AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY

USER INTERFACE DESIGN

23CSE113

LAB MANUAL

SUBMITTED BY	
NAME	DIVYA SAHITHI MASINENI
ROLL NO	AV.SC.U4CSE25129
YEAR/SEM/SECTION	1ST YEAR/ SEM-2/CSE-B
CREDITS	03

SUBMITTED TO	DR. RAJ KUMAR BATCHU
DESIGNATION	ASST.PROFFESSOR(SL Grade)
DEPARTMENT	SCHOOL OF COMPUTING

INDEX

S.NO	TITLE	PAGE NO	REMARKS
1	WEEK 1 : 1. Installation of VS CODE tool 2. Understanding basic of html structure 3. Understanding basic html tags	6-13	
2	WEEK 2 : 1. To design a simple web page using HTML that demonstrates the use of background color, background image, heading styling, and preformatted text 2.To create a static webpage using only HTML that displays basic information about Amrita Vishwa Vidyapeetham, including a heading, descriptive text, background styling through HTML attributes, and a table showing the list of campuses in different states. 3.To design a restaurants menu webpage using HTML and displaying menu items in a table format with proper layout and visual styling.	14-29	
3	WEEK 3 : 1. Construction of Class Timetable using html Elements 2 . Construction of web page Displaying Each Syllabus of each Subject Displaying as Image Format and also displaying the Corresponding time table of current semester	30-47	

INDEX

S.NO	TITLE	PAGE NO	REMARKS
4.	WEEK 4 : 1 . a college wants to collect student details through an online admission form design a registration form using only html tags(no css) with propern labels table alignment,background color and borders 2 . A Company wants to design a webpage that display two different forms side by side using iframes in which one should display a login form and other should display register form	48-57	
5	WEEK 5 : 1.To create a responsive card layout using HTML and CSS with three cards, each containing an image, title, and button 2. To design a mini website using iframe with navigation links (Home, About, Contact), where clicking on each link displays the respective content within the same page using internal or external CSS. 3. To design an Amrita login page using HTML and CSS that displays different campuses, where clicking on each campus shows its details on the same page	58-73	

INDEX

S.NO	TITLE	PAGE NO	REMARKS
6.	WEEK 6: 1. Design a responsive e-commerce application using the grid layout. This application should contain different categories of products such as electronics, clothing, home-appliances, books Based on the category we select corresponding products should be displayed on the sidebar.	74-81	
7	WEEK 7 : 1. create a github account and create a repository named UID lab with roll no and upload every week experiments and lab report in repository and make repository file	82-85	
8	WEEK 8 : 1.Create a webpage that displays a square that continuously rotates 360° using CSS animation. 2.Create a ball that moves up and down continuously only using CSS animations 3. Create a circle that moves horizontally and rotates at the same time.	86-95	
9.	WEEK 9 :Create an webpage using html, CSS, java script based on the given conditions: a)Create a button for each program, based on the button clicked the corresponding program has to be executed and should display the output. b)The logics for each button includes finding an even numbers, odd number, prime or not, factorial of a number, amstrong or not. c)Use appropriate designs to display the output.	96-105	

S.NO	TITLE	PAGE NO	REMARKS
10.	WEEK 10: (A).Write a Javascript conditional statement to find thesign of product of three numbers using functions.Display an alert box with specified sign. (B).Write a Javascript conditional statement to sort three numbers using functions .Display an alert box to show the result (C).Write a Javascript conditional statement to find the largest of five numbers using function and onclick event.Display an alert box to show result. (D).Write a Javascript for loop that will iterate from 0 to 15.For each iteration, it will check if the current number is odd or even and display a message to the screen. (E).Write a Javascript program to read 6 subjects marks from the student then compute the average marks of the student.Then this average is used to determine corresponding grade	106-143	
11.	WEEK 11 : 1. A website requires users to enter a username. The system should display a message while typing if the username is less than 5 characters. 2. A registration form should validate email format before submission. If invalid, the form should not be submitted. 3. A system should check password strength while typing and display whether it is weak or strong. 4. A form should validate a mobile number when the user leaves the input field. The number must be exactly 10 digits.	144-177	

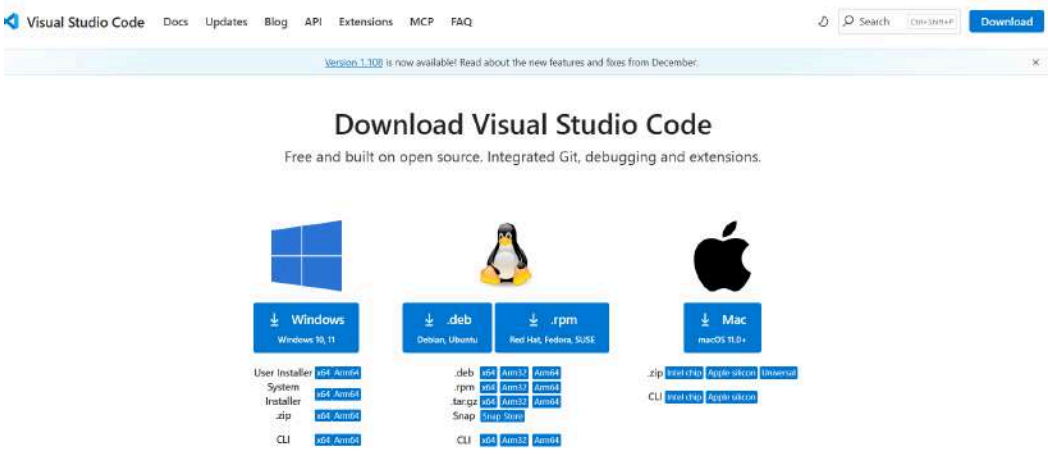
S.NO	TITLE	PAGE NO	REMARKS
11.	WEEK 11: 5. Design an HTML page, such that the registration form must validate: Name (not empty) Email (valid format) Password (minimum 6 characters)		
12.	WEEK 12 : installation and downloading od node js	178-180	

WEEK -1

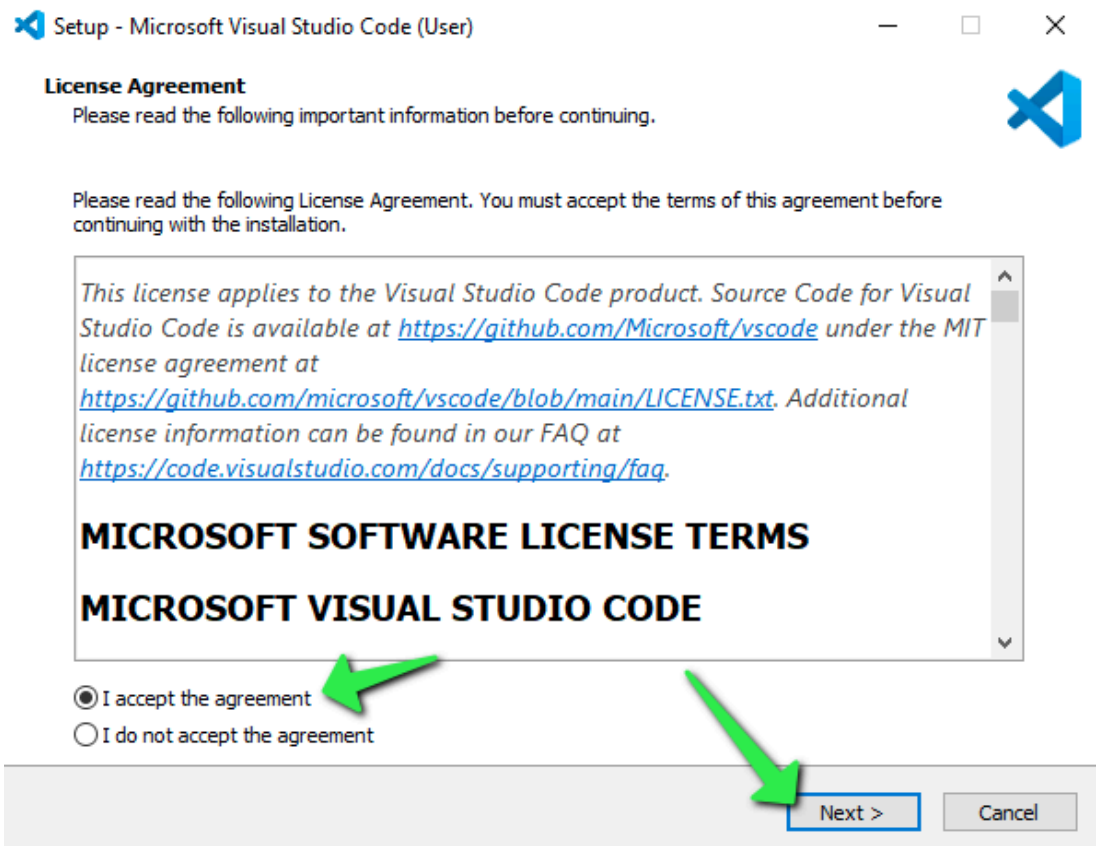
1. <u>AIM</u>: Steps to install VS CODE

<u>Procedure</u> :

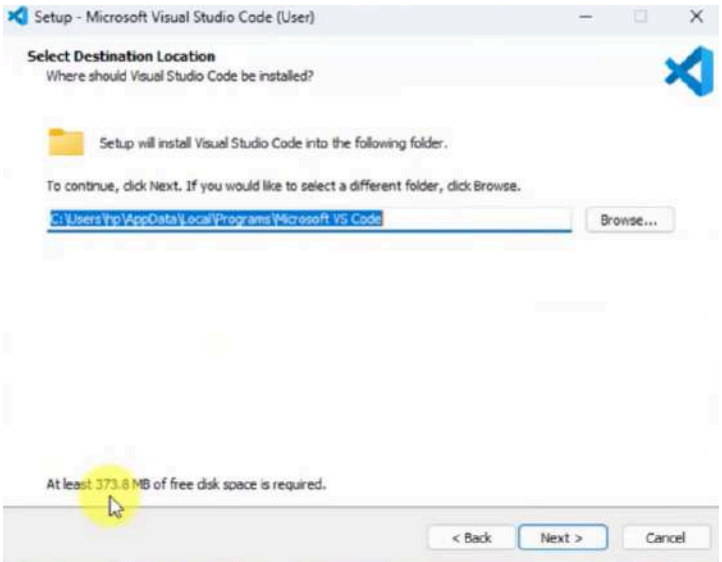
step - 1



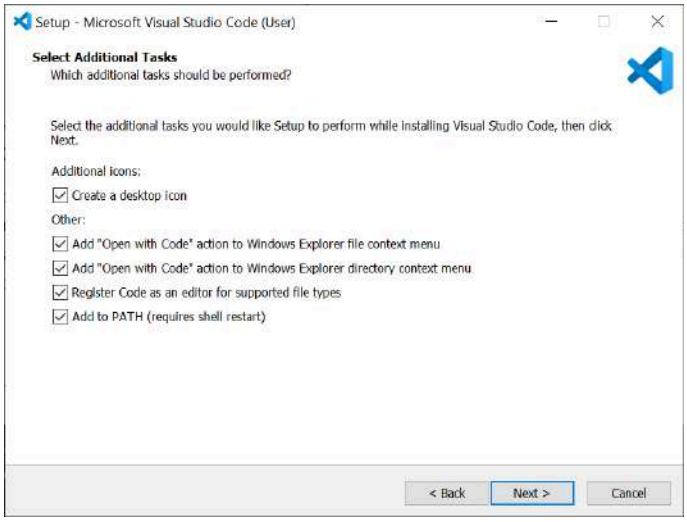
step - 2



step -3

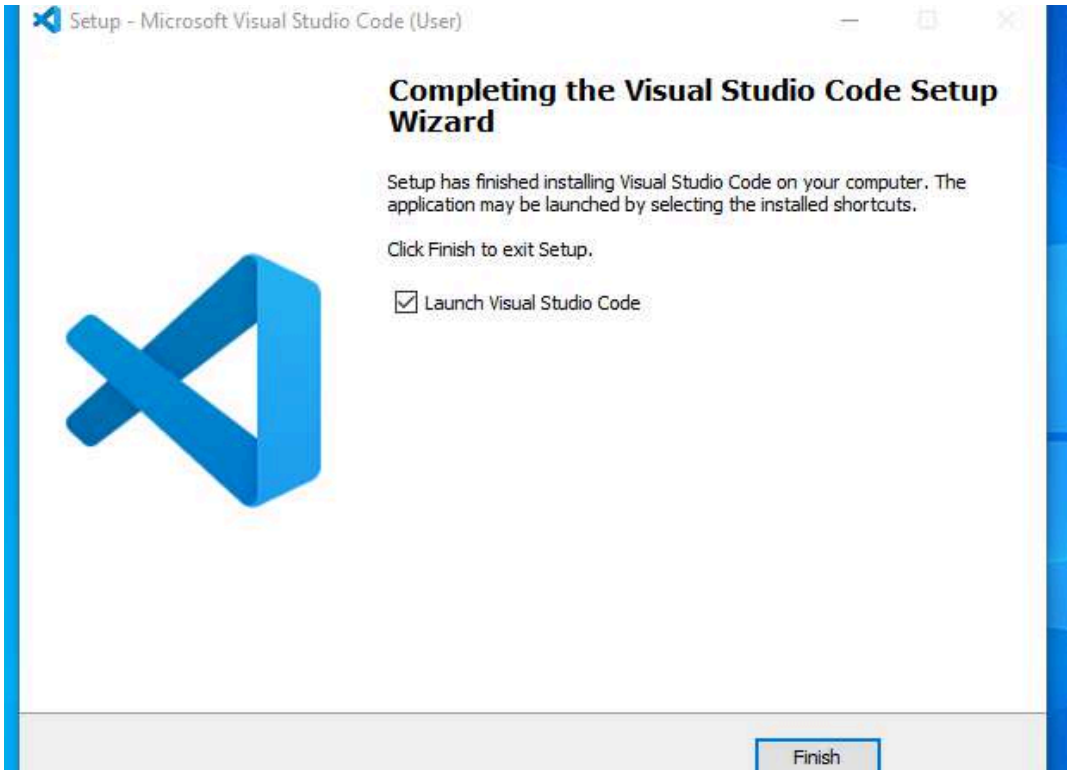


step - 4

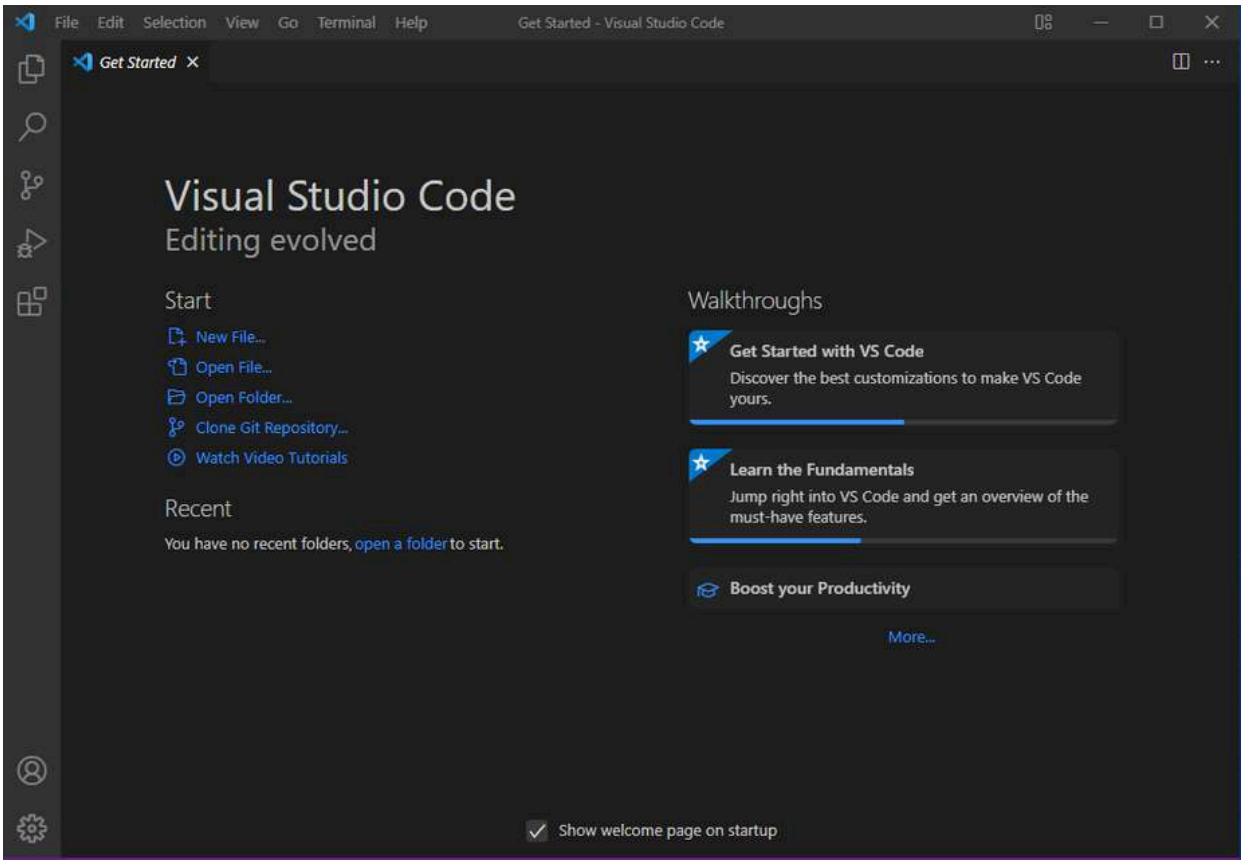


step - 5

step -6


Setup - Microsoft Visual Studio Code (User)
Completing the Visual Studio Code Setup Wizard
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Visual Studio Code
Finish

step -7


File Edit Selection View Go Terminal Help
Get Started - Visual Studio Code
Get Started
Visual Studio Code
Editing evolved
Start
New File...
Open File...
Open Folder...
Clone Git Repository...
Watch Video Tutorials
Recent
You have no recent folders, open a folder to start.
Walkthroughs
Get Started with VS Code
Discover the best customizations to make VS Code yours.
Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.
Boost your Productivity
More...
Show welcome page on startup

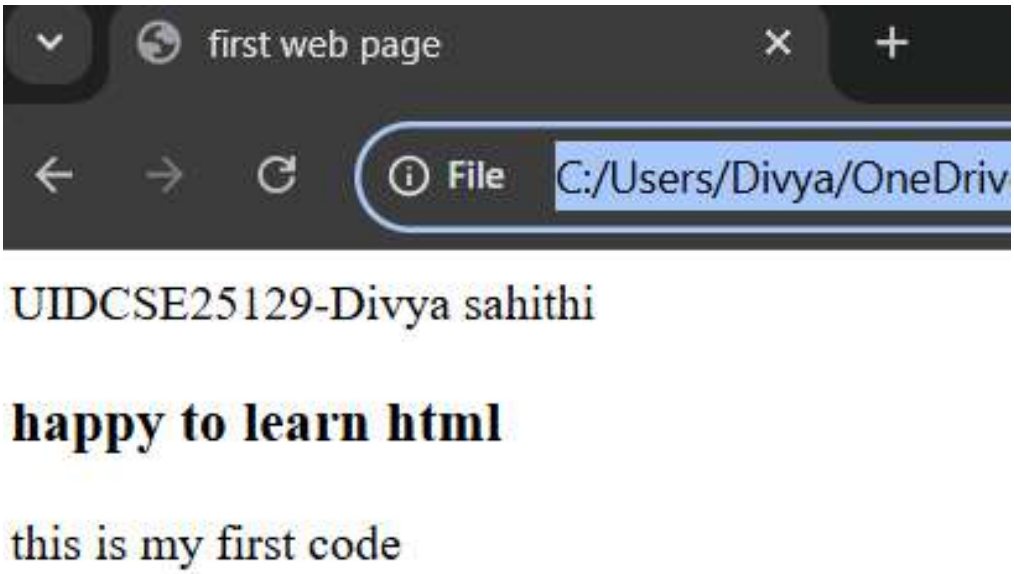
2. Aim : To learn the basics of html(basics.html)

Program:

UIDCSE25129-Divya sahithi

```
<!DOCTYPE html>
<html>
  <head>
    <title>first web page</title>
  </head>

  <body>
    <h3>happy to learn html</h3>
    <p>this is my first code</p>
  </body>
</html>
```

Output:



Explanation of the tags:

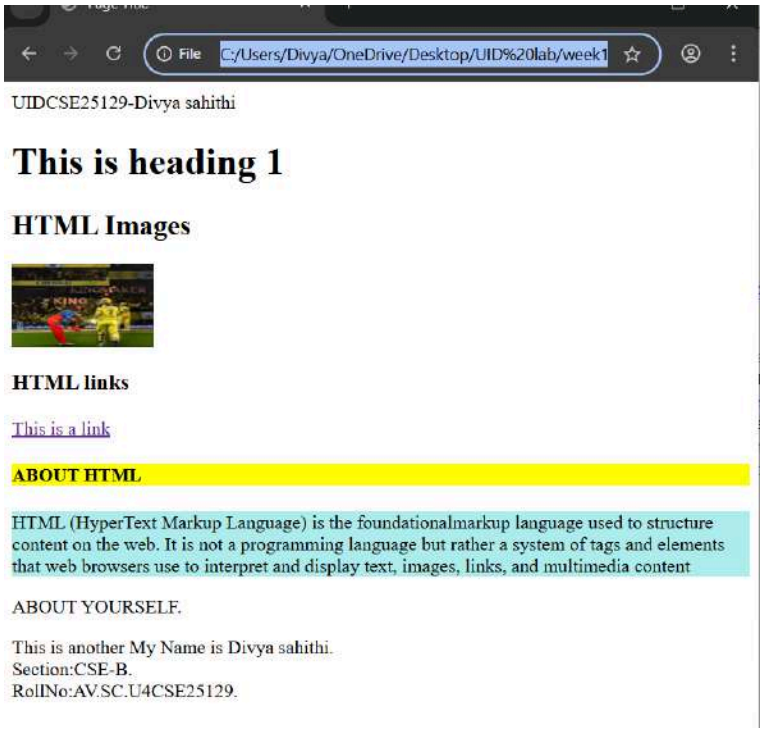
- HTML stands for Hyper Text Markup Language.
- HTML is used to create and structure web pages.
- It tells the web browser what to display and how to display it.
- **<DOCTYPE HTML> :** The <DOCTYPE html>element tells the browser , this is an HTML5 document
- **<html>**: The <html> element is the root element of an HTML page.
- **<head>:** The <head> element contains meta information about the HTML page.
- **<title>:** The <title> element specifies a title for the HTML page.
- **<body>:** The <body>element contains all visible content like text ,images ,links ,tables ,etc.
- **<h3>:** The <h3> element defines a large heading.
- **<p>:** The <p> element defines a paragraph.

3. Aim : understanding basic html tags

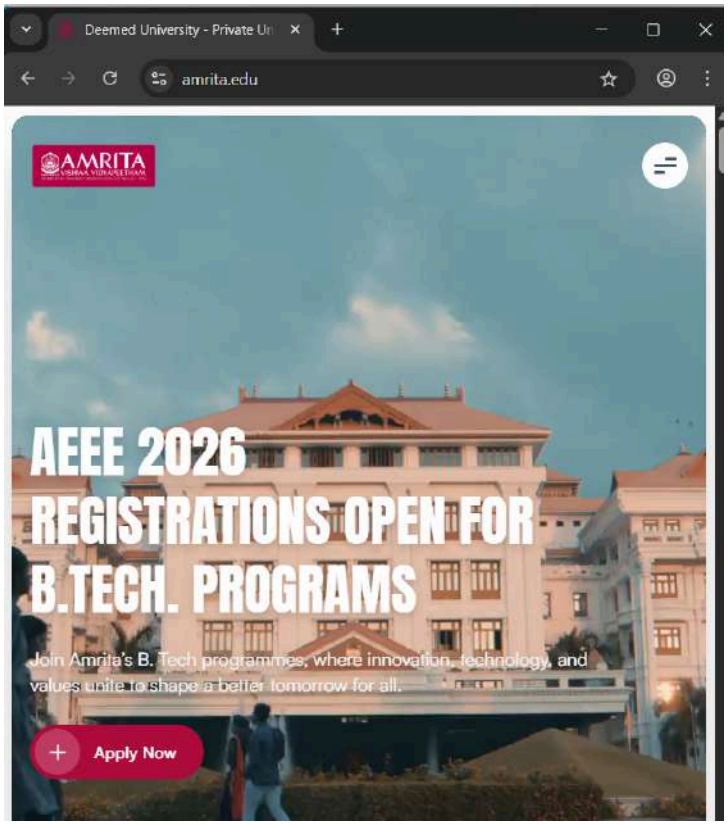
Program:UIDCSE25129-Divya sahithi

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
<body>
   <h1>This is heading 1</h1>
   <h2>HTML Images</h2>
   <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTXCOXM4q-IK0MFwcWErgc34P9crNwz2ZkaPg&s"
   alt="thala" width="120" height="70">
   <h3>HTML links</h3>
   <a href="https://www.amrita.edu">This is a link</a>
   <h4 style="background-color: yellow;">ABOUT HTML</h4>
   <p style="background-color: paleturquoise;">HTML (HyperText Markup Language)
      is the foundationalmarkup language used to structure content on the web.
      It is not a programming language but rather a system of tags and elements
      that web browsers use to interpret and display text, images, links,
      and multimedia content</p>
   <p> ABOUT YOURSELF.<br>
   <p> This is another
      My Name is Divya sahithi.<br> Section:CSE-B.<br>
      RollNo:AV.SC.U4CSE25129.<br>
</body>
</html>
```

Output:



If we click on the link we get amrita website like this



Explanation of the tags:

- **
:** The HTML <br> element defines a line break. it is an empty tag with no end tag
- **Style:** The HTML style attribute is used to add styles to an element such as color, font, size and more
- **<img> :** The <img> tag is a fundamental HTML element used to embed an image into a web page
- **<a> :** The HTML <a> tag ,which stands for “anchor”, is used to define a hyperlink.

WEEK -2

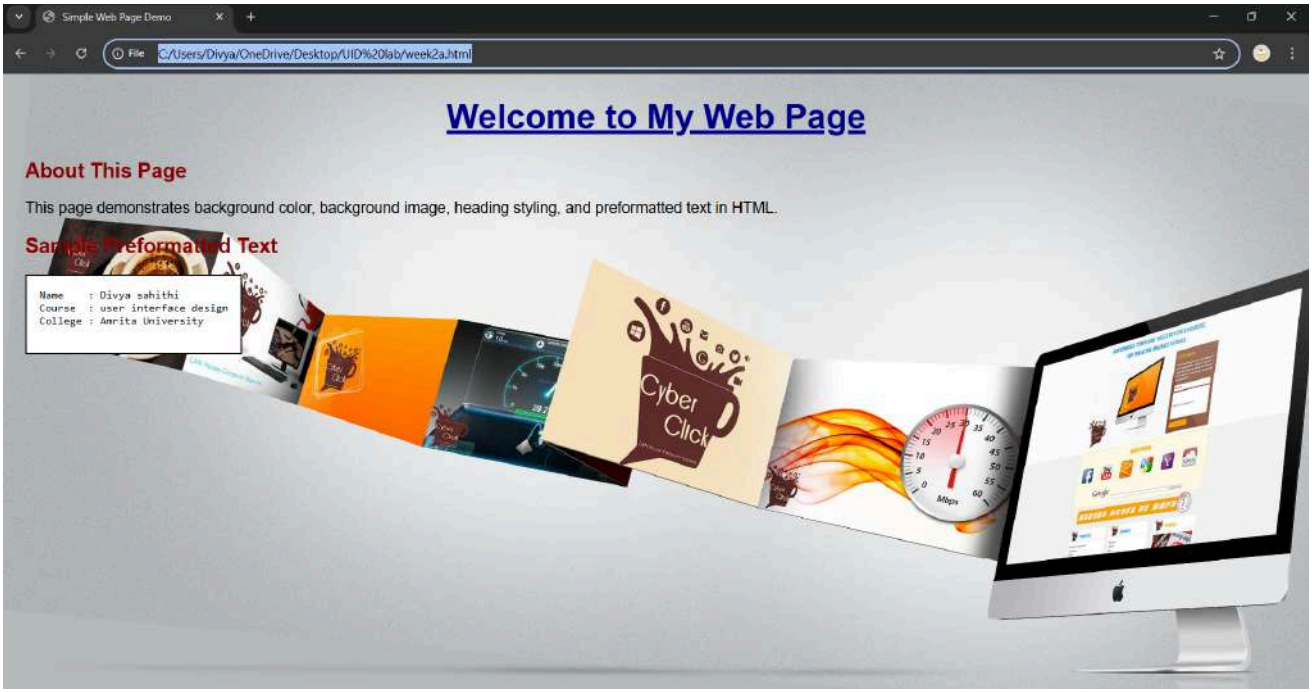
2a.Aim: To design a simple web page using HTML that demonstrates the use of background color, background image, heading styling, and preformatted text

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Simple Web Page Demo</title>
  <style>
    /* Background color and image */
    body {
      background-color: lightblue;  /* Background color */
      background-image: url("https://wallpaperaccess.com/full/2945044.jpg");
      background-repeat: no-repeat;
      background-size: cover;
      background-attachment: fixed;
      font-family: Arial, sans-serif   }
    /* Heading styling */
    h1 {
      color: darkblue;
      text-align: center;
      text-decoration: underline;
      font-size: 40px;     }
```

```
  h2 {   color: darkred;
         margin-left: 20px;}
      /* Paragraph styling */
      p {   font-size: 18px;
         margin-left: 20px;   }
      /* Preformatted text styling */
      pre {
         background-color: white;
         padding: 15px;
         border: 2px solid black;
         width: fit-content;
         margin-left: 20px;}
   </style>
</head>
<body>
   <h1>Welcome to My Web Page</h1>
   <h2>About This Page</h2>
   <p>This page demonstrates background color, background image, heading styling, and preformatted text in HTML.</p>
   <h2>Sample Preformatted Text</h2>
   <pre>
Name    : Divya sahithi
Course  : user interface design
College : Amrita University
   </pre></body></html>
```

Output:



Explanation of the tags:

- h1>, <h2> → Headings
- <p> → Paragraph
- <pre> → Preformatted text

2b.Aim : To create a static webpage using only HTML that displays basic information about Amrita Vishwa Vidyapeetham, including a heading descriptive text, background styling through HTML attributes, and a table showing the list of campuses in different states.

