

IT645 – Web and Mobile Development

**Project Report on**

**QuickGPT | MERN stack based AI chatbot Application**

**Submitted By :**

202412063 – ANJALI PATEL

**Submission Date :** 29/11/2025

# INDEX

|  |  |  |
| --- | --- | --- |
| SR. NO. | Content | Page no. |
| 1 | Project introduction | 01 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Project Description

QuickGPT is a MERN stack based AI chatbot web application inspired by ChatGPT. It allows users to chat with an AI model, generate images, and explore community-shared posts. The system includes a credit-based model in which users must purchase credits to use AI features. The project also provides stripe-based payment integration, JWT authentication and clean, responsive frontend UI.

# Project Objective

* + To develop a functional AI chatbot, using MERN stack
  + To allow users to generate text and images using AI
  + To implement a credit-based usage system
  + To integrate secure payments using stripe
  + To provide a clean and responsive UI using modern web technologies

# Technology Stack

* **Frontend**
  + React.js
  + Tailwind CSS
* **Backend**
  + Node.js
  + Express.js
  + MongoDB + mongoose
* **Authentication**
  + JSON web token
* **Payments**
  + Stripe API
* **AI Features**
  + Google Gemini API
* **Third-party API**
  + Official Joke API

# System Architecture

# React Client → Express Server → MongoDB

# ↑ ↓

# Stripe Checkout ← Webhooks

* **Flow Summary :** 
  + User interacts with frontend
  + API calls sent to backend
  + Backend verifies token and then processes request if valid tocken
  + AI answer generation and credits deducted
  + Payments handled via stripe checkout
  + MongoDB stores users, chats, messages, credits & transactions

# Main Functionality

* **User authentication :** 
  + User can register with name, email, password
  + Password is hashed using bcrypt
  + JWT token generated during login
  + Authenticated routes are protected using middleware
* **Chat system :** 
  + Users can create chats
  + Chats include messages between user and AI
  + Chat name can be edited
  + Chat can be deleted anytime
  + Messages are stored in mongoDB
* **AI Text generation :** 
  + Users can send prompts
  + System calls AI API
  + Returns response
  + 1 credit deducted for each request
* **AI Image generation :** 
  + Users can generate images using AI
  + 2 credits per image generation
  + Generated images can be published in community gallery
* **Credit purchase system :** 
  + Users can buy credits via stripe
  + Plans include predefined credit amounts
  + Stripe checkout session is created and user is redirected
  + Successful payments are confirmed via webhook
  + Corresponding credits added into user’s account
* **Community Page :** 
  + All users can view AI-generated community images
  + Each post shows image + user details + prompt
* **Third-Party API integration for random jokes :** 
  + A random joke is fetched from Official joke API
  + Showcased in demo pages

# API Documentation [overview]

# Auth

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| POST | /api/users/register | Register user |
| POST | /api/users/login | Login user |
| GET | /api/users/me | Get logged in user |

# Chat

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/chats | List all chats |
| POST | /api/chats | Create chat |
| GET | /api/chats/:id | Get chat |
| PUT | /api/chats/:id | Update chat |
| DELETE | /api/chats/:id | Delete chat |

# AI answer generation

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| POST | /api/messages/text | Text generation |
| POST | /api/messages/image | Image generation |

# Credits

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/credits/plans | Plans list |
| POST | /api/credits/purchase | Stripe session |
| GET | /api/ credits/me | User credits |
| GET | /api/ credits/transactions | Transactions |
| PATCH | /api/ credits/transactions/:id | Update |

# Stripe

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| POST | /api/stripe/webhook | Raw webhook endpoint |

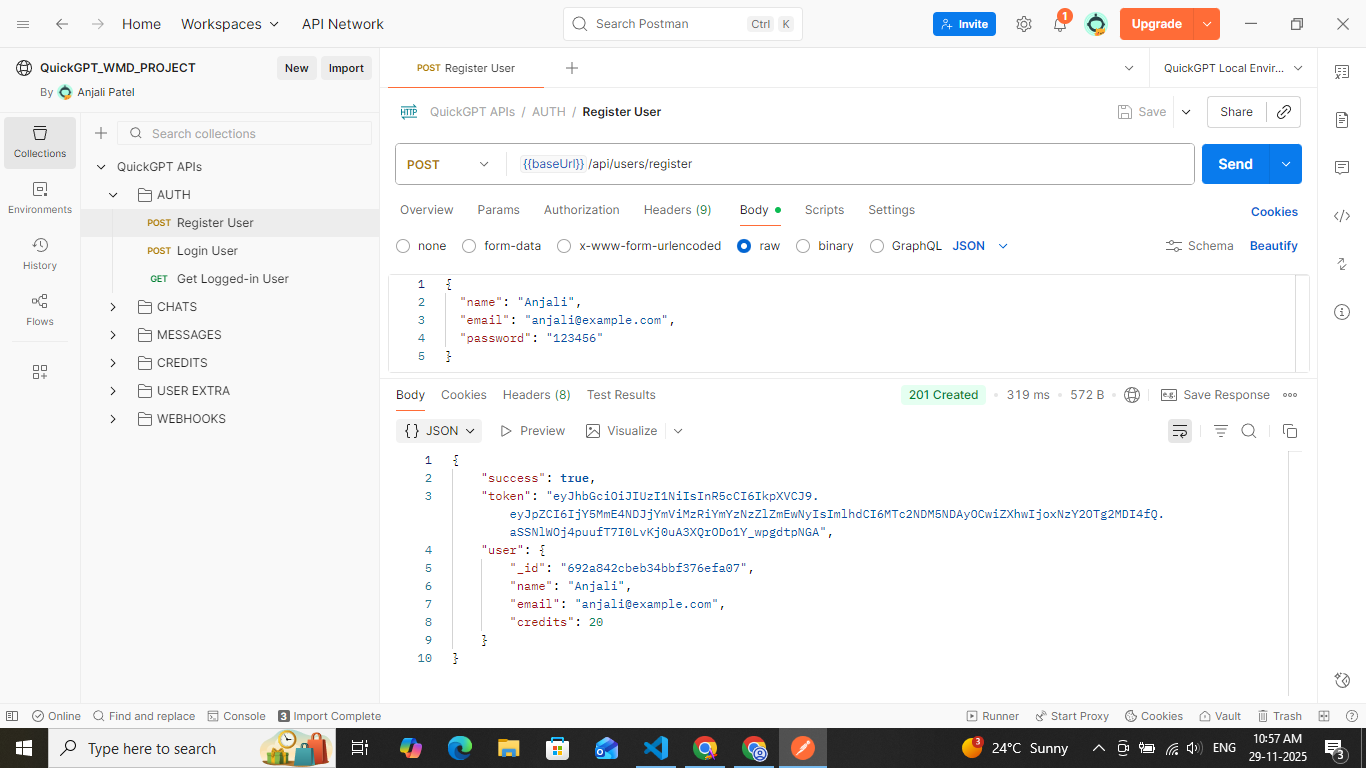
# Third-Party API

|  |  |  |
| --- | --- | --- |
| Method | Endpoint | Description |
| GET | /api/jokes/random | Generate random joke |

