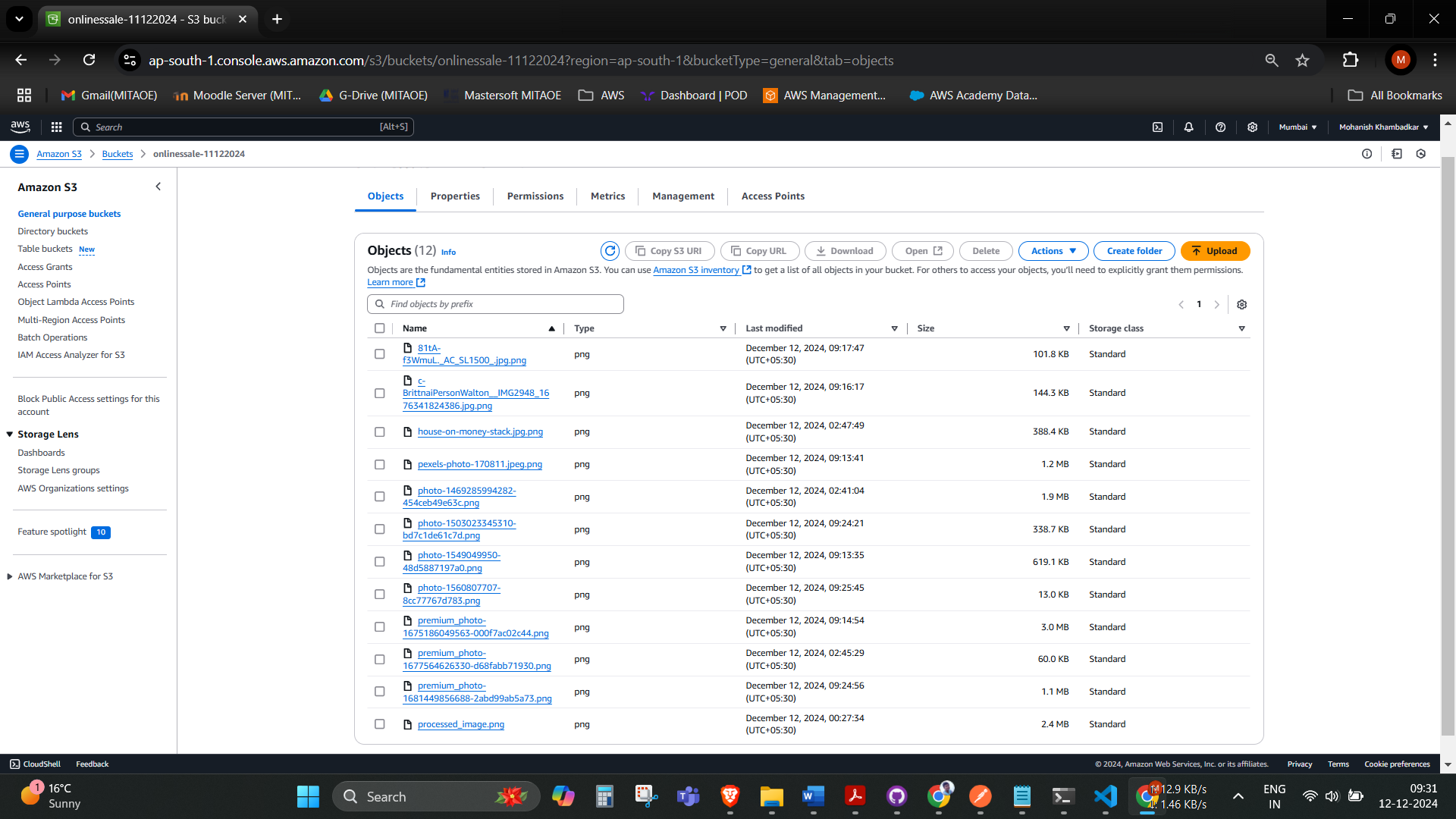
        "x\_max": 2800,

        "y\_max": 18

    }

}

**All the Processed images is storing in AWS S3 bucket**

****

**A brief explanation of the tools, frameworks, or libraries used**

**Tools used :**

* + Render:
* A cloud-based platform for deploying and hosting web applications and APIs.
* Used in this project to deploy the Flask application, providing scalability and secure HTTPS connections.
  + Postman:
* A widely used API testing tool.
* Used for sending HTTP requests to the /remove-background endpoint with JSON payloads and validating the API's functionality by examining the responses.

**Framework**

* + Flask:
* A micro web framework in Python designed for building web applications and APIs.
* In this project:
* Defines routes (/remove-background and /) to handle HTTP requests.
* Uses jsonify to format Python objects into JSON for API responses.
* Facilitates parsing incoming data with request.

**Libraries Used**

* + requests:
* A Python library for making HTTP requests.
* Used to download the image from the provided URL.
* OpenCV (cv2):
* An open-source library for computer vision and image processing.
* In this project:
  + Decodes images from bytes.
  + Processes the image to remove the background and replace it with a white background.
  + Crops the specified region of interest (ROI).
  + Applies grayscale conversion and binary thresholding.
  + NumPy (np):
* A library for efficient numerical and array operations.
* Used to handle and manipulate image data as multidimensional arrays.
  + AWS SDK for Python (boto3):
* A library for interfacing with AWS services, such as S3.
* Used to upload processed images to an S3 bucket with public access permissions.
  + BytesIO (from io):
* Used to handle image data as byte streams during encoding and uploading to S3.
  + Pillow (PIL):
* An image processing library in Python.
* Although imported, it’s not explicitly used in the current code. However, it could be helpful for additional image manipulations.
  + urllib.parse:
* Provides functions for parsing URLs.
* Used to extract filenames from the provided image URLs for generating valid S3 keys.
  + os:
* A standard Python library for interacting with the operating system.
* Used to manage file paths and extract filenames from URLs.