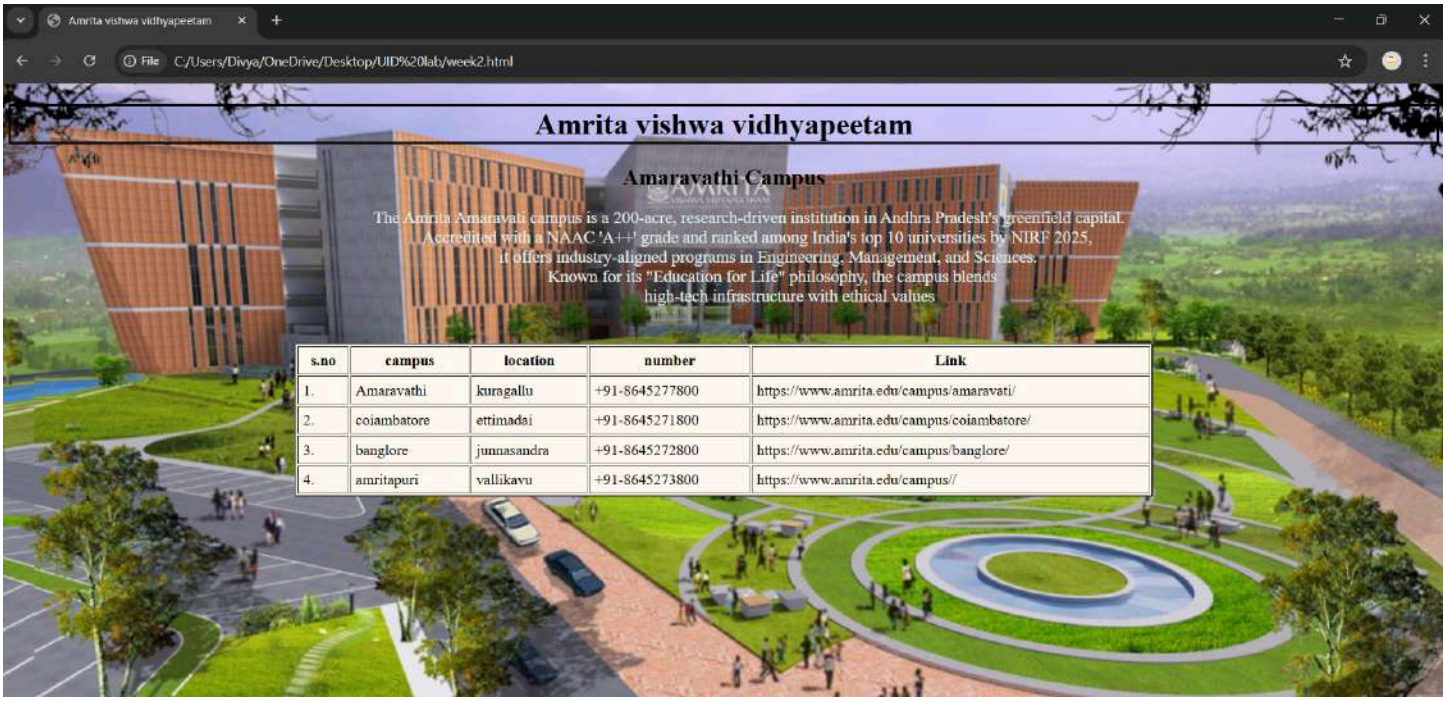
Procedure:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Amrita vishwa vidhyapeetam</title>
  </head>
 <body style="background-color:rgb(255, 244, 230);
  background-image:url(https://webfiles.amrita.edu/2022/03/nbElb5oz-amrita-amaravati-campus-banner-2022.jpg);
   background-size: cover ; background-repeat: no repeat;">
    <h1 style="text-align:center; color:rgb(0, 0, 0);border: 3px solid black;
    font-family: 'Times New Roman', Times, serif;"> Amrita vishwa vidhyapeetam</h1>
    <h2 style="text-align: center; color: black;font-family: 'Times New Roman', Times, serif;">
      Amaravathi Campus </h2>
    <pre style="text-align: center;font-family: 'Times New Roman', Times, serif;color:rgb(226, 239, 237);
    font-size: large;">
       The Amrita Amaravati campus is a 200-acre, research-driven institution in Andhra Pradesh's greenfield capital.
          Accredited with a NAAC 'A++' grade and ranked among India's top 10 universities by NIRF 2025,
            it offers industry-aligned programs in Engineering, Management, and Sciences.
              Known for its "Education for Life" philosophy, the campus blends
                high-tech infrastructure with ethical values
    </pre>
    <table border="2"; align="center"; cellpadding="5";cellspacing="10"; width="60%";
```

```
style="background-color:floralwhite;">
    <tr>
        <th>s.no</th>
        <th>campus</th>
        <th>location</th>
        <th>number</th>
        <th>Link</th>
    </tr>
     <tr>
        <td>1.</td>
        <td >Amaravathi</td>
        <td >kuragallu</td>
        <td>+91-8645277800</td>
        <td >https://www.amrita.edu/campus/amaravati/</td>
    </tr>
    <tr>
        <td>2.</td>
        <td >coiambatore</td>
        <td >ettimadai</td>
        <td>+91-8645271800</td>
        <td >https://www.amrita.edu/campus/coiambatore/</td>
    </tr>
    <tr>
  <td>3.</td>
        <td >banglore</td>
```

```
        <td >junnasandra</td>
        <td>+91-8645272800</td>
        <td >https://www.amrita.edu/campus/banglore/</td>
      </tr>
      <tr>
        <td>4.</td>
        <td >amritapuri</td>
        <td >vallikavu</td>
        <td>+91-8645273800</td>
        <td >https://www.amrita.edu/campus//</td>
      </tr>

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

Output:



s.no	campus	location	number	Link
1.	Amaravathi	kuragallu	+91-8645277800	https://www.amrita.edu/campus/amaravati/
2.	coiambatore	ettimadai	+91-8645271800	https://www.amrita.edu/campus/coiambatore/
3.	banglore	junnasandra	+91-8645272800	https://www.amrita.edu/campus/banglore/
4.	amritapuri	vallikavu	+91-8645273800	https://www.amrita.edu/campus//

Explanation of the tags:

- <h1>, <h2> – Heading tags for main title and subtitle.
- <pre> – Displays text in preformatted style, preserving spaces and line breaks.
- <table> – Creates a table to arrange data in rows and columns.
- Attributes: border, align, cellpadding, cellspacing, width.
- <tr> – Defines a table row.
- <th> – Table heading cell (bold and centered).
- <td> – Table data cell for normal content.
- Inline CSS (style attribute) – Used to set background color, image, font, alignment, and borders.

2cAim: To design a restaurants menu webpage using HTML and displaying menu items in a table format with proper layout and visual styling.

Procedure:

```
<!DOCTYPE html>

<html>

<head>

   <title>paradise restaurant - Luxury Edition</title>

   <link href="https://fonts.googleapis.com/css2?family=Playfair+Display:ital@1&display=swap"
rel="stylesheet">

   <style>

      body {   margin: 0;

         font-family: 'Playfair Display', serif;
         background: url("https://images.unsplash.com/photo-1552566626-52f8b828add9") no-repeat center
center fixed;

         background-size: cover;

         color: gold;}

      .header { background: rgba(0, 0, 0, 0.85);

         text-align: center;

         padding: 35px;

         font-size: 40px;

         letter-spacing: 3px;

         font-style: italic;

         color: gold;

         border-bottom: 3px solid;   }
```

```
.menu-section { width: 85%;
        margin: 40px auto;
        padding: 25px;
        border-radius: 15px;
        background: rgba(0, 0, 0, 0.75);
        box-shadow: 0 8px 25px rgba(0,0,0,0.8);
      h2 { text-align: center;
        font-style: italic;
        font-size: 28px;
        color: gold;
        border-bottom: 2px solid gold;
        padding-bottom: 10px;    }
table {width: 100%;
        border-collapse: collapse;
        margin-top: 20px;
        font-style: italic;}
 th, td {  padding: 14px;
        text-align: center;    }
   th {   color: gold;
        font-size: 20px;
        border-bottom: 2px solid gold;  }
```

```
tr {background: transparent; }
    tr:hover { background: rgba(255, 215, 0, 0.15);
        transition: 0.3s;    }
.price {   color: #ffd700;
           font-weight: bold;  }
  .footer { text-align: center;
           padding: 20px;
           background: rgba(0,0,0,0.9);
           color: gold;
           font-style: italic;
           border-top: 2px solid gold; }
   </style>
</head>
<body>
<div class="header">
   Paradise cusine
</div><div class="menu-section">
   <h2>Signature Starters</h2>
   <table>
      <tr>
        <th>Item</th>
           <th>Description</th>
           <th>Price</th>
      </tr>
```

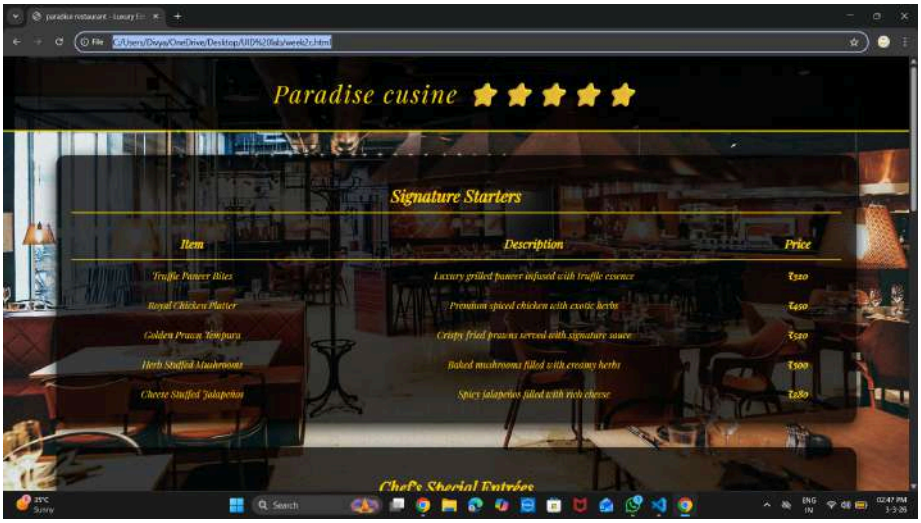
```
<tr>
        <td>Truffle Paneer Bites</td>
        <td>Luxury grilled paneer infused with truffle essence</td>
        <td class="price">₹320</td>
      </tr> <tr>
        <td>Royal Chicken Platter</td>
        <td>Premium spiced chicken with exotic herbs</td>
        <td class="price">₹450</td>
      </tr>  <tr>
        <td>Golden Prawn Tempura</td>
        <td>Crispy fried prawns served with signature sauce</td>
        <td class="price">₹520</td>
      </tr><tr>
        <td>Herb Stuffed Mushrooms</td>
        <td>Baked mushrooms filled with creamy herbs</td>
        <td class="price">₹300</td>
      </tr>  <tr>
        <td>Cheese Stuffed Jalapeños</td>
        <td>Spicy jalapeños filled with rich cheese</td>
        <td class="price">₹280</td>
      </tr>
    </table>
</div>
```

```
<div class="menu-section">
  <h2>Chef’s Special Entrées</h2>
  <table>  <tr>
        <th>Item</th>
        <th>Description</th>
        <th>Price</th> </tr>
     <tr>  <td>Imperial Mutton Biryani</td>
        <td>Slow-cooked aromatic rice with premium mutton</td>
        <td class="price">₹550</td>  </tr>
     <tr>      <td>Golden Cream Pasta</td>
        <td>Rich Alfredo sauce with fine herbs</td>
        <td class="price">₹380</td>  </tr>
     <tr>    <td>Grilled Salmon Steak</td>
        <td>Fresh salmon grilled to perfection</td>
        <td class="price">₹650</td>  </tr>
<tr> <td>Royal Butter Chicken</td>
        <td>Creamy tomato gravy with tender chicken</td>
        <td class="price">₹480</td> </tr>
     <tr>   <td>Luxury Veg Lasagna</td>
        <td>Layered pasta with cheese and vegetables</td>
        <td class="price">₹420</td>  </tr>
  </table></div>
```

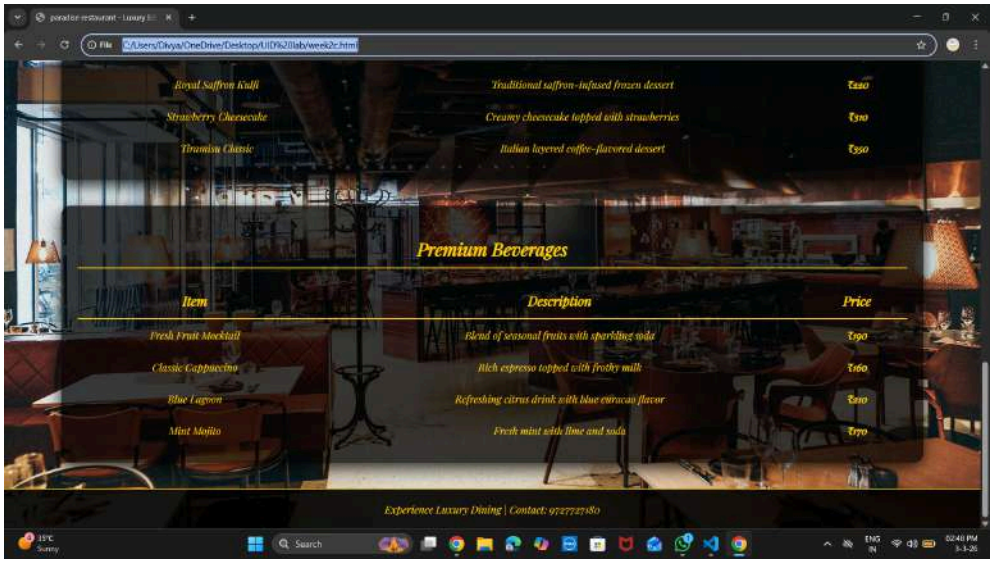
```
div class="menu-section">
  <h2>Luxury Desserts</h2>
  <table>
    <tr>   <th>Item</th>
      <th>Description</th>
      <th>Price</th>
    </tr><tr>
      <td>Belgian Chocolate Delight</td>
      <td>Premium dark chocolate layered dessert</td>
      <td class="price">₹290</td>
    </tr>  <tr>
      <td>Royal Saffron Kulfi</td>
      <td>Traditional saffron-infused frozen dessert</td>
      <td class="price">₹220</td>
    </tr> <tr>
      <td>Strawberry Cheesecake</td>
      <td>Creamy cheesecake topped with strawberries</td>
      <td class="price">₹310</td>
    </tr> <tr>
      <td>Tiramisu Classic</td>
      <td>Italian layered coffee-flavored dessert</td>
      <td class="price">₹350</td>
    </tr></table>
</div>
```

```
div class="menu-section">
   <h2>Premium Beverages</h2>
   <table>  <tr>  <th>Item</th>
         <th>Description</th>
         <th>Price</th></tr>
      <tr>   <td>Fresh Fruit Mocktail</td>
         <td>Blend of seasonal fruits with sparkling soda</td>
         <td class="price">₹190</td>  </tr>
      <tr> <td>Classic Cappuccino</td>
         <td>Rich espresso topped with frothy milk</td>
         <td class="price">₹160</td>
      </tr>  <tr>
         <td>Blue Lagoon</td>
         <td>Refreshing citrus drink with blue curacao flavor</td>
         <td class="price">₹210</td>
      </tr> <tr><td>Mint Mojito</td>
         <td>Fresh mint with lime and soda</td>
         <td class="price">₹170</td></tr>
   </table>
</div><div class="footer">
   Experience Luxury Dining | Contact: 9727727180
</div>
</body>
</html>
```

Output:


Paradise cusine
Signature Starters
Item
Description
Price
Truffle Paneer Bites
Luxury grilled paneer infused with truffle essence
₹320
Royal Chicken Platter
Premium spiced chicken with exotic herbs
₹450
Golden Prawn Tempura
Crispy fried prawns served with signature sauce
₹520
Herb Stuffed Mushrooms
Baked mushrooms filled with creamy herbs
₹300
Cheese Stuffed Jalapenos
Spicy jalapenos filled with rich cheese
₹280
Chef's Special Entrées

Chef's Special Entrées
Item
Description
Price
Imperial Mutton Biryani
Slow-cooked aromatic rice with premium mutton
₹550
Golden Cream Pasta
Rich Alfredo sauce with fine herbs
₹380
Grilled Salmon Steak
Fresh salmon grilled to perfection
₹650
Royal Butter Chicken
Creamy tomato gravy with tender chicken
₹480
Luxury Veg Lasagna
Layered pasta with cheese and vegetables
₹420
Luxury Desserts
Item
Description
Price
Belgian Chocolate Delight
Premium dark chocolate layered dessert
₹290
Royal Saffron Kulfi
Traditional saffron-infused frozen dessert
₹220


Royal Saffron Kulfi
Traditional saffron-infused frozen dessert
₹220
Strawberry Cheesecake
Creamy cheesecake topped with strawberries
₹310
Tiramisu Classic
Italian layered coffee-flavored dessert
₹350
Premium Beverages
Item
Description
Price
Fresh Fruit Mocktail
Blend of seasonal fruits with sparkling soda
₹190
Classic Cappuccino
Rich espresso topped with frothy milk
₹160
Blue Lagoon
Refreshing citrus drink with blue curacao flavor
₹220
Mint Mojito
Fresh mint with lime and soda
₹170
Experience Luxury Dining | Contact: 9727727180

Explanation of the tags:

- background: url(...) → Adds wallpaper
- rgba() → Transparent background
- border-radius → Rounded corners
- box-shadow → Shadow effect
- :hover → Effect when mouse moves over row
- font-style: italic → Italic text

<link> → External resource connection

<style> → Internal CSS

<body> → Visible content

 → Section container

<h2> → Sub heading

<table> → Creates table

<tr> → Table row

<th> → Table heading

<td> → Table data

WEEK -3

1. **Aim:** Construction of Class Timetable using html Elements

Program:

UIDlab-cse25129(Divya)

```
<!DOCTYPE html>
<html>
<head>
<title>Class Time Table</title>
<style>
.vertical {
   writing-mode: vertical-rl;
   transform: rotate(180deg);
   text-align: center;
   font-weight: bold;
   letter-spacing: 2px;
}
</style>
</head>
<body>
<h2 align="center">CLASS TIME TABLE</h2>

<table border="1" cellspacing="0" cellpadding="8" align="center">
```

<tr>

<th>Time / Day</th>

<th>Slot 1
08:55–09:45</th>

<th>Slot 2
09:45–10:35</th>

<th>10:35–10:45</th>

<th>Slot 3
10:45–11:35</th>

<th>Slot 4
11:35–12:25</th>

<th>Slot 5
12:25–01:15</th>

<th>01:15–02:05</th>

<th>Slot 6
02:10–03:00</th>

<th>03:00–03:10</th>

<th>Slot 7
03:10–04:00</th>

<th>Slot 8
04:00–04:50</th>

</tr>

<tr>

<th>Monday</th>

<td colspan="2">23MAT117 Linear Algebra LAB (BYOD)</td>

<td rowspan="5" bgcolor="lightgreen" class="vertical">SHORT BREAK</td>

<td>LIBRARY</td>

<td>23PHY115 Modern Physics</td>

<td>LIBRARY</td>

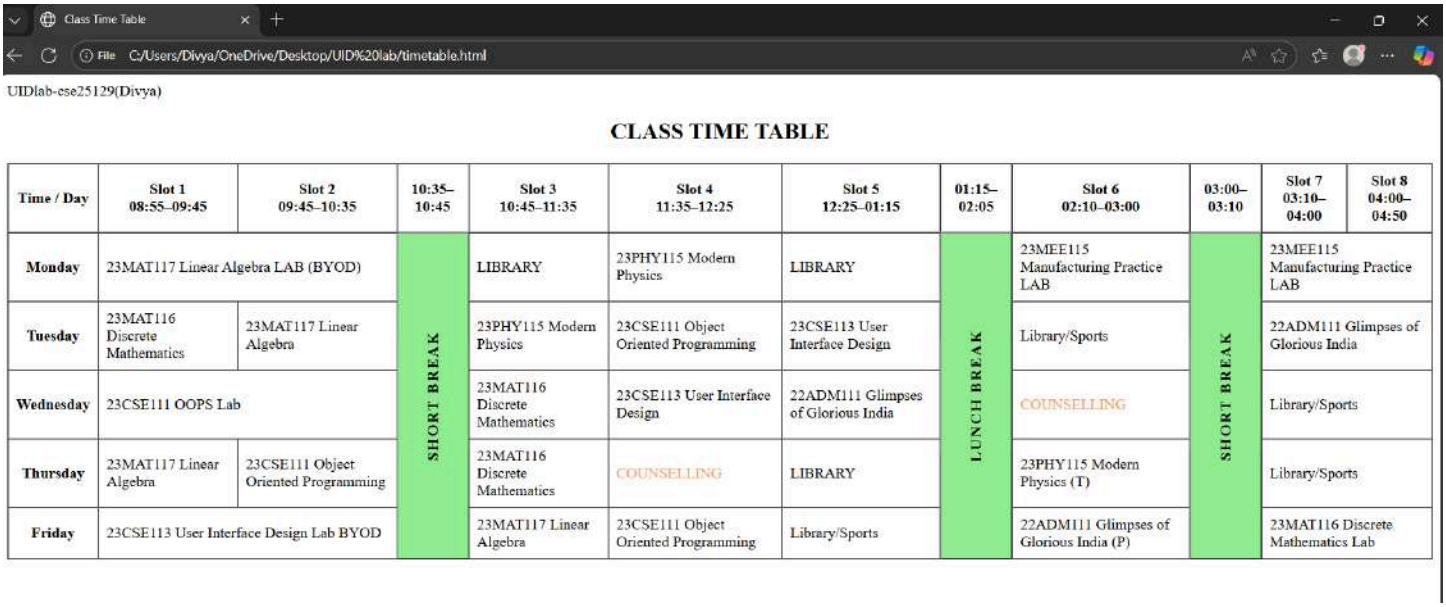
<td rowspan="5" bgcolor="lightgreen" class="vertical">LUNCH BREAK</td>

<td>23MEE115 Manufacturing Practice LAB</td>

```
<td rowspan="5" bgcolor="lightgreen" class="vertical">SHORT BREAK</td>

<td colspan="2">23MEE115 Manufacturing Practice LAB</td>

</tr>

<tr>

<th>Tuesday</th>

<td>23MAT116 Discrete Mathematics</td>

<td>23MAT117 Linear Algebra</td>

<td>23PHY115 Modern Physics</td>

<td>23CSE111 Object Oriented Programming</td>

<td>23CSE113 User Interface Design</td>

<td>Library/Sports</td>

<td colspan="2">22ADM111 Glimpses of Glorious India</td>

</tr>

<tr>

<th>Wednesday</th>

<td colspan="2">23CSE111 OOPS Lab</td>

<td>23MAT116 Discrete Mathematics</td>

<td>23CSE113 User Interface Design</td>

<td>22ADM111 Glimpses of Glorious India</td>

<td style="color:sandybrown;">COUNSELLING</td>

<td colspan="2">Library/Sports</td>

</tr>
```

```
<tr>
<th>Thursday</th>
<td>23MAT117 Linear Algebra</td>
<td>23CSE111 Object Oriented Programming</td>
<td>23MAT116 Discrete Mathematics</td>
<td style="color:sandybrown;">COUNSELLING</td>
<td>LIBRARY</td>
<td>23PHY115 Modern Physics (T)</td>
<td colspan="2">Library/Sports</td>
</tr>
<tr>
<th>Friday</th>
<td colspan="2">23CSE113 User Interface Design Lab BYOD</td>
<td>23MAT117 Linear Algebra</td>
<td>23CSE111 Object Oriented Programming</td>
<td>Library/Sports</td>
<td>22ADM111 Glimpses of Glorious India (P)</td>
<td colspan="2">23MAT116 Discrete Mathematics Lab</td>
</tr>
</table>
</body>
</html>
```

Output:



Class Time Table

File C:/Users/Divya/OneDrive/Desktop/UID%20lab/timetable.html

UIDlab-ese25129(Divya)

CLASS TIME TABLE

Time / Day	Slot 1 08:55–09:45	Slot 2 09:45–10:35	10:35–10:45	Slot 3 10:45–11:35	Slot 4 11:35–12:25	Slot 5 12:25–01:15	01:15–02:05	Slot 6 02:10–03:00	03:00–03:10	Slot 7 03:10–04:00	Slot 8 04:00–04:50
Monday	23MAT117 Linear Algebra LAB (BYOD)		SHORT BREAK	LIBRARY	23PHY115 Modern Physics	LIBRARY	LUNCH BREAK	23MEE115 Manufacturing Practice LAB	SHORT BREAK	23MEE115 Manufacturing Practice LAB	
Tuesday	23MAT116 Discrete Mathematics	23MAT117 Linear Algebra		23PHY115 Modern Physics	23CSE111 Object Oriented Programming	23CSE113 User Interface Design		Library/Sports		22ADM111 Glimpses of Glorious India	
Wednesday	23CSE111 OOPS Lab			23MAT116 Discrete Mathematics	23CSE113 User Interface Design	22ADM111 Glimpses of Glorious India		COUNSELLING		Library/Sports	
Thursday	23MAT117 Linear Algebra	23CSE111 Object Oriented Programming		23MAT116 Discrete Mathematics	COUNSELLING	LIBRARY		23PHY115 Modern Physics (T)		Library/Sports	
Friday	23CSE113 User Interface Design Lab BYOD			23MAT117 Linear Algebra	23CSE111 Object Oriented Programming	Library/Sports		22ADM111 Glimpses of Glorious India (P)		23MAT116 Discrete Mathematics Lab	

Explanation of the tags:

- <table> : is used to create the timetable structure with borders and spacing.
- <tr> :defines table rows for each day.
- <th> :is used for headings like time slots and days.
- <td> : is used to enter subjects and activities.
- colspan :merges columns for lab hours.
- rowspan : merges rows for short break and lunch break.
- bgcolor :gives background color to break cells.
- class="vertical" : applies vertical text style.
- style attribute changes text color for specific cells.

2 . Aim: Construction of web page Displaying Each Syllabus of each Subject Displaying as Image Format and also displaying the Corresponding time table of current semester

Program:

UIDlab-CSE25129(divya)

```
<!DOCTYPE html>
<html>
<head>
  <title>Syllabus, Faculty & Master Time Table</title>
  <style>
    body{
      font-family: Arial;
      background-color: #f2f6ff;
      text-align: center;
    }
    h2{
      margin: 25px 0 15px 0;
    }
    /* Gallery */
    .gallery{
      display: grid;
      grid-template-columns: 1fr 1fr ;
```

```
gap: 25px;
        width: 90%;
        margin: auto;
    }
    .card{
        background-color: white;
        padding: 10px;
        border-radius: 10px;
        border: 2px solid black;
    }
    .card img{
        width: 100%;
        border-radius: 8px;
    }
    .title{
        font-weight: bold;
        font-size: 18px;
        margin: 8px 0;
    }
    .faculty{
        color: #333;
        margin-bottom: 8px;
    }
```

```
/* Tables */
    table{
        border-collapse: collapse;
        margin: 30px auto;
        background-color: white;
        width: 95%;
    }
    th{
        background-color: #dbe4f2;
    }
    td, th{
        border: 1px solid black;
        padding: 8px;
        text-align: center;
    }
    .break{
        background-color: yellow;
        font-weight: bold;
    }
    .vertical {
       writing-mode: vertical-rl;
       transform: rotate(180deg);
       text-align: center;
       font-weight: bold;
        letter-spacing: 2px;
    }
</style>
```

```
</head>

<body>

<h2>SUBJECT SYLLABUS & FACULTY</h2>

<div class="gallery">

  <div class="card">

    <div class="title">23MAT116 – Discrete Mathematics</div>

    <div class="faculty">Faculty: Dr. Srivasa Rao Thota</div>

    <img src="C:\Users\Divya\OneDrive\Desktop\UID lab\syllabusproject\img1.png.png">

  </div>


  <div class="card">

    <div class="title">23MAT117 – Linear Algebra</div>

    <div class="faculty">Faculty: Mr. Sapavat Bixapathi</div>

    <img src="C:\Users\Divya\OneDrive\Desktop\UID lab\syllabusproject\img2.png.png">

  </div>

  <div class="card">

    <div class="title">23CSE111 – Object Oriented Programming</div>

    <div class="faculty">Faculty: Dr. Balaji TK</div>

    <img src="C:\Users\Divya\OneDrive\Desktop\UID lab\syllabusproject\img3.png.png">

  </div>

  <div class="card">

    <div class="title">23PHY115 – Modern Physics</div>

    <div class="faculty">Faculty: Dr. C R Kesavalu</div>

    <img src="C:\Users\Divya\OneDrive\Desktop\UID lab\syllabusproject\img4.png.png">

  </div>

  <div class="card">

    <div class="title">23CSE113 – User Interface Design</div>
```

```
    <div class="faculty">Faculty: Dr. Rajkumar Batchu<br>Assisting: Mr. Barge Bharadwaj</div>
    <img src="C:\Users\Divya\OneDrive\Desktop\UID lab\syllabusproject\img5.png.png">
  </div>
  <div class="card">
    <div class="title">23MEE115 – Manufacturing Practice Lab</div>
    <div class="faculty">Faculty: Dr. Sathies S.</div>
    <img src="C:\Users\Divya\OneDrive\Desktop\UID lab\syllabusproject\img6.png.png">
  </div>
</div>
<h2 align="center">Master Time Table CSE II Semester B Section</h2>
<table border="1" cellspacing="0" cellpadding="8" align="center">
<tr>
<th>Time / Day</th>
<th>Slot 1<br>08:55–09:45</th>
<th>Slot 2<br>09:45–10:35</th>
<th>10:35–10:45</th>
<th>Slot 3<br>10:45–11:35</th>
<th>Slot 4<br>11:35–12:25</th>
<th>Slot 5<br>12:25–01:15</th>
<th>01:15–02:05</th>
<th>Slot 6<br>02:10–03:00</th>
<th>03:00–03:10</th>
<th>Slot 7<br>03:10–04:00</th>
<th>Slot 8<br>04:00–04:50</th>
</tr>
```

```
<tr>
<th>Monday</th>
<td colspan="2">23MAT117 Linear Algebra LAB (BYOD)</td>
<td rowspan="5" bgcolor="lightgreen" class="vertical">SHORT BREAK</td>
<td>LIBRARY</td>
<td>23PHY115 Modern Physics</td>
<td>LIBRARY</td>
<td rowspan="5" bgcolor="lightgreen" class="vertical">LUNCH BREAK</td>
<td>23MEE115 Manufacturing Practice LAB</td>
<td rowspan="5" bgcolor="lightgreen" class="vertical">SHORT BREAK</td>
<td colspan="2">23MEE115 Manufacturing Practice LAB</td>
</tr>
<tr>
<th>Tuesday</th>
<td>23MAT116 Discrete Mathematics</td>
<td>23MAT117 Linear Algebra</td>
<td>23PHY115 Modern Physics</td>
<td>23CSE111 Object Oriented Programming</td>
<td>23CSE113 User Interface Design</td>
<td>Library/Sports</td>
<td colspan="2">22ADM111 Glimpses of Glorious India</td>
</tr>
```

```
<th>Wednesday</th>

<td colspan="2">23CSE111 OOPS Lab</td>

<td>23MAT116 Discrete Mathematics</td>

<td>23CSE113 User Interface Design</td>

<td>22ADM111 Glimpses of Glorious India</td>

<td style="color:sandybrown;">COUNSELLING</td>

<td colspan="2">Library/Sports</td>

</tr>

<tr>

<th>Thursday</th>

<td>23MAT117 Linear Algebra</td>

<td>23CSE111 Object Oriented Programming</td>

<td>23MAT116 Discrete Mathematics</td>

<td style="color:sandybrown;">COUNSELLING</td>

<td>LIBRARY</td>

<td>23PHY115 Modern Physics (T)</td>

<td colspan="2">Library/Sports</td>

</tr>

<tr>

<th>Friday</th>

<td colspan="2">23CSE113 User Interface Design Lab BYOD</td>

<td>23MAT117 Linear Algebra</td>

<td>23CSE111 Object Oriented Programming</td>

<td>Library/Sports</td>

<td>22ADM111 Glimpses of Glorious India (P)</td>

<td colspan="2">23MAT116 Discrete Mathematics Lab</td>

</tr>
```

<tr>

<th colspan="1">Course Code</th>

<th colspan="3">Course Name</th>

<th>L-T-P</th>

<th>Credits</th>

<th colspan="3">Faculty Name</th>

<th colspan="3">Assisting Faculty</th>

</tr>

<tr>

<td colspan="1">23MAT116</td>

<td colspan="3">Discrete Mathematics</td>

<td>3-0-2</td>

<td>4</td>

<td colspan="3">Dr. Srivasa Rao Thota</td>

<td colspan="3"></td>

</tr>

<tr>

<td colspan="1">23MAT117</td>

<td colspan="3">Linear Algebra</td>

<td>3-0-2</td>

<td>4</td>

<td colspan="3">Mr. Sapavat Bixapathi</td>

<td colspan="3"></td>

</tr>

<tr>

<td colspan="1">23CSE111</td>

<td colspan="3">Object Oriented Programming</td>

<td>3-0-2</td>

<td>4</td>

<td colspan="3">Dr. Balaji TK</td>

<td colspan="3" ></td>

</tr>

<tr>

<td colspan="1">23PHY115</td>

<td colspan="3">Modern Physics</td>

<td>2-1-0</td>

<td>3</td>

<td colspan="3">Dr. C R Kesavalu</td>

<td colspan="3"></td>

</tr>

<tr>

<td colspan="1">23CSE113</td>

<td colspan="3">User Interface Design</td>

<td>2-0-2</td>

<td>3</td>

<td colspan="3">Dr. Rajkumar Batchu</td>

<td colspan="3">Mr. Barge Bharadwaj</td>

</tr>

<tr>

<td colspan="1">23MEE115</td>

<td colspan="3">Manufacturing Practice</td>

<td>0-0-3</td>

<td>1</td>

<td colspan="3">Dr. Sathies S.</td>

<td colspan="3"></td>

</tr>

<tr>

<td colspan="1">22ADM111</td>

<td colspan="3">Glimpses of Glorious India</td>

<td>2-0-1</td>

<td>2</td>

<td colspan="3">Dr. Siva Panuganti</td>

<td colspan="3"></td>

</tr>

</table>

</body>

</html>

Output:

Syllabus, Faculty & Master Time

File C:/Users/Divya/OneDrive/Desktop/UID%20lab/syllabus.html

UIDlab-CSE25129(divya)

SUBJECT SYLLABUS & FACULTY

23MAT116 – Discrete Mathematics

Faculty: Dr. Srivasa Rao Thota

Syllabus

Unit 1
Logic, Mathematical Reasoning and Counting: Logic, Prepositional Equivalence, Predicate and Quantifiers, Theorem Proving, Recursive Definitions, Recursive Algorithms, Basics of Counting, Pigeonhole Principle, Permutation and Combinations.

Unit 2
Relations and Their Properties: Representing Relations, Closure of Relations, Partial Ordering, Equivalence Relations and partitions.
Advanced Counting Techniques and Relations: Recurrence Relations, Solving Recurrence Relations, Generating Functions, Solutions of Homogeneous Recurrence Relations, Divide and Conquer Relations, Inclusion-Exclusion.

Unit 3
Graph Theory: Graphs and Sub graphs, Isomorphism, matrices associated with graphs, degrees, walks, connected graphs, shortest path algorithm, Euler and Hamilton Graphs: Euler graphs, Euler's theorem, Fleury's algorithm for Eulerian trails, Hamilton cycles, Chinese-postman problem, approximate solutions of travelling salesman problem, Closest neighbour algorithm.

Lab Practice Problems: Verifications of logical statements, truth table, tautology, Recursive algorithms, Graph problems, degree, shortest path algorithm, Euler's algorithm and closest neighbour algorithm.