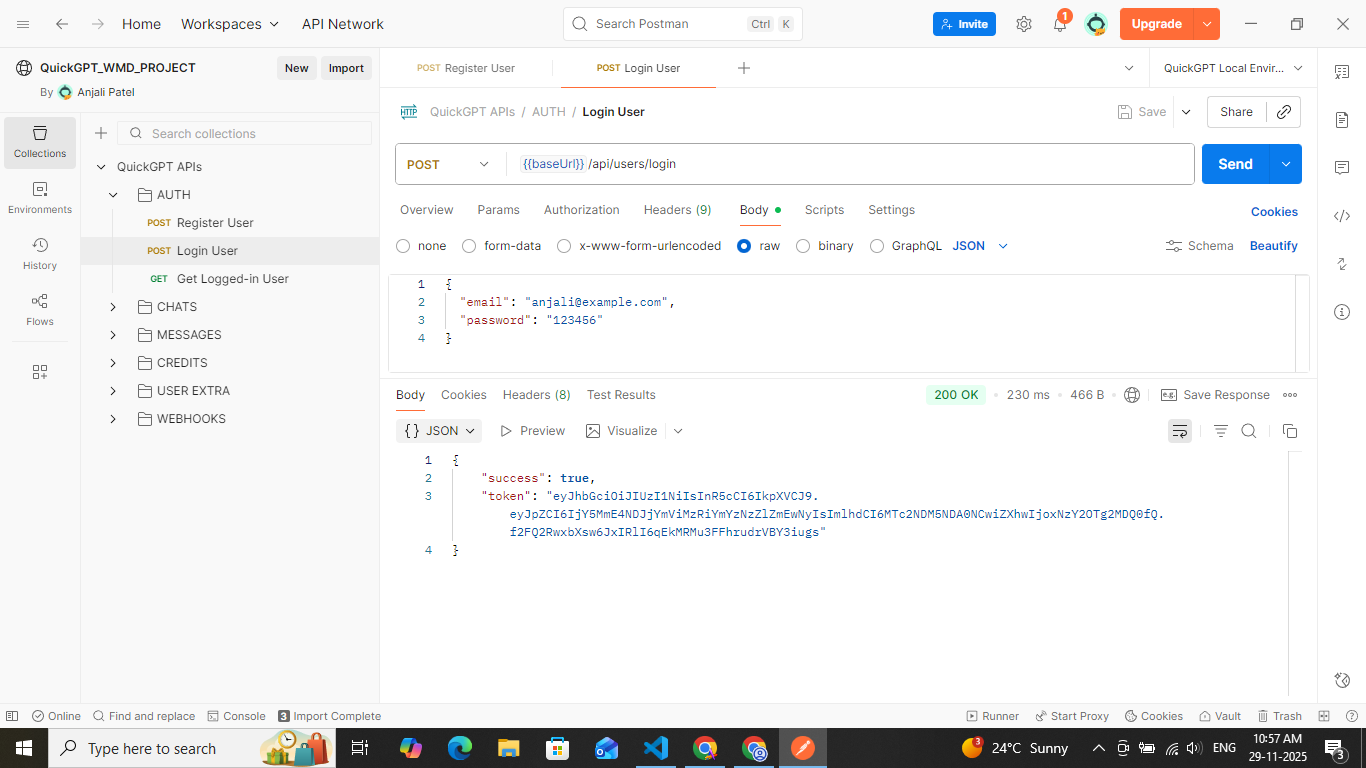
# System Snapshots

# Testing Snapshots

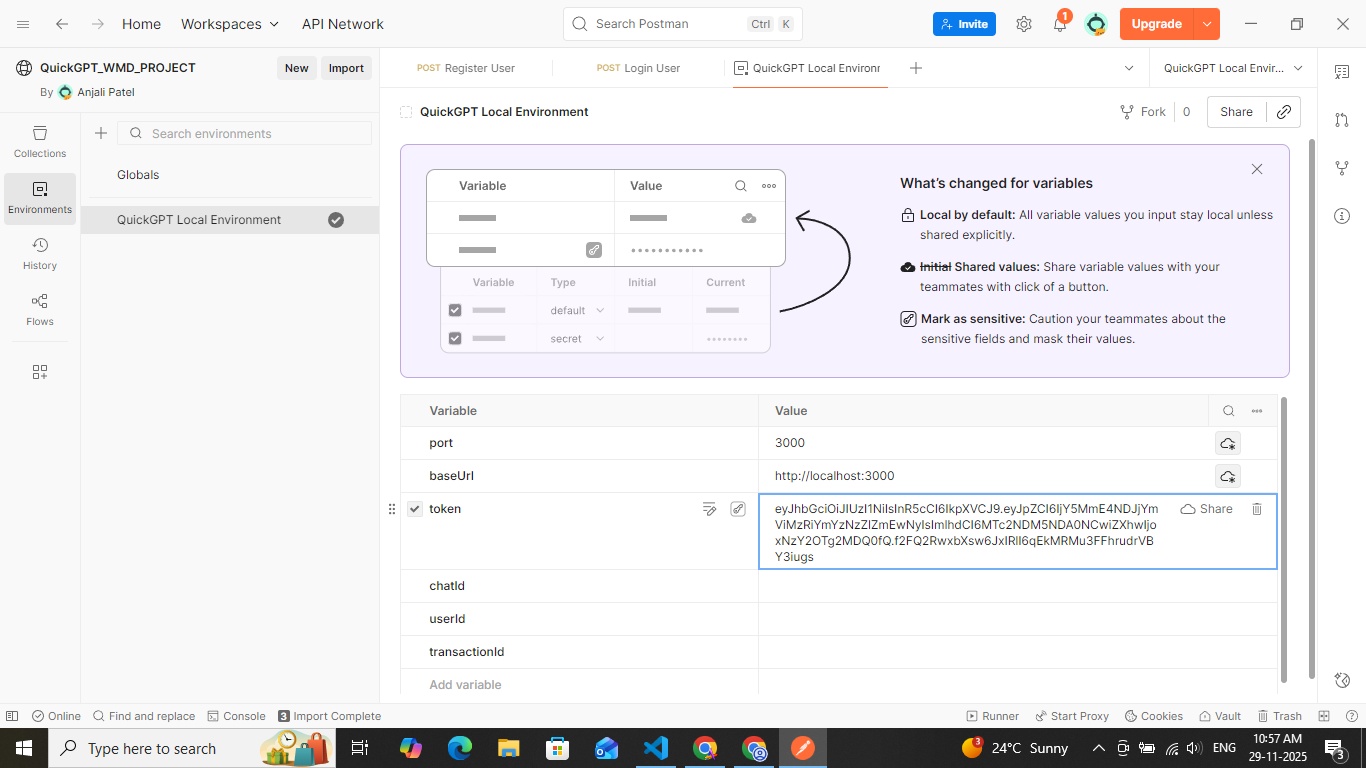
***[01] register user API***

******

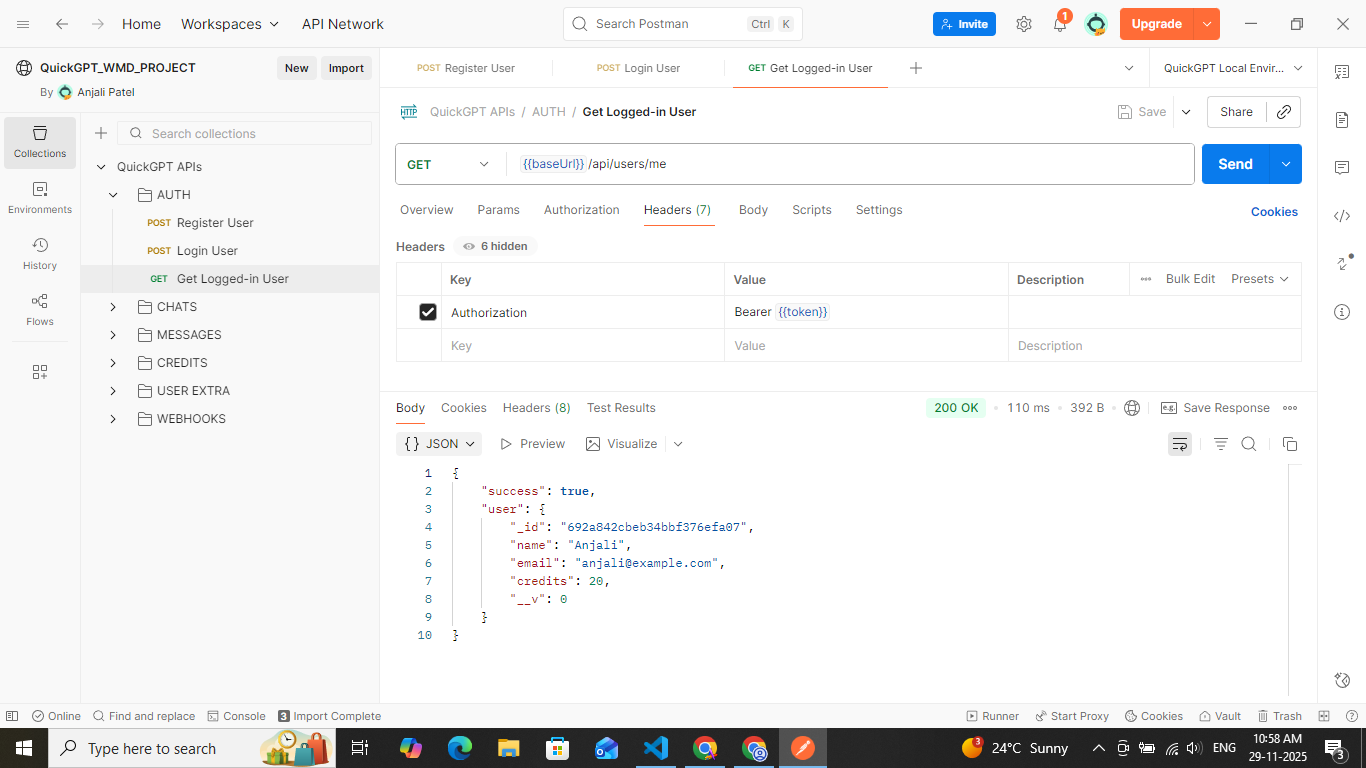
***[02] login user API***

******

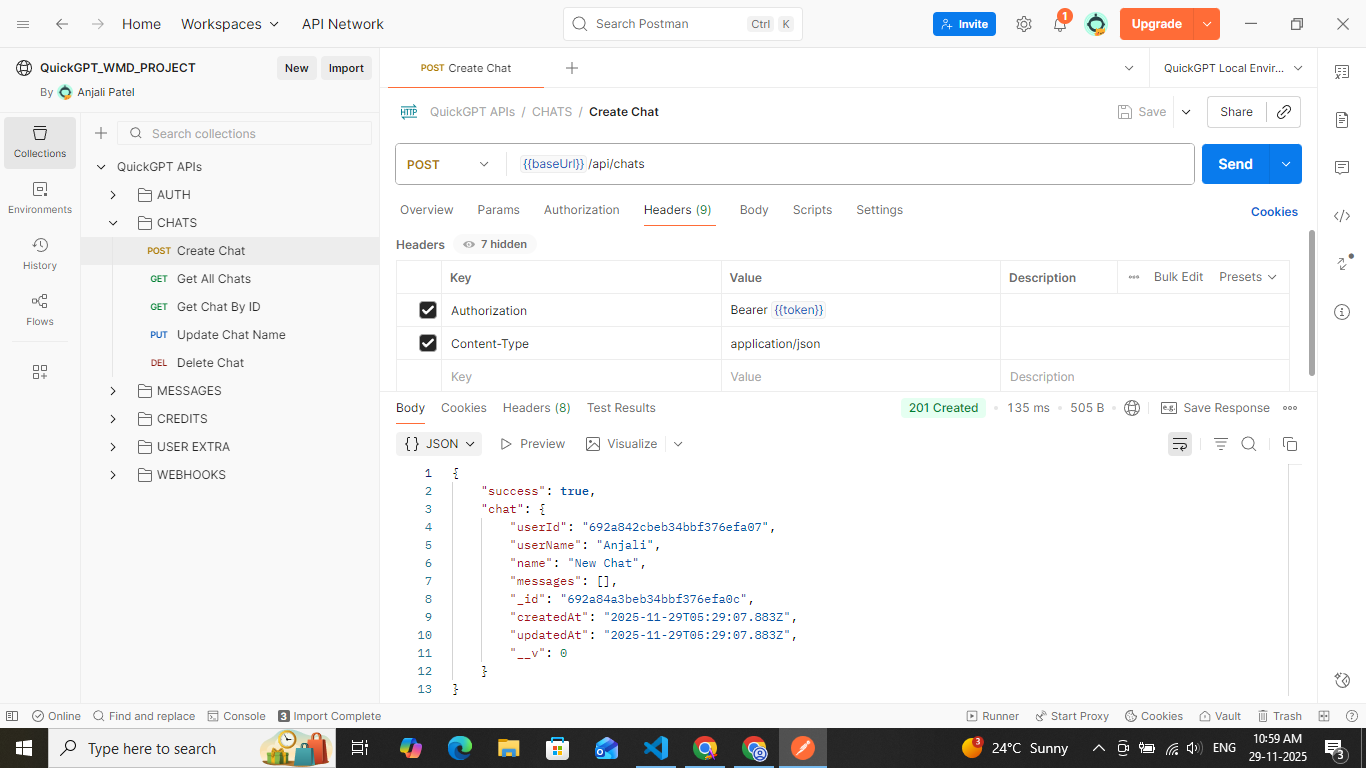
***[03] adding token to environment variable***

