IBRAHIM BADAMASI BABANGIDA UNIVERSITY LAPAI

SULEJA STUDY CENTER

DEPARTMENT OF COMPUTER SCIENCE

CSC 103: INTRO TO INTERNET AND WEB DESIGN

WEEK 1: Introduction to the Internet

- Course Overview
  - Introduction to the course
  - Objectives and expectations
- History of the Internet
  - Origins and evolution
  - Key milestones
- Understanding the Internet
  - How the Internet works (basic concepts)
  - Key terms (e.g., ISP, URL, HTTP, IP Address)

WEEK 2: Internet Technologies and Protocols

- Web Browsers
  - Types and functions of web browsers
  - Browser features and usage
- Internet Protocols
  - HTTP/HTTPS
  - FTP
  - TCP/IP
- Introduction to DNS
  - What is DNS
  - How DNS works

WEEK 3: Basics of Web Design

- Introduction to Web Design
  - Principles of good web design
  - Understanding user experience (UX) and user interface (UI)
- Web Design Tools
  - Overview of popular web design tools (e.g., Adobe XD, Figma)

- HTML Basics

  ° Structure of an HTML document

  ° Common HTML tags

WEEK 4: HTML in Detail

- HTML Elements

  ° Text elements

  ° Links, images, and media

- HTML Forms

  ° Creating forms

  ° Form elements and attributes

WEEK 5: Introduction to CSS

- CSS Basics

  ° What is CSS

  ° Linking CSS to HTML

- CSS Syntax

  ° Selectors, properties, and values

  ° Basic styling (color, fonts, layout)

WEEK 6: Advanced CSS

- Box Model

  ° Margin, border, padding, and content

- CSS Flexbox and Grid

  ° Flexbox layout

  ° Grid layout

- Responsive Design

  ° Media queries

  ° Mobile-first design

WEEK 7: Introduction to JavaScript

- JavaScript Basics
  - What is JavaScript
  - Linking JavaScript to HTML
- Basic JavaScript Syntax
  - Variables, data types, and operators
  - Functions and events

WEEK 8: Advanced JavaScript and DOM Manipulation

- DOM (Document Object Model)
  - Understanding the DOM
  - Selecting and manipulating DOM elements
- Event Handling
  - Adding interactivity
  - Common events and event listeners

WEEK 9: Web Development Frameworks

- Introduction to Frameworks
  - Overview of popular frameworks (e.g., Bootstrap, React)
- Using Bootstrap
  - Basic Bootstrap components
  - Building responsive layouts with Bootstrap

MODULE 1:

Definition: **The Internet is a massive network of computers worldwide that are connected and can communicate with each other like cables, WiFi, modem, etc. A global web that links millions of smaller networks. It's the backbone of modern digital communication, allowing us to share information quickly and easily.**

BRIEF HISTORY OF INTERNET

- **Origins: The Internet started as ARPANET, a project by the U.S. Department of Defense in the late 1960s to connect computers for research purposes.**
- **Development: Throughout the 1970s and 1980s, it grew as universities and research institutions began using it for sharing information and collaborating on projects.**
- **Commercialization: In the 1990s, the Internet became accessible to the public, leading to its rapid expansion and the birth of the World Wide Web.**

Key Components:

- **Hardware: This includes all the physical parts like servers, routers, and cables that form the infrastructure of the Internet.**
- **Protocols: These are rules that determine how data is transmitted over the Internet. The most important ones are TCP/IP, which ensure data can travel from one computer to another reliably.**
- **Domain Names and IP Addresses: Every computer on the Internet has an IP address, a unique identifier. Domain names are human-readable addresses (like www.example.com) that are easier to remember than numerical IP addresses.**

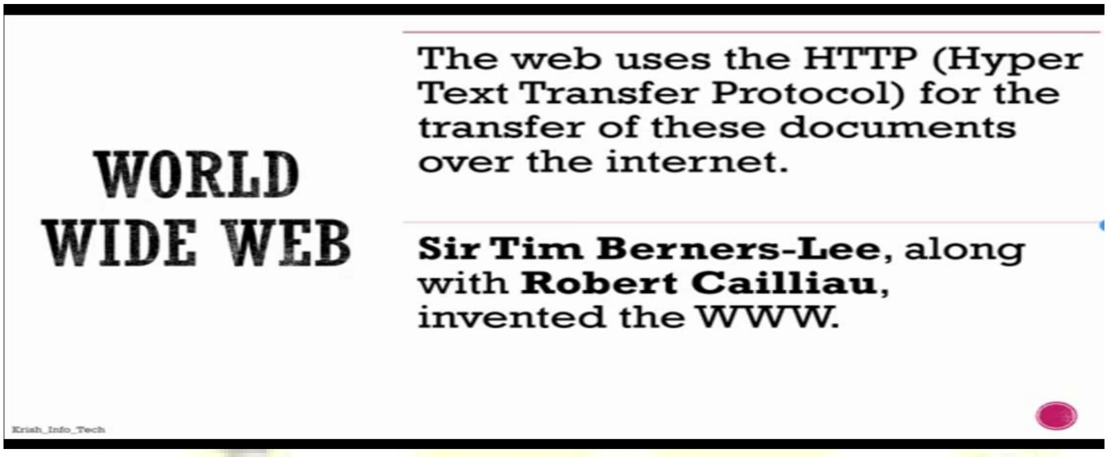
How It Works:

