



B.Tech Computer Science and Engineering

UI and UX Specialist | Lab



Copyright & Disclaimer

B. TECH CSE with Specialization in Frontend Development

Version 1.0.0

**Copyright and Trademark Information
for Partners/Stakeholders.**

Copyright © 2019 Xebia. All rights reserved.

Please note that the information contained in this classroom material is subject to change without notice. Furthermore, this material contains proprietary information that is protected by copyright. No part of this material may be photocopied, reproduced, or translated to another language without the prior consent of Xebia or ODW Inc. Any such complaints can be raised at sales@odw.rocks

The language used in this course is US English. Our sources of reference for grammar, syntax, and mechanics are from The Chicago Manual of Style, The American Heritage Dictionary, and the Microsoft Manual of Style for Technical Publications.

Acknowledgements

We would like to sincerely thank the experts who have contributed to and shaped B. TECH CSE with Specialization

SME

Hemachandra

is Agile Coach with 13+ Years' Experience in Project Management / Client Relationship Management / Scrum Master/Quality Engineering.

Course Reviewers.

Vipul Jain | Xebia

Divyanshu Sinha | Xebia



Xebia Group consists of seven specialized, interlinked companies: Xebia, Xebia Academy, XebiaLabs, StackState, GoDataDriven, Xpirit and Binx.io. With offices in Amsterdam and Hilversum (Netherlands), Paris, Delhi, Bangalore and Boston, we employ over 700 people worldwide. Our solutions address digital strategy; agile transformations; DevOps and continuous delivery; big data and data science; cloud infrastructures; agile software development; quality and test automation; and agile software security.



ODW is dedicated to provide innovative and creative solutions that contribute in growth of emerging technologies. As a learning experience provider, ODW strengths include providing unique, up to date content by combining industry best practices with leading edge technology. ODW delivers high quality solutions and services which focus on digital learning transformation.

Table of Contents

HTML and CSS Labs

- Lab 1 : Understanding of HTML
- Lab 2: Creating and Saving an HTML Document
- Lab 3: Creating a Sample HTML web page
- Lab 4: Creating a Sample HTML web page
- Lab 5: Formatting text in HTML
- Lab 6: Adding images to HTML Web pages
- Lab 7: Create HTML page with Div tag, div inside div tag
- Lab 8: Create an HTML page with span
- Lab 9: Create a sample Web Page that highlights all the 6 types of headings
- Lab 10: HTML Paragraphs

Exercise:

- Lab 11: Create a Web page with HTML Links
- Lab 12: Creating Lists in HTML
- Lab 13: Create a Web page using different HTML colors
- Lab 14: Creating tables in HTML document
- Lab 15: HTML attributes
- Lab 16: Adding attributes to existing HTML elements
- Lab 17: Create a Web page using HTML Core attributes
- Lab 18: Create a Web page using HTML Style attribute
- Lab 19: Create a Web page using HTML Class attribute
- Lab 20: Create a Web page using HTML generic attributes

Exercise:

- Lab 21: Create HTML Web page using src attribute
- Lab 22: Create a Web page by using HTML alt attribute
- Lab 23: Create HTML Web page using lang attribute
- Lab 24: Creating a simple form in HTML

Lab 25: Create an HTML Webpage using inline style sheet

Lab 26: Create a Web page using CSS selectors

Exercise :

Lab 27: Create an HTML Web page using CSS color

Lab 28: Create an HTML Web page using CSS background property

Lab 29: Create an HTML Web page using CSS border

Lab 30: Create an HTML Web page using CSS margin.

Lab 31: Create an HTML Web page using CSS padding

Lab 36: Create HTML Web page using CSS height/width property

Lab 32: Create an HTML Web page using CSS box property

Lab 33: Create HTML Web page using CSS outline property

Lab 34: Create HTML Web page using CSS font property

Lab 35: Create HTML Web page using CSS icons

Lab 36: Create HTML Web page using CSS links

Lab 37: Create HTML Web page using CSS lists

Lab 38: Create HTML Web page using CSS table

Lab 39: Create HTML Web page using CSS Opacity

Lab 40: Create HTML Web page using CSS Units

NodeJS Labs:

Lab 41: Exporting an object from a module using exports Object

Lab 42: Exporting nested objects and function from Module using exports Object

Lab 43: Writing to the server using request-response statements as a callback in createServer() function

Lab 44: Reading into a file asynchronously and writing code for handling error if file not found to read

Lab 45: Reading a text file on the server using http and fs module

Lab 46: Firing two events simultaneously simulating connection and receiving information from any database or server

Reactjs Labs

Lab 47: Using keyword Class, creating object and passing parameters to the method ()

Lab 48: Embed JavaScript functions as an expression inside the curly brackets{} for use in JSX

Lab 49: Using String Literal as JSX attribute with nesting elements

Lab 50: Designing a clock showing current time

Input Code and Resulting Output

Lab 51: Using both Props and State combined

React Native Labs

Lab 52: Creating a prop in React Native and using it multiple times.

Lab 53: Using Flex Dimension in different ways.

Lab 54: Updating the state using Button and styling the app

Lab 55: Working with Basic Button and Flex Layout

Lab 56: Using List View and Scroll View

Lab 57: Creating and using different types of touchable buttons

HTML and CSS Labs

Lab 1 : Understanding of HTML

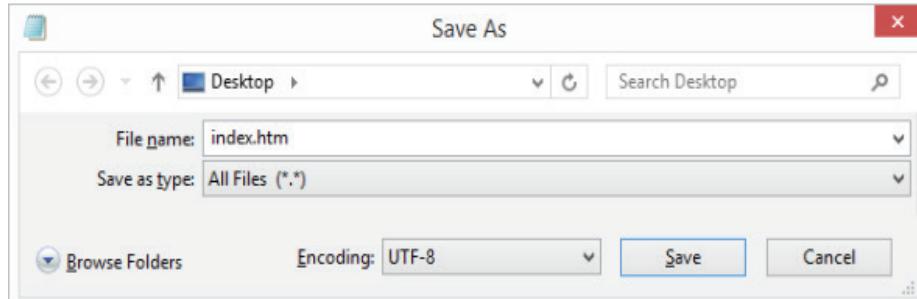
```
<!DOCTYPE html>
<html>
  <body>
    <h1> First Heading - My First HTML Course
    </h1>
    <p> First Paragraph - This is the first lab of
    course </p>
  </body>
</html>
```

Lab 2: Creating and Saving an HTML Document

- Step 1: Open Notepad on your PC
 - **Windows 7 or later:** Click on the Start Screen (the window symbol at the bottom left on your screen). Type Notepad.
 - **Windows 7 or earlier:** Open Start > Programs > Accessories > Notepad
- Step 2: Write HTML code in notepad

```
<!DOCTYPE html>
<html>
  <body>
    <TITLE> HTML Course </TITLE>
    <head> Welcome to course </head>
  </body>
</html>
```

- Step 3: Save the HTML Page
 - Save the file on your computer. Select File > Save as in the Notepad menu.
 - Name the file "index.html" and set the encoding to UTF-8 (which is the preferred encoding for HTML files).



- Step 4: View the HTML Page in Your Browser
 - Open the saved HTML file in your favorite browser (double click on the file, or right-click - and choose "Open with").

Note: Follow the same method to create HTML documents in other exercises.

