**1. What is a Web Application?**

A **web application** is a software program that runs on a **web server** and is accessed via a **web browser** over a network (commonly the Internet or Intranet). Unlike desktop applications, web apps do not need to be installed on the client device.

**Examples:**

* Gmail, Facebook, LinkedIn
* Online banking
* E-commerce sites (Amazon, Flipkart)

**2. Web Application Building Blocks**

A web application is typically built using the following key components:

**a) Frontend (Client-Side):**

* Technologies: HTML, CSS, JavaScript
* Frameworks: React, Angular, Vue.js
* Runs in the user's browser
* Responsible for UI and user interaction

**b) Backend (Server-Side):**

* Technologies: Java (Spring), Node.js, Python (Django/Flask), PHP
* Handles business logic, database interaction, authentication, etc.

**c) Database:**

* Stores data persistently
* Examples: MySQL, PostgreSQL, MongoDB

**d) Web Server/Container:**

* Software that handles HTTP requests (get,update(put),insert(post),delete)
* Examples: Apache, Nginx

**e) API Layer (Optional):**

* RESTful or GraphQL APIs allow communication between frontend and backend

**3. How to Access a Web Application?**

To access a web application:

1. Open a browser (e.g., Chrome, Firefox).
2. Enter the URL (e.g., https://www.example.com).
3. The browser sends an HTTP(S) request to the web server.
4. The server processes the request and sends back an HTML/CSS/JS response.
5. The browser renders the content for user interaction.

**4. How HTTP(FTP+SMTP) Works?**

**🔹 HTTP (HyperText Transfer Protocol) is the foundation of data communication on the Web.**

**Key Steps:**

1. **Client sends a request** (GET, POST, PUT, DELETE) to a server.
2. **Server processes the request** and returns a response (HTML, JSON, etc.).
3. HTTP is **stateless**, so each request is independent.

**Example:**

GET /home HTTP/1.1

Host: www.example.com

**Response:**

HTTP/1.1 404 Not Found

Content-Type: text/html

<body>Hello User</body>

**5. Web Application Multiple Tiers**

**🧱 Typical tiers in a web application architecture:**

| **Tier** | **Description** |
| --- | --- |
| **Presentation Tier** | Frontend UI layer (HTML, CSS, JS in browser) |
| **Application Tier** | Backend business logic (Java, Node.js, etc.) |
| **Data Tier** | Database system (e.g., PostgreSQL, MongoDB) |

**Example: 3-Tier Architecture**

1. **Client (Browser)** — sends HTTP requests
2. **Server (App Logic)** — processes request, calls DB
3. **Database** — stores/retrieves data

**6. How are Web Applications Deployed?**

Web application deployment is the process of **making your application live on the internet**.

**🔹 Steps in deployment:**

1. **Build the application**
   * Bundle frontend & backend code
2. **Choose a deployment platform:**
   * Cloud (AWS, Azure, GCP)
   * PaaS (Heroku, Vercel, Netlify)
   * On-premise server
3. **Set up the environment:**
   * OS, dependencies, databases
4. **Deploy the code:**
   * Use FTP, CI/CD pipelines (Jenkins, GitHub Actions)
5. **Run the server:**
   * Host on Apache/Nginx or containers like Docker
6. **Monitor & scale:**
   * Use tools like Prometheus, New Relic, Kubernetes

**Summary Table**

| **Topic** | **Key Points** |
| --- | --- |
| **What is a Web App?** | Software accessed via browser, runs on server |
| **Building Blocks** | Frontend, Backend, DB, Server, APIs |
| **How to Access** | Browser sends HTTP request, server responds |
| **How HTTP Works** | Stateless request-response protocol |
| **Multiple Tiers** | Presentation, Application, Data Access object/Repository |
| **Deployment** | Cloud hosting, CI/CD, Web server setup, Docker |