******

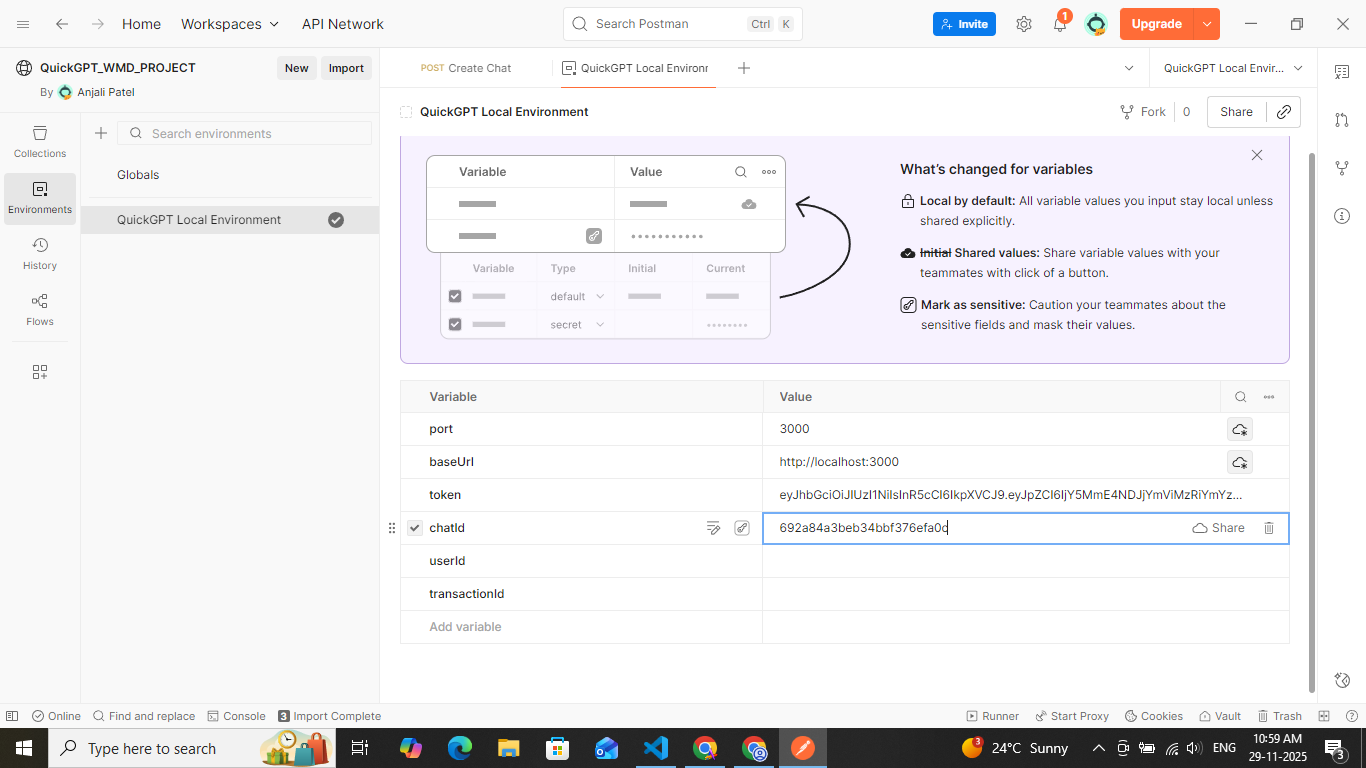
***[04] get logged-in user API***

******

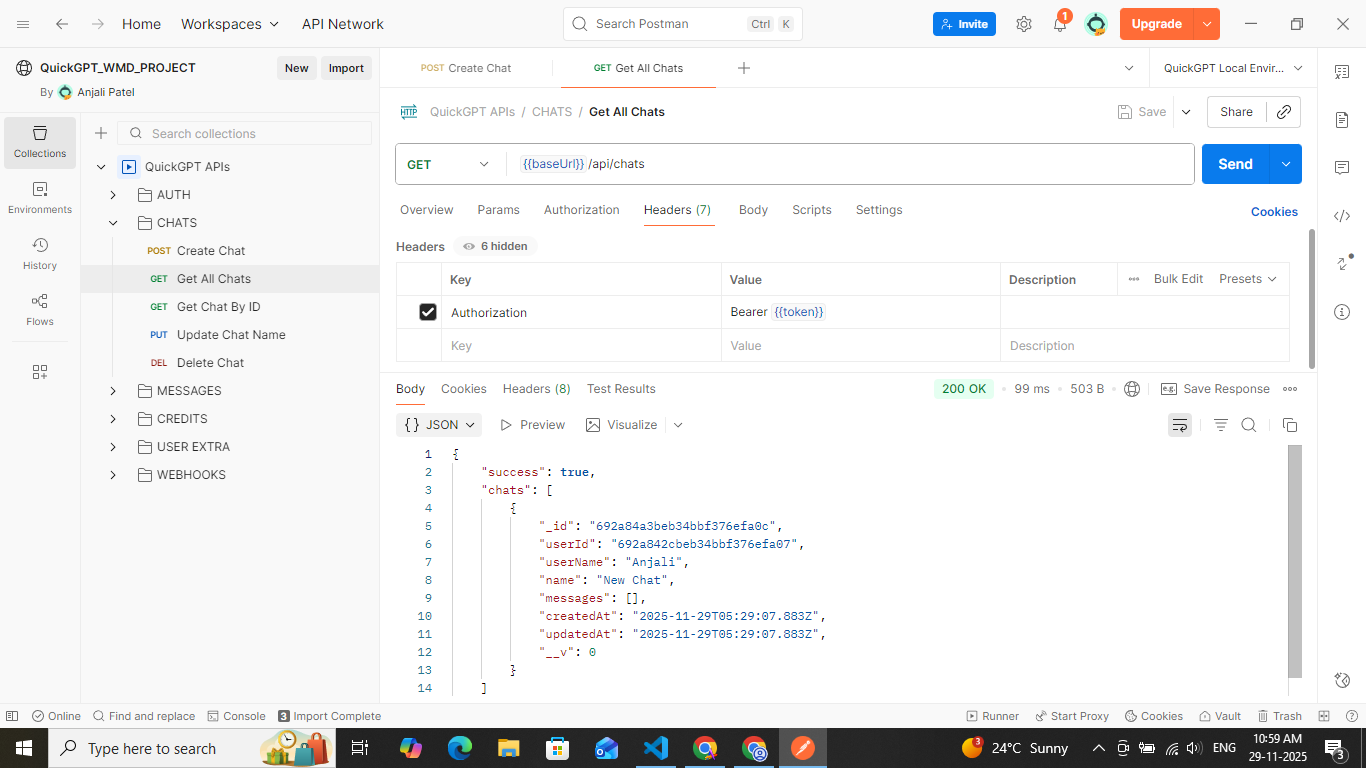
***[05] create chat API***

******

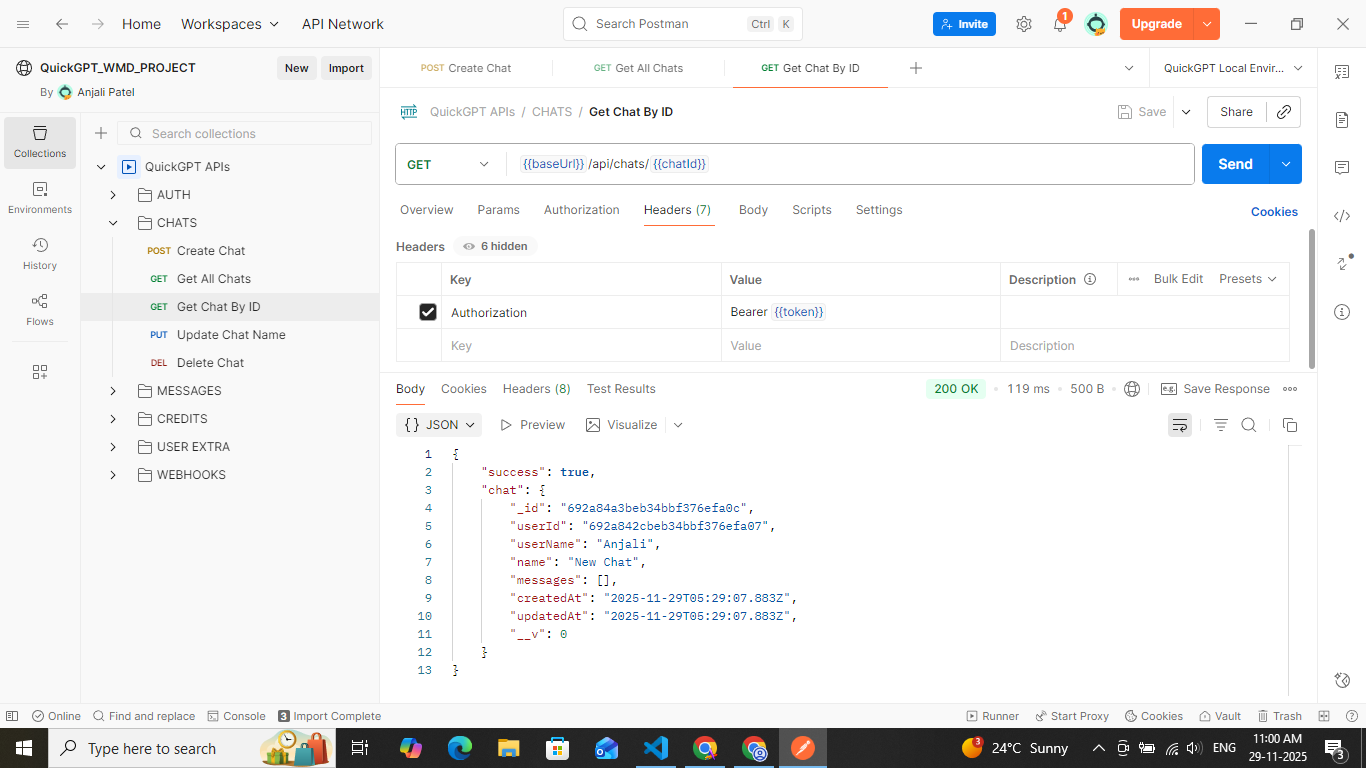
***[06] adding chatID-to environment variable***

