Course Outline

- Mockup Design With Figma or Adobe XD
- **Creating layout with HTML**
- **Designing layout with CSS**
- **Making interactive elements with JS**
- Using Bootstrap for creating fast and reliable templates
- Using Jquery to handle events
- Reactjs - Basics

Mockup Designing (Figma)

Psychology of Color in Web

Color is powerful. The psychology of color can help strengthen your brand, encourage sales, and even guide visitors toward specific pages or actions on your website. Studies show that people decide whether or not they like a product in 90 seconds or less and that 90% of that decision is based solely on color.

Typography

Typography is the use of type in a design. Typography seeks to create a greater meaning by thoughtful and deliberate selection of font, size, color, layout, alignment, and other factors that affect the design of type on a page. Sans serif

fonts are without serifs; no extra details are found on the end of each letter. Whereas Serif fonts have serifs:extra details are found on the end of each letter.

Using tools for Mockup Design

Mockplus, **Adobe XD**, Balsamiq, Moqups, MockplusiDoc, Fluid UI, Mockingbird,**figma**

Vector Design

Vector graphics use geometrical shapes and lines to represent images in computer graphics. Vector illustration is a popular technique of many digital illustrators worldwide. With their detailed, bright and crisp style, vector graphics have the power to give a modern, edgy and trendy look to any web design.

Designing various element for Web/Mobile

Here are the five elements I feel are important.

- Design and Information Architecture. Design and information architecture is quite different for a mobile site.
- Content. How content is displayed is another element you need to be aware of.
- Functionality.
- Call to Actions.
- Usability and Experience.
- Team/FAQ/

Mastering elements using different options

- Keep your website's topography consistent

- Don’t go overboard with hectic color schemes
- Put more thought into the images you utilize
- Learn more about SEO
- Keep inclusion and diversity in mind
- Always align your logo to the left

Using some important extension for Web/Mobile design

- DomFlags
- Fakefiller
- lightshot
- Sizzy
- Checkbot
- GistBox Clipper
- ColorZilla
- WhatFont
- LightShot
- Awesome Screenshot
- Clear Cache
- Web Developer Google Chrome Extension

Project Work (Designing Mockup)

Q. Design a landing page for any agency/community/portfolio. (5 Points)

Hint :

1) Header - logo on left side / Menu Links (Text) rightside
2) Image Slider - images
3) About Us -> Text and Images
4) Services -> Box (3) => icon/title/short info
5) Recent Blog Posts = 2 column layout with Image/title/date& time/short info / button(Read More)
6) Recent works/portfolio -> image (4 column)
7) Contact form
8) Footer {Logo, Subscription Form, Links, quick info about agency/community}

HTML/HTML5

HTML Elements

Html elements are basic elements to create webpage. HTML elements are listed below:

1. h1 => Heading
2. p => Paragraph
3. ul => Unordered List
4. ol=> Ordered List
5. img=> Image
6. a=> Anchor Tag(Link)
7. div => Division Block

8. meta => important information about particular page.
9. body => Main area of website
10. html => Document type.
11. link => Link is required to call any Resource for that page.

HTML 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"
- For Example Hari Bhusal Website, <img src="img_girl.jpg" height="100" width="100">

HTML Headings

- <h1>Heading 1</h1>
- *<h2>Heading 2</h2>*
- <h3>Heading 3</h3>
- <h4>Headi

- ng 4</h4>
- **<h5>Heading 5</h5>**
- <h6>Heading 6</h6>

HTML Text Formatting

- <b> - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

HTML Comments

To write HTML comments put <!--- and ---> at either end of the comment. HTML comments are notes to keep HTML code organized and are ignored by the browser.

<!--- This is a comment --->

<!---

This is a multi-line comment

that appears on multiple lines

because it is long.

