Royal University of Bhutan

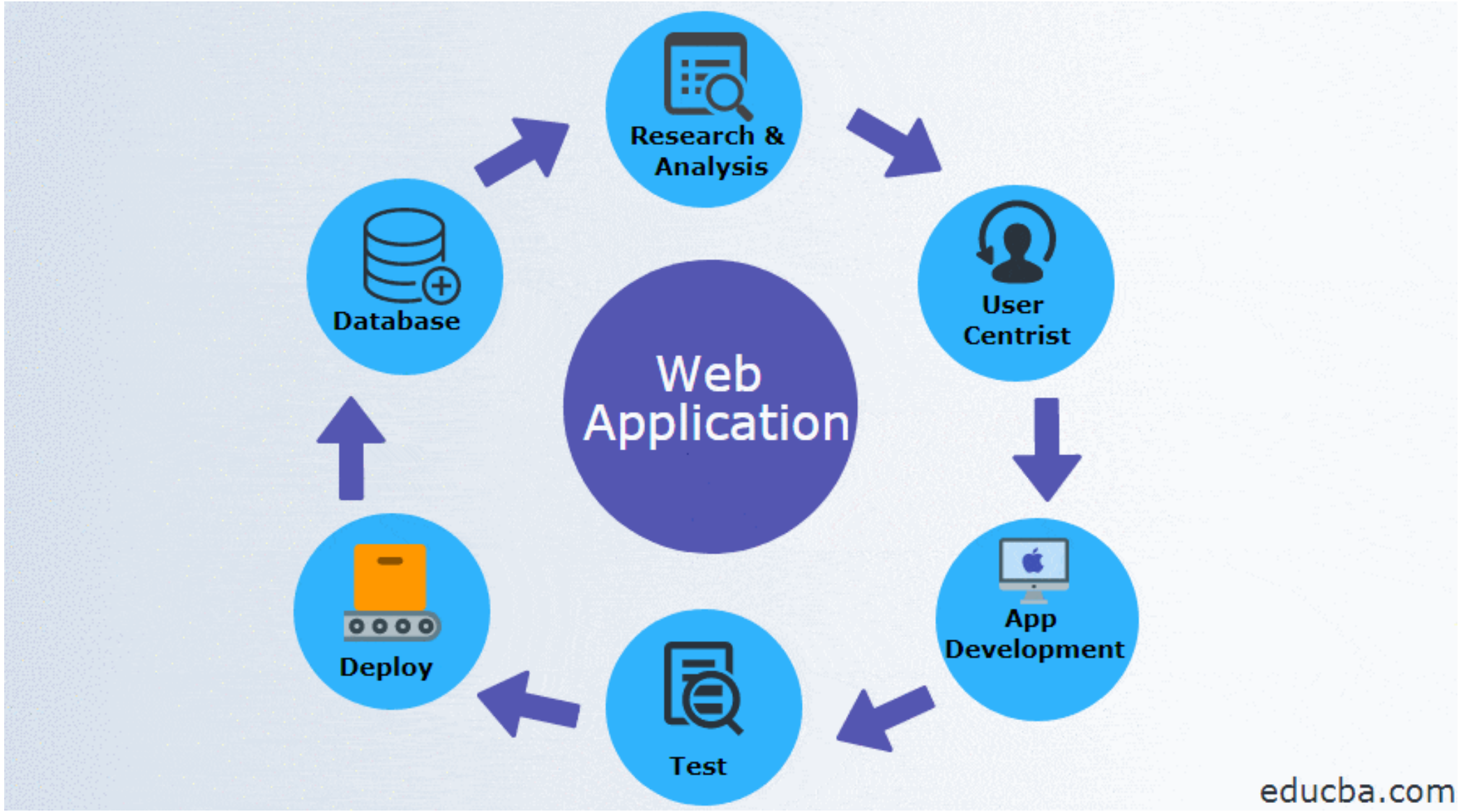
Unit I:
Introduction to Web Application Development (WAD)

Tutor: Pema Galey

Overview

In this session, you will learn about:

- Brief History of Web
- Internet & Web Protocols
- TCP/IP
- DNS
- HTTP/HTTPS
- Web Technologies
- Web App Development Process