Lab 3: Creating a Sample HTML web page

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<html>
  <head>
    <TITLE> HTML Course </TITLE>
  </head>
  <h1> First day in course </h1>
    <p> We are learning <b>bold</b> and printing 2
lines in a list: </p>
  <ul>
    <li> This is first line in my list </li>
      <li> This is second <i> line </i> in your list
    </li>
  </ul>
  </body>
</html>
```

Lab 4: Creating a Sample HTML web page

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<html>
  <head>
    <TITLE> HTML Course </TITLE>
  </head>
  <h1> This is my second webpage</h1>
  <ul>
    <li> Day 1
      <ul>
        <li> Introduction to HTML </li>
        <li> HTML Basics </li>
      </ul>
    </li>
    <ul>
      <li> Day 2
        <ul>
          <li> HTML Basics </li>
          <li> HTML Attributes </li>
        </ul>
      </li>
    </ul>
  </body>
</html>
```

Expected Output

This is my second webpage

- Day 1
 - Introduction to HTML
 - HTML Basics
- Day 2
 - HTML Basics
 - HTML Attributes

Lab 5: Formatting text in HTML

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<p><b> I am bold </b><p>
<p><code> I am computer code </code><p>
<p><em> I am emphasized</em></p>
<p><i> I am italics </i></p>
<p><small> I am small</small></p>
<p><mark> I am marked </mark></p>
<p><strong> I am strong </strong></p>
<p> I am <sub> subscript</sub></p>
<p> I am <sup> superscript </sup><p>
<p> <ins> I am inserted in document </ins></p>
<p><del> I am deleted from the document </del></p>
```

Expected Output

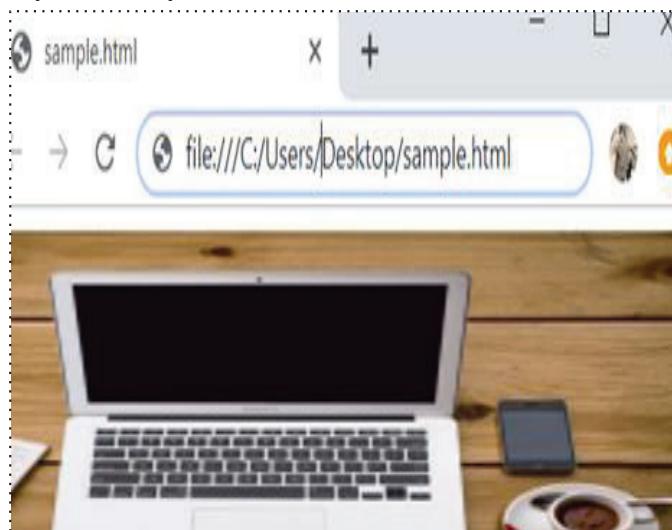
I am bold
I am computer code
<i>I am emphasized</i>
<i>I am italics</i>
I am small
I am marked
I am strong
I am _{subscript}
I am ^{superscript}
<u>I am inserted in document</u>
<u>I am deleted from the document</u>

Lab 6: Adding images to HTML Web pages

- Copy an image to C:\image.jpg.
- Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<html>
  <head>
    <style>
      img {
        width :100%;
      }
    </style>
  </head>
  <body>
    <img scr="C:/image.jpg" alt="HTML5 image"
      width="100" heighth="100">
    <img scr="C:/image.jpg" alt="HTML5 image" style
      ="width:100px;heighth=100px;">
  </body>
</html>
```

Expected Output



- Example 2: Div inside Div : Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<div id="Div" name="Div" title="Example Div Element"
style="font-family: Helvetica; font-size: 12pt; border:
1px solid black;">
    <div id="subDiv1" name="subDiv1" title="Subdivision
Div Element" style="color: #FF0000; border: 1px dotted
black;">
        <h5>Section 1</h5>
        <p>This is the first paragraph for content...</p>
        <p>Here's another content paragraph.</p>
    </div>
    <br />
    <div id="subDiv2" name="subDiv2" title="Subdivision
Div Element" style="color: #FF00FF; border: 1px dashed
black;">
        <h5>Section 2</h5>
        <p>This is the first paragraph for content...</p>
        <p>Here's another content paragraph.</p>
    </div>
</div>
```

Expected Output :

<p>Section 1</p> <p>This is the first paragraph for content...</p> <p>Here's another content paragraph.</p>	
<p>Section 2</p> <p>This is the first paragraph for content...</p> <p>Here's another content paragraph.</p>	

Lab 7: Create HTML page with Div tag, div inside div tag

Example 1: Single Div : Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<div id="Div" name="myDiv" title="Example Div Element"
style="color: #0900C4; font: Helvetica 12pt; border: 1px
solid black;">

<h5>Single Div</h5>

<p>This is the first paragraph for content...</p>

<p>Here's another content paragraph.</p>

</div>
```

Expected Output:



Example 2: Div inside Div : Open notepad and type the below code and click on the saved HTML file in your favorite browser

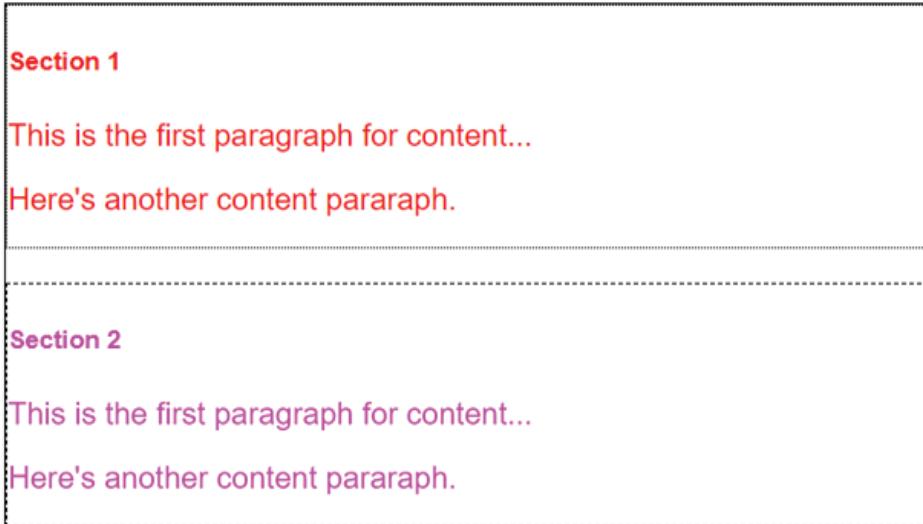
```
<div id="Div" name="Div" title="Example Div Element"
style="font-family: Helvetica; font-size: 12pt; border:
1px solid black;">

<div id="subDiv1" name="subDiv1" title="Subdivision Div
Element" style="color: #FF0000; border: 1px dotted
black;">

<h5>Section 1</h5>
```

```
<p>This is the first paragraph for content...</p>
<p>Here's another content paragraph.</p>
</div>
<br />
<div id="subDiv2" name="subDiv2" title="Subdivision
Div Element" style="color: #FF00FF; border: 1px dashed
black;">
<h5>Section 2</h5>
<p>This is the first paragraph for content...</p>
<p>Here's another content paragraph.</p>
</div>
</div>
```

Expected Output:



Lab 8: Create an HTML page with span

Example 1: Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<p>The <span style="color:green">span tag</span> is
used for grouping and applying styles to inline
elements..</p>
```

Expected Output:

The **span tag** is used for grouping and applying styles to inline elements..

Example 2: Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html>
<head>
<title> HTML Span tag </title>
</head>
<body>
<p>This is a paragraph <span
style="color:#FF0000;">
This is a paragraph </span> This is a
paragraph </p>
<p> <span style="color:#8866ff;">
This is another paragraph </span></p>

</body>
</html>
```

Expected Output:

This is a paragraph **This is a paragraph**This is a paragraph

This is another paragraph

Lab 9: Create a sample Web Page that highlights all the 6 types of headings

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<h1>Heading 1 - HTML Course</h1>
<h2>Heading 2 - HTML Course</h2>
```

```
<h3>Heading 3 - HTML Course</h3>
<h4>Heading 4 - HTML Course</h4>
<h5>Heading 5 - HTML Course</h5>
<h6>Heading 6 - HTML Course</h6>
```

Lab 10: HTML Paragraphs

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<p>This is first paragraph.</p>
<p>This is Second paragraph.</p>
<p>This is Third paragraph.</p>
```

Expected Output:

This is first paragraph.

This is Second paragraph.

This is Third paragraph.

Exercise:

Exercise 1: Create an HTML page and add 2 images.

Exercise 2: Create an HTML page of your choice using div and span tags.

Exercise 3: Create an HTML page that prints all 6 types on headings with different colors.

Lab 11: Create a Web page with HTML Links

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<!DOCTYPE html>
<html>
```

```

<body>

<h2>HTML Page Links</h2>

<p><a href="https://www.google.co.in">Google</a> is
the best search Engine.</p>

<p><a href="https://www.w3.org/">W3C</a> is a link
to a website on the World Wide Web.</p>

</body>
</html>

```

Expected Output

HTML Page Links

[Google](https://www.google.co.in) is the best search Engine.

[W3C](https://www.w3.org/) is a link to a website on the World Wide Web.

Lab 12: Creating Lists in HTML

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<html>
  <body>
    <h2>HTML Lists</h2>
    <ul> <b>Unordered-Lists</b>
      <li>HTML</li>
      <li>Java Script</li>
      <li>Agile</li>
    </ul>

```

```
<ol><b>Ordered-Lists</b>
    <li>Go to shopping site</li>
    <li>Put your selected items in cart</li>
    <li>Place the order</li>
</ol>
<dl><b>Definition List</b>
    <dt>HTML</dt>
    <dd>Hyper Text Mark-up Language.</dd>
    <dt>JS</dt>
    <dd>Java Script.</dd>
</dl>
</body>
</html>
```

Expected Output:

HTML Lists

Unordered-Lists

- HTML
- Java Script
- Agile

Ordered-Lists

1. Go to shopping site
2. Put your selected items in cart
3. Place the order

Definition List

HTML

Hyper Text Mark-up Language.

