1.FULL STACK DEVELOPMENT

Full stack development involves working on both the front-end and back-end of a web application or website. A full stack developer has the skills to handle all aspects of development, from designing user interfaces to managing servers and databases.

2.WEB DEVELOPMENT

Web development involves the creation and maintenance of websites and web applications.

3.WEB PAGE

* A web page is a document or resource of information that is suitable for the World Wide Web and can be accessed through a web browser.
* A web page typically contains text, images, videos, and hyperlinks to other web pages.
* It is identified by a unique URL (Uniform Resource Locator).

### **Components of a Web Page**

1. HTML (Hyper Text Markup Language)
2. CSS (Cascading Style Sheets)
3. JavaScript
4. Images and Multimedia
5. Hyperlinks

Types of Web Pages

* Static Web Pages

**Characteristics:** Fixed content; each user sees the same content.

* Dynamic Web Pages

**Characteristics:** Content can change based on user interaction or other factors.

4.PROGRAMMING LANGUAGE

A programming language is a formal language comprising a set of instructions that produce various kinds of output. Programming languages are used in computer programming to implement algorithms and manipulate data structures

5.APPLICATION

An application, often referred to as an "app," is a software program designed to perform specific tasks or functions for the user.

Applications can range from simple tools like calculators and text editors to complex systems like enterprise resource planning (ERP) systems and customer relationship management (CRM) platforms.

* Desktop Applications

**Examples:** Microsoft Office, Adobe Photoshop, VLC Media Player.

* Web Applications

**Examples:** Google Docs, Facebook, Gmail.

* Mobile Applications

**Examples:** WhatsApp, Instagram, Uber.

* Enterprise Applications

**Examples:** SAP ERP, Salesforce, Oracle Database.

* Embedded Applications

**Examples:** Software in smart TVs, car navigation systems, medical devices.

* Cloud Applications

**Examples:** Dropbox, Google Drive, Slack.

6.VIRTUAL MACHINE

A virtual machine (VM) is like a computer within your computer. It allows you to run a different operating system and applications as if they were on a separate physical machine.

7.WEB APPLICATION

A virtual machine (VM) is like a computer within your computer. It allows you to run a different operating system and applications as if they were on a separate physical machine.

Key Features of Web Applications

 **Accessibility**

* Accessible from any device with a web browser and internet connection, including desktops, laptops, tablets, and smartphones.

 **Cross-Platform Compatibility**

* Can run on multiple operating systems (Windows, macOS, Linux) without requiring modifications.

 **Automatic Updates**

* Updates and maintenance are handled on the server side, ensuring users always access the latest version without manual installations.

 **Scalability**

* Can handle a growing number of users and data through server upgrades or cloud services.

 **User Interaction**

* Often feature dynamic content, allowing real-time interactions and updates (e.g., forms, chat, notifications).

### Common Technologies Used

1. **Frontend (Client-Side)**
   * **HTML (HyperText Markup Language):** Structures the content of the web pages.
   * **CSS (Cascading Style Sheets):** Styles the appearance of the web pages.
   * **JavaScript:** Adds interactivity and dynamic features (e.g., form validation, animations).
2. **Backend (Server-Side)**
   * **Languages:** Python, Ruby, PHP, Java, Node.js, etc., handle business logic and data processing.
   * **Databases:** MySQL, PostgreSQL, MongoDB, etc., store and manage data.
   * **Web Frameworks:** Django, Ruby on Rails, Express, Flask, etc., facilitate faster development.
3. **APIs (Application Programming Interfaces)**
   * Enable communication between the frontend and backend, often used for data exchange (e.g., RESTful APIs).

8.STANDALONE APPLICATION

A standalone application is a software program that operates independently on a user's device without needing a constant connection to a server or the internet.

It is self-contained and does not rely on external resources to function.