Textbook(s)
Kenneth H. Rosen, "Discrete Mathematics and its Applications", Tata McGraw- Hill Publishing Company Limited, New Delhi, Sixth Edition, 2007.
James Strayer, "Elementary Number Theory", Waveland Press, 2002.

Reference(s)
R.P. Grimaldi, "Discrete and Combinatorial Mathematics", Pearson Education, Fifth Edition, 2007.
Thomas Koshy, "Discrete Mathematics with Applications", Academic Press, 2005.
Liu, "Elements of Discrete Mathematics", Tata McGraw- Hill Publishing Company Limited , 2004.

23MAT117 – Linear Algebra

Faculty: Mr. Sapavat Bixapathi

Syllabus

Unit 1
Vector Spaces: Vector spaces - Sub spaces - Linear Independence - Basis – Dimension; Inner Product Spaces: Inner products - Orthogonality - Orthogonal basis - Gram Schmidt Process - Change of basis - Orthogonal complements - Projection on subspace - Least Square Principle. QR- Decomposition.

Unit 2
Linear Transformations: Linear transformation - Relation between matrices and linear transformations - Kernel and range of a linear transformation - Change of basis - Nilpotent transformations. Symmetric and Skew Symmetric Matrices, Adjoint and Hermitian Adjoint of a Matrix, Hermitian, Unitary and Normal Transformations, Self-Adjoint and Normal Transformations.

Unit 3
Eigen values and Eigen vectors: Eigen Values and Eigen Vectors, Diagonalization, Orthogonal Diagonalization, Quadratic Forms, Diagonalizing Quadratic Forms, Conic Sections. Similarity of linear transformations - Diagonalization and its applications - Jordan form and rational canonical form. Case Studies: Applications on least square and image transformations.

Textbook(s)
Howard Anton and Chris Rorres, "Elementary Linear Algebra", Tenth Edition, John Wiley & Sons, 2010.

Reference(s)
Nabil Nassif, Jocelyne Erhel, Bernard Philippe, "Introduction to Computational Linear Algebra", CRC press, 2015.
Sheldon Axler, "Linear Algebra Done Right", Springer, 2014.
Gilbert Strang, "Linear Algebra for Learning Data", Cambridge press, 2019.
Kenneth Hoffmann and Ray Kunze, "Linear Algebra", Second Edition, Prentice Hall, 1971.
Mike Cohen, "Practical Linear Algebra for Data Science", Oreilly Publisher, 2022.

23CSE111 – Object Oriented Programming

Faculty: Dr. Balaji TK

23PHY115 – Modern Physics

Faculty: Dr. C R Kesavalu

Syllabus, Faculty & Master Time

File C:/Users/Divya/OneDrive/Desktop/UID%20lab/syllabus.html

Faculty: Dr. Balaji TK

Syllabus
Unit1
Structured to Object Oriented Approach by Examples - Object Oriented languages - Properties of Object Oriented system – UML and Object Oriented Software Development - Use case diagrams and documents as a functional model - Identifying Objects and classes - Representation of Objects and its state by Object Diagram - Simple Class using class diagram – Encapsulation - Data Hiding - Reading and Writing Objects - Class Level and Instance Level Attributes and Methods- JIVE environment for debugging.

Unit2
Aggregation and Composition using Class Diagram – Generalization using Class Diagram – Inheritance - Constructor and Over Riding – Visibility – Attribute – Parameter – Package - Local and Global - Polymorphism – Overloading - Abstract Classes and Interfaces.

Unit3
Exception Handling - Inner Classes - Wrapper classes – String - and String Builder classes – Number – Math – Random - Array methods - File Streams - Serialization - Generics - Collection framework - Comparator and Comparable - Vector and Arraylist - Iterator and Iterable.

Textbook(s)
Y.Daniel Liang, "Introduction to Java Programming", Tenth Edition, PHI, 2013.
Grady Booch and Robert A. Maksimchuk, "Object-oriented Analysis and Design with Applications", Third Edition, Pearson Education, 2009.

Reference(s)
Naughton P. and Schildt H., "Java2 Complete Reference", Eighth Edition, Tata McGraw- Hill, 2011.
Ali Bahrami, "Object Oriented Systems Development", Second Edition, McGraw-Hill,2008.
Jaime Niño, Fredrick A Hosck, "An Introduction to Programming and Object Oriented Design using Java", Wiley India Private Limited, 2010.

Faculty: Dr. C R Kesavalu

Syllabus

Unit 1
Origin of quantum theory of radiation: Black body radiation, photo-electric effect, Compton Effect – pair production and annihilation, De-Broglie hypothesis, description of waves and wave packets,group velocities. Evidence for wave nature of particles: Davisson-Germer experiment, Heisenberguncertainty principle.

Unit 2
Atomic structure: Historical Development of atomic structures: Thomson's Model, Rutherford's Model: Scattering formula and its predictions, Atomic spectra - Bohr's Model, Sommerfield's Model, The correspondence principle, nuclear motion, and atomic excitation, Application: Lasers.

Unit 3
Quantum mechanics: Wave function, Probability density, expectation values - Schrodinger equation – time dependent and independent, Linearity and superposition, expectation values, operators, Eigen functions and Eigen values

Unit 4
Application of 1D Schrodinger Wave equation: Free particle, Particle in a box, Finite potential well,Tunnel effect, Harmonic oscillator.

Unit 5
Intro to Quantum computing- Q bits- II Quantum correlations: Bell inequalities and entanglement,Schmidt decomposition, super dense coding, teleportation. Module

23CSE113 – User Interface Design

Faculty: Dr. Rajkumar Batchu
Assisting: Mr. Barge Bharadwaj

Syllabus

Unit 1
Introduction to Internet and Web design – Working of Web – Roles of Front-end, Back-end and Full stack development – Web Server – Internet protocols – Web Hosting – HTML – HTML Tags – Create a simple Web Page – Marking up Text – Adding Links – Adding Images – SVG – Table Markup – Embedded Media - Web Based Forms – HTML Forms - CSS for Presentation - Basic Style Sheet - A CSS Style Primer - Using Style Classes - Using Style IDs - Formatting Text - Advanced Typography with CSS3 - Working with Margins, Padding, Alignment, and Floating. Responsive web design. Introduction to version control systems.

Unit 2

23MEE115 – Manufacturing Practice Lab

Faculty: Dr. Sathies S.

Syllabus

1. **Additive Manufacturing Laboratory –12 hours**

Introduction to digital manufacturing. Introduction to Additive Manufacturing - types – additive manufacturing applications - Materials for 3D printing, CAD Modelling for Additive manufacturing, Slicing and STL file generation- G code generation - 3D printing of simple geometries.

2. **Mechanical Engineering Laboratory –12 hours**

Study of tools and equipment used for basic manufacturing processes.
Manual arc welding practice for making Butt and Lap joints - Soldering Practice
Introduction to Machine Tools and Machining Processes

3. **Automation lab –12 hours**

Design, simulate and test pneumatic and electro-pneumatic circuits. Introduction to PLC –PLC programming for

Reference(s)

Naughton P. and Schildt H., "Java2 Complete Reference", Eighth Edition, Tata McGraw- Hill, 2011.
Ali Bahrami, "Object Oriented Systems Development", Second Edition, McGraw-Hill 2008.
Jaime Niño, Frederick A Hosch, "An Introduction to Programming and Object Oriented Design using Java", Wiley India Private Limited, 2010.

Intro to Quanium computing- Q bits- II Quantum correlations: Bell inequalities and entanglement,Schmidt decomposition, super dense coding, teleportation. Module

23CSE113 – User Interface Design

Faculty: Dr. Rajkumar Batchu
Assisting: Mr. Barge Bharadwaj

Syllabus

Unit 1
Introduction to Internet and Web design – Working of Web – Roles of Front-end, Back-end and Full stack development – Web Server – Internet protocols – Web Hosting – HTML – HTML Tags – Create a simple Web Page – Marking up Text – Adding Links – Adding Images – SVG – Table Markup – Embedded Media - Web Based Forms – HTML Forms - CSS for Presentation - Basic Style Sheet - A CSS Style Primer - Using Style Classes - Using Style IDs - Formatting Text - Advanced Typography with CSS3 - Working with Margins, Padding, Alignment, and Floating, Responsive web design, Introduction to version control systems.

Unit 2
CSS Box Model and Positioning - Creating Layouts Using Modern CSS - Backgrounds and Borders - CSS Transformations and Transitions - Animating with CSS and the Canvas - Dynamic Web Pages – Web Scripting - JavaScript programming for Web Applications – Including JavaScript in HTML – HTML5 Form Controls and Validation - Document Object Model - DOM manipulation and events - Event Handlers

Unit 3
Overview of UI Frameworks – User Interface Design Concepts in Web Development – Accessibility – Introduction to Electron - Overview of Electron - Setting up an Electron project - Creating a basic desktop application with Electron - Object communication - Introduction to other popular frameworks and libraries (e.g., JavaFX, WPF, GTK).

Textbook(s)
Jennifer Kyrnin, Julie Meloni, "Sams Teach Yourself HTML, CSS, and JavaScript All in One", Third Edition, Pearson Education, Inc., 2019.

Reference(s)
Jennifer Niederst Robbin, "LEARNING WEB DESIGN A BEGINNER'S GUIDE TO HTML, CSS, JAVASCRIPT, AND WEB GRAPHICS", Fifth Edition, O'Reilly Media, Inc., 2018.
Jessica Minnick, "Responsive Web Design with HTML 5 and CSS", 9th Edition, Cengage Learning, Inc., 2021.
Electron JS.
https://www.electronjs.org/docs/latest
https://www.electronforge.io/

23MEE115 – Manufacturing Practice Lab

Faculty: Dr. Sathies S.

Syllabus

1. **Additive Manufacturing Laboratory –12 hours**
Introduction to digital manufacturing. Introduction to Additive Manufacturing - types – additive manufacturing applications - Materials for 3D printing. CAD Modelling for Additive manufacturing. Slicing and STL file generation- G code generation - 3D printing of simple geometries.
2. **Mechanical Engineering Laboratory –12 hours**
Study of tools and equipment used for basic manufacturing processes.
Manual arc welding practice for making Butt and Lap joints - Soldering Practice
Introduction to Machine Tools and Machining Processes
3. **Automation lab –12 hours**
Design, simulate and test pneumatic and electro-pneumatic circuits. Introduction to PLC –PLC programming for automation applications.
4. **Automobile Engineering lab –9 hours**
Overview of automobiles – components –functioning of various sub-systems; Power train, steering system, suspension system and braking system. Introduction to electric vehicles, hybrid vehicles, alternate fuels. Introduction to E Mobility.

Reference(s)
Lab Manual

Master Time Table CSE II Semester B Section

Time / Day	Slot 1 08:55–09:45	Slot 2 09:45–10:35	10:35–10:45	Slot 3 10:45–11:35	Slot 4 11:35–12:25	Slot 5 12:25–01:15	01:15–02:05	Slot 6 02:10–03:00	03:00–03:10	03:10–04:00	04:00–04:50
Monday	23MAT117 Linear Algebra LAB (BYOD)		SHORT BREAK	LIBRARY	23PHY115 Modern Physics	LIBRARY	LUNCH BREAK	23MEE115 Manufacturing Practice LAB	SHORT BREAK	23MEE115 Manufacturing Practice LAB	
Tuesday	23MAT116 Discrete Mathematics	23MAT117 Linear Algebra		23PHY115 Modern Physics	23CSE111 Object Oriented Programming	23CSE113 User Interface Design		Library/Sports		22ADM111 Glimpses of Glorious India	
Wednesday	23CSE111 OOPS Lab			23MAT116 Discrete Mathematics	23CSE113 User Interface Design	22ADM111 Glimpses of Glorious India		COUNSELLING		Library/Sports	
Thursday	23MAT117 Linear Algebra	23CSE111 Object Oriented Programming		23MAT116 Discrete Mathematics	COUNSELLING	LIBRARY		23PHY115 Modern Physics (T)		Library/Sports	
Friday	23CSE113 User Interface Design Lab BYOD			23MAT117 Linear Algebra	23CSE111 Object Oriented Programming	Library/Sports		22ADM111 Glimpses of Glorious India (P)		23MAT116 Discrete Mathematics Lab	

Course Code	Course Name	L-T-P	Credits	Faculty Name	Assisting Faculty
23MAT116	Discrete Mathematics	3-0-2	4	Dr. Srivasa Rao Thota	
23MAT117	Linear Algebra	3-0-2	4	Mr. Sapavat Bixapathi	
23CSE111	Object Oriented Programming	3-0-2	4	Dr. Balaji TK	
23PHY115	Modern Physics	2-1-0	3	Dr. C R Kesavalu	
23CSE113	User Interface Design	2-0-2	3	Dr. Rajkumar Batchu	Mr. Barge Bharadwaj
23MEE115	Manufacturing Practice	0-0-3	1	Dr. Sathies S.	
22ADM111	Glimpses of Glorious India	2-0-1	2	Dr. Siva Panuganti	

Explanation of the tags:

- : is used as a container to group content like gallery and cards.
- <img> : tag is used to display syllabus images using the src attribute.
- CSS classes like .gallery, .card, .title, .faculty, and .break are introduced for styling.
- display : grid is used to create a two-column gallery layout.
- margin : auto centers the content on the page.
- border-radius is used to give rounded corners to cards and images.
- border-collapse : collapse makes the table borders neat.
- font-family changes the text style of the whole page.
- Combined selector td, th applies common styles.
- HTML comments <!-- --> are used to organize the code into sections.

WEEK -4

1 . **Aim :** A college wants to collect student details through an online admission form design a registration form using only html tags(no css) with propern labels table alignment,background color and borders

Procedure :

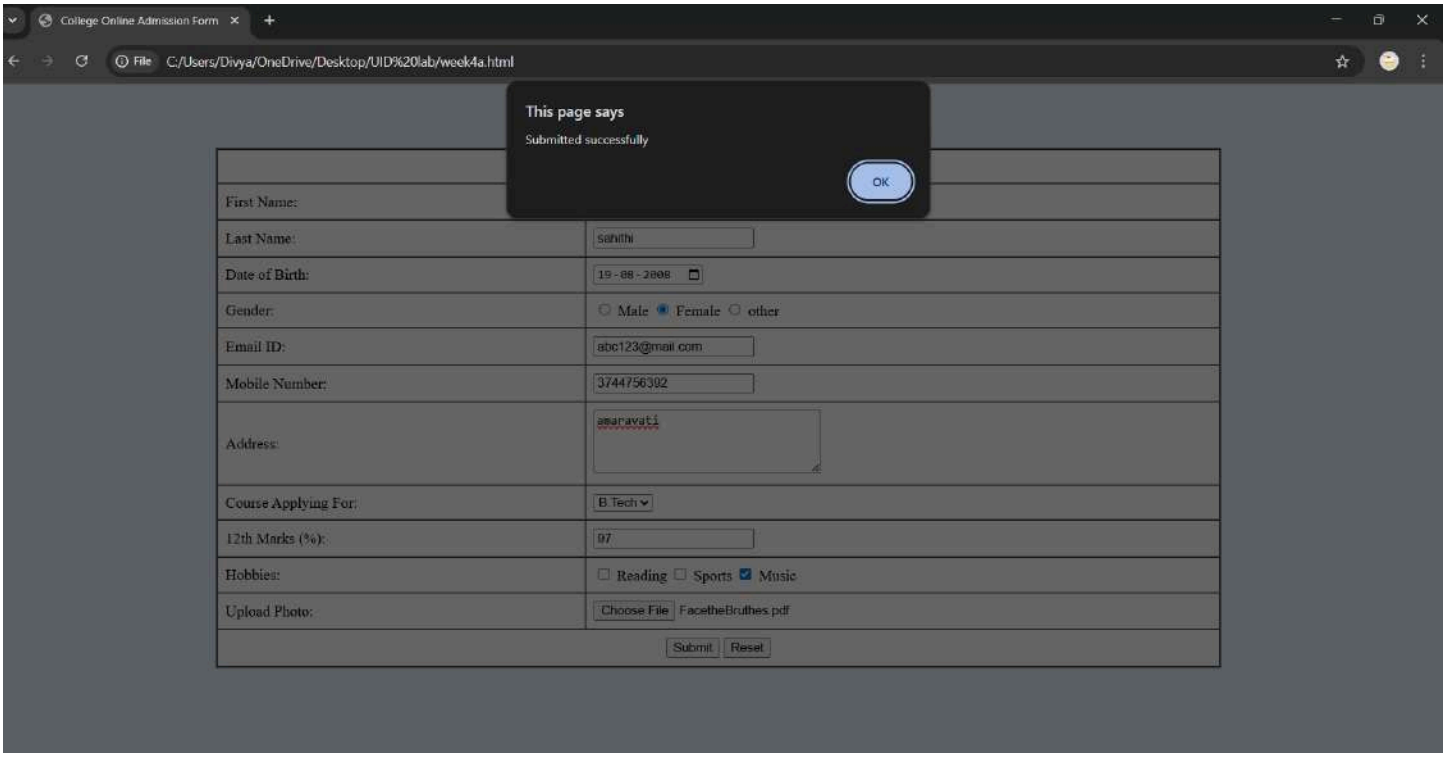
```
<!DOCTYPE html>
<html>
<head>
  <title>College Online Admission Form</title>
</head>
<body bgcolor="#e6f2ff">
  <h2 align="center">COLLEGE ONLINE ADMISSION FORM</h2>
  <form>
    <table border="2" align="center" cellpadding="8" cellspacing="0" width="70%" bgcolor="#ffffff">
      <tr>
        <td colspan="2" align="center"><b>Student Personal Details</b></td>
      </tr>
      <tr>
        <td><label for="fname">First Name:</label></td>
        <td><input type="text" id="fname" name="fname"></td>
      </tr>
```

```
<tr>
    <td><label for="lname">Last Name:</label></td>
    <td><input type="text" id="lname" name="lname"></td>
  </tr>
  <tr>
    <td><label for="dob">Date of Birth:</label></td>
    <td><input type="date" id="dob" name="dob"></td>
  </tr>
  <tr>
    <td><label>Gender:</label></td>
    <td>
      <input type="radio" name="gender"> Male
      <input type="radio" name="gender"> Female
      <input type="radio" name="gender"> other
    </td>
  </tr>
  <tr>
    <td><label for="email">Email ID:</label></td>
    <td><input type="email" id="email" name="email"></td>
  </tr>
  <tr>
    <td><label for="phone">Mobile Number:</label></td>
    <td><input type="tel" id="phone" name="phone"></td>
  </tr>
```

```
<tr>
    <td><label for="address">Address:</label></td>
    <td><textarea id="address" name="address" rows="4" cols="30"></textarea></td>
</tr>
<tr>
    <td><label for="course">Course Applying For:</label></td>
    <td>
        <select id="course" name="course">
            <option>B.Tech</option>
            <option>B.Sc</option>
            <option>B.Com</option>
            <option>BCA</option>
            <option>MCA</option>
        </select>
    </td>
</tr>
<tr>
    <td><label for="marks">12th Marks (%):</label></td>
    <td><input type="number" id="marks" name="marks"></td>
</tr>
<tr>
    <td><label>Hobbies:</label></td>
    <td>
```

```
<input type="checkbox"> Reading
            <input type="checkbox"> Sports
            <input type="checkbox"> Music
          </td>
        </tr>
        <tr>
          <td><label for="photo">Upload Photo:</label></td>
          <td><input type="file" id="photo" name="photo"></td>
        </tr>
        <tr>
          <td colspan="2" align="center">
            <button type="button" onclick="alert('Submitted successfully')">
              Submit
              </button>
            <input type="reset" value="Reset">
          </td>
        </tr>
      </table>
    </form>
</body>
</html>
```

Output:



Explanation of the tags:

- <form> – Collects user input
- colspan – Merges two columns into one.
- <label for> – Labels input fields.
- <input> – Takes user input (text, date, radio, email, tel, number, file, reset).
- <textarea> – Multi-line text input.
- <select> & <option> – Dropdown list.
- checkbox – Multiple selections.
- <button onclick> – Shows alert message.
- Attributes: id, name, align, bgcolor, rows, cols, border, width – control layout and identification.

2 . **AIM:** A Company wants to design a webpage that display two different forms side by side using iframes in which one should display a login form and other should display register form

Program :

<_____login form_____>

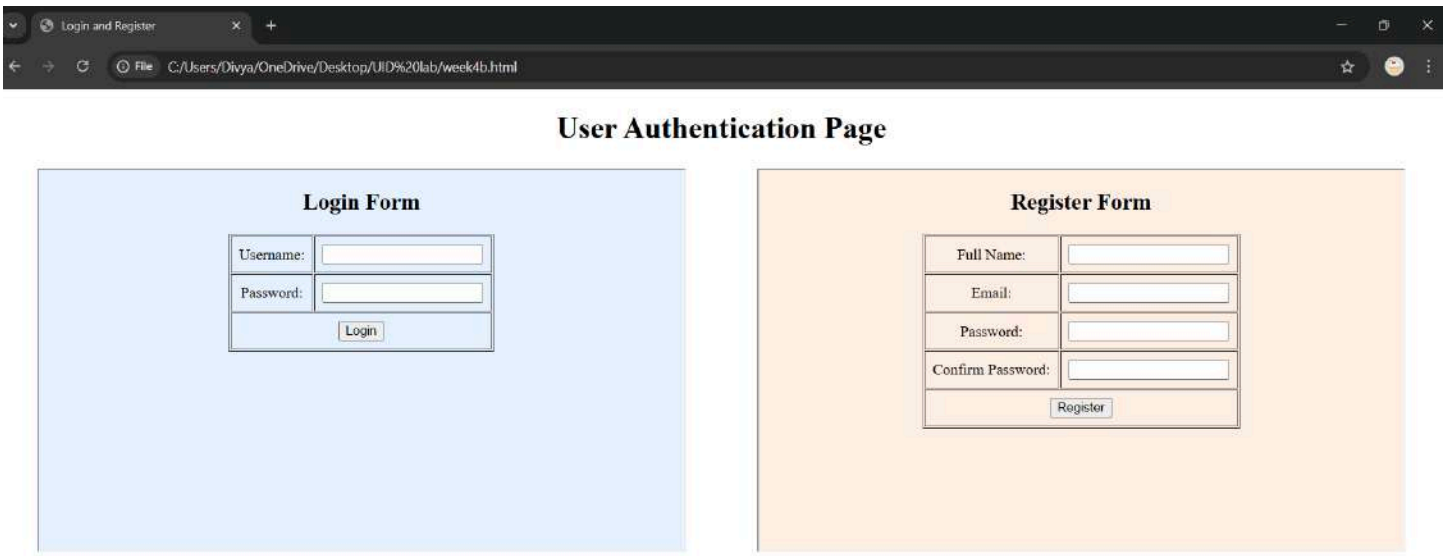
```
<!DOCTYPE html>
<html>
<head>
  <title>Login Form</title>
</head>
<body bgcolor="#e6f2ff">
  <h2 align="center">Login Form</h2>

  <form align="center">
    <table border="1" cellpadding="8" align="center">
      <tr>
        <td>Username:</td>
        <td><input type="text" name="uname"></td>
      </tr>
      <tr>
        <td>Password:</td>
        <td><input type="password" name="pass"></td>
      </tr>
      <tr>
```

```
  <td colspan="2" align="center">
                <input type="submit" value="Login">
            </td>
        </tr>
      </table>
  </form>
</body>
</html>
 <_______Register form_________>
<!DOCTYPE html>
<html>
<head>
  <title>Register Form</title>
</head>
<body bgcolor="#fff0e6">
  <h2 align="center">Register Form</h2>
  <form align="center">
      <table border="1" cellpadding="8" align="center">
        <tr>
            <td>Full Name:</td>
            <td><input type="text" name="fname"></td>
        </tr>
        <tr>
            <td>Email:</td>
```

```
  </head>
<body>
 <td><input type="email" name="email"></td>
        </tr>
        <tr>
            <td>Password:</td>
            <td><input type="password" name="pass"></td>
        </tr>
        <tr>
            <td>Confirm Password:</td>
            <td><input type="password" name="cpass"></td>
        </tr>
        <tr>
            <td colspan="2" align="center">
                <input type="submit" value="Register">
            </td>
        </tr>
      </table>
   </form>
</body>
</html>
 <_______Main page_________>
<!DOCTYPE html>
<html>
```

```
  <head>
  <title>Login and Register</title>
<h1 align="center">User Authentication Page</h1>
  <table width="100%">
    <tr>
      <td width="50%" align="center">
        <iframe src="login.html" width="90%" height="400"></iframe>
      </td>
      <td width="50%" align="center">
        <iframe src="register.html" width="90%" height="400"></iframe>
      </td>
    </tr>
  </table>
</body>
</html>
```

Output:



Explanation of the tags:

- **<iframe> (Inline Frame):** Displays another HTML page inside the current page.
- Two separate pages are created: **Login Form** and **Register Form**.
- A **main page** uses two iframes to show both forms **side by side**.
- **src attribute:** Specifies which page to load inside the iframe.
- **width & height:** Set the iframe size.
- A **table with two columns** is used to align the iframes in one row.
- Each iframe loads a different form page (login / register).

WEEK -5

1 . **Aim :** To create a responsive card layout using HTML and CSS with three cards, each containing an image, title, and button.

Procedure :

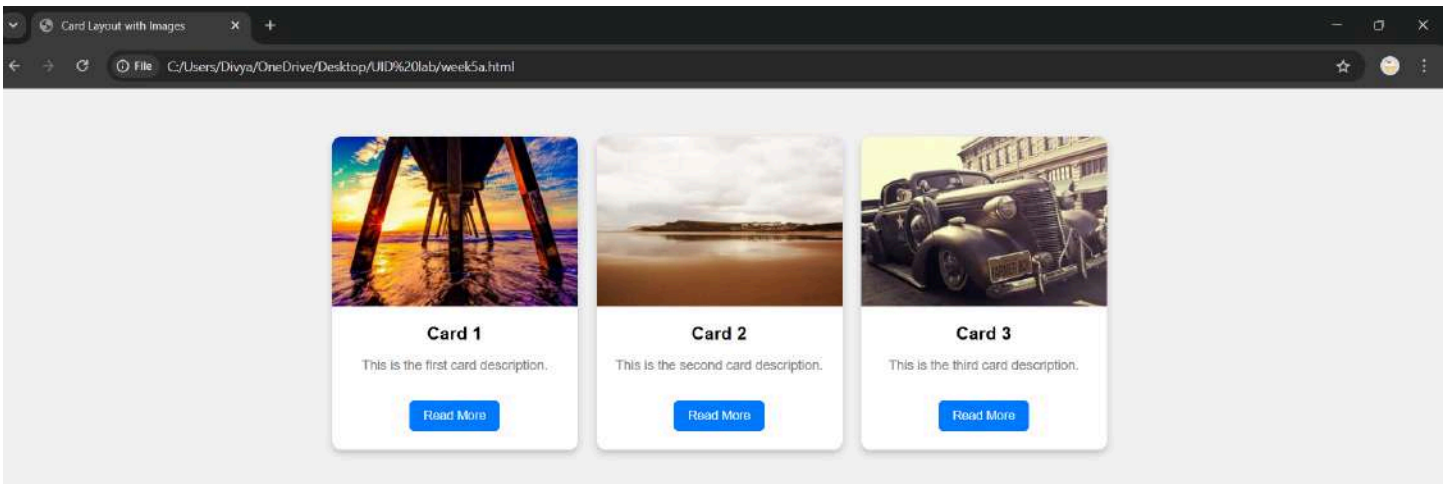
```
<!DOCTYPE html>
<html>
<head>
  <title>Card Layout with Images</title>
  <style>
    body{
      font-family: Arial, sans-serif;
      background-color: #f2f2f2;
    }
    .container{
      display: flex;
      justify-content: center;
      gap: 20px;
      margin-top: 50px;
    }
    .card{
      background-color: white;
```

```
width: 260px;
        border-radius: 10px;
        box-shadow: 0 4px 8px rgba(0,0,0,0.2);
        overflow: hidden;
        text-align: center;
    }
    .card img{
        width: 100%;
        height: 180px;
        object-fit: cover;
    }
    .card h3{
        margin: 15px 0 10px 0;
    }
    .card p{
        font-size: 14px;
        padding: 0 15px;
        color: gray;
    }
    .card button{
        margin: 15px 0 20px 0;
padding: 8px 15px;
        border: none;
        background-color: #007bff;
```

```
color: white;
            border-radius: 5px;
            cursor: pointer;
        }
        .card button:hover{
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="card">
            <img src="https://picsum.photos/300/200?1" alt="Image 1">
            <h3>Card 1</h3>
            <p>This is the first card description.</p>
            <button>Read More</button>
        </div>
        <div class="card">
            <img src="https://picsum.photos/300/200?2" alt="Image 2">
            <h3>Card 2</h3>
            <p>This is the second card description.</p>
            <button>Read More</button>
        </div>
<div class="card">
```

```
<img src="https://picsum.photos/300/200?3" alt="Image 3">

        <h3>Card 3</h3>

        <p>This is the third card description.</p>

        <button>Read More</button>

      </div>

    </div>

</body>

</html>
```

Output:



Explanation of the tags:

- flex → Align items in row
- box-shadow → Adds shadow
- border-radius → Rounded corners
- object-fit: cover → Adjust image without distortion
- :hover → Changes style when mouse is over element
- overflow: hidden → Hides extra content outside border

2.**Aim:** To design a mini website using iframe navigation links (Home, About, Contact), where clicking on each link displays the respective content within the same page using internal or external CSS.

Procedure :

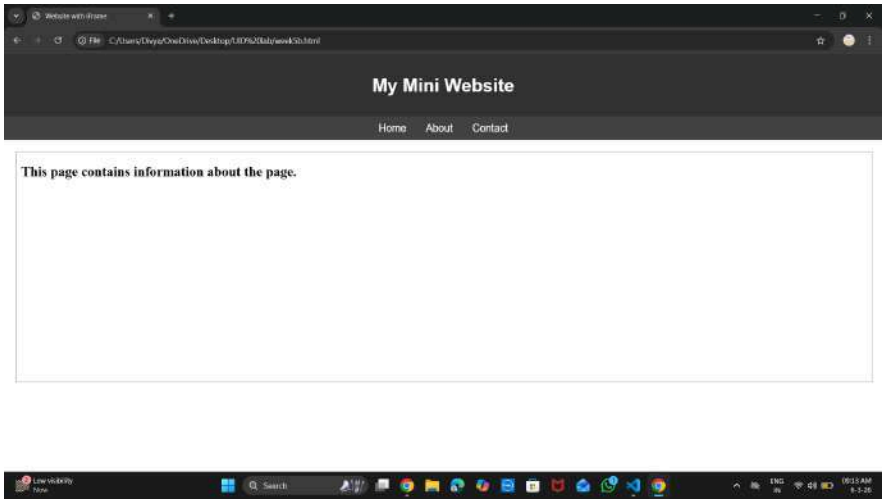
```
<------style.css------->

body {
   margin: 0;
   font-family: Arial, sans-serif; }
header {
   background-color: #333;
   color: white;
   padding: 15px;
   text-align: center;  }
nav {
   background-color: #444;
   text-align: center;
   padding: 10px;   }
nav a {
   color: white;
   text-decoration: none;
   margin: 15px;
   font-size: 18px;  }
nav a:hover {
   color: yellow; }
```

```
section {
    padding: 20px; }
iframe {
    width: 100%;
    height: 400px;
    border: 2px solid #ccc; }
<------home.html-------->
<!DOCTYPE html>
<html>
<body>
    <h2>This is my Website.Here is the content of this page.</h2>
</body>
</html>
<-----about.html------>
<!DOCTYPE html>
<html>
<body>
    <h2>This page contains information about the page.</h2>
</body>
</html>
<-----contact.html------->
<!DOCTYPE html>
<html>
```

```
<body>
    <h2>if having any issues regarding this</h2>
    <p>divya@2008gmail.com</p>
    <p>phone:1234567890</p>
</body>
</html>
<-------5b.html------>
<!DOCTYPE html>
<html>
<head>
    <title> Website with iFrame</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>My Mini Website</h1>
    </header>
    <nav>
        <a href="home.html" target="contentFrame">Home</a>
        <a href="about.html" target="contentFrame">About</a>
        <a href="contact.html" target="contentFrame">Contact</a>
    </nav>
```

```
<section>

    <iframe name="contentFrame" src="home.html"></iframe>  </section></body>

</html>
```

Output:



Explanation of the tags:

<header>

- Used for top section of webpage.

<nav>

- Used for navigation links.

<section>

- Used to divide content into sections.

