

## ❖ What happens when you type facebook.com on your browser and enter ? Write a full report on it.

- When you type "facebook.com" into your browser's address bar and press Enter, a series of complex steps occurs to load the webpage. This process involves multiple network protocols and layers of the OSI model, from DNS resolution to rendering the page. Below is a detailed breakdown of each step:

### Step-by-Step Process

#### 1. DNS Resolution

##### DNS Query Initiation:

- The browser first checks its DNS cache to see if it already knows the IP address for `facebook.com`.
- If not found in the cache, the browser checks the operating system's DNS cache.
- If still unresolved, the browser initiates a DNS query to resolve the domain name to an IP address.

##### DNS Resolver and Recursive Query:

- The query is sent to a DNS resolver, typically provided by your ISP.
- The resolver checks its cache; if the address is not cached, it performs a recursive query starting with the root DNS servers.

##### Root DNS Servers:

- The resolver contacts a root DNS server, which responds with the address of a Top-Level Domain (TLD) DNS server responsible for `.com` domains.

##### TLD DNS Servers:

- The resolver contacts the TLD DNS server, which responds with the address of the authoritative DNS server for `facebook.com`.

##### Authoritative DNS Server:

- The resolver contacts the authoritative DNS server, which responds with the IP address(es) for `facebook.com`.

##### Response to Browser:

- The DNS resolver returns the IP address to the browser, which caches it for future use.

#### 2. Establishing a TCP Connection

##### Socket Creation:

- The browser creates a socket to initiate a connection to the server.

### **Three-Way Handshake:**

- The browser sends a TCP SYN packet to the IP address on port 443 (HTTPS).
- The server responds with a SYN-ACK packet.
- The browser sends an ACK packet, establishing the connection.

## **3. SSL/TLS Handshake (HTTPS)**

### **Initiation:**

- The browser and server initiate an SSL/TLS handshake to establish a secure connection.

### **Handshake Steps:**

- The browser sends a ClientHello message with supported encryption methods.
- The server responds with a ServerHello message, selecting the encryption method and sending its SSL certificate.
- The browser verifies the server's certificate using the Certificate Authority (CA) public key.
- The browser and server exchange keys to establish a secure, encrypted connection.

## **4. HTTP Request**

### **Request Preparation:**

- The browser prepares an HTTP GET request for the URL  
`https://facebook.com`.

### **Sending the Request:**

- The request includes headers with information about the browser, accepted data formats, and other preferences.

### **Request Transmission:**

- The HTTP request is sent over the established TCP (and SSL/TLS) connection to the server.

## **5. Server Processing**

### **Request Handling:**

- The server receives the HTTP request and processes it. This might involve querying a database, accessing backend services, or generating dynamic content.

### **Response Preparation:**

- The server prepares an HTTP response, which typically includes a status line (e.g., 200 OK), response headers, and the requested content (e.g., HTML, CSS, JavaScript).

### **Sending the Response:**

- The server sends the HTTP response back to the browser over the established connection.

## **6. Rendering the Page**

### **Receiving the Response:**

- The browser receives the HTTP response and begins processing the content.

### **Parsing HTML:**

- The browser parses the HTML document and builds the Document Object Model (DOM) tree.

### **Loading Resources:**

- As the HTML is parsed, the browser identifies and requests additional resources (CSS, JavaScript, images) required to render the page.

### **Rendering:**

- The browser applies CSS to the DOM to create the Render Tree.
- JavaScript is executed to manipulate the DOM and handle events.
- The browser renders the content to the screen, displaying the complete web page to the user.

## **In-Depth Analysis of Each Step**

### **DNS Resolution**

- **Root Servers:** These are a critical component of the DNS hierarchy, providing information about where to find TLD servers.
- **Caching:** DNS caching at various levels (browser, OS, resolver) helps speed up the resolution process by reducing the need to repeatedly perform the same lookups.

### **Establishing a TCP Connection**

- **TCP/IP Model:** TCP operates at the Transport layer, ensuring reliable data transmission. IP operates at the Network layer, handling addressing and routing.

- **Three-Way Handshake:** This process ensures a reliable connection is established before data transmission begins.

### SSL/TLS Handshake (HTTPS)

- **Encryption:** SSL/TLS uses asymmetric encryption for the initial handshake and symmetric encryption for the session.
- **Certificate Authorities:** Trusted entities that issue SSL certificates, ensuring the legitimacy of the server.

### HTTP Request/Response

- **Headers:** HTTP headers carry important metadata about the request and response, such as content type, content length, and caching policies.
- **Status Codes:** The status line in the HTTP response provides information about the result of the request (e.g., 200 OK, 404 Not Found).

### Rendering the Page

- **DOM and Render Tree:** The DOM represents the structure of the HTML document, while the Render Tree represents how it should be displayed.
- **JavaScript Execution:** JavaScript can dynamically modify the DOM and CSSOM (CSS Object Model), affecting the final rendered page.
- **Painting and Compositing:** The final step in rendering, where the browser draws the visual representation of the web page on the screen.

### Conclusion

Typing "facebook.com" into a browser and pressing Enter triggers a sophisticated series of events that involve DNS resolution, TCP/IP communication, SSL/TLS security, HTTP protocol transactions, and complex rendering processes. Each step relies on multiple technologies and protocols working seamlessly together to deliver the web page to the user. This intricate process happens within milliseconds, providing a smooth and efficient user experience.