**FULL STACK DEVELOPER**

**FRONTEND+BACKEND**

**FRONT END BACKEND**

**HTML PYTHON**

**CSS . NET**

**JAVASCRIPT SPRING BOOT**

**BOOTSTRAP JAVA**

**REACT JS NODE JS**

**What are web/mobile/desktop applications?**

**web and desktop applications are easily viewed from any browser and thus no need to be installed where as mobile applications need to be installed from the app store**

**Application has two types:**

**1)Static application**

**2) Dynamic application**

**1)Static application:**

**Static web applications are commonly built using libraries and web frameworks like Angular , React, svelte,vue or blazer where server side rendering is not required.These applications include HTML CSS JAVA SCRIPT. a static app can appear the same for every user. These are not stored in any database or get updated whenever user visit, the data is stored in browser**

**2) Dynamic applications:**

**Dynamic websites generate content automatically based on user.Get functionality and responding to the actions ,it needs a server and database it can be expensive to build EX: Social media sites like Facebook,Twitter and forums.**

**Www/internet:**

**WWW: www stands for world wide web ,it is a collection of information which is accessed via the internet it is a subset of the internet ,www uses http address in the beginning it is known as NSFNET**

**INTERNET:**

**It is a global network of networks internet is means of connecting a computer to any other computer anywhere in the world ,it is superset of www , in earlier it was known as ARPANET ,internet uses the IP address**

**1)What are the prerequisites to become a front-end developer?**

**⦁ Must have a degree btech or bsc or similar field**

**⦁ Be proficient in coding languages like HTML,CSS,JAVASCRIPT and JQUERY**

**2)Roles and responsibilities of a front end developer**

**⦁ Front end developer has a responsibility to create or construct a web page in a beautiful way**

HTML

**HTML Stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content.**

**it has two concepts**

**1) Hypertext: link between multiple pages**

**2) Markup language: Text submit within the tag**

**tags are the one which is used to create a web pages**

**Features of HTML:**

* **It is easy to learn and easy to use.**
* **It is platform-independent.**
* **Images, videos, and audio can be added to a web page.**
* **Hypertext can be added to the text.**
* **It is a markup language.**

**Why learn HTML?**

* **It is a simple markup language. Its implementation is easy.**
* **It is used to create a website.**
* **Helps in developing fundamentals about web programming.**
* **Boost professional career.**

**Advantages:**

* **HTML is used to build websites.**
* **It is supported by all browsers.**
* **It can be integrated with other languages like CSS, JavaScript, etc.**

**Disadvantages:**

* **HTML can only create static web pages. For dynamic web pages, other languages have to be used.**
* **A large amount of code has to be written to create a simple web page.**
* **The security feature is not good.**

**Uses of HTML:**

**⦁ It is used to create a webpage with the help of tags,elements,attributes**

**⦁ Used to structure the webpage and its content**

**⦁ Navigating the internet**

**⦁ Embedding images and videos**

**⦁ Improving client side data storage and offline capabilities**

**⦁ Game development**

## Core concepts of HTML:

**⦁ Tags**

**⦁ Elements**

**⦁ Attributes**

**TAGS:**

**Tags are the keywords that are used to produce webpages in various formats.Tags are the building blocks of our application**

**It can be two things**

**1)Single tag :**

**The tag which doesn't have closing tag.It is self enclosing tag**

**it is called as a single tag/white tag/empty tag**

**<p>my paragraph</p>**

**2)Paired tag:**

**The tag which contains separately closing tag**

**ELEMENTS:**

**An HTML element is defined by a start tag, some content, and an end tag.The HTML element is everything from the start tag to the end tag.**

**1) Inline elements:**

* **If the elements are displayed in a line.An inline element does not start on a new line.An inline element only takes up as much width as necessary.**