<a>

- Anchor tag.
- Used to create hyperlinks.
- Important attributes:
  - href → Link address
  - target → Where to open the link

<iframe>

- Used to display another webpage inside current webpage.
- Important attributes:
  - src → Page to display
  - name → Used with targe

3. Aim: To design an Amrita login page using HTML and CSS that displays different campuses, where clicking each campus shows its details on the same page

Procedure :

```
<!DOCTYPE html>

<html>

<head>

   <title>Amrita Login Page</title>

   <style>

      body {

         margin: 0;

         font-family: Arial, sans-serif;

         background: linear-gradient(to right,light orange, pale pink);

          background: url(https://wallpapers.com/images/hd/cool-light-blue-l6csxyf32ij7t1or.jpg) no-repeat
center center/cover        }

      /* Header */

      .header {

         background-color:lightcoral;

         color: white;

         text-align: center;

padding: 15px;

         font-size: 28px;

         font-weight: bold;        }

      /* Navigation */
```

```
.nav {
        background-color:darkcyan;
        text-align: center;
        padding: 10px;        }
    .nav button {
        background: transparent;
        border: none;
        color:lightgoldenrodyellow;
        font-size: 16px;
        margin: 0 15px;
        cursor: pointer; }
    .nav button:hover {
        text-decoration: underline;       }
    /* Main Layout */
    .container {
        display: flex;
        justify-content: center;
        align-items: flex-start;
        margin: 40px;
        gap: 40px;  }
    /* Login Box */
    .login-box {
background: linear-gradient(to bottom,darkviolet,lightcoral);
        padding: 25px;
```

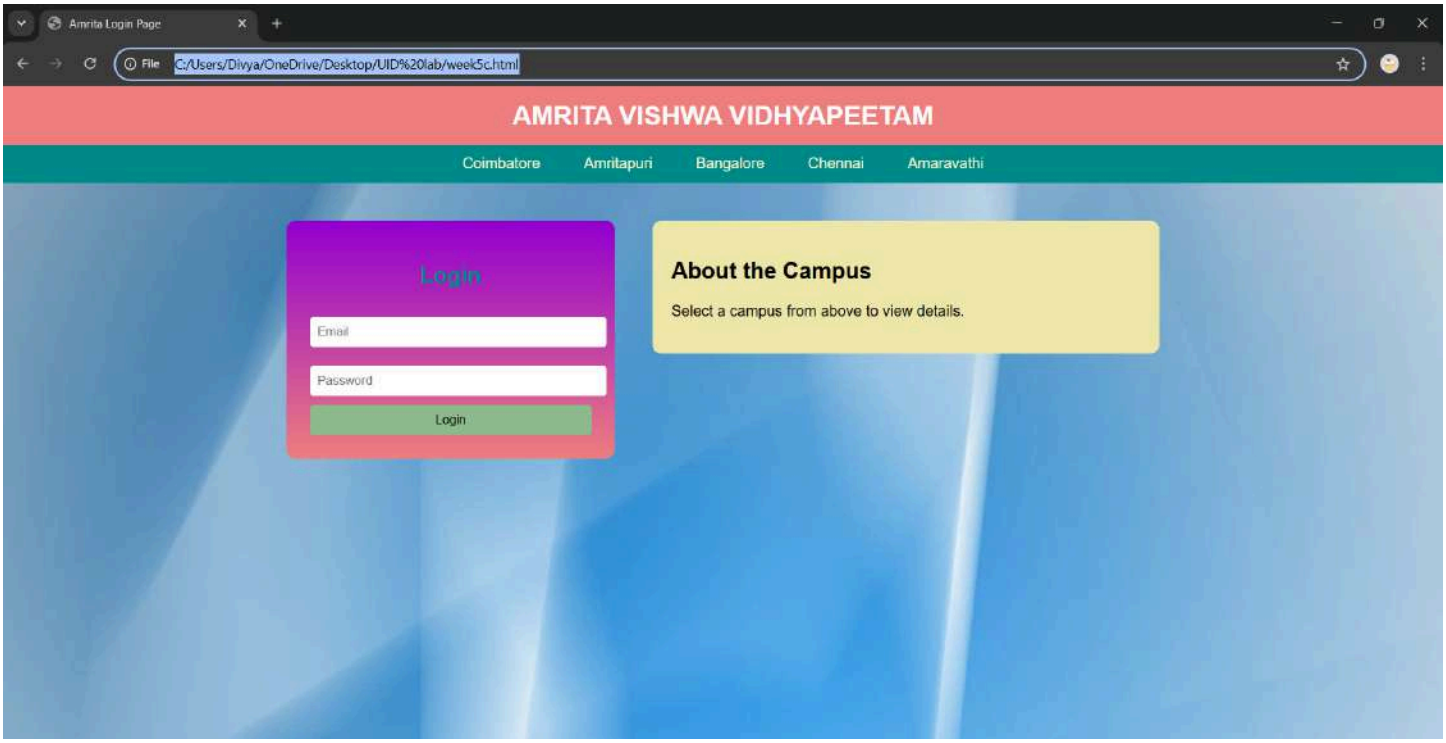
```
width: 300px;
        border-radius: 10px;
        color:darkcyan;    }
    .login-box h2 {
        text-align: center; }
    .login-box input {
        width: 100%;
        padding: 8px;
        margin: 10px 0;
        border: none;
        border-radius: 4px;}
    .login-box button {
        width: 100%;
        padding: 8px;
        background-color:darkseagreen;
        color:light turquoise;
        border: none;
        border-radius: 4px;
        cursor: pointer;}
    /* Campus Details */
    .campus-details {
background:palegoldenrod;
        width: 500px;
```

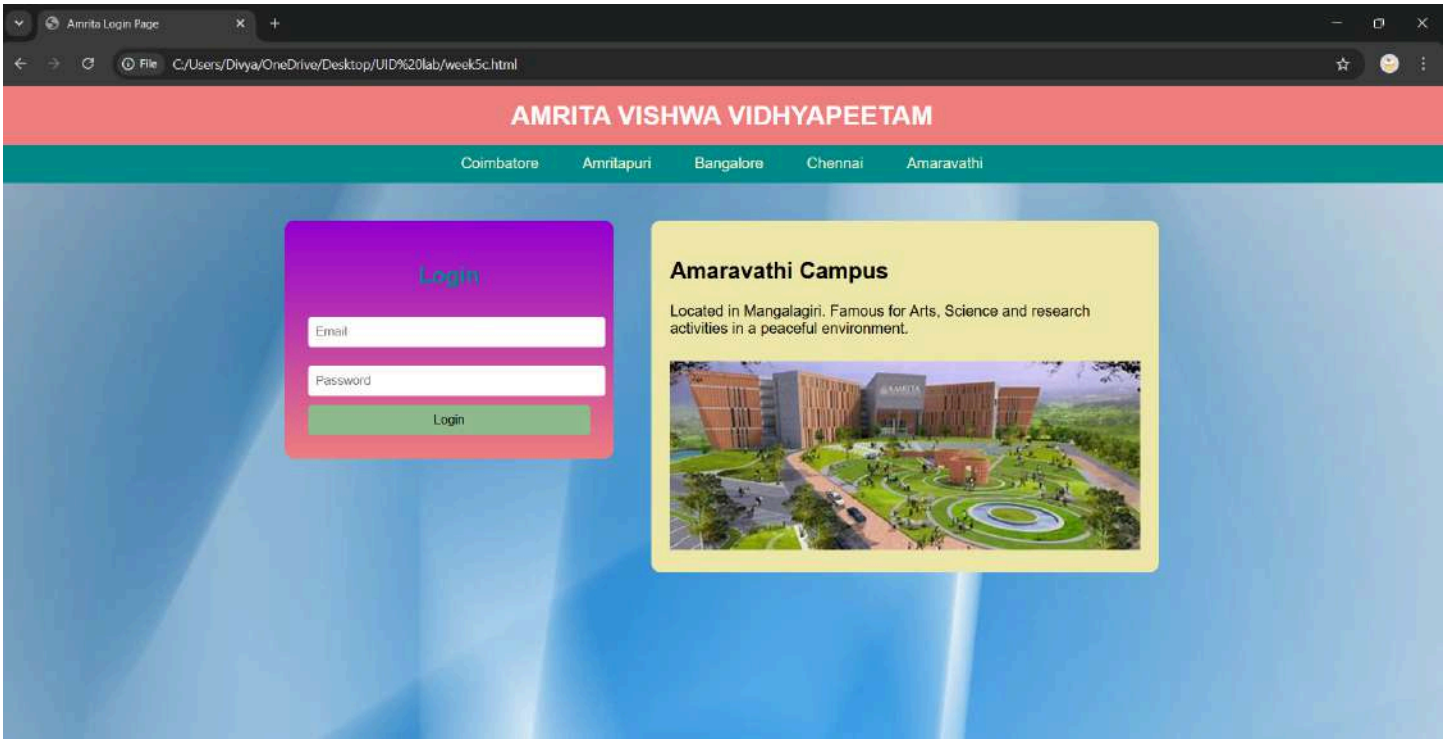
```
padding: 20px;
        border-radius: 10px;     }
      .campus-details img {
        width: 100%;
        height: 200px;
        object-fit: cover;
        margin-top: 10px;  }
  </style>
</head>
<body>
  <div class="header">
    AMRITA VISHWA VIDHYAPEETAM
  </div>
  <div class="nav">
    <button onclick="showCampus('coimbatore')">Coimbatore</button>
    <button onclick="showCampus('amritapuri')">Amritapuri</button>
    <button onclick="showCampus('bangalore')">Bangalore</button>
    <button onclick="showCampus('chennai')">Chennai</button>
    <button onclick="showCampus('Amaravathi')">Amaravathi</button>
  </div>
 <div class="container">
    <!-- Login Section -->
    <div class="login-box">
      <h2>Login</h2>
```

```
<input type="text" placeholder="Email">

        <input type="password" placeholder="Password">

        <button>Login</button>

      </div>

      <!-- Campus Info Section -->

      <div class="campus-details" id="campusInfo">

        <h2>About the Campus</h2>

        <p>Select a campus from above to view details.</p>

      </div>

    <script>

      function showCampus(campus) {

        let info = document.getElementById("campusInfo");

        if (campus === "coimbatore") {

          info.innerHTML = `

          <h2>Coimbatore Campus</h2>

          <p>Located in Tamil Nadu. Known for Engineering and Technology programs with modern
infrastructure.</p>

          <img src="https://tse3.mm.bing.net/th/id/OIP.elkN6U_aVSlLuBTcPX2q_QHaEK?
pid=Api&P=0&h=180" alt="Coimbatore Campus"> `;   }

        else if (campus === "amritapuri") {

          info.innerHTML = `

          <h2>Amritapuri Campus</h2>

          <p>Located in Kerala. Famous for Arts, Science and research activities in a peaceful environment.
</p>

    <img src="https://tse4.mm.bing.net/th/id/OIP.jPuLxk4ALi6yUzvKevF7fgHaEM?pid=Api&P=0&h=180"
alt="Amritapuri Campus">  `;}
```

```
else if (campus === "bangalore") {

            info.innerHTML = `

            <h2>Bangalore Campus</h2>

            <p>Located in Karnataka. Offers Engineering and Management programs with industry exposure.
</p>

            <img src="https://tse4.mm.bing.net/th/id/OIP.AxDXEuEdONmn1Poym0qaiQHaDt?
pid=Api&P=0&h=180" alt="Bangalore Campus">`;           }

        else if (campus === "chennai") {

            info.innerHTML = `

            <h2>Chennai Campus</h2>

            <p>Located in Tamil Nadu. Special focus on engineering disciplines and innovation.</p>

            <img src="https://tse2.mm.bing.net/th/id/OIP.5Cd4ekgT1P_EX7zhzH_D1AHaDp?
pid=Api&P=0&h=180" alt="Chennai Campus">

            `;

        }

        else if (campus === "Amaravathi") {

            info.innerHTML = `

            <h2>Amaravathi Campus</h2>

            <p>Located in Mangalagiri. Famous for Arts, Science and research activities in a peaceful
environment.</p>

            <img src="https://tse4.mm.bing.net/th/id/OIP.WIKj0_HGLKy7-vepQuwuLwHaDK?
pid=Api&P=0&h=180" alt="Amaravathi Campus">;}}

        </script>

</body>

</html>
```

Output:


Amrita Login Page
File
C:/Users/Divya/OneDrive/Desktop/UID%20lab/week5c.html
AMRITA VISHWA VIDHYAPEETAM
Coimbatore
Amritapuri
Bangalore
Chennai
Amaravathi
Login
Email
Password
Login
About the Campus
Select a campus from above to view details.


Amrita Login Page
File
C:/Users/Divya/OneDrive/Desktop/UID%20lab/week5c.html
AMRITA VISHWA VIDHYAPEETAM
Coimbatore
Amritapuri
Bangalore
Chennai
Amaravathi
Login
Email
Password
Login
Amaravathi Campus
Located in Mangalagiri. Famous for Arts, Science and research activities in a peaceful environment.

Explanation of the tags:

CSS Styling

body

- Sets background image.
- Removes margin.
- Applies font style.

.header

- Top banner.
- Background color: lightcoral.
- Center aligned white text.

.nav

- Navigation bar.
- Contains buttons for campuses.

.container

- Uses display: flex;
- Aligns login box and campus details side by side.

.login-box

- Styled login form.
- Gradient background.
- Rounded corners.

.campus-details

- Displays campus information.
- Contains heading, paragraph and image.

Navigation Buttons

```
<button onclick="showCampus('coimbatore')">Coimbatore</button>
```

- onclick calls JavaScript function.
- Sends campus name as parameter.

WEEK –6

Aim : Design a responsive e-commerce application using the grid layout. This application should contain different categories of products such as electronics, clothing, home-appliances, books. Based on the category we select corresponding products should be displayed on the sidebar.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>MyShop Online Store</title>
<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body{
font-family:Arial;
margin:0;
background:#f2f2f2;
}

/* HEADER */

.header{
background:#a8f15a;
color:white; padding:15px;
font-size:24px; text-
align:center;
}

/* LAYOUT */

.container{
display:flex; flex-
wrap:wrap;
```

```
}

/* SIDEBAR */

.sidebar{ flex:1 1
220px;
background:#aaf8fc;
padding:20px;
border-right:1px solid #ccc;
}

.sidebar h3{
margin-top:0;
}

.sidebar label{
display:block;
padding:10px;
cursor:pointer;
border-radius:4px;
}

.sidebar label:hover{
background:#eee;
}

/* RADIO BUTTONS */

input[type="radio"]{
display:none;
}

/* PRODUCTS */

.products{ flex:4
1 600px;
display:grid;
grid-template-columns:repeat(auto-fit,minmax(220px,1fr)); gap:20px;
padding:20px;
}
```

```
.card{
background:white;
padding:15px;
border-radius:6px;
text-align:center;
box-shadow:0 2px 8px rgba(0,0,0,0.1); transition:0.3s;
}

.card:hover{
transform:scale(1.05);
}

.card img{
width:100%;
height:180px; object-
fit:cover; border-
radius:5px;
}

.price{ color:green;
font-weight:bold;
}

.rating{ color:orange;
}

/* CATEGORY FILTER */

#electronics:checked ~ .container .products .card:not(.electronics){ display:none;
}

#men:checked ~ .container .products .card:not(.men){ display:none;
}

#women:checked ~ .container .products .card:not(.women){ display:none;
}
```

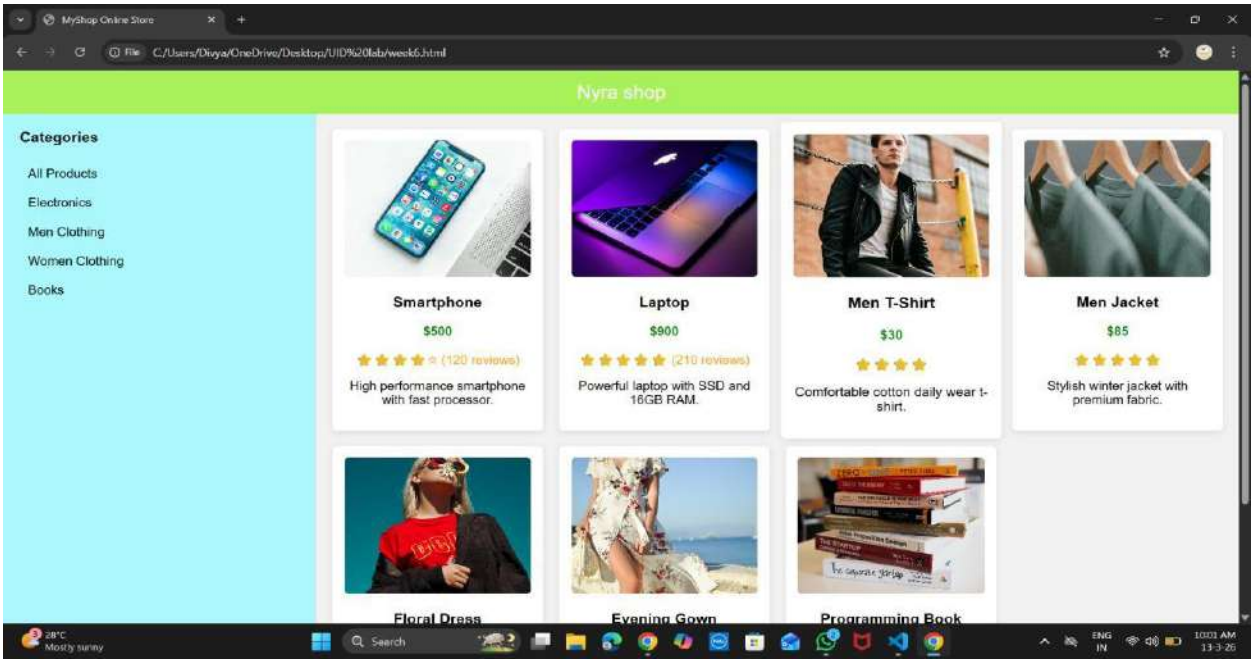
```
#books:checked ~ .container .products .card:not(.books){ display:none;
}

/* MOBILE RESPONSIVE */

@media(max-width:900px){

.products{
grid-template-columns:repeat(auto-fit,minmax(200px,1fr));
}

}

@media(max-width:768px){

.container{
flex-direction:column;
}

.sidebar{
width:100%;
border-right:none;
border-bottom:1px solid #ccc;
}

.products{
grid-template-columns:repeat(auto-fit,minmax(180px,1fr));
}

.header{
font-size:20px;
}

}

@media(max-width:480px){

.products{
grid-template-columns:1fr;
}
```

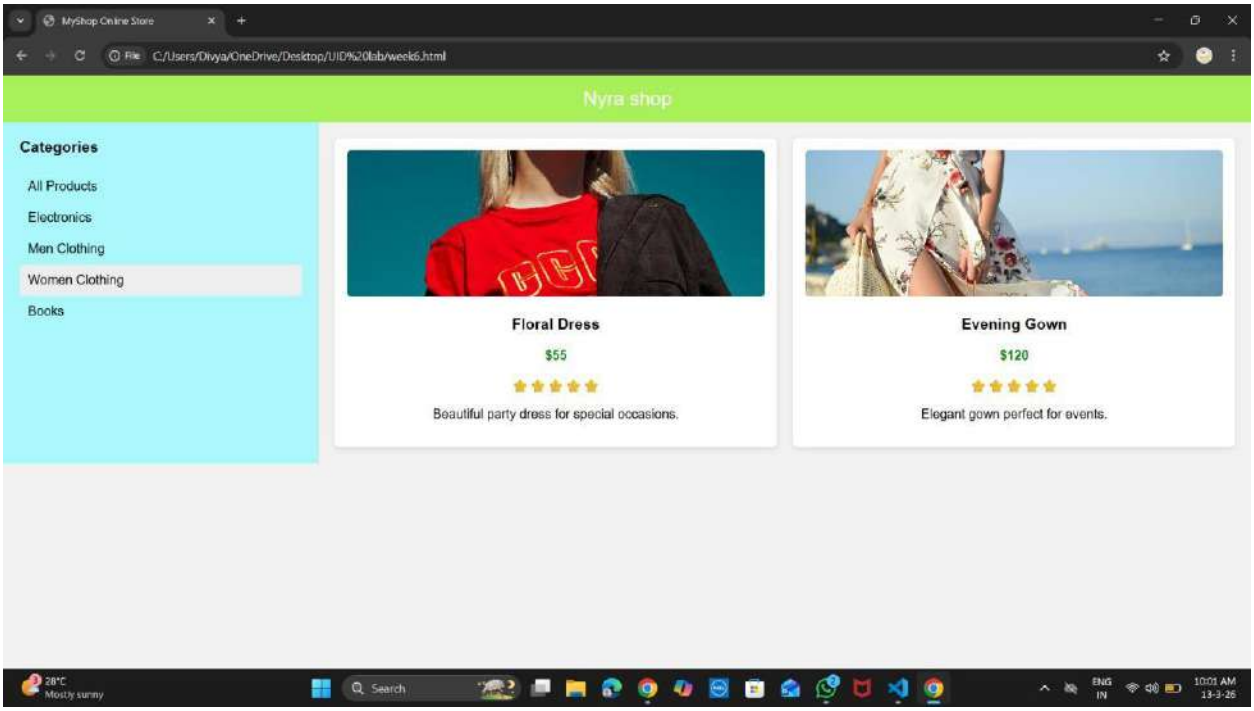
```
.card img{ height:160px;
}

}

</style>
</head>

<body>

<div class="header"> Nyra
  shop
</div>

<!-- RADIO BUTTONS -->

<input type="radio" name="cat" id="all" checked>
<input type="radio" name="cat" id="electronics">
<input type="radio" name="cat" id="men">
<input type="radio" name="cat" id="women">
<input type="radio" name="cat" id="books">

<div class="container">

<!-- SIDEBAR -->

<div class="sidebar">

<h3>Categories</h3>

<label for="all">All Products</label>
<label for="electronics">Electronics</label>
<label for="men">Men Clothing</label>
<label for="women">Women Clothing</label>
<label for="books">Books</label>

</div>

<!-- PRODUCTS -->
```

```
<div class="products">

<div class="card electronics">
<img src="https://images.unsplash.com/photo-1511707171634-5f897ff02aa9">
<h3>Smartphone</h3>
<p class="price">$500</p>
<p class="rating">⭐ ⭐ ⭐ ⭐ ☆  (120 reviews)</p>
<p>High performance smartphone with fast processor.</p>
</div>

<div class="card electronics">
<img src="https://images.unsplash.com/photo-1517336714731-489689fd1ca8">
<h3>Laptop</h3>
<p class="price">$900</p>
<p class="rating">⭐ ⭐ ⭐ ⭐ ⭐  (210 reviews)</p>
<p>Powerful laptop with SSD and 16GB RAM.</p>
</div>

<div class="card men">
<img src="https://images.unsplash.com/photo-1520975916090-3105956dac38">
<h3>Men T-Shirt</h3>
<p class="price">$30</p>
<p class="rating">⭐ ⭐ ⭐ ⭐</p>
<p>Comfortable cotton daily wear t-shirt.</p>
</div>

<div class="card men">
<img src="https://images.unsplash.com/photo-1523381210434-271e8be1f52b">
<h3>Men Jacket</h3>
<p class="price">$85</p>
<p class="rating">⭐ ⭐ ⭐ ⭐ ⭐</p>
<p>Stylish winter jacket with premium fabric.</p>
</div>

<div class="card women">
<img src="https://images.unsplash.com/photo-1529139574466-a303027c1d8b">
<h3>Floral Dress</h3>
<p class="price">$55</p>
<p class="rating">⭐ ⭐ ⭐ ⭐ ⭐</p>
<p>Beautiful party dress for special occasions.</p>
</div>
```