JS

Java Script.

Lab 13: Create a Web page using different HTML colors

Example 1: Changing background color

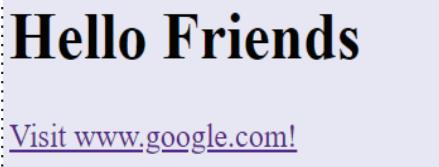
Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<html>
  <body bgcolor="#E6E6FA">
    <h1>Hello Friends</h1>
    <p><a href="https://www.google.com">Visit
      www.google.com!</a></p>
  </body>
</html>

```

Expected Output:



Hello Friends

Visit [www.google.com!](https://www.google.com)

Example 2: Use different colors for body and background.

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<!DOCTYPE html>
<html>
  <head>
    <title>HTML Colors by Name</title>
  </head>
  <body text = "blue" bgcolor = "orange">
    <p>Use different color names for body and table
      and see the result.</p>
    <table bgcolor = "black">
      <tr>
        <td>
          <font color = "white"> This text will
            appear white on black background.</font>
        </td>
      </tr>
    </table>
  </body>
</html>

```

Expected Output

Use different color names for body and table and see the result.

This text will appear white on black background.

Lab 14: Creating tables in HTML document

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<table border="1">
  <thead>
    <tr>
      <th>No.</th>
      <th>Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>John Prince</td>
      <td>johnp@mail.com</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Rickie May</td>
      <td>May@mail.com</td>
    </tr>
    <tr>
      <td>3</td>
      <td>Thomas Cook</td>
      <td>ThomasC@mail.com</td>
    </tr>
  </tbody>
</table>
```

```

        </tr>
    </tbody>
</table>
```

Lab 15: HTML attributes

Example: Use attributes Link, Paragraph and colored attribute and create a Web page

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```

<html>
    <body>
        <title> HTML Attribute Example </title>
        <p style="color:blue">HTML attribute example</p>
        <p title="I'm a tooltip">
            I am a Paragraph.
        </p>
        <a href="https://www.google.com/">This is a
            link</a>
    </body>
</html>
```

Expected Output:

HTML attribute example

I am a Paragraph.

This is a link

Lab 16: Adding attributes to existing HTML elements

Example: Add a new image to existing Web page

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```

<!DOCTYPE html>
<html>
    <body>
```

```
<title> HTML Attribute Example </title>
<p style="color:red">HTML attribute example</p>
<p title="I'm a tooltip">
This is a paragraph.</p>
<a href="https://www.google.com/">This is a
link</a>
<p style="color:red">Adding new image</p>

<p style="color:red">Increasing the size of
image</p>

</body>
</html>
```

Expected Output:

HTML attribute example

This is a paragraph.

This is a link

Adding new image



Lab 17: Create a Web page using HTML Core attributes

Example: Using HTML core attributes create a Web page

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html>
<head>
<style>
.red {
    background-color: red;
    color: white;
    padding: 8px;
}

.blue {
    background-color: blue;
    color: white;
    padding: 8px;
}
</style>
</head>
<body>
<p class ="red">Printing paragraph in red using
class and style attribute</p>
<p class="blue">Printing paragraph in blue
using class and style attribute</p>
</body>
</html>
```

Expected Output:

Printing paragraph in red using class and style attribute

Printing paragraph in blue using class and style attribute

Lab 18: Create a Web page using HTML Style attribute

Example 1: Printing text color in red using style attribute

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<!doctype html>
<html>
<head>
<style>
    p {
        color: red;
    }
</style>
</head>
<body>
    <p>Printing text in red using Style tag.</p>
</body>
</html>
```

Expected Output:

Printing text in red using Style tag.

Example 2: Using style tag to modify the font of text:

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<html>
<head>
```

```

<title>Font Family</title>
</head>
<body>
    <h1 style="font-family:commanders;">Example 1
    of Style tag</h1>
    <h2 style="font-family:Chaparral Pro
    Light;">Example 2 of Style tag</h2>
    <h3 style="font-family:algerian;">Example 3 of
    Style tag</h3>
    <p style="font-family:Castellar;">Example 4 of
    Style tag</p>
</body>
</html>

```

Expected Output:

Example 1 of Style tag

Example 2 of Style tag

EXAMPLE 3 OF STYLE TAG

EXAMPLE 4 OF STYLE TAG



Lab 19: Create a Web page using HTML Class attribute

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<html>
    <head>
        <style>
            .course {
                background-color: red;
                color: white;
                padding: 8px;

```

```

        }
    </style>
</head>

<body>
    <h2 class = "course">Day 1</h2>
    <p>We learn HTML</p>
    <h2 class = "course"> Day 2</h2>
    <p>We practise.</p>
    <h2 class = "course">Day 3</h2>
    <p>We are expert.</p>
</body>
</html>

```

Expected Output:**Day 1**

We learn HTML

Day 2

We practise.

Day 3

We are expert.

Lab 20: Create a Web page using HTML generic attributes

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<!DOCTYPE html>
<html>
<head>
<body bgcolor="#E6E6FA">
<div align="center">
Moving text to center!<br>
Changing the color of background using bgcolor
attribute!

```

```
</div>
</body>
</html>
```

Expected Output:

Moving text to center!
Changing the color of background using bgcolor attribute!

Exercise:

Exercise 1: Create HTML page and print symbols - dollar, euro and christmas tree

Exercise 2: Create HTML page with different text formats using class attribute

Exercise 3: Create HTML page with links to another Web page and an image



Lab 21: Create HTML Web page using src attribute

Example: Display image using src attribute. Copy image.jpg to C: Drive

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<!DOCTYPE html>
<html>
    <head>
        <title>Using Image in Webpage</title>
    </head>
    <body>
        <p>Displaying image using src</p>
        <img src = "C:/image.jpg" />
    </body>
</html>
```

Expected Output:



Lab 22: Create a Web page by using HTML alt attribute

Example 1: Display image using alt. attribute. We will use the same HTML code from Lab 23 and we add an extra alt attribute. Rename/Delete the image.jpg from C: Drive

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<html>
  <head>
    <title>Using Image in Webpage</title>
  </head>
  <body>
    <p>Displaying image using src</p>
    <img src = "C:/image.jpg" alt = "You see this
      coz loading image was a problem  " />
  </body>
</html>
```

Expected Output:

Displaying image using src

 You see this coz loading image was a problem

Lab 23: Create HTML Web page using lang attribute

Example 1: Use lang attribute with language - DE

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<html lang="en">
  <body>
    <h1>I am a heading <span lang="de-DE">Ich lerne
    HTML
    </span></h1>
  </body>
</html>
```

Expected Output

I am a heading Ich lerne HTML

Note: Refer this page to find all the supported languages

https://en.wikipedia.org/wiki/ISO_639-3#Special_codes

Lab 24: Creating a simple form in HTML

Example 1: Creating a basic user registration form

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html lang="en">
  <body>
    <form>
      <fieldset>
```

```
<legend>Registration</legend>
<label>Username: <input type="text"></label>
<label>Password: <input type="password">
</label>
</fieldset>
<fieldset>
<input type="radio" name="sex" id="male">
<label for="male">Male</label>
<input type="radio" name="sex" id="female">
<label for="female">Female</label>
</fieldset>
<fieldset>
<label for="file-select">Upload form:</label>
<input type="file" name="upload"
id="file-select">
</fieldset>

<fieldset>
<label for="address">Address:</label>
<textarea rows="3" cols="30" name="address"
id="address"></textarea>
</fieldset>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Expected Output

Registration

Username: Password:

Male Female

Upload form: Choose File No file chosen

Address:

Submit

Example 2: Validation of required attribute:

