UNIVERSITY OF INFORMATION TECHNOLOGY & SCIENCES

Lab Manual

CSE 256

Web Application Design Lab

University of Information Technology and Sciences

Primitives

Department offering the course: Department of Computer Science and Engineering

Entity offering the program: University of Information Technology and Sciences (UITS)

Course Title: Web Application Design

Course Code: CSE 256

Academic Level/Term: 2 / 2

Credit Hour: 1.5

Contact Hour per Week: 3

Facilities Required for Teaching and Learning:

- PC
- Internet Connection

Server Needed (If any): No

Additional Requirements (if any): As the course emphasizes on online assignment, a total of 25 functional PCs should be kept available.

Assessment

Term Assessment:

Name of the Term Assignment: Create a Website using the technologies taught in the course.

Basic Requirement of the Term Assignment: Understanding of HTML, CSS and Javascript

Total Duration of the Assignment: 14 weeks.

Presentation Requirement: Yes.

Demonstration Requirement: Depending on the term project, demonstration may be conducted by respective advisors.

Reporting Requirement: As per advisor's decision.

Mark Distribution of a Particular Deliverable: No.

Schedule of the Assignment: Each student should present his/her progress to his/her supervisor in every one/two-week interval.

Final Viva:

Viva should be conducted by the respective project advisor (not any panel of teachers).

Quiz:

Necessity of conducting quiz: Quiz needs to be conducted based on the modular programming paradigms. Quiz must not include specific project-oriented questions.

Type of questions: MCQ, True/False, Small code snippet (like, possible outputs, any existing bug etc.)

Weights of Assessments

Assessments (Based on UCAM)	**Percentage**	**Breakdown Marks**
Class Attendance	20 %	Class Attendance (10)
		Class Performance (10)
Lab Performance	30 %	Lab Test 1 (10)
		Lab Test 2 (20)
Quiz	30%	Quiz (20)
		Lab Report (10)
Viva	20%	Project Presentation (100)
		Viva (10)
Total	**100%**	**100**

Mark Distribution of Class Attendance

Range of Percentage	**Marks**
90% - 100%	10
80% - 89%	9
70% - 79%	7-8
60% - 69%	5-6
Below 50%	1-4
Below 20%	0

Rubric for Assessment of Lab Performance

Criteria	Levels		
	Novice (0-3)	**Accomplished (4-7)**	**Expert (8-10)**
A. Laboratory Skills			
1. Manipulative Skills	Members do not demonstrate needed skills	Members occasionally demonstrate needed skills	Members always demonstrate needed skills
2. Experimental Set-up	Members are unable to set up the materials.	Members are able to set up the materials with supervision.	Members are able to set up the materials with minimum supervision
3. Process skills	Members do not demonstrate targeted process skills	Members occasionally demonstrate targeted process skills	Members always demonstrate targeted process skills
4. Oral Communication	Members are able to answer questions with many errors	Members are able to correctly answer with few errors	Members are able to correctly answer all questions
5. Safety Precautions	Minimum do not follow safety precautions	Members follow safety precautions most of the times	Members follow safety precautions at all times
B. Work Habits			
6. Time Management	Members do not finish on time with incomplete data	Members finish on time with incomplete data	Members finish Ahead of time with complete data
7. Preparedness	Members do not bring the required materials	Members bring only some of the required materials	Members bring all the required materials
8. Teamwork	Members do not know its task and have no defined responsibilities	Members are on task and have defined responsibilities most of the time.	Members are on task and have defined responsibilities all the times
9. Cleanliness	Messy workplace during and after the experiment	Clean and orderly workplace with occasional mess during and after the experiment	Clean and orderly workplace at all times during and after the experiment
10. Ability to work independently	Members require supervision by the teacher at all times	Members require occasional supervision by the teacher	Members do not need to be supervised by the teacher

