Name: MD Imran Nazir Udoy

Id: 22303286

1. Domain Names

Definition: A domain name is like a contact name for a website, replacing hard-to-remember numerical IP addresses with readable words. Instead of typing 172.67.133.202 into your browser, you simply type example.com.

Example: Imagine you need to visit your friend's house. Instead of saying, "Go to house number 12345678," you'd say, "Go to 123 Uttara." Similarly, example.com is more memorable than an IP address.

- **Top-Level Domains (TLDs)**: These are the extensions at the end of domain names, such as .com, .org, or country-specific ones like .gov.bd for Bangladesh government sites. They categorize the domain and can give clues about its purpose (e.g., .org for organizations).

2. Hosting Server

Definition: A hosting server is a computer where all the files, databases, and content for a website are stored. It's what makes the website accessible when you type the domain name in your browser.

Example: Think of a hosting server as your favorite bakery. The bakery stores all the ingredients and baked goods (website files), and when customers come in, it serves them the products (web pages and content).

3. Name Servers

Definition: Name servers are part of the Domain Name System (DNS), which translates human-readable domain names into machine-readable IP addresses.

Procedure:

- **Step 1**: You type example.com into your browser. This starts a DNS lookup to find the matching IP address.
- **Step 2**: The name servers return the IP address of the hosting server.
- **Step 3**: The browser uses this IP address to connect to the hosting server and request the website's files.
- **Step 4**: The hosting server responds by sending back the files, and the browser displays the website on your screen.

4. Communication between Frontend, Server-Side, and Database

When you interact with a website (e.g., submitting a form), the **frontend** (what you see and click) sends a request to the **server**. The server processes this request and, if needed, fetches data from the **database**.

- **Process**:
  - The **frontend** collects user input and sends it to the server.
  - The **server** processes any necessary logic and interacts with the **database** if data is needed.
  - The **database** retrieves or updates data and sends it back to the server.
  - The **server** formats the response and sends it to the frontend.
  - The **frontend** displays this data to the user.

5. Relational Databases (SQL)

- **MySQL**: Ideal for general-use cases, known for being relatively fast and simple to use.
- **PostgreSQL**: Popular for applications requiring complex data structures and high data integrity.
- **Oracle Database**: Commonly used in industries that require robust data consistency and security, though it's often costlier and used for larger applications.

6. MVC (Model-View-Controller)

Definition: The MVC design pattern separates an application's logic, user interface (UI), and user interaction into three parts:

- **Model**: Represents data and business logic (e.g., database operations).
- **View**: The user interface (e.g., web pages or forms).
- **Controller**: Acts as a bridge between the model and the view, handling user input and updating the model and view accordingly.