******

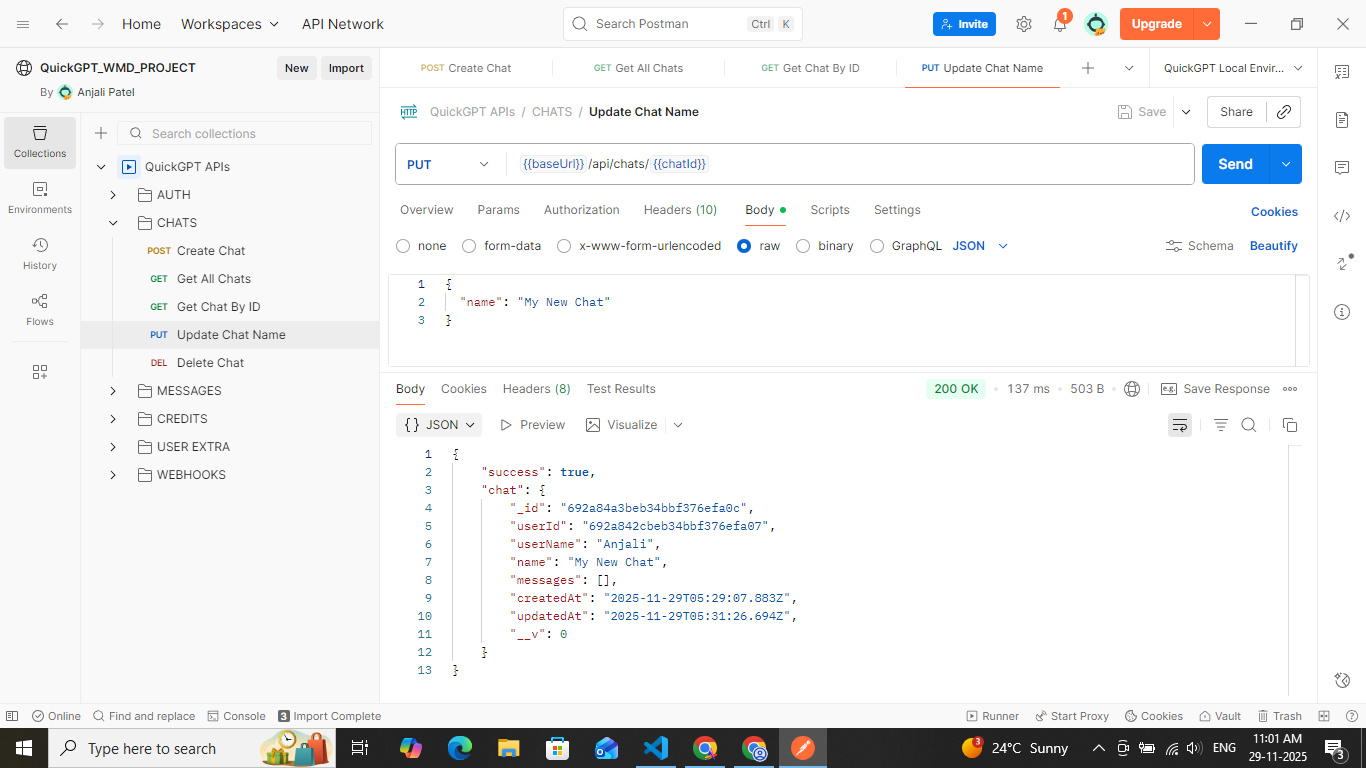
***[07] get all chats API***

******

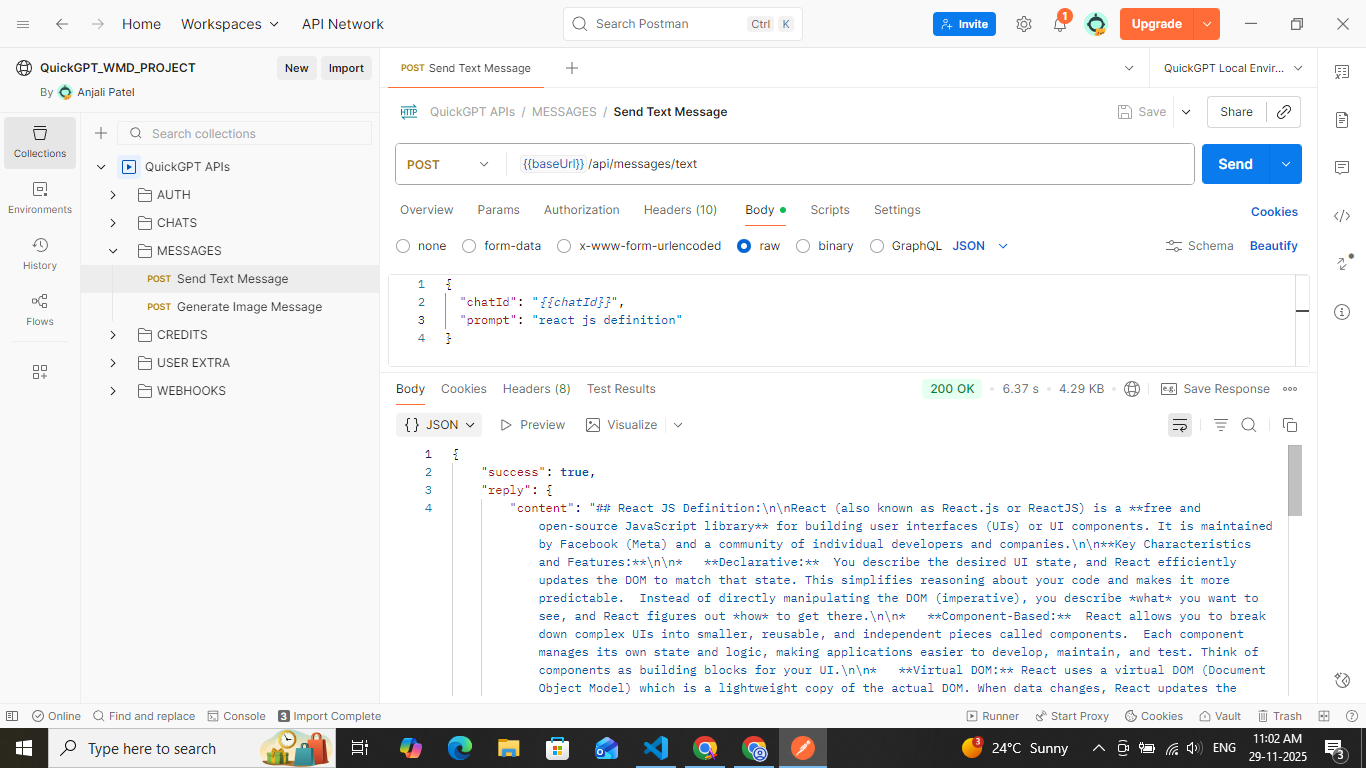
***[08] get chat by ID***

******

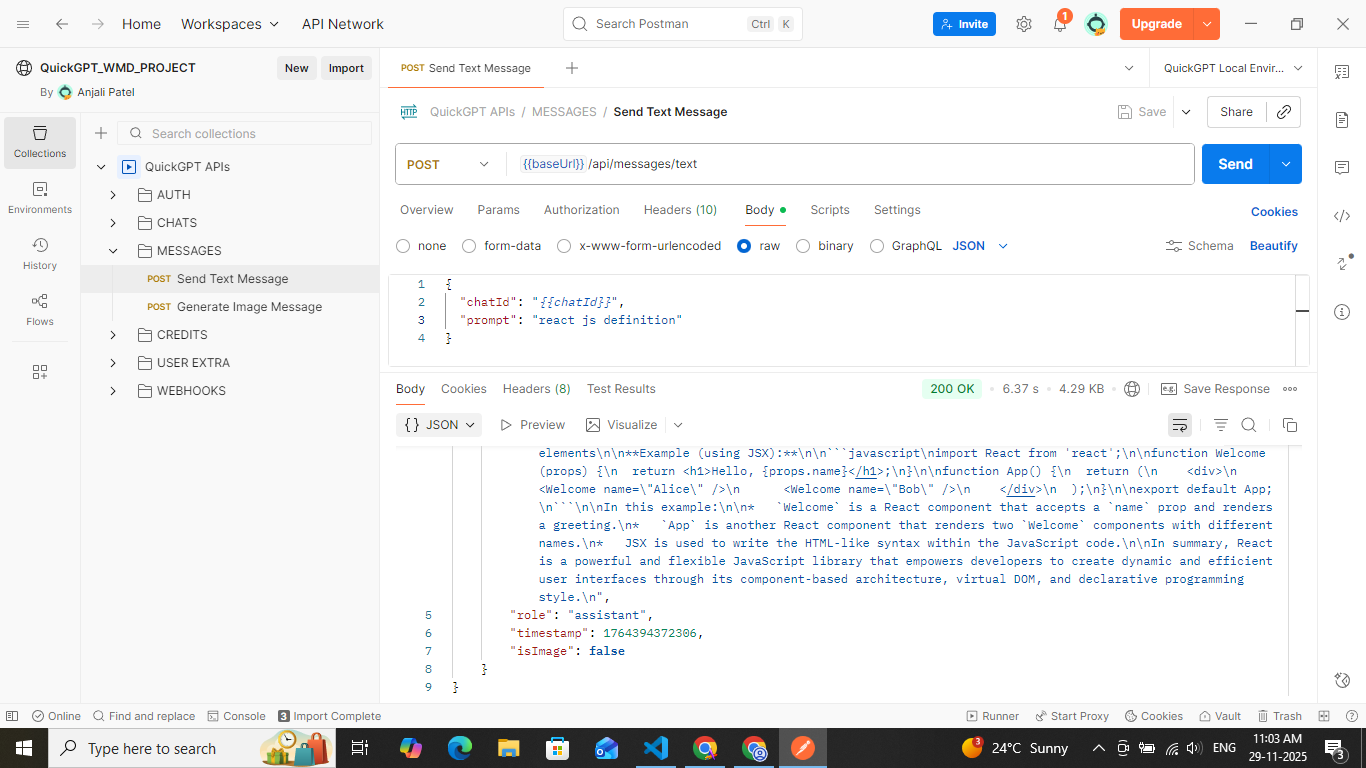
***[09] update chat name***

******

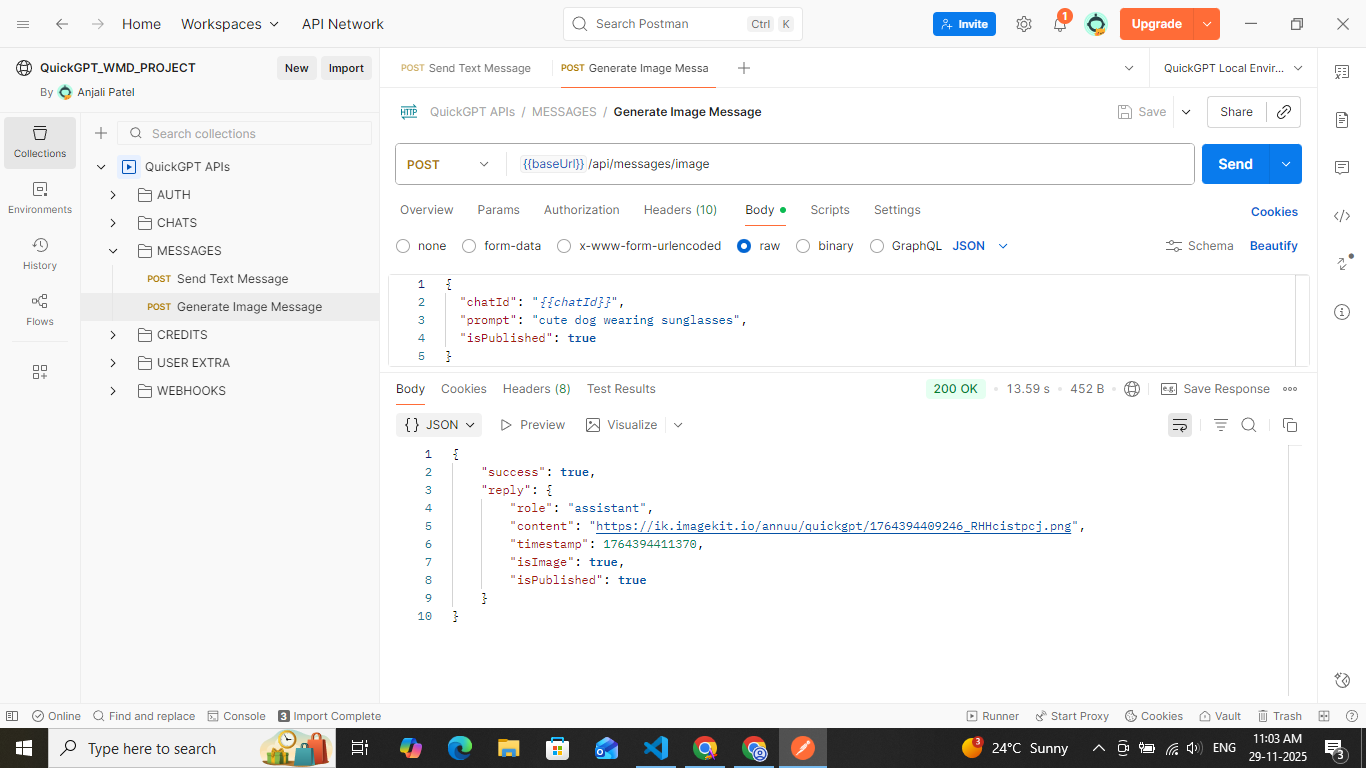
***[10] send text message***

******

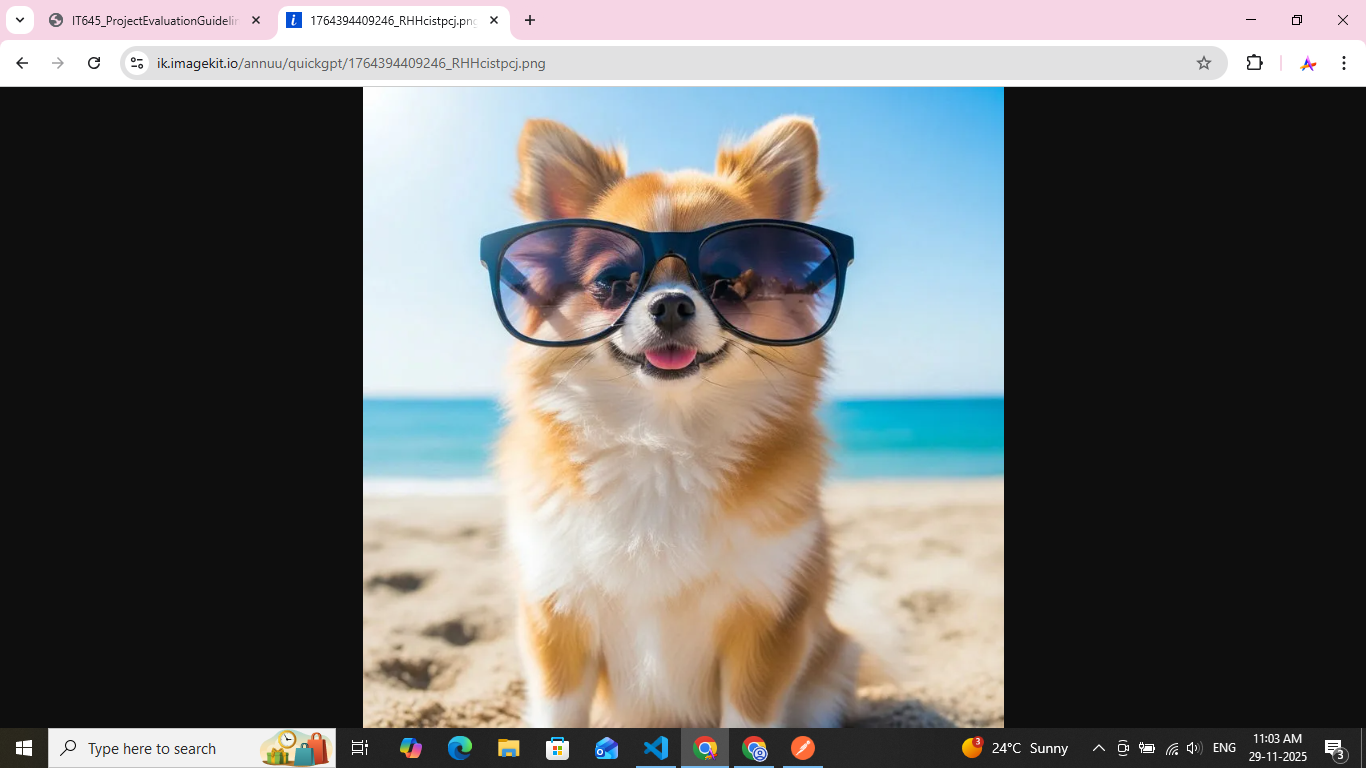
