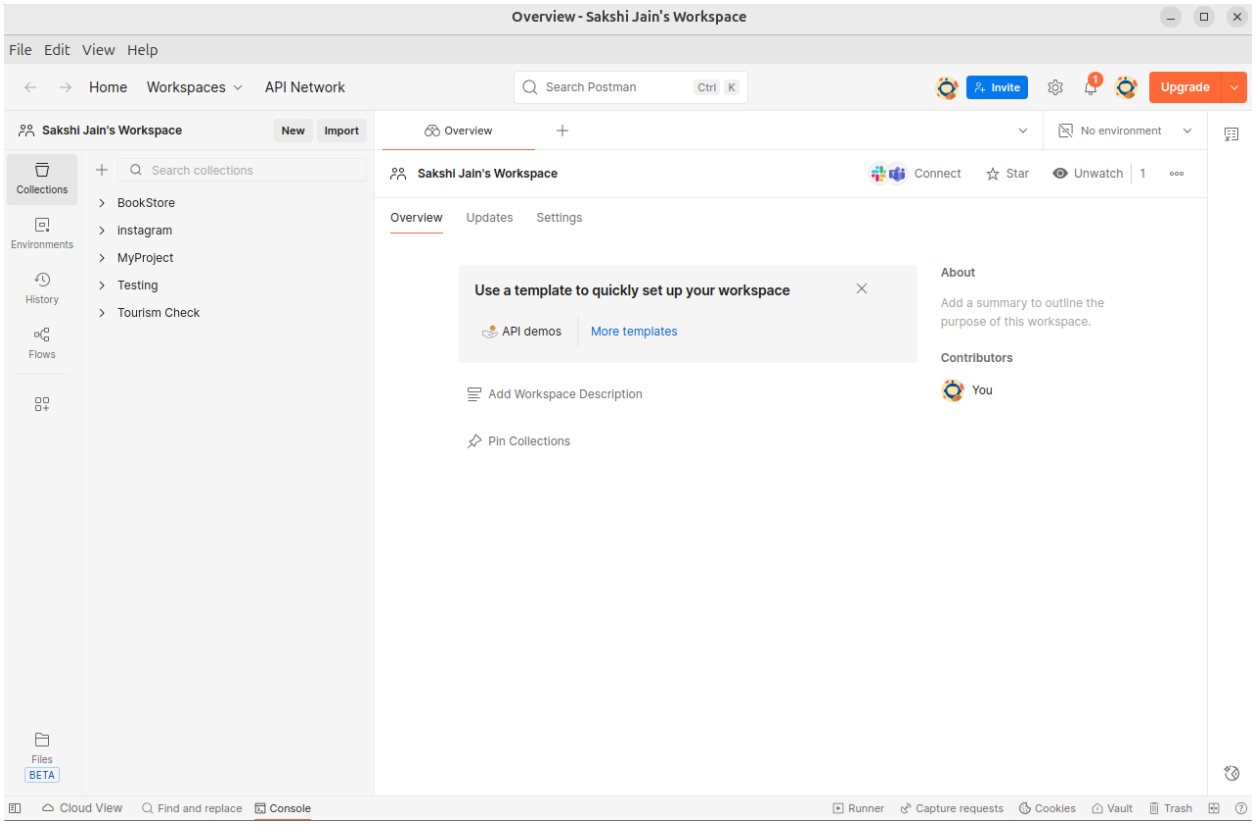
Task 1- Install and open Postman (or use an alternative like Hoppscotch).