```
<div class="card women">
<img src="https://images.unsplash.com/photo-1496747611176-843222e1e57c">
<h3>Evening Gown</h3>
<p class="price">$120</p>
<p class="rating">⭐ ⭐ ⭐ ⭐ ⭐</p>
<p>Elegant gown perfect for events.</p>
</div>
<div class="card books">
<img src="https://images.unsplash.com/photo-1512820790803-83ca734da794">
<h3>Programming Book</h3>
<p class="price">$25</p>
<p class="rating">⭐ ⭐ ⭐ ⭐ ⭐</p>
<p>Learn programming with practical examples.</p>
</div>
</div>
</div>
</body>
</html>
```

Output:





Explanation of the tags:

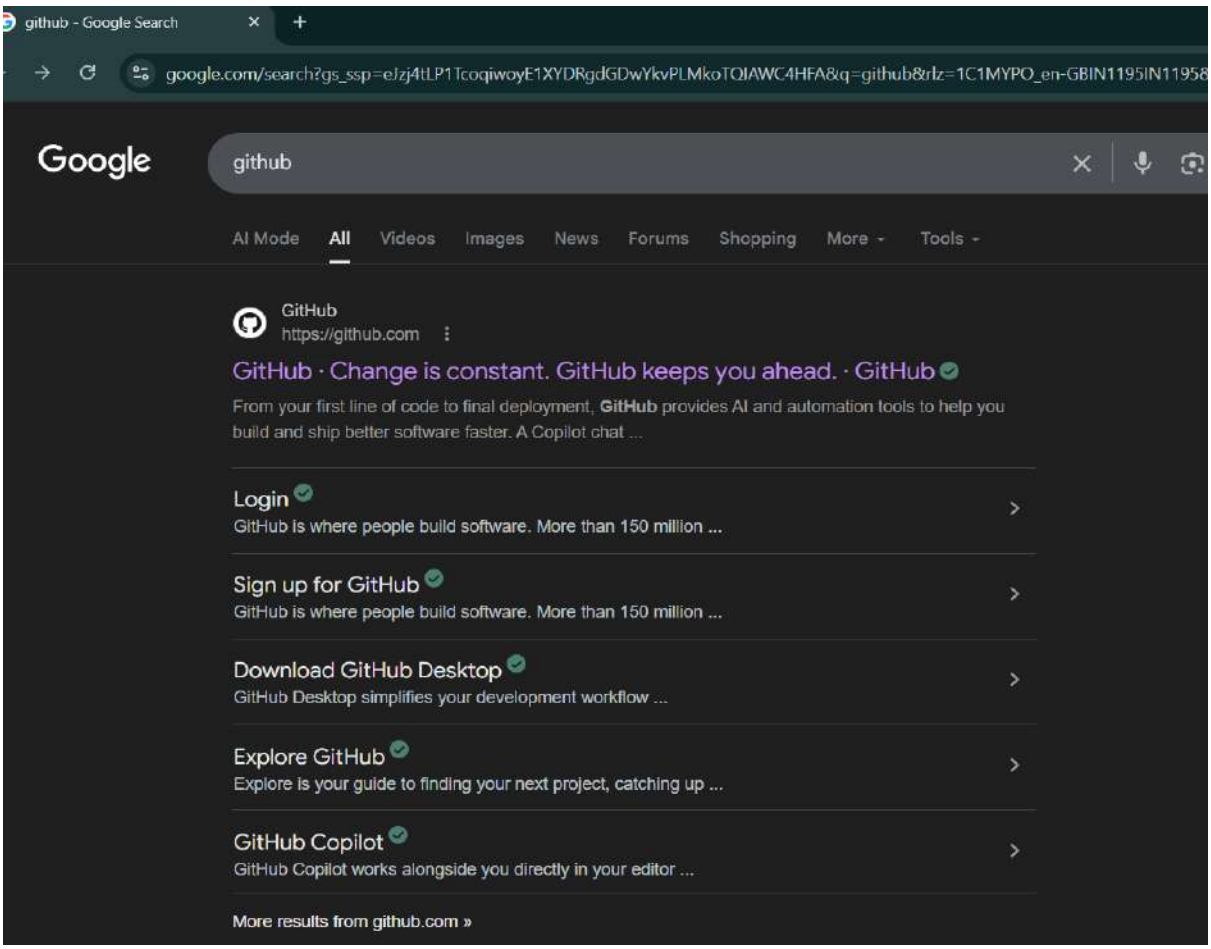
- **meta>**
  The meta tag provides **additional information about the webpage**. In this program, it is used to make the webpage **responsive on mobile devices**.
- **<style>**
  This tag is used to include **CSS styling inside the HTML file**. It controls the layout, colors, fonts, and design of the webpage.
- **<body>**
- ** **
  The div tag is used to **create sections or containers** in the webpage like header, sidebar, product cards, cart section, and footer.
- **<img>**
  The img tag is used to **display images** on the webpage. In this code it is used to show product images.
- **<button>**
  This tag creates **clickable buttons**, such as **Add to Cart** and **Buy Now**.

WEEK –7

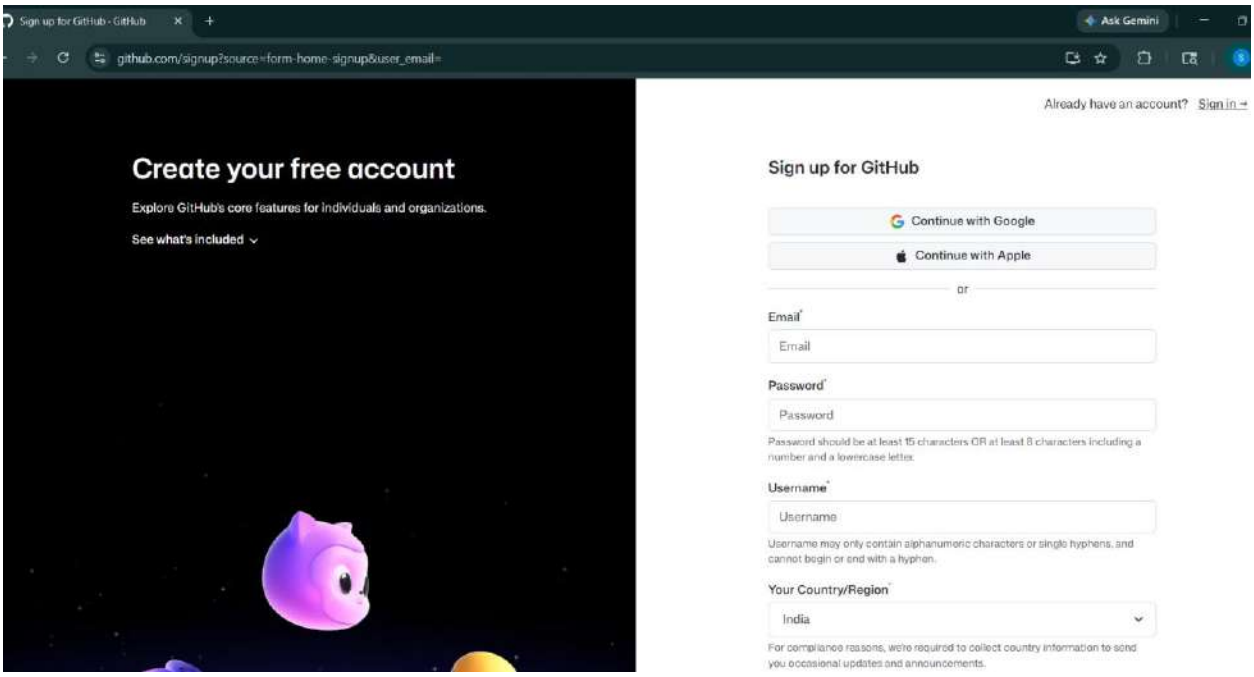
AIM: 1. create a github account and create a repository named UID lab with roll no and upload every week experiments and lab report in repository and make repository file

Procedure :

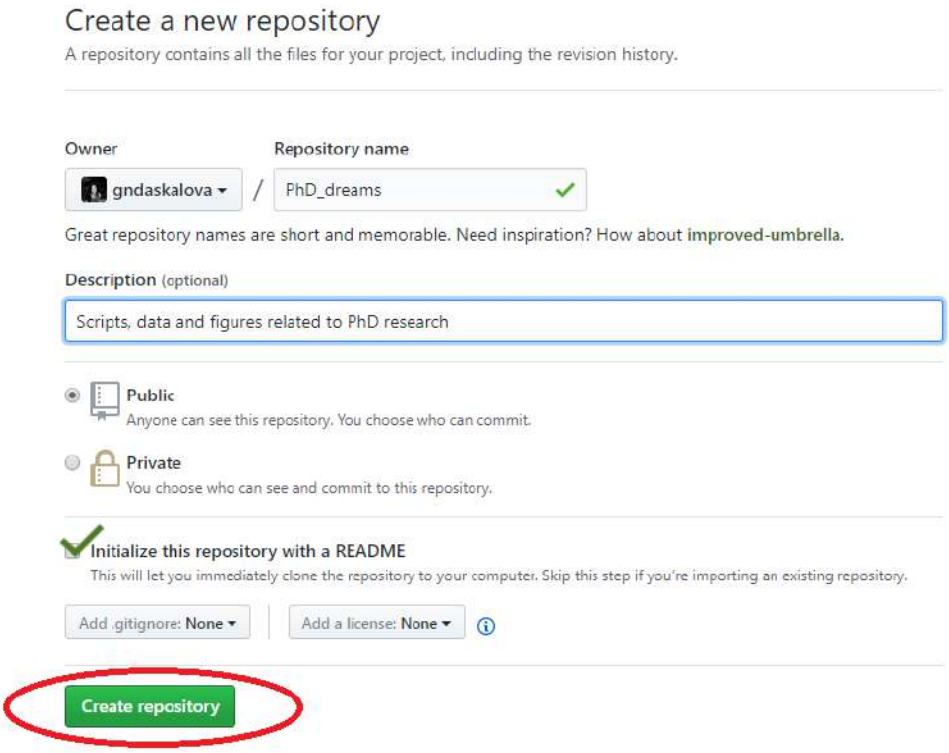
step – 1



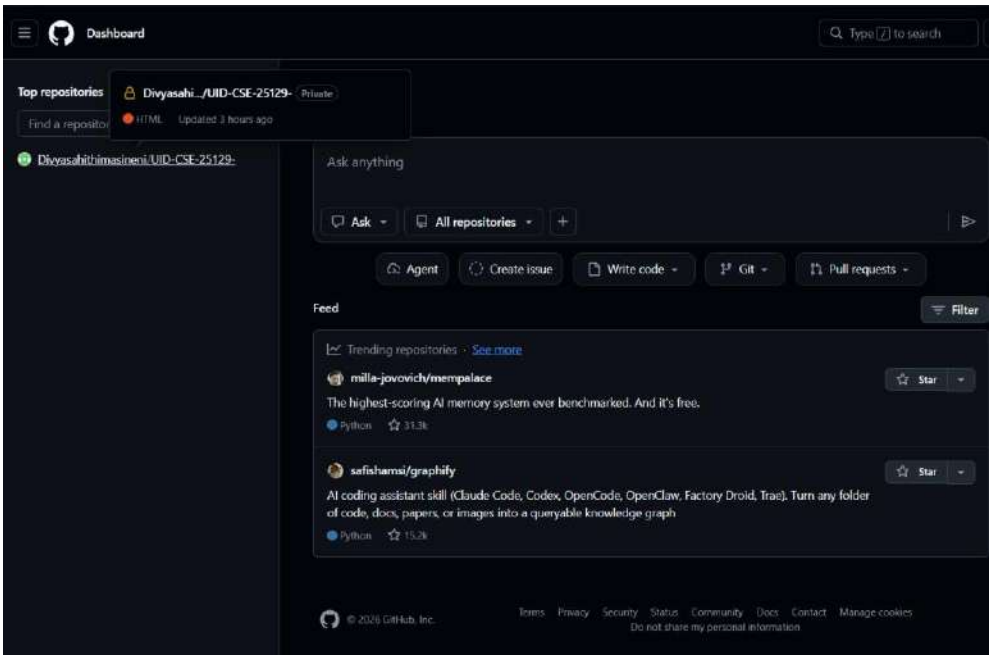
step – 2


Sign up for GitHub · GitHub
Ask Gemini
github.com/signup?source=form-home-signup&user_email=
Create your free account
Explore GitHub's core features for individuals and organizations.
See what's included
Already have an account? Sign in →
Sign up for GitHub
Continue with Google
Continue with Apple
or
Email*
Email
Password*
Password
Password should be at least 15 characters OR at least 8 characters including a number and a lowercase letter.
Username*
Username
Username may only contain alphanumeric characters or single hyphens, and cannot begin or end with a hyphen.
Your Country/Region*
India
For compliance reasons, we're required to collect country information to send you occasional updates and announcements.

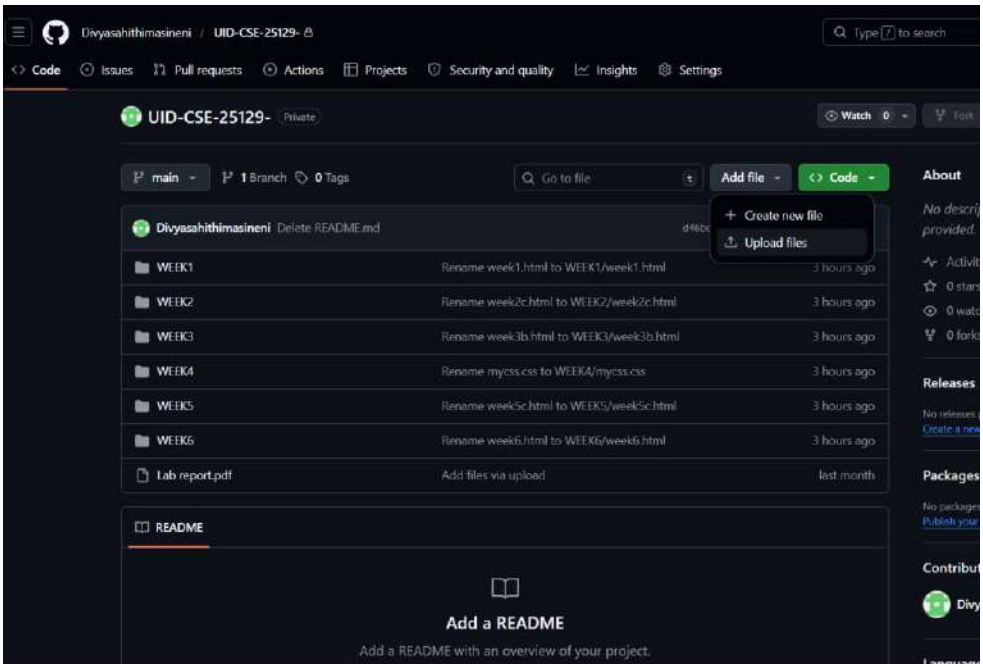
step – 3


Create a new repository
A repository contains all the files for your project, including the revision history.
Owner
Repository name
gndaskalova
/
PhD_dreams
Great repository names are short and memorable. Need inspiration? How about improved-umbrella.
Description (optional)
Scripts, data and figures related to PhD research
Public
Anyone can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.
Add .gitignore: None
Add a license: None
Create repository

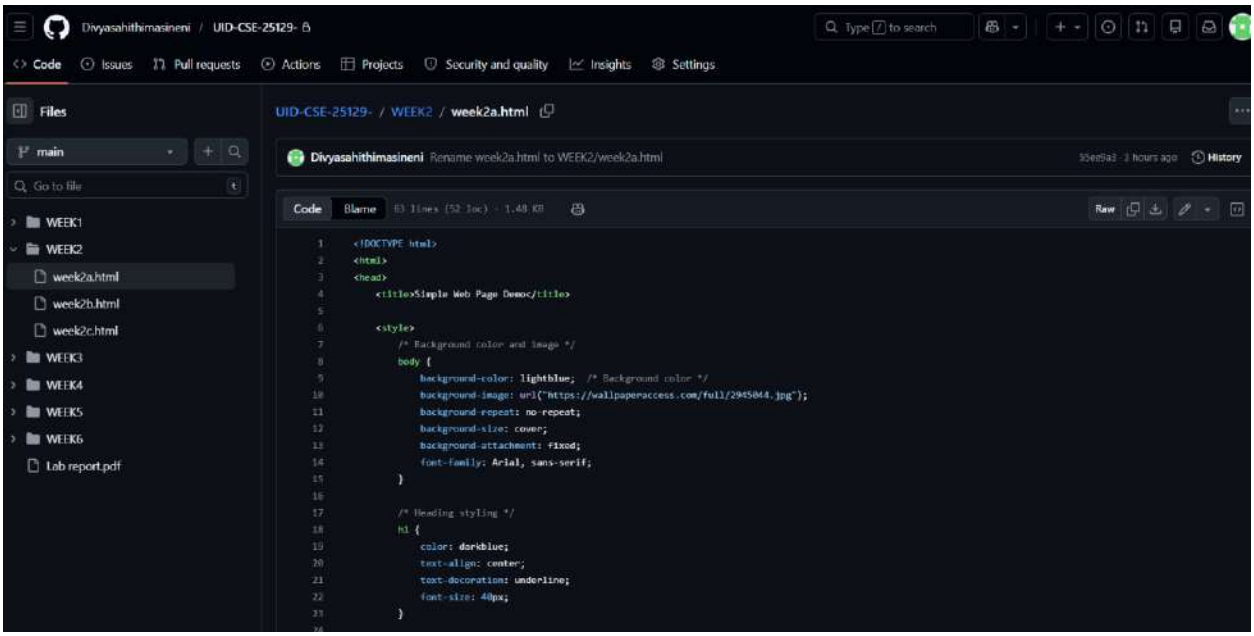
step – 4


Dashboard
Type / to search
Top repositories
Divyasahi.../UID-CSE-25129- Private
HTML Updated 3 hours ago
Divyasahithimasineni/UID-CSE-25129-
Ask anything
Ask
All repositories
Agent
Create issue
Write code
Git
Pull requests
Feed
Filter
Trending repositories · See more
milla-jovovich/mempalace
The highest-scoring AI memory system ever benchmarked. And it's free.
Python
Star
safishamsi/graphify
AI coding assistant skill (Claude Code, Codex, OpenCode, OpenClaw, Factory Droid, Trae). Turn any folder of code, docs, papers, or images into a queryable knowledge graph
Python
Star
© 2026 GitHub, Inc.
Terms Privacy Security Status Community Docs Contact Manage cookies
Do not share my personal information

step – 5


Divyasahithimasineni / UID-CSE-25129-
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security and quality
Insights
Settings
UID-CSE-25129- Private
Watch 0
main
1 Branch
0 Tags
Go to file
Add file
Code
Create new file
Upload files
About
Divyasahithimasineni Delete README.md
WEEK1
Rename week1.html to WEEK1/week1.html
WEEK2
Rename week2c.html to WEEK2/week2c.html
3 hours ago
WEEK3
Rename week3b.html to WEEK3/week3b.html
3 hours ago
WEEK4
Rename mycss.css to WEEK4/mycss.css
3 hours ago
WEEK5
Rename week5c.html to WEEK5/week5c.html
3 hours ago
WEEK6
Rename week6.html to WEEK6/week6.html
3 hours ago
Lab report.pdf
Add files via upload
last month
Releases
Packages
Contribut
README
Add a README
Add a README with an overview of your project.

step – 6


Divyasahithimasineni / UID-CSE-25129-
Type / to search
Code
Issues
Pull requests
Actions
Projects
Security and quality
Insights
Settings
Files
main
Go to file
WEEK1
WEEK2
week2a.html
week2b.html
week2c.html
WEEK3
WEEK4
WEEK5
WEEK6
Lab report.pdf
UID-CSE-25129- / WEEK2 / week2a.html
Divyasahithimasineni Rename week2a.html to WEEK2/week2a.html
History
Code
Blame
Raw
<!DOCTYPE html>
<html>
<head>
<title>Simple Web Page Demo</title>
<style>
/* Background color and image */
body {
background-color: lightblue; /* Background color */
background-image: url("https://wallpaperaccess.com/full/2945044.jpg");
background-repeat: no-repeat;
background-size: cover;
background-attachment: fixed;
font-family: Arial, sans-serif;
}
/* Heading styling */
h1 {
color: darkblue;
text-align: center;
text-decoration: underline;
font-size: 40px;
}

WEEK –8

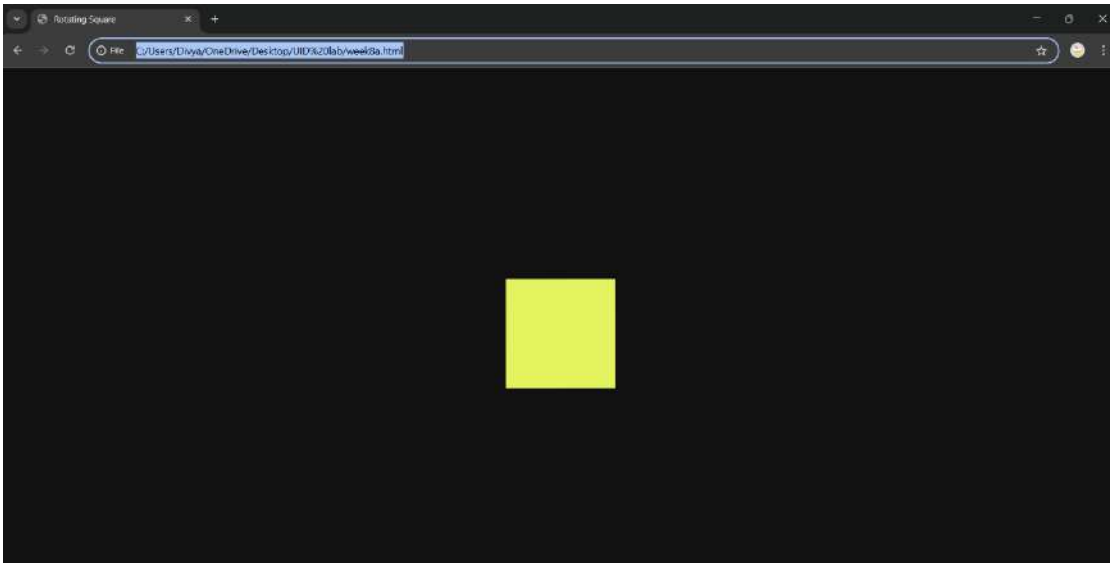
8a.Aim: Create a webpage that displays a square that continu rotates 360° using CSS animation.

Program:

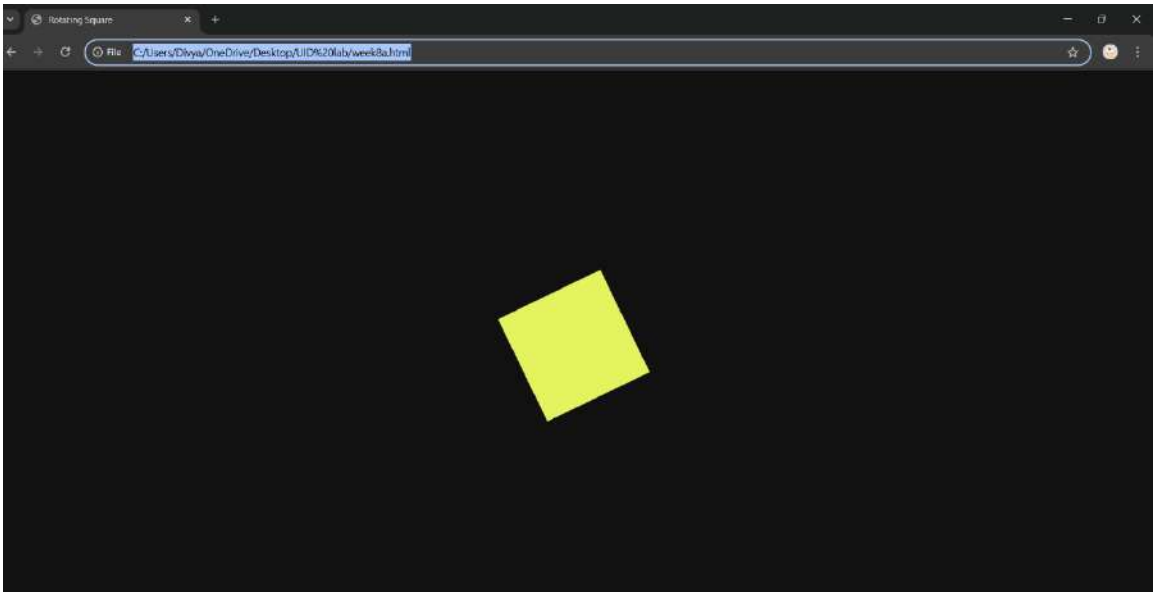
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Rotating Square</title>

<style>
    body {
        margin: 0;
        height: 100vh;
        display: flex;
        justify-content: center;
        align-items: center;
        background-color: #111;
    }

    .square {
        width: 150px;
        height: 150px;
        background-color: #e4f360;
        animation: rotateSquare 2s linear infinite;
    }

    @keyframes rotateSquare {
        from {
            transform: rotate(0deg);
        }
        to {
            transform: rotate(360deg);
        }
    }
</style>

</head>
<body>

<div class="square"></div>
```

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

Output:



This is square is rotating as we can we see the difference from two pictures



Explanation of tags:

<html lang="en">

- Main tag of webpage
- lang="en" means English language

<head>

- Contains page information (not visible)

<meta charset="UTF-8">

- Supports all characters properly

<style>

- Used to write CSS (design part)

Body

- margin: 0
- removes spaceheight: 100vh
- full screen heightdisplay: flex
- uses flexbox justify-content: center
- center horizontallyalign-items: center
- center vertically
- background-color: #111 → dark background

Square

- Creates the square design
- width & height → size of square
- background-color → gives color
- animation → makes it rotate

@keyframes

- rotate Square
- Defines animation steps
- from → start (0° rotation)
- to → end (360° rotation)

<body>

- → Visible part of webpage

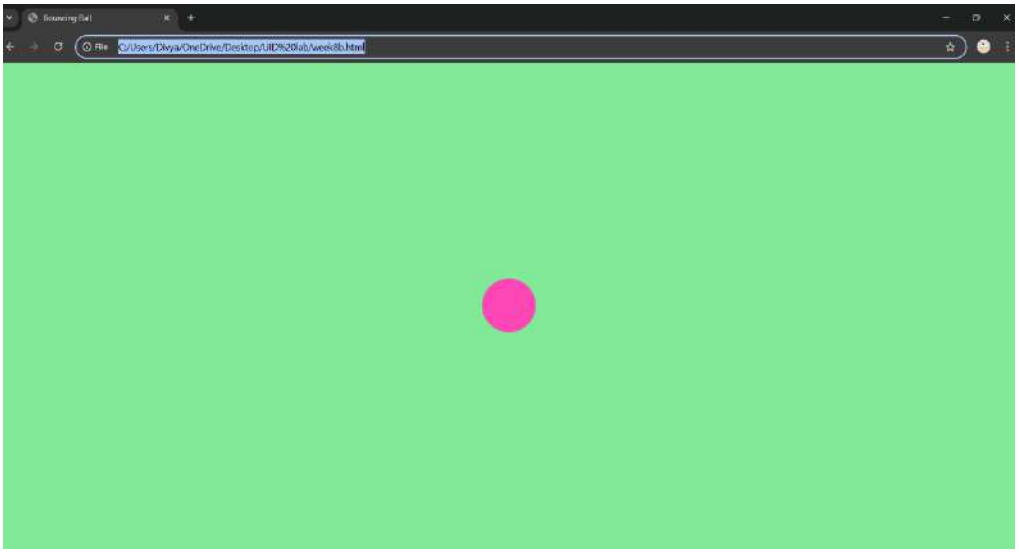
- Creates the square
- Connected to .square CSS

8b.Aim: Create a ball that moves up and down continuously only using CSS animations

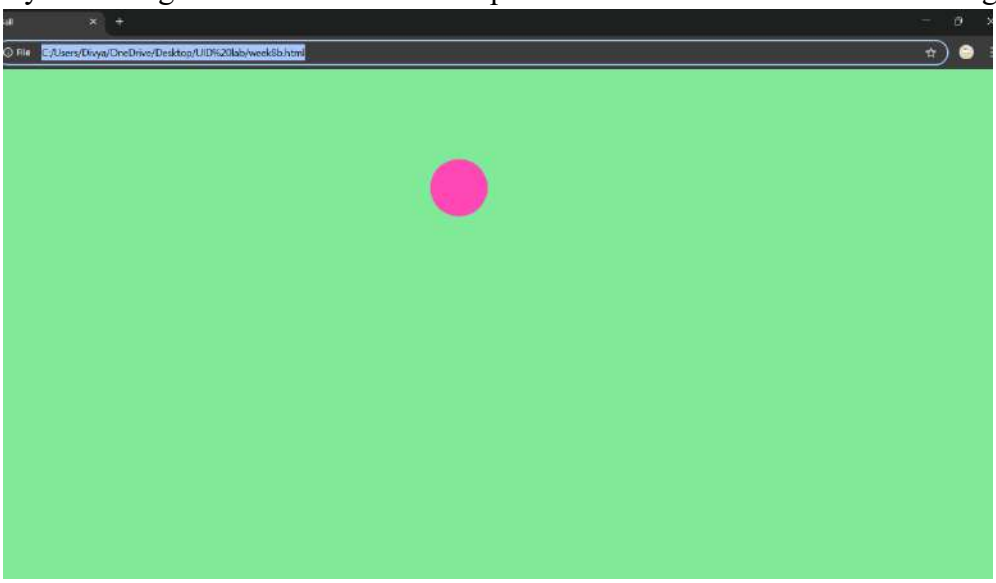
Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Bouncing Ball</title>

<style>
  body {
    margin: 0;
    height: 100vh;
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #80ea99;
  }
  .ball {
    width: 80px;
    height: 80px;
    background-color: #ff47b5;
    border-radius: 50%;
    animation: bounce 1s ease-in-out infinite alternate;
  }
  @keyframes bounce {
    from {
      transform: translateY(0);
    }
    to {
      transform: translateY(-200px);
    }
  }
</style>

</head>
<body>
<div class="ball"></div>
</body>
</html>
```

Output:



By observing difference between two pictures we can see the ball is bouncing



Explanation of tags:

<html lang="en">
- Main tag of webpage
- lang="en" means English language

<head>
- Contains page information (not visible)

<meta charset="UTF-8">
- Supports all characters properly

<style>
- Used to write CSS (design part)

body
- margin: 0 → removes space
- height: 100vh → full screen height
- display: flex → uses flexbox
- justify-content: center → center horizontally
- align-items: center → center vertically
- background-color: #80ea99 → light green background

.ball
- Creates the ball design
- width & height → size of ball
- background-color → gives pink color
- border-radius: 50% → makes it a circle
- animation → makes it bounce

animation: bounce 1s ease-in-out infinite alternate
- bounce → animation name
- 1s → duration
- ease-in-out → smooth movement
- infinite → repeats forever
- alternate → moves up and down

@keyframes bounce
- Defines animation steps

from
- transform: translateY(0) → starting position

to
- transform: translateY(-200px) → moves ball upward

<body>
- Visible part of webpage

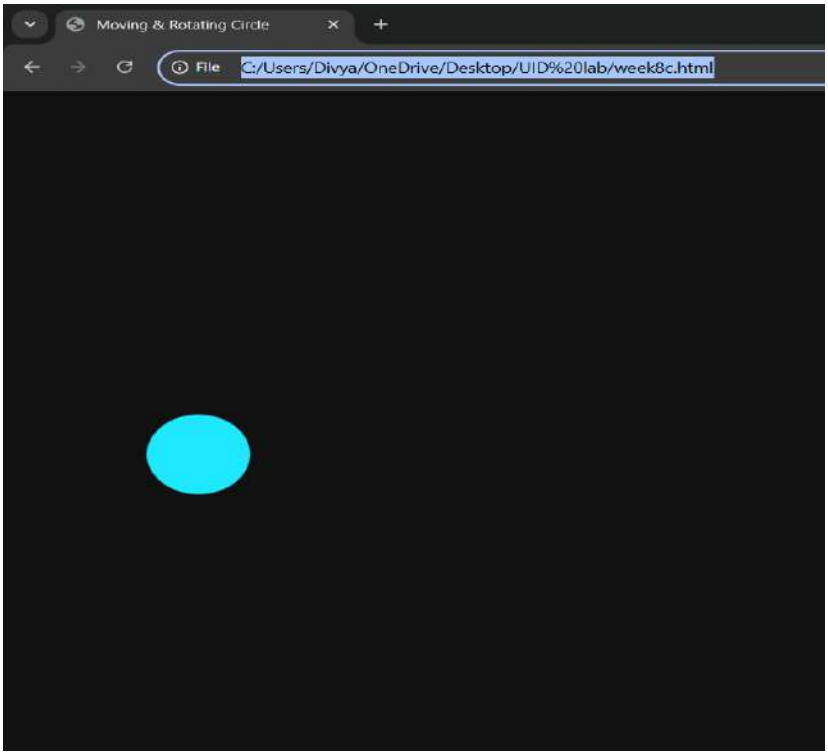
- Creates the ball
- Connected to .ball CSS

8c.Aim: Create a circle that moves horizontally and rotates at the same time.

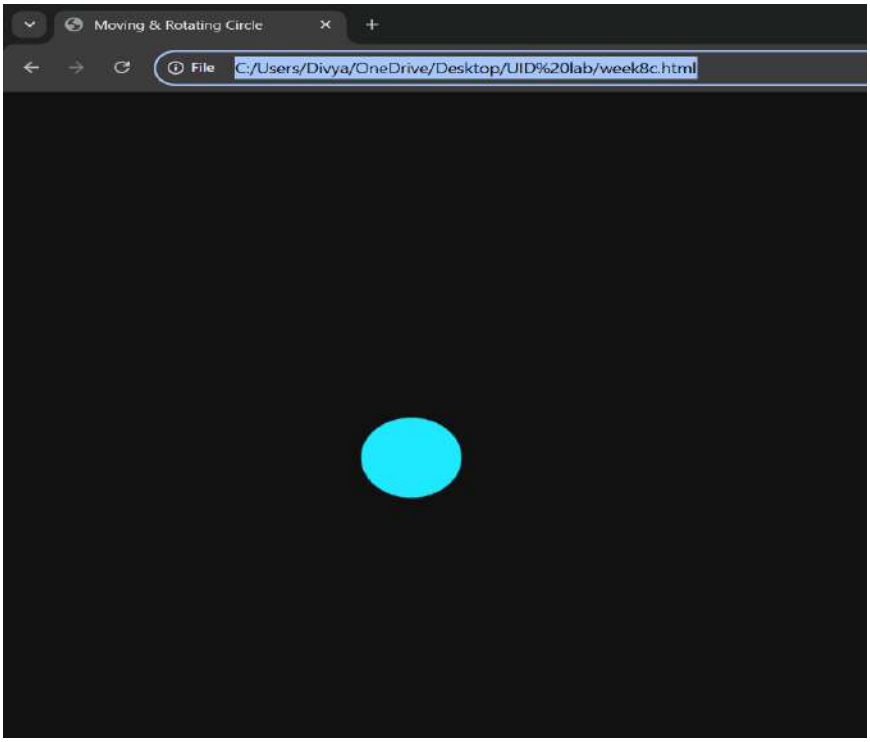
Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Moving & Rotating Circle</title>

<style>
  body {
    margin: 0;
    height: 100vh;
    display: flex;
    align-items: center;
    background-color: #111;
    overflow: hidden;
  }
  .circle {
    width: 80px;
    height: 80px;
    background-color: #1ee9ff;
    border-radius: 50%;
    position: relative;
    animation: moveRotate 3s linear infinite;
  }

  @keyframes moveRotate {
    0% {
      transform: translateX(0) rotate(0deg);
    }
    100% {
      transform: translateX(300px) rotate(360deg);
    }
  }
</style>

</head>
<body>

<div class="circle"></div>

</body>
</html>
```

Output:



By the difference between both we can see the ball is rotating



Explanation of tags:

<html lang="en">
- Main tag of webpage
- lang="en" means English language

<head>
- Contains page information (not visible)

<meta charset="UTF-8">
- Supports all characters properly

<style>
- Used to write CSS (design part)

body
- margin: 0 → removes space
- height: 100vh → full screen height
- display: flex → uses flexbox
- align-items: center → center vertically
- background-color: #111 → dark background
- overflow: hidden → hides extra content outside screen

.circle
- Creates the circle design
- width & height → size of circle
- background-color → gives blue color
- border-radius: 50% → makes it a circle
- position: relative → allows movement
- animation → makes it move and rotate

animation: moveRotate 3s linear infinite
- moveRotate → animation name
- 3s → duration
- linear → constant speed
- infinite → repeats forever

@keyframes moveRotate
- Defines animation steps

0%
- transform: translateX(0) rotate(0deg)
→ starting position (no movement, no rotation)

100%
- transform: translateX(300px) rotate(360deg)
→ moves right and completes full rotation

<body>
- Visible part of webpage

- Creates the circle
- Connected to .circle CSS

WEEK-9

AIM: Create an webpage using html, CSS, java script based on the given conditions:

a)Create a button for each program, based on the button clicked the corresponding program has to be executed and should display the output.

b)The logics for each button includes finding an even numbers, odd number, prime or not, factorial of a number, amstrong or not.

c)Use appropriate designs to display the output.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Week 9 JavaScript Programs</title>

  <style>
    body {
      font-family: Arial, sans-serif;
      background: linear-gradient(135deg, #dfe9f3, #ffffff);
      text-align: center;
      margin: 0;
      padding: 0;
    }

    .container {
      width: 80%;
      margin: 50px auto;
      background: white;
      padding: 30px;
      border-radius: 15px;
      box-shadow: 0 0 15px rgba(0,0,0,0.2);
```

```
}

h1 {
    color: #2c3e50;
}

input {
    padding: 12px;
    width: 250px;
    font-size: 16px;
    border: 2px solid #3498db;
    border-radius: 8px;
    margin: 15px;
}

button {
    padding: 12px 20px;
    margin: 10px;
    border: none;
    border-radius: 8px;
    background-color: #3498db;
    color: white;
    font-size: 16px;
    cursor: pointer;
}

button:hover {
    background-color: #21618c;
    transform: scale(1.05);
```

```
        }

        #output {
            margin-top: 25px;
            padding: 20px;
            font-size: 20px;
            font-weight: bold;
            color: #1b4f72;
            background-color: #eaf2f8;
            border-radius: 10px;
            min-height: 40px;
        }
    </style>
</head>

<body>

    <div class="container">
        <h1>Check the number</h1>

        <p>Enter a number:</p>
        <input type="number" id="numberInput" placeholder="Enter number">

        <br>

        <button onclick="findEven()">Even Number</button>
        <button onclick="findOdd()">Odd Number</button>
        <button onclick="checkPrime()">Prime or Not</button>
        <button onclick="findFactorial()">Factorial</button>
```

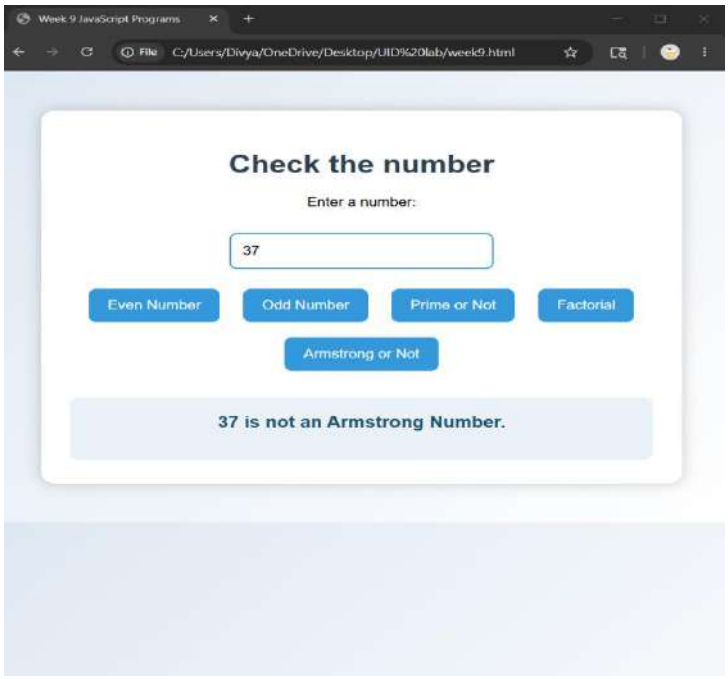
```
        <button onclick="checkArmstrong()">Armstrong or Not</button>

        <div id="output">Output will be displayed here</div>
    </div>

    <script>
        function getNumber() {
            return parseInt(document.getElementById("numberInput").value);
        }

        function findEven() {
            let num = getNumber();

            if (isNaN(num)) {
                document.getElementById("output").innerHTML = "Please enter a valid number.";
            }
            else if (num % 2 === 0) {
                document.getElementById("output").innerHTML = num + " is an Even Number.";
            }
            else {
                document.getElementById("output").innerHTML = num + " is not an Even
Number.";
            }
        }

        function findOdd() {
            let num = getNumber();

            if (isNaN(num)) {
```

```
            document.getElementById("output").innerHTML = "Please enter a valid number.";
        }
        else if (num % 2 !== 0) {
            document.getElementById("output").innerHTML = num + " is an Odd Number.";
        }
        else {
            document.getElementById("output").innerHTML = num + " is not an Odd
Number.";
        }
    }

    function checkPrime() {
        let num = getNumber();
        let isPrime = true;

        if (isNaN(num)) {
            document.getElementById("output").innerHTML = "Please enter a valid number.";
            return;
        }

        if (num <= 1) {
            isPrime = false;
        }
        else {
            for (let i = 2; i <= Math.sqrt(num); i++) {
                if (num % i === 0) {
                    isPrime = false;
                    break;
                }
```

```
            }
        }

        if (isPrime) {
            document.getElementById("output").innerHTML = num + " is a Prime Number.";
        }
        else {
            document.getElementById("output").innerHTML = num + " is not a Prime
Number.";
        }
    }

    function findFactorial() {
        let num = getNumber();
        let fact = 1;

        if (isNaN(num) || num < 0) {
            document.getElementById("output").innerHTML = "Please enter a positive
number.";
            return;
        }

        for (let i = 1; i <= num; i++) {
            fact = fact * i;
        }
        document.getElementById("output").innerHTML =
            "Factorial of " + num + " is " + fact + ".";
    }
    function checkArmstrong() {
        let num = getNumber();
```

```
        if (isNaN(num)) {
            document.getElementById("output").innerHTML = "Please enter a valid number.";
            return;
        }
        let original = num;
        let sum = 0;
        let digits = num.toString().length;

        while (num > 0) {
            let digit = num % 10;
            sum = sum + Math.pow(digit, digits);
            num = Math.floor(num / 10);
        }

        if (sum === original) {
            document.getevementById("output").innerHTML =
                original + " is an Armstrong Number.";
        }
        else {
            document.getElementById("output").innerHTML =
                original + " is not an Armstrong Number.";
        }
    }
  </script>