Research & Analysis
User Centrist
App Development
Test
Deploy
Database
Web Application
educba.com

Brief History of Web

- World Wide Web commonly called as WWW or WEB is the strongest worldwide information medium.
- Information available on web is always accessible to every user through computer which is connected with internet.
- Most of the time people consider WEB and Internet are the similar words. Whereas **Web** is the service which is functioned through the **Internet**.
- Tim Berners-Lee, a British scientist, invented the World Wide Web (WWW) in 1989 and he wrote first computer program for the web browser.
- In the year 1990 Sir Tim Berners-Lee wrote three different technologies which are HTML, URI and HTTP.

Internet and Web Protocols

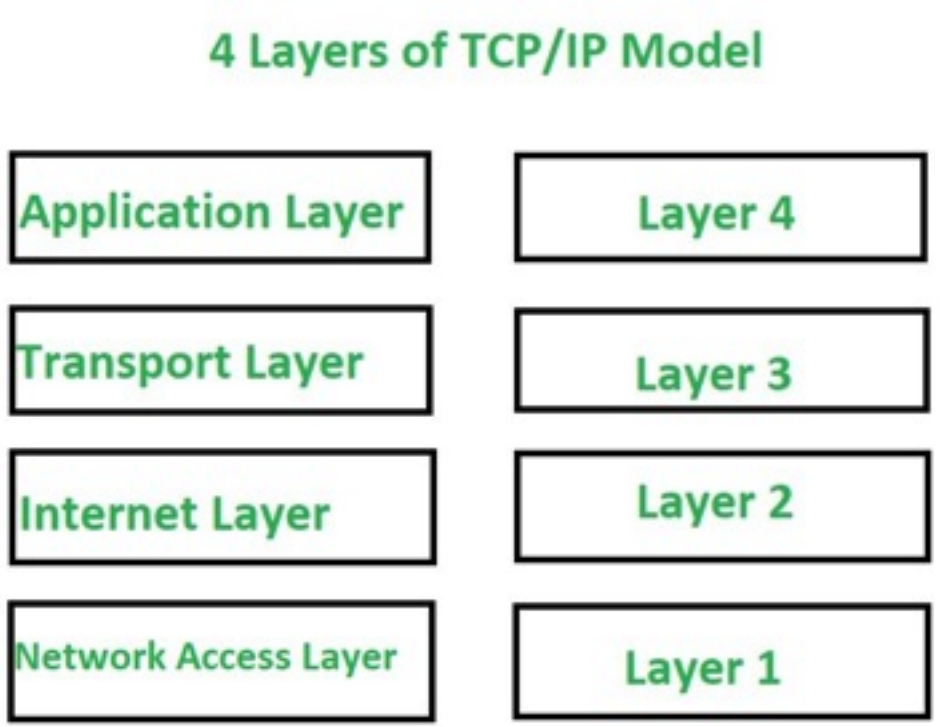
- Internet Protocols are a set of rules that governs the communication and exchange of data over the internet.
- Both the sender and receiver should follow the same protocols in order to communicate the data.
- The internet and many other data networks work by organizing data into small pieces called packets.

Types of Internet Protocols

1. TCP/IP(Transmission Control Protocol/ Internet Protocol)
2. SMTP(Simple Mail Transfer Protocol)
3. PPP(Point to Point Protocol)
4. FTP (File Transfer Protocol)
5. SFTP(Secure File Transfer Protocol)
6. HTTP(HyperText Transfer Protocol)
7. HTTPS(HyperText Transfer Protocol Secure)
8. TELNET(Terminal Network)
9. POP3(Post Office Protocol 3)
10. IPv4 and IPv6
11. ICMP (Internet Control Message Protocol)
12. UDP (User Datagram Protocol)
13. IMAP (Internet Message Access Protocol)
14. SSH (Secure Shell)