```
<head>
</head>
<body>

    <form name= "Field" action= "#">
        First name: <input type= "text" name= "fname" required>
        <input type= "submit" value= "Submit">
    </form>
</body>
```

Expected Output:

Without entering name click on Submit to see the validation message

First name: Submit

Please fill out this field.

Example 3: Validation of numbers in required format.

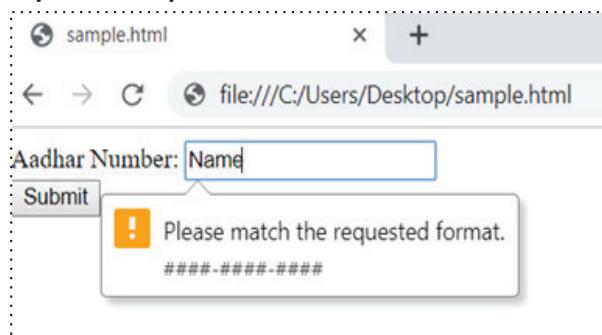
```
<form>
    <div>
```

```

<label>
    Aadhar Number:
    <input id="aadhar"
        required
        pattern="^\d{3}-\d{2}-\d{4}$"
        title="####-###-###" />
</label>
</div>
<div>
    <input type="submit" />
</div>
</form>

```

Expected Output



Lab 25: Create an HTML Webpage using inline style sheet

Example : Use CSS Internal style sheet and create a Web page

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<head>
<head>

<link rel="stylesheet" type="text/css"
href="style.css">

</head>
</head>
<body>
    <h1> </h2>

```

```

<h1 style="color:blue;margin-left:30px;">Printing
heading 1 in Blue using Inline style sheet</h1>
<h2 style="color:red;margin-left:30px;">Printing
heading 2 in Red using Inline style sheet</h2>
</body>

```

Expected Output:

**Printing heading 1 in Blue using
Inline style sheet**

Printing heading 2 in Red using Inline style sheet

Lab 26: Create a Web page using CSS selectors

Example1:Create HTML webpage using CSS element selector

Open notepad and type the below code and click on the saved HTML file in your favorite browser

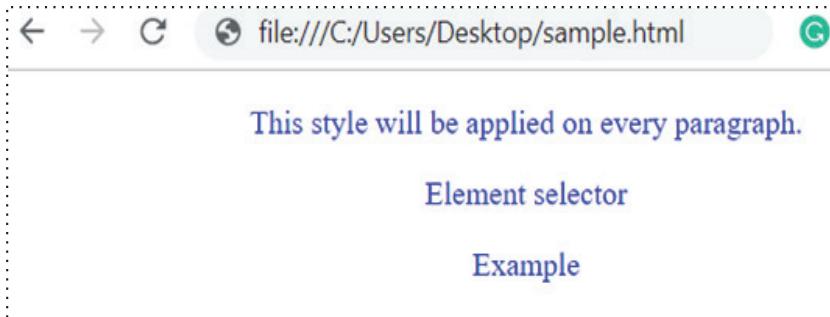
```

<!DOCTYPE html>
<html>
<head>
<style>
p{
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p>This style will be applied on every paragraph.</p>
<p id="para1">Element selector </p>
<p>Example</p>
</body>

```

```
</html>
```

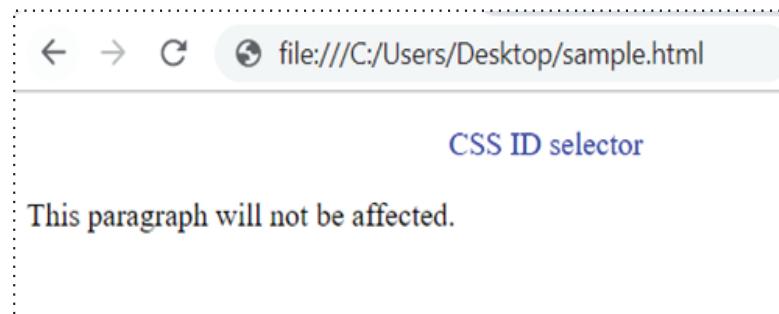
Expected Output:



Example-2: Create HTML webpage using CSS ID selector

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<!DOCTYPE html>
<html>
<head>
<style>
#para1 {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<p id="para1">CSS ID selector</p>
<p>This paragraph will not be affected.</p>
</body>
</html>
```

Expected Output:

Example 3: Create an HTML webpage using CSS class selector.

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
    text-align: center;
    color: blue;
}
</style>
</head>
<body>
<h1 class="center">This heading is blue and center-aligned (Class selector).</h1>
<p class="center">This paragraph is blue and center-aligned (Class selector).</p>
</body>
</html>
```

Expected Output:

**This heading is blue and center-aligned
(Class selector).**

This paragraph is blue and center-aligned (Class selector).

Exercise :

Exercise 1: Create a simple Webpage using Internal, external and inline style sheet

Exercise 2: Create a form for bank account registration

Exercise 3: Create a Web page using lang and src attribute



Lab 27: Create an HTML Web page using CSS color

Example: Using CSS Color attribute create an HTML web page.

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<!DOCTYPE html>
<html>
<head>
<style>
    body {
        color: red;
    }
    h1 {
        color: #00ff00;
    }
    p.ex {
        color: rgb(0,0,255);
    }
</style>
</head>
<body>
    <h1>Hello World!</h1>
    <p>This is my first web page using CSS color attribute.</p>
</body>
</html>
```

```

        }
</style>
</head>
<body>
    <h1>This is heading 1</h1>
    <p>This is an ordinary paragraph.</p>
    <p class="ex">This is a paragraph with
    class="ex". This text is blue.</p>
</body>
</html>

```

Expected Output:

This is heading 1

This is an ordinary paragraph.

This is a paragraph with class="ex". This text is blue.

Lab 28: Create an HTML Web page using CSS background property

- Example : Create HTML webpage using CSS background style. Create image.jpg in the same folder where the HTML file is created
- Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<!DOCTYPE html>
<html>
    <head>
        <style>
            body {
                background-image: url("image.jpg");
                background-repeat: repeat-y;
                background-color: grey;
                background-position: center;
            }
        </style>
    </head>
    <body>
        <h1>This is heading 1</h1>
        <p>This is an ordinary paragraph.</p>
        <p class="ex">This is a paragraph with
        class="ex". This text is blue.</p>
    </body>
</html>

```

```
</body>  
</html>
```

Expected Output :



¶

Lab 29: Create an HTML Web page using CSS border

Example : Using CSS border property create an HTML web page.

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
    p.dotted {border-style: dotted;}  
    p.dashed {border-style: dashed;}
```

```
    p.solid {border-style: solid; }
    p.double {border-style: double; }
    p.groove {border-style: groove; }
    p.ridge {border-style: ridge; }
    p.inset {border-style: inset; }
    p.outset {border-style: outset; }
    p.none {border-style: none; }
    p.hidden {border-style: hidden; }
    p.mix {border-style: dotted dashed solid
double; }

</style>
</head>
<body>
<h2>The border-style Property</h2>
<p>This property specifies what kind of border to
display:</p>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="none">No border.</p>
<p class="hidden">A hidden border.</p>
<p class="mix">A mixed border.</p>

</body>
</html>
```

Expected Output:

The border-style Property

This property specifies what kind of border to display:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

No border.

A hidden border.

A mixed border.

Lab 30: Create an HTML Web page using CSS margin.

Example 1: Display image using src attribute. Copy image.jpg to C: Drive
Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    margin-top: 100px;
```

```
        margin-bottom: 100px;  
        margin-right: 150px;  
        margin-left: 80px;  
        background-color: lightblue;  
    }  
</style>  
</head>  
<body>  
  
<h2>CSS margin properties</h2>  
  
<div>The text placed here has a top margin of 100px,  
a right margin of 150px, a bottom margin of 100px,  
and a left margin of 80px.</div>  
  
</body>  
</html>
```

Expected Output:

CSS margin properties

The text placed here has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.

