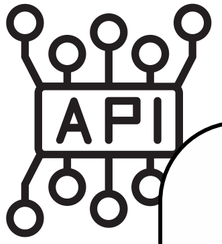


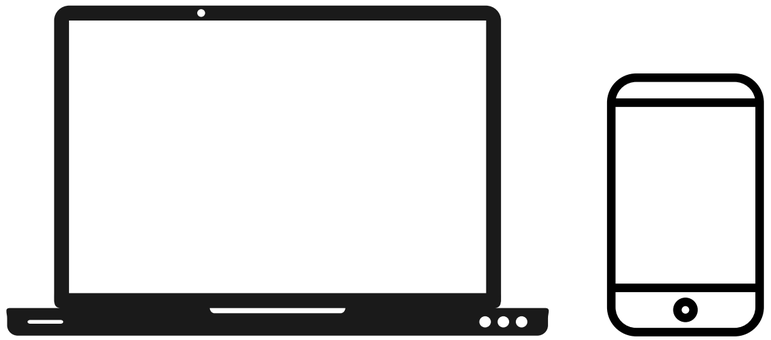
GITHUB CAMPUS CLUB

BACKEND WEB DEVELOPMENT WORKSHOP

CONTENTS

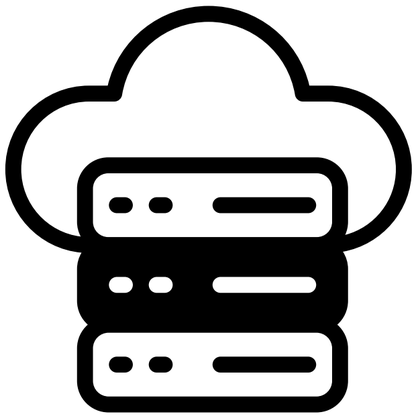
Introduction to Backend Web Development

Client



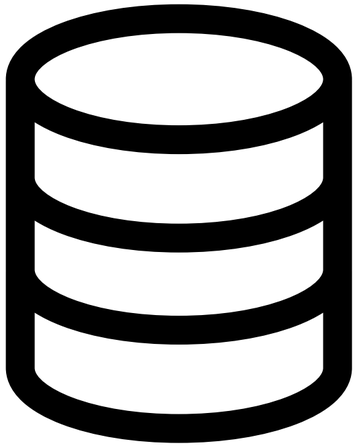
- The client is just the **frontend** part of a website.
- It provides a nice **Graphical User Interface (GUI)** for the users to interact with the website.
- It communicates with the **Backend server** through **API** to provide resources when user asks. (**Eg: searching for laptop in amazon.com**)
- The client is usually built with **HTML,CSS,J**S and with frameworks like **React, Vue, Angular etc..**

Server



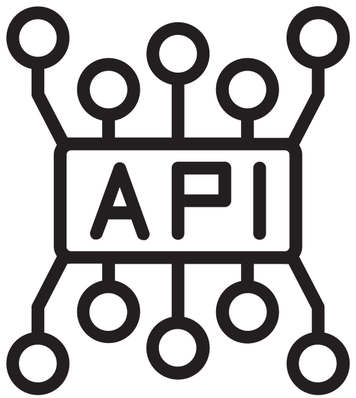
- A server is just a **computer.**
- The server **stores** and **manages** the website's data.
- Processes client requests, like **retrieving information**.
- Sends the **required data** back to the client for display.
- Handles tasks like **data storage**, **security**, and **processing**.
- Ensures the website functions correctly and responds to users.