### Key Features of Standalone Applications

1. **Independence**
   * Runs independently on a local machine, making it accessible without internet access.
2. **Self-Contained**
   * All necessary resources, such as libraries and assets, are bundled with the application.
3. **Installation Required**
   * Typically installed on a computer or device, often using an installer package.
4. **User Interface**
   * Provides a user interface that allows users to interact directly with the application.
5. **Performance**
   * Can offer better performance for resource-intensive tasks since it utilizes local hardware directly

9.SOFTWARE DEVELOPMENT LIFE CYCLE

The Software Development Life Cycle (SDLC) is a systematic process used for designing, developing, testing, and deploying software applications.

The SDLC typically consists of several Phases

.  **Planning**

* **Goal:** Define the project scope, objectives, and feasibility.
* **Activities:**
  + Identify stakeholders and gather initial requirements.
  + Conduct feasibility studies (technical, economic, legal).
  + Develop a project plan, including timelines and resource allocation.
* **Deliverables:** Project charter, feasibility study report, project plan.

 **Requirements Analysis**

* **Goal:** Understand and document what the software should do.
* **Activities:**
  + Gather detailed requirements from stakeholders through interviews, surveys, and observations.
  + Create use cases and user stories.
  + Develop requirement specifications.
* **Deliverables:** Requirement specification document, use cases, user stories.

**Design**

* **Goal:** Define the architecture and design of the system.
* **Activities:**
  + Create high-level system architecture.
  + Develop detailed design documents, including data models, interface designs, and system interactions.
  + Choose technology stack and tools.
* **Deliverables:** System architecture diagram, detailed design documents, technology stack choices.

**Testing**

* **Goal:** Ensure the software is free of defects and meets requirements.
* **Activities:**
  + Develop test plans and test cases.
  + Conduct various types of testing (unit, integration, system, acceptance).
  + Identify and fix defects.
* **Deliverables:** Test plans, test cases, test reports, defect logs.

**Deployment**

* **Goal:** Release the software to users.
* **Activities:**
  + Prepare deployment environment.
  + Deploy software to production servers.
  + Conduct post-deployment verification.
* **Deliverables:** Deployed software, deployment documentation, verification reports.

**Maintenance**

* **Goal:** Keep the software operational and up-to-date.
* **Activities:**
  + Monitor the software for issues.
  + Provide bug fixes, patches, and updates.
  + Enhance software with new features as needed.
* **Deliverables:** Maintenance reports, updated software, user feedback.

**10.URL**

A URL (Uniform Resource Locator) is a specific type of Uniform Resource Identifier (URI) that is used to specify the address of a resource on the internet.

It provides a means to access web pages, files, and other resources.

A typical URL consists of several parts:

1. **Scheme/Protocol:**
   * Indicates the protocol used to access the resource (e.g., http, https, ftp).
   * Example: https://
2. **Host/Domain Name:**
   * Specifies the domain name or IP address of the server hosting the resource.
   * Example: www.example.com
3. **Port (optional):**
   * Indicates the port number on the server (default for HTTP is 80, and HTTPS is 443).
   * Example: :8080
4. **Path:**
   * Refers to the specific resource or location on the server.
   * Example: /path/to/resource
5. **Query String (optional):**
   * Contains additional parameters for the resource, usually in key-value pairs, starting with a ?.
   * Example: ?key1=value1&key2=value2
6. **Fragment (optional):**
   * Refers to a specific section within the resource, starting with a #.
   * Example: #section1

### Example of a Complete URL

https://www.example.com:8080/path/to/resource?key1=value1&key2=value2#section1

11.UI

UI (User Interface) refers to the point of interaction between a user and a computer system, application, or website.

It encompasses everything that a user interacts with, including screens, buttons, icons, menus, and any other elements that facilitate user engagement.

12.UX

UX (User Experience) refers to the overall experience a user has when interacting with a product, service, or system.

It encompasses all aspects of the user's interaction, including usability, accessibility, design, and the emotional response elicited by the interaction.

The goal of UX design is to create products that provide meaningful and relevant experiences to users.

13.API

An API (Application Programming Interface) is a set of rules that lets different software applications talk to each other. Think of it as a way for apps to request information or services from one another.