Lab 31: Create an HTML Web page using CSS padding

Example 1: Create a Webpage with CSS padding

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>  
<html>
```

```
<head>
<style>
div {
    border: 1px solid black;
    background-color: lightgrey;
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
</style>
</head>
<body>
    <h2>CSS padding properties</h2>
    <div>This text placed here has a top padding of
    50px, a right padding of 30px, a bottom padding of
    50px, and a left padding of 80px.</div>
</body>
</html>
```

Expected Output:

CSS padding properties

This text placed here has a top padding of 50px, a right padding of 30px, a bottom padding of 50px, and a left padding of 80px.

Lab 36: Create HTML Web page using CSS height/width property

Example 1: Create HTML webpage using CSS height/width property

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    height: 150px;
    width: 70%;
    background-color: powderblue;
}
</style>
</head>
<body>
<h2>Setting the height/width of an element</h2>
<p>This div element has a height of 150px and a width of 70%:</p>
<div></div>
</body>
</html>
```

Expected Output:

Setting the height/width of an element

This div element has a height of 150px and a width of 70%:



Lab 32: Create an HTML Web page using CSS box property

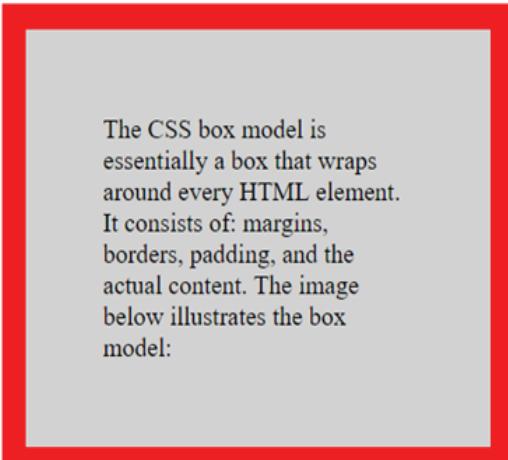
Example: Create an HTML webpage using CSS box property

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    background-color: lightgrey;
    width: 200px;
    border: 15px solid red;
    padding: 50px;
    margin: 20px;
}
</style>
</head>
<body>
<h2>CSS Box Model</h2>
<p>Demonstrating CSS box model</p>
<div>The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:</div>
</body>
</html>
```

Expected Output:**CSS Box Model**

Demonstrating CSS box model



The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

Lab 33: Create HTML Web page using CSS outline property

Example : Create HTML web page using CSS outline property

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {
    border: 1px solid black;
    outline-style: solid;
    outline-color: red;
}
p.ex2 {
    border: 1px solid black;
```

```

        outline-style: double;
        outline-color: grey;
    }
    p.ex3 {
        border: 1px solid black;
        outline-style: outset;
        outline-color: blue;
    }
</style>
</head>
<body>
<h2>The outline-color Property</h2>
<p class="ex1">A solid red outline.</p>
<p class="ex2">A double green outline.</p>
<p class="ex3">An outset yellow outline.</p>
</body>
</html>

```

Expected Output:**The outline-color Property**

A solid red outline.

A double green outline.

An outset yellow outline.

Lab 34: Create HTML Web page using CSS font property

Example: Create HTML web page using CSS font property

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<!DOCTYPE html>
<html>
```

```
<head>
<style>
    p.normal {
        font-style: normal;
    }
    p.italic {
        font-style: italic;
    }
    p.oblique {
        font-style: oblique;
    }
    h1 {
        font-size: 2.5em;
    }
    h2 {
        font-size: 1.875em;
    }
    p {
        font-size: 0.875em;
    }
</style>
</head>
<body>
    <p class="normal">Text in normal style.</p>
    <p class="italic">Text in italic style.</p>
    <p class="oblique">Texts in oblique style.</p>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <p>This is a paragraph.</p>
</body>
</html>
```

Expected Output:

Text in normal style.

Text in italic style.

Texts in oblique style.

This is heading 1

This is heading 2

This is a paragraph.



Lab 35: Create HTML Web page using CSS icons

Example: Create HTML web page using CSS icons

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html>
<head>
    <title>Font Awesome Icons</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
        href="https://use.fontawesome.com/releases/v5.7.0/css/all.css" integrity="sha384-1ZN37f5QGtY3VHgisS14W3ExzMWZxybE1SJSEsQp9S+oqd12jhcu+A56Ebc1zFSJ" crossorigin="anonymous">
</head>
<body>
    <p>Font Awesome icons:</p>
    <i class="fas fa-cloud"></i>
```

```

<i class="fas fa-heart"></i>
<i class="fas fa-car"></i>
<i class="fas fa-file"></i>
<i class="fas fa-bars"></i>
<p>Styled Font Awesome icons (size and
color):</p>
<i class="fas fa-cloud" style="font-
size:24px;"></i>
<i class="fas fa-cloud" style="font-
size:36px;"></i>
<i class="fas fa-cloud" style="font-
size:48px;color:red;"></i>
<i class="fas fa-cloud" style="font-
size:60px;color:lightblue;"></i>
</body>
</html>

```

Expected Output:

Font Awesome icons:



Styled Font Awesome icons (size and color):



Lab 36: Create HTML Web page using CSS links

Example : Create HTML web page using CSS links property

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```

<html>
  <head>
    <style type = "text/css">
      a:link {color: #000000}
      a:visited {color: #006600}
    </style>
  </head>
  <body>
    <a href="#">My Link</a>
  </body>
</html>

```

```
a:hover {color: #FFCC00}
a:active {color: #FF00CC}
</style>
</head>
<body>
    <a href="">This is a Link.</a>
    <p> Hover the mouse on the link </p>
</body>
</html>
```

Expected Output:

This is a Link.

Hover the mouse on the link

Lab 37: Create HTML Web page using CSS lists

Example: Create HTML web page using CSS lists property

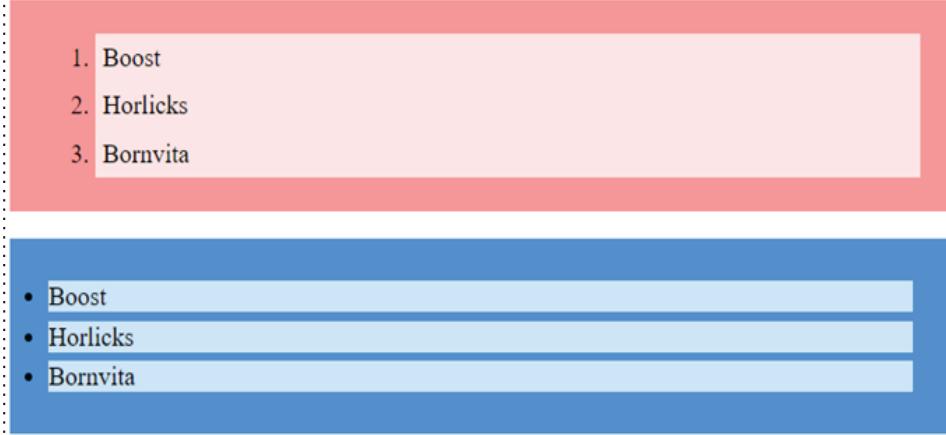
Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html>
<head>
    <style>
        ol {
            background: #ff9999 ;
            padding: 20px;
        }
        ul {
            background: #3399ff;
            padding: 20px;
        }
        ol li {
            background: #ffe5e5;
        }
    </style>
</head>
<body>
    <ol>
        <li>Item 1</li>
        <li>Item 2</li>
        <li>Item 3</li>
    </ol>
    <ul>
        <li>Item 1</li>
        <li>Item 2</li>
        <li>Item 3</li>
    </ul>
</body>
</html>
```

```
padding: 5px;
margin-left: 35px;
}
ul li {
background: #cce5ff;
margin: 5px;
}
</style>
</head>
<body>
<h1>Styling Lists With Colors:</h1>
<ol>
<li>Boost</li>
<li>Horlicks</li>
<li>Bournvita</li>
</ol>
<ul>
<li>Boost</li>
<li>Horlicks</li>
<li>Bournvita</li>
</ul>
</body>
</html>
```

Expected Output:

Styling Lists With Colors:



Lab 38: Create HTML Web page using CSS table

Example: Create HTML web page using CSS table property

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<html>
<head>
    <style type = "text/css">
        table.empty {
            width:350px;
            border-collapse:separate;
            empty-cells:hide;
        }
        td.empty {
            padding:5px;
            border-style:solid;
            border-width:1px;
            border-color:#999999;
```

```

        }
</style>
</head>
<body>

    <table class = "empty">
        <tr>
            <th></th>
            <th>B.Tech</th>
            <th>M.Tech</th>
        </tr>

        <tr>
            <th>SEM A</th>
            <td class = "empty">value</td>
            <td class = "empty">value</td>
        </tr>

        <tr>
            <th>SEM B</th>
            <td class = "empty">value</td>
            <td class = "empty"></td>
        </tr>
    </table>

