Assignment: 2 Introduction to HTTP Requests

1. Define HTTP and explain its importance in web communication.

HTTP (HyperText Transfer Protocol) is a communication protocol used for transferring data between web browsers and servers over the internet. It defines how requests and responses are structured, enabling users to access web pages, send data, and interact with online applications.

- Importance of HTTP:
- Foundation of the Web: Enables web browsing and data exchange.
- Stateless Communication: Each request is independent, improving scalability.
- Supports Various Data Formats: Handles HTML, JSON, XML, images, and more.
- Used in APIs: Web services and RESTful APIs rely on HTTP for communication.

2. Describe the different types of HTTP requests (GET, POST, PUT,

DELETE, etc.).

1. GET: This method is used to retrieve data from a server. It does not modify the data and is commonly used for fetching webpages, API responses, or other resources. For example, when you enter a URL in a browser, it sends a GET request to fetch the page.
2. POST: This method is used to send data to a server to create a new resource. It is often used when submitting forms, creating new user accounts, or sending data to an API. A POST request typically includes a request body containing the data to be added.
3. PUT: The PUT method is used to update an existing resource on the server. It replaces the entire resource with the new data sent in the request body. For instance, updating a user's profile information might involve sending a PUT request to modify the user's details.
4. DELETE: This request method is used to remove a specified resource from the server. If you delete a post on a social media site, for example, a DELETE request is sent to the server to remove it.
5. PATCH: The PATCH method is similar to PUT but is used for partial updates. Instead of replacing the entire resource, it modifies only specific fields. If a user wants to update only their email address without changing their entire profile, a PATCH request would be used.

3. Explain the components of an HTTP request (method, URL, headers, body).

An HTTP request consists of:

1. Method – Specifies the action (GET, POST, etc.).
2. URL (Uniform Resource Locator) – Identifies the resource (e.g., https://example.com/users).
3. Headers – Carry metadata (e.g., authentication tokens, content type).
4. Body – Contains data (only for methods like POST, PUT, PATCH).

4. Discuss HTTP status codes and their meanings.

HTTP status codes are three-digit responses sent by a server to indicate the outcome of a request. They are grouped into different categories based on their purpose:

- 1xx (Informational): These codes indicate that the request has been received and is being processed. For example, 100 Continue means the server has received the request headers and the client can proceed with sending the body.
- 2xx (Success): These codes confirm that the request was successfully processed. The most common code, 200 OK, means the request was successful, while 201 Created is used when a resource is successfully created (e.g., a new user is added to a database).
- 3xx (Redirection): These codes indicate that further action is required to complete the request. 301 Moved Permanently means the requested resource has been permanently moved to a new URL, and 302 Found is used for temporary redirection.
- 4xx (Client Errors): These codes indicate that the request was incorrect or cannot be processed due to an issue from the client's side. 400 Bad Request means the server could not understand the request due to invalid syntax. 401 Unauthorized indicates authentication is required, and 404 Not Found means the requested resource does not exist.
- 5xx (Server Errors): These codes indicate that the server failed to fulfill a valid request. 500 Internal Server Error is a general error when something unexpected happens on the server. 503 Service Unavailable means the server is temporarily overloaded or undergoing maintenance**.**

5. Explain the difference between HTTP and HTTPS.

HTTP (HyperText Transfer Protocol) and HTTPS (HyperText Transfer Protocol Secure) are both used for communication over the web, but HTTPS provides an added layer of security.

HTTP transmits data in plain text, making it vulnerable to interception by attackers. In contrast, HTTPS encrypts data using SSL/TLS (Secure Sockets Layer/Transport Layer Security), ensuring secure communication between the client and server.

Websites using HTTP are more susceptible to cyber threats such as man-in-the-middle attacks, where an attacker can intercept and alter data during transmission. HTTPS prevents such threats by encrypting the data before it is sent, making it unreadable to unauthorized parties.

6. Describe how authentication and security are handled in HTTP requests.

Authentication Methods:

- Basic Authentication: Uses username:password encoded in Base64 (not secure).
- Token-Based Authentication: Uses API keys or JWT (JSON Web Tokens).
- OAuth 2.0: Secure method for third-party authentication (e.g., logging in via Google).

Security Measures:

- HTTPS (SSL/TLS Encryption): Encrypts HTTP traffic.
- CORS (Cross-Origin Resource Sharing): Restricts which domains can access an API.
- Rate Limiting & Firewalls: Prevents excessive requests and attacks.