**Here are the inline elements in HTML:**

**<a> <abbr> <acronym> <b> <bdo> <big> <br><button> <cite> <code> <dfn> <em> <i> <img><input> <kbd> <label> <map> <object> <output> <q><samp> <script> <select> <small> <span> <strong> <sub><sup> <textarea> <time> <tt> <var>**

**2) Block-level elements:**

* **A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.A block-level element always takes up the full width availableTwo commonly used block elements are: <p> and <div>.The <p> element defines a paragraph in an HTML document.The <div> element defines a division or a section in an HTML document.**

**<div> this is a block level<div>**

**Here are the block-level elements in HTML:**

**<address> <article> <aside> <blockquote> <canvas> <dd><div> <dl> <dt> <fieldset> <figcaption> <figure> <footer> <form> <h1>-<h6> <header> <hr><li> <main> <nav> <noscript> <ol> <p><pre> <section> <table> <tfoot> <ul> <video>**

**3) Inline block elements:**

* **Always whatever clements content is placed inside a web (or) container that content act as a block level element acts as a inline**

**ATTRIBUTES:**

**⦁ All HTML elements can have attributes**

**⦁ Attributes provide additional information about elements**

**⦁ Attributes are always specified in the start tag**

**⦁ Attributes usually come in name/value pairs like: name="value"**

**⦁ Attributes are**

**1) Pre defined:**

* **which have been defined already by using html.world wide can use**

**2) User defined:**

* **define a own attribute,element specific attribute**

**How can we create a sample web page?**

**<!doctype html>---->it is not a tag instructor tell the browser which type of html u are using**

**<html> define root elements of our html document**

**<head> Defines the meta data of the html**

**<title></title> it defines the title of our html page ,always it displays on browser task bar**

**<body></body> it defines the website /webpage design , all the content we will write in a body**

**</head>**

**</html>**

## **TEXT RELATED TAGS**

**1)Heading tag**

**2)Article tag**

**3)Aside tag**

**4)List tag**

**5)Table tag**

**6)Paragraph**

**1) Heading tag:**

**⦁ Html headings are titles or subtitles that you want to display on the web page html are defined with the <h1>to <h6> tags**

**⦁ <h1> defines the most importent heading <h6> defines the least importance hading**

**⦁ Each and every page contain only one h1tag h1 tag canot be repeated multiple times**

**⦁ h2,h3 we can call multiple times,all the headings are block level**

**2) Article tag:**

**⦁ If we want make a lengthy point of view,it can take the high priority to display the content**

**⦁ 1----------150 characters we use artical**

**⦁ 1-----------250 characters we use Aside tag**

**⦁ 1--------------250 above we use paragraph**

**3)Aside tag:**

**The <aside> tag defines some content aside from the content it is placed in. The aside content should be indirectly related to the surrounding content. Tip: The <aside> content is often placed as a sidebar in a document.**

**4) List tag:**

**We want to list the every paragraph into lists we can use list tag**

**lists are majorly 4 types**

**i)order list:**

**An ordered list starts with the <ol> tag. Each list item starts with the <li> tag.**

**the list items will be marked with numbers by default:**

**Example**

**<ol>**

**<li>Coffee</li>**

**<li>Tea</li>**

**<li>Milk</li>**

**</ol>**

**ii)unordered list:**

**An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.**

**The list items will be marked with bullets (small black circles) by default:**

**Example**

**<ul>**

**<li>Coffee</li>**

**<li>Tea</li>**

**<li>Milk</li>**

**</ul>**

**iii)Definition list: -----it is not recommended**

**iv)Description list:**

**HTML also supports description lists.**

**A description list is a list of terms, with a description of each term.**

**The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:**

**Example:**

**<dl>**

**<dt>Coffee</dt>**

**<dd>- black hot drink</dd>**

**<dt>Milk</dt>**

**<dd>- white cold drink</dd>**

**</dl>**

**5)Table tag:**

**⦁ The <table> tag defines an HTML table.**

**⦁ An HTML table consists of one <table> element and one or more <tr>, <th>, and <td> elements.**

**⦁ The <tr> element defines a table row, the <th> element defines a table header, and the <td> element defines a table cell.**