</body>
</html>
```

Output:



Explanation of Tags:

1.
    - Declares the document as an HTML5 document.
2.
    - Root element of the webpage.
    - All HTML code is written inside this tag.
3.
    - Contains metadata, title, and CSS styles.
4.
    - Specifies the title displayed on the browser tab.
5.
    - Contains CSS code used for designing the webpage.
6.
    - Contains all visible contents of the webpage.
7.
    - Used as a container to group webpage elements.
8.

- Displays the main heading of the webpage.

9.

- Displays paragraph text or instructions.

10.

- Accepts user input.
- type="number" allows only numeric values.
- id="numberInput" is used to access the input using JavaScript.

11.

- Inserts a line break.

12.

- Creates clickable buttons.
- onclick attribute calls the corresponding JavaScript function when clicked.

13.

- Contains JavaScript code for program execution.

JavaScript Functions

1. **getNumber()**
   - Reads the number entered by the user from the input box.
2. **findEven()**
   - Checks whether the entered number is even using num % 2 == 0.
3. **findOdd()**
   - Checks whether the entered number is odd using num % 2 != 0.
4. **checkPrime()**
   - Determines whether the number is prime by checking divisibility from 2 to the square root of the number.
5. **findFactorial()**
   - Calculates the factorial of a number using a loop.
6. **checkArmstrong()**
   - Checks whether a number is an Armstrong number by summing the powers of its digits and comparing the result with the original number.

WEEK-10

Aim10a: Write a Javascript conditional statement to find thesign of product of three numbers using functions.Display an alert box with specified sign.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Sign of Product</title>

  <style>
    body {
      font-family: "Segoe UI", Arial, sans-serif;
      background: linear-gradient(135deg, #36d1dc, #5b86e5);
      height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
    }

    .box {
      background: white;
      padding: 30px;
      border-radius: 12px;
      width: 320px;
      box-shadow: 0 8px 25px rgba(0,0,0,0.2);
      text-align: center;
      transition: 0.3s;
    }
```

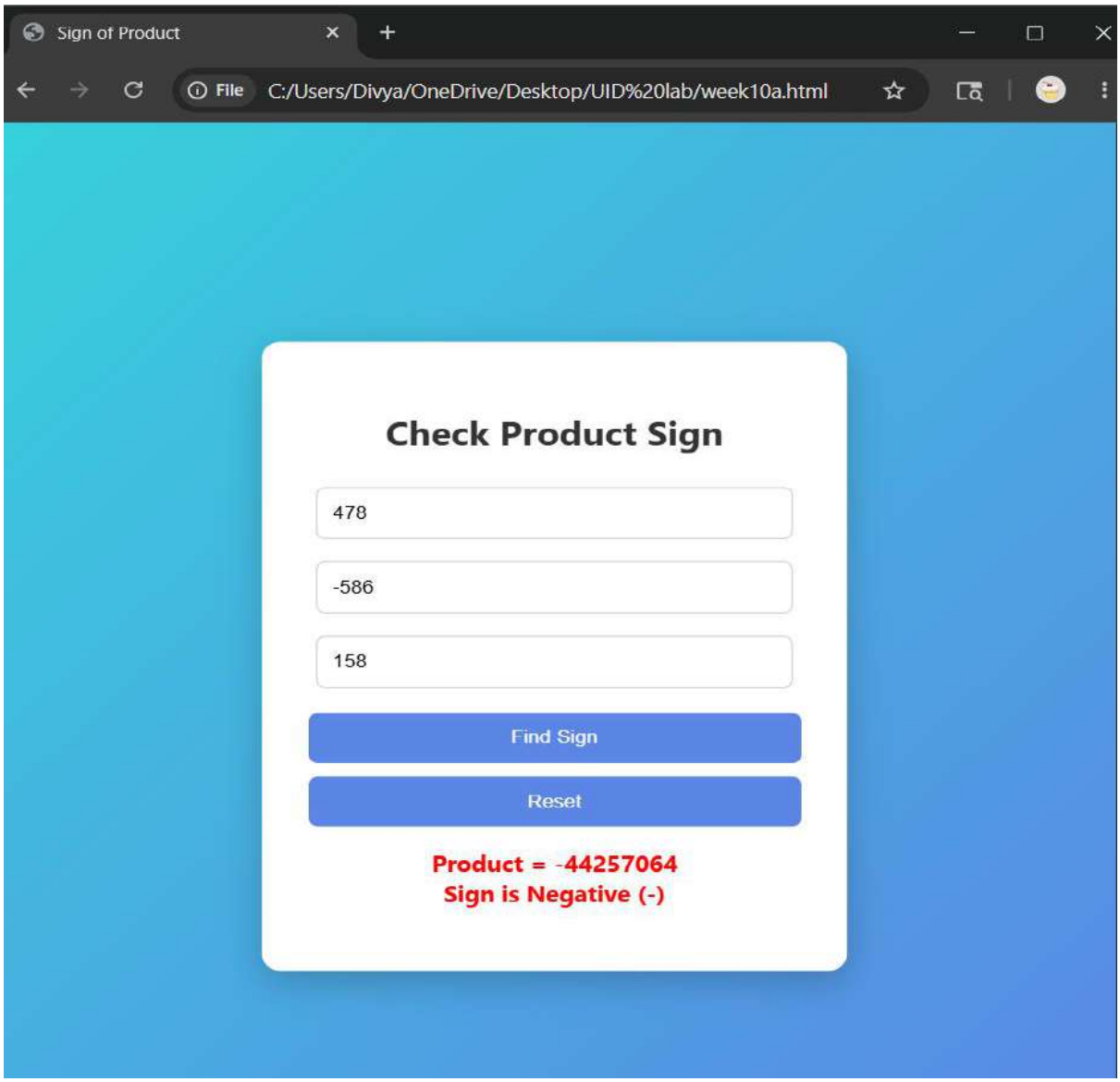
```
.box:hover {
    transform: scale(1.03);
}

h2 {
    margin-bottom: 15px;
    color: #333;
}

input {
    width: 90%;
    padding: 10px;
    margin: 8px 0;
    border: 1px solid #ccc;
    border-radius: 6px;
    transition: 0.3s;
}

input:focus {
    border-color: #5b86e5;
    outline: none;
    box-shadow: 0 0 6px rgba(91,134,229,0.4);
}

button {
    margin-top: 10px;
    padding: 10px;
    width: 100%;
    border: none;
```

```
            border-radius: 6px;
            background: #5b86e5;
            color: white;
            cursor: pointer;
            transition: 0.3s;
        }

        button:hover {
            background: #3f6fd1;
        }

        #result {
            margin-top: 15px;
            font-weight: bold;
            transition: 0.3s;
        }
    </style>
</head>

<body>

<div class="box">
    <h2>Check Product Sign</h2>

    <input type="number" id="num1" placeholder="Enter first number">
    <input type="number" id="num2" placeholder="Enter second number">
    <input type="number" id="num3" placeholder="Enter third number">

    <button onclick="checkSign()">Find Sign</button>
```

```
    <button onclick="resetFields()">Reset</button>

    <p id="result"></p>
</div>

<script>

function checkSign() {
    let a = document.getElementById("num1").value;
    let b = document.getElementById("num2").value;
    let c = document.getElementById("num3").value;
    let res = document.getElementById("result");

    // Validation
    if (a === "" || b === "" || c === "") {
        alert("Please enter all numbers");
        return;
    }

    let product = a * b * c;

    // Logic
    if (product > 0) {
        res.innerHTML = "Product = " + product + "<br>Sign is Positive (+)";
        res.style.color = "green";
        alert("The sign is plus (+)");
    }
    else if (product < 0) {
        res.innerHTML = "Product = " + product + "<br>Sign is Negative (-)";
```

```
        res.style.color = "red";
        alert("The sign is minus (-)");
    }
    else {
        res.innerHTML = "Product = 0<br>Sign is Zero (0)";
        res.style.color = "orange";
        alert("The product is zero (0)");
    }
}

// Reset function
function resetFields() {
    document.getElementById("num1").value = "";
    document.getElementById("num2").value = "";
    document.getElementById("num3").value = "";
    document.getElementById("result").innerHTML = "";
}

// Enter key support
document.addEventListener("keydown", function(e) {
    if (e.key === "Enter") {
        checkSign();
    }
});

</script>

</body>
</html>
```

Output:



Explanation of tags:

Additional Tags and Attributes Used in the Program

1. **class Attribute**
2.
    - The class attribute is used to apply CSS styles to an element.
    - Here, the class name box is used to create a styled container.
3. **id Attribute**
4. <input type="number" id="num1">
    - The id attribute uniquely identifies an HTML element.

- JavaScript uses this id to access the element.

5. **placeholder Attribute**
6. placeholder="Enter first number"
    - Displays a hint inside the input field.
    - The text disappears when the user starts typing.
7. **type="number"**
8. <input type="number">
    - Restricts input to numeric values only.
    - Prevents entering text characters.
9. **onclick Attribute**
10. <button onclick="checkSign()">
    - Executes a JavaScript function when the button is clicked.
11. **Paragraph Tag with ID**
12.
    - Used to display the output dynamically.
    - JavaScript updates its content using the id "result".

New CSS Properties Used

1. **background: linear-gradient()**
2. background: linear-gradient(135deg,#36d1dc,#5b86e5);
    - Creates a gradient background with multiple colors.
3. **display: flex**
4. display: flex;
    - Enables Flexbox layout for easy alignment.
5. **justify-content: center**
6. justify-content: center;
    - Centers content horizontally.
7. **align-items: center**
8. align-items: center;

- Centers content vertically.

9. **border-radius**
10. border-radius: 12px;
    - Creates rounded corners.
11. **box-shadow**
12. box-shadow: 0 8px 25px rgba(0,0,0,0.2);
    - Adds shadow around the box.
13. **transform: scale()**
14. transform: scale(1.03);
    - Slightly enlarges the box when hovered.
15. **transition**
16. transition: 0.3s;
    - Creates smooth animation effects.

New JavaScript Methods Used

1. **document.getElementById()**
    - Accesses an HTML element using its id.
2. **.value**
    - Retrieves the value entered by the user.
3. **.innerHTML**
    - Changes or displays content inside an HTML element.
4. **alert()**
    - Displays a popup message.
5. **addEventListener()**
    - Attaches an event to an element.
    - Used here to detect the Enter key.
6. **e.key**
    - Identifies which keyboard key was pressed.

Aim10b: Write a Javascript conditional statement to sort three numbers using functions .Display an alert box to show the result

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Sort Three Numbers</title>

  <style>
    body {
      font-family: "Segoe UI", Arial, sans-serif;
      background: linear-gradient(135deg, #ff9a9e, #fad0c4);
      height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
    }

    .box {
      background: white;
      padding: 30px;
      border-radius: 12px;
      width: 320px;
      box-shadow: 0 8px 25px rgba(0,0,0,0.2);
      text-align: center;
    }
    .box:hover {
      transform: scale(1.03);
    }
```

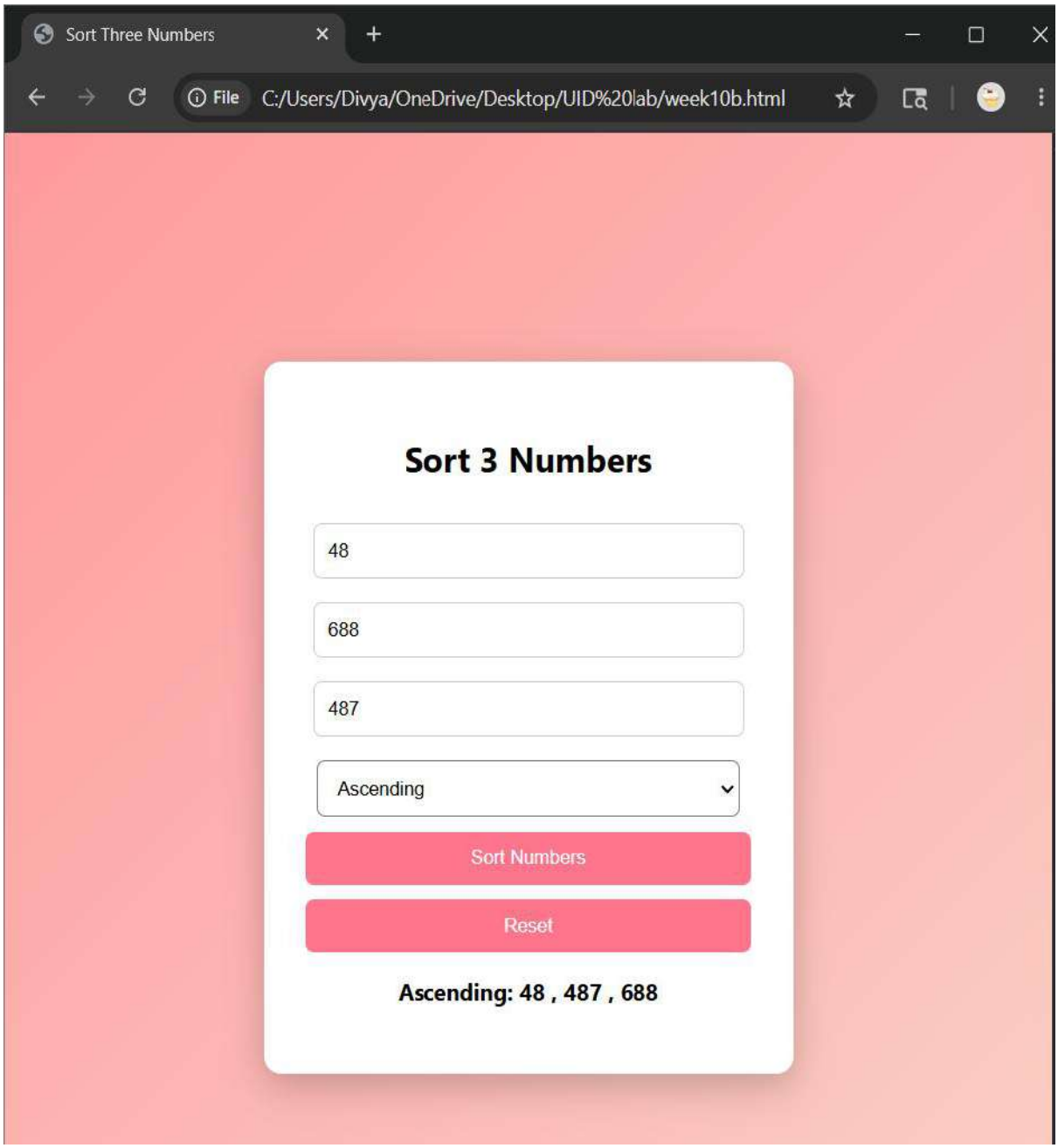
```
input {
    width: 90%;
    padding: 10px;
    margin: 8px 0;
    border: 1px solid #ccc;
    border-radius: 6px;
}
input:focus {
    border-color: #ff758c;
    outline: none;
    box-shadow: 0 0 6px rgba(255,117,140,0.4);
}
select {
    width: 95%;
    padding: 10px;
    margin-top: 8px;
    border-radius: 6px;
}

button {
    margin-top: 10px;
    padding: 10px;
    width: 100%;
    border: none;
    border-radius: 6px;
    background: #ff758c;
    color: white;
    cursor: pointer;
```

```
        }

        button:hover {
            background: #e64965;
        }

        #result {
            margin-top: 15px;
            font-weight: bold;
        }
    </style>
</head>

<body>

<div class="box">
    <h2>Sort 3 Numbers</h2>

    <input type="number" id="num1" placeholder="Enter first number">
    <input type="number" id="num2" placeholder="Enter second number">
    <input type="number" id="num3" placeholder="Enter third number">

    <!-- Sorting Option -->
    <select id="order">
        <option value="desc">Descending</option>
        <option value="asc">Ascending</option>
    </select>

    <button onclick="sortNumbers()">Sort Numbers</button>
```

```
    <button onclick="resetFields()">Reset</button>

    <p id="result"></p>
</div>

<script>

function sortNumbers() {
    let a = parseFloat(document.getElementById("num1").value);
    let b = parseFloat(document.getElementById("num2").value);
    let c = parseFloat(document.getElementById("num3").value);
    let order = document.getElementById("order").value;
    let res = document.getElementById("result");

    if (isNaN(a) || isNaN(b) || isNaN(c)) {
        alert("Please enter all numbers");
        return;
    }
    let largest, middle, smallest;
    // Same logic
    if (a >= b && a >= c) {
        largest = a;
        if (b >= c) { middle = b; smallest = c; }
        else { middle = c; smallest = b; }
    }
    else if (b >= a && b >= c) {
        largest = b;
        if (a >= c) { middle = a; smallest = c; }
        else { middle = c; smallest = a; }
```

```
    }
    else {
        largest = c;
        if (a >= b) { middle = a; smallest = b; }
        else { middle = b; smallest = a; }
    }

    let output;

    if (order === "asc") {
        output = "Ascending: " + smallest + " , " + middle + " , " + largest;
    } else {
        output = "Descending: " + largest + " , " + middle + " , " + smallest;
    }

    res.innerHTML = output;
    alert(output);
}

function resetFields() {
    document.getElementById("num1").value = "";
    document.getElementById("num2").value = "";
    document.getElementById("num3").value = "";
    document.getElementById("result").innerHTML = "";
}

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

Output:



Explanation of tags:

Explanation of New Tags and Attributes Used in the Program

1. **Tag**
2. <select id="order">
    - Creates a drop-down list.
    - Allows the user to choose a sorting order.
    - Here it is used to select Ascending or Descending order.

3. **Tag**
4. <option value="desc">Descending</option>
5. <option value="asc">Ascending</option>
    - Defines items inside the drop-down list.
    - The value attribute stores the value selected by the user.
6. **value Attribute**
7. value="asc"
    - Stores the actual value associated with an option.
    - JavaScript reads this value to determine the sorting order.

Additional CSS Concepts Used

1. **input:focus**
2. input:focus
    - Applies styles when the input field is selected.
    - Changes border color and shadow effect.
3. **select Styling**
4. select {
5. width:95%;
6. padding:10px;
7. }
    - Styles the drop-down menu.
    - Makes it consistent with input fields.

Additional JavaScript Functions and Methods

1. **parseFloat()**
2. parseFloat(value)
    - Converts input values into decimal numbers.
    - Allows sorting of both integers and decimal values.
3. **isNaN()**

4. isNaN(a)
    - Checks whether the entered value is a valid number.
    - Returns true if the value is not a number.
5. **Logical AND Operator (&&)**
6. a >= b && a >= c
    - Checks multiple conditions together.
    - Both conditions must be true.
7. **else if Statement**
8. else if (b >= a && b >= c)
    - Tests another condition if the previous condition is false.

Program Logic

1. User enters three numbers.
2. User selects Ascending or Descending order from the drop-down menu.
3. The program finds:
    - Largest number
    - Middle number
    - Smallest number
4. Based on the selected option:
    - Ascending → Smallest to Largest
    - Descending → Largest to Smallest
5. The sorted result is displayed on the webpage and in an alert box.

Aim10c: .Write a Javascript conditional statement to find the largest of five numbers using function and onclick event.Display an alert box to show result.

Program:

```
<!DOCTYPE html>
<html>
<head>
   <title>Largest of Five Numbers</title>

   <style>
      body {
         font-family: "Segoe UI", Arial, sans-serif;
         background: linear-gradient(135deg, #84fab0, #8fd3f4);
         height: 100vh;
         display: flex;
         justify-content: center;
         align-items: center;
      }

      .box {
         background: white;
         padding: 30px;
         border-radius: 12px;
         width: 330px;
         box-shadow: 0 8px 25px rgba(0,0,0,0.2);
         text-align: center;
         transition: 0.3s;
      }
```

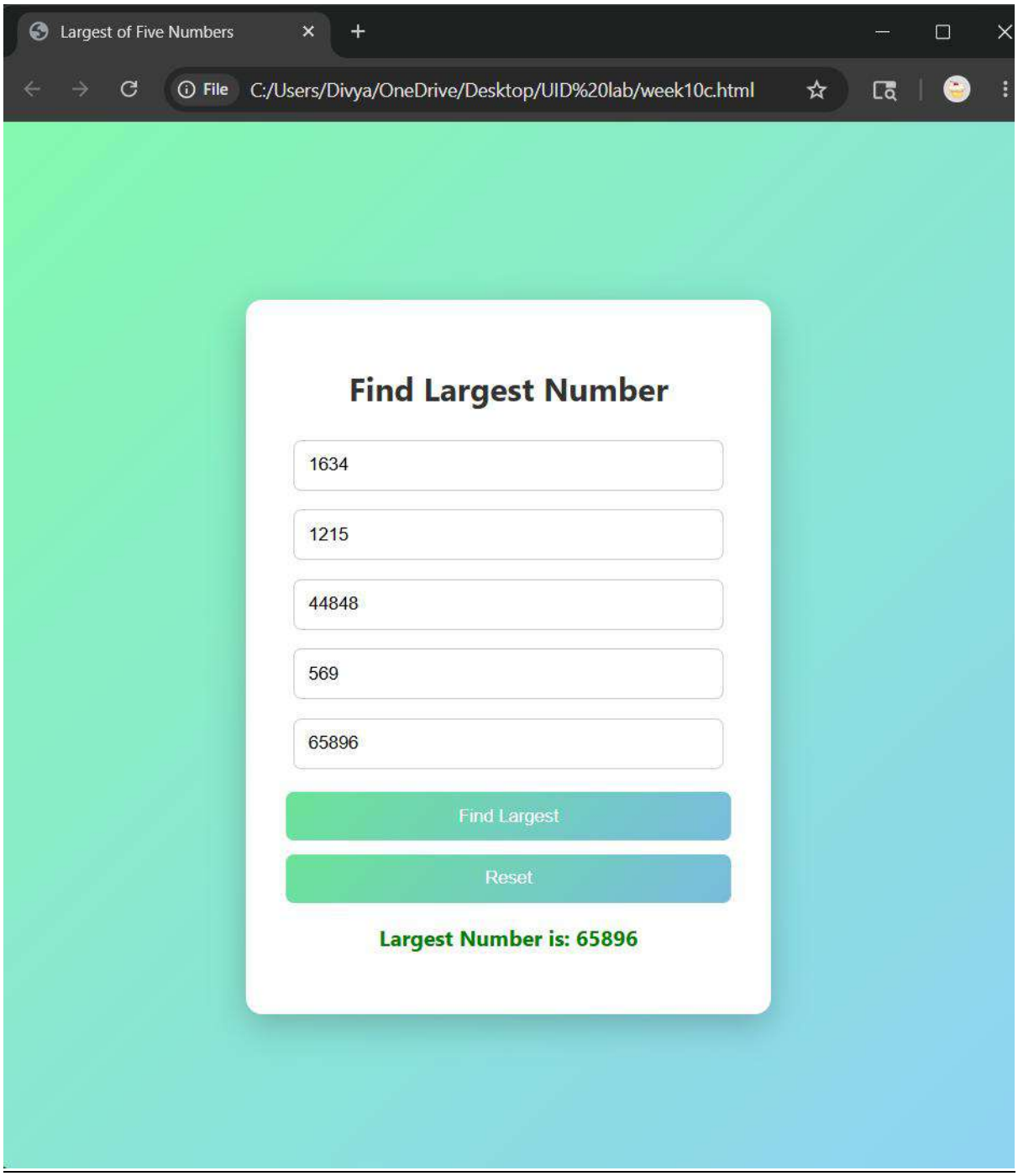
```
.box:hover {
    transform: scale(1.03);
}

h2 {
    margin-bottom: 15px;
    color: #333;
}

input {
    width: 90%;
    padding: 10px;
    margin: 7px 0;
    border: 1px solid #ccc;
    border-radius: 6px;
    transition: 0.3s;
}

input:focus {
    border-color: #4caf50;
    outline: none;
    box-shadow: 0 0 6px rgba(76,175,80,0.4);
}

button {
    margin-top: 10px;
    padding: 10px;
    width: 100%;
    border: none;
```

```
            border-radius: 6px;
            background:  linear-gradient(135deg, #6ce398, #78bcdf);
            color: white;
            cursor: pointer;
            transition: 0.3s;
        }

        button:hover {
            background:   linear-gradient(135deg, #50be78, #4e85a0);
        }

        #result {
            margin-top: 15px;
            font-weight: bold;
        }
    </style>
</head>

<body>

<div class="box">
    <h2>Find Largest Number</h2>

    <input type="number" id="num1" placeholder="Enter number 1">
    <input type="number" id="num2" placeholder="Enter number 2">
    <input type="number" id="num3" placeholder="Enter number 3">
    <input type="number" id="num4" placeholder="Enter number 4">
    <input type="number" id="num5" placeholder="Enter number 5">
```

```
    <button onclick="findLargest()">Find Largest</button>
    <button onclick="resetFields()">Reset</button>

    <p id="result"></p>
</div>

<script>

function findLargest() {
    let a = parseFloat(document.getElementById("num1").value);
    let b = parseFloat(document.getElementById("num2").value);
    let c = parseFloat(document.getElementById("num3").value);
    let d = parseFloat(document.getElementById("num4").value);
    let e = parseFloat(document.getElementById("num5").value);

    let res = document.getElementById("result");
    // Validation
    if (isNaN(a) || isNaN(b) || isNaN(c) || isNaN(d) || isNaN(e)) {
        alert("Please enter all numbers");
        return;
    }
    // Conditional logic
    let largest;

    if (a >= b && a >= c && a >= d && a >= e) {
        largest = a;
    }
    else if (b >= a && b >= c && b >= d && b >= e) {
        largest = b;
```

```
    }
    else if (c >= a && c >= b && c >= d && c >= e) {
        largest = c;
    }
    else if (d >= a && d >= b && d >= c && d >= e) {
        largest = d;
    }
    else {
        largest = e;
    }
    let output = "Largest Number is: " + largest;
    res.innerHTML = output;
    res.style.color = "green";
    alert(output);
}
// Reset function
function resetFields() {
    document.getElementById("num1").value = "";
    document.getElementById("num2").value = "";
    document.getElementById("num3").value = "";
    document.getElementById("num4").value = "";
    document.getElementById("num5").value = "";
    document.getElementById("result").innerHTML = "";
}

</script>

</body>
</html>
```

Output:



Explanation of tags:

1. **Multiple Input Tags**
2. <input type="number" id="num1">

3. <input type="number" id="num2">
4. <input type="number" id="num3">
5. <input type="number" id="num4">
6. <input type="number" id="num5">
    - Five input fields are used to accept five numbers from the user.
    - Each input field has a unique id for identification in JavaScript.
7. **placeholder Attribute**
8. placeholder="Enter number 1"
    - Displays hint text inside the input box.
    - Helps the user understand what to enter.
9. **onclick Attribute**
10. <button onclick="findLargest()">
    - Calls the JavaScript function when the button is clicked.
11. **Result Paragraph**
12.
    - Used to display the largest number dynamically.

Additional CSS Properties Used

1. **Linear Gradient Background**
2. background: linear-gradient(135deg, #84fab0, #8fd3f4);
    - Creates a smooth color transition in the background.
3. **Transition Property**
4. transition: 0.3s;
    - Creates smooth animation effects.
5. **Hover Effect**
6. .box:hover
    - Executes styles when the mouse pointer moves over the box.
7. **Transform Property**
8. transform: scale(1.03);

- Enlarges the container slightly when hovered.

JavaScript Functions Used

1. **parseFloat()**
2. parseFloat(value)
   - Converts input values into decimal numbers.
3. **isNaN()**
4. isNaN(a)
   - Checks whether the entered value is a valid number.
5. **Logical AND Operator (&&)**
6. a >= b && a >= c && a >= d && a >= e
   - Ensures all conditions are true before executing the statement.
7. **style.color**
8. res.style.color = "green";
   - Changes the color of the displayed result.

Program Logic

1. User enters five numbers.
2. The program validates that all fields contain numbers.
3. Using if-else conditions, each number is compared with the others.
4. The largest value is stored in the variable largest.
5. The result is displayed on the webpage and in an alert box.

Example:

Input:
10, 45, 25, 60, 35

Output:
Largest Number is: 60

Aim10d: Write a Javascript for loop that will iterate from 0 to 15.For each iteration, it will check if the current number is odd or even and display a message to the screen.

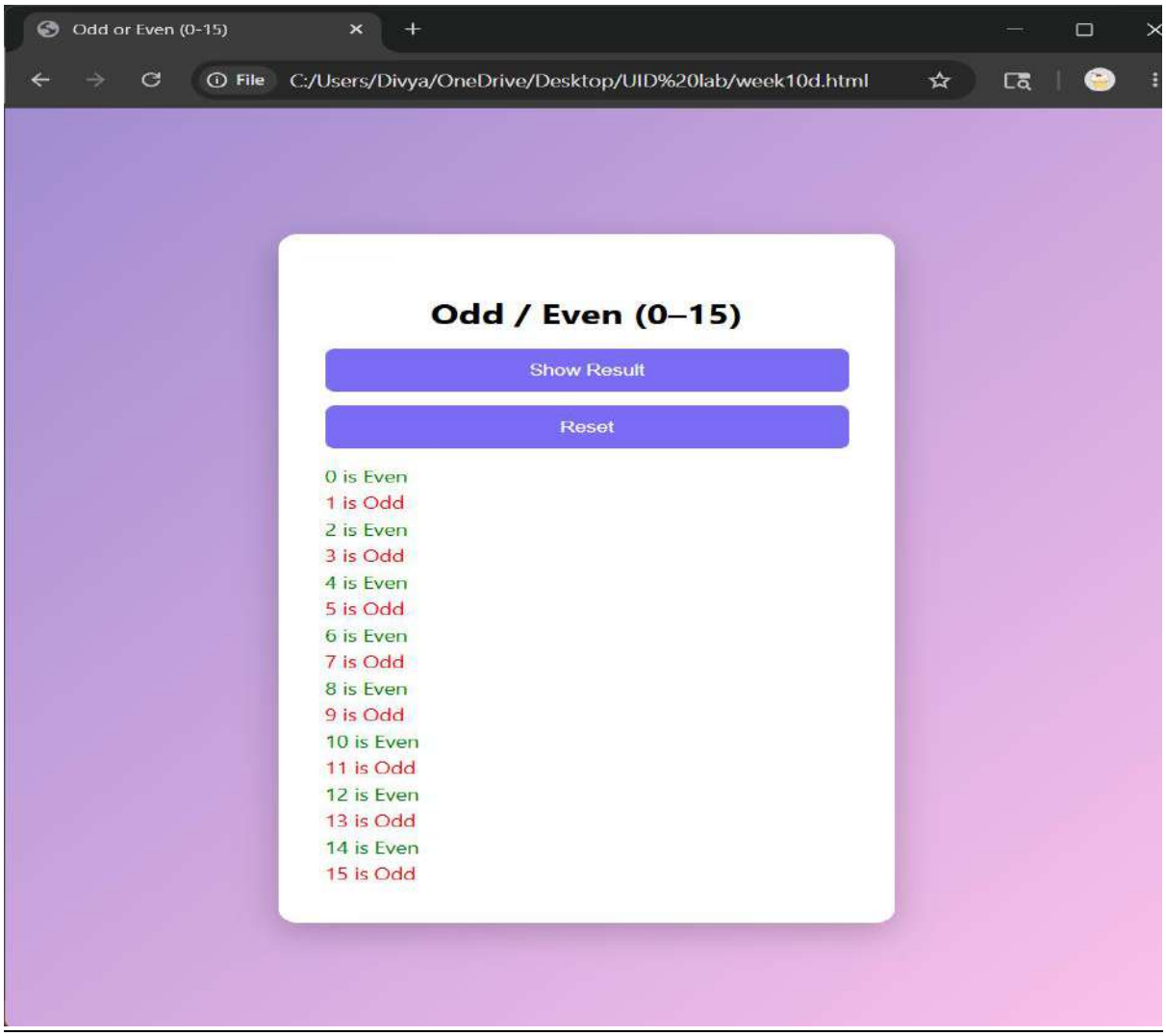
Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Odd or Even (0-15)</title>

  <style>
    body {
      font-family: "Segoe UI", Arial, sans-serif;
      background: linear-gradient(135deg, #a18cd1, #fbc2eb);
      height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
    }

    .box {
      background: white;
      padding: 30px;
      border-radius: 12px;
      width: 340px;
      box-shadow: 0 8px 25px rgba(0,0,0,0.2);
      text-align: center;
    }

    h2 {
      margin-bottom: 15px;
```

```
        }

        button {
            padding: 10px;
            width: 100%;
            border: none;
            border-radius: 6px;
            background: #7b6cf6;
            color: white;
            cursor: pointer;
            margin-bottom: 12px;
        }
        button:hover {
            background: #5b4be0;
        }

        #result {
            text-align: left;
            font-size: 14px;
            line-height: 1.6;
        }
        .even { color: green; }
        .odd { color: red; }
    </style>
</head>

<body>

<div class="box">
```

```
    <h2>Odd / Even (0–15)</h2>

    <button onclick="checkNumbers()">Show Result</button>
    <button onclick="resetResult()">Reset</button>

    <div id="result"></div>
</div>
<script>
function checkNumbers() {
    let output = "";
    let res = document.getElementById("result");
    for (let i = 0; i <= 15; i++) {
        if (i % 2 === 0) {
            output += "<span class='even'>" + i + " is Even</span><br>";
        } else {
            output += "<span class='odd'>" + i + " is Odd</span><br>";
        }
    }
    res.innerHTML = output;
    alert("Check results displayed on screen");
}
function resetResult() {
    document.getElementById("result").innerHTML = "";
}