- **Data Transmission: Information on the Internet is broken into smaller packets. These packets travel independently across the network and are reassembled at the destination.**

- **Client-Server Model: When you use the Internet, your device (the client) requests information from another computer (the server). The server then provides the requested data, such as web pages or emails.**

Services and Uses:

- **World Wide Web: This is a system of interlinked documents (web pages) accessed through web browsers.**



WORLD WIDE WEB

The web uses the HTTP (Hyper Text Transfer Protocol) for the transfer of these documents over the internet.

Sir Tim Berners-Lee, along with **Robert Cailliau**, invented the WWW.

Krish_Info_Tech

- **Email: A method for sending electronic messages between people.**
- **Social Media: Platforms where people can interact, share content, and communicate.**
- **E-commerce: Buying and selling goods and services online.**
- **Streaming Services: Websites and applications that deliver audio and video content over the Internet.**

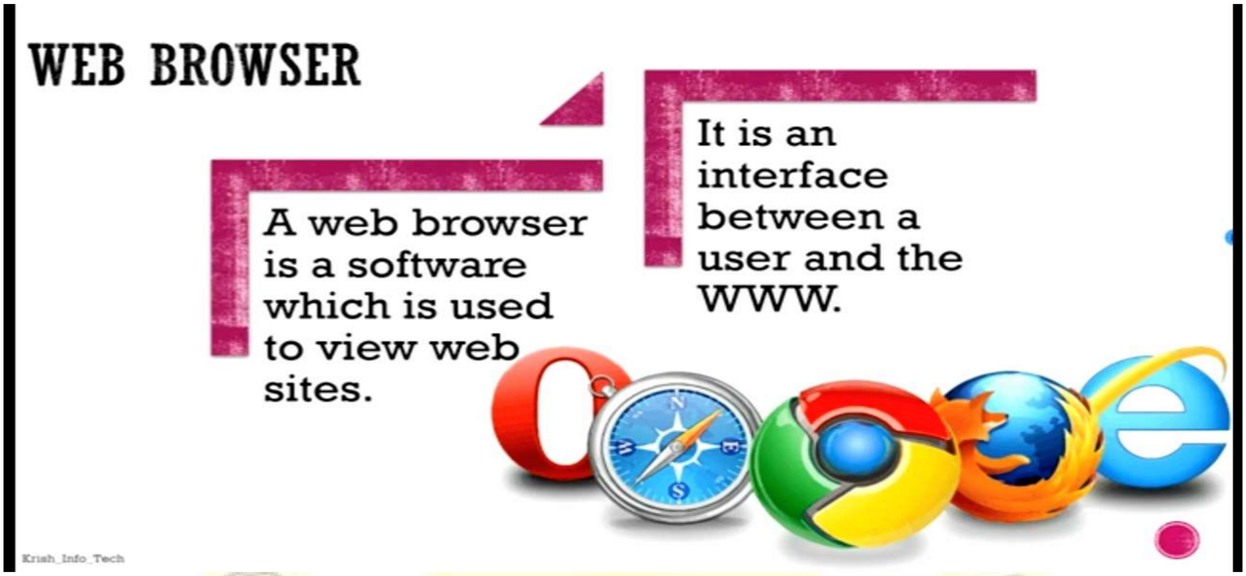
Importance:

- **Communication: The Internet makes it possible to communicate instantly with people anywhere in the world.**
- **Information Access: It provides vast amounts of information on almost any topic imaginable.**
- **Economic Impact: The Internet has revolutionized business, enabling new models like online shopping and digital marketing.**
- **Social Interaction: Social media and other platforms enable people to connect, share, and collaborate.**

MODULE 2:

INTERNET TECHNOLOGIES AND PROTOCOLS

Web Browsers

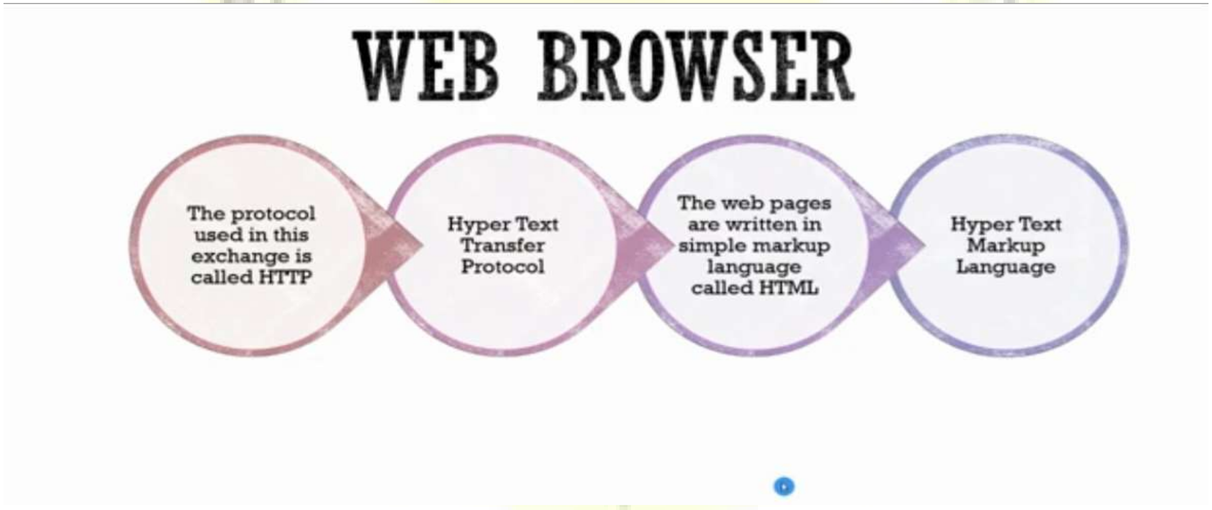


WEB BROWSER

A web browser is a software which is used to view web sites.

It is an interface between a user and the WWW.

Krish_Info_Tech



WEB BROWSER

The protocol used in this exchange is called HTTP

Hyper Text Transfer Protocol

The web pages are written in simple markup language called HTML

Hyper Text Markup Language

Types and Functions of Web Browsers:

- **Types: There are several types of web browsers available, each with its unique features. Common examples include Google Chrome, Mozilla Firefox, Microsoft Edge, Safari, and Opera.**

- **Functions: Web browsers are software applications used to access and view websites. They interpret and display HTML, CSS, and JavaScript code, allowing users to interact with web content.**

BROWSER FEATURES AND USAGE

• Tabs: Allow users to open multiple web pages in a single window.

• Bookmarks: Enable users to save and organize their favorite web pages for quick access.

History: Tracks the websites visited by the user, making it easy to revisit previous pages.

• Extensions/Add-ons: Small software modules that add specific features to a browser, such as ad blockers or password managers.

• Privacy Modes: Features like Incognito Mode (Chrome) or Private Browsing (Firefox) prevent the browser from saving history, cookies, and site data.

INTERNET PROTOCOLS

HTTP/HTTPS:

HTTP (HyperText Transfer Protocol): The foundation of data communication on the web, HTTP is used to transfer hypertext documents. It defines how messages are formatted and transmitted, and how web servers and browsers should respond to various commands.

• HTTPS (HTTP Secure): An extension of HTTP, HTTPS is used for secure communication over a computer network. It encrypts the data transferred between the web browser and server, providing a secure channel to prevent eavesdropping and tampering.

FTP (File Transfer Protocol):

FTP: A standard network protocol used to transfer files from one host to another over the Internet. It is commonly used for uploading and downloading files to and from a server.

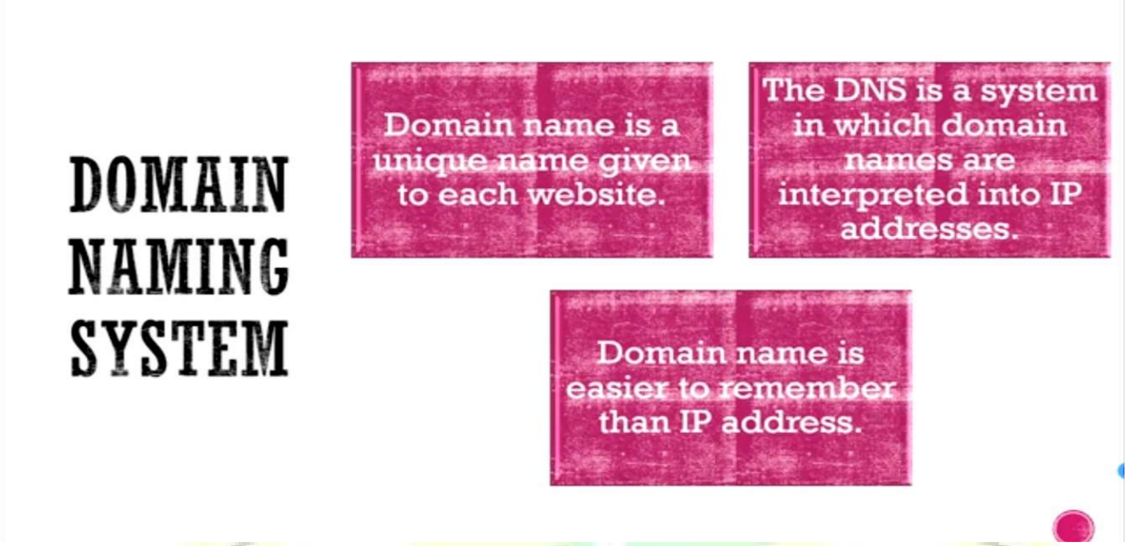
TCP/IP (Transmission Control Protocol/Internet Protocol):