TCP/IP

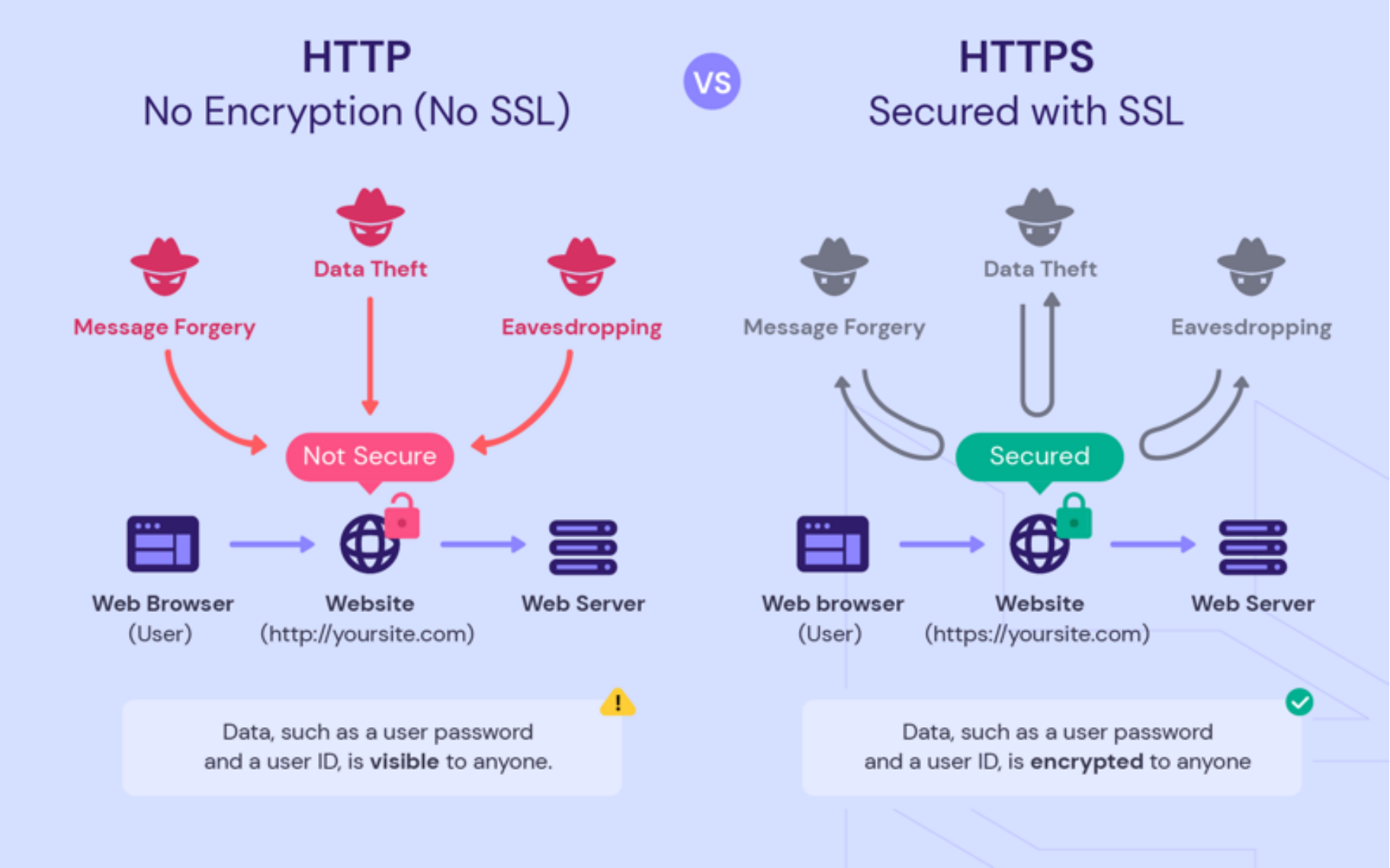
- A set of standard rules that allows different types of computers to communicate with each other.
- The IP protocol ensures that each computer that is connected to the Internet is having a specific serial number called the IP address



HTTP/HTTPS

HTTP	HTTPS
HyperText Transfer Protocol	HyperText Transfer Protocol Secure
It is not secure and unreliable	It is secure and reliable
Default port 80	Default 443
Operates at Application layer	Operates at Transport layer
No encryption and subject to main-in-middle	It is designed to withstand such attacks and considered secured against such attacks