</script>

</body>
</html>
```

Output:



Explanation of tags:

1.
2.
    - Used to display multiple output lines.
    - Unlike the <p> tag, the tag can easily contain several formatted lines and HTML elements.
3. **class Attribute**
4.
5.
    - Used to apply different CSS styles.

- Here, even numbers are displayed in green and odd numbers in red.

6. **Tag**
7. 0 is Even
    - An inline HTML element.
    - Used to apply styles to specific text without creating a new line.
8. **Tag**
9.
    - Inserts a line break.
    - Displays each result on a separate line.

Additional CSS Properties Used

1. **line-height**
2. line-height: 1.6;
    - Controls spacing between lines.
    - Improves readability of the output.
3. **Text Alignment**
4. text-align: left;
    - Aligns the output text to the left side.
5. **Class Selectors**
6. .even { color: green; }
7. .odd { color: red; }
    - Styles elements based on their class names.
    - Even numbers appear green.
    - Odd numbers appear red.

JavaScript Concepts Used

1. **for Loop**
2. for(let i = 0; i <= 15; i++)
    - Repeats a block of code multiple times.

    - Here it checks numbers from 0 to 15.
3. **Modulo Operator (%)**
4. i % 2 === 0
    - Finds the remainder after division.
    - If remainder is 0, the number is even.
5. **String Concatenation**
6. output +=
    - Appends new text to the existing output string.
7. **innerHTML**
8. res.innerHTML = output;
    - Displays formatted HTML content inside the result division.

Program Logic

1. User clicks the "Show Result" button.
2. A for loop runs from 0 to 15.
3. Each number is checked:
    - If divisible by 2 → Even.
    - Otherwise → Odd.
4. Results are stored in a string.
5. Even numbers are displayed in green.
6. Odd numbers are displayed in red.
7. The output is shown inside the result area.

Example Output:

0 is Even

1 is Odd

2 is Even

3 is Odd

...

15 is Odd

Aim10e: Write a Javascript program to read 6 subjects marks from the student then compute the average marks of the student.Then this average is used to determine corresponding grade

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>Student Grade Calculator</title>

  <style>
    body {
      font-family: Arial, sans-serif;
      background: linear-gradient(135deg,rgb(245, 245, 162),rgb(167, 201, 245));
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      margin: 0;
    }

    .container {
      background: #fff;
      padding: 30px;
      border-radius: 15px;
      width: 350px;
      text-align: center;
      box-shadow: 0 10px 25px rgba(0,0,0,0.3);
    }
```

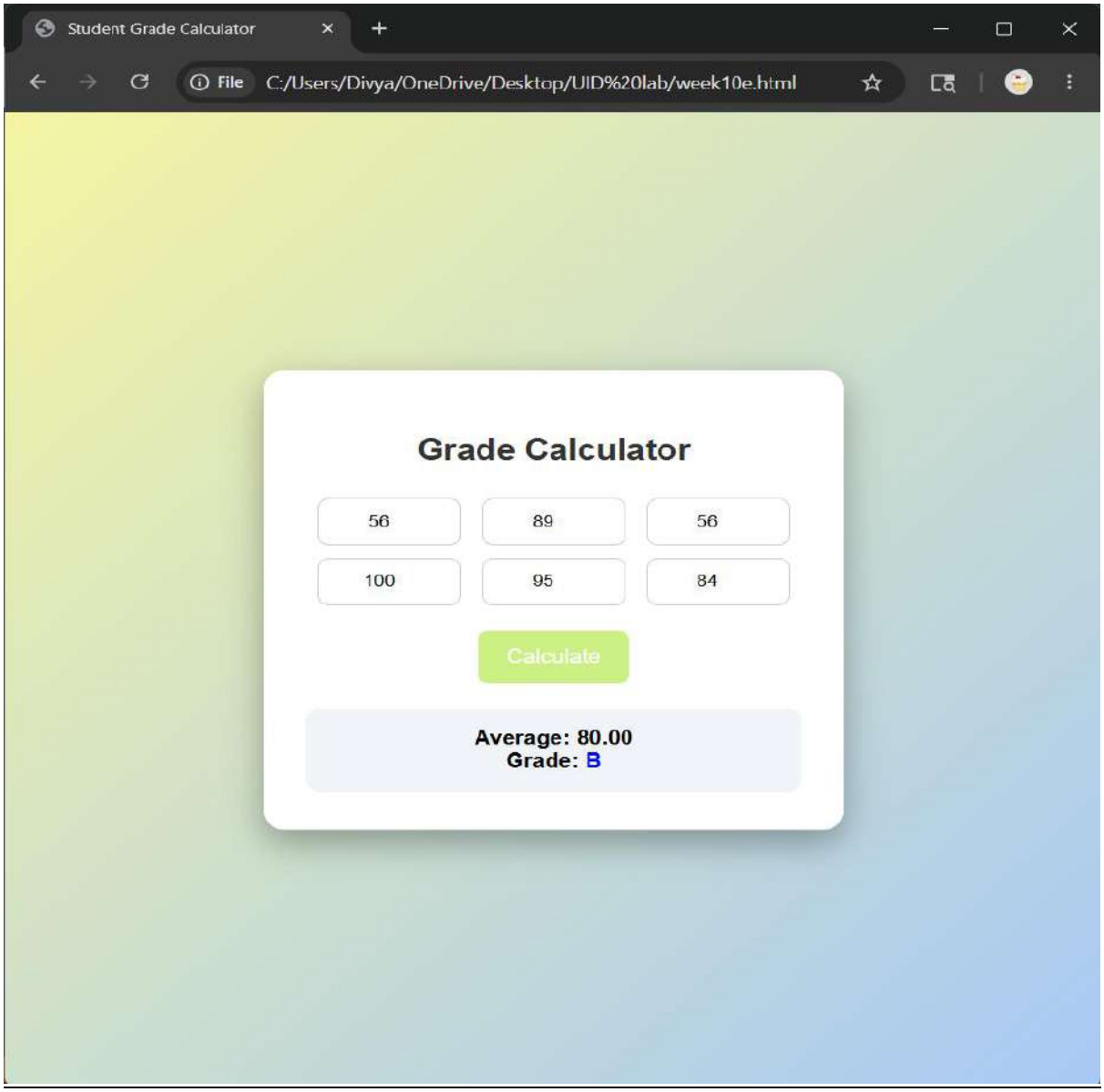
```
    h2 {
        margin-bottom: 20px;
        color: #333;
    }

    .inputs input {
        width: 80px;
        padding: 10px;
        margin: 5px;
        border-radius: 8px;
        border: 1px solid #ccc;
        text-align: center;
        transition: 0.3s;
    }

    .inputs input:focus {
        border-color: #667eea;
        outline: none;
    }


button {
        margin-top: 15px;
        padding: 12px 20px;
        border: none;
        border-radius: 8px;
        background: #cbf185;
        color: white;
        font-size: 16px;
```

```
            cursor: pointer;
            transition: 0.3s;
        }

        button:hover {
            background: #5a67d8;
        }

        #output {
            margin-top: 20px;
            padding: 15px;
            border-radius: 10px;
            background: #f1f5f9;
            font-weight: bold;
        }



        .grade-A { color: green; }
        .grade-B { color: blue; }
        .grade-C { color: orange; }
        .grade-D { color: brown; }
        .grade-F { color: red; }
    </style>
</head>

<body>

    <div class="container">
        <h2> Grade Calculator</h2>
```

```
    <div class="inputs">
        <input type="number" id="m1" placeholder="S1">
        <input type="number" id="m2" placeholder="S2">
        <input type="number" id="m3" placeholder="S3"><br>
        <input type="number" id="m4" placeholder="S4">
        <input type="number" id="m5" placeholder="S5">
        <input type="number" id="m6" placeholder="S6">
    </div>

    <button onclick="calculateGrade()">Calculate</button>

    <div id="output">Enter marks and click Calculate</div>
  </div>

  <script>

   function calculateGrade() {

let m1 = Number(document.getElementById("m1").value);
        let m2 = Number(document.getElementById("m2").value);
        let m3 = Number(document.getElementById("m3").value);
        let m4 = Number(document.getElementById("m4").value);
        let m5 = Number(document.getElementById("m5").value);
        let m6 = Number(document.getElementById("m6").value);

        let avg = (m1 + m2 + m3 + m4 + m5 + m6) / 6;
        let grade = "";
        let gradeClass = "";
```

```
            if (avg < 60) {
                grade = "F";
                gradeClass = "grade-F";
            } else if (avg < 70) {
                grade = "D";
                gradeClass = "grade-D";
            } else if (avg < 80) {
                grade = "C";
                gradeClass = "grade-C";
            } else if (avg < 90) {
                grade = "B";
                gradeClass = "grade-B";
            } else if (avg <= 100) {
                grade = "A";
                gradeClass = "grade-A";
            } else {
                grade = "Invalid";
            }



            document.getElementById("output").innerHTML =
                "Average: " + avg.toFixed(2) +
                "<br>Grade: <span class='" + gradeClass + "'>" + grade + "</span>";
        }
    </script>

</body>
</html>
```

Output:



Explanation of tags:

Explanation of Tags and Concepts Used in the Program

1.
2.
    - Groups all subject mark input fields together.
    - Makes it easier to apply common CSS styles.
3. **Placeholder Values**
4. placeholder="S1"
    - Indicates Subject 1, Subject 2, etc.

- Helps users know where to enter marks.

5. **Output Division**
6.
    - Displays the calculated average and grade.
    - Updated dynamically using JavaScript.

Additional CSS Concepts Used

1. **Class Selectors for Grades**
2. .grade-A { color: green; }
3. .grade-B { color: blue; }
4. .grade-C { color: orange; }
5. .grade-D { color: brown; }
6. .grade-F { color: red; }
    - Different colors are assigned to different grades.
    - Improves readability of the result.
7. **Box Shadow**
8. box-shadow: 0 10px 25px rgba(0,0,0,0.3);
    - Adds shadow around the container.
    - Gives a modern card-like appearance.

JavaScript Functions and Methods Used

1. **Number() Function**
2. Number(value)
    - Converts the input value into a numeric value.
    - Required for performing calculations.
3. **Average Calculation**
4. avg = (m1+m2+m3+m4+m5+m6)/6
    - Calculates the average marks of six subjects.
5. **toFixed() Method**

6. avg.toFixed(2)
    - Displays the average with two decimal places.
    - Example:
      85.6666 → 85.67
7. **else if Ladder**
8. else if(avg < 70)
    - Used to check multiple grade ranges.

Grade Criteria

- Average below 60 → Grade F
- Average 60 to 69 → Grade D
- Average 70 to 79 → Grade C
- Average 80 to 89 → Grade B
- Average 90 to 100 → Grade A

Program Logic

1. User enters marks for six subjects.
2. JavaScript converts the inputs into numbers.
3. The average marks are calculated.
4. Based on the average, a grade is assigned.
5. The grade is displayed in a specific color.
6. The average and grade are shown on the screen.

Example:

Input:
80, 85, 90, 88, 92, 95

Average:
88.33

Output:
Grade B

WEEK-11

Aim11a: A website requires users to enter a username. The system should display a message while typing if the username is less than 5 characters

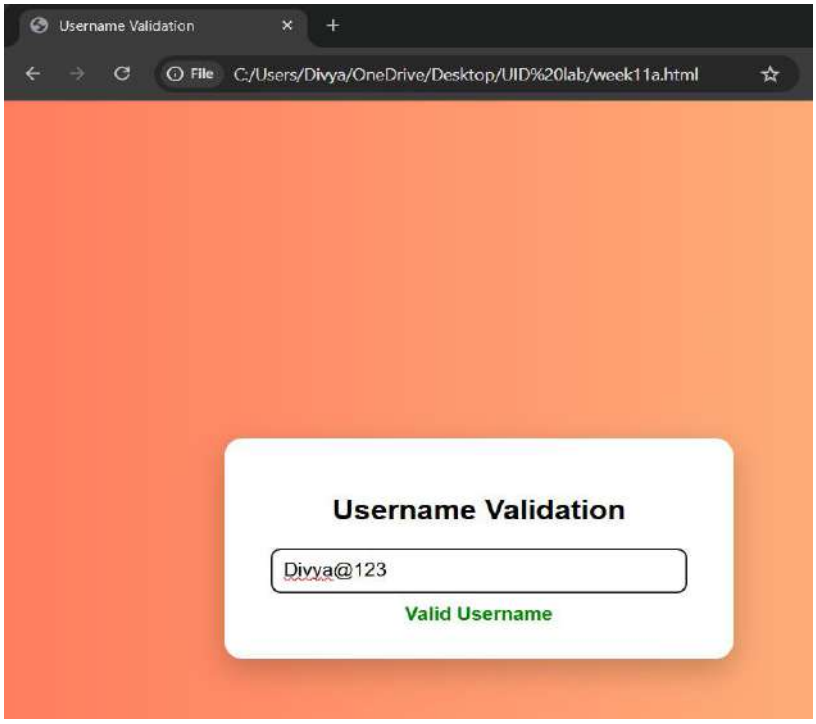
Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Username Validation</title>

<style>
  body {
    font-family: Arial, sans-serif;
    background: linear-gradient(to right, #ff7e5f, #feb47b);
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }

  .container {
    background: white;
    padding: 30px;
    border-radius: 15px;
    width: 350px;
    text-align: center;
    box-shadow: 0 10px 25px rgba(0,0,0,0.2);
  }
```

```
    h2 {
        margin-bottom: 20px;
    }

    input {
        width: 90%;
        padding: 10px;
        font-size: 16px;
        border-radius: 8px;
        border: 1px solid #ccc;
    }

    .message {
        margin-top: 10px;
        font-weight: bold;
    }

    .error {
        color: red;
    }
    .success {
        color: green;
    }
</style>
</head>
<body>

<div class="container">
```

```
    <h2>Username Validation</h2>

    <input type="text" id="username" placeholder="Enter username"
onkeyup="validateUsername()">

    <div id="message" class="message"></div>
</div>
<script>
function validateUsername() {
    let username = document.getElementById("username").value;
    let message = document.getElementById("message");

    if (username.length === 0) {
        message.innerHTML = "";
    }
    else if (username.length < 5) {
        message.innerHTML = "Username must be at least 5 characters";
        message.className = "message error";
    }
    else {
        message.innerHTML = "Valid Username";
        message.className = "message success";
    }
}
</script>

</body>
</html>
```

Output:



Explanation of tags:

Explanation of Tags and Concepts Used in the Program

1. **lang Attribute**
2. <html lang="en">
    - Specifies that the webpage language is English.
    - Helps browsers and search engines understand the language.
3. **meta Charset Tag**
4. <meta charset="UTF-8">
    - Defines the character encoding of the webpage.
    - Supports special symbols and multiple languages.
5. **type="text"**
6. <input type="text">
    - Allows the user to enter text values.
    - Used here for entering a username.
7. **onkeyup Event**
8. onkeyup="validateUsername()"

- Executes the JavaScript function whenever a key is released.
- Provides real-time validation while typing.

9. **Message Division**
10.
    - Displays validation messages.
    - The class provides styling, and the id allows JavaScript access.

Additional CSS Concepts Used

1. **Class Selector**
2. .message
    - Applies styles to all elements with class "message".
3. **Error Class**
4. .error {
5. color: red;
6. }
    - Displays error messages in red color.
7. **Success Class**
8. .success {
9. color: green;
10. }
    - Displays success messages in green color.

JavaScript Concepts Used

1. **length Property**
2. username.length
    - Returns the number of characters in the username.
3. **className Property**
4. message.className = "message error";
    - Dynamically changes the CSS class of an element.

    - Used to switch between error and success styles.
5. **Real-Time Validation**
6. validateUsername()
    - Checks the username as the user types.
    - No need to click a button.

Validation Rules

- If the username is empty → No message displayed.
- If username length is less than 5 characters → Error message shown.
- If username length is 5 or more characters → Success message shown.

Program Logic

1. User enters a username.
2. The onkeyup event calls the validation function.
3. JavaScript counts the number of characters.
4. If the username has fewer than 5 characters:
    - Error message is displayed in red.
5. If the username has 5 or more characters:
    - Success message is displayed in green.
6. The message updates instantly while typing.

Example:

Input:
abc

Output:
Username must be at least 5 characters

Input:
divya12

Output:
Valid Username

Aim11b: A registration form should validate email format before submission. If invalid, the form should not be submitted

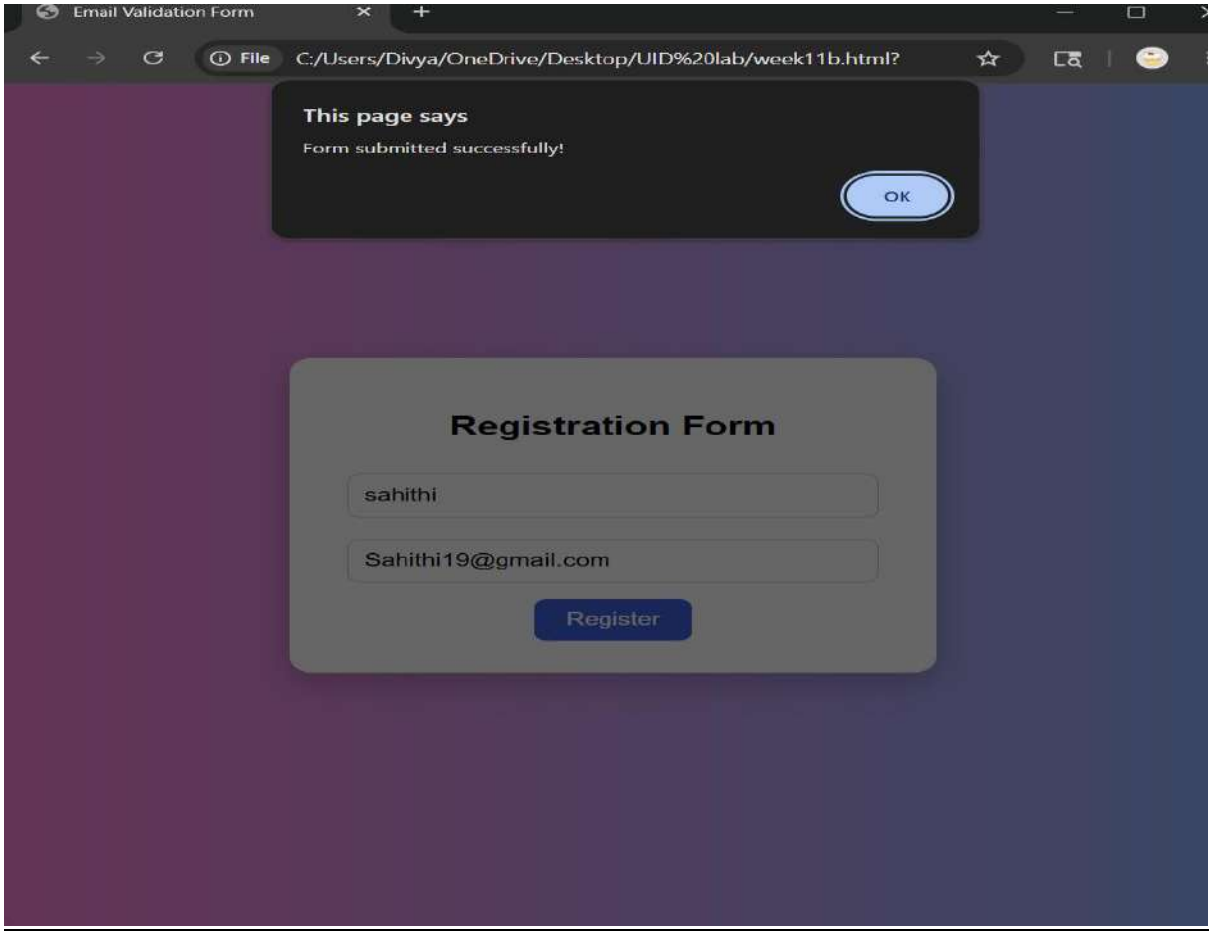
Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Email Validation Form</title>

<style>
  body {
    font-family: Arial, sans-serif;
    background: linear-gradient(to right, #36d1dc, #5b86e5);
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }
  .container {
    background: white;
    padding: 30px;
    border-radius: 15px;
    width: 350px;
    box-shadow: 0 10px 25px rgba(0,0,0,0.2);
    text-align: center;
  }
  h2 {
    margin-bottom: 20px;
  }
```

```
    input {
        width: 90%;
        padding: 10px;
        margin: 10px 0;
        border-radius: 8px;
        border: 1px solid #ccc;
        font-size: 16px;
    }
    button {
        padding: 10px 20px;
        border: none;
        border-radius: 8px;
        background-color: #5b86e5;
        color: white;
        font-size: 16px;
        cursor: pointer;
    }
    button:hover {
        background-color: #3b5edb;
    }
    .error {
        color: red;
        font-size: 14px;
        margin-top: 5px;
    }
</style>
</head>
<body>
```

```
<div class="container">
    <h2>Registration Form</h2>

    <form onsubmit="return validateForm()">
        <input type="text" placeholder="Enter Name" required><br>
        <input type="email" id="email" placeholder="Enter Email"><br>
        <div id="errorMsg" class="error"></div>
        <button type="submit">Register</button>
    </form>
</div>
<script>
function validateForm() {
    let email = document.getElementById("email").value;
    let errorMsg = document.getElementById("errorMsg");
    // Simple email regex
    let pattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;

    if (!pattern.test(email)) {
        errorMsg.innerHTML = "Please enter a valid email address";
        return false; // Stops form submission
    }
    errorMsg.innerHTML = "";
    alert("Form submitted successfully!");
    return true;
}
</script>

</body>
</html>
```

Output:



Explanation of tags:

1. **Tag**
2. <form onsubmit="return validateForm()">
    - Creates a form to collect user information.
    - The onsubmit event calls the validation function before form submission.
3. **onsubmit Event**
4. onsubmit="return validateForm()"
    - Executes when the user submits the form.
    - If the function returns false, form submission is stopped.
5. **required Attribute**
6. <input type="text" required>
    - Makes the field mandatory.
    - The form cannot be submitted if this field is empty.

7. **type="email"**
8. <input type="email">
    - Used specifically for email input.
    - Helps browsers recognize email addresses.
9. **type="submit"**
10. <button type="submit">
    - Creates a submit button.
    - Submits the form after validation.

Additional CSS Concepts Used

1. **Font Size**
2. font-size: 16px;
    - Controls the size of text.
3. **Error Message Class**
4. .error {
5. color: red;
6. }
    - Displays validation errors in red color.

JavaScript Concepts Used

1. **Regular Expression (Regex)**
2. let pattern = /^[^]+@[^]+\.[a-z]{2,3}$/;
    - Used to validate the email format.
    - Checks whether the email follows the correct structure.
3. **test() Method**
4. pattern.test(email)
    - Checks whether the email matches the pattern.
    - Returns true or false.
5. **Logical NOT Operator (!)**

6. if (!pattern.test(email))
    - Reverses the result.
    - Executes the block if the email format is invalid.
7. **return false**
8. return false;
    - Stops form submission.
9. **return true**
10. return true;
    - Allows form submission.

Program Logic

1. User enters name and email address.
2. User clicks the Register button.
3. The validateForm() function is called.
4. A regular expression checks the email format.
5. If the email is invalid:
    - Error message is displayed.
    - Form submission is stopped.
6. If the email is valid:
    - Success message is shown.
    - Form submission continues.

Example:

Input:
divya@gmail

Output:
Please enter a valid email address

Input:
divya@gmail.com

Output:
Form submitted successfully!

Aim11c: A system should check password strength while typing and display whether it is weak or strong.

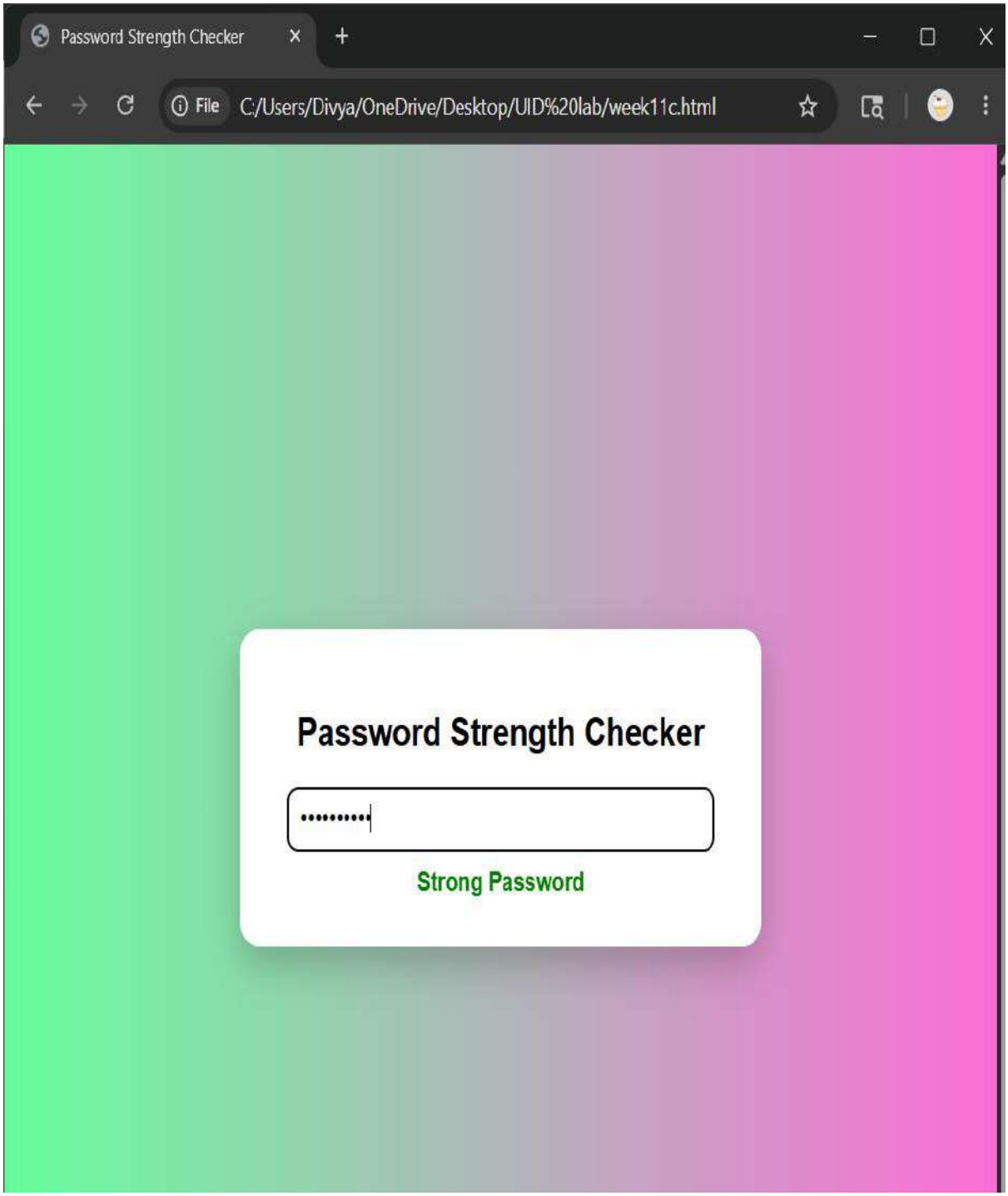
Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Password Strength Checker</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background: linear-gradient(to right, #ff9966, #ff5e62);
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }
  .container {
    background: white;
    padding: 30px;
    border-radius: 15px;
    width: 350px;
    text-align: center;
    box-shadow: 0 10px 25px rgba(0,0,0,0.2);
  }
  h2 {
    margin-bottom: 20px;
  }
```

```
    input {
        width: 90%;
        padding: 10px;
        font-size: 16px;
        border-radius: 8px;
        border: 1px solid #ccc;
    }
    .strength {
        margin-top: 10px;
        font-weight: bold;
    }
    .weak {
        color: red;
    }
    .medium {
        color: orange;
    }
    .strong {
        color: green;
    }
</style>
</head>
<body>
<div class="container">
    <h2>Password Strength Checker</h2>
    <input type="password" id="password" placeholder="Enter Password"
onkeyup="checkStrength()">


    <div id="result" class="strength"></div>
```

```
</div>
<script>
function checkStrength() {
    let password = document.getElementById("password").value;
    let result = document.getElementById("result");
    let strength = 0;
    if (password.length >= 6) strength++;
    if (/[A-Z]/.test(password)) strength++;
    if (/[0-9]/.test(password)) strength++;
    if (/[@$!%*?&]/.test(password)) strength++;
    if (password.length === 0) {
        result.innerHTML = "";
    }
    else if (strength <= 1) {
        result.innerHTML = "Weak Password";
        result.className = "strength weak";
    }
    else if (strength === 2 || strength === 3) {
        result.innerHTML = "Medium Password";
        result.className = "strength medium";
    }
    else {
        result.innerHTML = "Strong Password";
        result.className = "strength strong";
    }
}
</script>
</body>
</html>
```

Output:



Explanation of Tags

1. **type="password"**
2. <input type="password">
    - Used for password input.
    - Characters entered by the user are hidden for security.

3. **onkeyup Event**
4. onkeyup="checkStrength()"
    - Executes the JavaScript function whenever the user releases a key.
    - Enables real-time password strength checking.
5. **Strength Output Division**
6.
    - Displays the password strength message.
    - Updated dynamically using JavaScript.

Additional CSS Concepts Used

1. **Strength Class**
2. .strength
    - Provides common styling for the password strength message.
3. **Weak Password Class**
4. .weak {
5. color: red;
6. }
    - Displays weak passwords in red color.
7. **Medium Password Class**
8. .medium {
9. color: orange;
10. }
    - Displays medium-strength passwords in orange color.
11. **Strong Password Class**
12. .strong {
13. color: green;
14. }
    - Displays strong passwords in green color.

JavaScript Concepts Used

1. **Password Length Check**
2. password.length >= 6
    - Checks whether the password contains at least 6 characters.
3. **Regular Expression (Regex)**
4. /[A-Z]/
    - Checks for uppercase letters.
5. **Digit Validation**
6. /[0-9]/
    - Checks whether the password contains numbers.
7. **Special Character Validation**
8. /[@$!%*?&]/
    - Checks for special characters.
9. **test() Method**
10. .test(password)
    - Determines whether the password matches the specified pattern.
11. **Strength Counter**
12. strength++
    - Increments the strength score whenever a condition is satisfied.
13. **className Property**
14. result.className = "strength strong";
    - Dynamically changes the CSS class according to password strength.

Password Strength Rules

- Length of at least 6 characters.
- Contains an uppercase letter.
- Contains a numeric digit.
- Contains a special character.

Program Logic

1. User enters a password.
2. The onkeyup event calls the checkStrength() function.
3. The program checks:
    - Password length
    - Uppercase letters
    - Numbers
    - Special characters
4. A strength score is calculated.
5. Based on the score:
    - Weak Password → Red color
    - Medium Password → Orange color
    - Strong Password → Green color
6. The result is displayed instantly while typing.

Example:

Input:
abc123

Output:
Medium Password

Input:
Divya@123

Output:
Strong Password

Aim11d: A form should validate a mobile number when the user leaves the input field.

The number must be exactly 10 digits.

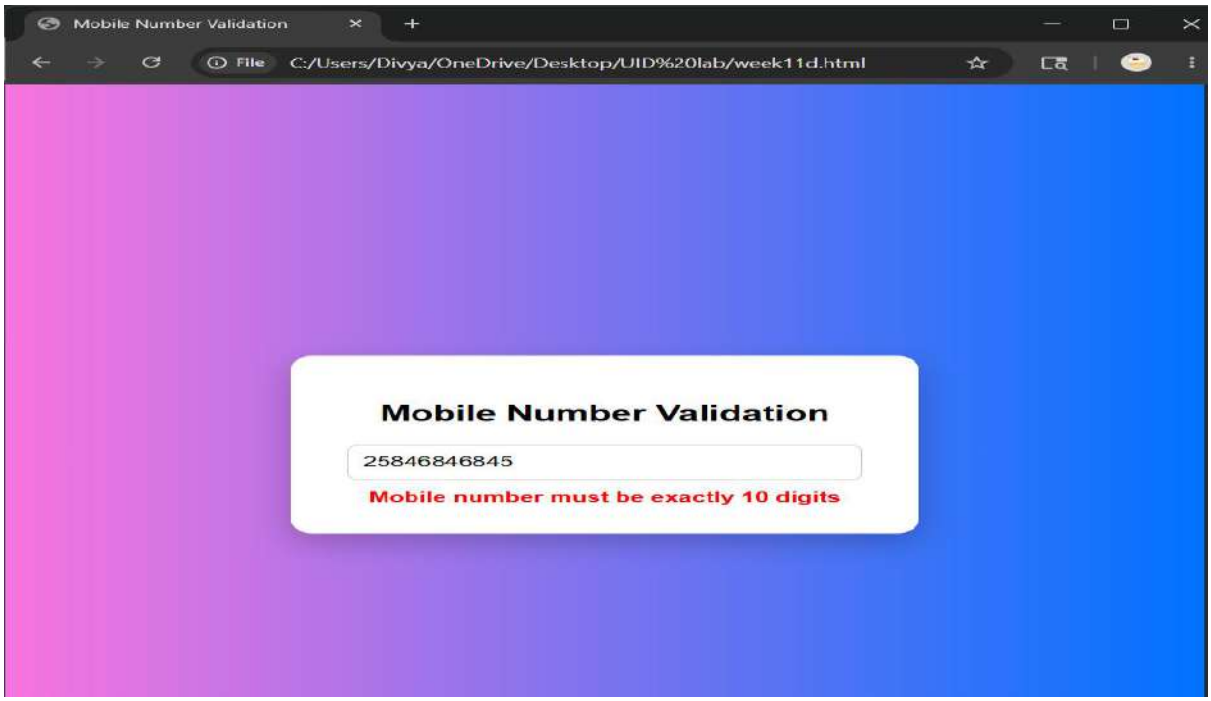
Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Mobile Number Validation</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background: linear-gradient(to right, #00c6ff, #0072ff);
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }
  .container {
    background: white;
    padding: 30px;
    border-radius: 15px;
    width: 350px;
    text-align: center;
    box-shadow: 0 10px 25px rgba(0,0,0,0.2);
  }

  h2 {
    margin-bottom: 20px;
  }
```

```
    input {
        width: 90%;
        padding: 10px;
        font-size: 16px;
        border-radius: 8px;
        border: 1px solid #ccc;
    }

    .message {
        margin-top: 10px;
        font-weight: bold;
    }

    .error {
        color: red;
    }

    .success {
        color: green;
    }
</style>
</head>

<body>

<div class="container">
    <h2>Mobile Number Validation</h2>
```

```
    <input type="text" id="mobile" placeholder="Enter Mobile Number"
onblur="validateMobile()">


    <div id="message" class="message"></div>
</div>


<script>
function validateMobile() {
    let mobile = document.getElementById("mobile").value;
    let message = document.getElementById("message");


    // Check if exactly 10 digits
    let pattern = /^[0-9]{10}$/;


    if (mobile.length === 0) {
        message.innerHTML = "";
    }
    else if (!pattern.test(mobile)) {
        message.innerHTML = "Mobile number must be exactly 10 digits";
        message.className = "message error";
    }
    else {
        message.innerHTML = "Valid Mobile Number";
        message.className = "message success";
    }
}
</script>
</body>
</html>
```

Output:



Explanation of Tags :

1. **onblur Event**
2. onblur="validateMobile()"
    - Executes the JavaScript function when the input field loses focus.
    - Validation occurs after the user clicks outside the input box.
3. **Input Field**
4. <input type="text" id="mobile">
    - Accepts the mobile number from the user.
    - The id attribute helps JavaScript access the entered value.
5. **Message Division**
6.
    - Displays validation messages.
    - Shows whether the mobile number is valid or invalid.