Rubric for Assessment of Lab Experiment

Parameter	LOW (1)	MEDIUM (2-3)	HIGH (4-5)
Background Knowledge	Sufficient study and half of the questions have been answered correctly.	Adequate study and more than half of the questions have been answered correctly.	Thorough study and all the questions have been answered correctly.
Select Appropriate Equipment & Tools	Satisfactory selection of equipment's and tools to achieve desired objective without proper reasoning.	Satisfactory selection of equipments and tools to achieve desired objective along with proper reasoning.	Relevant and smart selection of equipments and tools to achieve desired objective along with proper reasoning.
Design	Inappropriate Design and contains few technical errors.	Design documentation is appropriately detailed and structured for the intended Purpose with less information	Design documentation is appropriately detailed and structured for the intended Purpose.
Logic Construction	Significant findings are summarized. Acceptable conclusion. Acceptable suggestion for further research.	Significant findings are summarized. Good conclusion. Good suggestion for further research.	Significant findings are summarized. Precisely concluded. Excellent suggestion for further research.
Documentation	The requirements of document writing are not properly addressed.	Document meets all prescribed requirements.	Document meets all requirements and it is prepared in original and creative way to engage readers.

Rubric for Lab Participation

Proficient (5)	Adequate (4-3)	Substandard (2)	Unacceptable (1)
Students demonstrate an accurate understanding of the lab objectives and concepts. The student can correctly answer questions and if appropriate, can explain concepts to fellow classmates. Students are eager to participate and assist when needed.	Students arrive on time to the lab, but may be unprepared. Answers to questions are basic and superficial suggesting that concepts are not fully grasped.	Student tardiness or unpreparedness makes it impossible to fully participate. If able to participate, students have difficulty explaining key lab concepts.	Student was absent from the lab or did not participate. There was no attempt to make prior arrangements to make up the lab.

Rubric for Lab Report

Excellent (9-10)	Good (6-8)	Average (3-5)	Poor (1-2)
Students demonstrate an accurate understanding of the lab objectives and concepts. Questions are answered completely and correctly. Graphs are neat, creative and include complete titles and accurate units. Errors, if any are minimal	Students have a basic knowledge of content, but may lack some understanding of some concepts. Questions are answered fairly well and/or graphs could have been done more neatly, accurately or with more complete information.	Students have problems with both the graphs and the answers. Student appears to have not fully grasped the lab content and the graph(s) possess multiple errors.	Students turn in lab reports late or the report is so incomplete and/or so inaccurate that it is unacceptable.

Rubric for Lab Viva

Poor (1-2)	Average (3-5)	Good (6-8)	Excellent (9-10)
Tells 1 or 2 events or key facts	Tells some of the events or key Facts	Tells many events, in sequence for the most part, or tells many key facts	Tells most events in sequence or tells most key facts
Includes few or no important details from topics	Includes some important details no Important details from topics	Includes many important details from topics	Includes most important details and key language or vocabulary from topics
Refers to 1 or 2 characters or topics using pronouns (he, she, it, they)	Refers to 1 or 2 characters or topics by generic name or label	Refers to many characters or topics	Refers to all characters or topics by specific name
Responds with incorrect information	Responds with some misinterpretation	Responds with literal interpretation	Responds with interpretation that shows higher level thinking
Provides limited or no response to teacher questions and prompts	Provides some response to teacher questions and prompts	Provides adequate response to teacher questions and prompts	Provides insightful response to teacher questions or prompts

Rubric for Lab Final

Excellent (9-10)	The student completed assigned tasks without any help from the instructor and showed the results appropriately.
Good (6-8)	The student completed assigned tasks with minimal help from the instructor and showed the results appropriately.
Average (3-5)	The student could not complete all assigned tasks and showed partial results
Poor (1-2)	The student did not complete assigned tasks.

