USER INTERFACE DESIGN

23CSE113 LAB MANUAL

SUBMITTED BY	
NAME	K. Sree Harshith
ROLL.NO	AV.SC.U4CSE25204
YEAR/SEM/SECTION	1ST YEAR/SEM-2/CSE-C
CREDITS	03

SUBMITTED TO	DR. RAJ KUMAR BATCHU
DESIGNATION	ASST.PROFESSOR (S1. Grade)
DEPARTMENT	SCHOOL OF COMPUTING

INDEX

S. No	TITLE	PAGE NO.	REMARKS
1	WEEK 1: 1. Installation of VS Code 2. To learn basics of html 3. Understanding basic HTML tags	4-10	
2	WEEK 2: a) Design a basic webpage with heading, paragraph, pre, table by adding styles.	11-13	
3	WEEK 3: a) Create class Timetable using HTML elements. b) Create an HTML webpage that should display syllabus of all the subjects as an image format also the timetable.	14-29	
4	WEEK 4: a) Design a registration form using HTML tags. b) Design a webpage with two different forms in one webpage using Iframes.	30-36	
5	WEEK 5: a) Create a card layout using tag. Consider three cards and a button for each card using HTML and CSS. b) Navigation using iframe (mini website). Create 3 basic pages like contact.html, home.html, and about.html. c) Design Amrita Login page by showing different campuses on homepage upon clicking on each campus the details about the campus should be displayed on the same page, use HTML and CSS to design stylish webpage.	37-48	
6	WEEK 6: 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.	49-59	

	Based on the category we select corresponding products should be displayed on the sidebar.		
7	WEEK 7: Create a GitHub repository and push all the lab exercise programs into the repository.	60-62	
8	WEEK 8: a) Create a webpage that displays a square that continuously rotates 360 degrees using CSS animations. b) Create a ball that moves up and down continuously using CSS animation only. c) Create a circle that moves horizontally and rotates at the same time.	63-69	
9	WEEK 9: Create a 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, Armstrong or not. c)Use appropriate designs to display the output.	70-76	
10	WEEK 10: a) Write a JavaScript conditional statement to find the sign of product of three numbers using functions. Display an alert box with the specified sign. b) Write a JavaScript conditional statement to sort three numbers. Display an alert box to show the result. c) Write a Java script conditional statement to find the largest of five numbers using function and onclick event. d) Write a Java script 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.	77-86	
11	WEEK 11: a) A website requires users to enter a username. The system should display a	87-97	

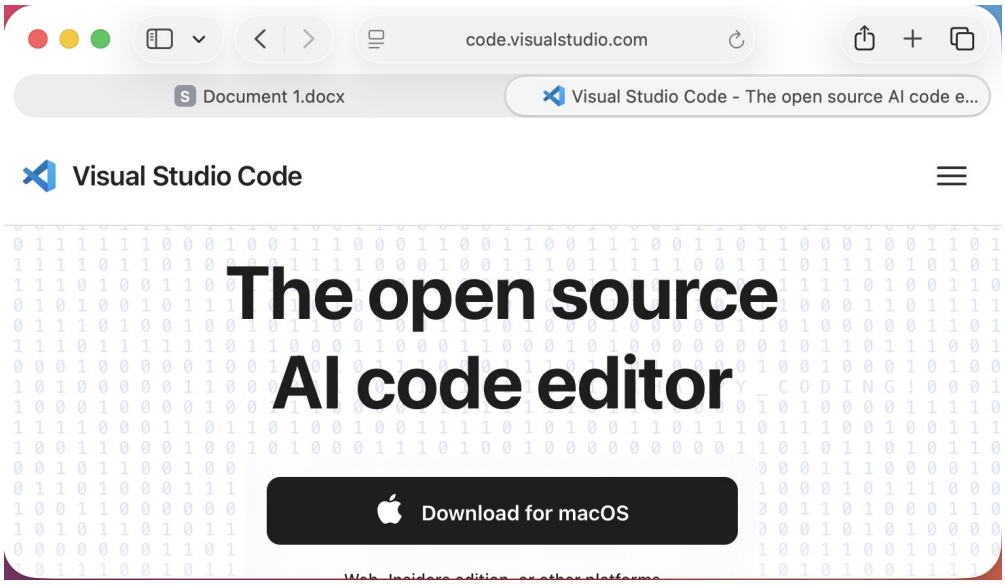
	message while typing if the username is less than 5 characters. b) A registration form should validate email format before submission. If invalid, the form should not be submitted. c) A system should check password strength while typing and display whether it is weak or strong. d) A form should validate a mobile number when the user leaves the input field. The number must be exactly 10 digits. e) 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 of Electron framework.	98-101	

WEEK – 1

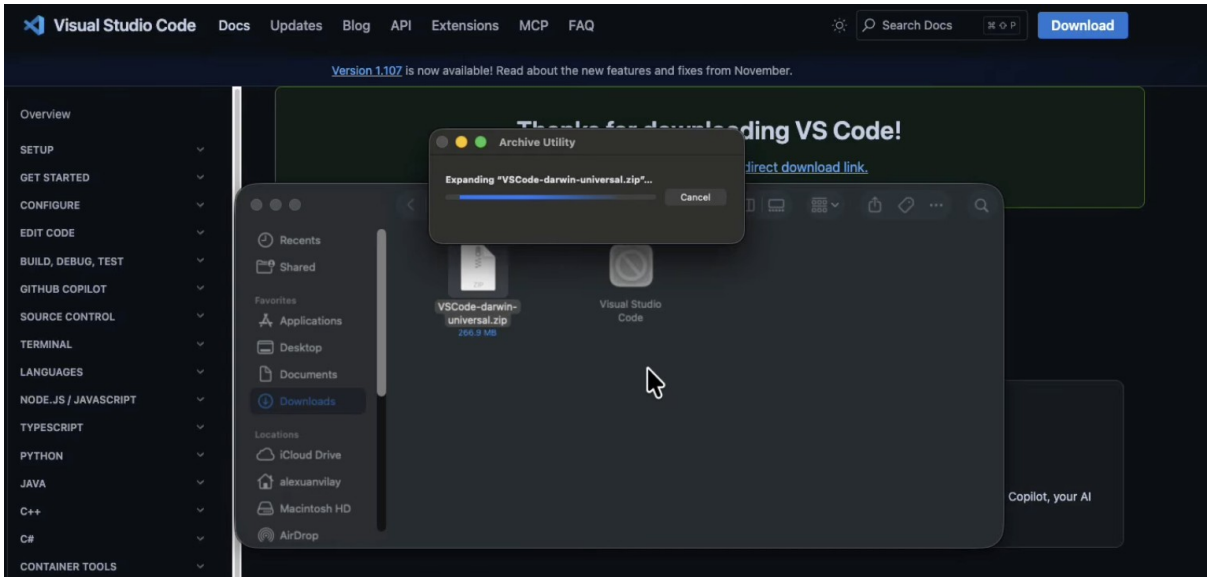
1. Aim: Installation of VS Code

Procedure:

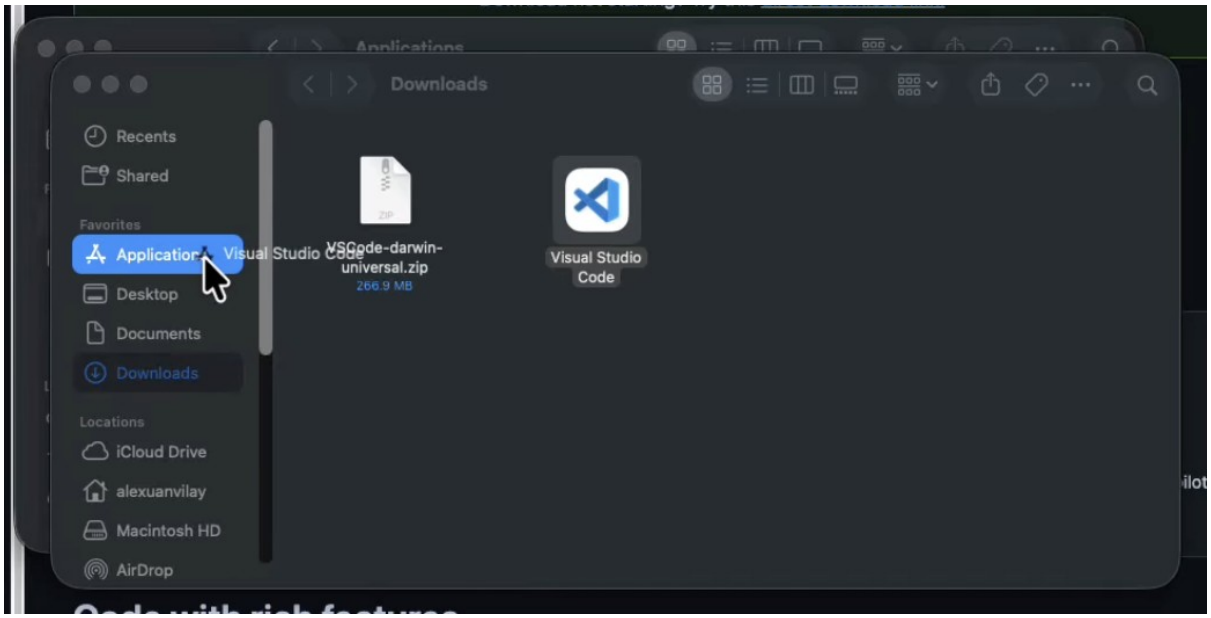
Step 1:



Step 2:



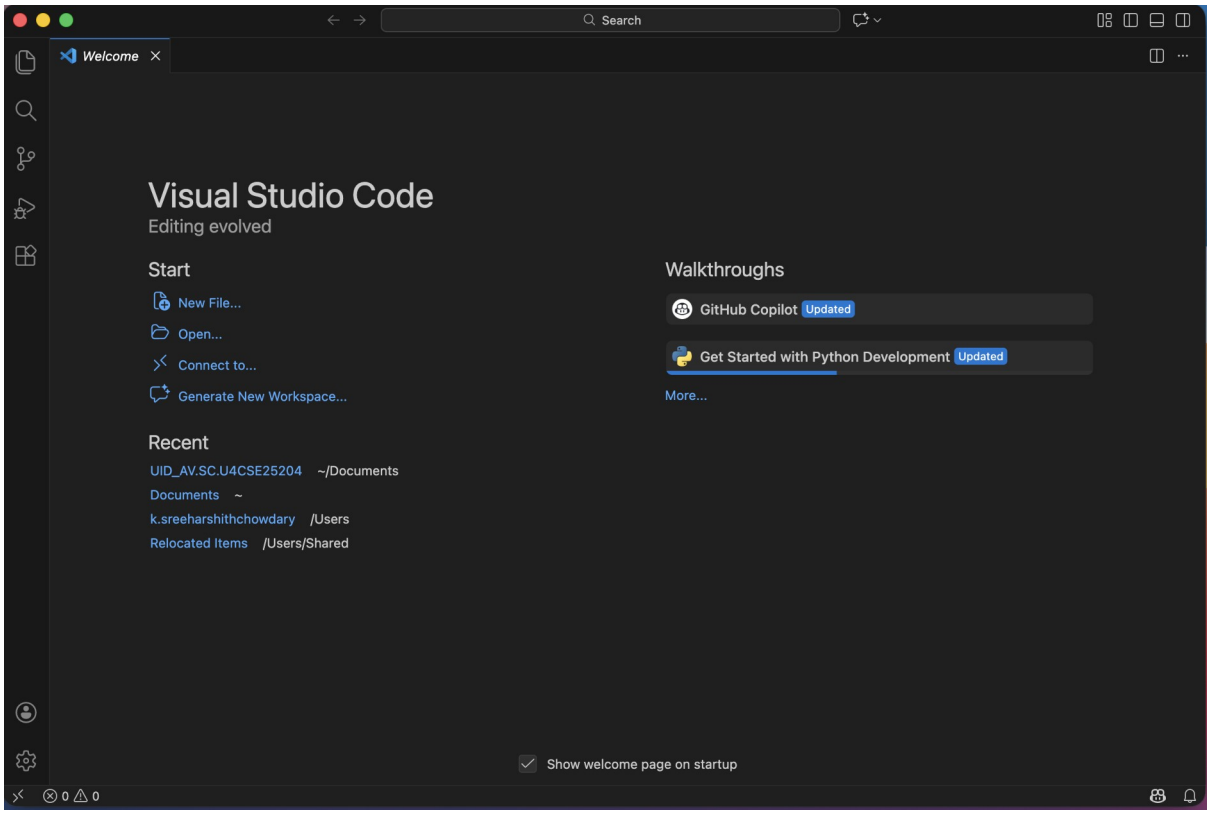
Step 3:


Downloads
Recents
Shared
Favorites
Applications
Desktop
Documents
Downloads
Locations
iCloud Drive
alexuanvilay
Macintosh HD
AirDrop
Visual Studio Code
VSCode-darwin-universal.zip
266.9 MB
Visual Studio Code

Step 4:

Finder
File
Edit
View
Go
Window
Help
Fri 30 Jan 10:36 PM
Applications
Productivity & Finance
Utilities
Social
Entertainment
Creativity
Other
Microsoft 365 C...
Music
Notes
Numbers
OneDrive
Pages
Passwords
Phone
Photo Booth
Photos
Podcasts
Python Launcher
Reminders
Safari
Siri
Stocks
System Settings
The Unarchiver
TV
Visual Studio Co...
Voice Memos
Weather
WhatsApp
Other
Activity Monitor
AirPort Utility
Audio MIDI Setup
Automator
Bluetooth File E...
ColorSync Utility
Console
PDF Documents
Screenshots
OneDrive - Amrit_tham
untitled folder
3h 15m

Step 5:


Search
Welcome
Visual Studio Code
Editing evolved
Start
New File...
Open...
Connect to...
Generate New Workspace...
Recent
UID_AV.SC.U4CSE25204 ~/Documents
Documents ~
k.sreeharshithchowdary /Users
Relocated Items /Users/Shared
Walkthroughs
GitHub Copilot Updated
Get Started with Python Development Updated
More...
Show welcome page on startup

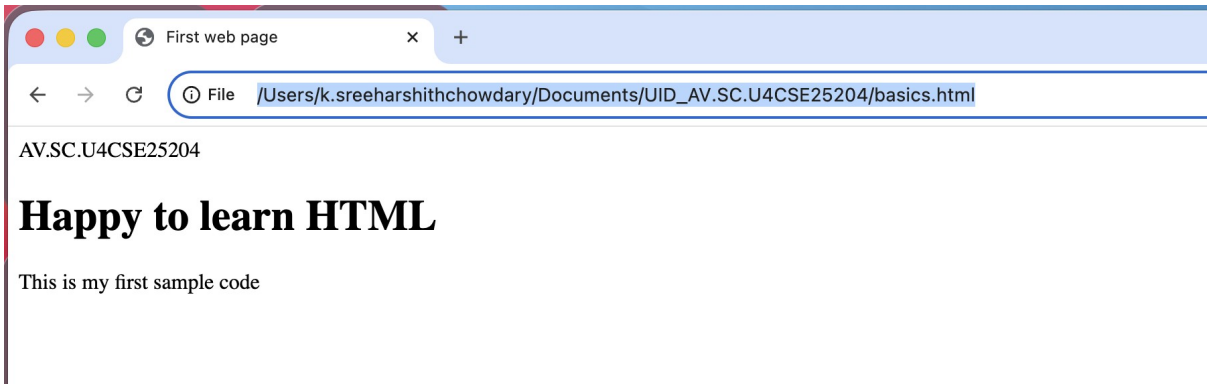
2. Aim: To learn the basics of HTML

Program:

AV.SC.U4CSE25204

```
<!DOCTYPE html>
<html>
<head>
<title>First web page</title>
</head>
<body>
<h1>Happy to learn HTML</h1>
<p>This is my first sample code</p>
</body>
</html>
```

Output:



Tags Used:

- 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

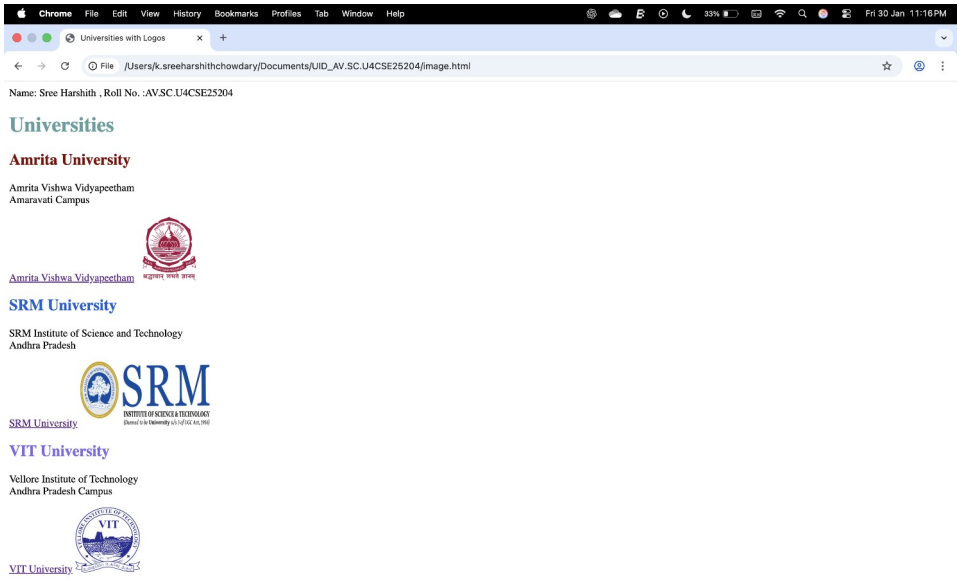
Program:

Name: Sree Harshith , Roll No. :AV.SC.U4CSE25204

```
<!DOCTYPE html>
<html>
<head>
<title>Universities with Logos</title>
</head>
<body>
<h1 style="color:cadetblue">Universities</h1>
<h2 style="color:maroon">Amrita University</h2>
<p>Amrita Vishwa Vidyapeetham <br> Amaravati Campus</p>
<a href = "https://www.amrita.edu">Amrita Vishwa Vidyapeetham</a>
<img src="amrita_logo.png" alt="Amrita Logo" width="100" height="100">
<h2 style="color:rgb(20, 95, 216)">SRM University</h2>
<p>SRM Institute of Science and Technology<br>Andhra Pradesh</p>
<a href = "https://www.srmist.edu.in">SRM University</a>
<img src="srm_logo.png" alt="SRM Logo" width="200" height="100">
<h2 style="color:mediumslateblue">VIT University</h2>
```

```
<p>Vellore Institute of Technology<br>Andhra Pradesh Campus</p>
<a href = "https://www.vit.ac.in">VIT University</a>
<img src="vit_logo.png" alt="VIT Logo" width="100" height="100">
</body>
</html>
```

Output:



HTML TAGS AND ATTRIBUTES USED:

- <hr> – Inserts a horizontal line to separate content
- <h1> – Defines the largest heading
- <pre> – Displays preformatted text (keeps spaces and line breaks)
- <table> – Creates a table
- <caption> – <Gives a title to the table
- <tr> – Defines a table row
- <th> – Defines a table header cell
- <td> – Defines a table data cell
- <b> – Displays text in bold

WEEK-2

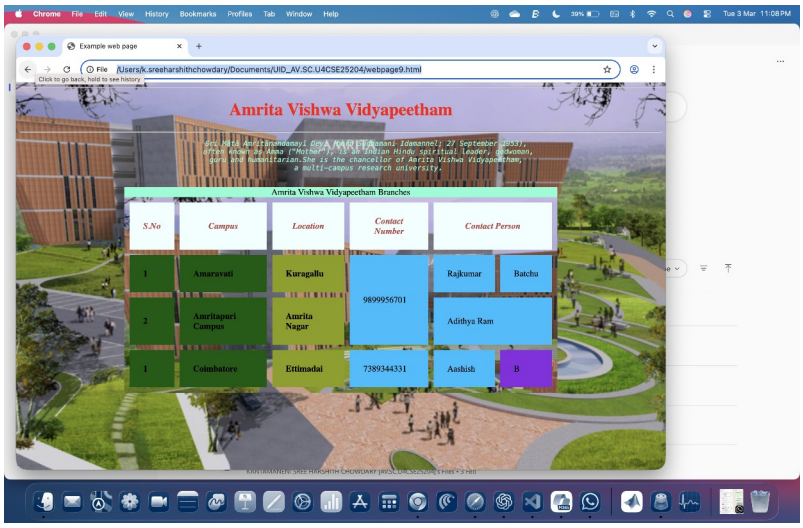
Aim: a) Design a basic webpage with heading, paragraph, pre, table by adding styles.