***[11] send text message API 2***

******

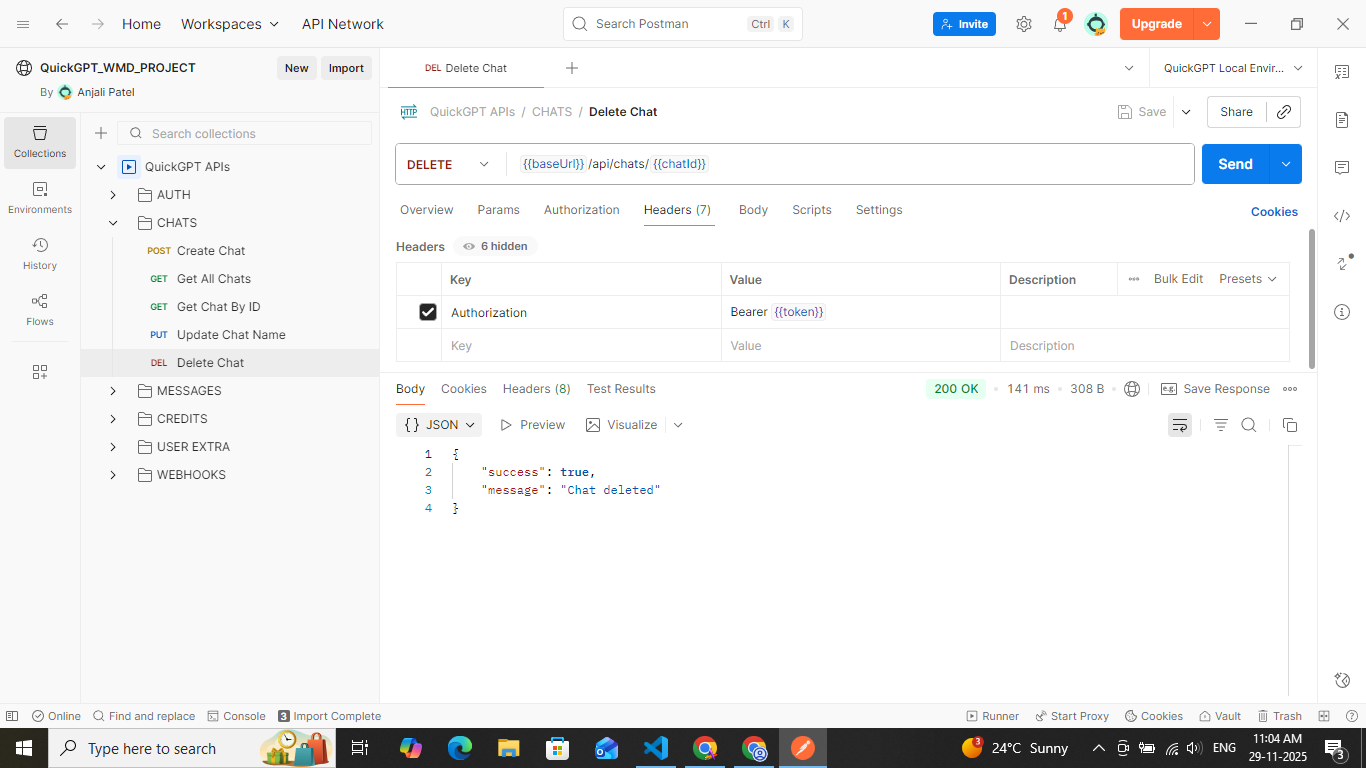
***[12] generate image message***

******

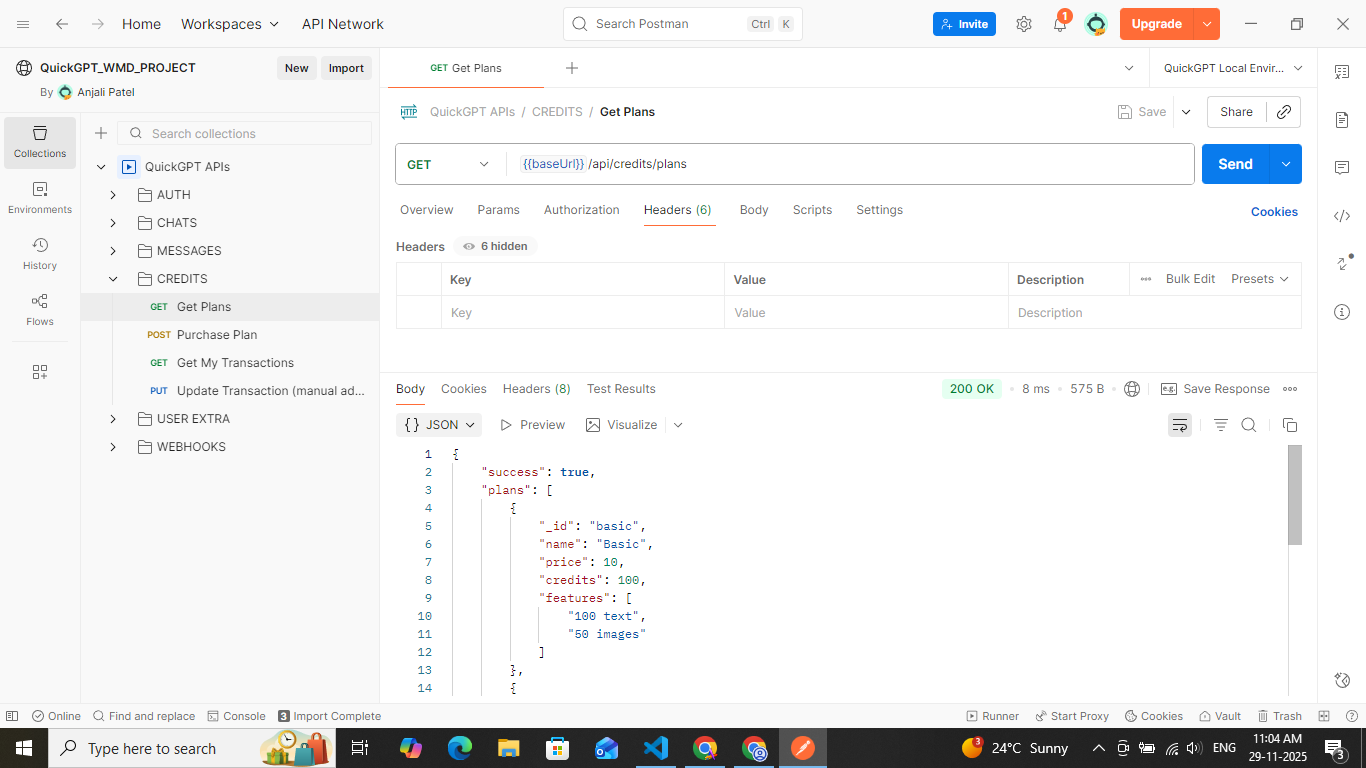
***[13] getting generated image***

******

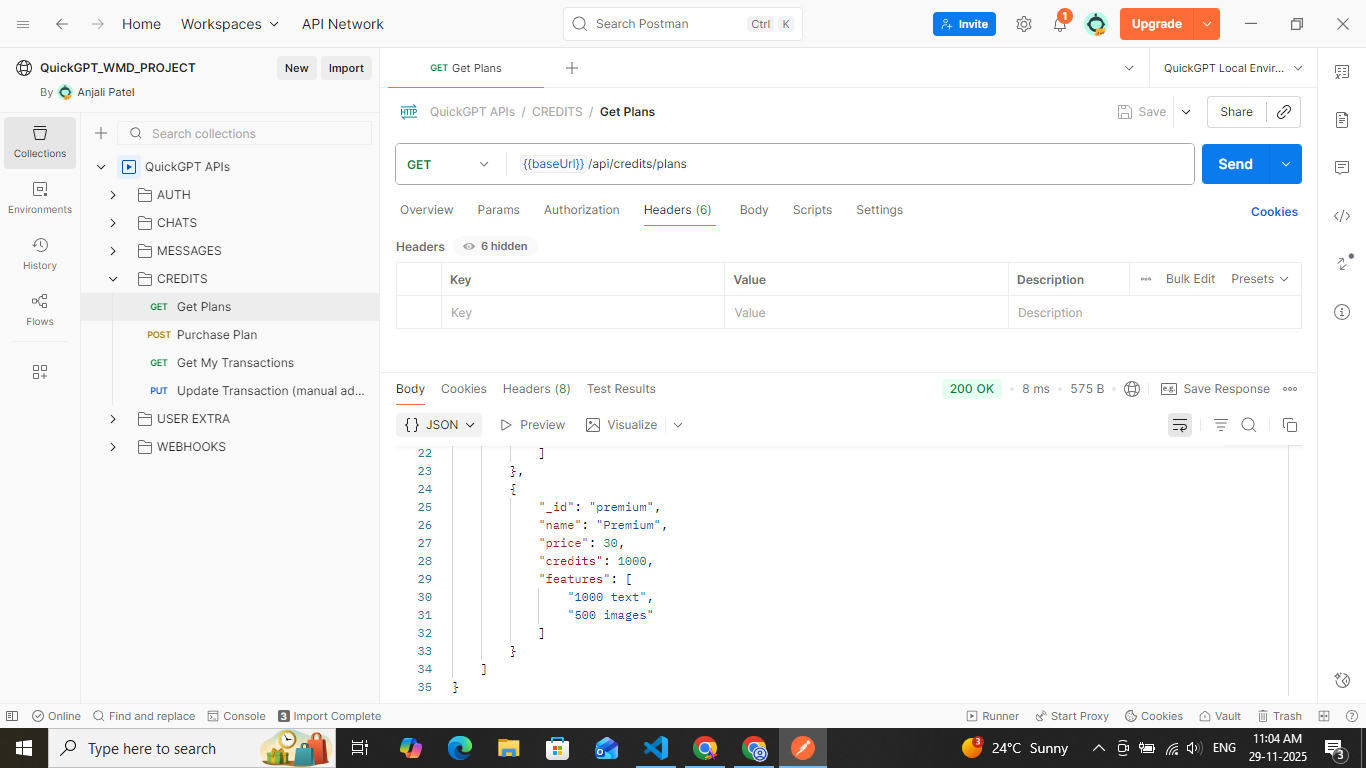