HTTP/HTTPS


HTTP
No Encryption (No SSL)
VS
HTTPS
Secured with SSL
Message Forgery
Data Theft
Eavesdropping
Not Secure
Web Browser
(User)
Website
(http://yoursite.com)
Web Server
Data, such as a user password and a user ID, is visible to anyone.
Message Forgery
Data Theft
Eavesdropping
Secured
Web browser
(User)
Website
(https://yoursite.com)
Web Server
Data, such as a user password and a user ID, is encrypted to anyone

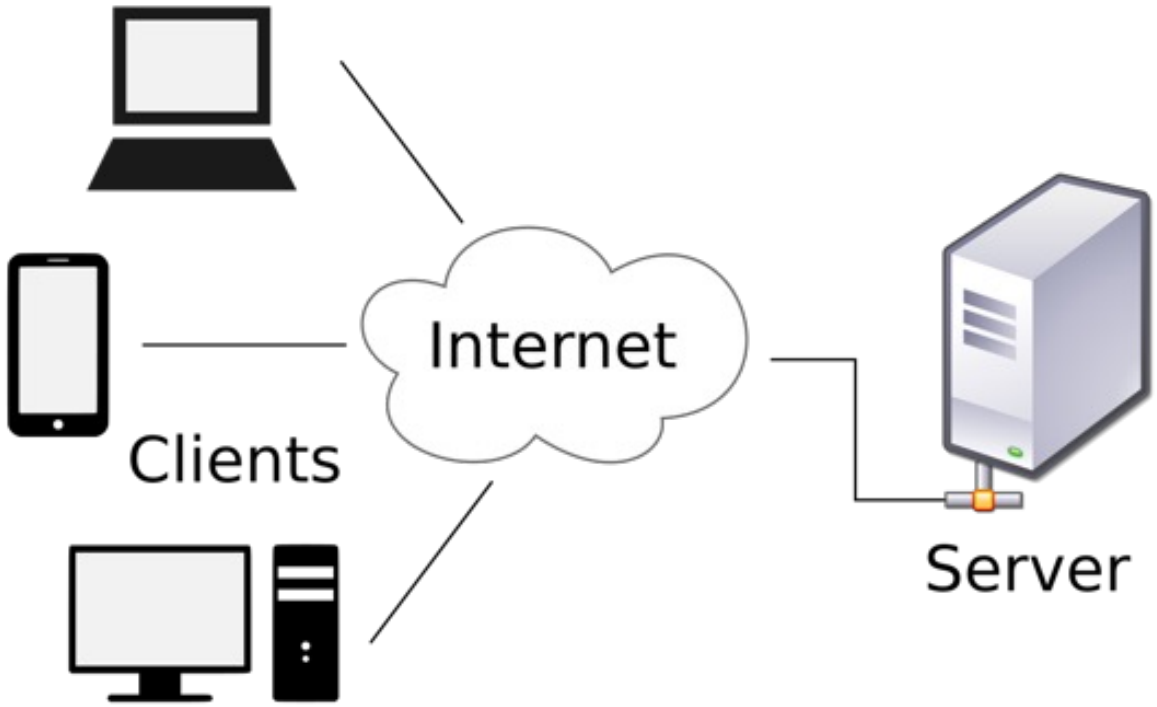
Domain Name System (DNS)

- The DNS is the phonebook of the Internet. Humans access information online through domain names, like cst.edu.bt or tashicell.com
- Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.

- Example:
  - www.cst.edu.bt - 103.133.216.194
  - www.tashicell.com - 118.103.136.91