Program:

```
<html>
<head>
<title>Example web page</title>
</head>
<body background="campus.jpg">
<hr style="color:fuchsia">
<h1 style="color:red;text-align: center;">Amrita Vishwa Vidyapeetham</h1>
<hr style="color:fuchsia">
<pre style="color:aquamarine;font-style: italic;text-align: center;">
Sri Mātā Amritānandamayī Devi (born Sudhamani Idamannel; 27 September 1953),
often known as Amma ("Mother"), is an Indian Hindu spiritual leader, godwoman,
guru and humanitarian.She is the chancellor of Amrita Vishwa Vidyapeetham,
a multi-campus research university.
</pre>
<table width="800" border:"2" cellpadding="25" cellspacing="10" align="center"
background="campus.jpg">
<caption style="background-color: aquamarine;color: black;">Amrita Vishwa Vidyapeetham
Branches</caption>
<tr>
<th style="background-color: azure;color:rgb(186, 70, 70);font-style: italic;">S.No</th>
<th style="background-color: azure;color:rgb(186, 70, 70);font-style: italic;">Campus</th>
<th style="background-color: azure;color:rgb(186, 70, 70);font-style: italic;">Location</th>
<th style="background-color: azure;color:rgb(186, 70, 70);font-style: italic;">Contact
Number</th>
```

```
<th colspan="2" style="background-color: azure;color:rgb(186, 70, 70);font-style:
italic;">Contact Person</th>
</tr>
<tr>
<td style="background-color: rgb(6, 91, 6);"><b>1</b></td>
<td style="background-color: rgb(6, 91, 6);"><b>Amaravati</b></td>
<td style="background-color: rgb(138, 158, 7);"><b>Kuragallu</b></td>
<td rowspan="2" style="color:black;background-color: deepskyblue;">9899956701</td>
<td style="background-color:deepskyblue;">Rajkumar</td>
<td style="background-color: deepskyblue;">Batchu</td>
</tr>
<tr>
<td style="background-color: rgb(6, 91, 6);"><b>2</b></td>
<td style="background-color: rgb(6, 91, 6);"><b>Amritapuri Campus</b></td>
<td style="background-color: rgb(138, 158, 7);"><b>Amrita Nagar</b></td>
<td colspan="2" style="background-color:deepskyblue;">Adithya Ram</td>
</tr>
<tr>
<td style="background-color: rgb(6, 91, 6);"><b>1</b></td>
<td style="background-color: rgb(6, 91, 6);"><b>Coimbatore</b></td>
<td style="background-color: rgb(138, 158, 7);"><b>Ettimadai</b></td>
<td rowspan="2" style="color:black;background-color: deepskyblue;">7389344331</td>
<td style="background-color:deepskyblue;">Aashish</td>
<td style="background-color: blueviolet;">B</td>
</tr>
</table>
</body>
</html>
```

Output:



Explanation of tags:

- <html> – defines the root of the HTML document
- <head> – contains metadata of the webpage
- <title> – sets the title shown in the browser tab
- <body> – contains all visible content
- background (body) – sets background image of the page
- <hr> – creates a horizontal line
- <h1> – defines a main heading
- <pre> – displays text exactly as written (keeps spaces and line breaks)
- <table> – creates a table
- width – sets table width
- border – defines thickness of table border
- cellpadding – space inside cells
- cellspacing – space between cells
- align – aligns table
- background (table) – sets background image for table
- <caption> – gives title to the table
- <tr> – defines a table row
- <th> – defines a header cell
- <td> – defines a data cell
- rowspan – merges cells vertically
- colspan – merges cells horizontally
- color – changes text color
- background-color – sets background color
- text-align – aligns text
- font-style – changes font style

WEEK 3

a) Aim: Create class timetable using HTML elements.

Program:

AV.SC.U4CSE25204

K. SREE HARESHITH

```
<html>
<head>
<title>Timetable</title>
</head>
<table width="1200" border="2" align="center" cellspacing="0">
<caption style="background-color: yellow;">Dr. Raj Timetable</caption>
<tr>
<th rowspan="2">Time/Day</th>
<th>Slot 1</th>
<th>Slot 2</th>
<th>Break</th>
<th>Slot 3</th>
<th>Slot 4</th>
<th>Slot 5</th>
<th>Lunch Break</th>
<th>Slot 6</th>
<th>Slot 7</th>
<th>Slot 8</th>
</tr>
<tr>
<th>8:50-9:40</th>
<th>9:40-10:30</th>
<th>10:30-10:40</th>
<th>10:40-11:30</th>
```

```
<th>11:30-12:20</th>
<th>12:20-1:10</th>
<th>1:10-2:30</th>
<th>2:30-3:20</th>
<th>3:20-4:10</th>
<th>4:10-5:00</th>
</tr>
<tr>
<th>Monday</th>
<td></td>
<td></td>
<td rowspan="5" style="writing-mode: vertical-rl;background-color: yellow;text-align:
center;">Break</td>
<td></td>
<td></td>
<td></td>
<td rowspan="5" style="writing-mode: vertical-rl;background-color: yellow;text-align:
center;">Lunch Break</td>
<td colspan="3" style="background-color: lightskyblue;">21CSE003 PROBLEM SOLVING
AND ALGORITHMIC THINKING LAB</td>
</tr>
<tr>
<th>Tuesday</th>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td colspan="3">23CSE081 Computer Programming Lab</td>
```

```
</tr>
<tr>
<th>Wednesday</th>
<td colspan="2">23CSE004 Computer Programming</td>
<td></td>
<td></td>
<td></td>
<td colspan="3" style="background-color: olive;">19CSE201 Advanced Programming
language Lab(Sec-A,CSE-3)</td>
</tr>
<tr>
<th>Thursday</th>
<td></td>
<td></td>
<td></td>
<td>23CSE008 CHE-C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<th>Friday</th>
<td></td>
<td></td>
<td>23CSE004 Computer Programming</td>
<td></td>
<td></td>
<td colspan="2">23CSE Computer Hardware Essentials Lab(Sec-C,CSE-1)</td>
```

```
<td></td>
</tr>
</table>
<br><br>
<table width="1200" border="2" cellspacing="0" align="center">
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Credits</th>
<th>Faculty Name</th>
<th>Assisting Faculty</th>
</tr>
<tr>
<th>23MAT116</th>
<th>Discrete Mathematics</th>
<th>3-0-2</th>
<th>4</th>
<th>Dr.Srivivasa Rao Thota</th>
<th></th>
</tr>
<tr>
<th>23MAT117</th>
<th>Linear Algebra</th>
<th>3-0-2</th>
<th>4</th>
<th>Mr.Sapavat Bixapathi</th>
<th></th>
</tr>
```

<tr>
<th>23CSE111</th>
<th>Object Oriented Programming</th>
<th>3-0-2</th>
<th>4</th>
<th>Dr.SATYA RAJENDRA SINGH R </th>
<th>Dr. Balaji TK</th>
</tr>
<tr>
<th>23PHY115</th>
<th>Modern Physics</th>
<th>2-1-0</th>
<th>3</th>
<th>Dr.Soumya</th>
<th></th>
</tr>
<tr>
<th>23CSE113</th>
<th>User Interface Design</th>
<th>2-0-2</th>
<th>3</th>
<th>Dr.Rajkumar Batchu </th>
<th>Mr. Barge Bharadwaj</th>
</tr>
<tr>
<th>23MEE115</th>
<th>Manufacturing Practice</th>
<th>0-0-3</th>
<th>1</th>

```
<th>Dr. Sathies S.</th>
<th></th>
</tr>
<tr>
<th>22ADM111</th>
<th>Glimpses of Glorious India </th>
<th>2-0-1</th>
<th>2</th>
<th>Dr. Siva Panuganti</th>
<th></th>
</tr>
</table>
</html>
```

Output:

file:///Users/k.sreeharshithchowdary/Documents/UID_AV.SC.U4CSE25204/timetable

Document 1.docx | Timetable

AV.SC.U4CSE25204 K. SREE HARESHITH

Dr. Raj Timetable

Time/Day	Slot 1	Slot 2	Break	Slot 3	Slot 4	Slot 5	Lunch Break	Slot 6	Slot 7	Slot 8
	8:50-9:40	9:40-10:30	10:30-10:40	10:40-11:30	11:30-12:20	12:20-1:10	1:10-2:30	2:30-3:20	3:20-4:10	4:10-5:00
Monday			Break				Lunch Break	21CSE003 PROBLEM SOLVING AND ALGORITHMIC THINKING LAB		
Tuesday								23CSE081 Computer Programming Lab		
Wednesday	23CSE004 Computer Programming							19CSE201 Advanced Programming language Lab(Sec-A,CSE-3)		
Thursday					23CSE008 CHE-C					
Friday				23CSE004 Computer Programming				23CSE Computer Hardware Essentials Lab(Sec-C,CSE-1)		

Course Code	Course Name	L-T-P	Credits	Faculty Name	Assisting Faculty
23MAT116	Discrete Mathematics	3-0-2	4	Dr.Srivivasa Rao Thota	
23MAT117	Linear Algebra	3-0-2	4	Mr.Sapavat Bixapathi	
23CSE111	Object Oriented Programming	3-0-2	4	Dr.SATYA RAJENDRA SINGH R	Dr. Balaji TK
23PHY115	Modern Physics	2-1-0	3	Dr.Soumya	
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	

b) Aim: Create an HTML webpage that should display syllabus of all the subjects as an image format also the timetable.

Program:

AV.SC.U4CSE25204

K. SREE HARESHITH

```
<html>
<head>
<title>Timetable</title>
</head>
<table width="1200" border="2" align="center" cellspacing="0">
<caption style="background-color: yellow;">Dr. Raj Timetable</caption>
<tr>
<th rowspan="2">Time/Day</th>
<th>Slot 1</th>
<th>Slot 2</th>
<th>Break</th>
<th>Slot 3</th>
<th>Slot 4</th>
<th>Slot 5</th>
<th>Lunch Break</th>
<th>Slot 6</th>
<th>Slot 7</th>
<th>Slot 8</th>
</tr>
<tr>
<th>8:50-9:40</th>
```

<th>9:40-10:30</th>

<th>10:30-10:40</th>

<th>10:40-11:30</th>

<th>11:30-12:20</th>

<th>12:20-1:10</th>

<th>1:10-2:30</th>

<th>2:30-3:20</th>

<th>3:20-4:10</th>

<th>4:10-5:00</th>

</tr>

<tr>

<th>Monday</th>

<td></td>

<td></td>

<td rowspan="5" style="writing-mode: vertical-rl;background-color: yellow;text-align: center;">Break</td>

<td></td>

<td></td>

<td></td>

<td rowspan="5" style="writing-mode: vertical-rl;background-color: yellow;text-align: center;">Lunch Break</td>

<td colspan="3" style="background-color: lightskyblue;">21CSE003 PROBLEM SOLVING AND ALGORITHMIC THINKING LAB</td>

</tr>

<tr>

<th>Tuesday</th>

<td></td>

<td></td>

<td></td>

```
<td></td>
<td></td>
<td colspan="3">23CSE081 Computer Programming Lab</td>
</tr>
<tr>
<th>Wednesday</th>
<td colspan="2">23CSE004 Computer Programming</td>
<td></td>
<td></td>
<td></td>
<td colspan="3" style="background-color: olive;">19CSE201 Advanced Programming
language Lab(Sec-A,CSE-3)</td>
</tr>
<tr>
<th>Thursday</th>
<td></td>
<td></td>
<td></td>
<td>23CSE008 CHE-C</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<th>Friday</th>
<td></td>
<td></td>
<td>23CSE004 Computer Programming</td>
```

```
<td></td>
<td></td>
<td colspan="2">23CSE Computer Hardware Essentials Lab(Sec-C,CSE-1)</td>
<td></td>
</tr>
</table>
<br><br>
<table width="1200" border="2" cellspacing="0" align="center">
<tr>
<th>Course Code</th>
<th>Course Name</th>
<th>L-T-P</th>
<th>Credits</th>
<th>Faculty Name</th>
<th>Assisting Faculty</th>
</tr>
<tr>
<th>23MAT116</th>
<th>Discrete Mathematics</th>
<th>3-0-2</th>
<th>4</th>
<th>Dr.Srivivasa Rao Thota</th>
<th></th>
</tr>
<tr>
<th>23MAT117</th>
<th>Linear Algebra</th>
<th>3-0-2</th>
<th>4</th>
```

<th>Mr.Sapavat Bixapathi</th>

<th></th>

</tr>

<tr>

<th>23CSE111</th>

<th>Object Oriented Programming</th>

<th>3-0-2</th>

<th>4</th>

<th>Dr.SATYA RAJENDRA SINGH R </th>

<th>Dr. Balaji TK</th>

</tr>

<tr>

<th>23PHY115</th>

<th>Modern Physics</th>

<th>2-1-0</th>

<th>3</th>

<th>Dr.Soumya</th>

<th></th>

</tr>

<tr>

<th>23CSE113</th>

<th>User Interface Design</th>

<th>2-0-2</th>

<th>3</th>

<th>Dr.Rajkumar Batchu </th>

<th>Mr. Barge Bharadwaj</th>

</tr>

<tr>

<th>23MEE115</th>

```
<th>Manufacturing Practice</th>
<th>0-0-3</th>
<th>1</th>
<th>Dr. Sathies S.</th>
<th></th>
</tr>
<tr>
<th>22ADM111</th>
<th>Glimpses of Glorious India </th>
<th>2-0-1</th>
<th>2</th>
<th>Dr. Siva Panuganti</th>
<th></th>
</tr>
</table>
<body>
<h2>Syllabus</h2>
<h3>Discrete Mathematics</h3><br>
<img src="D_M.png" alt="Syllabus Image">
<h3>Linear Algebra</h3><br>
<img src="L_A.png" alt="Syllabus Image">
<h3>Object Oriented Programming</h3><br>
<img src="OOP.png" alt="Syllabus Image">
<h3>Manufacturing Practice</h3><br>
<img src="M_PRAC.png" alt="Syllabus Image">
<h3>User Interface Design</h3><br>
<img src="U_I_D.png" alt="Syllabus Image">
<h3>Modern Physics</h3><br>
<img src="M_PHY.png" alt="Syllabus Image">
```

<h3>Glimpses of Glorious India</h3>

<img src="GGI.png" alt="Syllabus Image">

</body>

</html>

Output:

file:///Users/k.sreeharshithchowdary/Documents/UID_AV.SC.U4CSE25204/timetable

Document 1.docx | Timetable

AV.SC.U4CSE25204 K. SREE HARESHITH

Dr. Raj Timetable

Time/Day	Slot 1	Slot 2	Break	Slot 3	Slot 4	Slot 5	Lunch Break	Slot 6	Slot 7	Slot 8
	8:50-9:40	9:40-10:30	10:30-10:40	10:40-11:30	11:30-12:20	12:20-1:10	1:10-2:30	2:30-3:20	3:20-4:10	4:10-5:00
Monday			Break				Lunch Break	21CSE003 PROBLEM SOLVING AND ALGORITHMIC THINKING LAB		
Tuesday								23CSE081 Computer Programming Lab		
Wednesday	23CSE004 Computer Programming							19CSE201 Advanced Programming language Lab(Sec-A,CSE-3)		
Thursday					23CSE008 CHE-C					
Friday				23CSE004 Computer Programming				23CSE Computer Hardware Essentials Lab(Sec-C,CSE-1)		

Course Code	Course Name	L-T-P	Credits	Faculty Name	Assisting Faculty
23MAT116	Discrete Mathematics	3-0-2	4	Dr.Srivivasa Rao Thota	
23MAT117	Linear Algebra	3-0-2	4	Mr.Sapavat Bixapathi	
23CSE111	Object Oriented Programming	3-0-2	4	Dr.SATYA RAJENDRA SINGH R	Dr. Balaji TK
23PHY115	Modern Physics	2-1-0	3	Dr.Soumya	
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	

Syllabus

Discrete Mathematics

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

file:///Users/k.sreeharshithchowdary/Documents/UID_AV.SC.U4CSE25204/timetable

Document 1.docx | Timetable

22ADM111	Glimpses of Glorious India	2-0-1	2	Dr. Siva Panuganti	

Syllabus

Discrete Mathematics

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 traveling salesman problem. Closest neighbour algorithm.