***[14] delete chat API***

******

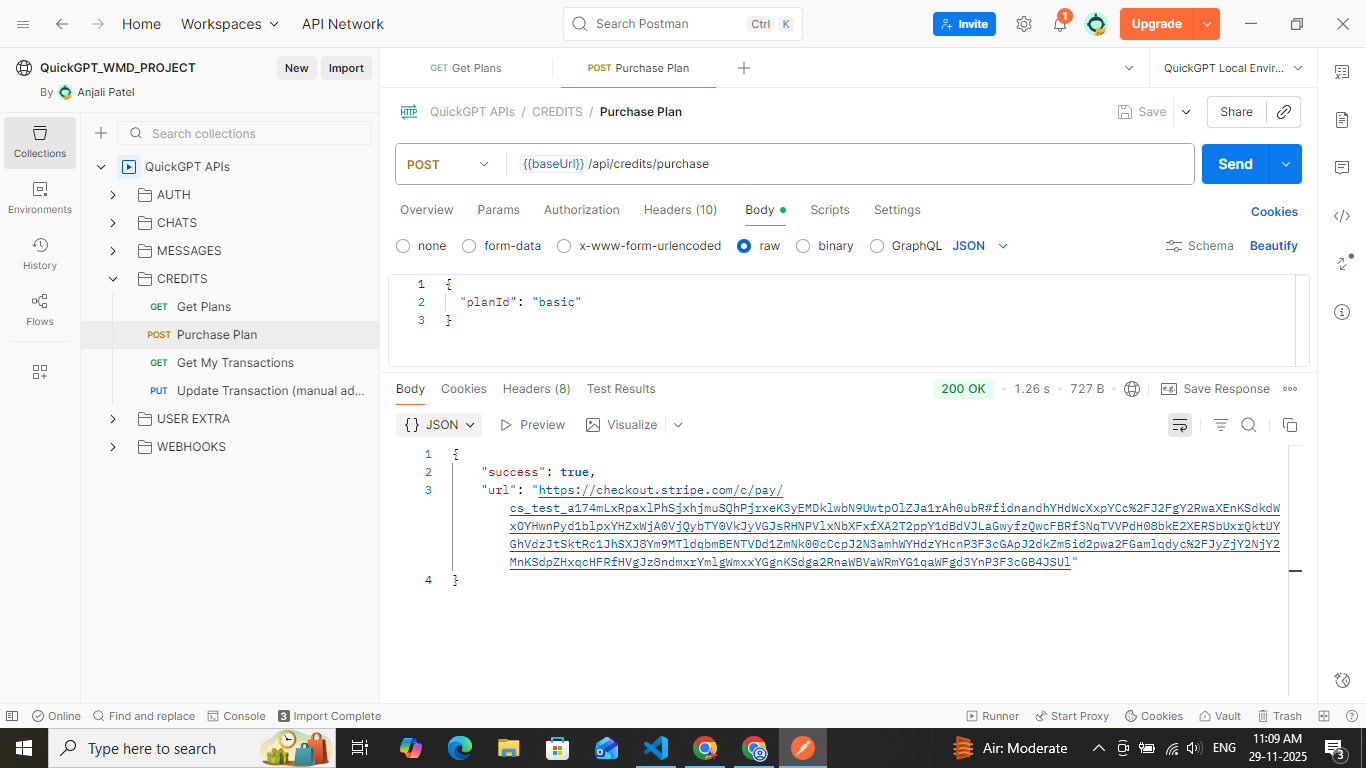
***[15] get plans API***

******

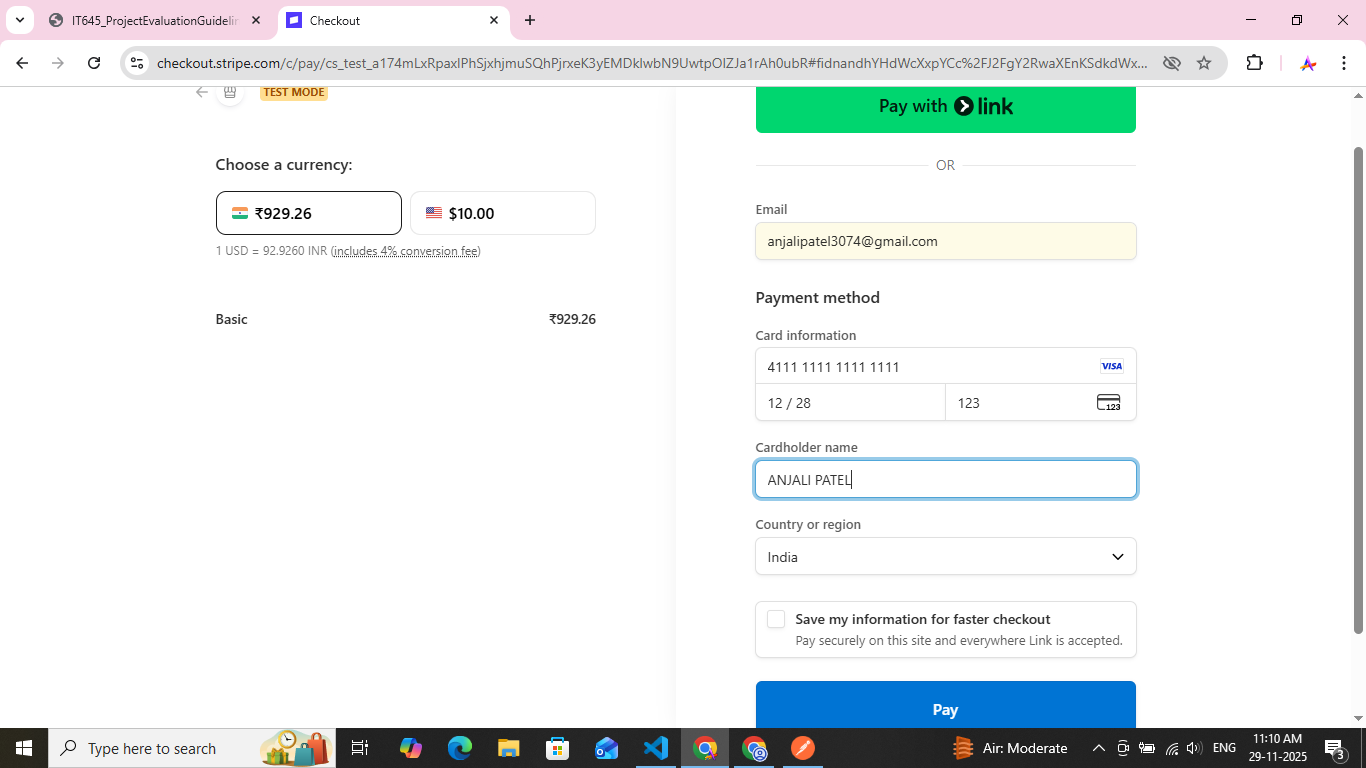
***[16] get plans API 2***

******

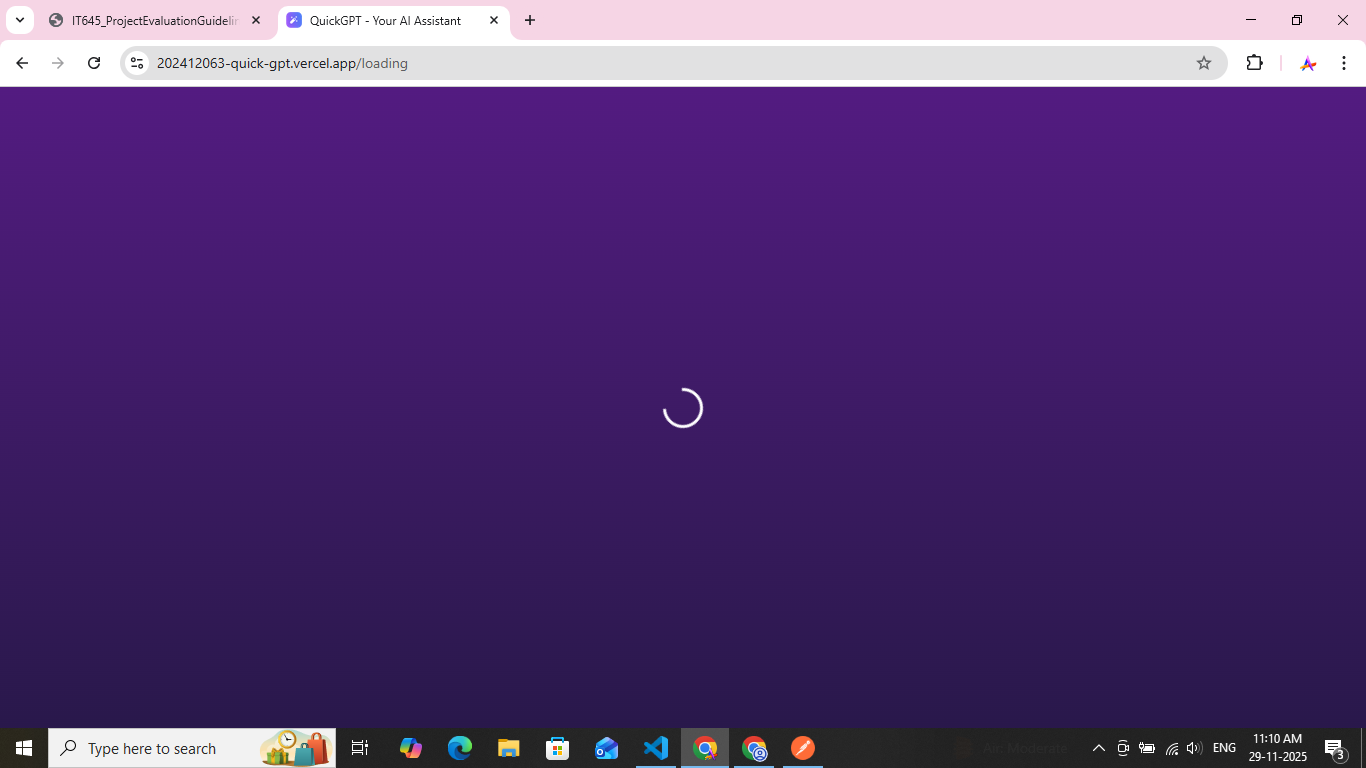
***[17] purchase plan***

******

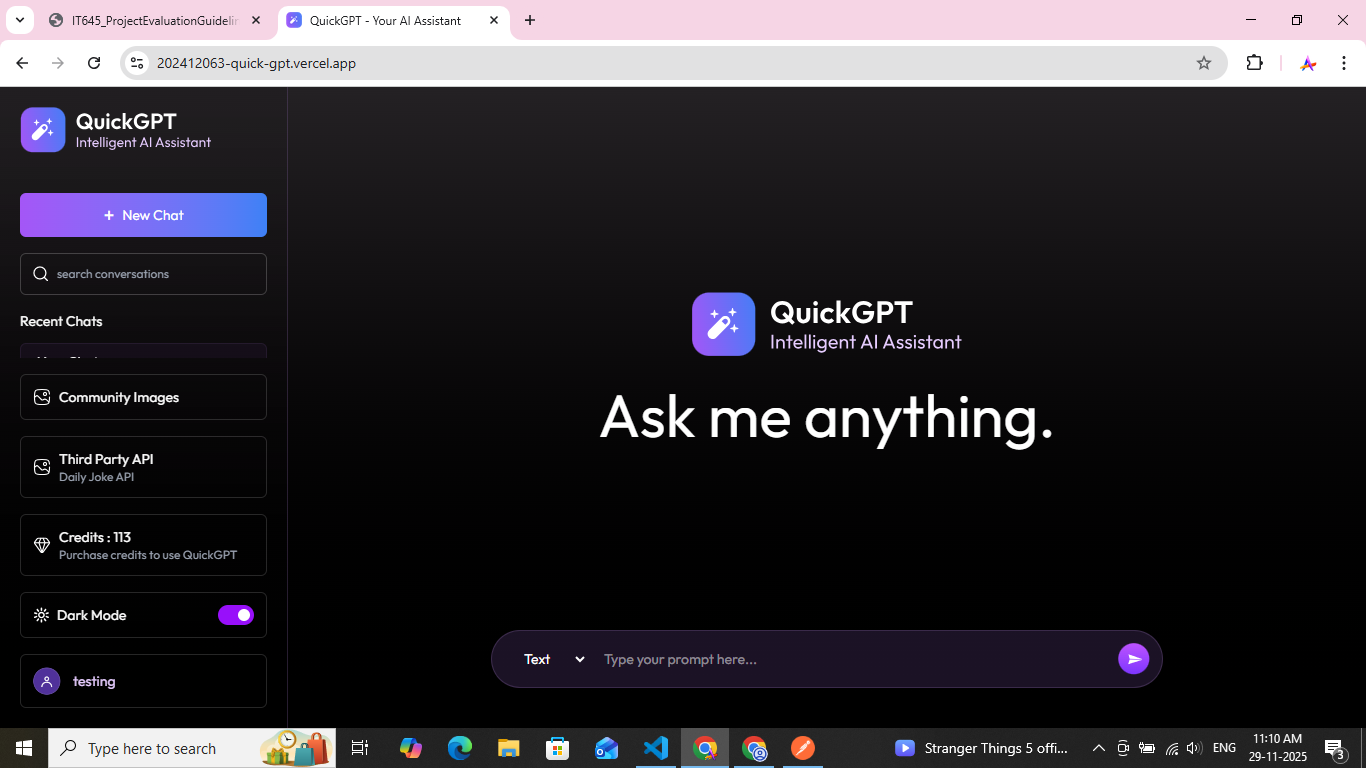
***[18] payment processing***

******

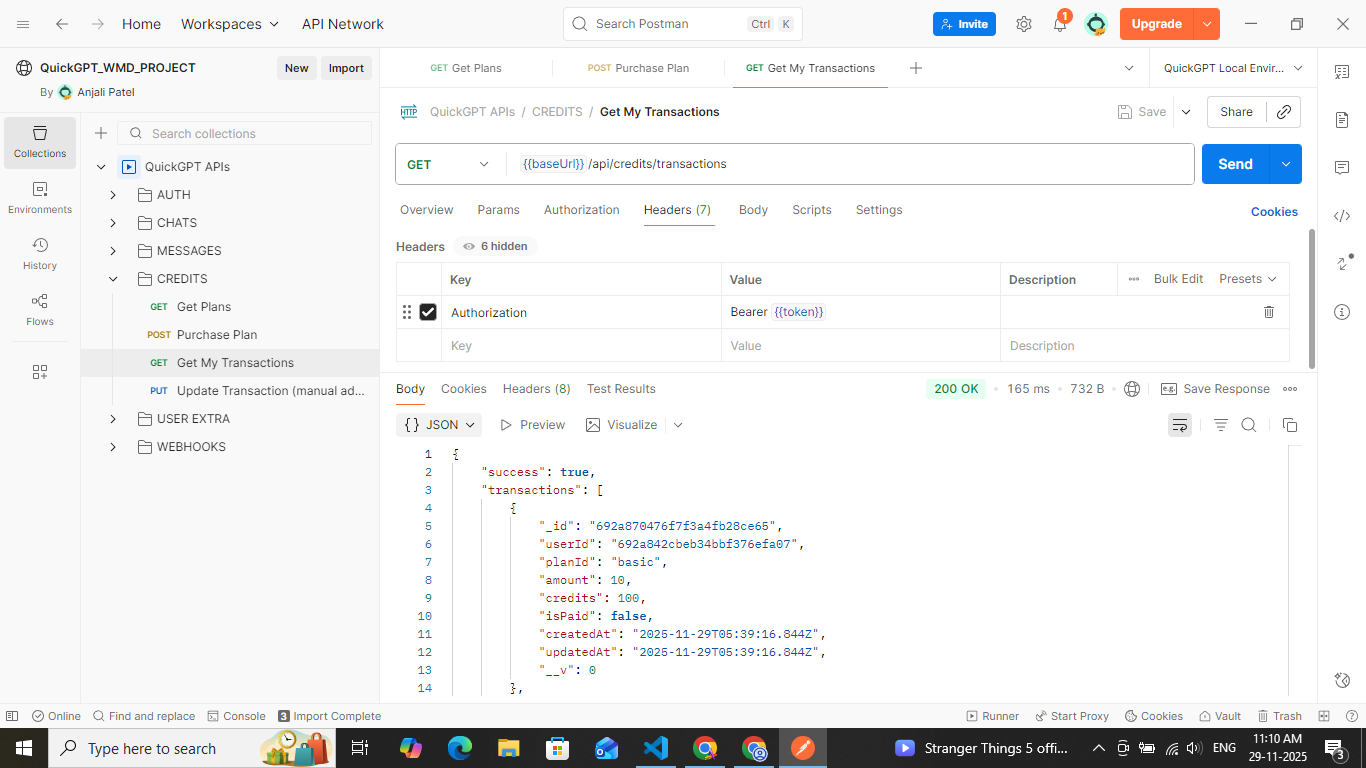