TCP/IP: The fundamental suite of protocols that governs the Internet. TCP ensures reliable data transmission between devices, while IP handles addressing and routing data packets to their destination. Together, they enable different networks to communicate effectively.

INTRODUCTION TO DNS

What is DNS (Domain Name System):

DNS: The system that translates human-readable domain names (like www.example.com) into numerical IP addresses (like 192.0.2.1) that computers use to identify each other on the network.



DOMAIN NAMING SYSTEM

Domain name is a unique name given to each website.

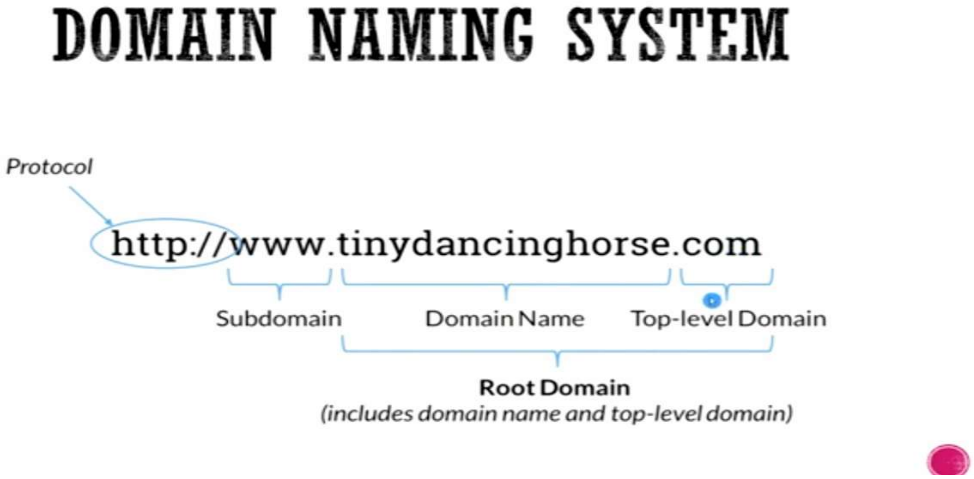
The DNS is a system in which domain names are interpreted into IP addresses.

Domain name is easier to remember than IP address.

How DNS Works:

Domain Name Entry: When you type a domain name into your web browser, the browser first checks its cache to see if it already knows the corresponding IP address.

Query DNS Resolver: If the IP address is not cached, the browser sends a query to a DNS resolver, usually provided by your Internet Service Provider (ISP).



DOMAIN NAMING SYSTEM

Protocol
http://www.tinydancinghorse.com
Subdomain
Domain Name
Top-level Domain
Root Domain
(includes domain name and top-level domain)

Recursive Query: The DNS resolver then queries other DNS servers in a hierarchical order. It starts with the root DNS servers, which point to the DNS servers responsible for the top-level domain (TLD) like .com or .org.TLD Servers: The TLD servers direct the query to the authoritative DNS servers for the specific domain.

Authoritative DNS Servers: These servers contain the actual IP address for the requested domain and return it to the DNS resolver.

IP Address Returned: The resolver sends the IP address back to the browser, which can then connect to the web server hosting the site.

By understanding these technologies and protocols, you can better comprehend how the Internet functions and how to use it more effectively.

MODULE 3:

BASICS OF WEB DESIGN

Introduction to Web Design

Web design is the process of creating the visual and interactive aspects of a website. It involves planning and building the elements that make up a website, including layout, content, graphics, and user interface. Good web design combines aesthetics and functionality to provide an engaging and effective user experience.

Principles of Good Web Design

1. Clarity and Simplicity:

Clarity: Information should be presented clearly and logically, making it easy for users to understand.

Simplicity: Avoid clutter and unnecessary elements. Focus on essential features to make navigation intuitive.

2. Consistency: Layout: Use a consistent layout across all pages to provide a cohesive experience.

° Color Scheme and Typography: Maintain a consistent color palette and typography to enhance brand identity and readability.

3. Responsiveness:

° Adaptability: Design should adapt to different screen sizes and devices, ensuring a seamless experience on desktops, tablets, and smartphones.