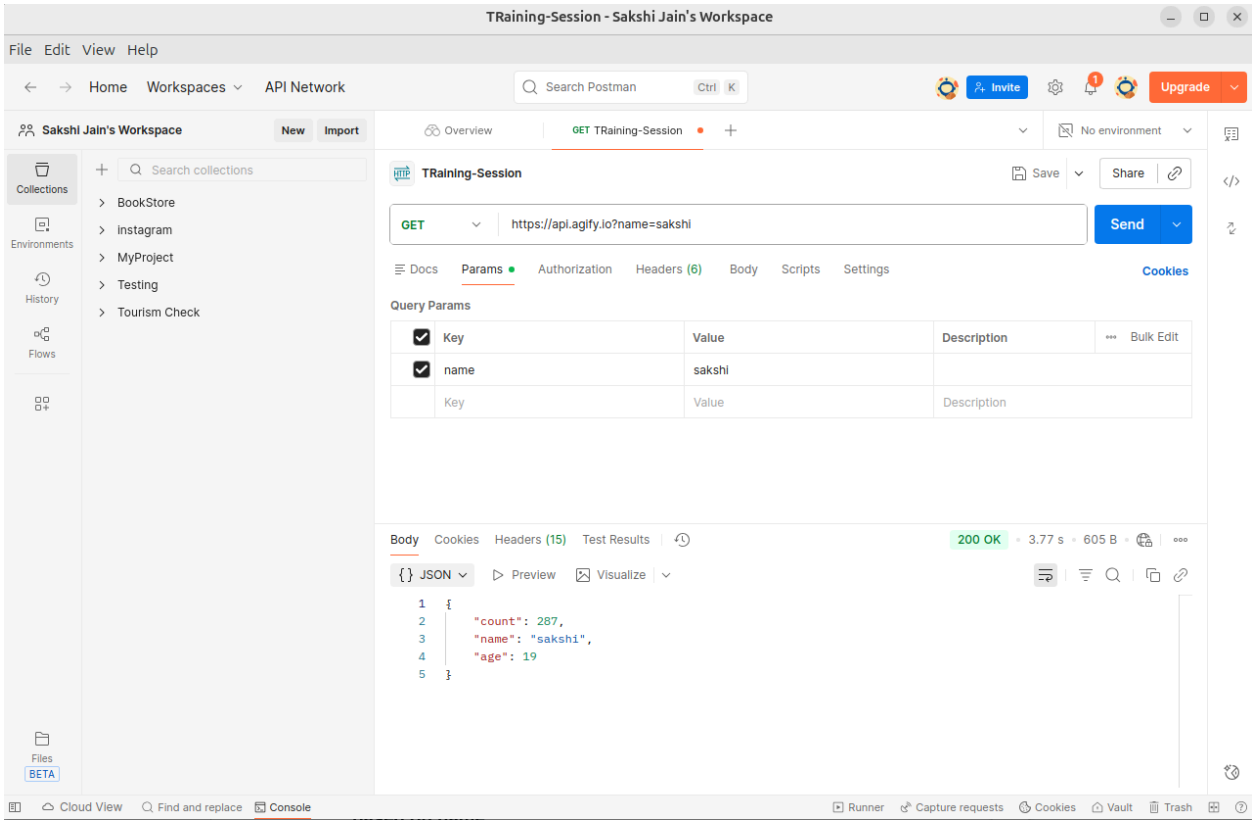
- **Will be using Postman**



Task -2] Make a GET request to a public API

(e.g., https://jsonplaceholder.typicode.com/posts/1).

- **I have researched about public apis on website - https://apipheny.io/free-api/**
- **Selected the public REST API for get request : Predicting the age based on name**
- **API URL:** https://api.agify.io?name=sakshi
- **Output:**

