TASK 13

What is an API?

An API (Application Programming Interface) allows two software applications-to, communicate with each other.
It acts as a middleman that takes a request, processes it, and returns a response.

2 What is a Web API?

A Web API is an API that works over the internet using HTTP.
It enables communication between applications on different systems.
Example:
A mobile app fetching data from a server.

3 What are Public APIs?

Public APIs are available for anyone to use with minimal restrictions.
They often require an API key and may have usage limits.
Example:
Google Maps API, Weather API.

4 What is the OpenAPI Standard?

The OpenAPI standard is a specification for documenting and designing APIs.
It helps developers understand endpoints, request/response formats, and errors easily.

5 What are Private / Internal APIs?

Private APIs are used only within an organization.
They provide access to internal systems and data and require strict authentication.
Example:
Company internal HR or inventory systems.

6 What are Partner APIs?

Partner APIs are shared between trusted business partners.
Access is limited to approved organizations for specific use cases.
Example:
Bank APIs used by payment apps like Google Pay.

7 What is a Monolithic API?

A Monolithic API is built as a single large application handling all functions.
All components are tightly connected.

Advantage: Simple to develop
Disadvantage: Hard to scale and modify

8 What is a Microservices API?
A Microservices API architecture splits functionality into small, independent services.
Each service handles one specific task.
Example:
Separate services for login, payment, and orders.

9 What is a Composite API?
A Composite API combines multiple API calls into a single request.
It reduces network calls and improves efficiency.
Example:
One request to create user, send email, and store profile data.

10 What is a Unified API?
A Unified API provides a single interface to access multiple different APIs.
It hides complexity from the client.
Example:
One finance app accessing multiple bank APIs.

11 What is a REST API?
A REST API uses HTTP methods to access and manipulate resources.
It is stateless and commonly uses JSON.
Methods: GET, POST, PUT, DELETE

12 REST Architectural Principles
- Client–server separation
- Stateless requests
- Cacheable responses
- Layered system

13 What is a SOAP API?
A SOAP API is a protocol that uses XML for communication.
It is strict, secure, and supports complex transactions.
Used in: Banking and enterprise systems.

14 What is an RPC API?
An RPC API allows a client to call a function on a remote server.
It focuses on actions, not resources.

Example:
startProcess() or generateReport()

1️⃣5️⃣ What is gRPC?

gRPC is a high-performance RPC framework developed by Google.
It uses Protocol Buffers and HTTP/2 for fast communication.
Used in: Microservices and internal systems.

1️⃣6️⃣ What is GraphQL?

GraphQL is a query language for APIs that allows clients to request exactly the data they need.
It usually has a single endpoint.
Advantage: No over-fetching
Disadvantage: More complex to design

1️⃣7️⃣ REST vs GraphQL (Brief)

- REST: Fixed endpoints, simple, widely used
- GraphQL: Flexible queries, efficient data fetching

1️⃣8️⃣ Which API should beginners focus on?

Beginners should focus on:

- REST APIs
- Public & Private APIs
- Microservices basics

SOAP, gRPC → optional / advanced

COMING TO CONCEPT OF API CREDENTIALS
HEADERS: use to say the behaviour and security of the response.
Compare and discuss about the two url which constructs on api.