Linear Algebra

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

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.

Object Oriented Programming

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.

Manufacturing Practice

Syllabus

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

file:///Users/k.sreeharshithchowdary/Documents/UID_AV.SC.U4CSE25204/timetable
Document 1.docx
Timetable

Manufacturing Practice

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.

User Interface Design

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

file:///Users/k.sreeharshithchowdary/Documents/UID_AV.SC.U4CSE25204/timetable

Document 1.docx | Timetable

Modern Physics

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

Glimpses of Glorious India

Syllabus

Face the Brutes, Role of Women in India, Acharya Chanakya, God and Iswara, Bhagavad Gita: From Soldier to Samsarin to Sadhaka, Lessons of Yoga from Bhagavad Gita, Indian Soft powers, Preserving Nature through Faith, Ancient Indian Cultures (Class Activity), Practical Vedanta, To the World from India, Indian Approach to Science.

HTML Tags and Attributes Used:

- <table> – creates a table
- <caption> – gives a title to a table
- <tr> – defines a table row
- <th> – defines a table header cell
- <td> – defines a table data cell
-
 – inserts a line break
- <h2> – defines a second-level heading
- <img> – displays an image
- width – sets the width of the table
- border – specifies table border thickness
- align – aligns the table to center
- cellspacing – sets space between table cells
- rowspan – merges cells vertically
- colspan – merges cells horizontally
- style – applies inline CSS styles
- src – specifies the image source
- alt – provides alternate text for an image

WEEK 4

Aim: a) Design a registration form using HTML tags.

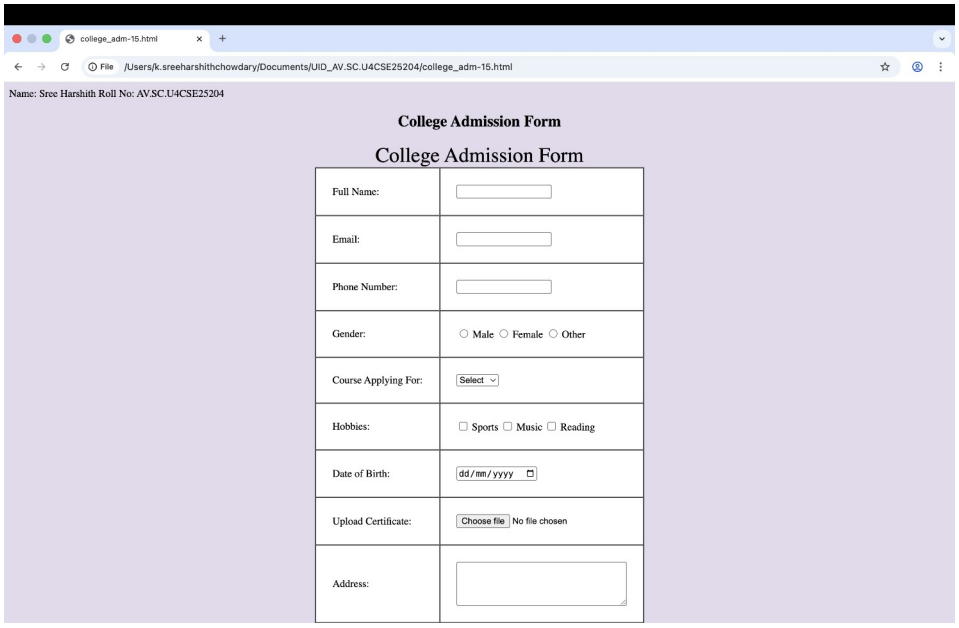
Program:

```
<!DOCTYPE html>
<html>
<body style="background-color: rgb(227, 217, 235);">
<h2 align="center">College Admission Form</h2>
<center>
<form>
<table border="1" cellpadding="25" cellspacing="0" style="background-color: white;">
<caption style="text-align:center; font-size:200%;">
College Admission Form
</caption>
<tr>
<td>Full Name:</td>
<td><input type="text" name="fname"></td>
</tr>
<tr>
<td>Email:</td>
<td><input type="email" name="email"></td>
</tr>
<tr>
<td>Phone Number:</td>
<td><input type="number" name="phone"></td>
</tr>
<tr>
<td>Gender:</td>
<td><input type="radio" name="gender" value="male"> Male
<input type="radio" name="gender" value="female"> Female
```

```
<input type="radio" name="gender" value="other"> Other
</td>
</tr>
<tr>
<td>Course Applying For:</td>
<td><select name="course">
<option value="select">Select</option>
<option value="btech">B.Tech</option>
<option value="mtech">M.Tech</option>
<option value="bba">BBA</option>
<option value="mba">MBA</option>
</select>
</td>
</tr>
<tr>
<td>Hobbies:</td>
<td><input type="checkbox" name="hobbies" value="sports"> Sports
<input type="checkbox" name="hobbies" value="music"> Music
<input type="checkbox" name="hobbies" value="reading"> Reading
</td>
</tr>
<tr>
<td>Date of Birth:</td>
<td><input type="date" name="dob"></td>
</tr>
<tr>
<td>Upload Certificate:</td>
<td><input type="file" name="certificate"></td>
</tr>
```

```
<tr>
<td>Address:</td>
<td><textarea rows="4" cols="30" name="address"></textarea></td>
</tr>
<tr>
<td colspan="2" align="center">
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</td>
</tr>
</table>
</form>
</center>
</body>
</html>
```

Output:



Aim: b) Design a webpage with two different forms in one webpage using Iframes.

Program:

```
<!DOCTYPE html>
<html>
<body style="background-color: lightgoldenrodyellow;">
<h2 align="center">Two Forms Displayed Using Iframes</h2>
<table align="center">
<tr>
<td>
<iframe src="iframe_login.html" width="350" height="350">
</iframe>
</td>
<td>
<iframe src="iframe_registration.html" width="350" height="350">
</iframe>
</td>
</tr>
</table>
</body>
</html>
```

Iframe_login:

```
<!DOCTYPE html>
<html>
<body>
<h3 align="center">User Login</h3>
```

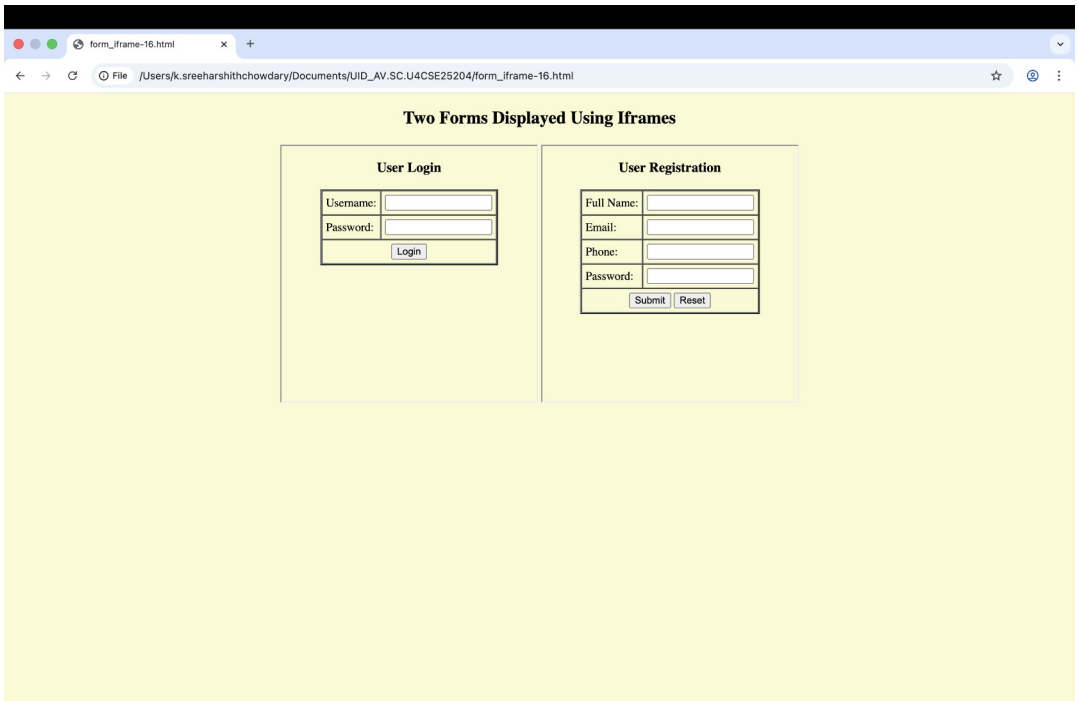
```
<form>
<table border="2" cellpadding="5" cellspacing="0" align="center">
<tr>
<td>Username:</td>
<td><input type="text" name="username"></td>
</tr>
<tr>
<td>Password:</td>
<td><input type="password" name="password"></td>
</tr>
<tr>
<td colspan="2" align="center">
<input type="submit" value="Login">
</td>
</tr>
</table>
</form>
</body>
</html>
```

Iframe_registration:

```
<!DOCTYPE html>
<html>
<body>
<h3 align="center">User Registration</h3>
<form>
<table border="2" cellpadding="5" cellspacing="0" align="center">
<tr>
<td>Full Name:</td>
```

```
<td><input type="text" name="fname"></td>
</tr>
<tr>
<td>Email:</td>
<td><input type="email" name="email"></td>
</tr>
<tr>
<td>Phone:</td>
<td><input type="text" name="phone"></td>
</tr>
<tr>
<td>Password:</td>
<td><input type="password" name="password"></td>
</tr>
<tr>
<td colspan="2" align="center">
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</td>
</tr>
</table>
</form>
</body>
</html>
```

Output:



HTML Tags and Attributes Used:

• <form> – creates a form to collect user input

• <input> – creates input fields

• <select> – creates a dropdown list

• <option> – defines options inside a dropdown

• <textarea> – creates a multi-line text input area

• type – specifies the type of input field

• name – assigns a name to form elements

• value – sets the value of input elements

• text – for Full Name and Phone

• email – for Email input

• password – for Password input

• submit – to submit the form

• reset – to clear the form

WEEK 5

a) Create a card layout using tag. Consider three cards and a button for each card using HTML and CSS.

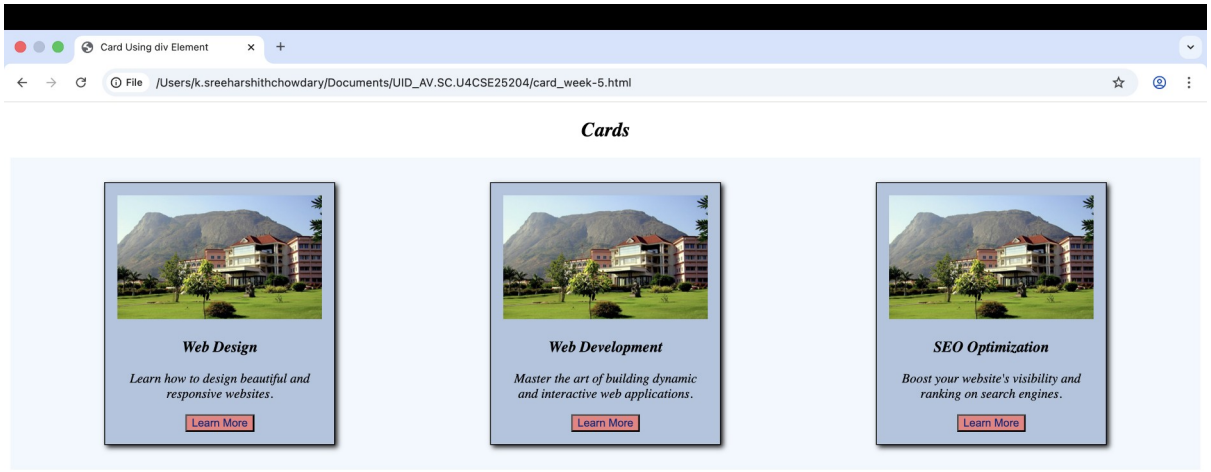
Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Card Using div Element</title>
<style>
body {
background-color: white;
font-style: italic;}
.container{
display: flex;
flex-wrap: wrap;
flex-direction: row;
background-color:aliceblue;
border: 2px 2px 2px solid black;
box-shadow: 5px rgb(29, 27, 27);
justify-content: center;
padding: 20px;
text-align: center;
}
.card {
background-color: lightsteelblue;
border: 1px solid black;
box-shadow: 3px 3px 6px rgb(29, 27, 27);
margin: 10px auto;
```

```
padding: 15px;
width: 250px;
}
.card img{
width: 100%;
height: auto;
}
.card button{
background-color: lightcoral;
color: rgb(12, 39, 126);
text-align: center;
}
.card button:hover{
background-color: rgb(10, 6, 231);
color: rgb(12, 39, 126);
text-align: center;
}
</style>
</head>
<body>
<h2 align="center">Cards</h2>
<div class="container">
<div class="card">
<img src="campus2.jpg" alt="Image">
<h3>Web Design</h3>
<p>Learn how to design beautiful and responsive websites.</p>
<button>Learn More</button>
</div>
<div class="card">
```

```
<img src="campus2.jpg" alt="Image">
<h3>Web Development</h3>
<p>Master the art of building dynamic and interactive web applications.</p>
<button>Learn More</button>
</div>
<div class="card">
<img src="campus2.jpg" alt="Image">
<h3>SEO Optimization</h3>
<p>Boost your website's visibility and ranking on search engines.</p>
<button>Learn More</button>
</div>
</div>
</body>
</html>
```

Output:



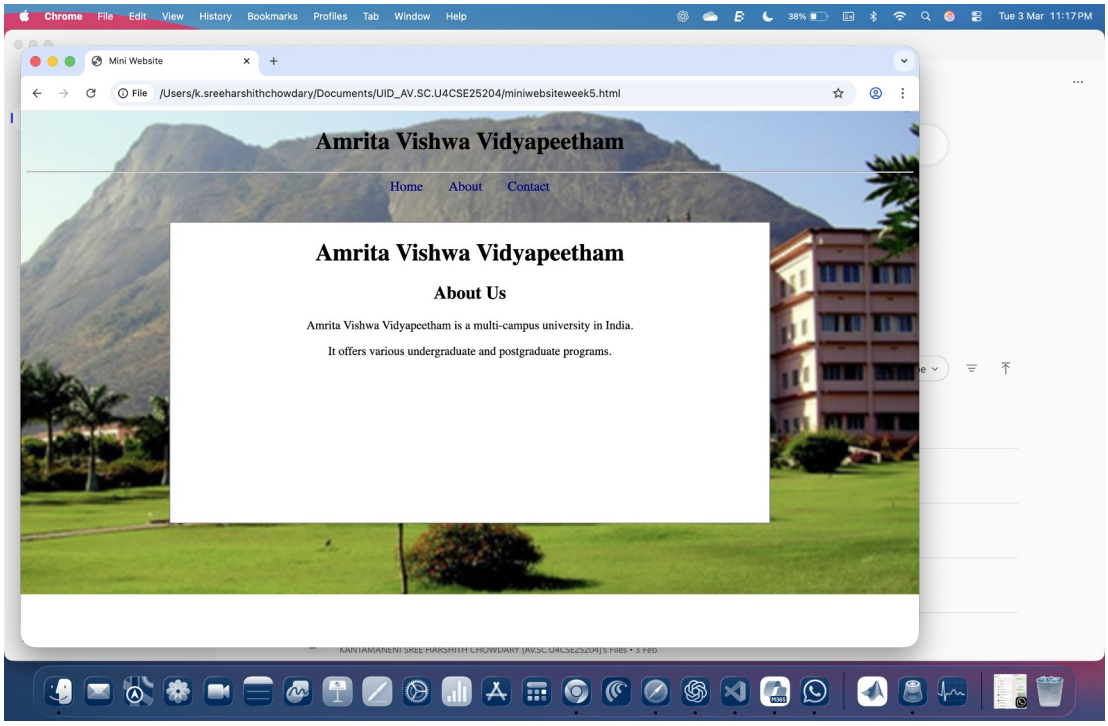
b) Navigation using iframe (mini website). Create 3 basic pages like contact.html, home.html, and about.html.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Mini Website</title>
<style>
body {
background-image: url("campus2.jpg");
background-size: cover;
background-repeat: no-repeat;
background-position: center;
}
iframe {
background-color: white;
border: 1px solid gray;
}
a {
text-decoration: none;
color: darkblue;
font-size: 18px;
margin: 0 15px;
}
a:hover {
text-decoration: underline;
}
</style>
</head>
```

```
<body>
<h1 align="center">Amrita Vishwa Vidyapeetham</h1>
<hr>
<center>
<a href="home.html" target="homepage">Home</a>
<a href="about.html" target="homepage">About</a>
<a href="contact.html" target="homepage">Contact</a>
</center>
<br><br>
<center>
<iframe name="homepage" width="800" height="400">
</iframe>
</center>
</body>
</html>
```

Output:



c) Design Amrita Login page by showing different campuses on homepage upon clicking on each campus the details about the campus should be displayed on the same page, use HTML and CSS to design stylish webpage.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Mini Website</title>
<style>
body {
background-color: bisque;
font-family: Arial;
}
a {
text-decoration: none;
color: darkblue;
font-size: 18px;
margin: 0 15px;
}
a:hover {
text-decoration: underline;
}
iframe {
border: 1px solid gray;
background-color: white;
}
</style>
</head>
<body>
```