***[19] redirected after sucessful payment***

******

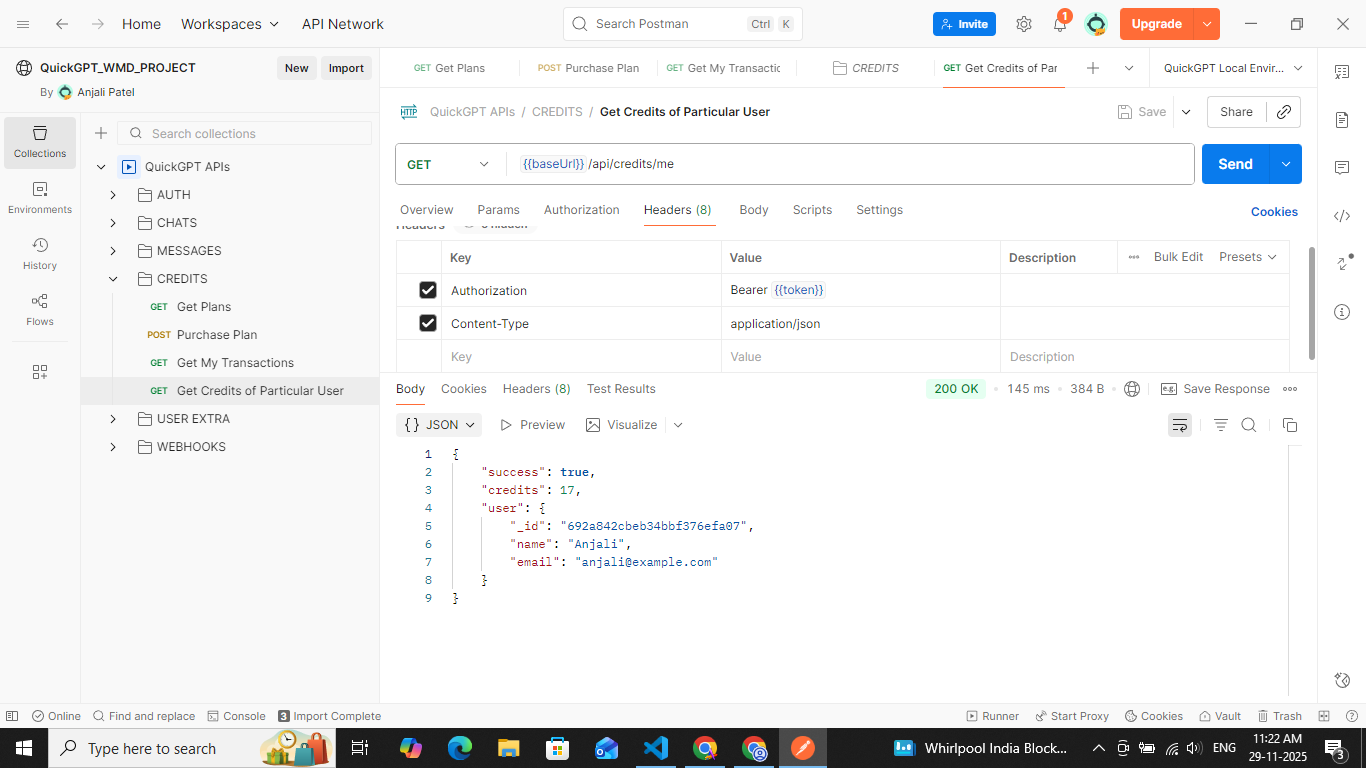
***[20] updated credits***

******

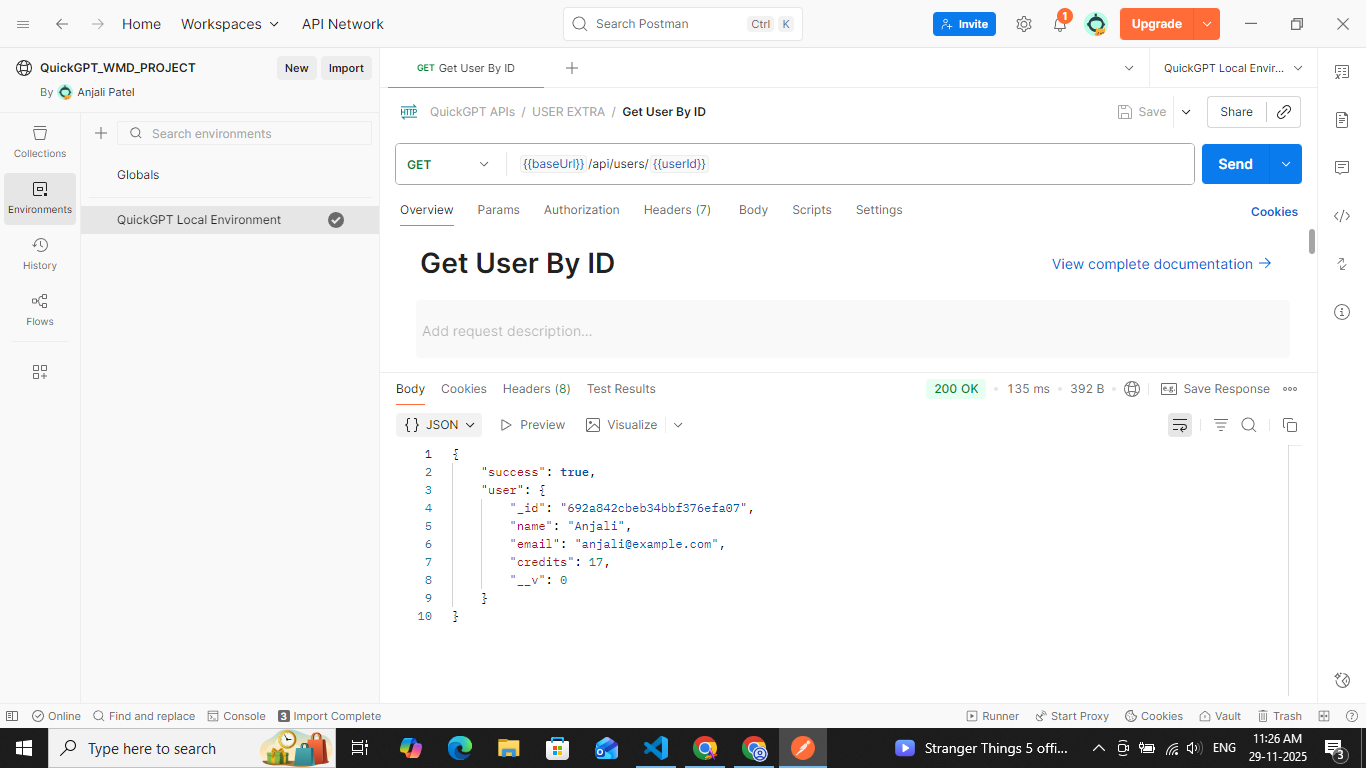
***[21] get my transactions API***

******

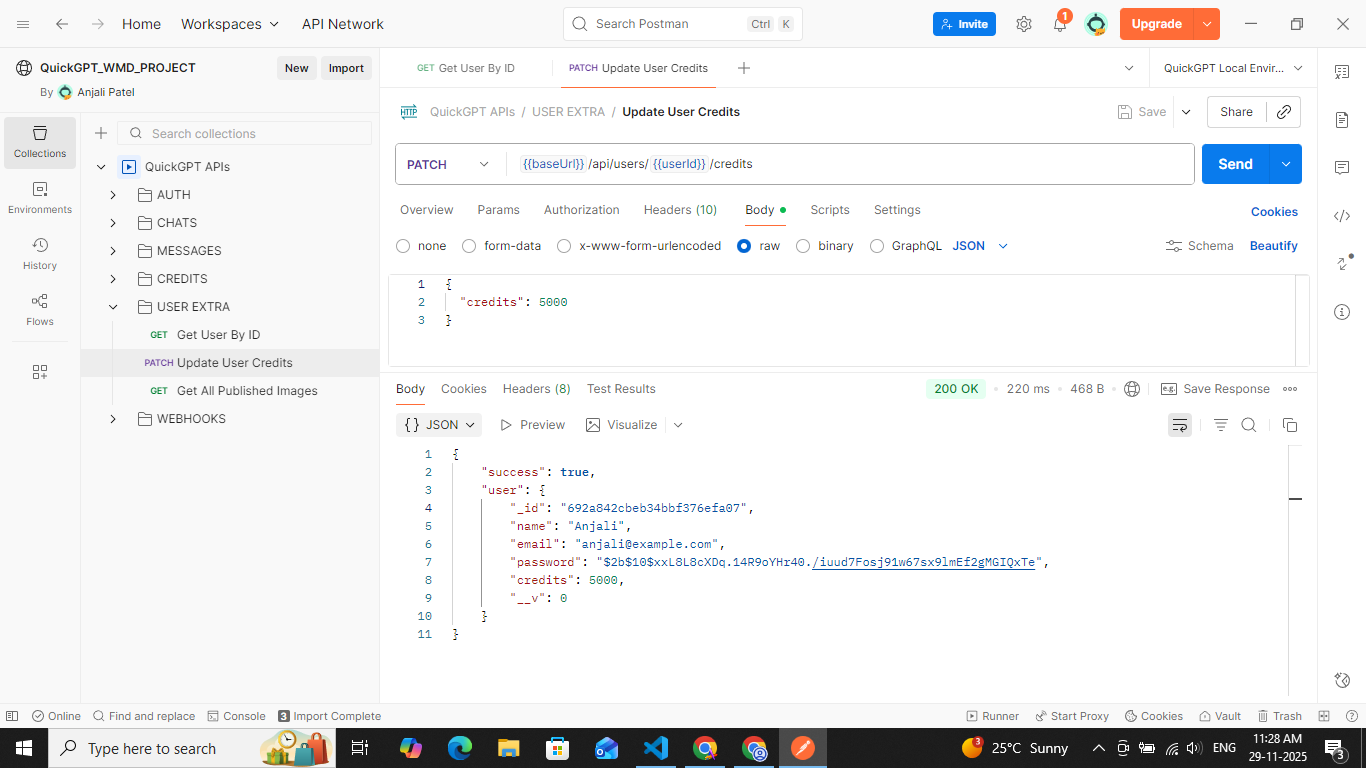
***[22] get credits of particular user API***