**⦁ An HTML table may also include <caption>, <colgroup>, <thead>, <tfoot>, and <tbody> elements.**

**Creation of a table:**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title></title>**

**</head>**

**<body>**

**<table border="2" cellspacing="0" cellpadding="10" align="center">**

**<tr>**

**<th>S NO</th>**

**<th>NAME</th>**

**<th>PH NUMBER</th>**

**<th>EMAIL</th>**

**<th>COURSE</th>**

**<th>SKILLS</th>**

**</tr>**

**<tr>**

**<td>01</td>**

**<td>Ramadevi</td>**

**<td>8106192412</td>**

**<td>munichandrarama@gmail.com</td>**

**<td>Btech</td>**

**<td>Html</td>**

**</tr>**

**<tr>**

**<td>01</td>**

**<td>Ramadevi</td>**

**<td>8106192412</td>**

**<td>munichandrarama@gmail.com</td>**

**<td>Btech</td>**

**<td>Html</td>**

**</tr>**

**</table>**

**</body>**

**</html>**

**Cell Spacing :**

**space between cells that is white space between the edges of the adjacent cells**

**Cellpadding:**

**Space between the border of a table cell and its content that means white space between the cell edge and the content of the cell**

IMAGES

**⦁ Web Page =text content+Graphics+multimedia+forms+layout**

**⦁ main motto=We should present the data /content inside a webpage in a beautiful way**

**⦁ image tag: this don't have closing tag only d\self enclosing tag**

**⦁ Every single tag must be self enclosed**

**⦁ <img src="file path" alt=""/>**

**⦁ Src= it a mandatory attribute adding extra information to an element**

**⦁ value = attribute it holds the alternative value of the image**

**⦁ File path can be two types:**

**1) Absolute path:**

* **Relative file path are paths that links to a local file in the same folder or on the same server**

**2)Relative path:**

* **An absolute path is usually used to specify the path to a file that is located on another network resource. It is a complete URL to a file or page**

**Image extensions:**

**.jpeg .jpg------> Low resolution .png(Portable network graphic)---------->High resolution**

**svg--->(scalable Vector graphic) more powerful**

**Web P(recommended .jfif**

**Example of imges:**

**<body>**

**<img src="https://www.achieversit.com/assets/images/logo-white.png" alt="logo"/>**

**<img src="C:\Users\91810\OneDrive\Desktop\New folder\marriage.jpeg" alt="marriage photo" width="100"/>**

**</body>**

## 

## FORMS

**⦁ An HTML form is used to collect user input.**

**⦁ Why should we collect the data from a user ? for the future use**

**⦁ How to collect the data from the user ? using forms and form controls**

**⦁ Form is a tag it has a starting and ending it is paired tag**

**⦁ Rules:**

**1)Every form control must be placed in a form tag**

**⦁ 2) create labels and controls**

**⦁ Why do we have to create a label?**

**⦁ If anything without a label end users are not going to understand**

**⦁ In advanced html without labels also we can educate the users by using "placeholder".**

**⦁ Every form control must have a name**

**⦁ TYPE: type is an attribute which defines which type of control we are creating. the value of a type attribute is a predefined**

**⦁ Name is user defined**

**⦁ Input is a single tag must be self enclosed**

**⦁ Relevant values we have to give**

**⦁ We have to submit form must---submit is control**

**⦁ Every control has its own importance .Must have to know for which element that is supposed to give.**

**⦁ Whenever a user inputting the details, there is no need to give a value attribute.**

**⦁ LABELS: A <label> is used to create a caption for a form control.Whenever the label you would like to map to the respective input element make sure that label"for" attribute value and input "ID" must be the same.**