° Fluid Grids and Flexible Images: Use flexible grid layouts and scalable images to adjust to varying screen sizes.

4. Accessibility:

1° Inclusivity: Ensure the website is usable by people with disabilities, following accessibility standards like the Web Content Accessibility Guidelines (WCAG).

° Keyboard Navigation: Make sure all interactive elements can be accessed and used via keyboard.

5. Visual Hierarchy: Guiding Attention: Use size, color, contrast, and spacing to create a visual hierarchy that guides users' attention to the most important elements.

° Prioritization: Arrange content in a way that reflects its importance, making it easy for users to find key information quickly.

6. Loading Speed:

° Optimization: Optimize images, scripts, and other resources to minimize load times, providing a better user experience and improving search engine rankings.

° Performance: Use techniques like caching, compression, and asynchronous loading to enhance performance.

Understanding User Experience (UX) and User Interface (UI)

User Experience (UX):

Definition: UX focuses on the overall experience of the user as they interact with the website, aiming to make it enjoyable and efficient.

Components: Includes usability, accessibility, information architecture, and user research.

Goal: To create a smooth, intuitive, and satisfying interaction that meets users' needs and expectations.

User Interface (UI)

Definition: UI deals with the visual and interactive elements of a website, such as buttons, icons, and menus.

Components: Includes layout, visual design, typography, and interactive elements.

Goal: To make the interface aesthetically pleasing and easy to use, enhancing the overall user experience.

Web Design Tools

Overview of Popular Web Design Tools: Adobe XD: A powerful design and prototyping tool used to create user interfaces and user experiences for web and mobile apps. It offers features like vector design, wireframing, and interactive prototypes Figma: A cloud-based design tool that allows real-time collaboration on interface design projects. It supports vector graphics, prototyping, and design systems, making it ideal for team projects.

HTML Basics

Structure of an HTML Document: HTML (HyperText Markup Language) is the standard language used to create web pages. An HTML document typically includes the following structure:

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>:
```

Declares the document type and version of HTML being used.:

The root element that contains all the content of the web page.: Contains meta-information about the document, such as the title, character set, and links to style

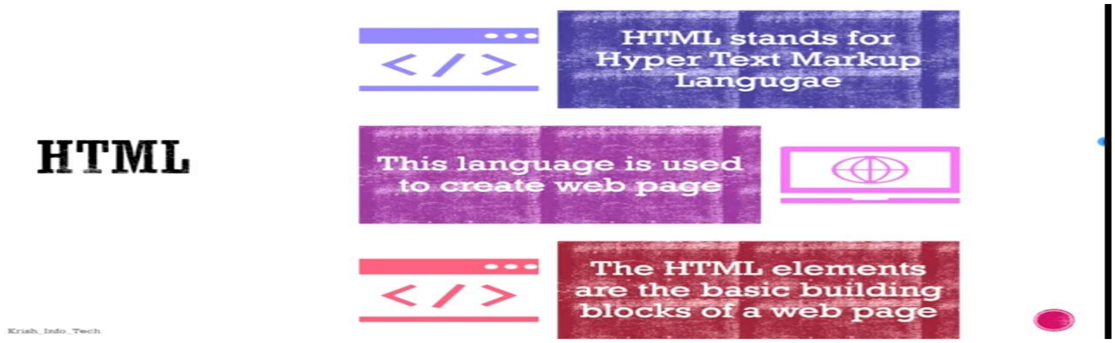
Sheets.:

Sets the title of the web page, which appears in the browser tab.: Contains the content of the web page, such as text, images, and other media.

Common HTML Tags: to : Heading tags, with being the highest level and the lowest.: Paragraph tag, used to define blocks of text.: Anchor tag, used to create hyperlinks.: Image tag, used to embed images in a web page. and : Unordered and ordered list tags, used to create lists.: List item tag, used within and to define individual items.: Division tag, used to group block-level content.: Inline container tag, used to group inline content.

MODULE 4:

HTML IN DETAIL


HTML
HTML stands for Hyper Text Markup Langugae
This language is used to create web page
The HTML elements are the basic building blocks of a web page
Krish_Info_Tech

HTML Elements

Text Elements:

Headings: <h1>, <h2>, etc. for different levels of headings. Paragraphs: <p> for paragraphs.

Bold and Italic: <b> for bold text, <i> for italic text.

Lists: <ul> for unordered lists, <ol> for ordered lists, <li> for list items.

Divisions and Spans: <div> for block-level division, <span> for inline division.

Links, Images, and Media

Links: <a href="URL">link text</a> for creating hyperlinks.

Images (<img>): Embeds images in a web page. The src attribute specifies the path to the image, and the alt attribute provides alternative text for screen readers.