```
<h1 align="center">Amrita Vishwa Vidyapeetham</h1>
<hr>
<center>
<a href="amaravati.html" target="content">Amaravati</a>
<a href="bengaluru.html" target="content">Bengaluru</a>
<a href="coimbatore.html" target="content">Coimbatore</a>
<a href="chennai.html" target="content">Chennai</a>
</center>
<br><br>
<table align="center">
<tr>
<td>
<iframe src="iframe_login.html" width="250" height="400"></iframe>
</td>
<td>
<iframe name="content" width="800" height="400"></iframe>
</td>
</tr>
</table>
</body>
</html>
```

amaravati.html:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
background-image: url("campus.jpg");
```

```
background-size: cover;
background-repeat: no-repeat;
background-position: center;
}
</style>
</head>
<body>
<center>
<h1>Amaravati Campus</h1>
<p>The Amrita Vishwa Vidyapeetham Amaravati campus is a 100-acre
co-educational facility located in Kuragallu, Andhra Pradesh.
Established in 2021, it is part of the prestigious
Amrita Vishwa Vidyapeetham university system, which holds an
A++ grade from NAAC.</p>
</center>
</body>
</html>
```

bengaluru.html:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
background-image: url("bengaluru.jpg");
background-size: cover;
background-repeat: no-repeat;
background-position: center;
}
```

```
</style>
</head>
<body>
<center>
<h1>Bengaluru Campus</h1>
<p>The Amrita Vishwa Vidyapeetham Bengaluru campus is a 25- to 50-acre
facility located in Kasavanahalli, off Sarjapur Road in South Bengaluru.
Established in 2002, the campus integrates modern academic infrastructure
with a serene environment featuring green lawns and a temple atmosphere.</p>
</center>
</body>
</html>
```

Coimbatore.html:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
background-image: url("campus2.jpg");
background-size: cover;
background-repeat: no-repeat;
background-position: center;
}
</style>
</head>
<body>
<center>
<h1>Coimbatore Campus</h1>
```

```
<p>The Amrita Vishwa Vidyapeetham Coimbatore campus is the headquarters and
first campus of the university system, situated on a lush 400-acre site
at the foothills of the Western Ghats in Ettimadai. As of late 2025 and
early 2026, the campus continues to lead in national rankings and
placement outcomes.</p>
</center>
</body>
</html>
```

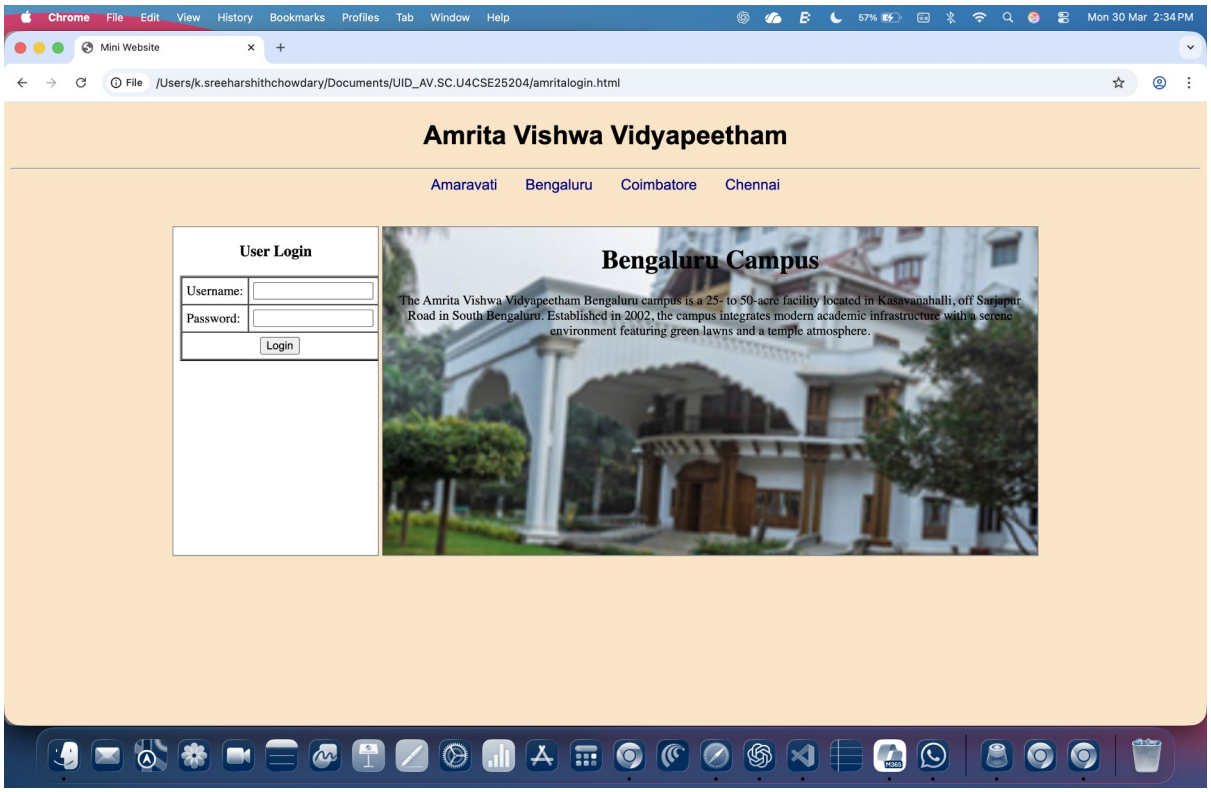
Chennai.html:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
background-image: url("chennai.jpg");
background-size: cover;
background-position: center;
}
</style>
</head>
<body>
<center>
<h1>Chennai Campus</h1>
<p>The Amrita Vishwa Vidyapeetham Chennai campus, established in 2019,
is a 13.5-acre modern facility located in Vengal Village,
Thiruvallur District. As part of a university ranked 8th best in India
by NIRF 2025, the campus primarily focuses on high-demand engineering
specialisations and research.</p>
```

</body>

</html>

Output:



Explanation of tags:

- <style> Used to write internal CSS to style the webpage elements.
- <hr> Creates a horizontal line to separate sections of the webpage.
- Defines a container used to group elements together for styling and layout.
- class Specifies a class name used to apply CSS styles to multiple elements.
- <img> Used to display images on the webpage.
- src Specifies the path of the image file.
- alt Provides alternate text for the image if it cannot be displayed.

- <button> Creates a clickable button element.
- <a> Defines a hyperlink used to navigate between pages.
- href Specifies the URL or page the link should open.
- target Specifies where to open the linked document.
- Used to center content horizontally on the webpage.
- <iframe> Used to embed another webpage within the current webpage.
- name Assigns a name to the iframe so links can target it.
- width Specifies the width of the iframe.
- height Specifies the height of the iframe.
- background-image is CSS property used to set an image as the background of the webpage.
- background-size Specifies how the background image should be scaled.
- background-repeat Controls whether the background image repeats or not.
- background-position Sets the starting position of the background image.
- text-decoration CSS property used to remove or add decoration like underline to links.
- color Specifies the text color.
- font-size Defines the size of the text.
- margin Adds space around elements.
- border Defines the border around an element.
- box-shadow Adds a shadow effect around an element.
- padding Adds space inside an element between the content and border.
- display Specifies the display behavior of an element.
- flex-wrap Allows flex items to wrap onto multiple lines.
- flex-direction Specifies the direction of flex items.
- justify-content Aligns flex items horizontally inside the container.

WEEK-6

(6) 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>
<style>
.container {
display: grid;
min-height: 100vh;
grid-template-rows: auto auto 1fr auto;
grid-template-columns: 200px 1fr;
grid-template-areas:
"header header"
"sale sale"
"sidebar products"
"footer footer";
gap: 10px;
padding: 10px;
}
.header {
grid-area: header;
background-color: #0568e9;
padding: 20px;
text-align: center;
font-size: 24px;
```

```
}
.sale {
grid-area: sale;
background-color: #f4f007;
padding: 20px;
text-align: center;
font-size: 18px;
}
.sidebar {
grid-area: sidebar;
background-color: #ddd8d8;
padding: 20px;
}
.sidebar label{
display:block;
background:white;
padding:10px;
margin:8px 0;
border-radius:5px;
cursor:pointer;
font-weight:bold;
}

.sidebar label:hover{
background:#0568e9;
color:white;
}
.sidebar h3 {
color: black;
```

```
margin-top: 0;
font: bold;
}
.products {
grid-area: products;
background-color: #e0e0e0;
display: grid;
grid-template-columns: repeat(2, 1fr);
gap: 8px;
padding: 10px;
}
.card {
background-color: white;
padding: 6px;
margin-bottom: 10px;
border-radius: 5px;
text-align: center;
}
.card img {
width: 100%;
height: 140px;
object-fit: contain;
border-radius: 6px;
}
.footer {
grid-area: footer;
background-color: #0568e9;
padding: 20px;
text-align: center;
```

```
font-size: 14px;
}
button {
background-color: #0568e9;
color: white;
padding: 10px 20px;
text-align: center;
font-size: 16px;
margin-top: 10px;
cursor: pointer;
border-radius: 5px;
}
button:hover {
background-color: #034a9e;
}
.card{
display:none;
}

#electronics:checked ~ .container .electronics{
display:block;
}

#clothing:checked ~ .container .clothing{
display:block;
}

#home:checked ~ .container .home{
display:block;
```

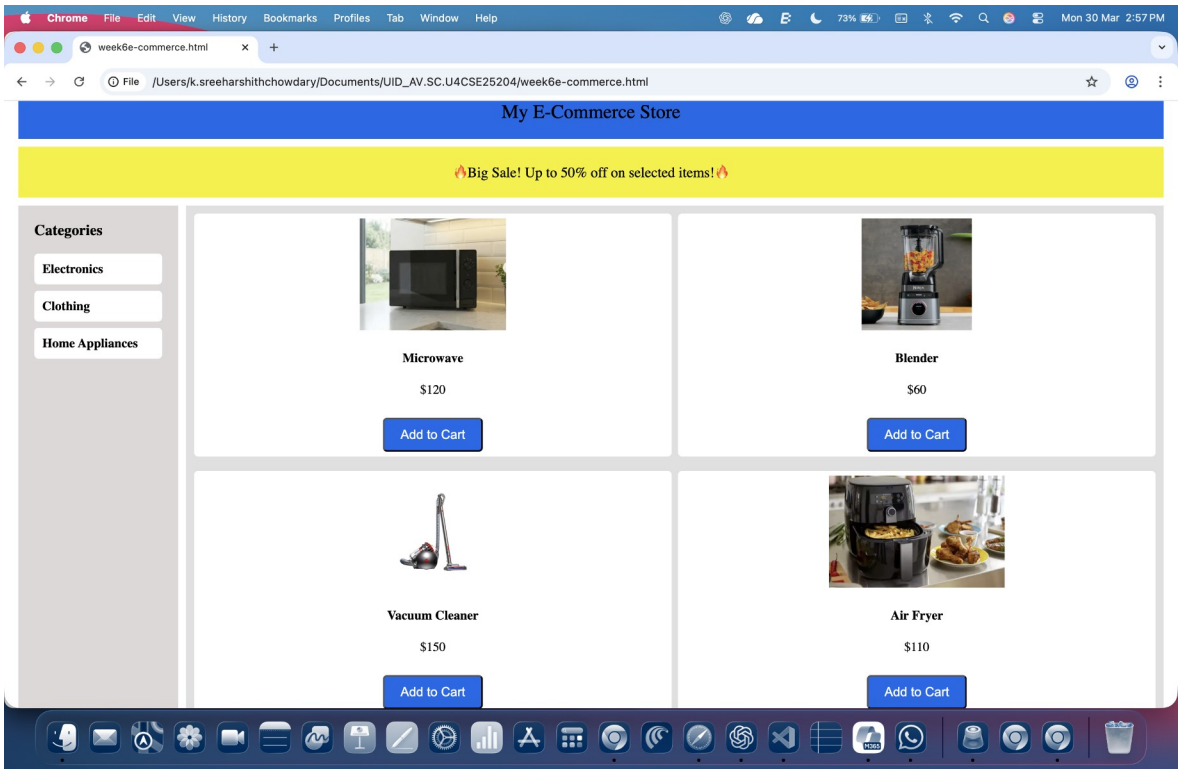
```
}
</style>
</head>
<body>
<input type="radio" name="category" id="electronics" hidden>
<input type="radio" name="category" id="clothing" hidden>
<input type="radio" name="category" id="home" hidden>
<div class="container">
<div class="header">
My E-Commerce Store
</div>
<div class="sale">
🔥Big Sale! Up to 50% off on selected items!🔥
</div>
<div class="sidebar">
<h3>Categories</h3>
<label for="electronics">Electronics</label>
<label for="clothing">Clothing</label>
<label for="home">Home Appliances</label>
</div>
<div class="products">
<div class="card electronics">
<img src="phone.jpg" alt="Product 1">
<h4>Phone</h4>
<p>$199</p>
<button>Add to Cart</button>
</div>

<div class="card electronics">
```

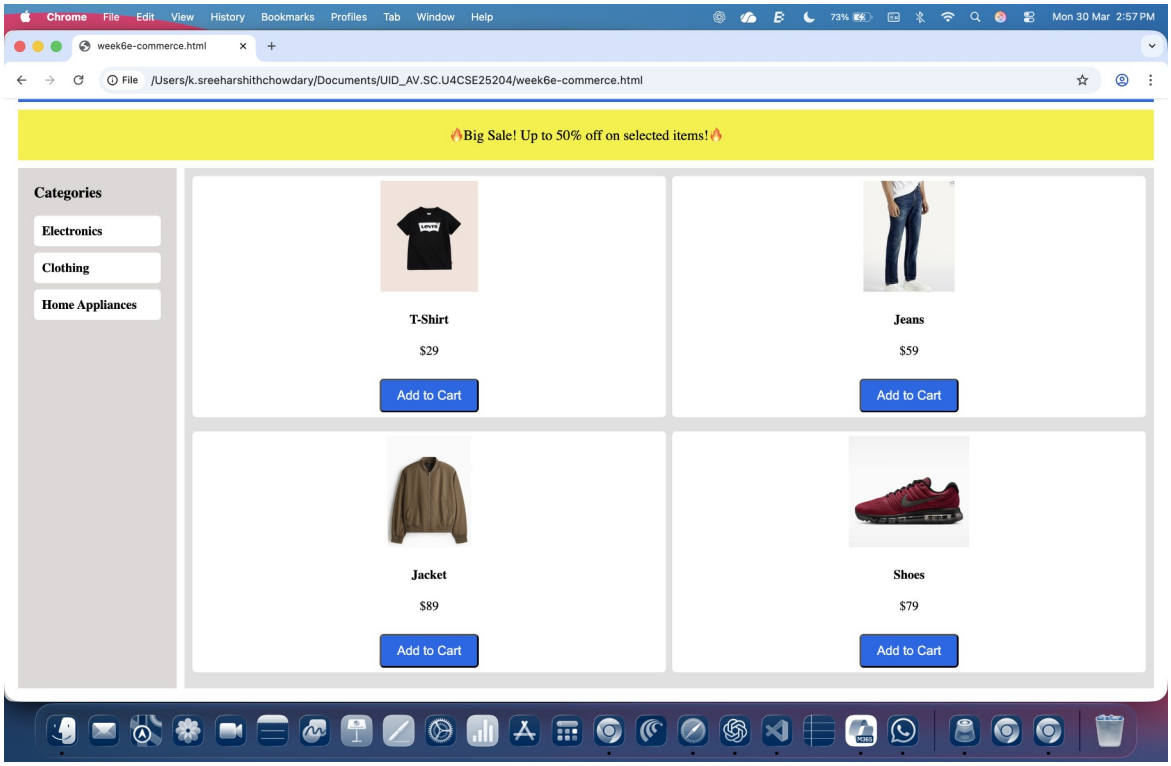
```
<img src="laptop.jpg" alt="Product 2">
<h4>Laptop</h4>
<p>$799</p>
<button>Add to Cart</button>
</div>

<div class="card electronics">
<img src="headphone.jpg" alt="Product 3">
<h4>Headphones</h4>
<p>$49</p>
<button>Add to Cart</button>
</div>

<div class="card electronics">
<img src="smartwatch.jpg" alt="Product 4">
<h4>Smart Watch</h4>
<p>$129</p>
<button>Add to Cart</button>
</div>

<div class="card clothing">
<img src="tshirt.jpg" alt="Product 5">
<h4>T-Shirt</h4>
<p>$29</p>
<button>Add to Cart</button>
</div>

<div class="card clothing">
<img src="jeans.jpg" alt="Product 6">
```