--->

HTML Lists

Ordered HTML

Eg.

```
<ol>
 <li>Coffee</li>
 <li>Tea</li>
 <li>Milk</li>
</ol>
```

Unordered HTML

Eg.

```
<ul>
 <li>Coffee</li>
 <li>Tea</li>
 <li>Milk</li>
</ul>
```

Definition Lists

Eg.

```
<dl>
 <dt>Coffee</dt>
```

```
 <dd>Black hot drink</dd>
 <dt>Milk</dt>
 <dd>White cold drink</dd>
</dl>
```

HTML tables

```
<table>
 <tr>
  <th>Company</th>
  <th>Contact</th>
  <th>Country</th>
 </tr>
 <tr>
  <td>student.name</td>
  <td>student.addres</td>
  <td>student.country</td>
 </tr>
 <tr>
  <td>Centro comercial Moctezuma</td>
  <td>Francisco Chang</td>
  <td>Mexico</td>
 </tr>
```

```
</table>
```

HTML Forms

https://www.w3schools.com/html/html_form_input_types.asp

```
<form>
 <label for="fname">First name:</label><br>
 <input type="text" id="fname" name="fname"><br>
 <label for="lname">Last name:</label><br>
 <input type="text" id="lname" name="lname">
<input type="submit" value="Submit">
</form>
```

HTML Images

```
<img src="img_girl.jpg" alt="Girl in a jacket">
```

HTML Anchor

The HTML anchor tag defines a hyperlink that links one page to another page. It can create hyperlink to other web page as well as files, location, or any URL. The "href" attribute is the most important attribute of the HTML a tag. and which links to destination page or URL.

- Text Links - Go to Google

- Email Links - Send Feedback
- Image Links - <img src="smiley.gif" alt="HTML tutorial" >
- Phone Links - CLICK TO CALL

Block Elements

<p>, <h1>, <h2>, <h3>, <h4>, <h5>, <h6>, <ul>, <ol>, <dl>, <pre>, <hr />, <blockquote>, and <address>

Inline Elements

<b>, <i>, <u>, , , <sup>, <sub>, <big>, <small>, <li>, <ins>, <del>, <code>, <r>, <dfn>, <kbd>, and <var>

HTML Entities

[space]	non-breaking space		
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
¢	cent	¢	¢
£	pound	£	£

¥	yen	¥	¥
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

HTML Symbols

Char	Number	Comment
©	©	copyright icon
®	®	registered trade mark sign
™	™	trade mark sign
@	@	at symbol

HTML5 Audio & Video

Method	Description
addTextTrack()	Adds a new text track to the audio/video
canPlayType()	Checks if the browser can play the specified audio/video type
load()	Re-loads the audio/video element
play()	Starts playing the audio/video
pause()	Pauses the currently playing audio/video

Example

```
<!DOCTYPE html>
<html>
<body>
<button onclick="playVid()" type="button">Play Video</button>
<button onclick="pauseVid()" type="button">Pause Video</button><br>
<video id="myVideo" width="320" height="176">
  <source src="mov_bbb.mp4" type="video/mp4">
  <source src="mov_bbb.ogg" type="video/ogg">
  Your browser does not support HTML5 video.
</video>
<script>
var vid = document.getElementById("myVideo");
function playVid() {
  vid.play();
}
function pauseVid() {
  vid.pause();
}
</script>
<p>Video courtesy of <a href="https://www.bigbuckbunny.org/"
target="_blank">Big Buck Bunny</a>.</p>
</body>
</html>
```

HTML5 Web Storage

With web storage, web applications can store data locally within the user's browser.

Before HTML5, application data had to be stored in cookies, included in every server request. Web storage is more secure, and large amounts of data can be stored locally, without affecting website performance.

Unlike cookies, the storage limit is far larger (at least 5MB) and information is never transferred to the server.

