**Text-to-Image API Documentation**

**Base URL**

* Local: http://127.0.0.1:5000/

**Endpoints**

**1. Generate Image Based on Text Input**

* **URL**: /generate
* **Method**: POST
* **Description**: Generates an image based on the provided text prompt using Stable Diffusion, analysis it with CLIP, and performs segmentation with SAM.

**Request Format**

* **Content-Type**: application/json
* **Body Parameters**:
  + prompt: *(required)* The text input that will be used to generate an image.

**Example Request**

{

"prompt": "a mom holding a baby"

}

**Response Format**

* **Response Parameters**:
  + **prompt**: The text prompt used for image generation.
  + **generated\_image**: The generated image encoded in Base64 format.
  + **clip\_results**: The CLIP analysis results, showing the matching scores for predefined concepts.
  + **sam\_masks**: Segmentation masks generated by SAM (list of masks).
  + **sam\_scores**: Confidence scores for each segmentation mask (list of scores).

**Example Successful Response**

**Given prompt**: Mom handling baby

A screenshot of a computer

Description automatically generated

**Setup Instructions**

1. **Install dependencies** (Docker or directly):
   * Flask
   * Flask-CORS
   * torch
   * diffusers
   * opencv-python-headless
   * pillow
   * clip-by-openai
   * segment-anything
   * accelerate (optional for optimization)

**Using Docker:**

After creating image and container

**CMD:**

docker run -p 5000:5000 text-to-image-app

**Testing:**

You can test the API using **Postman** or any other API testing tool.

**Example Request in Postman:**

1. **Method**: POST
2. **URL**: http://127.0.0.1:5000/generate
3. **Body**: Set to raw JSON:

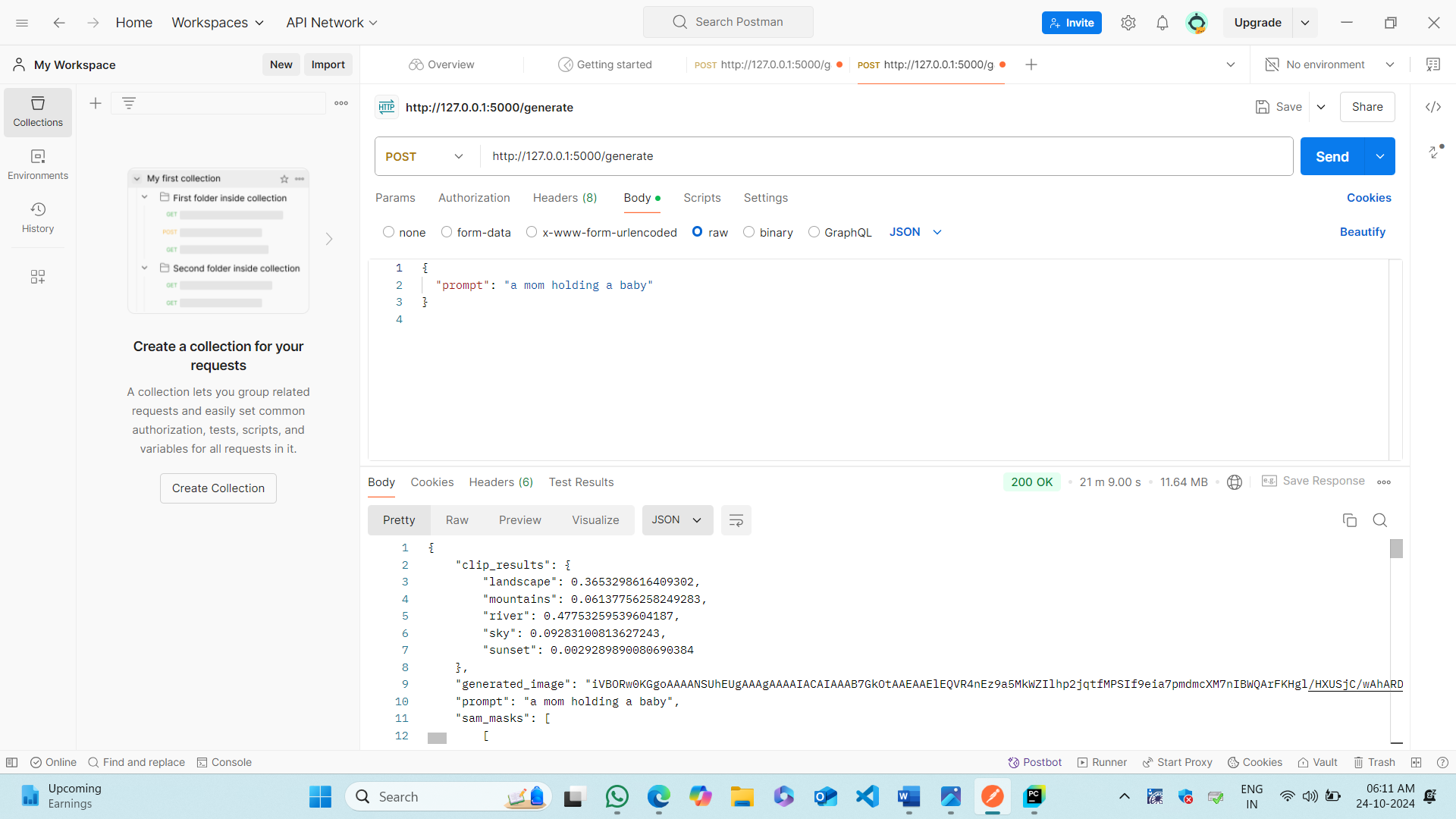
**raw**

{

"prompt": "a mom holding a baby"

}

**Result in Postman:**



**In Pycharm:**

In pycharm after running flask server

A screenshot of a computer

Description automatically generated