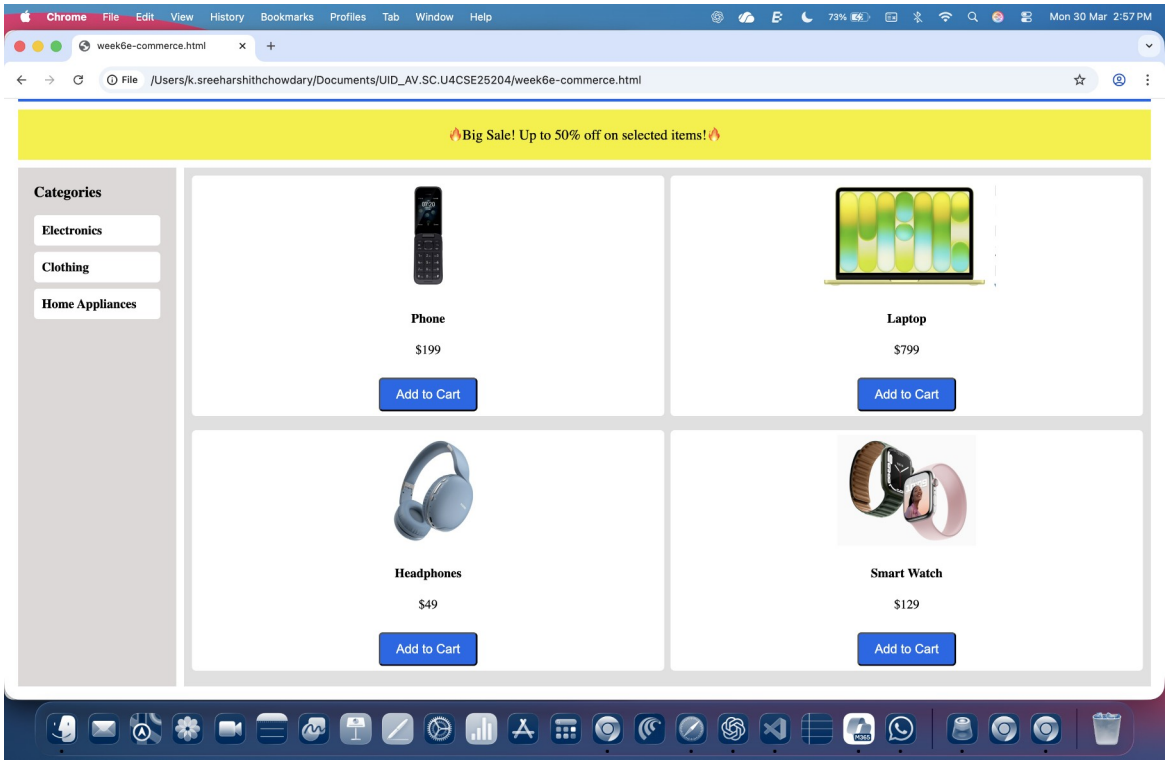
```
<h4>Jeans</h4>
<p>$59</p>
<button>Add to Cart</button>
</div>

<div class="card clothing">
<img src="jacket.jpg" alt="Product 7">
<h4>Jacket</h4>
<p>$89</p>
<button>Add to Cart</button>
</div>

<div class="card clothing">
<img src="shoes.jpg" alt="Product 8">
<h4>Shoes</h4>
<p>$79</p>
<button>Add to Cart</button>
</div>

<div class="card home">
<img src="oven.jpg">
<h4>Microwave</h4>
<p>$120</p>
<button>Add to Cart</button>
</div>

<div class="card home">
<img src="blender.jpg">
<h4>Blender</h4>
```

```
<p>$60</p>
<button>Add to Cart</button>
</div>

<div class="card home">
<img src="vacuumcleaner.jpg">
<h4>Vacuum Cleaner</h4>
<p>$150</p>
<button>Add to Cart</button>
</div>

<div class="card home">
<img src="airfryer.jpg">
<h4>Air Fryer</h4>
<p>$110</p>
<button>Add to Cart</button>
</div>

</div>
</div>
<div class="footer">
&copy; 2026 My E-Commerce Store | All rights reserved.
</div>
</div>
</body>
</html>
```

Output:


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 30 Mar 2:57 PM
week6e-commerce.html
File /Users/k.sreeharshithchowdary/Documents/UID_AV.SC.U4CSE25204/week6e-commerce.html
My E-Commerce Store
🔥Big Sale! Up to 50% off on selected items!🔥
Categories
Electronics
Clothing
Home Appliances
Microwave
$120
Add to Cart
Blender
$60
Add to Cart
Vacuum Cleaner
$150
Add to Cart
Air Fryer
$110
Add to Cart


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 30 Mar 2:57 PM
week6e-commerce.html
File /Users/k.sreeharshithchowdary/Documents/UID_AV.SC.U4CSE25204/week6e-commerce.html
🔥Big Sale! Up to 50% off on selected items!🔥
Categories
Electronics
Clothing
Home Appliances
T-Shirt
$29
Add to Cart
Jeans
$59
Add to Cart
Jacket
$89
Add to Cart
Shoes
$79
Add to Cart


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon 30 Mar 2:57 PM
week6e-commerce.html
File /Users/k.sreeharshithchowdary/Documents/UID_AV.SC.U4CSE25204/week6e-commerce.html
🔥Big Sale! Up to 50% off on selected items!🔥
Categories
Electronics
Clothing
Home Appliances
Phone
$199
Add to Cart
Laptop
$799
Add to Cart
Headphones
$49
Add to Cart
Smart Watch
$129
Add to Cart

Tags And Attributes used:

• Defines a container used to group elements together for layout and styling.

• <h3> Defines a level-3 heading used for the “Categories” title.

• <h4> Defines a level-4 heading used for product names.

• <p> Defines a paragraph used to display product prices.

• <img> Used to display product images on the webpage.

• <button> Creates a clickable button element used for actions like “Add to Cart”.

• <label> Defines a label for an input element and allows users to select categories by clicking it.

• <input> Used to create input controls such as radio buttons for category selection.

• class Specifies a class name used to apply CSS styles to multiple elements.

• id Specifies a unique identifier for an element.

• src Specifies the path of the image file to be displayed.

• alt Provides alternate text for the image if it cannot be displayed.

• type Specifies the type of input element (in this case, radio buttons).

• name Groups radio buttons together so only one category can be selected at a time.

• for Associates a label with a specific input element.

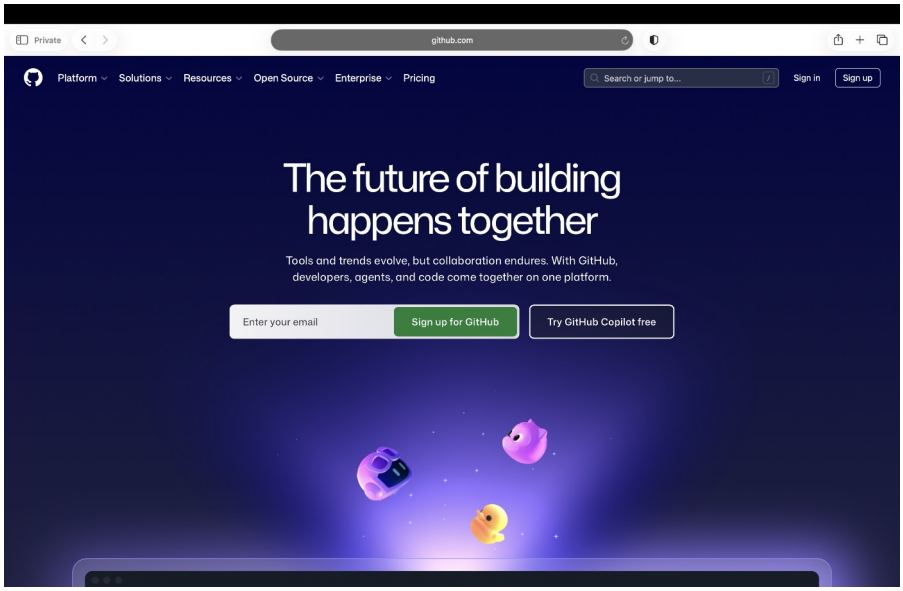
• hidden Hides the radio button input from the webpage.

WEEK 7

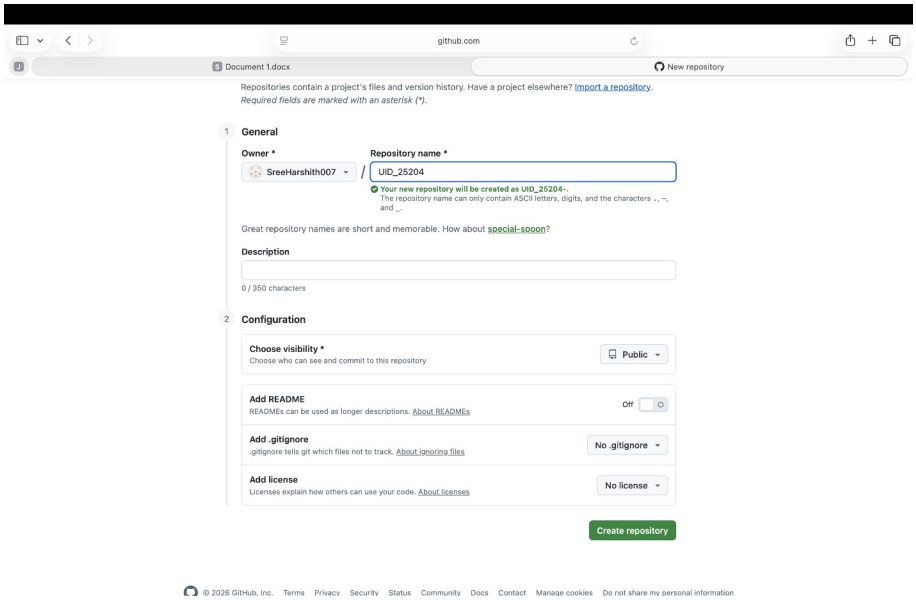
Aim: Create a GitHub account, a repository in that account and push all 6 weeks code files into the repository.

Procedure:

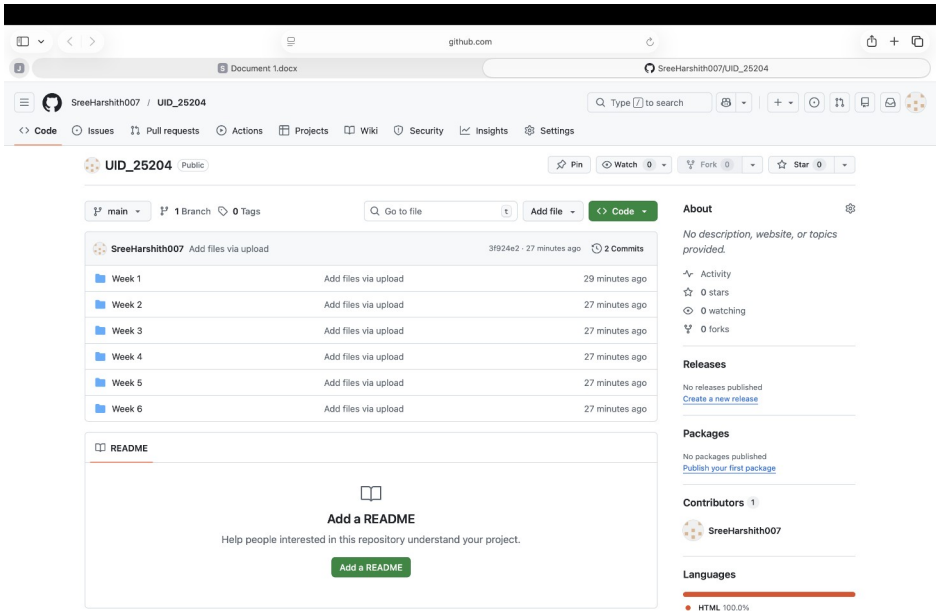
Step 1: Create a GitHub account.



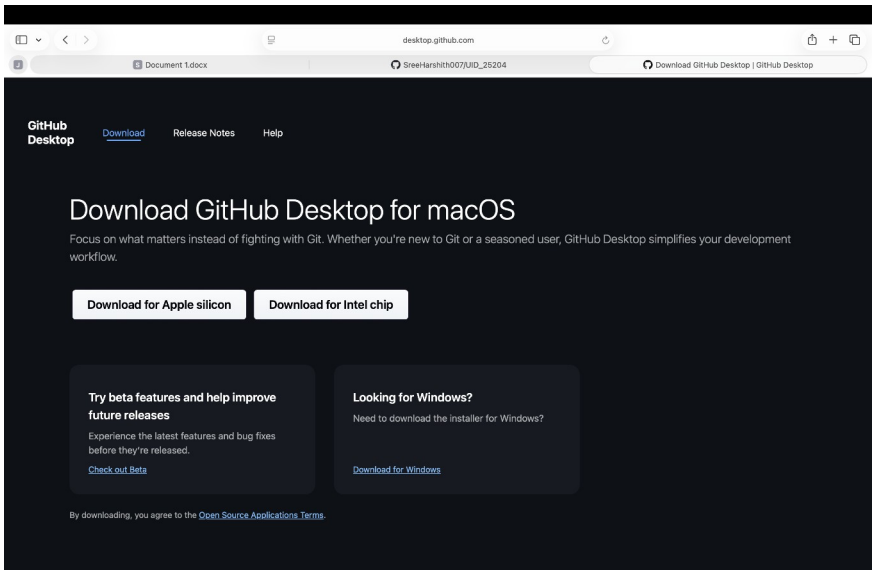
Step 2: Create a repository



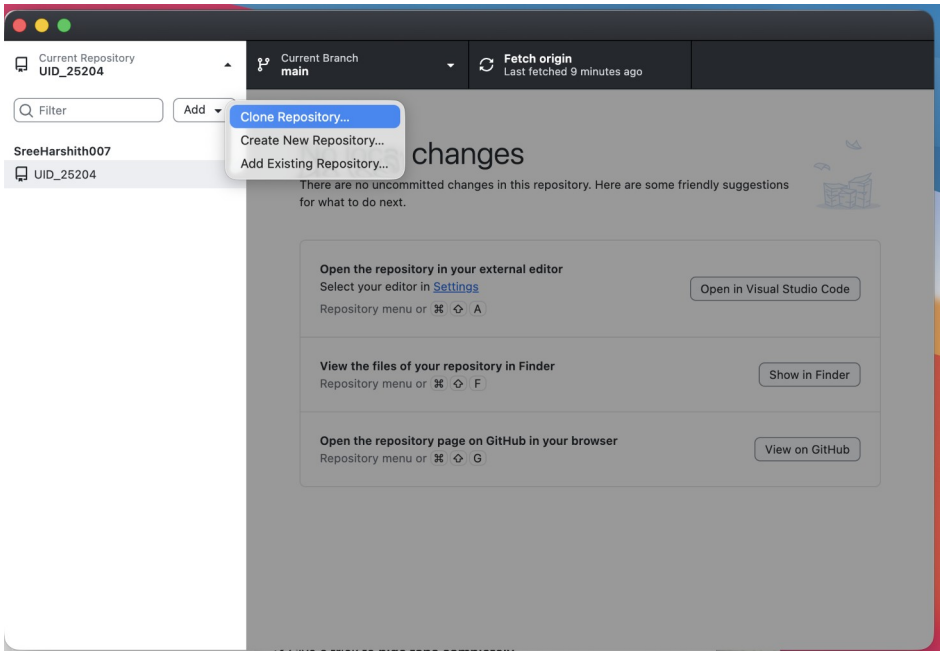
Step 3: Add files to repository.



Step 4: Download GitHub Desktop



Step 5: login to desktop and create a clone repository of UID_25204 in GitHub Desktop



Link To The Repository:

https://github.com/SreeHarshith007/UID_25204.git

WEEK 8

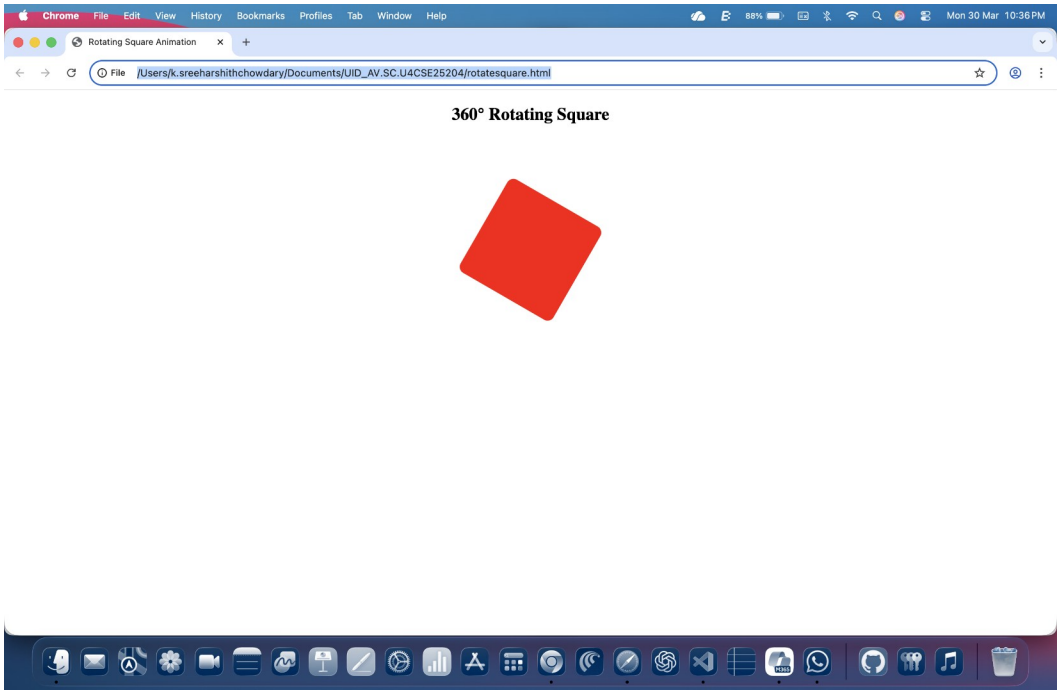
Aim: a) Create a webpage that displays a square that continuously rotates 360 degrees using CSS animations.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Rotating Square Animation</title>
<style>
body {
background-color:white;
}
.square {
width: 150px;
height: 150px;
background-color: red;
border-radius:10px;
margin: 100px auto;
animation-name: rotatesquare;
animation-duration: 2s;
animation-iteration-count: infinite;
animation-timing-function: linear;
}
@keyframes rotatesquare {
0% {
transform: rotate(0deg);
}
50% {
transform: rotate(180deg);
```