Reference Material

Essential Books/Papers:

1. HTML & CSS: Design and Build Websites
2. JavaScript and JQuery: Interactive Front-End Web Development

Recommended Reference books:

1. Front-End Web Development: The Big Nerd Ranch Guide

Website (if any): https://www.w3schools.com/ , https://developer.mozilla.org/en-US/

Handout (if any): NONE

Others: NONE

Index

Experiments

Do’s & Don'ts in Laboratory

Do’s

1. Students should be punctual and regular to the laboratory.

2. Students should come to the lab in-time with a proper dress code.

3. Students should maintain discipline all the time and obey the instructions.

4. Students should carry observation and record completed in all aspects.

5. Students should be at their concerned experiment table; unnecessary moments are restricted.

6. Students should follow the indent procedure to receive and deposit the components from the lab technician.

7. While doing the experiments any failure/malfunction must be reported to the faculty.

8. Students should check the connections of the circuit properly before switching ON the power supply.

9. Students should verify the reading with the help of the lab instructor after completion of the experiment.

10. Students must ensure that all switches are in the lab OFF position, all the connections are removed.

11. At the end of practical class the apparatus should be returned to the lab technician and

take back the indent slip.

12. After completing your lab session SHUTDOWN the systems, TURNOFF the power switches and arrange the chairs properly.

13. Each experiment should be written in the record note book only after getting a signature from the lab in charge in the observation notebook.

Don’ts

1. Don’t eat and drink in the laboratory.

2. Don’t touch electric wires.

3. Don’t turn ON the circuit unless it is completed.

4. Avoid making loose connections.

5. Don’t leave the lab without permission.

6. Don’t bring mobiles into the laboratory.

7. Do not open any irrelevant sites on the computer.

8. Don’t use a flash drive on computers

1. Vision of CSE Department:

To scaffold the next generation of engineers and scientists in a student-centered learning environment to reach digital fluency in Computer Science & Engineering. The challenge is to make the students competent and skilled leaders in the wake of the ever-changing and challenging global work environment of the 21st century.

2. Mission of CSE Department:

- To provide outstanding education, training and research to the students with a view to making them good human beings as well.
- To build skilled engineers who can efficiently serve the society and nation as a whole.
- To enhance research activities to global level and also maintain industrial collaboration.

3. Course Objectives:

1. Realization of Boolean logic by using OR, AND, NOT, NAND, NOR, XOR gate.
2. Realization of Universal Gates (NOR and NAND)
3. Design and testing of a Half adder, Half Subtractor, Full adder and Full subtractor.
4. Verification of SR latch and different types of Flip-Flops
5. Construction of combinational logic circuits
6. Design of synchronous asynchronous and modulo counter

4. Course Outcomes (COs):

1. **Understanding** the basic components and design methodologies of a web application.
2. **Learning** web Programming Language HTML and designing the skeleton structure of a web page
3. **Learning** CSS ,Bootstrap and Javascript to design the front end user interface of a web application.
4. **Evaluate** existing systems and build new systems by applying state of the art technology.

6. Program Outcomes (POs)

PO_1 Engineering knowledge: Apply knowledge of mathematics, natural science and engineering fundamentals.
PO_2 Problem analysis: Identify, formulate, research literature and analyze complex engineering problems searching substantiated conclusions.
PO_3 Design/development of solutions: Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration.
PO_4 Investigation: Conduct investigations of complex problems using research-based knowledge and research methods.
PO_5 Modern tool usage: Create, select and apply appropriate techniques, resources, and modern engineering and IT tools.
PO_6 The engineer and society: Apply reasoning informed by contextual knowledge Modern tool usage: Create, select and apply appropriate techniques, resources, and modern engineering and IT tools. to assess societal, health, safety, legal and cultural issues.
PO_7 Environment and sustainability: Understand and evaluate the sustainability and impact of professional engineering work in the solution of complex engineering problems.
PO_8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
PO_9 Individual work and teamwork: Function effectively as an individual, and as a member or leader in diverse teams and in multi-disciplinary settings.
PO_10 Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO_11 Project management and finance: Demonstrate knowledge and understanding of engineering management principles, economic decision-making and apply these to multidisciplinary environments.
PO_12 Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