</body>
</html>

```

Expected Output:

	B.Tech	M.Tech
SEM A	value	value
SEM B	value	

Lab 39: Create HTML Web page using CSS Opacity

Example: Create HTML web page using CSS opacity property

Open notepad and type the below code and click on the saved HTML file in your favorite browser.

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Opacity</title>
    <style>
        div {
            height: 50px;
            padding: 20px;
            background-color: #808080;
            font-size: 22px;
        }
        #opacity1 {
            opacity: 0.2;
            filter: Alpha(opacity=20);
        }
        #opacity2 {
            opacity: 0.5;
            filter: Alpha(opacity=50);
        }
        #opacity3 {
            opacity: 1;
            filter: Alpha(opacity=100);
        }
    </style>
</head>

<body>
    <div id="opacity1">Opacity Value: 0.2</div>
    <div id="opacity2">Opacity Value: 0.5</div>
```

```
<div id="opacity3">Opacity Value: 1</div>
</body>
</html>
```

Expected Output:



Lab 40: Create HTML Web page using CSS Units

Example : Create HTML web page using CSS units property

Open notepad and type the below code and click on the saved HTML file in your favorite browser

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
font-size: 16px;
}
.slogan {
font-size: 1.2em;
padding: 1.2em;
```

NodeJS Labs:

Lab 41: Exporting an object from a module using exports Object

Learn and Practice the code for exporting an object from one module to another using exports Object.

- Write and execute the code for exporting an object from one module to another using exports Object.
- Create two files: moduleOne.js and moduleTwo.js

ModuleOne.js

```
var webCourse = {
  own_name: "Rhea" ,
  found_year: 2017
};

module.exports.own_name = webCourse.own_name;
module.exports.fond_year = webCourse.found_year;
```

ModuleTwo.js

```
var e1 = require('./moduleOne.js');

console.log("Owner of webCourse is", e1.own_name);
console.log("Founding Year of webCourse is",
e1.found_year);
```

Lab 42: Exporting nested objects and function from Module using exports Object

Learn and Practice the code for exporting nested objects and function from Module using exports Object

- Write and execute the code for exporting nested objects and function from Module using exports Object
- Create two files: myXmodule.js and App.js

myXmodule.js

```
const myXdigital= {
```

```

company = "DigitalMedia",
name: {own_name: "Rhea" ,found_year: 2017},
branches = function()
{
    return ["Noida", "Delhi"];
}
}

module.exports.exp = myXdigital;
App.js
var e1 = require ('./myXmodule.js');

console.log("Name of the company is", e1.exp.company);
console.log (e1.exp.name);
console.log ("Its Branches are located in:",
e1.exp.branches ());

```

Lab 43: Writing to the server using request-response statements as a callback in createServer() function

Learn and Practice the code for writing to the server using request-response statements as a callback in createServer() function.

- Write and execute the code for writing to the server using request-response statements as a callback in createServer() function.
- Pass the callback with two arguments of request (req) and response (res) within createServer() function
- Write on the server using res.write() and the request using res.end()

App.js

```

const http = require('http');

const server = http.createServer((req, res) => {
    if (req.url === '/') {
        res.write ('Welcome to DigitalMedia');
        res.end();
    }
})
server.listen(8000);

```

```

        }
    }) ;

server.listen(3000);

console.log('Listening to the port 3000...');

Listening on port 3000...

```

Lab 44: Reading into a file asynchronously and writing code for handling error if file not found to read

Learn and Practice the code for reading into a file asynchronously and writing code for handling error if file not found to read.

- Write and execute the code for reading into a file asynchronously and writing code for handling error if file not found to read.
- Create two files: myXinput.txt and App.js
- Write a code to read the data in the file myXinput.txt. Assume any arbitrary name of the file not present in the directory: not-there.txt and write a code for handling and displaying an error when the file not found

myXinput.txt

```
Welcome to DigitalMedia
A place where you'll learn all about coding,
programming language, and
designing databases from 'front' to 'back' literally!
```

App.js

```
var fs = require("fs");

fs.readFile('myXinput.txt', function (err, data) {
    if (err) {
        return console.error(err);
    }
    console.log(`\nAsynchronous read \n
${data.toString()}`);
});

fs.readFile('not-there.txt', function (err, data) {
```

```

        if (err) {
            return console.error(err);
        }
        console.log(`\nAsynchronous read \n
${data.toString()}\n`);
    });
}

```

Lab 45: Reading a text file on the server using http and fs module

Learn and Practice the code for reading a text file on the server using http and fs module.

- Write any text in a newly created text file: myXinput.txt
- Write and execute the code for reading a text file on the server using http and fs module.

App.js

```

const http = require ('http');
var fs = require ('fs');
http.createServer(function (req, res){
    fs.readFile ('myXinput.txt').function(err, data){
        res.writeHead(200, {'Content-Type':
'text/html'});
        res.write(data);
        res.end();
    });
}).listen(3000);

```

Lab 46: Firing two events simultaneously simulating connection and receiving information from any database or server

Learn and Practice the code firing two events simultaneously simulating connection and receiving information from any database or server

- Write and execute the code for firing two events simultaneously simulating connection and receiving information from any database or server

```
// Importing events module
```

```
var events = require('events');
var eventEmitter = new events.EventEmitter();

// Creating an event handler
var connectHandler = function connected() {
    console.log('Connection to DigitalMedia
successful.');

    // Firing the event 'info_received'
    eventEmitter.emit('info_received');

}

// Binding the connection event with the handler
eventEmitter.on('DigitalConnect', connectHandler);

// Binding the data_received event with the anonymous
function
eventEmitter.on('info_received', function(){
    console.log('Information received successfully.');
});

// Firing the connection event
eventEmitter.emit('DigitalConnect');
console.log("Program Ended.");

background-color: #ccc;
}
ul {
    font-size: .8em;
}
```

```
</style>
</head>
<body>
    Welcome to CSS Units module
<p class="slogan">Welcome to CSS Units module</p>

<ul>
<li>Top level
    <ul>
        <li>Second level
            <ul>
                <li>Third level
                    <ul>
                        <li>Fourth level
                            <ul>
                                <li>Fifth level</li>
                            </ul>
                        </li>
                    </ul>
                </li>
            </ul>
        </li>
    </ul>
</li>
</ul>
</body>
</html>
```

Expected Output:

Welcome to CSS Units module

Welcome to CSS Units module

- Top level
 - Second level
 - Third level
 - Fourth level
 - Fifth level

Reactjs Labs

Lab 47: Using keyword *Class*, creating object and passing parameters to the method ()

Learn and Practice the code for passing parameter to the method ()in the Class

- Use two id(s), pass a parameter to method: Courses (Module);
- Create two Objects

```

        ○ let myCourse1 = new Course("JavaScript");
        ○ let myCourse2 = new Course("React and Redux");
<html> <body>

<h3>Passing parameter to Method in the Class </h3>
<p id="Xba1"></p> <p id="Xba2"></p>

<script>
class Course {
    constructor(CourseName) {
        this.CourseName = CourseName;
    }
}

Courses(Module) {
    return Module + this.CourseName;
} }

let myCourse1 = new Course("JavaScript");
document.getElementById("Xba1").innerHTML =
myCourse1.Courses("Participants have learned ");

let myCourse2 = new Course("React and Redux");
document.getElementById("Xba2").innerHTML =
myCourse2.Courses("Currently, Participants are learning ");
</script> </body> </html>
```

Passing parameter to Method in the Class

Participants have learned JavaScript

Currently, Participants are learning React and Redux

Lab 48: Embed JavaScript functions as an expression inside the curly brackets{} for use in JSX

Learn and Practice the code for embedding JS functions as an expression inside curly brackets{} for use in JSX

- Write and execute the code for embedding JS functions as an expression inside curly brackets{} for use in JSX

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App.css';
function fullName(user) {
  return user.fName + ' ' + user.lName;
}
const user = {
  fName: 'Kevin',
  lName: 'Nash',
};
const element = <h2> Hello, {fullName(user)}!</h2>;
ReactDOM.render(element,
document.getElementById('root'));
export default App;
```

Hello, Kevin Nash!