```
}
100% {
transform: rotate(360deg);
}
}
</style>
</head>
<body>
<h2 style="text-align:center;">360° Rotating Square</h2>
<div class="square"></div>
</body>
</html>
```

Output:



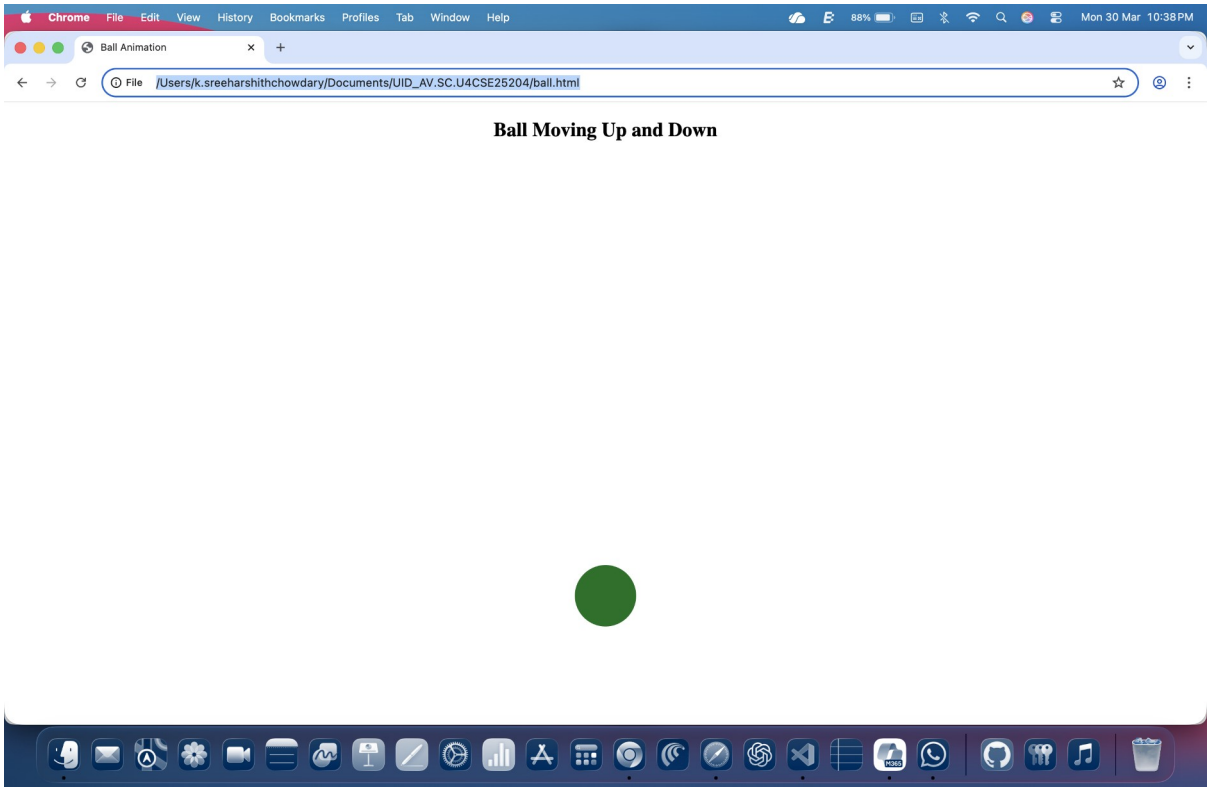
Aim: b) Create a ball that moves up and down continuously using css animation only.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Ball Animation</title>
<style>
body {
background-color: white;
}
.ball {
width: 75px;
height: 75px;
background-color: rgb(5, 113, 32);
border-radius: 50%;
margin: 0 auto;
position: relative;
animation-name: moveUpDown;
animation-duration: 2s;
animation-iteration-count: infinite;
animation-direction: alternate;
}
@keyframes moveUpDown {
from {
top: 0px;
}
```

```
to {
top: 500px;
}
}
</style>
</head>
<body>
<h2 style="text-align:center;">Ball Moving Up and Down</h2>
<div class="ball"></div>
</body>
</html>
```

Output:



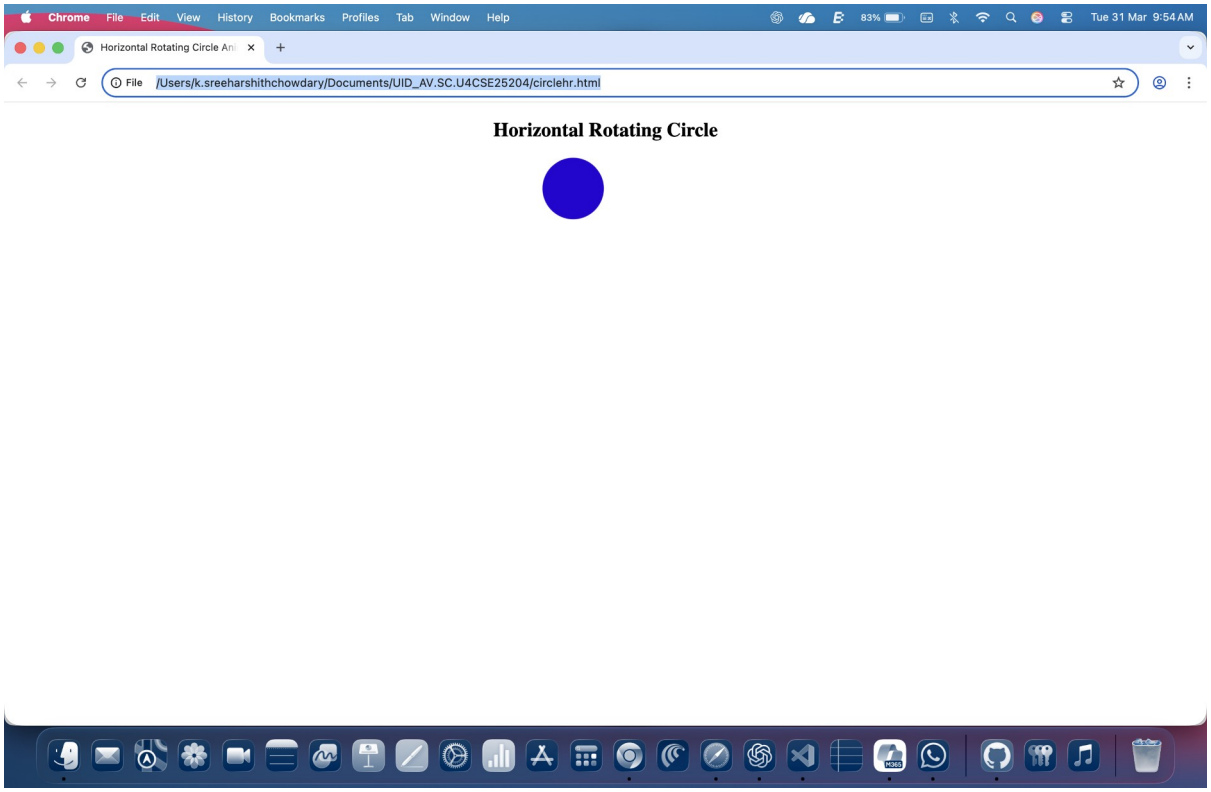
Aim: c) Create a circle that moves horizontally and rotates at the same time.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Horizontal Rotating Circle Animation</title>
<style>
body {
background-color: white;
}
.circle {
width: 75px;
height: 75px;
background-color:rgb(38, 4, 212);
border-radius: 50%;
position: relative;
animation: moveRotate 3s linear infinite;
}
@keyframes moveRotate {
0% {
left: 0px;
transform: rotate(0deg);
}
100% {
left: 1000px;
transform: rotate(360deg);
}
```

```
}
</style>
</head>
<body>
<h2 style="text-align:center;"> Horizontal Rotating Circle</h2>
<div class="circle"></div>
</body>
</html>
```

Output:



Explanation of tags:

1)<style>: It is used to write CSS inside an HTML document.

2)@Keyframes: The CSS @keyframes rule is used to control the steps in an animation sequence by defining CSS styles for points along the animation sequence.

3)animation-name: The animation-name property specifies a name for the @keyframes animation.

4)animation duration: The animation-duration property defines how long an animation should take to complete one cycle.

5)animation-iteration-count: The animation-iteration-count property specifies the number of times an animation should be played.

6)animation-timing-function: The animation-timing-function specifies the speed curve of an animation.

7)position: relative: The position property specifies the positioning type for an element. Element is positioned relative to its normal position in the document flow.

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.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Number Programs</title>
<style>
body {
margin: 0;
font-family: Arial;
background-color: #e6e6e6;
}
.header {
background-color: #1e6bd6;
color: white;
padding: 15px;
text-align: center;
font-size: 24px;
}
.container {
display: grid;
```

```
grid-template-columns: repeat(2, 1fr);
gap: 20px;
padding: 20px;
}
.card {
background: white;
padding: 20px;
border-radius: 10px;
text-align: center;
}
input {
padding: 10px;
width: 70%;
margin: 10px 0;
}
button {
background-color: #1e6bd6;
color: white;
border: none;
padding: 10px 15px;
margin: 5px;
cursor: pointer;
border-radius: 5px;
}
button:hover {
background-color: #1555a5;
}
.output {
margin-top: 10px;
```

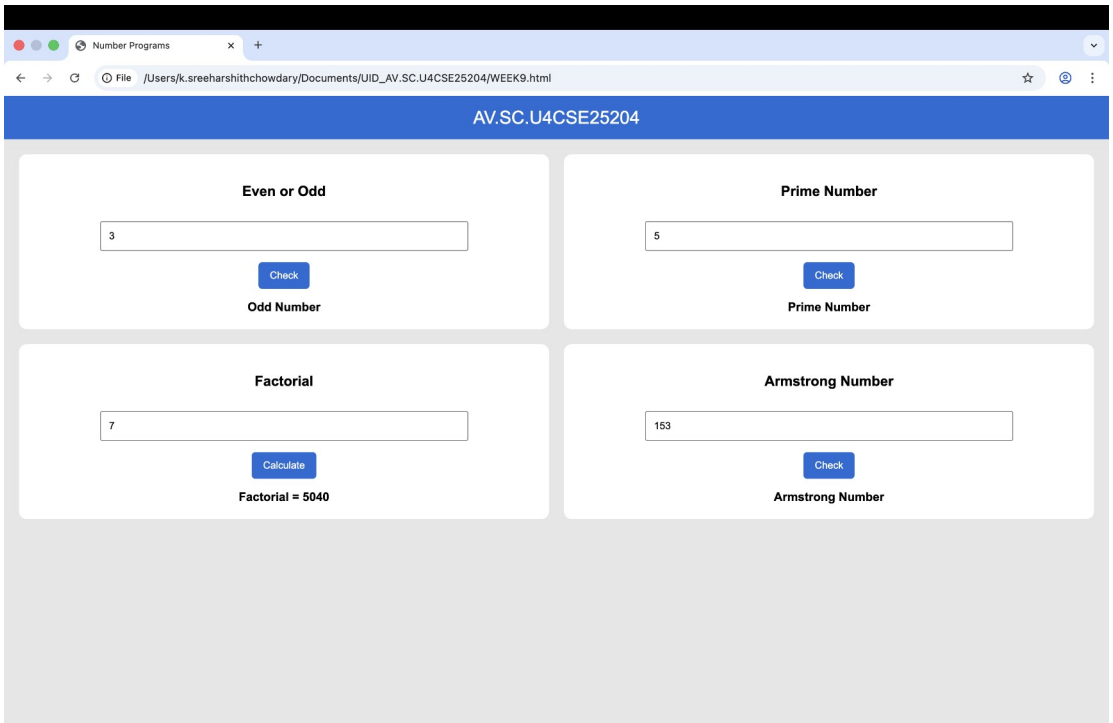
```
font-weight: bold;
}
</style>
</head>
<body>
<div class="header">AV.SC.U4CSE25204</div>
<div class="container">
<div class="card">
<h3>Even or Odd</h3>
<input type="number" id="eo">
<br>
<button onclick="checkEO()">Check</button>
<div class="output" id="eoOut"></div>
</div>

<div class="card">
<h3>Prime Number</h3>
<input type="number" id="prime">
<br>
<button onclick="checkPrime()">Check</button>
<div class="output" id="primeOut"></div>
</div>

<div class="card">
<h3>Factorial</h3>
<input type="number" id="fact">
<br>
<button onclick="factorial()">Calculate</button>
<div class="output" id="factOut"></div>
```

```
</div>

<div class="card">
<h3>Armstrong Number</h3>
<input type="number" id="arm">
<br>
<button onclick="checkArmstrong()">Check</button>
<div class="output" id="armOut"></div>
</div>
</div>
<script>
function checkEO() {
let n = parseInt(document.getElementById("eo").value);
if (n % 2 == 0) {
document.getElementById("eoOut").innerText = "Even Number";
} else {
document.getElementById("eoOut").innerText = "Odd Number";
}
}
function checkPrime() {
let n = parseInt(document.getElementById("prime").value);
let flag = true;
if (n <= 1) {
flag = false;
} else {
for (let i = 2; i <= Math.sqrt(n); i++) {
if (n % i == 0) {
flag = false;
break;
```

```
}
}
}
if (flag) {
document.getElementById("primeOut").innerText = "Prime Number";
} else {
document.getElementById("primeOut").innerText = "Not Prime";
}
}
function factorial() {
let n = parseInt(document.getElementById("fact").value);
let f = 1;

if (n < 0) {
document.getElementById("factOut").innerText = "Invalid input";
return;
}
for (let i = 1; i <= n; i++) {
f *= i;
}
document.getElementById("factOut").innerText = Factorial = " + f;
}
function checkArmstrong() {
let n = parseInt(document.getElementById("arm").value);
let temp = n;
let sum = 0;
while (temp > 0) {
let r = temp % 10;
sum += r * r * r;
```

```
temp = Math.floor(temp / 10);
}

if (sum == n) {
document.getElementById("armOut").innerText = "Armstrong Number";
} else {
document.getElementById("armOut").innerText = "Not Armstrong";
}
}
</script>
</body>
</html>
```

Output:



Explanation of Tags:

- <style> Used to apply internal CSS for designing the webpage layout and appearance.
- Defines sections such as header, container, and cards for grouping elements and layout.
- <input> Used to take numeric input from the user.
- <button> Creates clickable buttons to execute JavaScript functions.
- <script> Contains JavaScript code used to implement logic and interactivity.
- class Specifies a class name used to apply CSS styles to multiple elements (e.g., header, container, card).
- id Specifies a unique identifier for elements (used to access them in JavaScript).
- type Defines the type of input field (in this case, number input).
- onclick Executes a JavaScript function when the button is clicked.
- color Defines text color.
- padding Adds space inside elements.
- margin Controls spacing outside elements.
- display: grid Creates a grid layout for arranging cards.
- grid-template-columns Defines number of columns in grid (2 columns).
- gap Adds space between grid items.
- border-radius Rounds the corners of elements.
- text-align Aligns text to center.
- cursor Changes cursor style when hovering over buttons.
- document.getElementById() - Used to access input and output elements.
- InnerText - Used to display output inside HTML elements.

WEEK 10

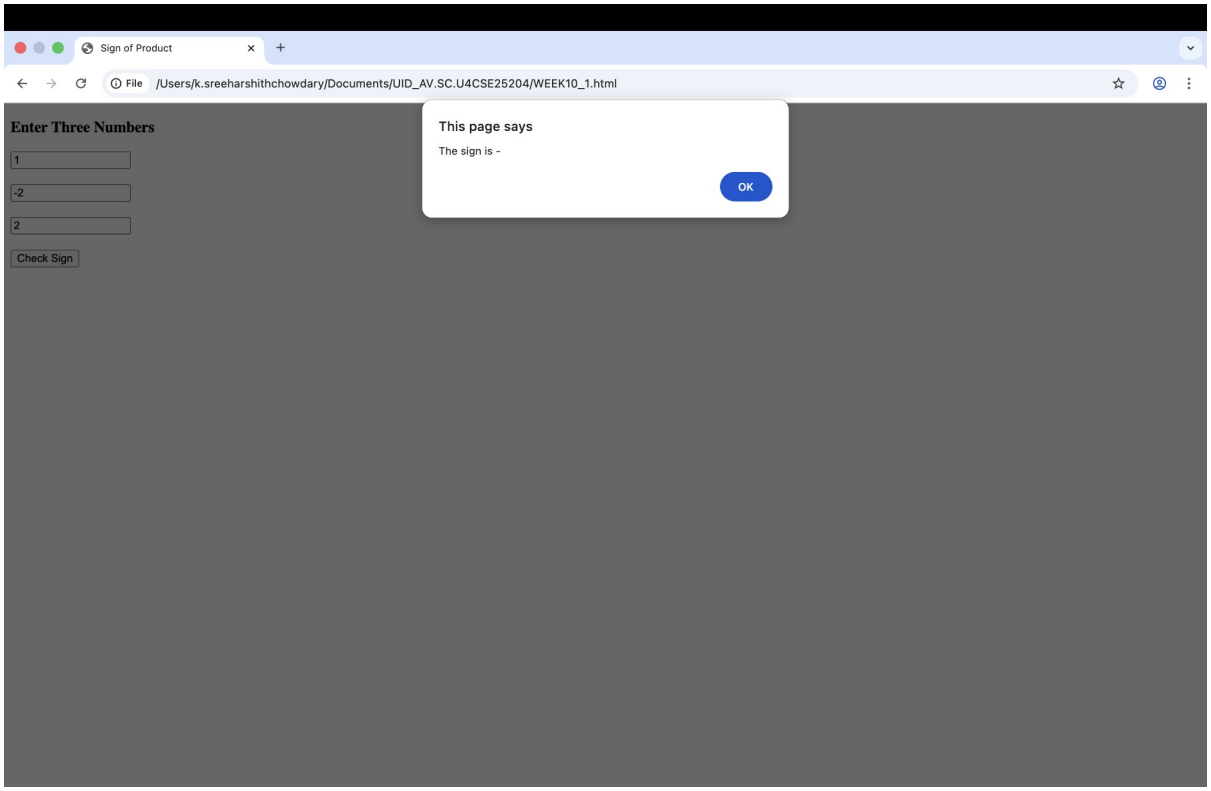
a)Write a Javascript conditional statement to find the sign of product of three numbers using functions. Display an alert box with the specified sign.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Sign of Product</title>
</head>
<body>
<h3>Enter Three Numbers</h3>
<input type="number" id="n1" placeholder="First number"><br><br>
<input type="number" id="n2" placeholder="Second number"><br><br>
<input type="number" id="n3" placeholder="Third number"><br><br>
<button onclick="findSign()">Check Sign</button>
<script>
function findSign() {
let a = parseInt(document.getElementById("n1").value);
let b = parseInt(document.getElementById("n2").value);
let c = parseInt(document.getElementById("n3").value);
let product = a * b * c;
if (product > 0) {
alert("The sign is +");
} else if (product < 0) {
alert("The sign is -");
} else {
alert("The sign is 0");
}
```

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

Output:



b) Write a Javascript conditional statement to sort three numbers. Display an alert box to show the result.

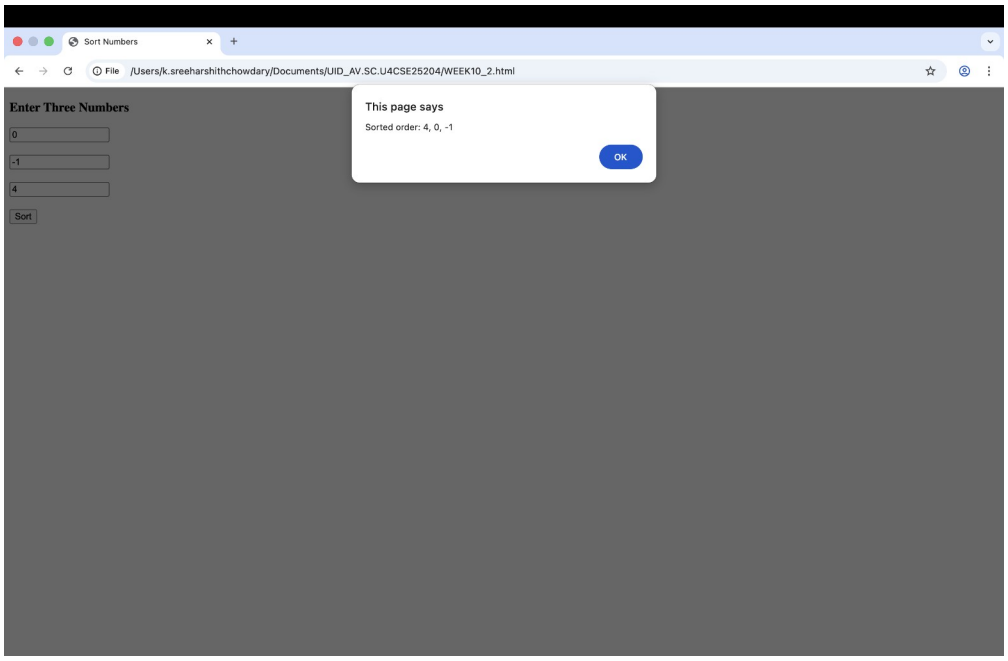
Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Sort Numbers</title>
</head>
<body>
<h3>Enter Three Numbers</h3>
```

```
<input type="number" id="n1" placeholder="First number"><br><br>
<input type="number" id="n2" placeholder="Second number"><br><br>
<input type="number" id="n3" placeholder="Third number"><br><br>
<button onclick="sortNumbers()">Sort</button>
<script>
function sortNumbers() {
let a = parseInt(document.getElementById("n1").value);
let b = parseInt(document.getElementById("n2").value);
let c = parseInt(document.getElementById("n3").value);
let first, second, third;
if (a >= b && a >= c) {
first = a;
if (b >= c) {
second = b;
third = c;
} else {
second = c;
third = b;
}
}
else if (b >= a && b >= c) {
first = b;
if (a >= c) {
second = a;
third = c;
} else {
second = c;
third = a;
}
```

```
}
else {
first = c;
if (a >= b) {
second = a;
third = b;
} else {
second = b;
third = a;
}
}
alert("Sorted order: " + first + ", " + second + ", " + third);
}
</script>
</body>
</html>
```