7. Mapping of the Course with COs and POs:

Course Outcome (CO) of the Course	Program Outcome (PO)											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1			3									
CO2		3										
CO3				3								
CO4				3								

Experiment No. 01:

Experiment Name / Title: Introduction to Web Design & Applications: Basic HTML Tags

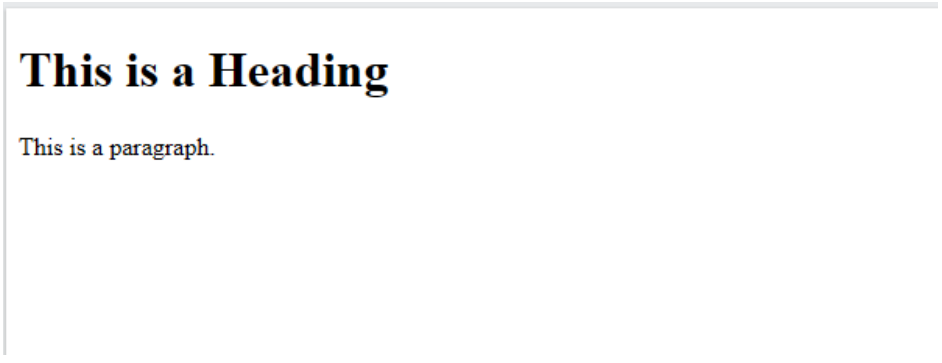
Aim or Objectives

1. To be able to create basic html elements in the web page.

Code or Script

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

Output:



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

<h1>HTML Links</h1>
<p><a href="https://www.google.com/">Visit Google!</a></p>
</body>
</html>
```

Output:

HTML Links

Visit Google!

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

<h2>HTML Image</h2>
<img src="pic_trulli.jpg" alt="Trulli" width="500" height="333">

</body>
</html>
```

Output:

HTML Image

Experiment No. 02:

Experiment Name / Title: HTML Tags to demonstrate Lists.

Objectives: To be able to create the following HTML element

- Unordered List
- Ordered List
- Definition List

Unordered List:

```
<html>
<head>
<title> Creating Unordered List </title>
</head>
<body bgcolor="pink">
<h1 align="center"> Creating Unordered List</h1>
<h1 align="center">List of Colleges in Kurnool</h1>
<ul type="square">
<li>GPREC</li>
<li>RGMCET</li>
<li>GPCET</li>
</ul>
</body>
</html>
```

Output:

Ordered List:

<html>

<head>

<title> Creating Order List </title>

</head>

<body bgcolor=”pink”>

<h1 align=”center”> Creating Order List</h1>

<h1 align=”center”>List of branches in RGMCET</h1>

<ol type=”A”>

*
CSE
*

*
IT
*

*
ECE
*

*
EEE
*

*
CIVIL
*

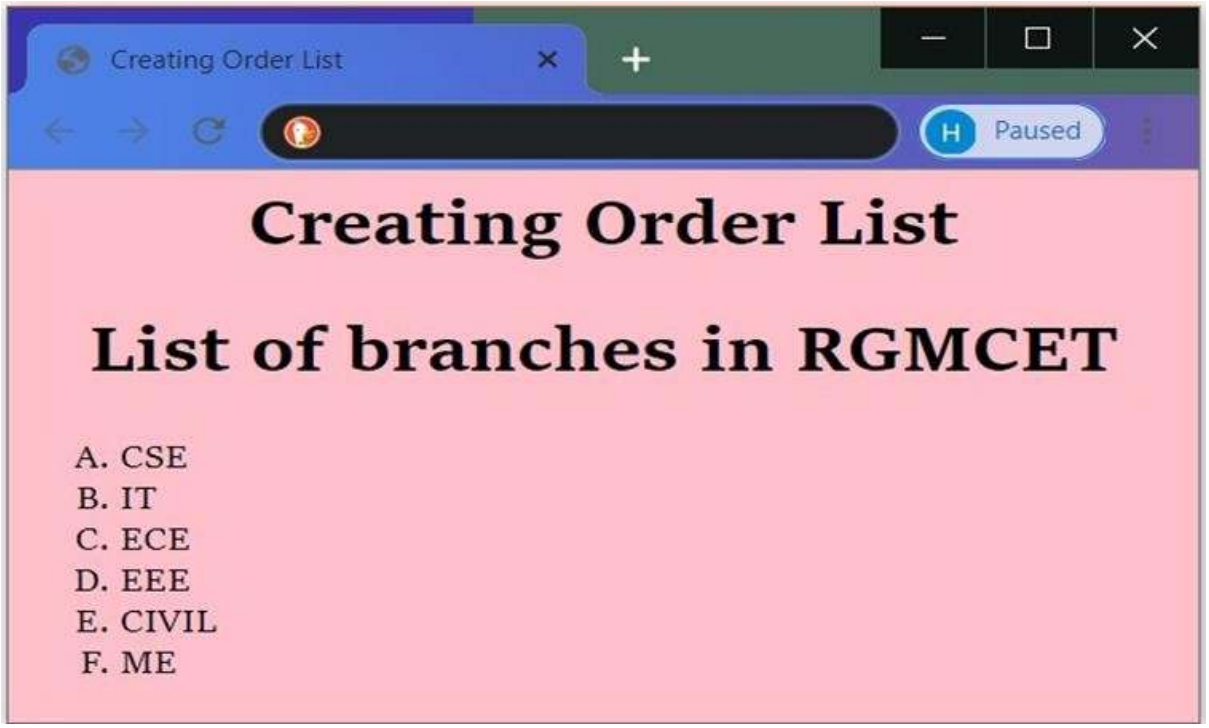
*
ME
*

*
*

</body>

</html>

Output:



Definition List:

<html>

<head>

<title>Creating Definition List</title>

</head>

<body bgcolor=”pink”>

<h1 align=”center”>Definition List</h1>

<dl>

<dt>CSE<dd>Computer Science & Engineering

<dt>ECE<dd>Electronics & Communication Engineering

<dt>IT<dd>Information Technology

<dt>EEE<dd>Electrical & Electronics Engineering

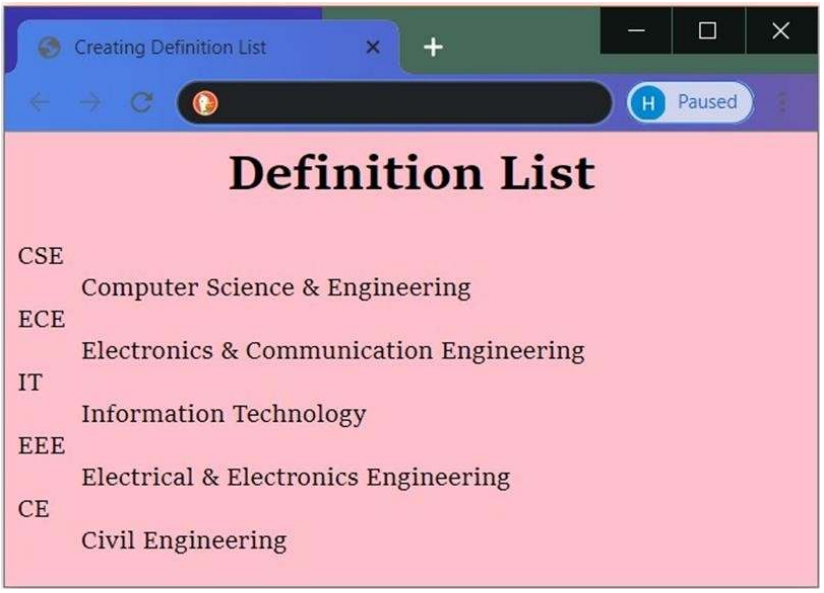
<dt>CE<dd>Civil Engineering

</dl>

</body>

</html>

Output:



Nested List:

<html>

<head>

<title>Nested Lists</title>

</head>

<body bgcolor=”pink”>

<h1 align=”center”>List of Colleges in Kurnool</h1>

<ol>

*
Kurnool
*

<ul>

*
GPREC
*

*
BITS
*

*
GPCET
*

*
*

*
Nandyala
*

<ul>

*
RGMCET
*

*
SREC
*

*
*

*
*

</body>

</html>

Output:

Experiment No. 03:

Experiment Name / Title: Write a HTML program for demonstrating Hyperlinks.