Web storage is per origin (per domain and protocol). All pages, from one origin, can store and access the same data

HTML5 web storage provides two objects for storing data on the client:

window.localStorage - stores data with no expiration date

window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)

Eg:

```
// Store

localStorage.setItem("lastname", "Smith");

// Retrieve

document.getElementById("result").innerHTML =
localStorage.getItem("lastname");
```

HTML5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.

The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

EG:

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000000;">
</canvas>
```

HTML5 SVG

```
<svg width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />
</svg>
```

HTML5 Drag & drop

```
<!DOCTYPE HTML>
<html>
<head>
<script>
function allowDrop(ev) {
```

```
  ev.preventDefault();
}

function drag(ev) {
  ev.dataTransfer.setData("text", ev.target.id);
}

function drop(ev) {
  ev.preventDefault();
  var data = ev.dataTransfer.getData("text");
  ev.target.appendChild(document.getElementById(data));
}
</script>
</head>
<body>

<div id="div1" ondrop="drop(event)"
ondragover="allowDrop(event)"></div>

<img id="drag1" src="img_logo.gif" draggable="true"
ondragstart="drag(event)" width="336" height="69">

</body>
</html>
```

HTML5 Geolocation

Locate the User's Position

The HTML Geolocation API is used to get the geographical position of a user.

Since this can compromise privacy, the position is not available unless the user approves it.

sasa

```
<script>
var x = document.getElementById("demo");
function getLocation() {
  if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(showPosition);
  } else {
      x.innerHTML = "Geolocation is not supported by this browser.";
  }
}

function showPosition(position) {
  x.innerHTML = "Latitude: " + position.coords.latitude +
  "<br>Longitude: " + position.coords.longitude;
}
</script>
```

Example explained:

```
Check if Geolocation is supported

If supported, run the getCurrentPosition() method. If not, display a
message to the user

If the getCurrentPosition() method is successful, it returns a
coordinates object to the function specified in the parameter
(showPosition)

The showPosition() function outputs the Latitude and Longitude
```

HTML5 Semantics

Many web sites contain HTML code like: , , to indicate navigation, header, and footer.

A semantic element clearly describes its meaning to both the browser and the developer.

Examples of non-semantic elements: and - Tells nothing about its content.

Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.

In HTML there are some semantic elements that can be used to define different parts of a web page:

<**article**> : Forum posts, Blog posts, User comments, Product cards, Newspaper articles

<aside>: The <aside> content should be indirectly related to the surrounding content.

<details>

<**figcaption**>

<**figure**> : We can place image inside Figure Tag

<**footer**>: authorship information, copyright information, contact information, sitemap, back to top links, related documents

<**header**>: one or more heading elements (<h1> - <h6>), logo or icon, authorship information, menu links

<**main**>

<**mark**>

<**nav**> : menu links

<**section**> : Chapters, Introduction, News items, Contact information

<**summary**>

<**time**>

HTML Meta Tags

Important for SEO purpose

```
<head>
 <meta charset="UTF-8">
 <meta name="description" content="Free Web tutorials">
 <meta name="keywords" content="html,css,javascript,php">
 <meta name="author" content="John Doe">
 <meta name="revised" content="05/12/2013" />
 <meta http-equiv="refresh" content="5;
 url=https://example.com/">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

- **Keywords**

  The keywords are generally written in lower case, and separated with a comma.

- **Description**

  A meta description is the small blurb that appears underneath your website on the **SERP** that includes information about your page. It is designed to

provide users with a brief summary of the content on your page so that users know if the page will answer their question.

- **Author**

  Specifies the name of the author of the document

- **Revised**

  Meta revised tag is used to tell Search Engine bots (Google, Yahoo!, Bing etc.) the Last Modified Date and Time of any Webpage in which you have made certain changes. Web crawlers may decide to update their index by reading this tag info on a web page.

- **Charset**

  It contains the value i.e character_set which specify the character encoding for the HTML document. Values: UTF-8: It specify the character encoding for Unicode. ISO-8859-1: It specify the character encoding for the Latin alphabet.