Output:



c) Write a Java script conditional statement to find the largest of five numbers using function and onclick event.

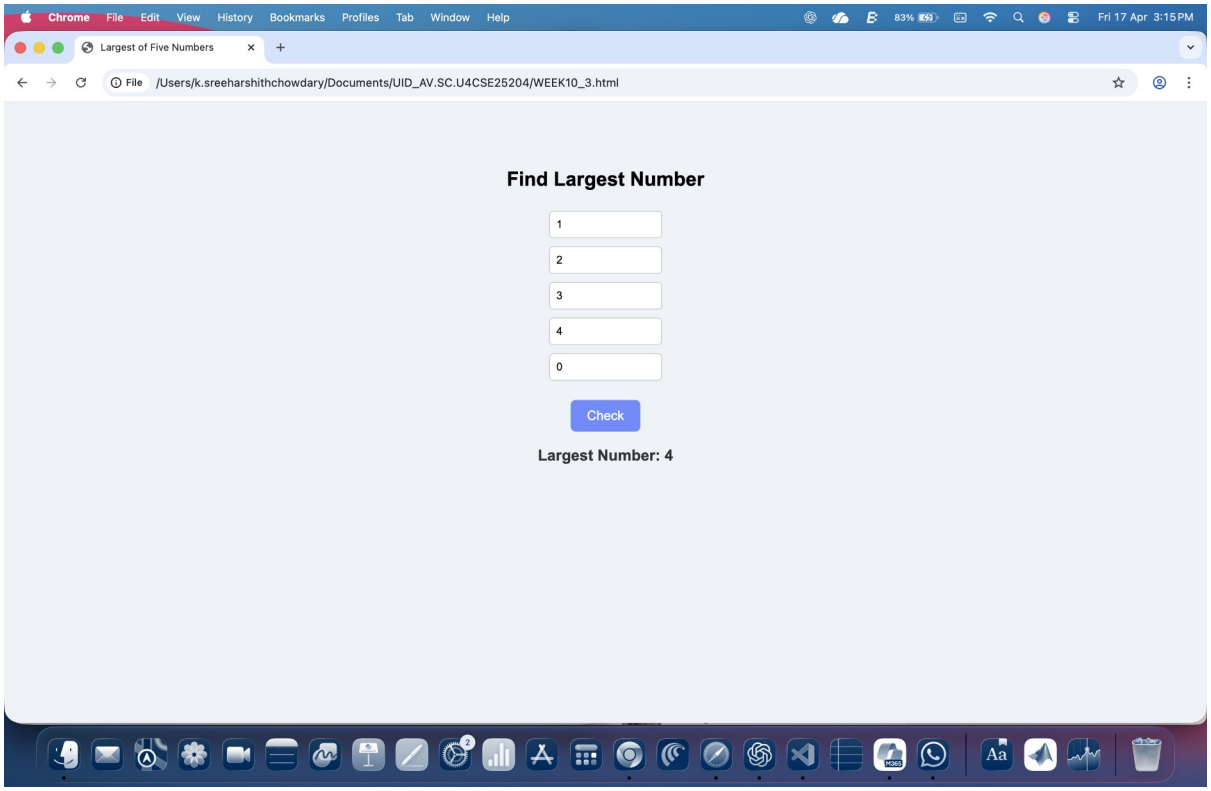
Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Largest of Five Numbers</title>
<style>
body {
font-family: Arial, sans-serif;
background-color: #eef2f7;
text-align: center;
margin-top: 80px;
}
input {
padding: 8px;
margin: 5px;
width: 120px;
border-radius: 5px;
border: 1px solid #ccc;
}
button {
padding: 10px 20px;
font-size: 16px;
background-color: #6c8cff;
color: white;
border: none;
border-radius: 6px;
```

```
cursor: pointer;
}
button:hover {
background-color: #4a6ee0;
}
h3 {
color: #333;
}
</style>
</head>
<body>
<h2>Find Largest Number</h2>
<input type="number" id="n1"><br>
<input type="number" id="n2"><br>
<input type="number" id="n3"><br>
<input type="number" id="n4"><br>
<input type="number" id="n5"><br><br>
<button onclick="check()">Check</button>
<h3 id="result"></h3>
<script>
function check() {
let a = parseInt(document.getElementById("n1").value);
let b = parseInt(document.getElementById("n2").value);
let c = parseInt(document.getElementById("n3").value);
let d = parseInt(document.getElementById("n4").value);
let e = parseInt(document.getElementById("n5").value);
let largest = a;
if (b > largest) largest = b;
if (c > largest) largest = c;
```

```
if (d > largest) largest = d;
if (e > largest) largest = e;
document.getElementById("result").innerHTML = "Largest Number: " + largest;
}
</script>
</body>
</html>
```

Output:



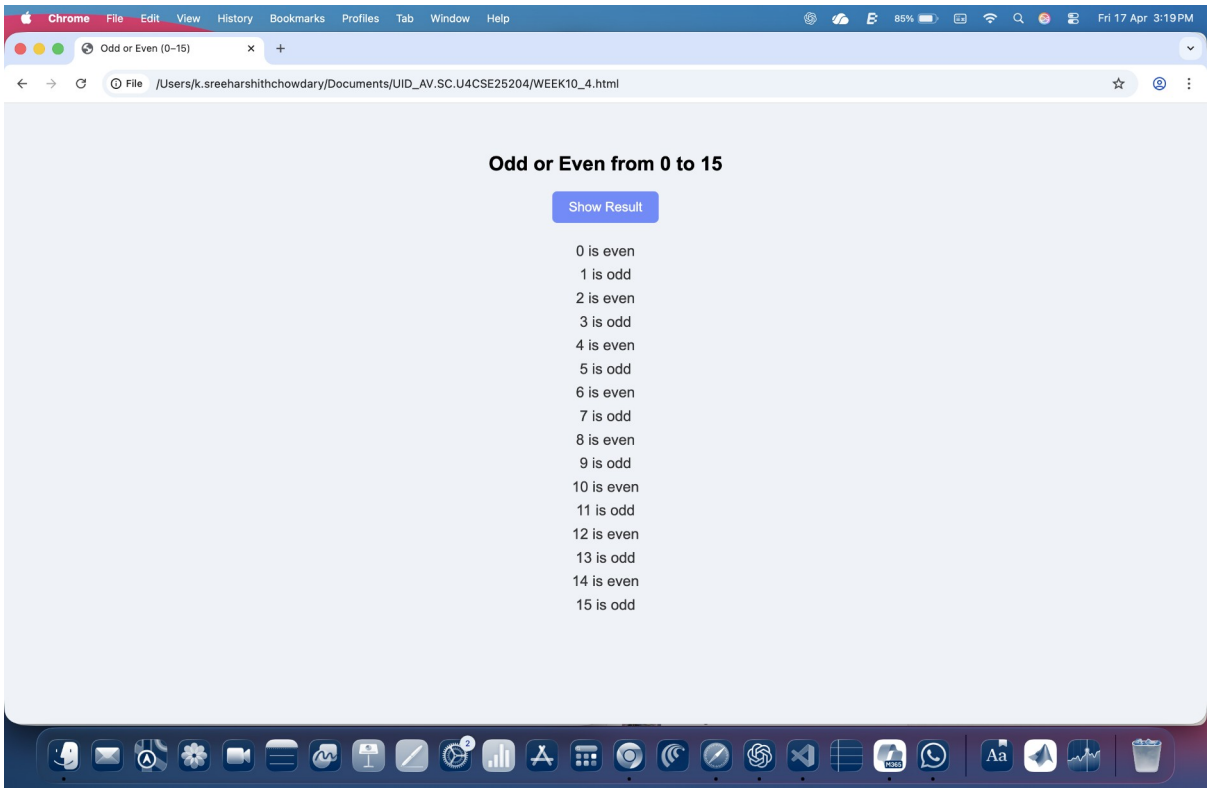
d) Write a Java script for loop that will iterate from 0 to 15 for each iteration it will check if the current number is odd or even and display a message to the screen.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Odd or Even (0–15)</title>
<style>
body {
font-family: Arial, sans-serif;
text-align: center;
background-color: #eef2f7;
margin-top: 60px;
}
button {
padding: 10px 20px;
font-size: 16px;
background-color: #6c8cff;
color: white;
border: none;
border-radius: 6px;
cursor: pointer;
}
button:hover {
background-color: #4a6ee0;
}
#output {
margin-top: 20px;
```

```
font-size: 18px;
color: #333;
line-height: 1.6;
}
</style>
</head>
<body>
<h2>Odd or Even from 0 to 15</h2>
<button onclick="checkNumbers()">Show Result</button>
<div id="output"></div>
<script>
function checkNumbers() {
let result = "";
for (let i = 0; i <= 15; i++) {
if (i % 2 === 0) {
result += i + " is even<br>";
} else {
result += i + " is odd<br>";
}
}
document.getElementById("output").innerHTML = result;
}
</script>
</body>
</html>
```

Output:



Explanation of Tags:

- <input> Used to take numeric input from the user.
-
 Inserts a line break for spacing between elements.
- <button> Creates a clickable button to execute the function.
- <script> Contains JavaScript code for logic and functionality.
- type="number" Specifies that the input field accepts only numeric values.
- id Assigns a unique identifier to each input field (n1, n2, n3).
- placeholder Displays hint text inside input boxes.
- onclick Calls the JavaScript function findSign() when the button is clicked.
- document.getElementById() - Used to access input values entered by the user.
- parseInt() - Converts the input values from string to integer.
- alert() - Displays the result (sign of the product) in a popup box.
- innerHTML write content inside HTML element

WEEK 11

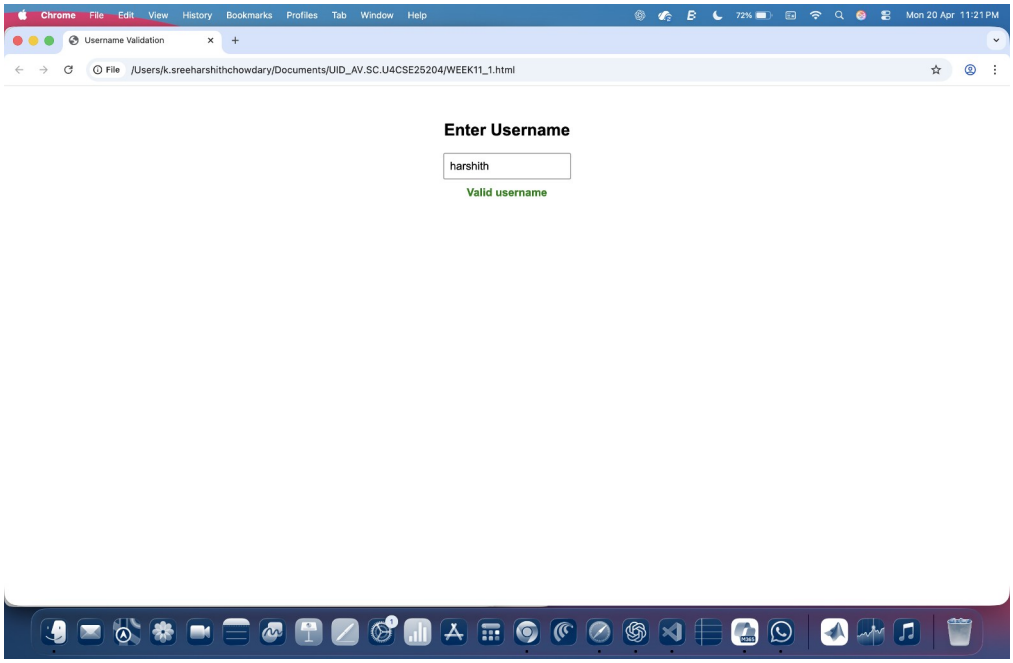
a) A website requires users to enter a username. The system should display a message while typing if the username is less than 5 characters.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Username Validation</title>
<style>
body {
font-family: Arial;
text-align: center;
margin-top: 50px;
}
input {
padding: 8px;
font-size: 16px;
}
#msg {
margin-top: 10px;
font-weight: bold;
}
</style>
</head>
<body>
<h2>Enter Username</h2>
<input type="text" id="username" oninput="checkUsername()">
<p id="msg"></p>
```

```
<script>
function checkUsername() {
let uname = document.getElementById("username").value;
let message = document.getElementById("msg");
if (uname.length < 5) {
message.innerHTML = "Username must be at least 5 characters";
message.style.color = "red";
} else {
message.innerHTML = "Valid username";
message.style.color = "green";
}
}
</script>
</body>
</html>
```

Output:



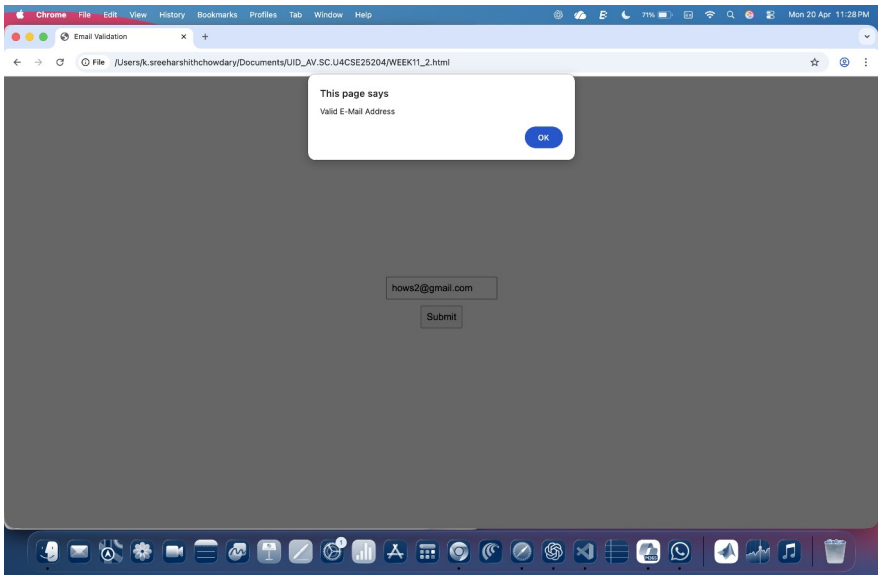
b) A registration form should validate email format before submission. If invalid, the form should not be submitted.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Email Validation</title>
<style>
body {
display: flex;
justify-content: center; /* horizontal center */
align-items: center; /* vertical center */
height: 100vh;
margin: 0;
font-family: Arial;
}
.container {
text-align: center;
}
input, button {
padding: 8px;
font-size: 16px;
margin: 5px;
}
</style>
<script>
function validateEmail() {
var email = document.getElementById("email").value;
var pattern = /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
```

```
if (!pattern.test(email)) {
alert("Invalid E-Mail Address");
return false;
}
alert("Valid E-Mail Address");
return true;
}
</script>
</head>
<body>
<div class="container">
<form onsubmit="return validateEmail()">
<input type="text" id="email" placeholder="Enter Email" required><br>
<button type="submit">Submit</button>
</form>
</div>
</body>
</html>
```