Database



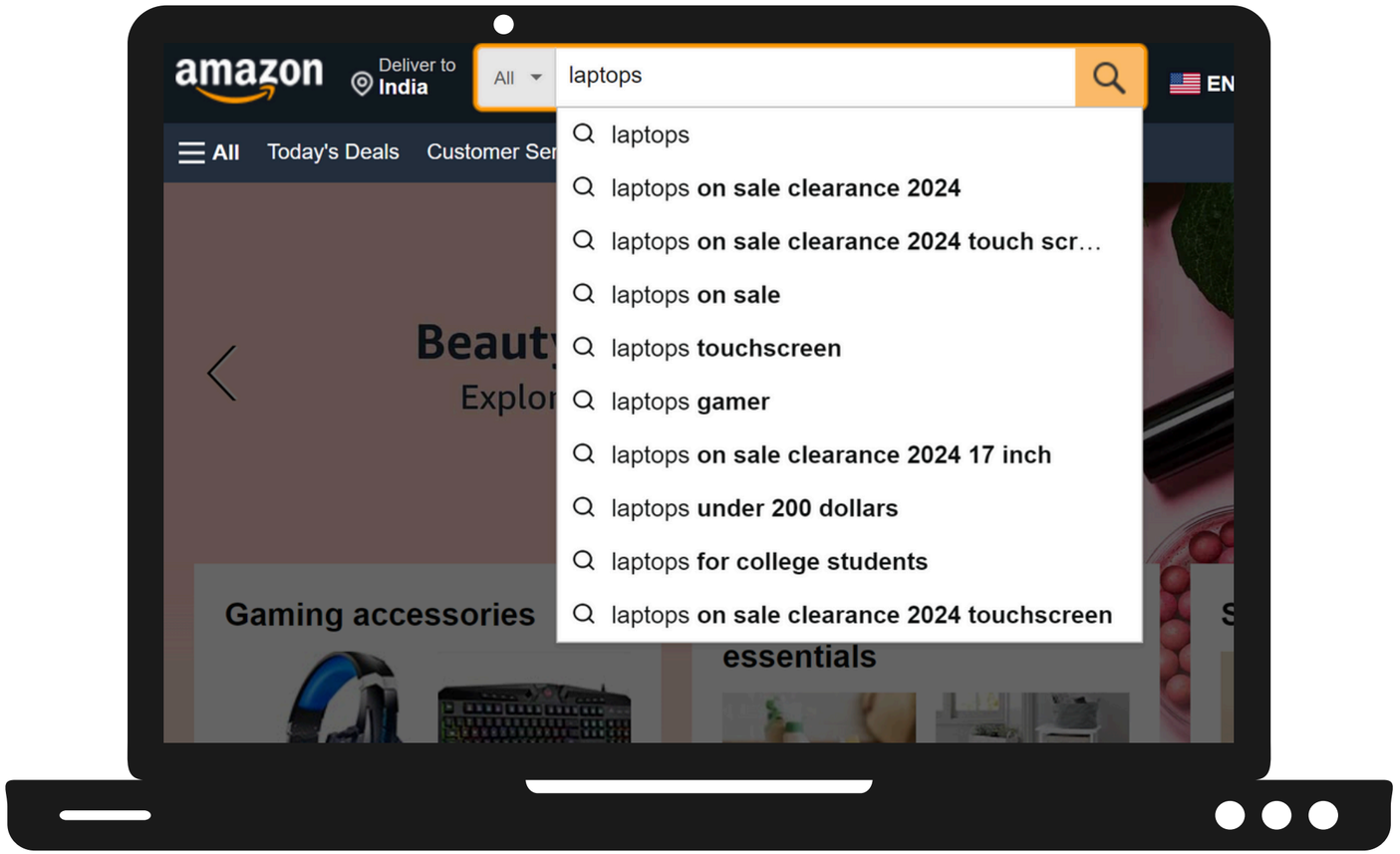
- A database **stores and organizes the website’s data.**
- Holds information like **user accounts or product details**.
- The **server connects** to the **database to retrieve or update data.**
- Helps **manage large amounts of data efficiently.**
- Essential for keeping the website functioning smoothly.

API (Application **P**rogramming **I**nterface**)**



- An **API** allows **different software systems** to **communicate**.
- Acts as a **bridge** between the **client** (frontend) and **server** (backend).
- The **client** sends a **request** to the **API** to **get or update data**.
- Makes it **easier to connect** parts of a website or integrate external services.
- Helps access functionality without needing to know internal details.

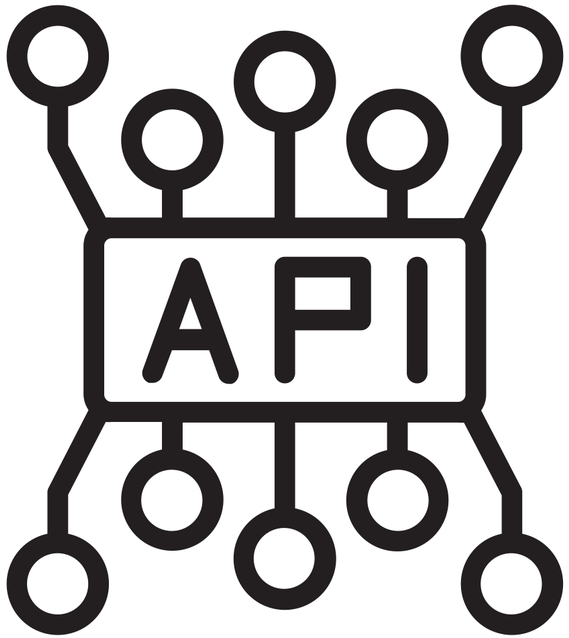
How it works...?



