# **Made by: manthar ali**

# **Web engineering**

Web: The system of websites and online resources you access through the internet. (internet application)

Engineering (methodology): The process of designing and building things using a systemic way certified by an organization.

Web Engineering: Using engineering methods to create and maintain websites and web applications.

## **Software Engineering vs Web Engineering**

Software engineering: the process of designing and building all kinds of software applications, such as desktop programs and mobile apps.

Web engineering: the process of designing and building websites and web applications that run on the internet.

web application: an application you use through a web browser, like online email or a banking app.

website: a collection of web pages accessible via the internet, often for information or business purposes.

software: general term for programs or applications used on computers or devices, including both web and nonweb applications.

program: a single piece of software designed to perform a specific task, like a word processor or a game.

**Web applications come in various types, each serving different purposes. Here are some common types:**

1. **Static web applications**: display fixed content that doesn’t change based on user interaction, like simple informational websites.

2. **Dynamic web applications**: generate and display content dynamically based on user input or other factors, like social media platforms or news websites.

3. Ecommerce applications: enable online buying and selling of goods and services, such as online stores or marketplaces.

4. Content management systems (cms): allow users to create, manage, and modify digital content, like wordpress or joomla.

5. Web portals: provide access to various services and information through a single interface, like a company intranet or a news aggregation site.

**Simple definitions of common term:**

Web Page: A single document on the internet, like a page of a book.(executed on web browser)

Web Program: A software application that runs in a web browser, such as an online calculator or email client.

Website: A collection of related web pages accessible through a common domain name.

Browser: A tool (like Chrome or Firefox) used to view and interact with websites and web pages.

URL: The web address used to access a specific web page or resource on the internet.

**How web browser works in a few simple steps:**

1. User input: you enter a url (web address) into the browser’s address bar.

2. Request: the browser sends a request to the server that hosts the website.

3. Server response: the server sends back the web page's files (html, css, javascript) to the browser.

4. Rendering: the browser processes these files and displays the web page on your screen.

5. Interaction: you interact with the web page, and the browser may send more requests or update the page as needed.

# **Web architecture**

a web application architecture is a model that defines how the components of a web app interact with each other.

For instance, a client-server architecture, where the client is the user interface and the server is the back-end, is a common model.

Web architecture consists of several components, including the client, the server, the network, and the database.

# **Why is Web Architecture Important?**

Web architecture is important because it plays a crucial role in the performance, scalability, and maintenance of a website or web application.



# What is the 3-Tier Architecture?

The 3-Tier Architecture, also known as the three-layer architecture, is a client-server software architecture that separates an application into three distinct layers, or tiers. The purpose of this architecture is to improve modularity, maintainability, scalability, reliability, and flexibility of the software system.

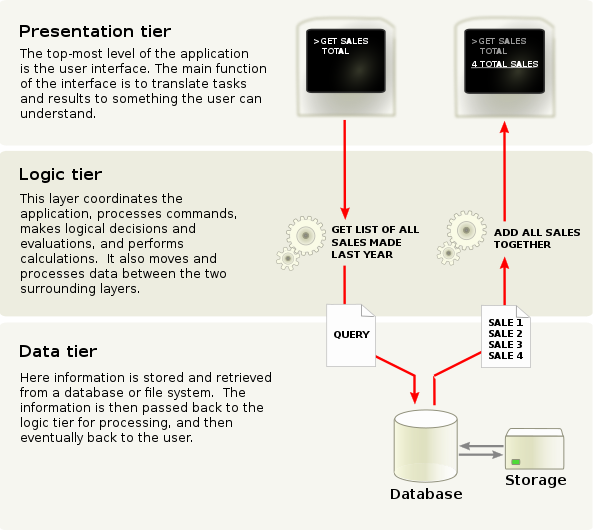
The 3-Tier Architecture divides an application into three layers to make it easier to manage and scale:

1. Client-Tier: This is the user interface where you interact with the app. For example, a web page or mobile app screen where you enter and view information.

2. Server-Tier: This layer handles the app’s logic and processes data. For example, it manages user logins, retrieves data from a database, and runs the app’s core functions.

3. Data-Tier: This layer stores and retrieves data. For example, it includes the database where user information is saved and the code that reads from and writes to this database.

In short, the client-tier is what you see, the server-tier handles the app’s operations, and the data-tier manages the data.