Additional CSS Concepts Used

1. **Message Class**

2. .message
    - Provides common formatting for validation messages.
3. **Error Class**
4. .error {
5. color: red;
6. }
    - Displays invalid mobile number messages in red.
7. **Success Class**
8. .success {
9. color: green;
10. }
    - Displays valid mobile number messages in green.

JavaScript Concepts Used

1. **Regular Expression (Regex)**
2. /^[0-9]{10}$/
    - Validates the mobile number format.
    - Ensures that exactly 10 digits are entered.
3. **Square Brackets []**
4. [0-9]
    - Specifies any digit from 0 to 9.
5. **Curly Braces {10}**
6. {10}
    - Indicates that exactly 10 digits must be present.
7. **test() Method**
8. pattern.test(mobile)
    - Checks whether the entered mobile number matches the pattern.
9. **Logical NOT Operator (!)**
10. !pattern.test(mobile)

    - Executes the validation error block when the pattern does not match.

11. **className Property**
12. message.className = "message error";
    - Dynamically changes the CSS class of the message element.

Validation Rules

- Mobile number must contain exactly 10 digits.
- Only numeric values are allowed.
- Less than 10 digits → Invalid.
- More than 10 digits → Invalid.

Program Logic

1. User enters a mobile number.
2. When the user clicks outside the input field, the onblur event is triggered.
3. The validateMobile() function is executed.
4. A regular expression checks whether the number contains exactly 10 digits.
5. If invalid:
    - Error message is displayed in red.
6. If valid:
    - Success message is displayed in green.

Example:

Input:
987654321

Output:
Mobile number must be exactly 10 digits

Input:
9876543210

Output:
Valid Mobile Number

Aim11e: Design an HTML page, such that the registration form must validate:

Name (not empty)

Email (valid format)

Password (minimum 6 characters)

Program:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Registration Form Validation</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background: linear-gradient(to right, #667eea, #764ba2);
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }
  .container {
    background: white;
    padding: 30px;
    border-radius: 15px;
    width: 350px;
    box-shadow: 0 10px 25px rgba(0,0,0,0.2);
    text-align: center;
  }

  h2 {
```

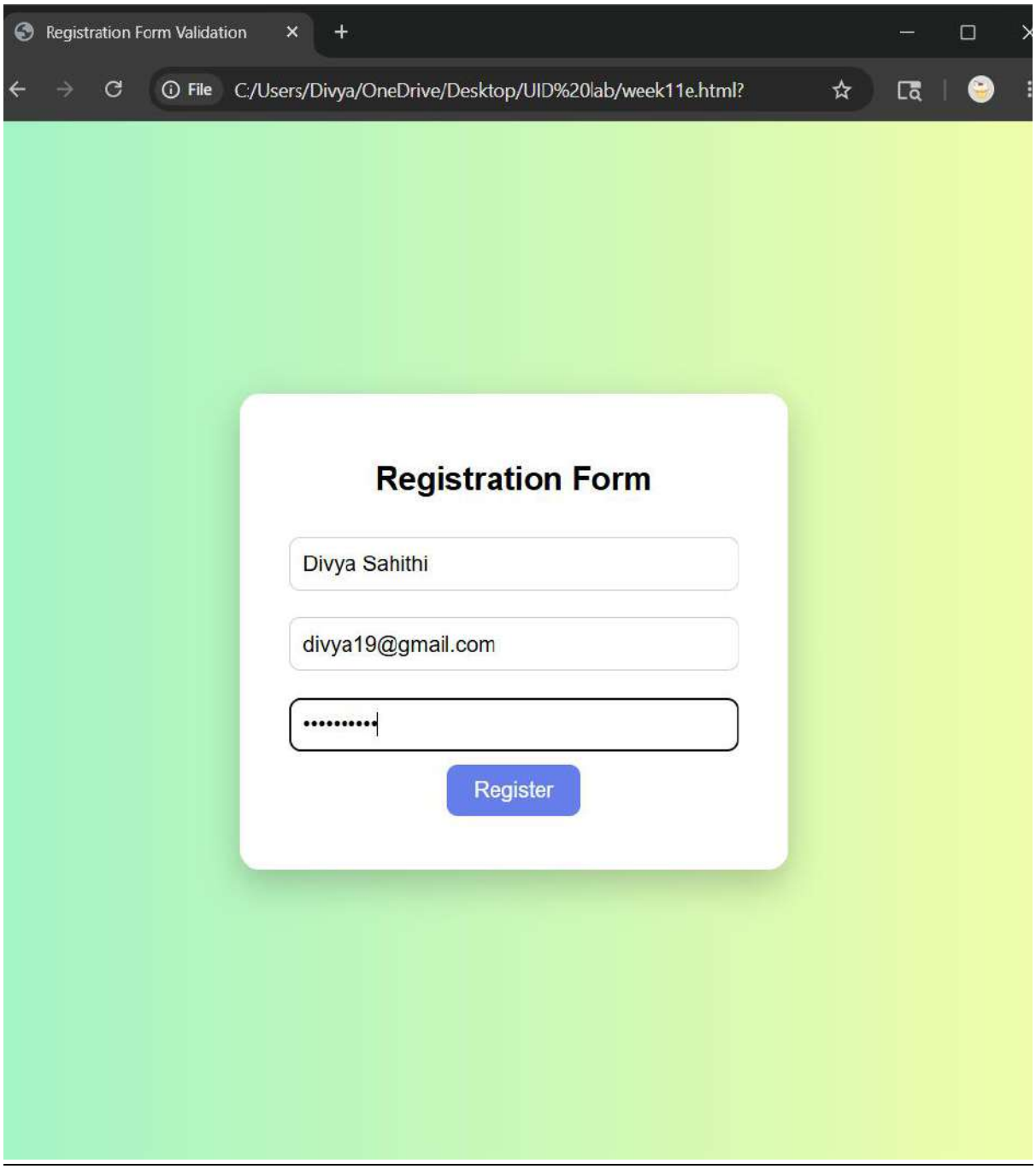
```
    margin-bottom: 20px;
}

input {
    width: 90%;
    padding: 10px;
    margin: 10px 0;
    border-radius: 8px;
    border: 1px solid #ccc;
    font-size: 16px;
}

button {
    padding: 10px 20px;
    border: none;
    border-radius: 8px;
    background-color: #667eea;
    color: white;
    font-size: 16px;
    cursor: pointer;
}

button:hover {
    background-color: #5a67d8;
}

.error {
    color: red;
    font-size: 14px;
```

```
        text-align: left;
        margin-left: 15px;
    }

    .success {
        color: green;
        font-weight: bold;
        margin-top: 10px;
    }
</style>
</head>

<body>

<div class="container">
    <h2>Registration Form</h2>

    <form onsubmit="return validateForm()">
        <input type="text" id="name" placeholder="Enter Name"><br>
        <div id="nameError" class="error"></div>

        <input type="text" id="email" placeholder="Enter Email"><br>
        <div id="emailError" class="error"></div>

        <input type="password" id="password" placeholder="Enter Password"><br>
        <div id="passError" class="error"></div>

        <button type="submit">Register</button>
```

```
        <div id="successMsg" class="success"></div>
    </form>
</div>

<script>
function validateForm() {

    let name = document.getElementById("name").value.trim();
    let email = document.getElementById("email").value.trim();
    let password = document.getElementById("password").value;

    let nameError = document.getElementById("nameError");
    let emailError = document.getElementById("emailError");
    let passError = document.getElementById("passError");
    let successMsg = document.getElementById("successMsg");

    // Clear previous messages
    nameError.innerHTML = "";
    emailError.innerHTML = "";
    passError.innerHTML = "";
    successMsg.innerHTML = "";

    let valid = true;

    // Name validation
    if (name === "") {
        nameError.innerHTML = "Name cannot be empty";
        valid = false;
    }
```

```
    // Email validation
    let emailPattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;
    if (!emailPattern.test(email)) {
        emailError.innerHTML = "Enter a valid email";
        valid = false;
    }

    // Password validation
    if (password.length < 6) {
        passError.innerHTML = "Password must be at least 6 characters";
        valid = false;
    }

    // If all valid
    if (valid) {
        successMsg.innerHTML = "Registration Successful!";
    }

    return valid; // prevents submission if false
}
</script>

</body>

</html>
```

Output:



Explanation of Tags

1. **Multiple Error Divisions**
2.
3.
4.
    - Separate divisions are used to display validation errors for each field.

- Helps users identify which input contains an error.

5. **Success Message Division**
6.
    - Displays a success message when all validations are passed.
7. **type="password"**
8. <input type="password">
    - Hides the password characters while typing.

Additional CSS Concepts Used

1. **Text Alignment**
2. text-align: left;
    - Aligns error messages to the left side.
3. **Margin Property**
4. margin-left: 15px;
    - Adds spacing from the left side.
5. **Success Class**
6. .success {
7. color: green;
8. }
    - Displays successful registration messages in green.

JavaScript Concepts Used

1. **trim() Method**
2. name.trim()
    - Removes extra spaces from the beginning and end of a string.
    - Prevents users from entering only blank spaces.
3. **Boolean Variable**
4. let valid = true;
    - Stores validation status.

- Changes to false when any validation fails.

5. **Email Regular Expression**
6. /^[^]+@[^]+\.[a-z]{2,3}$/
    - Validates the email format.
7. **Password Length Validation**
8. password.length < 6
    - Checks whether the password contains at least 6 characters.
9. **Clearing Previous Messages**
10. nameError.innerHTML = "";
    - Removes old error messages before performing new validation.
11. **return valid**
12. return valid;
    - Returns true if all validations pass.
    - Returns false if any validation fails and prevents form submission.

Validation Rules

1. Name field cannot be empty.
2. Email must follow a valid email format.
3. Password must contain at least 6 characters.

Program Logic

1. User enters Name, Email, and Password.
2. User clicks the Register button.
3. The validateForm() function is executed.
4. Name field is checked for empty values.
5. Email is checked using a regular expression.
6. Password length is validated.
7. If any validation fails:
    - Error messages are displayed below the corresponding fields.

8. If all validations pass:
    - "Registration Successful!" message is displayed.

Example:

Input:
Name: Divya

Email: divya@gmail.com

Password: divya123

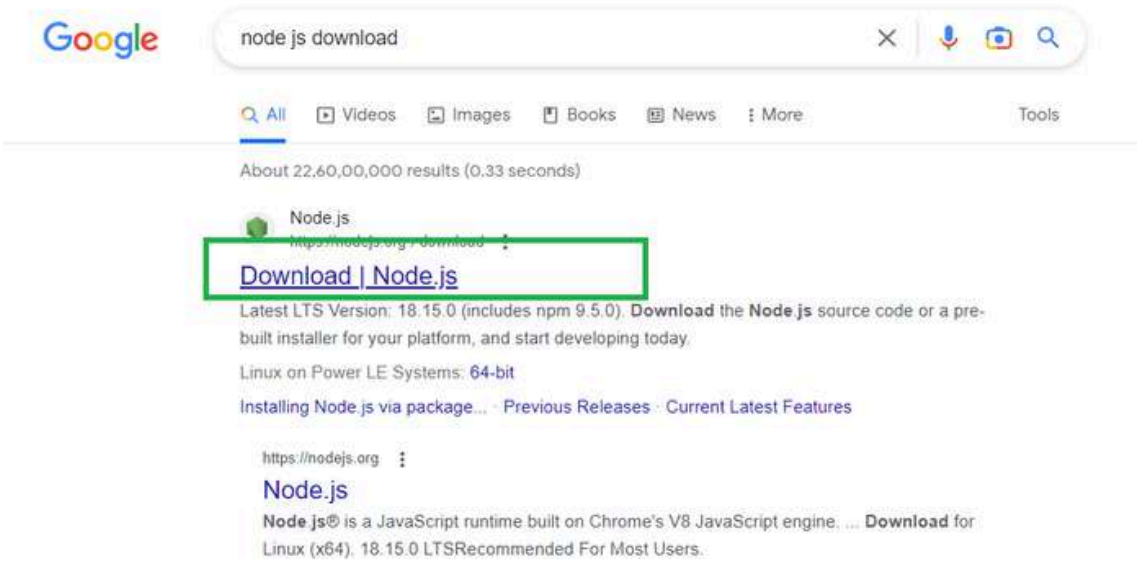
Output:
Registration Successful!

WEEK -12

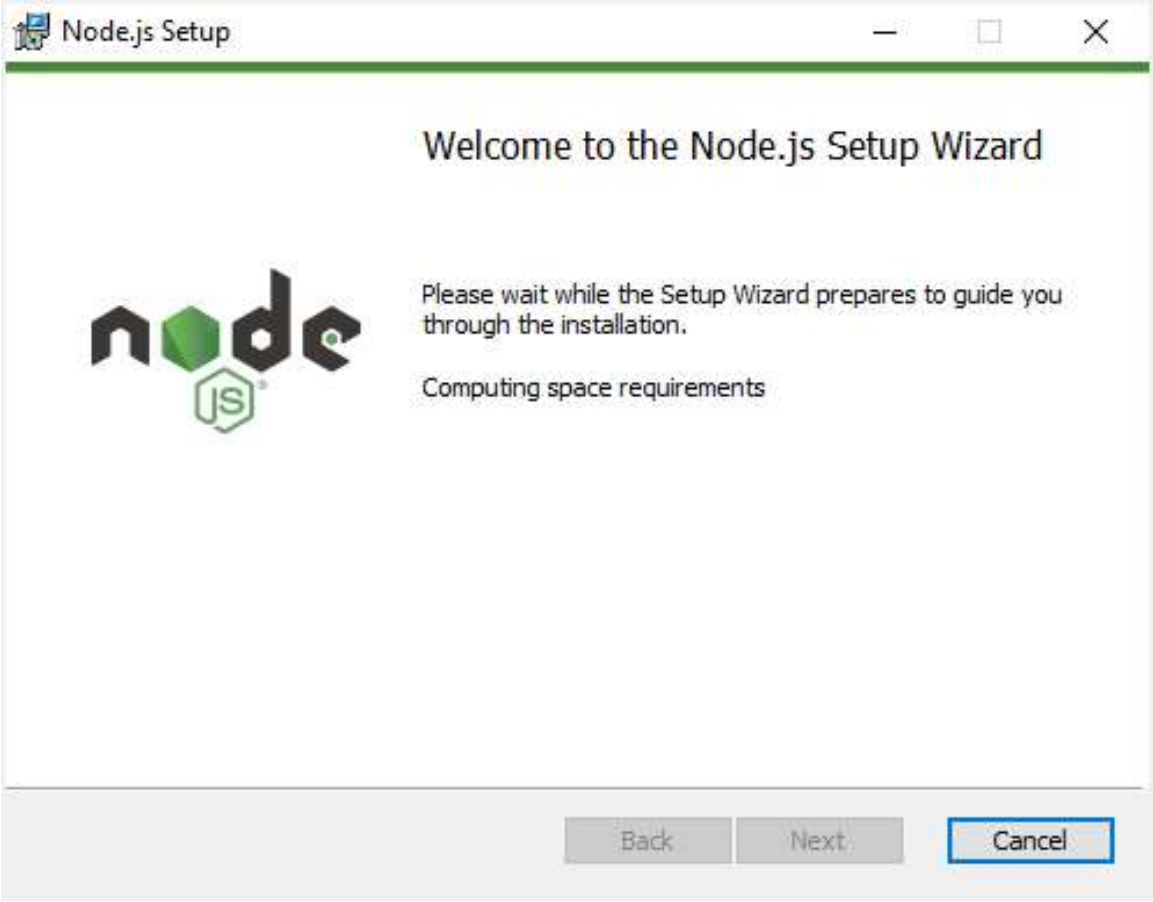
1. AIM: Steps to install NODE Js

PROCEDURE:

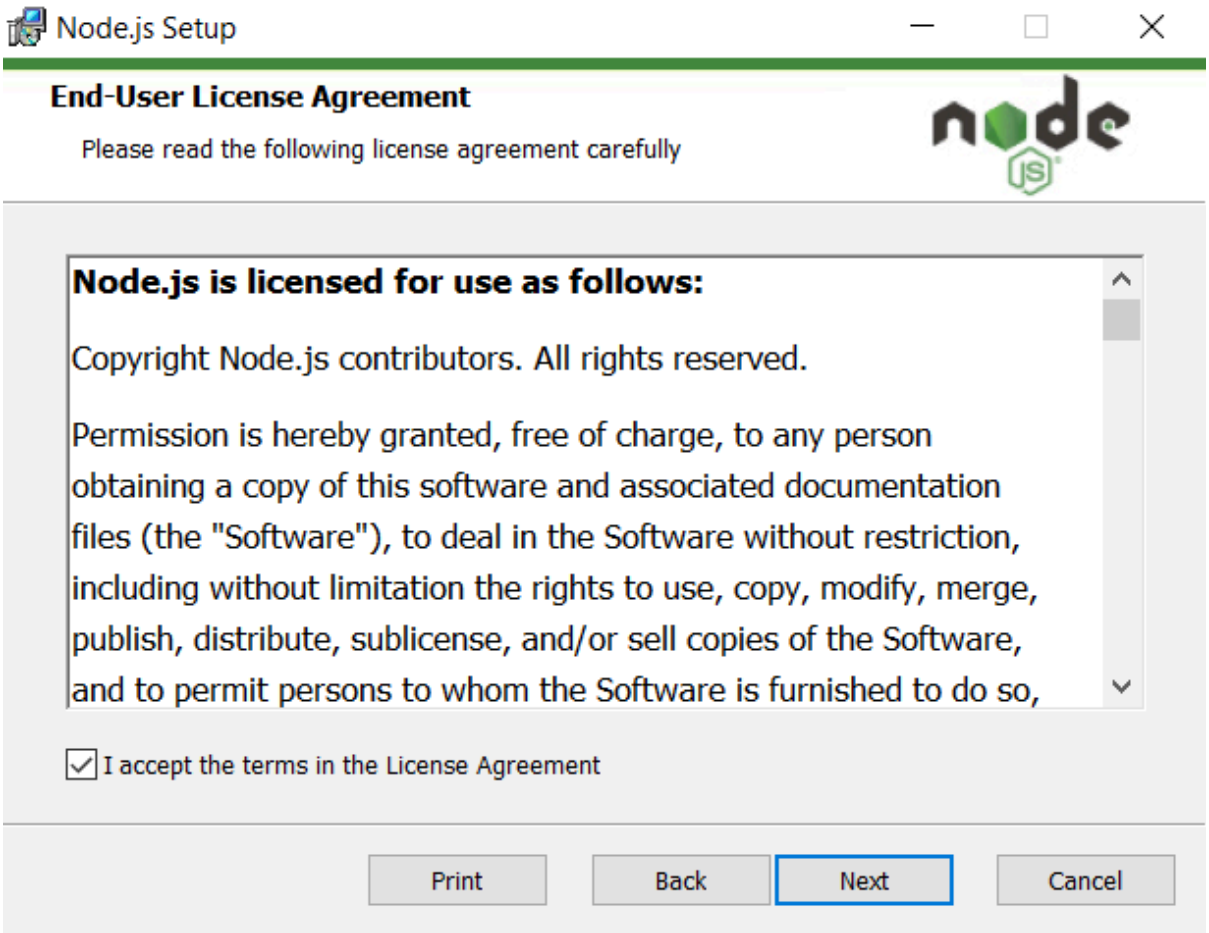
STEP-1



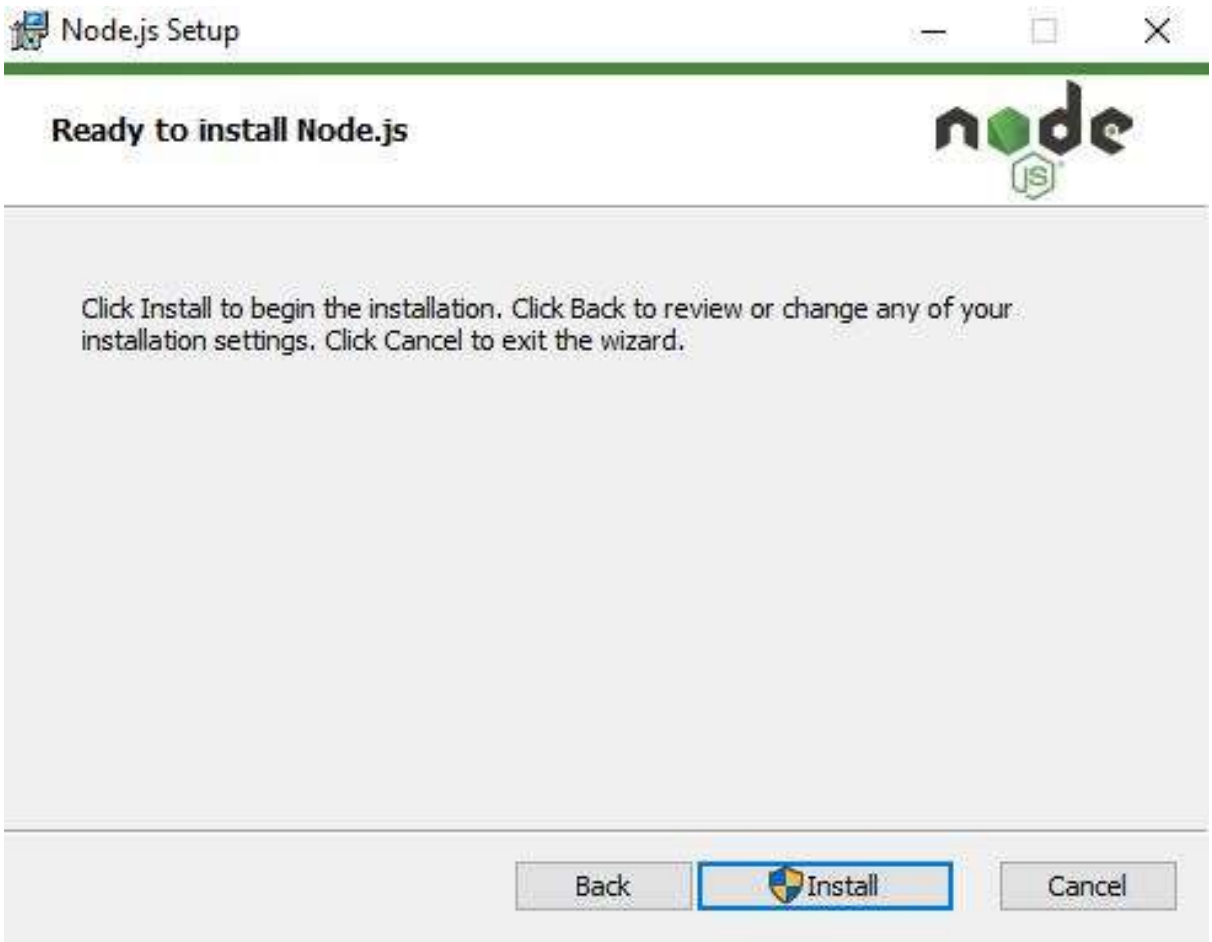
STEP-2



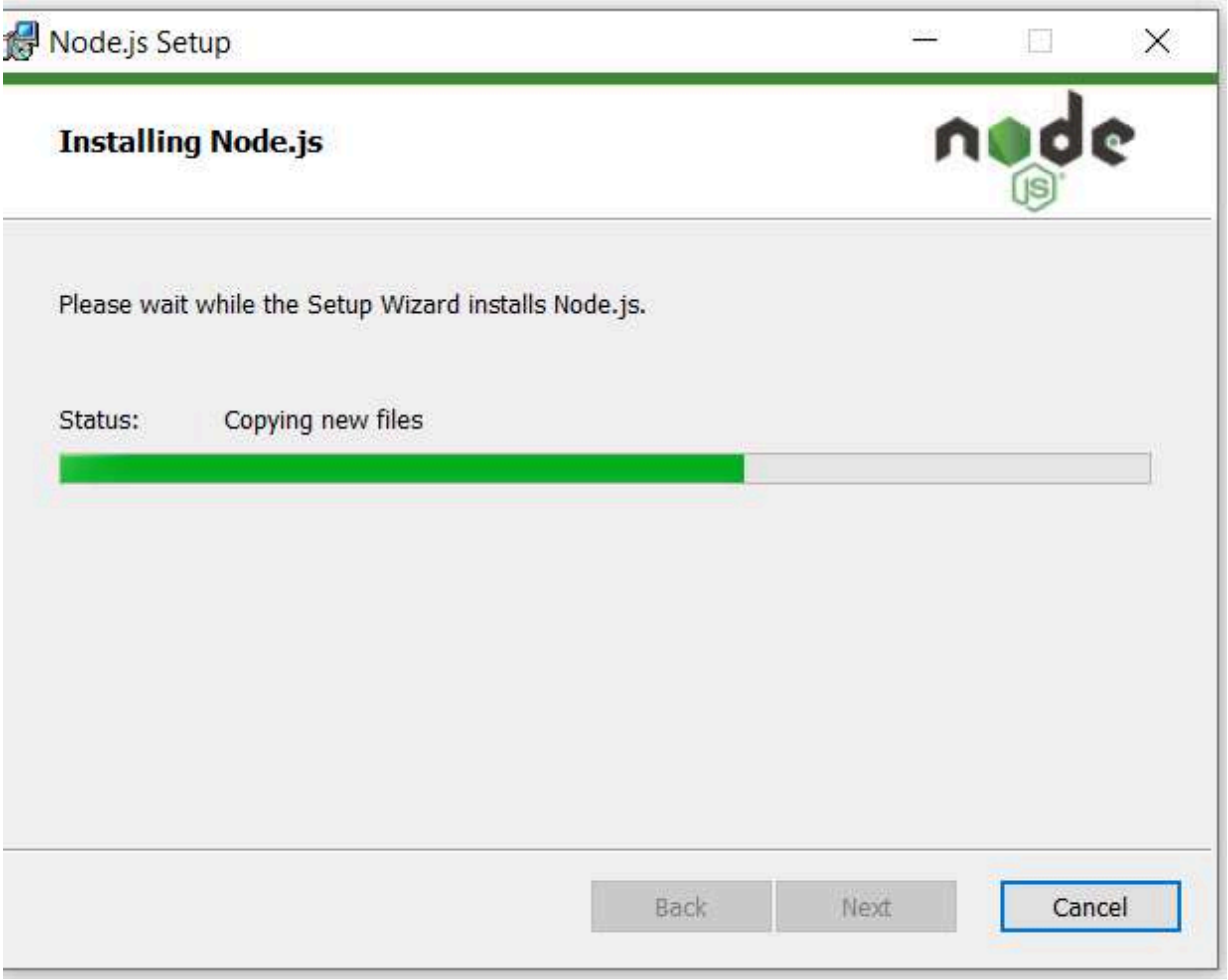
STEP-3


Node.js Setup
End-User License Agreement
Please read the following license agreement carefully
Node.js is licensed for use as follows:
Copyright Node.js contributors. All rights reserved.
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so,
I accept the terms in the License Agreement
Print
Back
Next
Cancel

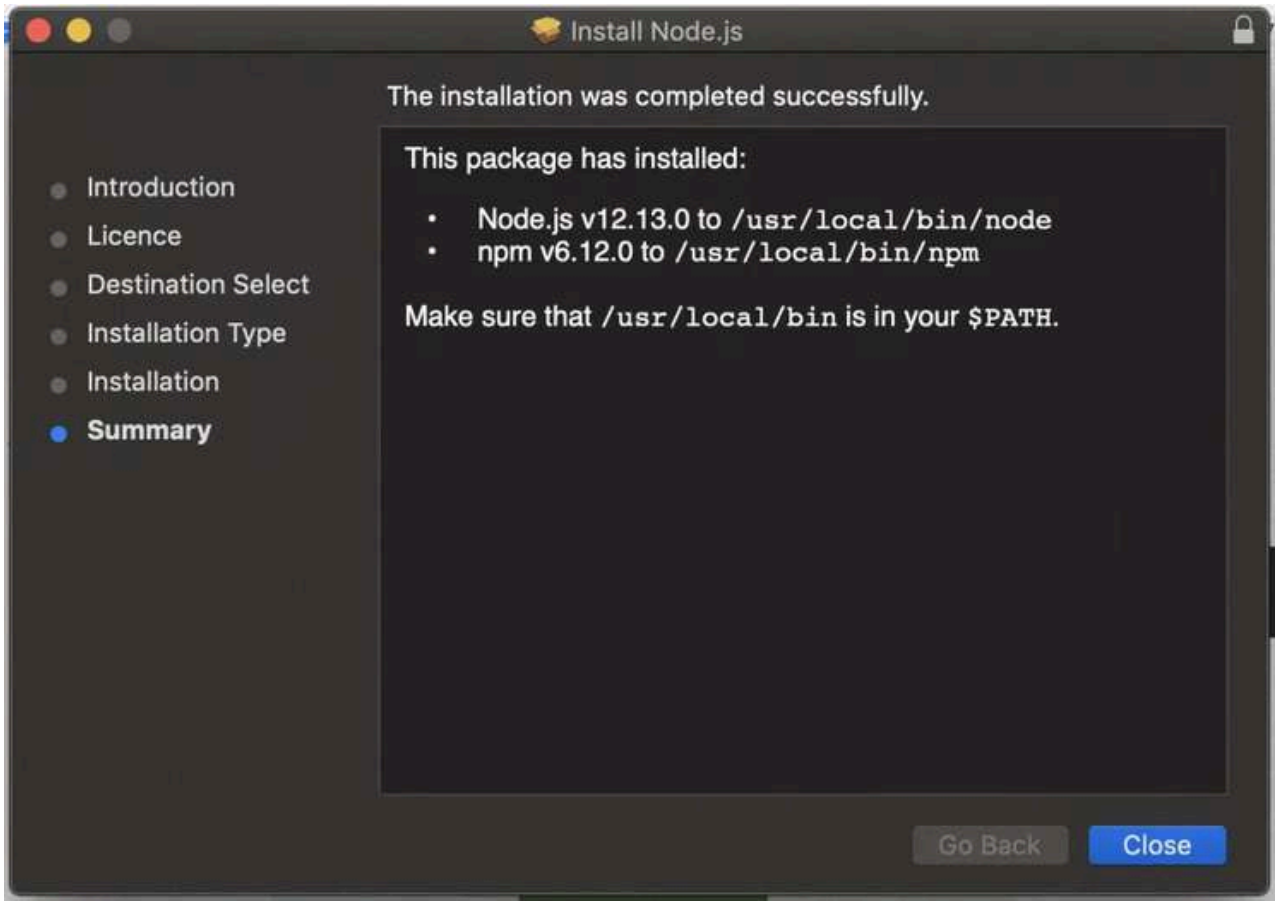
STEP-4


Node.js Setup
Ready to install Node.js
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.
Back
Install
Cancel

STEP-5


Node.js Setup
Installing Node.js
node
Please wait while the Setup Wizard installs Node.js.
Status:
Copying new files
Back
Next
Cancel

STEP-6


Install Node.js
The installation was completed successfully.
Introduction
Licence
Destination Select
Installation Type
Installation
Summary
This package has installed:
Node.js v12.13.0 to /usr/local/bin/node
npm v6.12.0 to /usr/local/bin/npm
Make sure that /usr/local/bin is in your $PATH.
Go Back
Close