### Simple Explanation:

* **Imagine a Restaurant:** The menu is like the API. It lists the dishes you can order (the functions available) and how to request them (the rules).
* **How It Works:** You place your order (make a request), and the kitchen (another application) prepares your food and sends it back to you (returns the data).

14.IP(INTERNET PROTOCOL)

An IP (Internet Protocol) address is a unique identifier assigned to each device connected to a network that uses the Internet Protocol for communication.

It serves two main purposes: identifying the host or network interface and providing the location of the device in the network.

15.DEVOPS

DevOps is a practice that combines software development (Dev) and IT operations (Ops) to improve collaboration and efficiency.

### It aims to shorten the time it takes to develop, test, and release software while ensuring higher quality.

### Explanation:

* **Teamwork:** DevOps encourages developers and operations teams to work together closely.

16.AWS

* AWS (Amazon Web Services) is a comprehensive cloud computing platform provided by Amazon.
* It offers a wide range of services that enable businesses and individuals to build, deploy, and manage applications and services in the cloud.

17.AZURE

Azure, officially known as Microsoft Azure, is a cloud computing platform and service offered by Microsoft.

It provides a wide range of cloud services, including those for computing, analytics, storage, and networking, allowing users to build, deploy, and manage applications through Microsoft-managed data centers.

18.CLOUD COMPUTING

Cloud computing is a way to access and use computer services—like storage, servers, and software—over the internet instead of relying on local computers or servers.

### Simple Explanation:

* **Access Anytime, Anywhere:** You can use these services from any device with an internet connection.

19.FRONT END

Front-end refers to the part of a website or application that users interact with directly.

It encompasses everything that users see and engage with, including the layout, design, buttons, text, images, and overall user experience.

1. HTML (Hyper Text Markup Language)
2. CSS (Cascading Style Sheets)
3. Java Script
4. Angular JS
5. React JS

20.BACK END

Front-end refers to the part of a website or application that users interact with directly.

It encompasses everything that users see and engage with, including the layout, design, buttons, text, images, and overall user experience.

1. PHP
2. Go
3. Node JS
4. Java
5. Python

21.LIBRARY

A library is a collection of pre-written code that helps programmers do specific tasks without having to write all the code themselves.

It’s like a toolbox with tools that make building software faster and easier.

### Simple Explanation:

* **Reusable Code:** Libraries contain functions or components that you can use in your projects.
* **Saves Time:** Instead of coding everything from scratch, you can just use what's in the library.
* **Examples:** Popular libraries include jQuery for JavaScript and NumPy for Python