Lab 49: Using String Literal as JSX attribute with nesting elements

Learn and Practice the code to use String literal as JSX attribute with nesting elements

- Write a code using JSX attribute with nesting elements, where attribute is used with <p> element and displayed as output given below.

```
import React, { Component } from 'react';
class App extends Component{
  render(){
    return(
```

```

        <div>
          <h2> Course </h2>
          <h2>Hello, Participants</h2>
          <p myXAttribute = "demo">'Getting comfortable
using JSX Attributes?' </p>
        </div>
      ) ;
    }
}

export default App;

```

Lab 50: Designing a clock showing current time

Learn and Practice the code for designing the clock showing current time

- Write and execute the code using properties this.state for designing the clock showing current time
- Use class Time, and Use this.state.date.toLocaleTimeString inside the render () method

Input Code and Resulting Output

```

import React from 'react';
import ReactDOM from 'react-dom';
class Time extends React.Component {
  constructor(props) {
    super(props);
    this.state = {date: new Date()};
  }
  render() {
    return (
      <div>
        <h2>It's Time to React with Course !</h2>
        <h2>It is
{this.state.date.toLocaleTimeString()}</h2>
      </div>
    )
  }
}

export default Time;

```

```

        ) ;
    }
}

ReactDOM.render(
  <Time />,
  document.getElementById('root')
) ;

```

Lab 51: Using both Props and State combined

Learn and Practice the code to use both Props and State

- Write a code combining the use of both Props and State to display the output
- Use: this.state = {
 name: "Course",
}
- Use class DIG for using Props with property: this.props.digProp

```
(App.js)
import React, { Component } from 'react';
class App extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      name: "Course "
    }
  }
  render() {
    return (
      <div>
        <DIG digProp = {this.state.name}/>
      </div>
    ) ;
  }
}
```

```
        }
    }

class DIG extends React.Component {
    render() {
        return (
            <div>
                <h1>State & Props Example</h1>
                <h3>Welcome to {this.props.digProp}</h3>
                <p>Using both State and Props
            combined</p>
            </div>
        ) ;
    }
}

export default App;
(Main.js)

import React from 'react';
import ReactDOM from 'react-dom';
import App from './App.js';

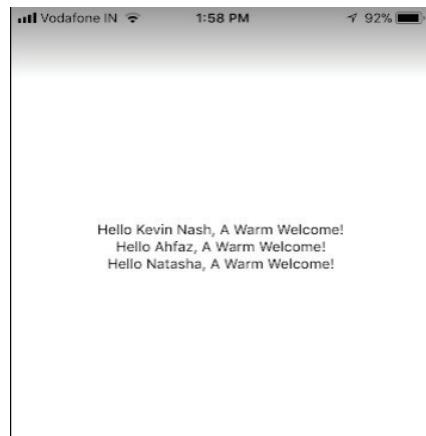
ReactDOM.render(<App/>,document.getElementById('app'));
```

React Native Labs

Lab 52: Creating a prop in React Native and using it multiple times.

Write and execute the code for creating a prop in React Native and using it multiple times

```
import React, { Component } from 'react';
import { Text, View } from 'react-native';
class Participants extends Component {
  render() {
    return (
      <View style={{alignItems: 'center'}}>
        <Text>Hello {this.props.name}, A Warm Welcome!</Text>
      </View>
    );
  }
}
export default class Greetings extends Component {
  render() {
    return (
      <View style={{alignItems: 'center', top: 200}}>
        <Participants name='Kevin Nash' />
        <Participants name='Ahfaz' />
        <Participants name='Natasha' />
      </View>
    );
  }
}
```

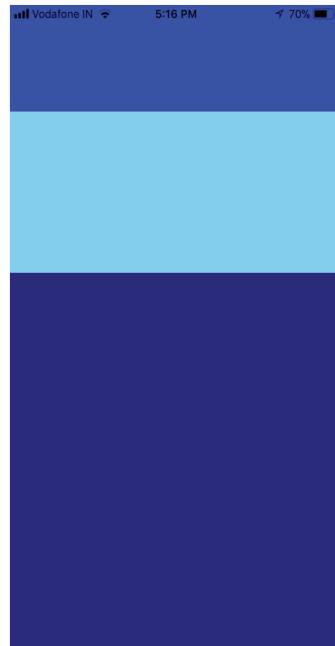


Lab 53: Using Flex Dimension in different ways.

Write a code using flex dimensions where:

- 3 different colors need to be displayed in 2:3:7 ratio
- Height of the parent view be 500 with display ratio 2:3:5

```
import React, { Component } from 'react';
import { View } from 'react-native';
export default class DimensionsFlex extends Component {
  render() {
    return (
      <View style={{flex: 1}}>
        <View style={{flex: 2, backgroundColor: 'blue'}}/>
        <View style={{flex: 3, backgroundColor: 'skyblue'}} />
        <View style={{flex: 7, backgroundColor: 'darkblue'}} />
      </View>
    );
  }
}
```



```
import React, { Component } from 'react';
import { View, Text } from 'react-native';
export default class DimensionsFlex extends Component {
  render() {
    return (
      <View style={{height: 500}}>
        <View style={{flex: 2, backgroundColor: 'blue'}}/>
          <View style={{flex: 3, backgroundColor: 'skyblue'}} />
            <View style={{flex: 5, backgroundColor: 'darkblue'}} />
          </View>
      );
    }
}
```



Lab 54: Updating the state using Button and styling the app

- Write and execute the code for updating the state through using button and styling the app
- Use StyleSheet and Button components to style and create interface for updating the state respectively. Use StyleSheet.create to define multiple styles

```
import React, { useState } from 'react';
import { StyleSheet, Text, View, Button } from 'react-native';

export default function App() {
  <Text> Hello Participants </Text>
  const [name, setName] = useState('Rachna Institute Manav');

  const [course, setCourse] = useState({ name: 'React', term: 2018 });

  const clickHandler = () => {
    setName('Manav Rachna Institute');
    setCourse({ name: 'React Native', term: 2019 });
  };
}
```

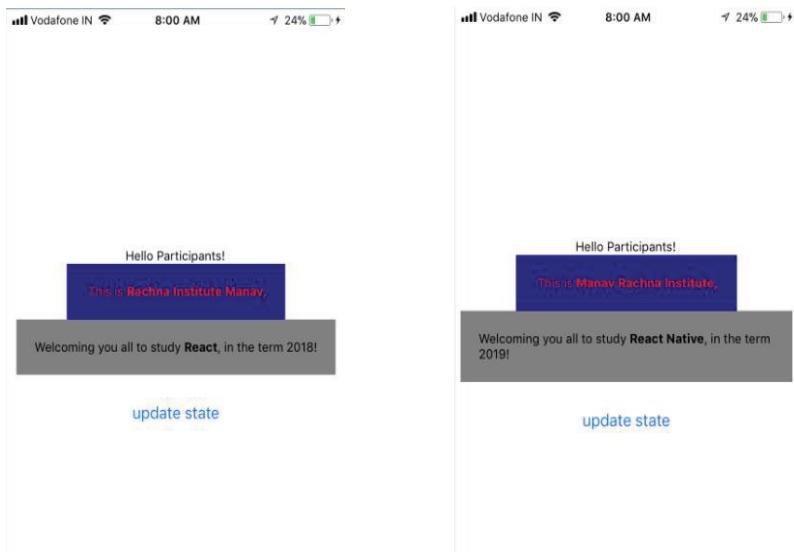
```
return (
  <View style={styles.container}>
    <Text> Hello Participants! </Text>
    <View style={styles.header}>
      <Text><Text style = {styles.textred}> This is
<Text style =
{styles.boldText}>{name} ,</Text></Text></Text>
    </View>
    <View style={styles.body}>
      <Text>Welcoming you all to study <Text style =
{styles.boldText}>{course.name}</Text>, in the term
{course.term}!</Text>
    </View>
    <View style={styles.buttonContainer}>
      <Button title='update state'
onPress={clickHandler} />
    </View>
  </View>
) ;
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: 'white',
    alignItems: 'center',
    justifyContent: 'center',
  },
  header: {
    backgroundColor: 'darkblue',
    padding: 20,
  },
  body: {
    backgroundColor: 'grey',
    padding: 20,
```

```

} ,
textred: {
  color: 'red',
} ,
buttonContainer: {
  marginTop: 20,
} ,
boldText: {
  fontWeight: 'bold',
}
}) ;

```



Lab 55: Working with Basic Button and Flex Layout

- Write and execute the code for working with basic button and designing an app using flex layout
- Use justifyContent and flexDirect to style the app and design layout. Use <Button> and its onPress attribute to pop an alert message.