**⦁ Query parameters:Defined set of parameters attached to the end of url,when information submitting through the url**

**Form Attributes:**

**⦁ METHODS : (Get and post)**

**⦁ GET: When the information submitting through the url we can use GET method ,It is not secured**

**⦁ POST: if the information is submitted through the http method we use POST,it is secured . If your form contains any sensitive data that means password,credit card cvv numbers, secret codes etc...**

**⦁ Action attribute: Which going to define the action when the form get submitted to whom have to submit backend url**

**⦁ ENCTYPE : (encryption type ) enctype="multipart/form-data"**

**⦁ It is required when form contain any binary data from the user like docs,jpeg**

**⦁ Auto complete: by default it is ON,It display all historical values in the form**

**⦁ NOVALIDATE : html5 validations are not fully handled."required": mandatory fields we have fill , to bypass it to the javascript we use novalidate**

**<body>**

**<form method="GET" action="#">**

**<label for="user">NAME</label>**

**<input type="text" name="username" id="user" required/>**

**<label for="email">email</label>**

**<input type="email" name="email" id="email" required/>**

**<label for="num">phone number</label>**

**<input type="text" name="phone number" id="num" required />**

**<input type="submit" name="register" />**

**</form>**

**</body>**

**⦁ Reset : A form is already created user is going to enter the values at certain point of time he realised inside a form control is wrong using reset we can delete all the data**

**⦁ Submit : The submit button allows the user to send form data to web server or backend of the form**

**⦁ Button: The <button> tag defines a clickable button.**

**⦁ Input type button is the choice.It will not submit the data.**

**⦁ Button ---------- The button is clickable button**

**⦁ Submit------------- Submit the data to the backend**

**⦁ reset---------------The button is reset the form data its initial values**

**⦁ Dropdown: Whenever we would like to display multiple options the user is going to select any one option we use dropdown .**

**⦁ Radio buttons: Definition and Usage. The <input type="radio"> defines a radio button. Radio buttons are normally presented in radio groups (a collection of radio buttons describing a set of related options). Only one radio button in a group can be selected at the same time.**

**<label for="gender">Gender : </label>**

**<label for="male">Male</label>**

**<input type="radio" name=" gender" id="male"/>**

**<label for="female">Female</label>**

**<input type="radio" name=" gender" id="female"/>**

**⦁ Checkbox: checkbox are used to let the select one (or) more options from a predefined set of options.Check box input control are created using the "input" element with a type attribute having a values as "checkbox".<input type="checkbox">**

**⦁ < label for="language">Languages : </label>**

**<label for="telugu" >Telugu</label>**

**<input type="checkbox" name="Telugu" id="language" required/>**

**<label for="english">English </label>**

**<input type="checkbox" name="English" id="language" required/>**

**<label for="hindi">Hindi </label>**

**<input type="checkbox" name="hindi" id="hindi" required/>**

**⦁ File select: It is used to allow the user to select a local file and send it as an attachment to the web server.**

**<label for="file">Photo : </label>**

**<input type="file" name="upload" id="file" required/>**

**⦁ Text area: Text area is a multiple line text input control that allows the user to provide a description or text in multiple lines . text area input control is created using the "textarea" element**

**<label for="dec">Address : </label><br>**

**<textarea rows="5" cols="50" name="address" id="dec"></textarea>**

**⦁ Select box: Select boxes are used to allow users to select one or more options from a pulldown list of optionsSelect boxes are created using two elements which are "select" and "option".list items are defined within the select element.**

**<label for="cr">Course Name : </label>**

**<select name="course" id="cr">**

**<option value="frontend" > frontend</option>**

**<option value="Backend" > Backend </option>**

**<option value="full stack" > full stack</option>**

**</select>**

**Differences between checkbox and radio buttons?**

**===>Checkbox and radio buttons are elements for making a selection. Check box allows the user to choose items from a fixed number of alternatives ,while radio buttons allow the user to select only one option from a list of several predefined alternatives.**