- **Refresh**

  Meta refresh is a method of instructing a web browser to automatically refresh the current web page or frame after a given time interval, using an HTML meta element with the http-equiv parameter set to "refresh" and a content parameter giving the time interval in seconds

- **Robots**

  Meta robots tag is a tag that tells search engines what to follow and what not to follow. It is a piece of code in the <head> section of your webpage. It's a simple code that gives you the power to decide about what pages you want

to hide from search engine crawlers and what pages you want them to index and look at.

- FOLLOW – a command for the search engine crawler to follow the links in that webpage
- INDEX – a command for the search engine crawler to index that webpage
- NOFOLLOW – a command for the search engine crawler NOT to follow the links in that webpage
- NOINDEX – a command for the search engine crawler NOT to index that webpage

HTML iframes

iFrame is short for Inline Frame and is a powerful element in web design. You have probably seen countless YouTube videos embedded on sites other than YouTube. An iFrame can insert all sorts of media. So, you may be wondering how it's done. Chances are the web designer put an iFrame element within that page.

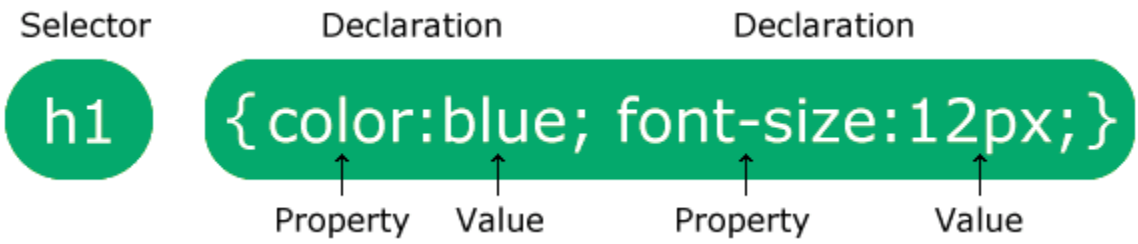
Eg: Youtube Video, Google Map , facebook page embedd

MODULE 2 - CSS

CSS Introduction

CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

CSS Syntax



H1,h2,h3,p,ol,li, table{

} => Element Selector

.class{

} => Class Selector

#idselector{

}=> id Selector

[role=”button”]{

}=> Attribute Selector

The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

Example:

.classSelector{

background: red;

}

<h1 class=”classSelector”>Some Content</h1>

Inline CSS

An inline style may be used to apply a unique style for a single element.

To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style. The internal style is defined inside the <style> element, inside the head section.

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}
h1 {
  color: maroon;
  margin-left: 40px;
}
</style>

</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!

Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

CSS Selectors

Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
.class1.class2{}	.name1.name2	Selects all elements with both *name1* and *name2* set within its class attribute

.class1 .class2	.name1 .name2	Selects all elements with *name2* that is a descendant of an element with *name1*
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element.class	p.intro	Selects all <p> elements with class="intro"
element,element	div, p	Selects all elements and all <p> elements
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element

element+element	div + p	Selects the first <p> element that is placed immediately after elements
element1~element2	p ~ ul	Selects every element that is preceded by a <p> element
[*attribute*]	[target]	Selects all elements with a target attribute
[*attribute=value*]	[target=_blank]	Selects all elements with target="_blank"
[*attribute~=value*]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[*attribute|=value*]	[lang|=en]	Selects all elements with a lang attribute value equal to "en" or starting with "en-"
[*attribute^=value*]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"

[*attribute$=value*]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[*attribute*=value*]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"
		date:8/24
:active	a:active	Selects the active link
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
:checked	input:checked	Selects every checked <input> element
:default	input:default	Selects the default <input> element

:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children (including text nodes)
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> element that is the first child of its parent
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the input element which has focus