```
<img src="image.jpg" alt="Description of image">
```

Media: <video>, <audio>, <iframe> for embedding multimedia content.

HTML FORMS

Creating Forms

Basic Structure of HTML Forms: Forms are created using the <form> tag. The action attribute specifies where to send the form data, and the method attribute specifies how to send the data (GET or POST).

```
<form action="/submit" method="post">
  <!-- Form elements go here -->
```

</form>

Form Element: <form> as a container for input elements.

Action and Method: Attributes like action (URL to submit data) and method (GET or POST).

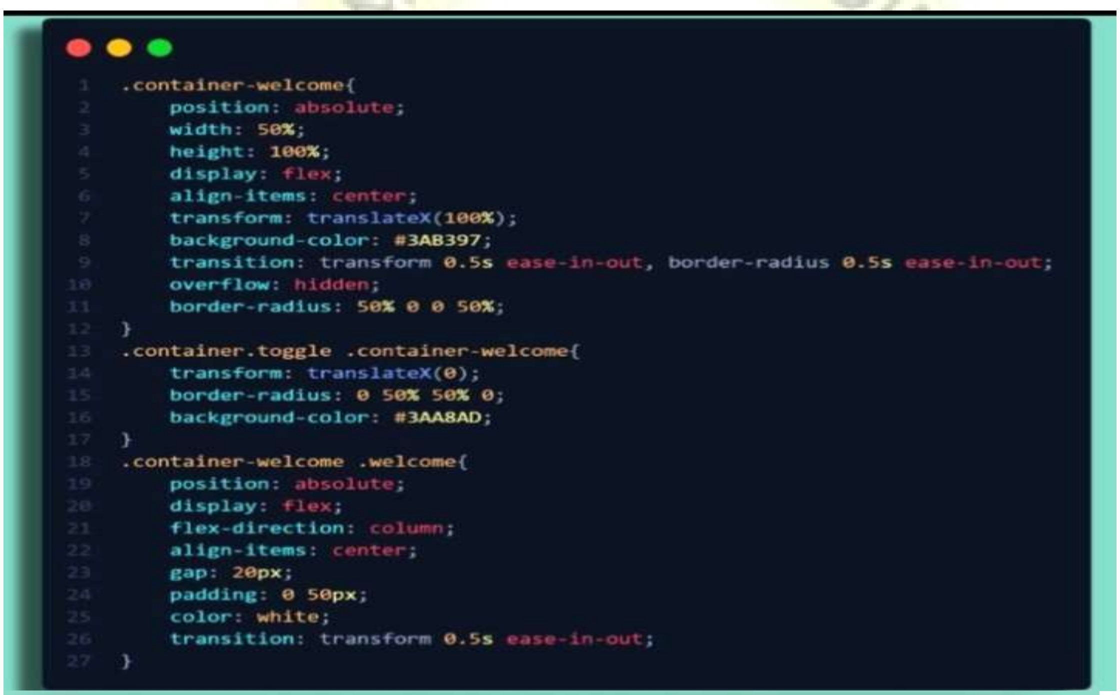
Form Elements and Attributes Input Fields: <input type="text">, <input type="password">, <input type="email">, etc.

Labels: <label> to associate text labels with form elements.

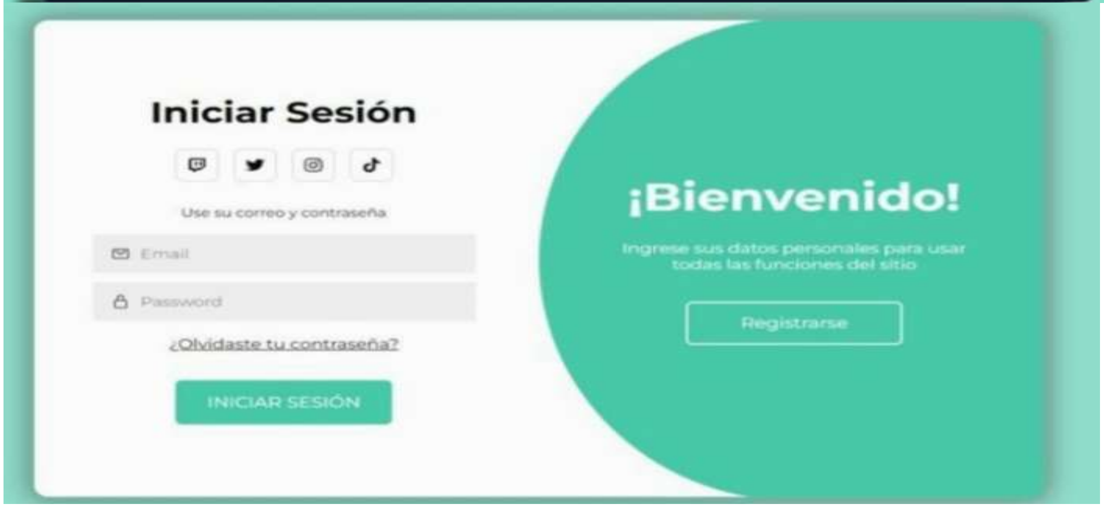
Buttons: <button>, <input type="submit">, <input type="reset">.

Select Boxes: <select> and <option> for dropdown lists.

Text Areas: <textarea> for multi-line text input.