**Differences between Textbox and Textarea ?**

**====>The difference between the two is that input textbox will allow you to add one line of text, While the text area will allow you to add multiple lines of text**

**Differences between button and submit ?**

**=====>If you want to submit the form to your server you should use "submit",however if you want to do something from the client side by clicking the "button" . Input type button is the choice.It will not submit the form.**

## Features of HTML5

**⦁ Form elements**

**⦁ Form Attributes**

**⦁ Validations**

**⦁ Layouts**

**⦁ Web storage**

**⦁ Geo Location**

**⦁ Drag and Drop**

**⦁ SVG**

**⦁ CANVAS**

**⦁ Form elements are form controls**

**⦁ Without attribute we can't send the form to backend**

**⦁ Form validations restrict the data to mandatory by using the Required,Novalidate,method.**

**Semantic tags:**

**A semantic elements clearly describe its meaning to both the browser and the developer**

**EX: <header>,<footer>, <article>**

**Non Semantic tags:**

**Unlike semantic elements,they don't have any meaning. They don't tell anything about the content that contain.They can be used with different attributes to mark up semantic common to group**

**EX: <span>,<div>**

**Uses of semantic tags:**

**By adding semantic tags to your document,you provide additional information to the document.Semantic tags make it clear to the browser what the meaning of page and its content is**

**Why we Use:**

**Make your code easier to understand making the source code more readable for other developers**

**Versions of HTML:**

**HTML---1.0**

**HTML---2.0**

**HTML---3.0**

**HTML---4.0**

**DHTML--Dynamic HTML**

**XHTML---Extensible HTML---->Introduced rules and regulations**

**HTML---5.0**

**Now we are using HTML5 Version**

## LAYOUTS

**It is blueprint used to arrange web page in a well defined manner.Easy to understand and uses HTML tags to customize web design elements**

**HTML layout tags:**

**A group of tags that define document layout include Header,Footer,Main,Nav,Article**

**Multimedia tags in HTML5:**

**⦁ Multimedia comes in many different formats. It can be almost anything you can hear or see, like images, music, sound, videos, records, films, animations, and more.**

**⦁ Web pages often contain multimedia elements of different types and formats.**

**Multimedia Formats**

**⦁ Multimedia elements (like audio or video) are stored in media files.The most common way to discover the type of a file, is to look at the file extension.Multimedia files have formats and different extensions like: .wav, .mp3, .mp4, .mpg, .wmv, and .avi.**

**⦁ There are many video formats out there.**

**⦁ The MP4, WebM, and Ogg formats are supported by HTML.**

**⦁ The MP4 format is recommended by YouTube.**

**HTML Audio:**

**⦁ The HTML <audio> element is used to play an audio file on a web page.**

**⦁ The controls attribute adds audio controls, like play, pause, and volume.**

**⦁ The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.**

**⦁ The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.**

**Example:**

**<audio controls muted autoplay loop>**

**<source src="C:\Users\91810\OneDrive\Desktop\html1\forms.html\harsha song.mpeg">**

**</audio>**

**HTML VIDEO:**

**⦁ The HTML <video> element is used to show a video on a web page.**

**⦁ The controls attribute adds video controls, like play, pause, and volume.**

**⦁ It is a good idea to always include width and height attributes. If height and width are not set, the page might flicker while the video loads.**

**⦁ The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.**

**⦁ The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.**

**Example:**

**<video controls width="200" muted autoplay loop poster="C:\Users\91810\OneDrive\Desktop\html1\forms.html\harsha.jpeg">**

**<source src="C:\Users\91810\OneDrive\Desktop\html1\forms.html\birthday vedio.mp4">**

**</video>**

**⦁ Muted---->to mute song or video**

**⦁ Autoplay loop------>To start a video automatically, use the autoplay attribute:**

**⦁ Poster -------> It means showing the image is different and the video is different.**

**⦁ Canvas and SVG used for 2D,3D animations (or) to draw the Graphics**