Objective:

- **Navigation from one page to another**
- **Navigation within the page**

Navigation from one page to another:

<html>

<head>

<title>Setting Hyperlink colors</title>

</head>

<body bgcolor="pink" link=”green” vlink=”blue” alink=”red”>

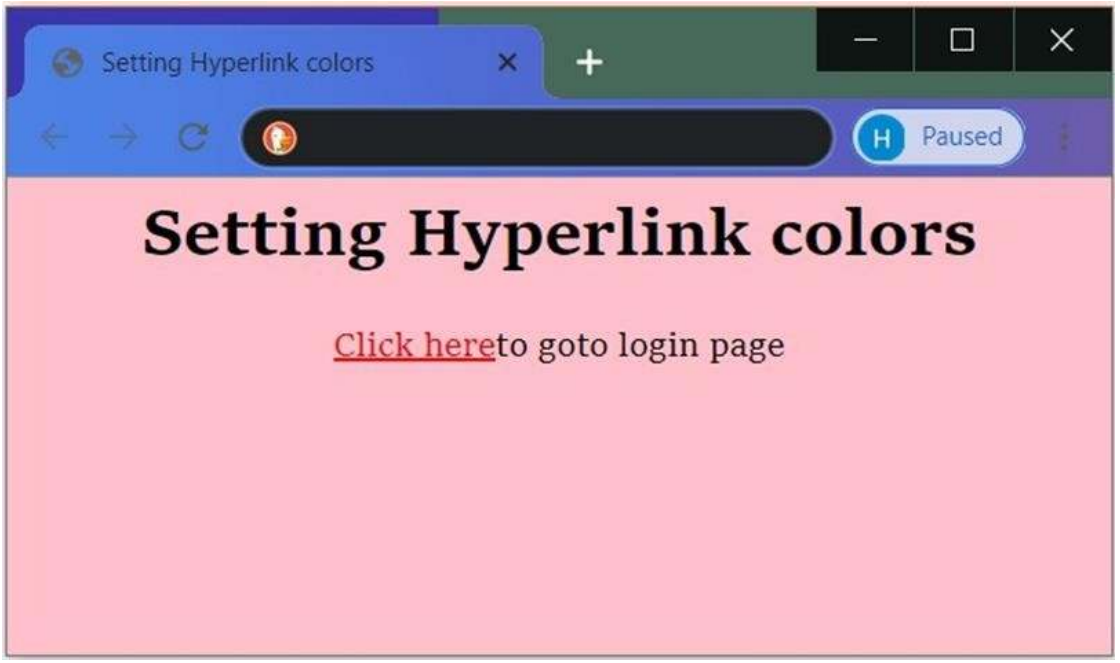
* <h1>Setting Hyperlink colors</h1>*

Click hereto goto login page

</body>

</html>

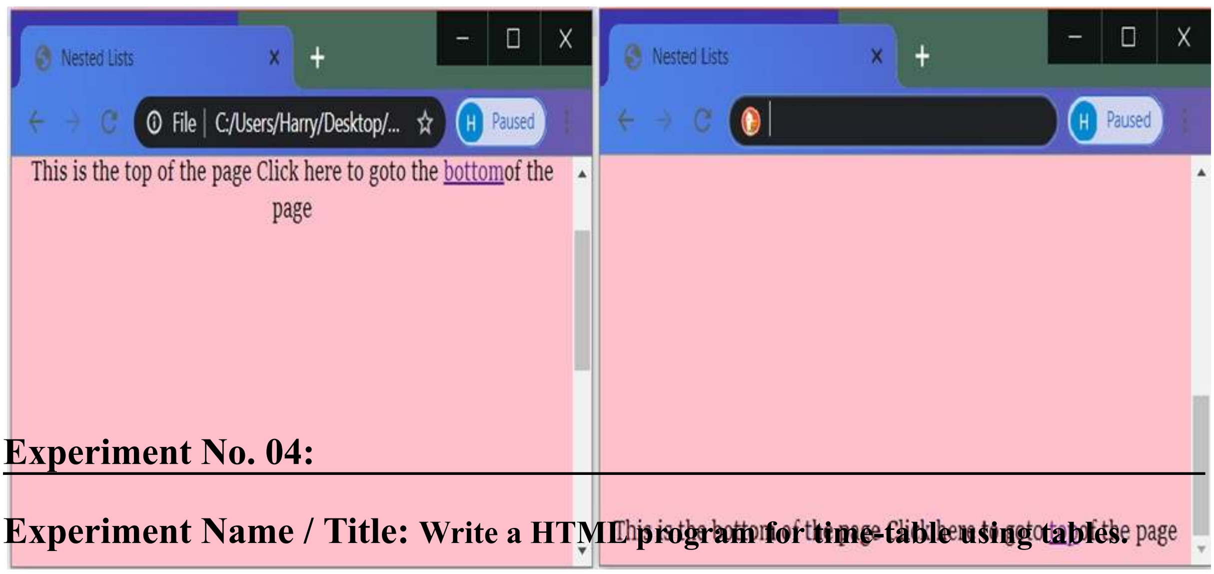
Output:



Navigation within the page:

```
<html>

<head>

<title>Nested Lists</title>

</head>

<body bgcolor="pink">

<center><h1>Linking to a section in a page</h1>

<a name=”top”>This is the top of the page</a>

Click here to goto the <a target=”#bottom”>bottom</a>of the page

<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>

<br><br><br><br><br>

<a name=”bottom”>This is the bottom of the page</a> Click here to goto <a target=”#top”>top</a>of the page

</center>

</body>

</html>
```

Output:



Experiment No. 04:

Experiment Name / Title: Write a HTML program for time-table using tables.

Objective: Showing different information using table in a webpage

