Question-1: Explain the purpose, and importance of the OSI Model, and briefly describe each layer of the OSI model with an example.

The **OSI (Open Systems Interconnection) Model** is a conceptual framework used to understand and standardize how different networking protocols interact to enable communication between computer systems. It breaks down complex networking functions into **seven distinct layers**, each responsible for specific tasks.

Importance:

- Provides a universal language and reference model for networking.
- Helps vendors develop interoperable hardware and software.
- Simplifies troubleshooting by isolating issues to specific layers.
- Enables modular engineering and flexible network design.

The 7 Layers of the OSI Model (Bottom to Top):

Layer Number	Layer Name	Purpose	Example
7	Application	Interface for end-user services; provides network services to applications.	Web browsers (HTTP), Email clients (SMTP)
6	Presentation	Translates data formats, encryption, compression.	SSL/TLS encryption, JPEG/ASCII conversion
5	Session	Manages sessions, establishes, maintains, and terminates connections.	Managing login sessions in remote desktop
4	Transport	Provides end-to-end communication, error correction, flow control.	TCP (reliable connection), UDP (fast, unreliable)
3	Network	Handles logical addressing and routing of data packets.	IP addressing, routers
2	Data Link	Provides node-to-node data transfer, error detection/correction on physical link.	MAC addressing, switches, Ethernet protocol
1	Physical	Transmits raw bitstream over physical medium.	Cables, hubs, electrical signals

Q2. Explain the purpose, and importance of the TCP/IP, and briefly describe the different layers of the TCP/IP model.

Purpose and Importance of TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) is the fundamental communication protocol suite that powers the Internet and most modern networks. Its purpose is to enable reliable, end-to-end data communication across diverse interconnected networks.

Importance:

- It's the foundation of the Internet and local networks.
- Provides standard protocols for data transmission, addressing, routing, and error handling.
- Ensures interoperability between different devices and networks worldwide.

TCP/IP Layers (4 Layers, Brief Description)

Layer	**Purpose**	**Example Protocols**
Application	Interfaces with user applications and provides protocols for email, file transfer, web, etc.	HTTP, FTP, SMTP, DNS
Transport	Provides reliable (TCP) or unreliable (UDP) data delivery between hosts.	TCP, UDP
Internet	Handles logical addressing and routing of packets across networks.	IP (IPv4, IPv6), ICMP
Network Access (Link)	Manages physical transmission of data over network hardware.	Ethernet, Wi-Fi, ARP

Q3. Create a comparison table between OSI and TCP/IP model.

Aspect	**OSI Model**	**TCP/IP Model**
Number of Layers	7	4
Layers	Application, Presentation, Session, Transport, Network, Data Link, Physical	Application, Transport, Internet, Network Access (Link)
Development	Developed as a theoretical standard by ISO	Developed based on practical protocols for ARPANET/Internet

Aspect	**OSI Model**	**TCP/IP Model**
Approach	Strict layered approach with clear separation	More flexible, layers sometimes overlap in function
Application Layer	Separate Application, Presentation, Session layers	Single Application layer combines these functions
Transport Layer	Supports TCP and UDP	Supports TCP and UDP
Network Layer	Handles routing with IP	Internet layer primarily uses IP for routing
Physical & Data Link	Separate Physical and Data Link layers	Combined into Network Access (Link) layer
Usage	Used as a reference model for understanding and designing networks	Protocol suite actually used on the Internet
Protocol Independence	Protocol independent	Protocol specific (focused on TCP/IP protocols)
Standardization	ISO standard	Developed by DARPA, IETF standards

Q4. What is DNS, and how does it resolve domain names?

DNS (Domain Name System) is like the Internet’s phonebook—it translates human-friendly domain names (like example.com) into IP addresses (like 192.0.2.1) that computers use to communicate.

How it resolves domain names:

1. You type a domain (e.g., example.com) into your browser.
2. Your computer asks a DNS resolver (usually your ISP’s server) for the IP address.
3. The resolver checks its cache; if not found, it queries root servers.
4. Root servers direct it to the TLD (Top-Level Domain) servers (like .com servers).
5. TLD servers point to the authoritative name servers for the domain.
6. Authoritative servers respond with the domain’s IP address.
7. The resolver sends the IP back to your computer, which connects to the website.

HTTP Request-Response Model

Q5. Provide a brief description of the HTTP request and response model along with an illustrative diagram.

Purpose:
HTTP (HyperText Transfer Protocol) is the foundation of data communication on the web. It works as a **request-response protocol** between a client (usually a browser) and a server.

How it works:

1. **HTTP Request:**
   The client sends a request to the server. This request includes:
   - A request line (method like GET, POST, etc., and the URL)
   - Headers (metadata like content type, user agent)
   - Optional body (for methods like POST)
2. **HTTP Response:**
   The server processes the request and sends back a response, which includes:
   - A status line (status code like 200 OK, 404 Not Found)
   - Headers (content type, caching info, etc.)
   - Body (the requested resource, e.g., HTML page, JSON data)

```
+-----------------+                    +------------------+
|    Client       |                    |      Server      |
|  (Web Browser)|                      | (Web Server)     |
+-----------------+                    +------------------+
       |                                      |
       | 1. HTTP Request (e.g., GET /index.html)            |
       |--------------------------------------------------->|
       |                                      |
       |                    2. Process Request              |
       |                                      |
       | 3. HTTP Response (e.g., 200 OK + HTML content) |
       |<---------------------------------------------------|
       |                                                    |
```

Q6. What is CORS? Explain its purpose and why developers frequently encounter CORS errors during development.

CORS (Cross-Origin Resource Sharing) is a browser security feature that restricts web pages from making requests to a different domain than the one that served the web page, protecting users from malicious sites.

Its purpose is to allow servers to specify who can access their resources from different origins.

Developers often face CORS errors during development because browsers block cross-origin requests if the server does not explicitly permit them via CORS headers. This usually happens when frontend and backend run on different localhost ports or domains without proper CORS configuration