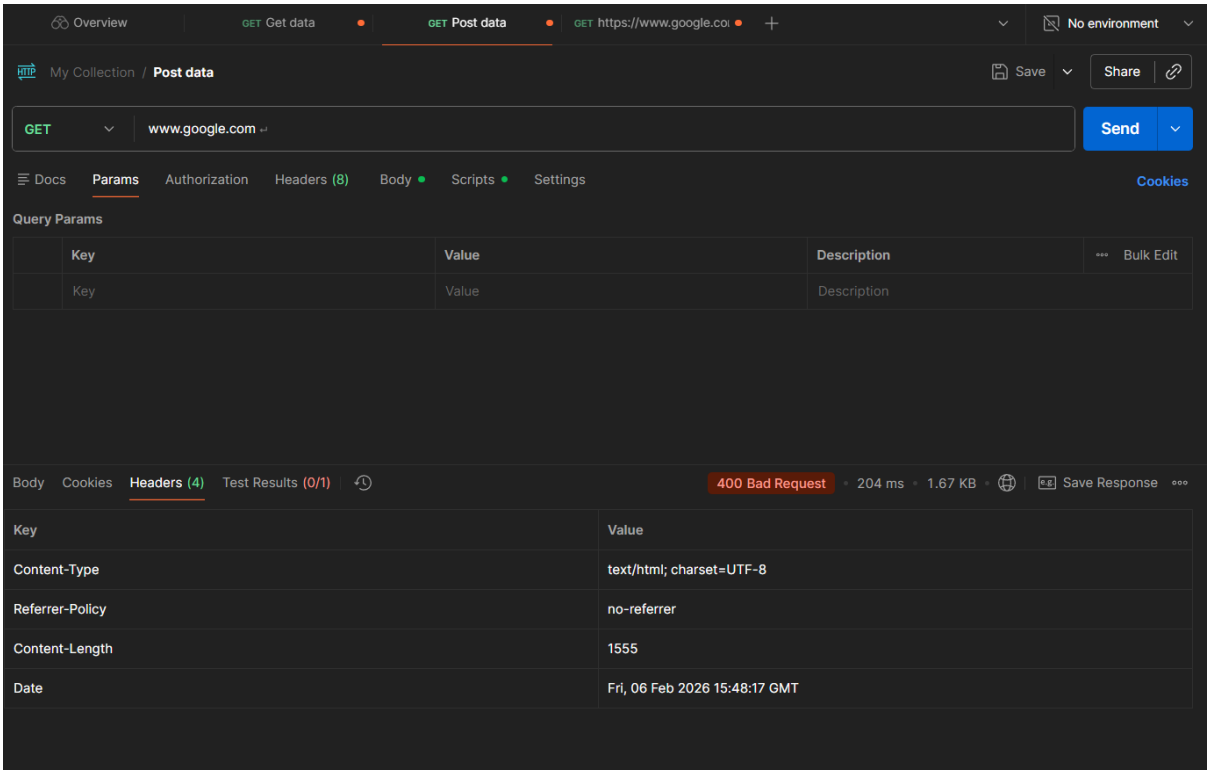
1. **Normal http:**

1) Content type

2) Browser should render HTML
3) Character encoding defined

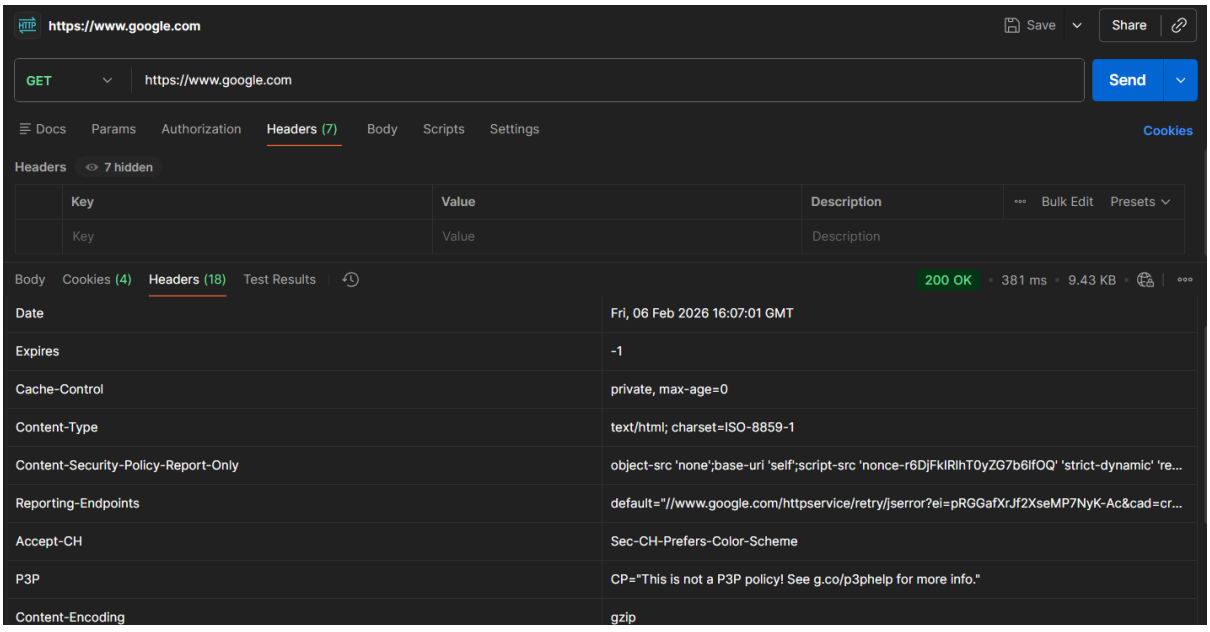
2) Content length

- Useful for performance & integrity checks



2)Second URL – HTTPS:

- Content and security: blocks the unsage objects.
- There is no embedded in other sites
- Authentication: using cookies
- GWS server



HTTP response headers were analyzed to understand security controls enforced by the server. Secure responses included headers such as Content-Security-Policy, X-Frame-Options, and Secure cookies, which protect against XSS, clickjacking, and session attacks. Comparison with basic responses highlighted the importance of HTTPS and security headers in protecting web and API communication.