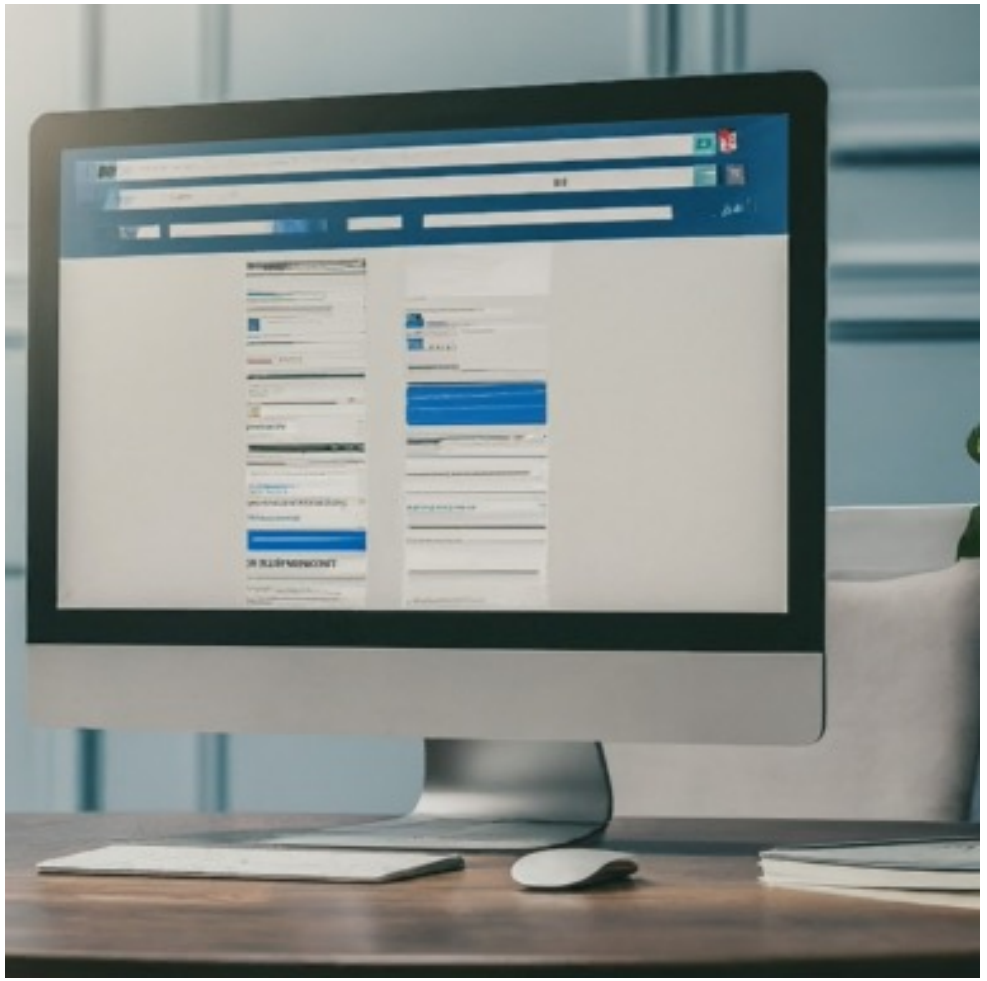
Client-Server Technology

- Client-Server Architecture: The cornerstone of web applications.
- Clients (web browsers): Software applications requesting information from servers.
- Servers: Powerful machines delivering requested data and running web applications.
- Communication Protocol: HTTP enables clients and servers to understand each other.