22.3-TIER ARCHITECTURE

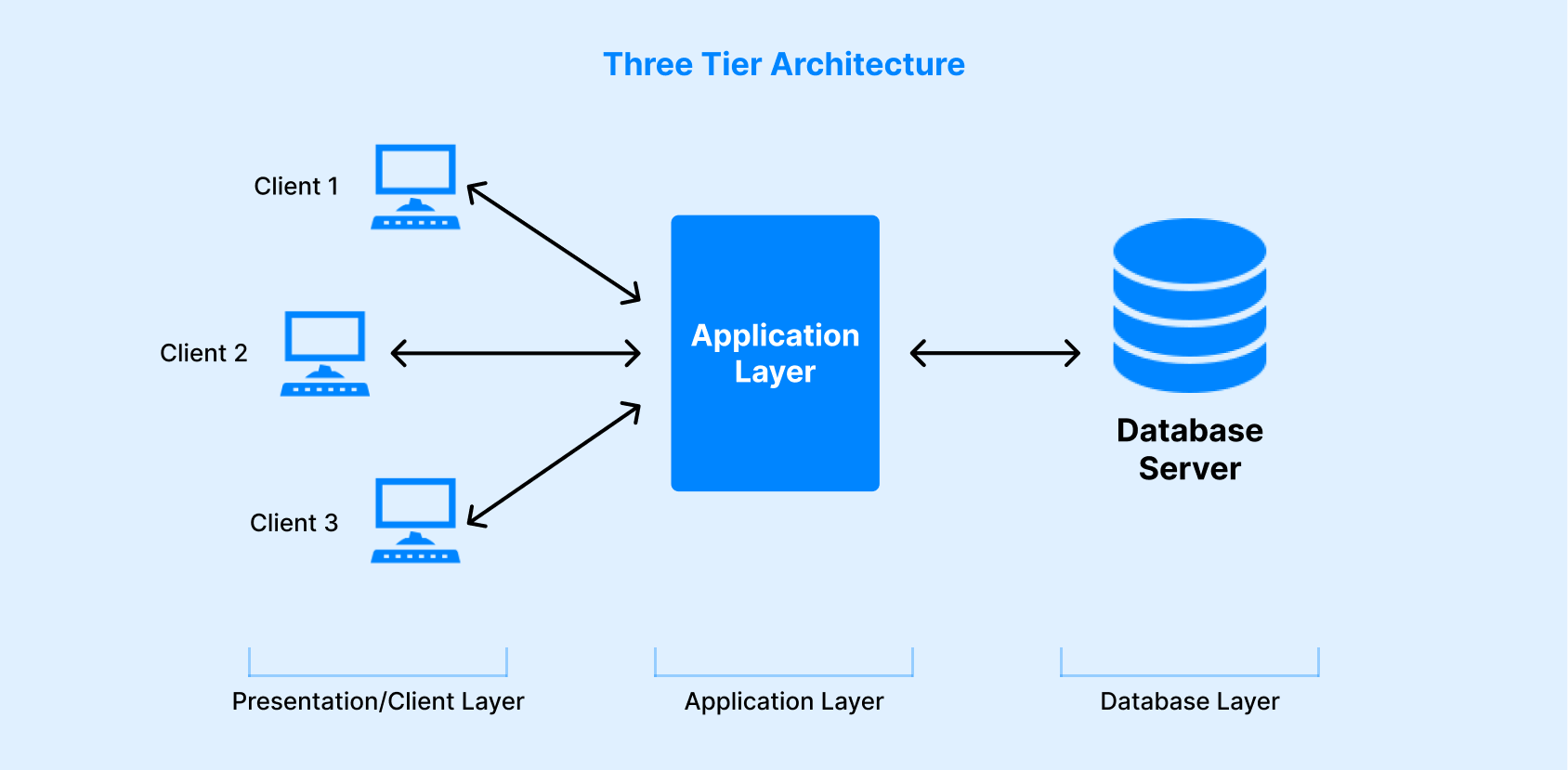
Three-tier architecture is a way of designing software applications that separates them into three main parts:

1. **Presentation Tier (User Interface):** This is where users interact with the application. It's the part you see and use, like a website or an app screen.
2. **Application Tier (Business Logic):** This is where the application processes the user's actions. It makes decisions, performs calculations, and runs the core functions of the application.
3. **Data Tier (Database):** This is where the application stores and retrieves data. It includes databases and the systems that manage them.

**Simple Example:**

Imagine a library system:

1. **Presentation Tier:** The library website where you search for books and check your account.
2. **Application Tier:** The system that processes your search request, checks your account details, and manages borrowing rules.
3. **Data Tier:** The database that stores information about all the books, users, and transactions.



23.SERVER

A server is a powerful computer that provides services, resources, or data to other computers (clients) over a network.

Servers play a crucial role in managing network resources, hosting websites, storing data, and running applications.

Types of Servers

1. Web Server
2. Database Server
3. File Server
4. Mail Server
5. Application Server
6. Proxy Server

24.EC2 INSTANCE

An EC2 instance is a virtual server provided by Amazon Web Services (AWS) as part of its Elastic Compute Cloud (EC2) service. It allows you to run applications on the cloud without needing to manage physical hardware.

**Key Features of EC2 Instances**

1. Scalability
2. Flexibility
3. Cost-Effectiveness
4. Security
5. Reliability

25.DEPLOYMENT

Deployment is the process of making an application or website available for people to use. It involves taking the finished product, moving it to a live server, and ensuring it runs properly so users can access it. In simple terms, deployment is about launching your project so everyone can see and interact with it.