:fullscreen	:fullscreen	Selects the element that is in full-screen mode
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects input elements with a value within a specified range
:indeterminate	input:indeterminate	Selects input elements that are in an indeterminate state
:invalid	input:invalid	Selects all input elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)
:last-child	p:last-child	Selects every <p> element that is the last child of its parent

:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
::marker	::marker	Selects the markers of list items
:not(*selector*)	:not(p)	Selects every element that is not a <p> element
:nth-child(*n*)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(*n*)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(*n*)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child

:nth-of-type(*n*)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects input elements with no "required" attribute
:out-of-range	input:out-of-range	Selects input elements with a value outside a specified range
::placeholder	input::placeholder	Selects input elements with the "placeholder" attribute specified
:read-only	input:read-only	Selects input elements with the "readonly" attribute specified

:read-write	input:read-write	Selects input elements with the "readonly" attribute NOT specified
:required	input:required	Selects input elements with the "required" attribute specified
:root	:root	Selects the document's root element
::selection	::selection	Selects the portion of an element that is selected by a user
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all input elements with a valid value
:visited	a:visited	Selects all visited links

CSS Comments

```css
/* This is a single-line comment */
p {
  color: red;
}

/*
This is
a multi-line
comment
*/

p {
 color: red;
}
```

CSS Background

- `background-color`
- `background-image`
- `background-repeat`
- `background-attachment`
- `background-position`
- `background` **(shorthand property)**

CSS Border

Border Style

The following values are allowed:

- `dotted` - Defines a dotted border
- `dashed` - Defines a dashed border
- `solid` - Defines a solid border
- `double` - Defines a double border
- `groove` - Defines a 3D grooved border. The effect depends on the border-color value
- `ridge` - Defines a 3D ridged border. The effect depends on the border-color value
- `inset` - Defines a 3D inset border. The effect depends on the border-color value
- `outset` - Defines a 3D outset border. The effect depends on the border-color value
- `none` - Defines no border
- `hidden` - Defines a hidden border

The `border-style` property can have from one to four values (for the top border, right border, bottom border, and the left border).

CSS Outline

- `outline-style`
- `outline-color`
- `outline-width`
- `outline-offset`
- `outline`

CSS Float

The float property is used for positioning and formatting content e.g. let an image float left to the text in a container. The float property can have one of the following values:

left - The element floats to the left of its container

right - The element floats to the right of its container

none - The element does not float (will be displayed just where it occurs in the text). This is default

inherit - The element inherits the float value of its parent

In its simplest use, the float property can be used to wrap text around images

```
.clearfix::after {
content: "";
```

```
clear: both;

display: table;

}
```

CSS Clear

The clear property controls the flow next to floated elements. The clear property specifies what should happen with the element that is next to a floating element.

Value	Description
none	Default. The element is not pushed below left or right floated elements
left	The element is pushed below left floated elements
right	The element is pushed below right floated elements
both	The element is pushed below both left and right floated elements
initial	Sets this property to its default value. Read about initial
inherit	Inherits this property from its parent element. Read about inherit