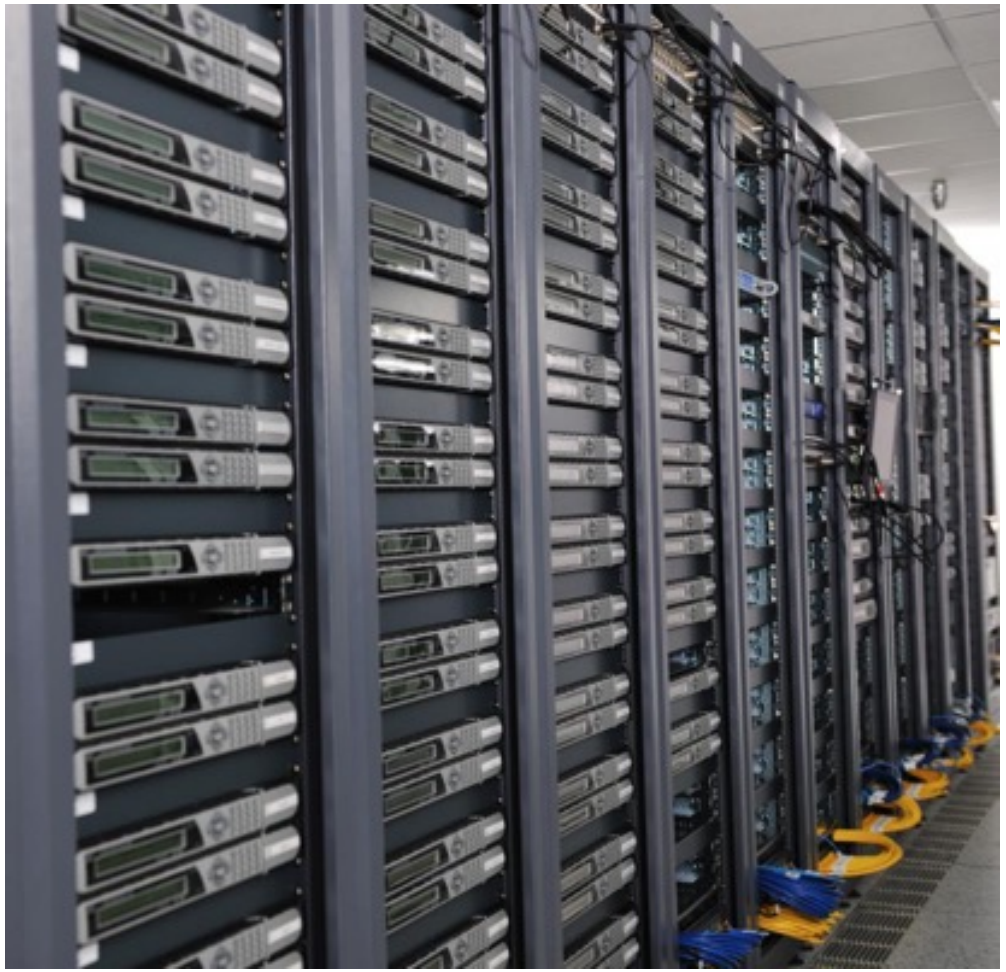
The Web Browser: Your Window to the Web

- Web browser: Software that interprets and displays web pages.
- Popular browsers: Chrome, Firefox, Safari, Edge.
- Rendering engine: Turns code into the visual elements you see.
- JavaScript engine: Enables interactivity and dynamic content.



The Web Server: The Heart of the Operation

- Web server: Software that receives client requests and delivers content.
- Popular web servers: Apache, Nginx, Microsoft IIS.
- Serves static files (HTML, CSS, images) and dynamic content (generated on-the-fly).
- Security measures: Protects sensitive information and ensures smooth operation.



Front-End vs. Back-End

- Front-End vs. Back-End: **Two Sides** of the same Coin for the application development
- Front-End: The user interface you interact with (design, visual elements, interactivity).
- Languages: HTML, CSS, JavaScript (React, Angular Vue.js).
- Back-End: The internal workings of the application (data storage, logic, server communication).
- Languages: Python, Java, PHP, Ruby.

Overview of Technology Stacks

- Technology stack: The combination of tools and languages used to build a web application.
- Front-End stack: Frameworks (React, Angular), libraries (Bootstrap), CSS preprocessors (Sass).
- Back-End stack: Programming languages (Python, Java), databases (MySQL, PostgreSQL), web servers (Apache, Nginx).
- Choosing the right stack: Depends on project requirements, performance needs, and developer expertise.