Output:



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

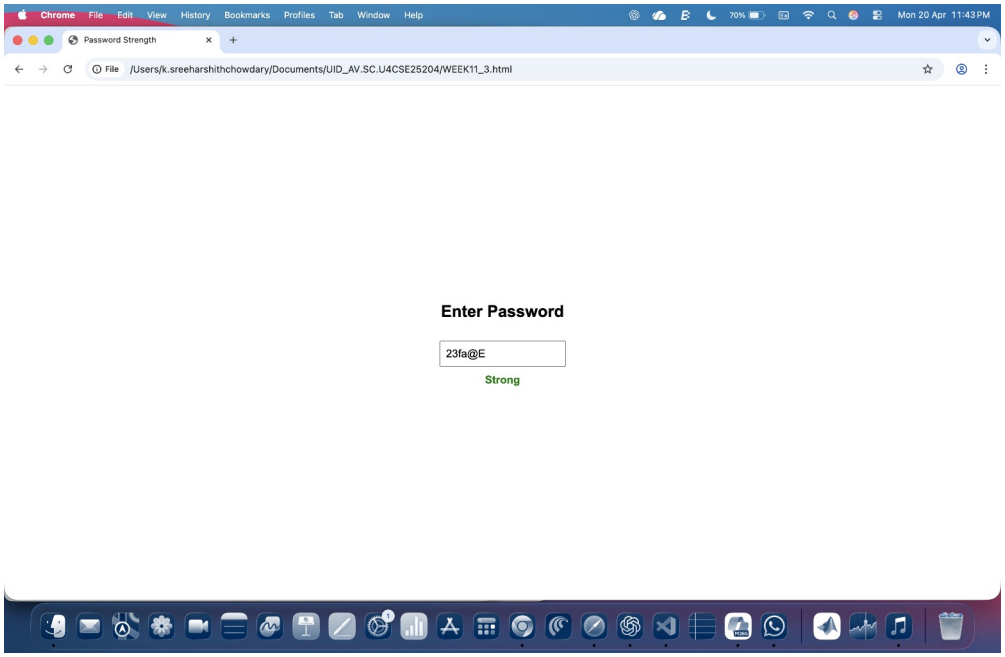
Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Password Strength</title>
<style>
body {
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
margin: 0;
font-family: Arial;
}
.container {
text-align: center;
}
input {
padding: 8px;
font-size: 16px;
margin-top: 10px;
}
#strength {
margin-top: 10px;
font-weight: bold;
}
</style>
```

```
<script>
function checkPassword() {
let pwd = document.getElementById("password").value;
let strength = document.getElementById("strength");
let hasUpper = /[A-Z]/.test(pwd);
let hasLower = /[a-z]/.test(pwd);
let hasNumber = /[0-9]/.test(pwd);
let hasSpecial = /[@$!%*?&]/.test(pwd);
if (pwd.length < 6) {
strength.innerHTML = "Weak";
strength.style.color = "red";
}
else if (hasUpper && hasLower && hasNumber && hasSpecial) {
strength.innerHTML = "Strong";
strength.style.color = "green";
}
else {
strength.innerHTML = "Medium";
strength.style.color = "orange";
}
}
</script>
</head>
<body>
<div class="container">
<h2>Enter Password</h2>
<input type="text" id="password" oninput="checkPassword()" placeholder="Enter
Password">
<p id="strength"></p>
```

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

Output:



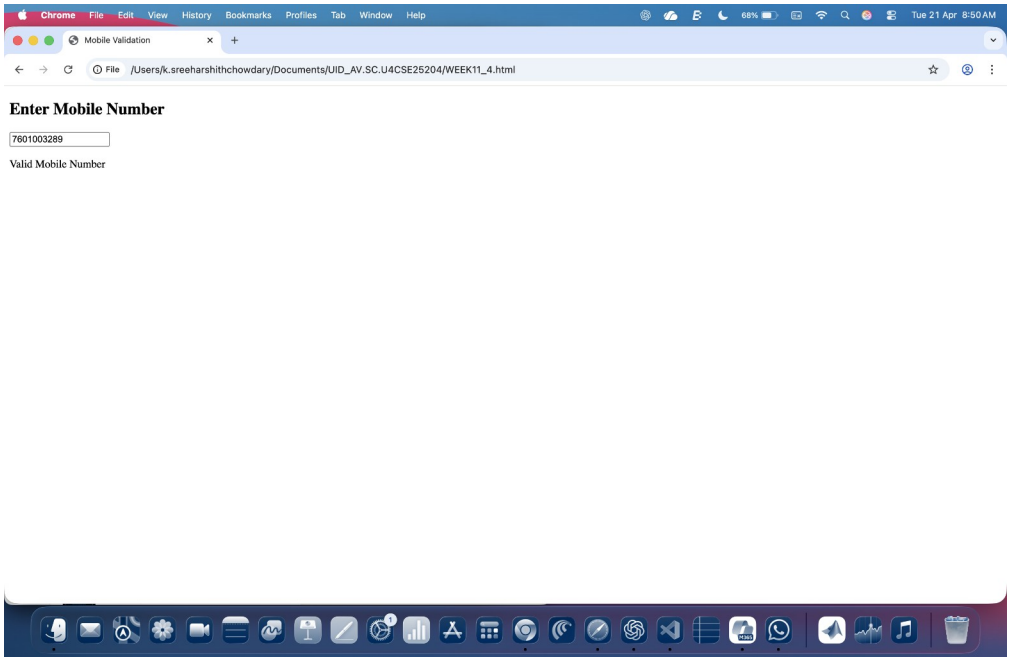
d) A form should validate a mobile number when the user leaves the input field. The number must be exactly 10 digits.

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Mobile Validation</title>
</head>
<body>
<h2>Enter Mobile Number</h2>
<input type="text" id="mobile" onblur="checkMobile()">
```

```
<p id="msg"></p>
<script>
function checkMobile() {
let number = document.getElementById("mobile").value;
let message = document.getElementById("msg");
if (number.length == 10 && number >= 0) {
message.innerText = "Valid Mobile Number";
} else {
message.innerText = "Invalid Mobile Number";
}
}
</script>
</body>
</html>
```

Output:



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

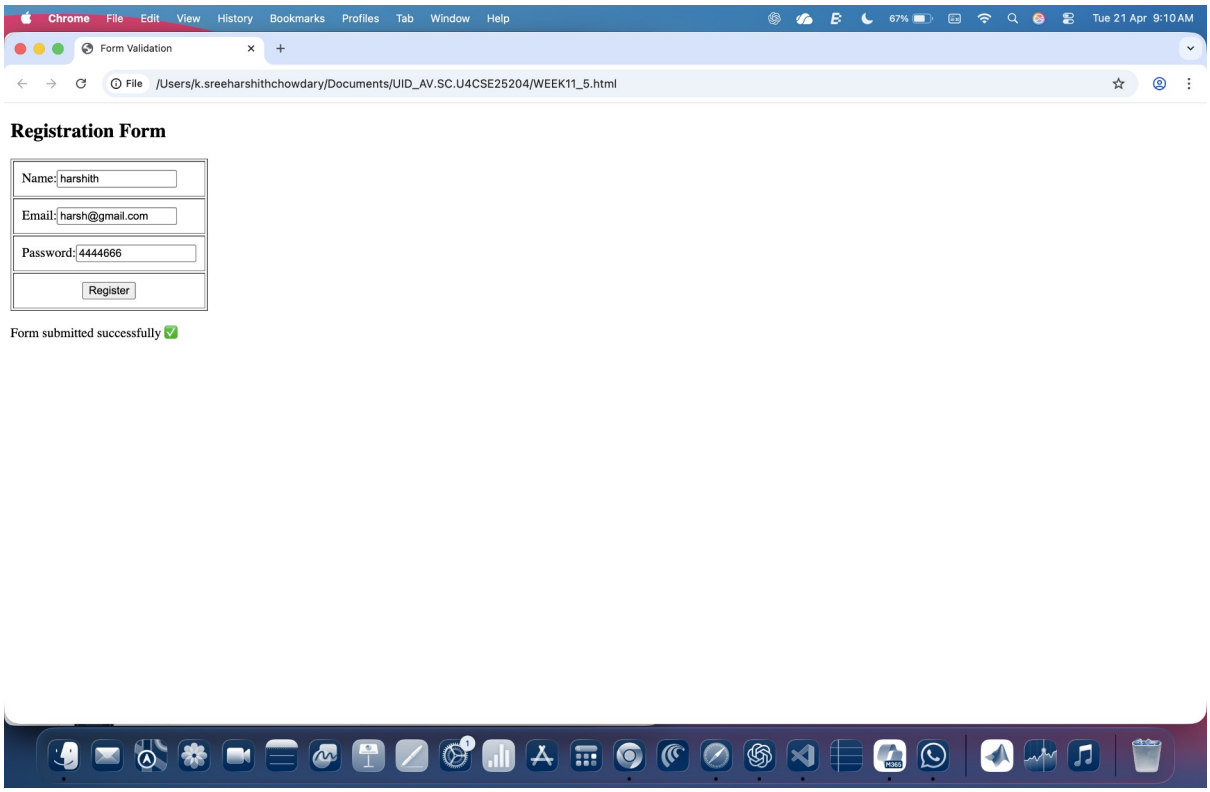
Name (not empty), Email (valid format), Password (minimum 6 characters).

Program:

```
<!DOCTYPE html>
<html>
<head>
<title>Form Validation</title>
</head>
<body>
<h2>Registration Form</h2>
<form onsubmit="return validateForm()">
<table border="1" cellpadding="10">
<tr>
<td>Name:<input type="text" id="name"></td>
</tr>
<tr>
<td>Email:<input type="text" id="email"></td>
</tr>
<tr>
<td>Password:<input type="text" id="password"></td>
</tr>
<tr>
<td colspan="2" align="center">
<input type="submit" value="Register">
</td>
</tr>
</table>
```

```
</form>
<p id="msg"></p>
<script>
function validateForm() {
let name = document.getElementById("name").value;
let email = document.getElementById("email").value;
let password = document.getElementById("password").value;
let pattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/;
if (name == "") {
document.getElementById("msg").innerHTML = "Name cannot be empty";
return false;
}
if (!email.match(pattern)) {
document.getElementById("msg").innerHTML = "Invalid Email";
return false;
}
if (password.length < 6) {
document.getElementById("msg").innerHTML = "Password must be at least 6 characters";
return false;
}
document.getElementById("msg").innerHTML = "Form submitted successfully ✅";
return false;
}
</script>
</body>
</html>
```

Output:



Explanation of tags and attributes:

1) <script>: It is used to write or include JavaScript code inside a webpage.

2) A function in JavaScript is a block of code designed to perform a specific task. It runs only when it is called (invoked).

3) getElementById(): It is a JavaScript method used to access an HTML element using its id attribute.

4) if-else: A conditional statement used to execute code based on a condition.

5) <table>: the <table> tag is used to arrange data into rows and columns.

6) <tr>: Creates a horizontal row.

7) <td>: <td> tag stands for table Data.

WEEK 12

Aim: Steps to install Electron framework.

Procedure:

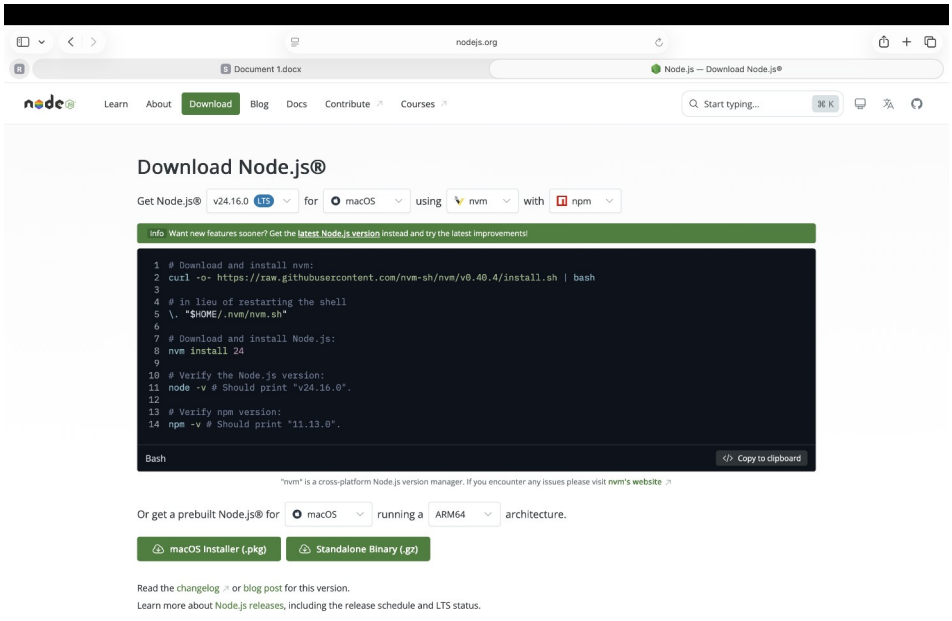
Step 1: Go to the web browser and download node.js.

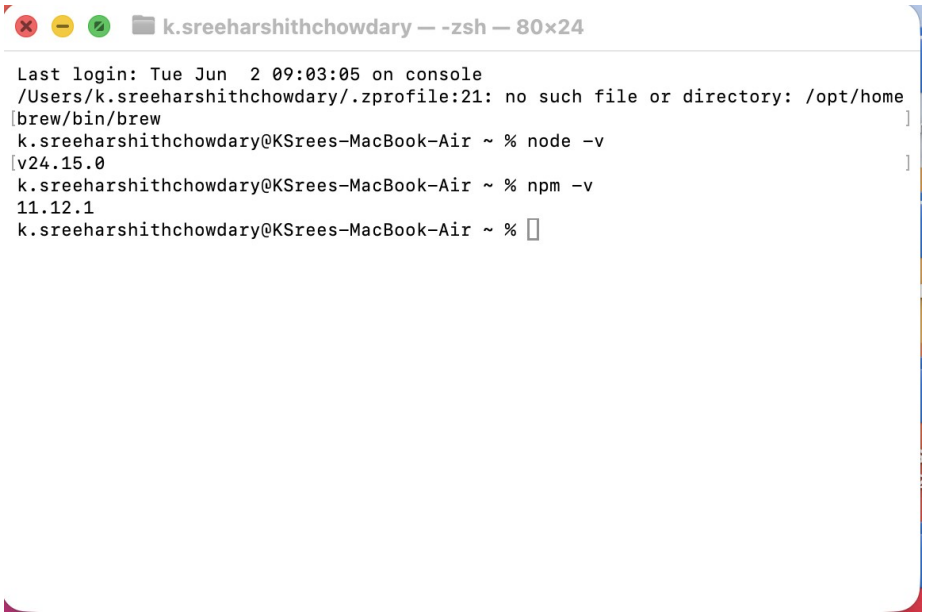
Step 2: After installing the Node.js, complete the setup process.

Step 3: After installing node.js, Open the terminal and navigate to the folder where your project is located.

Step 4: And give the prompt “node –v” and also "npm –v".

Step 5: If you get the versions then it is successfully installed.





Create Project Folder:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                zsh - electron-app
/Users/k.sreeharshithchowdary/.zprofile:21: no such file or directory: /opt/homebrew/bin/brew
k.sreeharshithchowdary@KSrees-MacBook-Air untitled folder 2 % mkdir electron-app
cd electron-app
k.sreeharshithchowdary@KSrees-MacBook-Air electron-app %
                                                                                Ln 1, Col 1   Spaces: 4   UTF-8
```

Step 6: Create a new folder for the Electron project using mkdir.

Step 7: Move into the project folder using cd.

- This helps keep all project files organized.

Step 8:

Initialize Node Project:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS                                zsh - electron-app
k.sreeharshithchowdary@KSrees-MacBook-Air electron-app % npm init -y
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "type": "commonjs"
}

k.sreeharshithchowdary@KSrees-MacBook-Air electron-app %
                                                                                Ln 1, Col 1   Spaces: 4   UTF
```

- The above command creates the package.json file.
- This file stores:

1.Project name

2.Version

3. Main entry file

4. Scripts

5. Dependencies

- Without package.json file, npm cannot manage project properly.

Step 9:

Install Electron:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

/Users/k.sreeharshithchowdary/.zprofile:21: no such file or directory: /opt/homebrew/bin/brew
k.sreeharshithchowdary@KSrees-MacBook-Air untitled folder 2 % npm install electron --save-dev

added 23 packages in 3s

2 packages are looking for funding
  run `npm fund` for details
k.sreeharshithchowdary@KSrees-MacBook-Air untitled folder 2 %
Ask about commands                                   Auto
zsh electro...
zsh
Ln 9, Col 3   Spaces: 2   UTF-8
```

• Downloads Electron package

• Saves it in node_modules

• Adds it to devDependencies

• --save-dev→Because Electron is needed during development.

Step 10:

Configure package.json:

```
index.html   main.js   render.js   style.css   package.json
electron-app > {} package.json > main
1  {
2    "name": "electron-app",
3    "version": "1.0.0",
4    "description": "",
5    "main": "main.js",
     ▷Debug
6    "scripts": {
7      "start": "electron ."
8
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "type": "commonjs"
14 }
15
```

Main→Tells Electron which file starts the application.

Start→Defines command used when we run:

Electron → This command runs Electron in the current project folder.

Create Main Application Files:

Required Files:

```
main.js
index.html
style.css
renderer.js
```

Step 11: main.js→Controls app lifecycle and creates windows

Step 12: index.html→User interface

Step 13: style.css→Design and layout

Step 14: renderer.js→Handles user interaction

Write main.js

• Imports Electron modules

• Creates application window

• Loads UI

```
package.json   main.js   index.html   style.
main.js > ...
const { app, BrowserWindow } = require('electron');

function createWindow() {
  const win = new BrowserWindow({
    width: 800,
    height: 600
  });

  win.loadFile('index.html');
}

app.whenReady().then(createWindow);
```

Run Application:

Command:

Bash

```
npm start
```