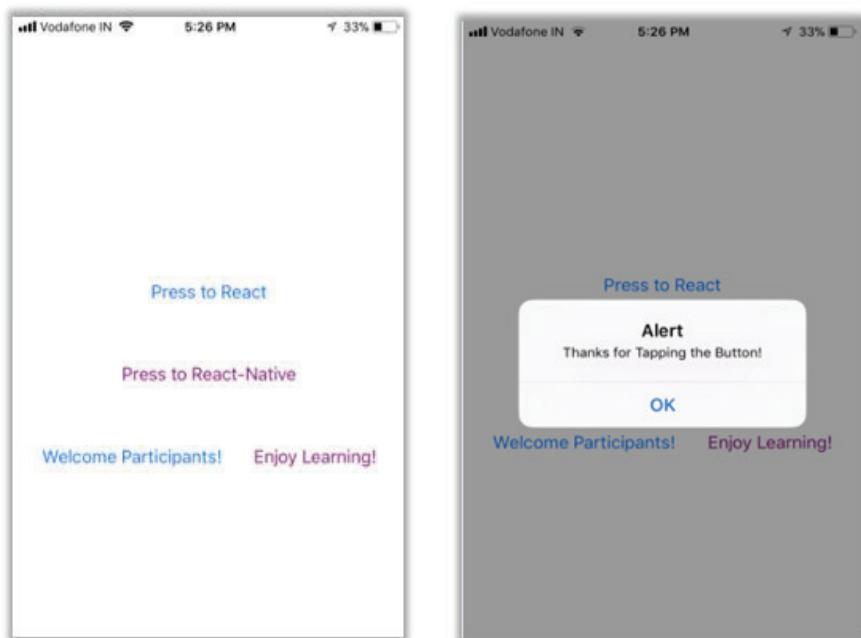
```

import React, { Component } from 'react';
import { Button, StyleSheet, View } from 'react-native';
export default class PrimitiveButton extends Component

```

```
{  
  _onPressButton() {  
    alert('Thanks for Tapping the Button!')  
  }  
  render() {  
    return (  
      <View style={styles.container}>  
        <View style={styles.buttonContainer}>  
          <Button  
            onPress={this._onPressButton}  
            title="Press to React"  
          />  
        </View>  
        <View style={styles.buttonContainer}>  
          <Button  
            onPress={this._onPressButton}  
            title="Press to React-Native"  
            color="#841584"  
          />  
        </View>  
        <View  
          style={styles.alternativeLayoutButtonContainer}>  
          <Button  
            onPress={this._onPressButton}  
            title="Welcome Participants!"  
          />  
          <Button  
            onPress={this._onPressButton}  
            title="Enjoy Learning!"  
            color="#841584"  
          />  
        </View>  
      </View>  
    )  
  }  
}
```

```
) ;  
}  
}  
  
const styles = StyleSheet.create({  
  container: {  
    flex: 1,  
    justifyContent: 'center',  
  },  
  buttonContainer: {  
    margin: 20  
  },  
  alternativeLayoutButtonContainer: {  
    margin: 20,  
    flexDirection: 'row',  
    justifyContent: 'space-between'  
  }  
});
```



Lab 56: Using List View and Scroll View

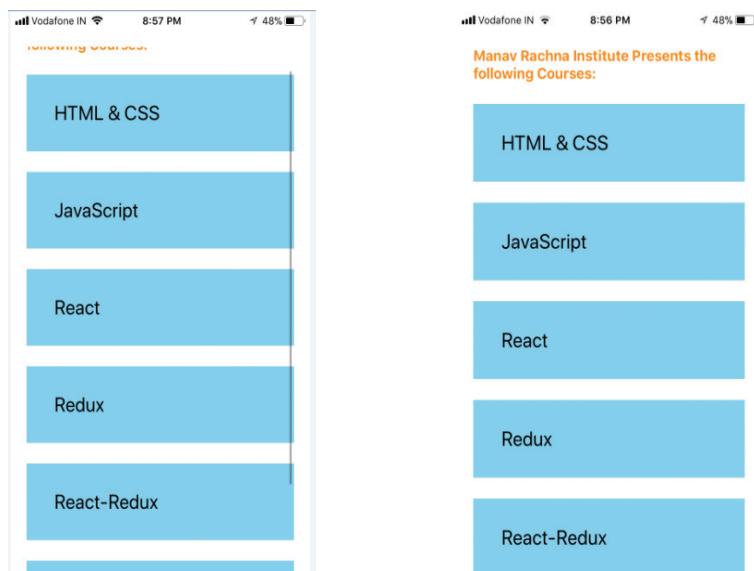
Write and execute the code for creating List and Scroll View in your application

- Use separate key or IDs for separate items
- Use <ScrollView> for imparting the Scroll View inside the app

```
import React, { useState } from 'react';
import { StyleSheet, Text, View, ScrollView } from
'react-native';

export default function App() {
  const [Courses] = useState([
    { name: 'HTML & CSS', key: '1' },
    { name: 'JavaScript', key: '2' },
    { name: 'React', key: '3' },
    { name: 'Redux', key: '4' },
    { name: 'React-Redux', key: '5' },
    { name: 'MySQL', key: '6' },
    { name: 'MongoDB', key: '7' },
  ]);
  return (
    <View style={styles.container}>
      <ScrollView>
        <Text><Text style = {styles.boldText}>Manav
        Rachna Institute Presents the following
        Courses:</Text></Text>
        {Courses.map(course => (
          <View key={course.key}>
            <Text style = {styles.course}>
              {course.name}</Text>
            </View>
          ))}
        </ScrollView>
      </View>
    );
}
```

```
const styles = StyleSheet.create({
  container: {
    flex: 1,
    paddingTop: 40,
    paddingHorizontal: 20,
    backgroundColor: '#fff',
  },
  boldText: {
    color: 'darkorange',
    fontSize: 18,
    fontWeight: 'bold',
  },
  course: {
    marginTop: 24,
    padding: 30,
    backgroundColor: 'skyblue',
    fontSize: 24,
  },
});
```



Lab 57: Creating and using different types of touchable buttons

- Write a code for creating and using different types of touchable buttons using TouchableHighlight, TouchableOpacity and TouchableWithoutFeedback
- Create separate alert messages when someone presses and/or long-presses the button

```
import React, { Component } from 'react';
import { StyleSheet, Text, TouchableHighlight,
TouchableOpacity,
TouchableWithoutFeedback, View } from 'react-native';
export default class MyXTouchableButtons extends
Component {
  _onPressButton() {
    alert('Thank you for tapping the button!')
  }
  _onLongPressButton() {
    alert('Thank you for long-pressing the button!')
  }
  render() {
    return (
      <View style={styles.container}>
        <Text><Text style = {styles.boldText}>Manav
Rachna Institute Presents Touchable Buttons
</Text></Text>
        <TouchableHighlight
onPress={this._onPressButton} underlayColor="white">
          <View style={styles.button}>
            <Text style={styles.buttonText}>Button that
gets Highlighted on Touch</Text>
          </View>
        </TouchableHighlight>

        <TouchableOpacity
onPress={this._onPressButton}>
          <View style={styles.button}>
```

```
        <Text style={styles.buttonText}>Button  
getting Opaque to Touch</Text>  
    </View>  
  </TouchableOpacity>  
  
  <TouchableWithoutFeedback  
    onPress={this._onPressButton}  
  >  
    <View style={styles.button}>  
      <Text style={styles.buttonText}>My  
Touchable Button Without Feedback</Text>  
    </View>  
  </TouchableWithoutFeedback>  
  <TouchableHighlight  
    onPress={this._onPressButton}  
    onLongPress={this._onLongPressButton}  
    underlayColor="white">  
    <View style={styles.button}>  
      <Text style={styles.buttonText}>Here,  
you've to Long Press the Touch</Text>  
    </View>  
  </TouchableHighlight>  
  </View>  
);  
}  
}  
const styles = StyleSheet.create({  
  container: {  
    paddingTop: 60,  
    alignItems: 'center'  
  },  
  button: {  
    marginTop: 20,  
    marginBottom: 30,  
  },
```

```
width: 260,  
alignItems: 'center',  
backgroundColor: 'steelblue'  
) ,  
buttonText: {  
  textAlign: 'center',  
  padding: 20,  
  color: 'white'  
) ,  
boldText: {  
  color: 'black',  
  fontSize: 16,  
  fontWeight: 'bold',  
}  
}  
} );
```