Client

Sends a request to API

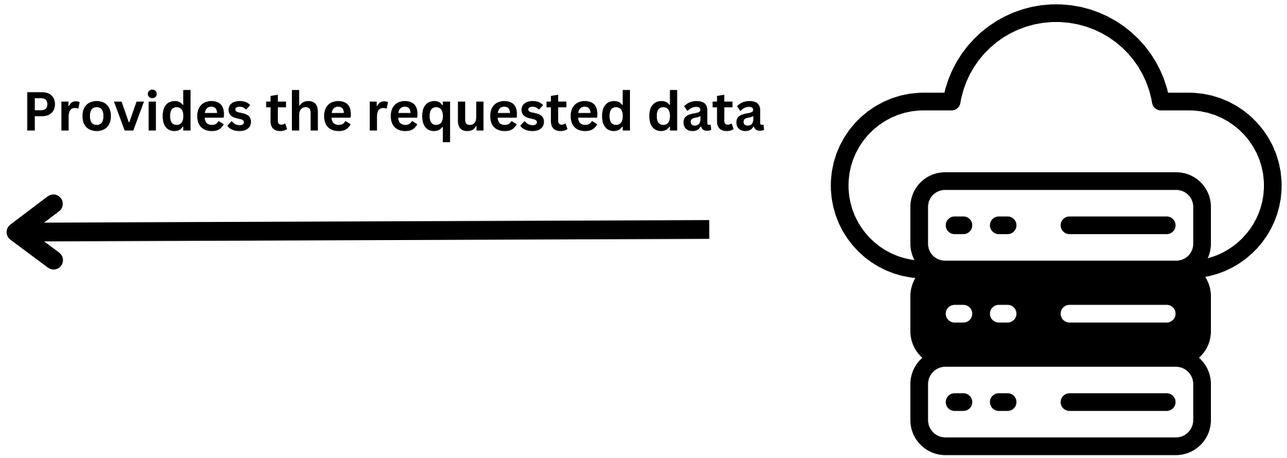
Receives the request



Communicates with server and database to fetch requested data

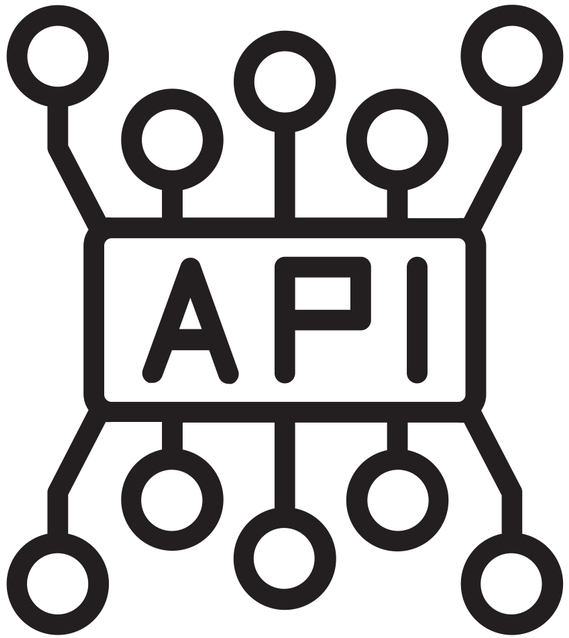
Amazon API

Provides the requested data



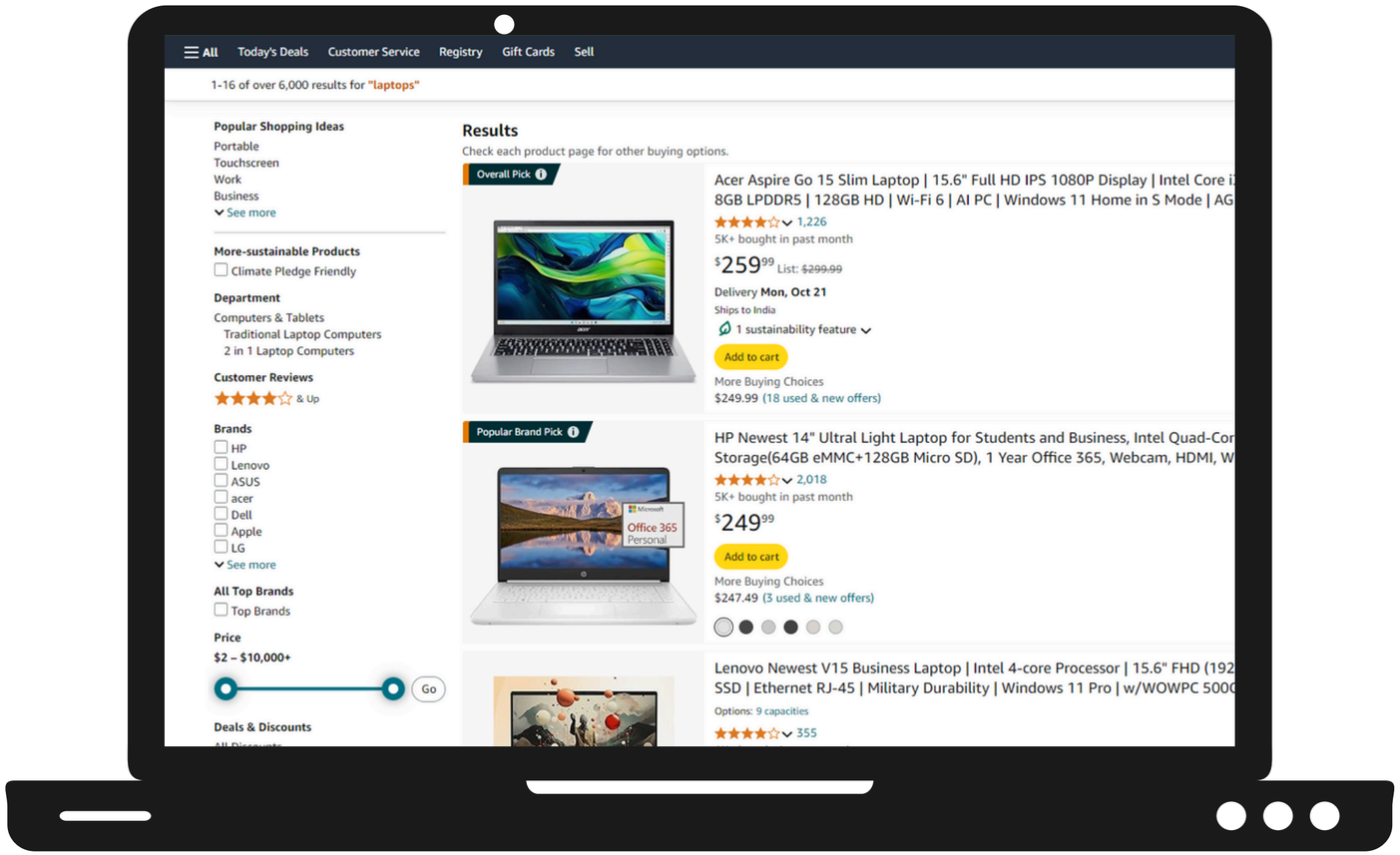
Amazon servers

Sends the data to client



Receives the data from the database

Amazon API



Receives the data form API and display it

Client

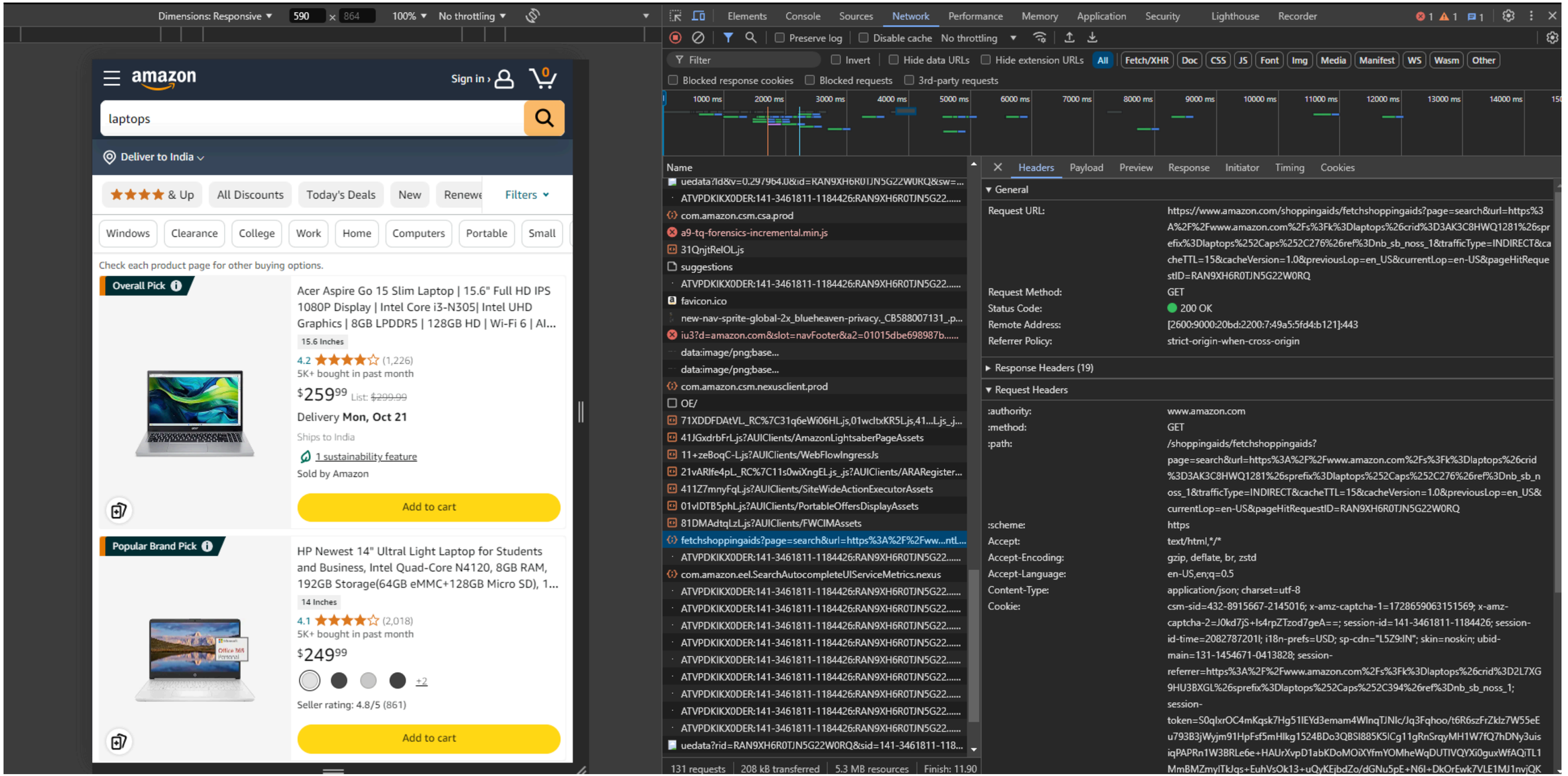
This is how it works...

Client (your PC or mobile)

Amazon API

Amazon servers

Visit the Network tab to view the requests and responses



Web Protocols

Web protocols define how data is transmitted over the internet.

- **HTTP**: Transmits web pages between client and server.
- **HTTPS**: Secured version of HTTP, encrypting data for security.
- **FTP**: Transfers files between client and server.
- **SMTP:** Sends emails from one server to another.
- **TCP/IP**: Manages data packet transmission across the internet.
- **WebSocket**: Enables real-time, two-way communication.

Environment Setup - Quickstart

Quickly jump into a preconfigured workspace using **repl.it**

Environment Setup - Quickstart

- Go to repl.it
- Login into your account or signup for a new account
- Click on *Create Repl* in the top left navbar
- Search for *Flask* in the templates
- Provide appropriate title
- Click on Run