**Server side and client side scripting**

A scripting language is a programming language used to write small blocks of code for automating tasks and controlling applications. It is usually interpreted, running line-by-line at runtime, and is designed to integrate with other languages and systems. Client-Side Scripting and Server-Side Scripting are two types of scripting used in web development.

Client-Side Scripting: Code executed in the user's browser.

Visible to the user, used for frontend operations, collects user input, not ideal for complex tasks, generally less secure.

Languages: HTML, CSS, JavaScript.

Server-Side Scripting: Code executed on the server before sending data to the user’s browser.

Characteristics: Not visible to the user, used for backend operations, processes user input, handles complex computations, more secure.

Languages: Node.js, PHP, Python, Java.

**Introduction to Programming Models**

Synchronous Programming

Synchronous means that code is executed sequentially from top to bottom. Each statement is completed before the next one begins.

For example, in the following synchronous code:

firstStatement();

secondStatement();

thirdStatement();

**output**

firstStatement() will execute fully before secondStatement() begins, and secondStatement() will finish before thirdStatement() starts.

Asynchronous Programming

Due to this nature of synchronous programming, sometimes important instructions get blocked due to some previous instructions, which causes a delay in the user interface. Asynchronous code execution allows to execution next instructions immediately and doesn't block the flow because of previous instructions.

console.log('First');

// Set timeout for 2 seconds

setTimeout(() => console.log('Second'), 2000);

console.log('Third');

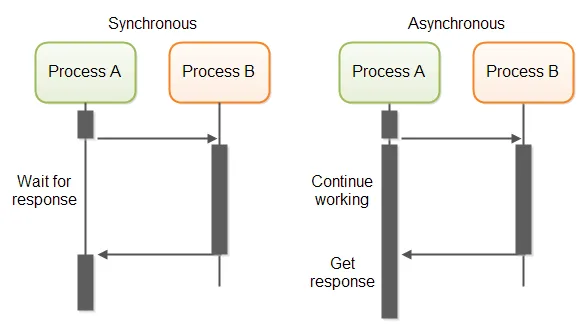
output

First

Third

Second

As we can see in the output of the above code example, Third gets printed before Second, because of the asynchronous execution of the code. Here setTimeout() function waits for 2 seconds, and in the meantime, the next instruction gets executed without waiting for the previous one to complete the execution.



**AJAX IMPLemantation**

AJAX (Asynchronous JavaScript and XML) is a technique used in web development to create interactive and dynamic web applications. It allows web pages to update content asynchronously without reloading the entire page, by sending and receiving data in the background using JavaScript and XML (or JSON).

An example of AJAX implementation is a live search feature on a website. As you type a query into a search box, AJAX sends the query to the server and retrieves search results without refreshing the entire page. The results are then displayed dynamically in real-time, providing a seamless user experience.

Is a technology which is used to talk with web servers asynchronously using JavaScript.

Load web content without reloading

# lab web engineering

Html

Boilerplate

H1 – h6

Marquee

CSS

Internal

External

Inline

Selectors(tag selector,id selector

Javascript

**PHP (Hypertext Preprocessor)**

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

A PHP script is executed on the server, and the plain HTML result is sent back to the browser.

Basic PHP Syntax

A PHP script can be placed anywhere in the document.

A PHP script starts with <?php and ends with ?>:

Basic syntax

<?php

echo "Hello World!";

?>

Note: PHP statements end with a semicolon (;).

What Can PHP Do?

• PHP can generate dynamic page content

• PHP can create, open, read, write, delete, and close files on the server

• PHP can collect form data

• PHP can send and receive cookies

• PHP can add, delete, modify data in your database

• PHP can be used to control user-access

• PHP can encrypt data

## **PHP Case Sensitivity**

In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are not case-sensitive.

In the example below, all three echo statements below are equal and legal:

**Note:** However; all variable names are case-sensitive!

## **Creating (Declaring) PHP Variables**

a variable starts with the $ sign, followed by the name of the variable

### **Example**

$x = 5;

$y = "John";

## **Output Variables**

The PHP echo statement is often used to output data to the screen.

The following example will show how to output text and a variable:

### **Example**

$txt = "W3Schools.com";

echo "I love $txt!";

PHP supports the following data types:

* String
* Integer
* Float (floating point numbers - also called double)
* Boolean
* Array
* Object
* NULL
* Resource