******

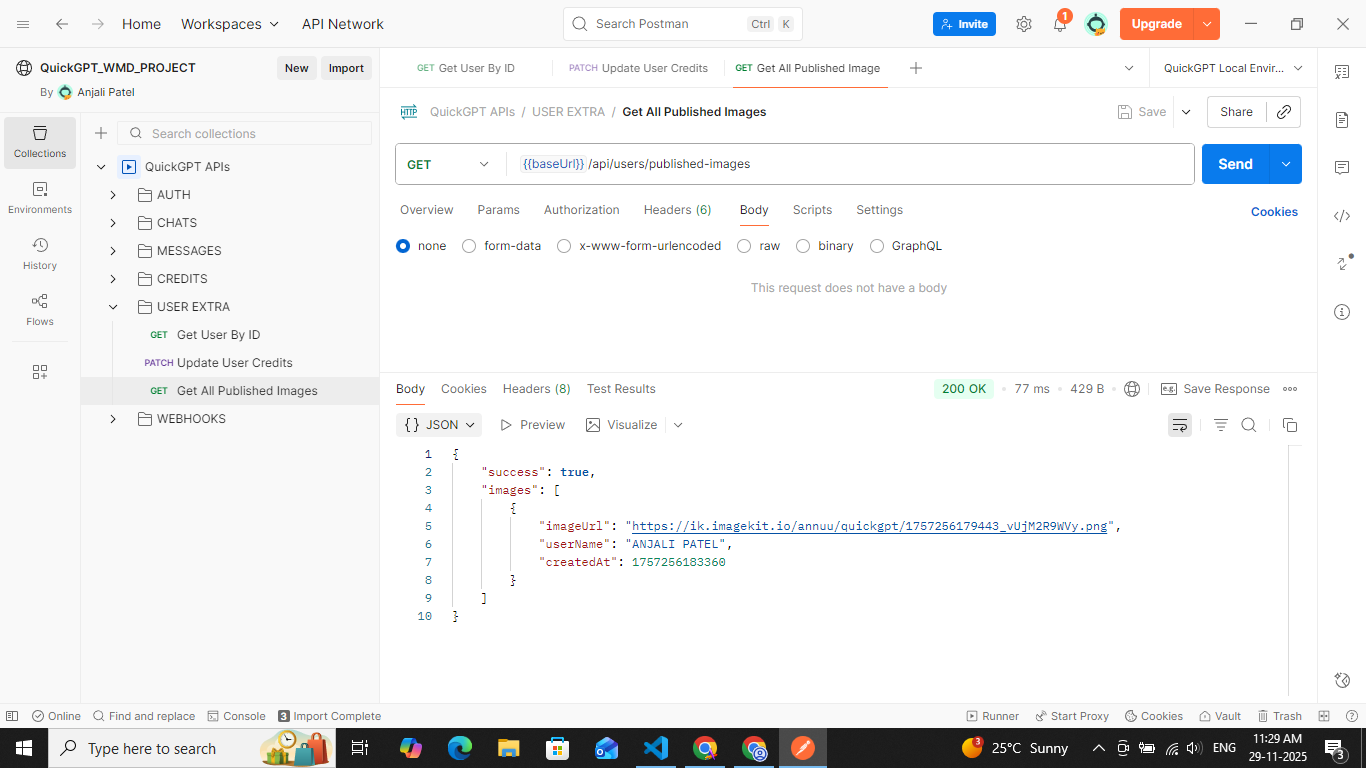
***[23] get user by ID API***

******

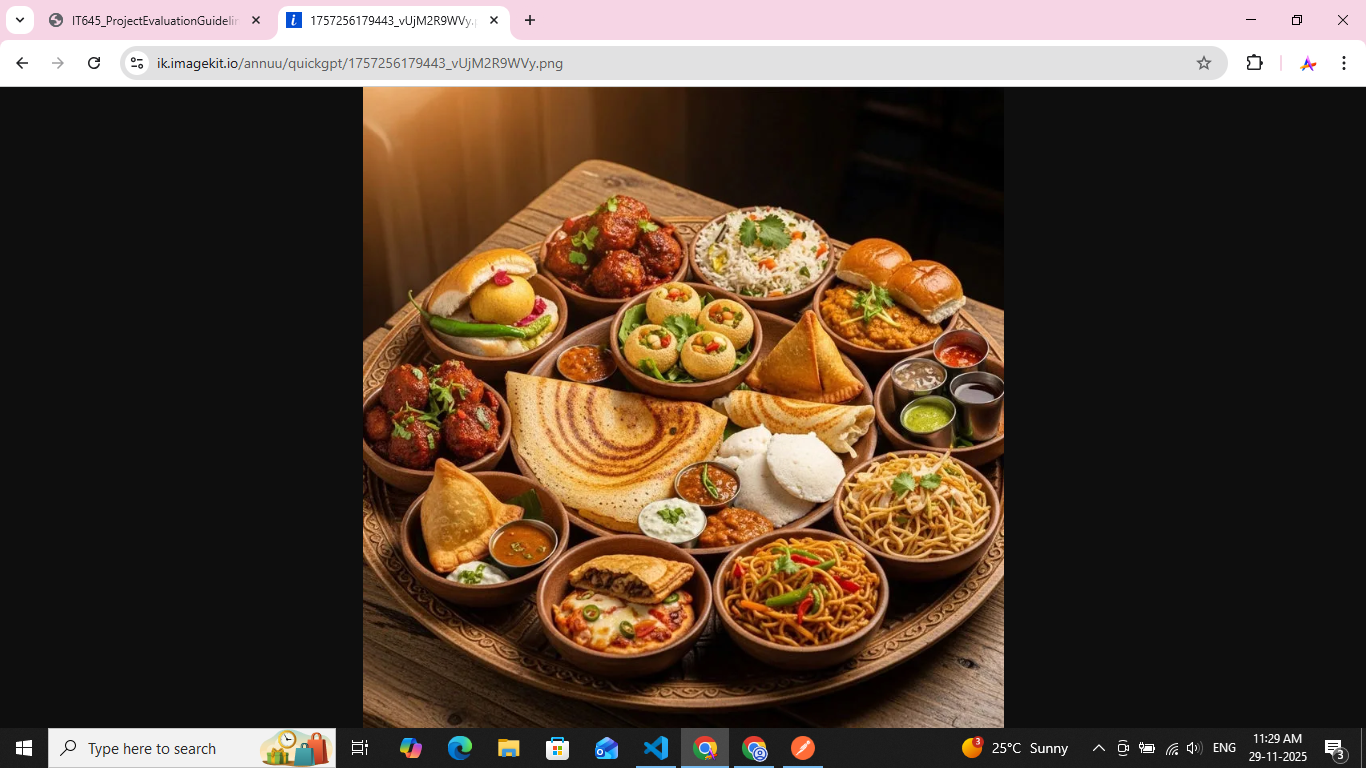
***[24] update user credits API***

******

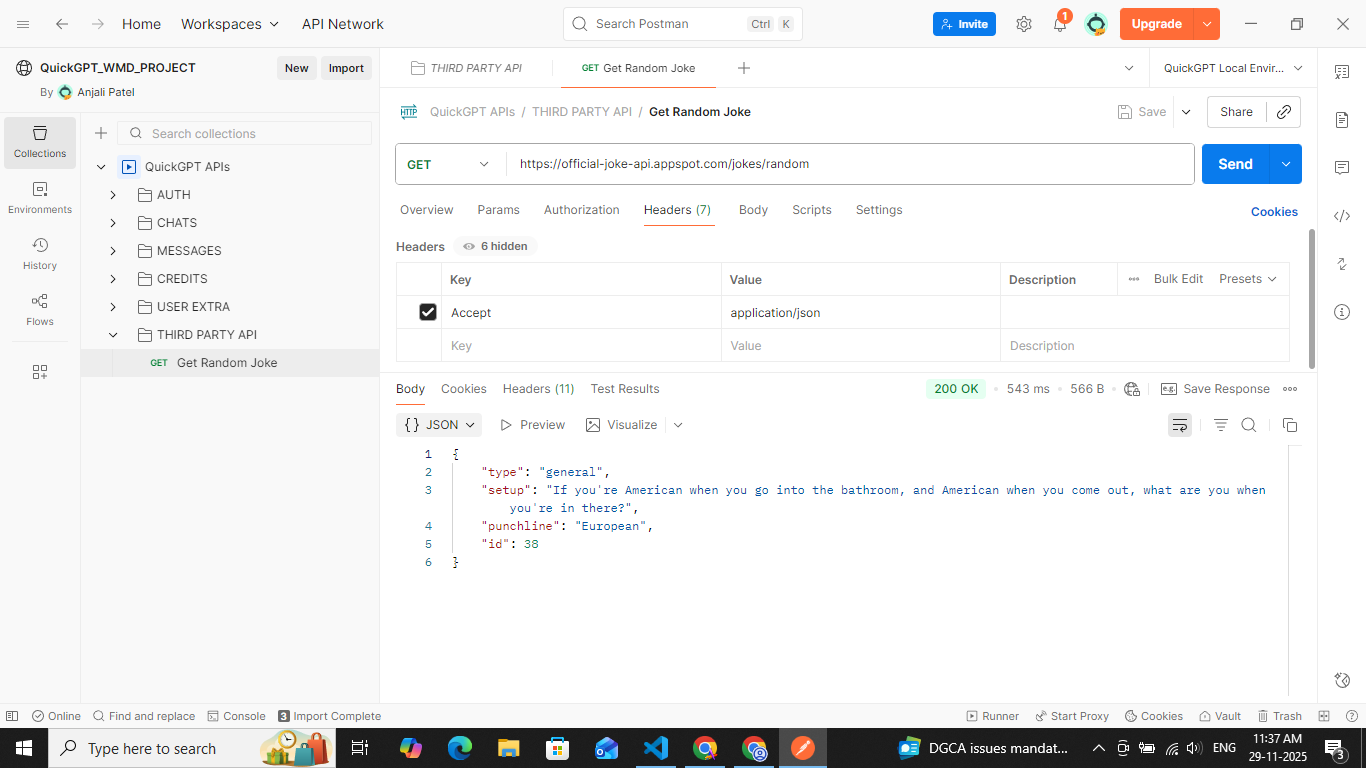
***[25] get all published images API***

******

***[26] getting published image from link***

******

***[27] get random joke third party API***

******

# Third-Party libraries & tools used

* **Frontend**
  + React
  + Tailwind CSS
  + React router
  + Context API
  + axios
* **Backend**
  + express
  + mongoose
  + cors
  + jsonwebtoken
  + bcryptjs
  + morgan
  + dotenv
* **Payments**
  + stripe
* **AI**
  + Google Gemini
* **Other tools**
  + Postman : for API testing
  + npm
  + Git + GitHub
  + VS code editor

# Links

* **GitHub repository link**
  + <https://github.com/AP-anjali/QuickGPT_WMD_PROJECT>
* **Deployed website link**
  + **Frontend :** <https://202412063-quick-gpt.vercel.app/>
  + **Backend :** <https://quick-gpt-server-plum.vercel.app/>