https://shorturl.at/TDqjY

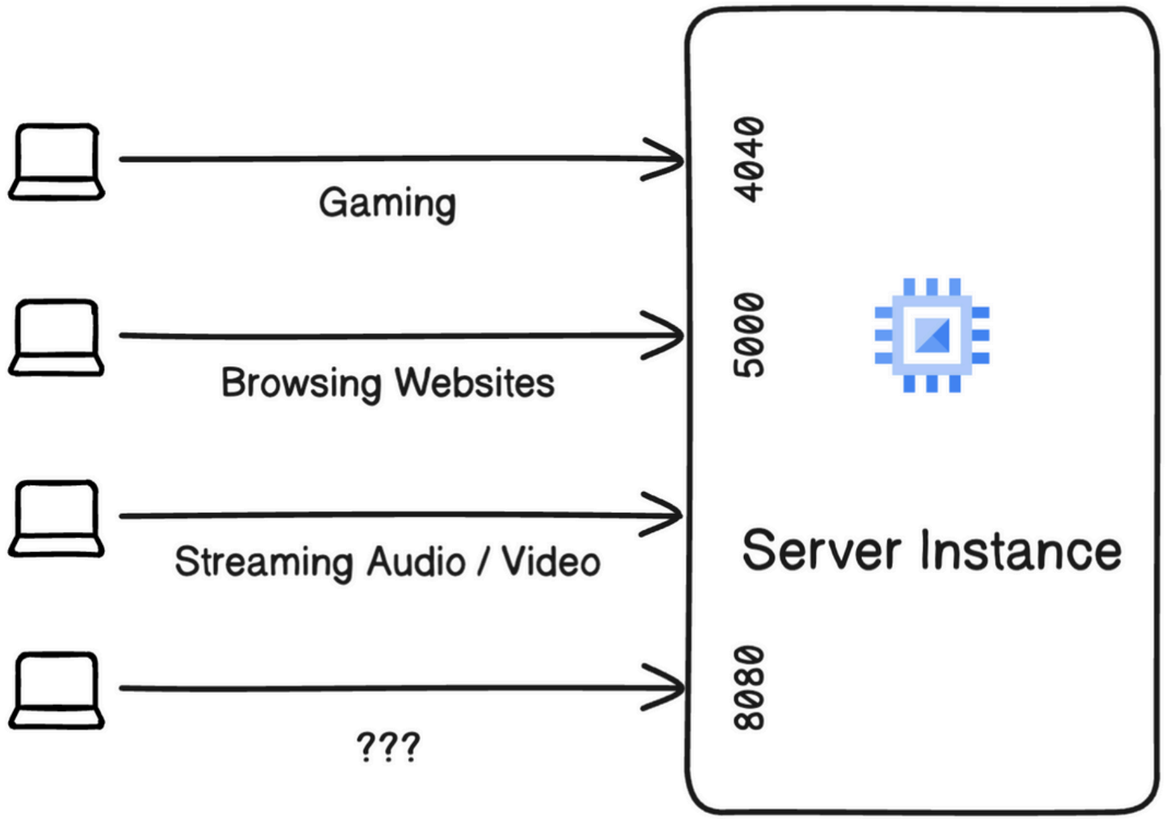
Guide / Hackpack for Flask

Setup API’s Locally

The actual way to use it

```python
from flask import Flask, request

app = Flask(__name__)


@app.route('/')
def index():
    return 'Hello from GHCC!' # feel free to change this string

# add new routes from here

# stop adding new routes here

if __name__ == '__main__':
  app.run(host='0.0.0.0', port=5000)
```

```python
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```


Gaming
Browsing Websites
Streaming Audio / Video
???
4040
5000
8080
Server Instance

```python
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
```

http://127.0.0.1:**5000**

or yes, http://localhost:**5000**

```python
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

http://127.0.0.1:**8080**

or yes, http://localhost:**8080**

Commonly used (and restricted) PORTs

Port	Protocol	Description
80	HTTP	Default port for web servers
443	HTTPS	Default port for secure web servers
21	FTP	Default port for file transfer protocol
22	SSH	Default port for secure shell connections
25	SMTP	Default port for sending emails
3306	MySQL	Default port for MySQL databases

```
 * Serving Flask app 'main'
 * Debug mode: off
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://127.0.0.1:8080
 * Running on http://192.168.29.68:8080
Press CTRL+C to quit
```

How APIs are actually used IRL

Demonstration with real application

http://127.0.0.1:**4160** ❌

http://localhost:**4160** ❌

http://35.132.22.140:**4160** ❌

https://algorithm.akashshanmugaraj.com ✅

```
server {
        server_name algorithm.akashshanmugaraj.com;
        location / {
                proxy_pass http://localhost:4160;
        }
```

That’s it 🎉

Now you are one step closer to a Backend Engineer!