Full Stack Developer

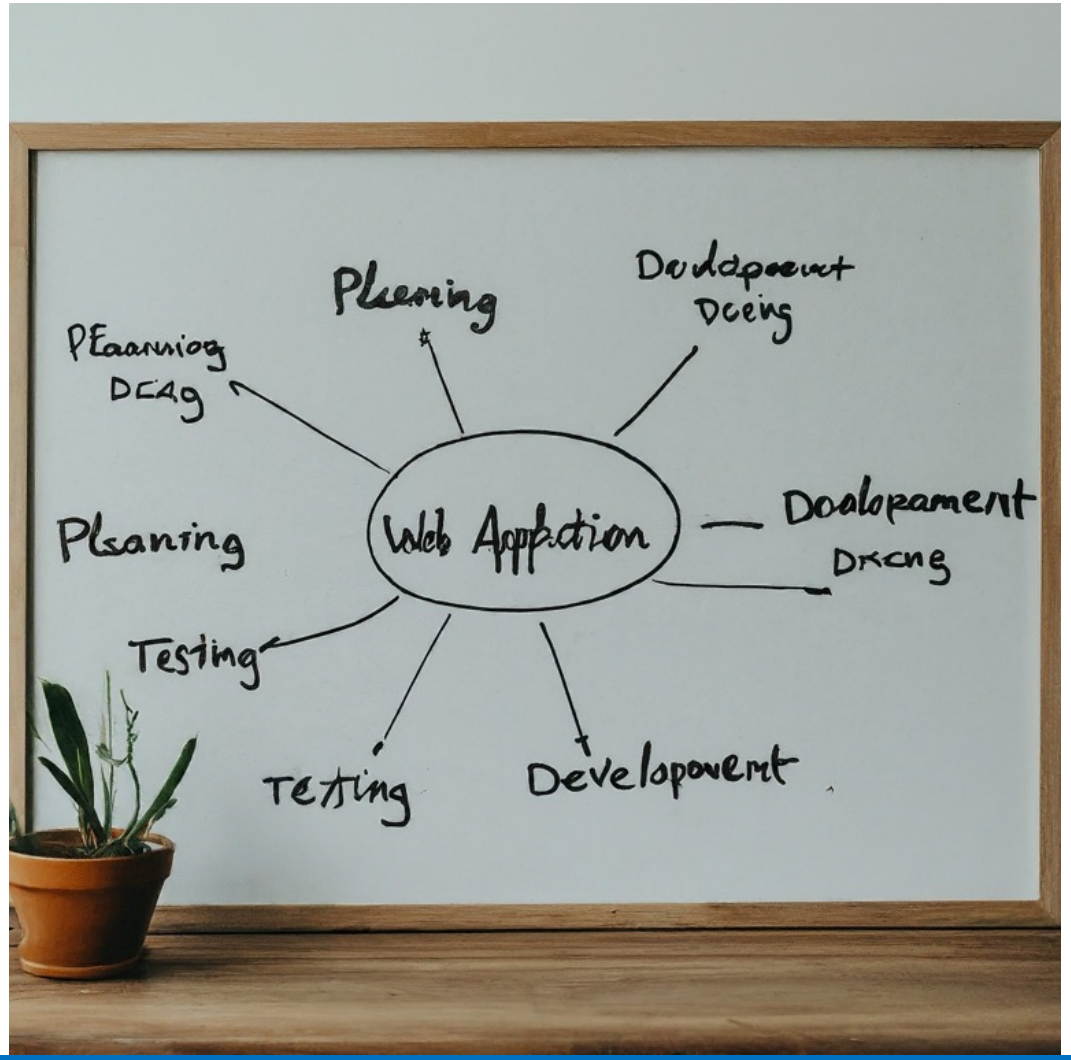
- Full Stack Developer: The Versatile Architect of Web Applications
- Full Stack Developer: Understands and works on both front-end and back-end development.
- Builds complete web applications, from user interface to server functionality.
- Possesses diverse skillset in various programming languages and technologies.
- Benefits: Flexibility, holistic understanding of the application, faster development.
- Challenges: Keeping up with both technology areas, and difficult to adopt all the technologies.

Website vs. Web App

- Website: Primarily static information display (blogs, portfolios, online brochures).
- Web App: Interactive functionalities for specific tasks (e-commerce, social media, online banking).
- Factors to consider: Purpose, user interaction needs, complexity of features.

Web Application Development Process

- Web Application Development Process: **From Idea to Reality**
- **Planning**: Defining goals, scope, target audience, and technology stack.
- **Design**: Creating user interface mockups and wireframes.
- **Development**: Building the front-end and back-end components.
- **Testing**: Ensuring functionality, performance, and security.
- **Deployment**: Launching the application on a web server and monitoring its performance.



Task (Peer Work)

- Select any for the Website or Web App Development
- Develop the basic document following the web application development process:
  - Plan - Plan your idea
  - Design - How to design?
  - Development - How to develop?
  - Deployment - How and where to deploy?

Thank you!