```css
.container-welcome{
    position: absolute;
    width: 50%;
    height: 100%;
    display: flex;
    align-items: center;
    transform: translateX(100%);
    background-color: #3AB397;
    transition: transform 0.5s ease-in-out, border-radius 0.5s ease-in-out;
    overflow: hidden;
    border-radius: 50% 0 0 50%;
}
.container.toggle .container-welcome{
    transform: translateX(0);
    border-radius: 0 50% 50% 0;
    background-color: #3AA8AD;
}
.container-welcome .welcome{
    position: absolute;
    display: flex;
    flex-direction: column;
    align-items: center;
    gap: 20px;
    padding: 0 50px;
    color: white;
    transition: transform 0.5s ease-in-out;
}
```



Ingrese sus datos personales para usar todas las funciones del sitio

MODULE 5:

INTRODUCTION TO CSS

CSS Basics

What is CSS (Cascading Style Sheets): Introduction to CSS as a style sheet language used to describe the presentation of a document written in HTML. CSS controls the layout of multiple web pages all at once and allows for separation of content (HTML) and design (CSS).

Linking CSS to HTML: Various methods to incorporate CSS into

HTML documents:

Inline CSS: Adding CSS styles directly within HTML elements using the style attribute, e.g., <p style="color: blue;">This is a paragraph.</p>.

Internal CSS: Writing CSS within a <style> tag inside the <head> section of an HTML document, e.g.,<head>

```
  <style>
    p { color: blue; }
  </style>
</head>
```

External CSS: Linking to an external CSS file using the <link> tag, which provides better organization and reusability of styles, e.g.,<head>

```
  <link rel="stylesheet" href="styles.css"></head>
```

CSS Syntax

Selectors, Properties, and Values:

- Selectors: Identifying HTML elements to style using different types of selectors:

- Element Selector: Selects all elements of a specified type, e.g., p { }

- Class Selector: Selects elements with a specific class attribute, e.g., .classname { }

ID Selector: Selects an element with a specific id attribute, e.g., #idname { }

- Attribute Selector: Selects elements with a specific attribute, e.g., [type="text"] { }
- Pseudo-classes: Select elements based on their state, e.g., :hover { }, :first-child { }
- Properties and Values: Defining style rules by specifying properties and their values, e.g., color: blue;, font-size: 16px;.
- Basic Styling (Color, Fonts, Layout):Color: Setting text and background colors using color names, hex codes, RGB, RGBA, HSL, and HSLA values, e.g., color: #0000FF;, background-color: rgb(0, 0, 255);.
- Fonts: Applying font styles including font-family, font-size, font-weight, font-style, and line-height, e.g.,body {

```
  font-family: Arial, sans-serif;
  font-size: 16px;
  font-weight: bold;
  font-style: italic;
  line-height: 1.5;
}
```

Layout: Using CSS for layout purposes, such as:

Width and Height: Setting dimensions for elements, e.g., width: 100px;, height: auto;.

Padding, Margin, and Border: Controlling spacing around and within elements, e.g.,.box {

```
  padding: 10px;   margin: 20px;   border: 1px solid #000;   }
```

MODULE 6:

ADVANCED CSS

Box Model

Margin, Border, Padding, and Content:

Understanding how the CSS box model works and how it affects the layout of elements.

- Content: The actual content of the box, where text and images appear.

- Padding: The space between the content and the border. Padding increases the size of the box but does not affect the distance from other boxes.

- Border: The line that surrounds the padding (if any) and content. Borders can have different styles, widths, and colors.

- Margin: The space outside the border, separating the element from other elements. Margins can collapse, meaning adjoining margins of adjacent elements may combine to form a single margin.

Example:

```
.box {
  width: 200px;
  padding: 10px;
  border: 2px solid black;
  margin: 15px;
}
```

CSS Flexbox and Grid

• Flexbox Layout: A layout model designed for one-dimensional content, providing a more efficient way to lay out, align, and distribute space among items in a container.

- Flex Container: An element with display: flex; that holds flex items.

- Flex Items: The children of the flex container.Main Axis and Cross Axis: The primary axis (horizontal or vertical) along which flex items are arranged and the perpendicular axis.

Example:

```
.container {
  display: flex;
  justify-content: space-between; /
* Distributes space between items */
  align-items: center; /* Aligns
 items along the cross axis */
}
```

• Grid Layout: A powerful layout system for two-dimensional content, allowing for the creation of complex and responsive grid-based layouts.

- Grid Container: An element with display: grid;.
- Grid Items: The children of the grid container.
- Grid Template Columns and Rows: Defines the structure of the grid.

Example:

```
.grid-container {
  display: grid;
  grid-template-columns: repeat(3, 1fr); /* Creates three equal columns */
  grid-gap: 10px; /* Space between
 grid items */
}
```

Responsive Design

• Media Queries: Techniques to apply different styles for different screen sizes and devices.

° Syntax: Using the @media rule to apply CSS rules based on conditions like screen width, height, orientation, and resolution. Example:

```
  @media (max-width: 600px) {
 .container {
    flex-direction: column; /*
 Stack items vertically on small
```

```
screens */
  }
}
```

Mobile-First Design:

Designing and coding the website primarily for mobile devices first, then progressively enhancing the design for larger screens.

- **Principles: Ensuring the site works well on smaller screens and gradually adding more complex layouts and features for larger screens.**

  Example:

```
        .container {display: flex;
  flex-direction: column; /*
 Default mobile layout */
}
@media (min-width: 768px) {
  .container {
    flex-direction: row; /* Change
 to row layout on larger screens */
  }
```

MODULE 7:

INTRODUCTION TO JAVASCRIPT

JavaScript Basics

What is JavaScript ? JavaScript is a programming language that enables interactive web pages. It allows you to create dynamic content, control multimedia, animate images, and handle various user interactions.

Linking JavaScript to HTML

JavaScript can be embedded directly in HTML using the <script> tag or linked as an external file with <script src="path/to/file.js"></script>.

Basic JavaScript Syntax

Variables, Data Types, and Operators:

- Variables: Used to store data. Declared with var, let, or const (e.g., let name = "John";).

- Data Types: Include strings ("Hello"), numbers (42), booleans (true), arrays ([1, 2, 3]), and objects ({key: "value"}).

- Operators: Perform operations on variables and values.

Examples include arithmetic (+, -), comparison (==, !=), and logical (&&, ||).

Functions and Events

Functions: Blocks of code designed to perform particular tasks, defined using the function keyword (e.g., function greet() { console.

log("Hello"); }).

Events: Actions or occurrences that happen in the browser, like clicks or key presses. Events can trigger functions to run.

MODULE 8:

ADVANCED JAVASCRIPT AND DOM MANIPULATION

DOM (Document Object Model)

-Understanding the DOM

° The DOM is a representation of the document structure of a web page. It allows JavaScript to access and manipulate HTML elements and styles dynamically.

° Essentially, it treats the HTML document as a tree structure where each node represents an element, attribute, or text.

-Selecting and Manipulating DOM Elements:

° Selecting Elements: Use methods like getElementById, getElementsByClassName, getElementsByTagName, querySelector, and querySelectorAll to target specific elements in the document.

° Manipulating Elements: Once selected, you can modify elements by changing their content (inner HTML, text Content), styles (style property), attributes (set Attribute), or adding/removing classes (class List).

-Event Handling

-Adding Interactivity:

° JavaScript allows you to make your web pages interactive by responding to user actions like clicks, hovers, or form submissions.

-Common Events and Event Listeners:

° Common Events: Examples include click, mouse over, mouse out, key down, key up, load, submit, etc.

° Event Listeners: Use addEventListener to attach a function (event handler) to a specific event on an element. For example, element.addEventListener('click', function) will execute the specified function when the element is clicked.

MODULE 9:

WEB DEVELOPMENT FRAMEWORKS

Introduction to Frameworks

-Overview of Popular Frameworks

Frameworks simplify web development by providing pre-written code for common tasks. Popular frameworks include:

° Bootstrap: A CSS framework for building responsive, mobile-first websites.

° React: A JavaScript library for building user interfaces, particularly single-page applications.

-Using Bootstrap

Bootstrap offers pre-designed UI components like navigation bars, buttons, modals, forms, and more that can be easily integrated into web projects.

• Basic Bootstrap Components:

° Layout Components: Container, grid system, rows, and columns.

° Content Components: Typography, tables, images, and figures.

° Forms: Form controls, layout options, validation.

° Components: Buttons, alerts, badges, progress bars, and more.

° Utilities: Classes for spacing, colors, display properties, and more.

-Building Responsive Layouts with Bootstrap

Bootstrap's grid system uses rows and columns to create flexible and responsive layouts that adjust automatically to different screen sizes. Utility classes help with spacing, alignment, and visibility.

° Grid System: Bootstrap’s grid system uses a series of containers, rows, and columns to layout and align content. It’s built with flexbox and is fully responsive.

° Responsive Design: Use Bootstrap’s responsive utility classes to adjust the layout based on the screen size (e.g., col-xs-*, col-sm-*, col-md-*, col-lg-*).

° Customizing Bootstrap: Overriding Bootstrap styles with custom CSS or using pre-built themes to match the design requirements.