CSS Display

- **Grid**
- **Flex**
- **None**

- **Block**
- **Inline-block**
- **Table**
- **Table-cell**
- **Table-column**

CSS Font

CSS Line Height

CSS Margin

CSS Padding

CSS Overflow

CSS Opacity

CSS Position

CSS Pseudo Class & Element

CSS Vertical Align

CSS White Space

CSS Word Wrap

CSS Visibility

CSS Counter

CSS3

CSS3 Rounded Corner

CSS3 Border Images

CSS3 Multi Background

CSS3 Color

CSS3 Gradients

CSS3 Shadow

CSS3 Text

CSS3 Web font

CSS3 2d transform

CSS3 3d transform

CSS3 Animation

CSS3 Multi columns

CSS3 User Interface

CSS3 Box Sizing * => Jquery height/width bata explain garne

CSS Variables

CSS3 Image Reflection

CSS3 object-fit

CSS3 Viewport

CSS3 Media Queries Responsive

CSS Flexbox

flex-direction

flex-wrap

flex-flow

justify-content

align-items

align-content

order

flex-grow

flex-shrink

flex-basis

flex

Align-self

CSS Grid Module

grid-column-gap

grid-row-gap

grid-gap

grid-template-columns

grid-template-rows

grid-template-areas

grid-column-start

grid-column-end

grid-row-start

grid-row-end

justify-items

align-items

Align-content

Browser Support

Supported by Internet Explorer with prefix -ms-

Supported by Firefox with prefix -moz-

Supported by Chrome with prefix -webkit-

-o-box-shadow: 0px 0px 10px rgba(0,0,0,0.4)

Supported by Safari with prefix -webkit-

Supported by Opera with prefix -webkit-

Once the above topics are covered, the students will make a project on:

E-commerce Website

Sass / SCSS

VS CODE EXTENSION - Live Sass Compiler

- **Preprocessing**
- **Variables**
- **Nesting**
- **Partials**
- **Import**
- **Mixins**
- **Inheritance**
- **Operators**

Once the above topics are covered, the students will make a project on:

News Portal Website

JavaScript

Data Types

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures

Let, Var and Const

https://www.freecodecamp.org/news/var-let-and-const-whats-the-difference/#:~:text=var%20variables%20can%20be%20updated,const%20variables%20are%20not%20initialized.

Operators

Conditionals, loop

Functions

Array and Objects

Document Object Model

jQuery

jQuery Syntax

jQuery Selectors

jQuery Events

jQuery Effects

jQuery Hide/Show

jQuery Fade

jQuery Slide

jQuery Add

jQuery Remove

jQuery CSS Classes

jQuery Animate

jQuery stop ()

jQuery Callback

jQuery Chaining

Bootstrap 4

Content

Reboot

Typography

Code

Images

Tables

Figures

Components

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Collapse

Dropdowns

Forms

Input group

Jumbotron

List group

Modal

Navs

Navbar

Pagination

Popovers

Progress

Scrollspy

Tooltips

Utilities

Utilities

Utilities

Borders

Clearfix

Close icon

Colors

Display

Embed

Flex

Float

Image replacement

Position

Screenreaders

Shadows

Sizing

Spacing

Text

Vertical align

Visibility

Once the above topics are covered, the students will make a project on:

Travel & Tours Website

Bootstrap 5

- **Breakpoints**
- **Containers**
- **Grid**
- **Columns**
- **Gutters**
- **Utilities**
- **Z-index**
- **Reboot**
- **Typography**
- **Images**
- **Tables**

Figures

Overview

Form control

Select

Checks & radios

Range

Input group

Floating labels

Layout

Validation

Accordion

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Close button

Collapse

Dropdowns

List group

Modal

Navs & tabs

Navbar

Offcanvas

Pagination

Popovers

Progress

Scrollspy

Spinners

Toasts

Tooltips

Clearfix

Colored links

Ratio

Position

Visually hidden

Stretched link

Text truncation

API

Background

Borders

Colors

Display

Flex

Float

Interactions

Overflow

Position

Shadows

Sizing

Spacing

Text

Vertical align

Visibility

Approach

Icons

Once the above topics are covered, the students will make a project on:

Corporate Website

React

React Get Started

React ES6

React Render HTML

React JSX

React Components

React Props

React State

React Lifecycle

React Events

React router

React Lifting State Up

React Forms

React CSS

React Hook- useState, useEffect

Once the above topics are covered, the students will make a project on:

News Portal Website

Domain Registration / Web Hosting

Domain Registration

Web Hosting

cPanel/webmail

File upload using FileZilla etc.

GitHub

Tips for job interview

As per the Institute Rules, four of the following projects will be done by the Student themselves! For Final Project

NGO Website (1 projects)

Travel & Tours Website (1 project)

E-commerce Website (1 project)

Corporate Website (1 projects)

News Portal Website (1 project)