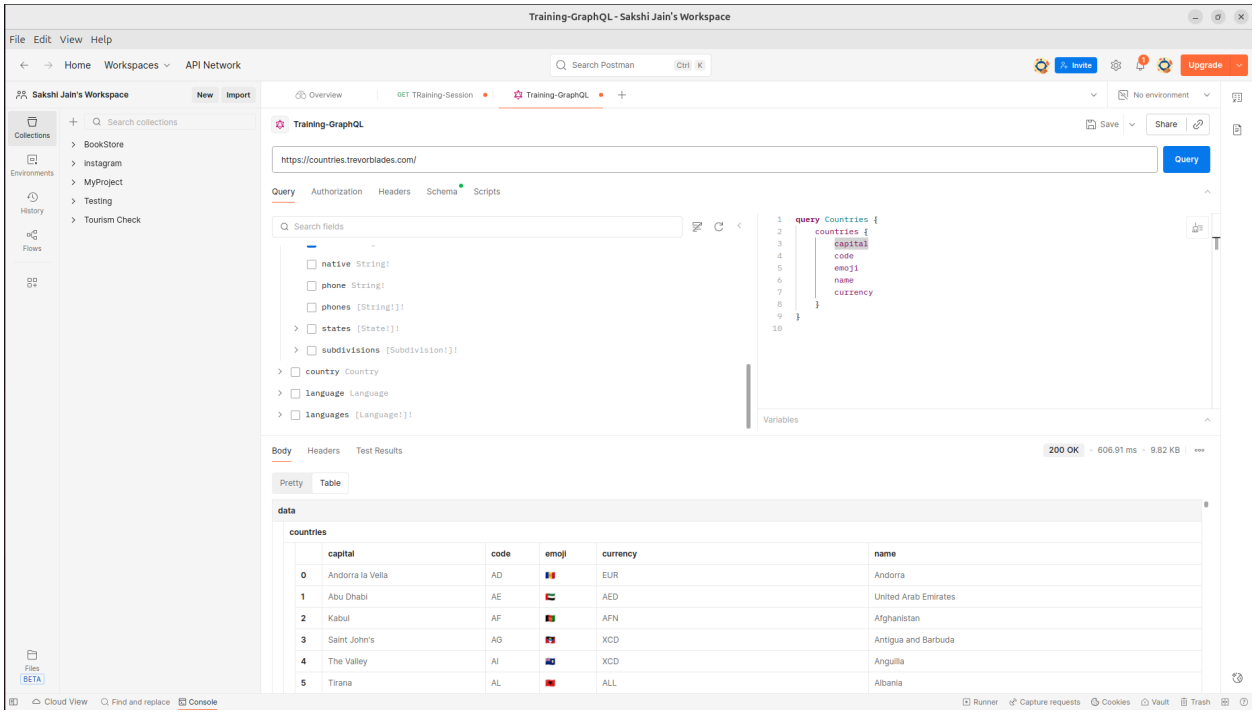


Now trying GraphQL Free public api:

I used : https://countries.trevorblades.com/

And Query Schema:

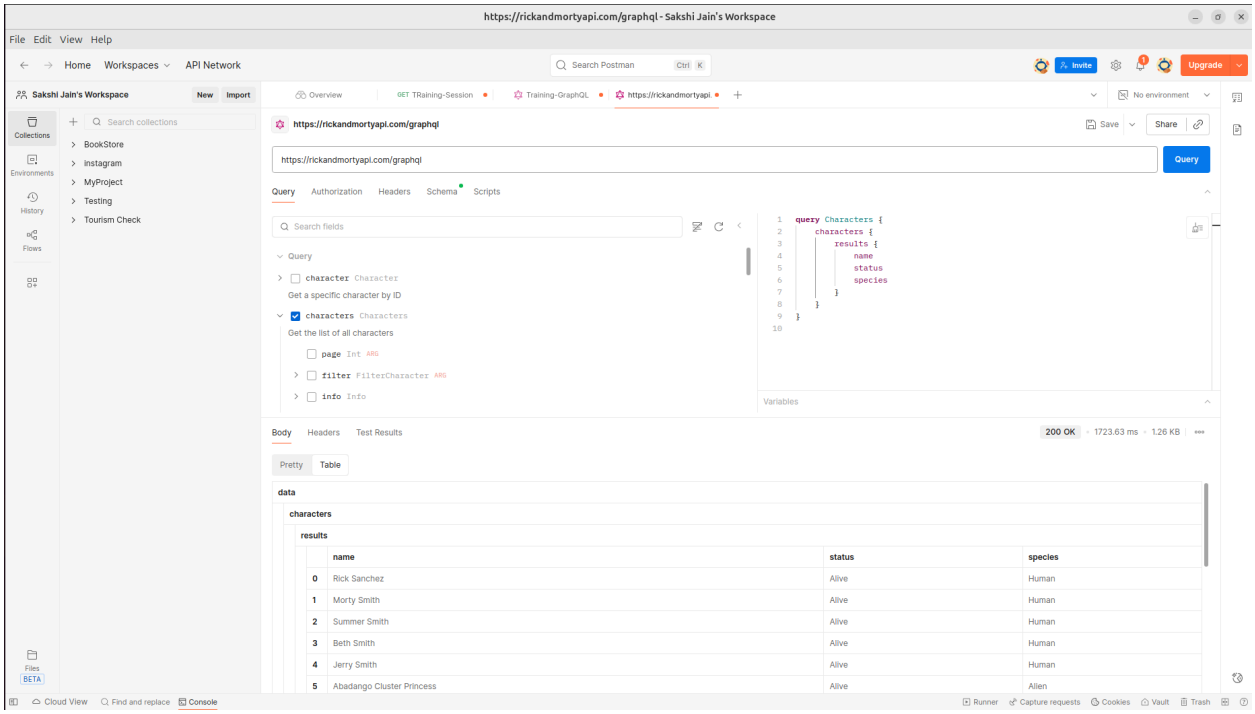
```
query {
  countries {
    code
    name
    capital
    emoji
  }
}
```



Again I checked - Rick and Morty API
URL: https://rickandmortyapi.com/graphql

SCHEMA

```
query {
  characters(page: 1) {
    results {
      name
      species
      status
    }
  }
}
```



Task 3: Make a POST request to https://jsonplaceholder.typicode.com/posts with a JSON body:

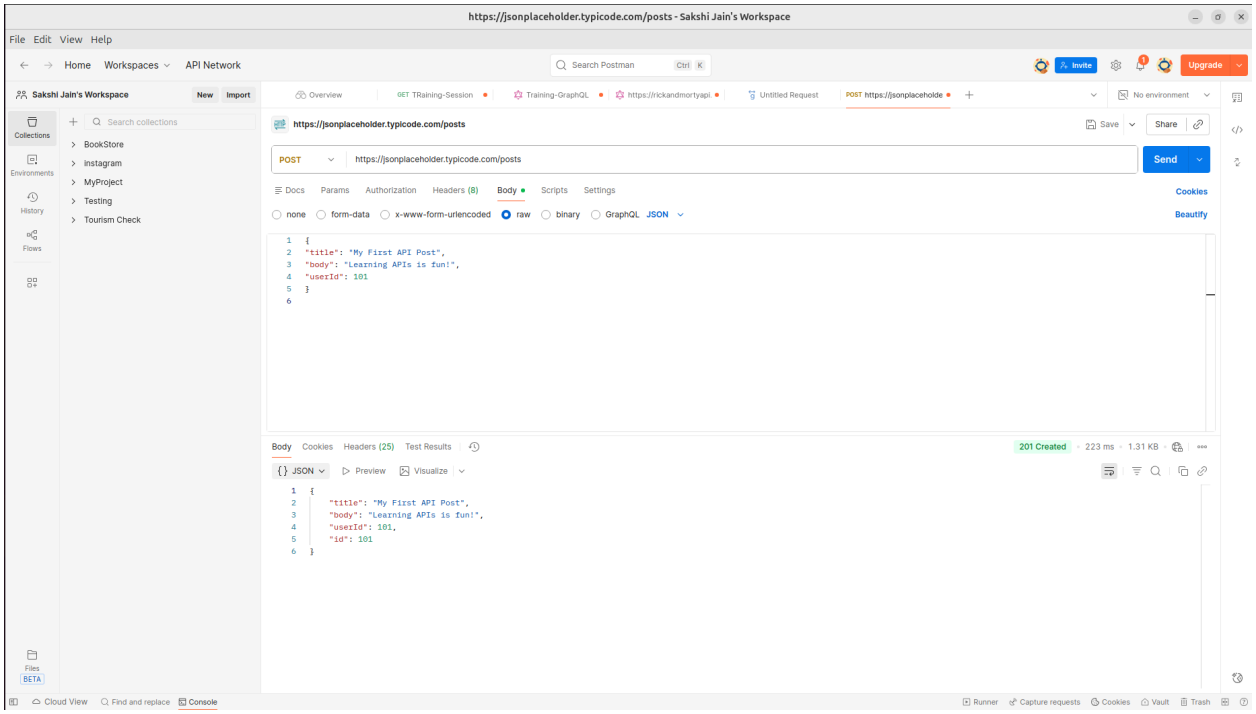
{

"title": "My First API Post",

"body": "Learning APIs is fun!",

"userId": 101

}



Task 4: In your browser DevTools Network tab, trigger a request to any website (e.g., your own site or google.com) and inspect:

- URL
- Method
- Request headers
- Response headers
- Body (if any)

Write a short explanation (5–8 lines) describing what happened from request → response.

When I first loaded https://bharat-yatra-frontend.vercel.app/, the browser already had a cached version of the page.
So it sent a GET request with cache validation headers (**if-none-match and if-modified-since**).

The server responded with 304 Not Modified, indicating the content had not changed and no response body was sent.

After disabling cache and reloading, the browser requested the page again without using cached data.
This time the server returned 200 OK along with the full HTML document. The browser then rendered the page and loaded additional assets like JavaScript and CSS.

What I observed in each part

URL : https://bharat-yatra-frontend.vercel.app/

Method : GET

Headers

- 304 request included cache headers like if-none-match and if-modified-since
- 200 response included content-type, content-length, etag, and caching headers

Body

- 304 → No response body (browser reused cached HTML)
- 200 → HTML content was returned in the response body

Basic difference between 304 and 200

- 304 Not Modified → Content unchanged, load from cache, faster, no body
- 200 OK → Fresh content sent by server, full response body included

While observing and analysing I got to know about: f-none-match and if-modified-since
And then i checked about it how this works

if-none-match

This works with something called an ETag (entity tag).

The server previously sent an ETag like:
"d00238250aee6422ae39b18d3e775984"

The browser stores it.

On the next request, the browser sends:
if-none-match: "d00238250aee6422ae39b18d3e775984"

If the current version does not match this tag, then send me the new content.

If the tag matches, the server replies 304 Not Modified.
If it doesn't, the server sends 200 OK with new content.

if-modified-since

This is a **time-based check**.

- The server earlier said:
  last-modified: Thu, 08 Jan 2026 07:11:47 GMT
- The browser later sends:
  if-modified-since: Wed, 07 Jan 2026 03:49:13 GMT

Only send the page again if it was changed after this time.

If nothing changed since that timestamp → **304**
If it changed → **200** with content

Lesser precise than if-none-match