Code Snippet:

```
<html>

<head>

<title>Timetable</title>

</head>

<body>

<h1 align="center"><fontcolor="Salmon">Timetable of III
CSE</font></h1><br>

<table align="center" border="2" cellspacing="0" cellpadding="15">

<tr align="center" valign=="middle">

<th>DAY</th>

<th>I</th>

<th>II</th>

<th rowspan="7"><b>T<br>E<br>A<br><br>B<br>R<br>E<br>A<br>K</b></th>

<th>III</th>

<th>IV</th>

<th
rowspan="7"><b>L<br>U<br>N<br>C<br>H<br><br>B<br>R<br>E<br>A<br>K<
/ b></th>

<th>V</th>

<th>VI</th>

<th>VII</th>

</tr>

<tr align="center">

<th>MON</th>
```

```
<td>IS</td>

<td>WT</td>

<td>SEM</td>

<td>OOAD</td>


<td>SCI</td>

<td>C#</td>

<td>COMP</td>


<tr align="center">

<th>TUE</th>

<td>AP</td>

<td>AP Lab</td>

<td colspan="2">AP Lab</td>

<td>WT</td>

<td>IS</td>

<td>OOAD</td>

</tr>

<tr align="center">

<th>WED</th>

<td>WT</td>

<td>IS</td>

<td>C#</td>

<td>SCI</td>

<td colspan="3">MOOC'S</td>
```

```
</tr>
<tr align="center">
<th>THU</th>
<td>IS</td>
<td>LIB</td>
<td>OOAD</td>
<td>WT</td>
<td colspan="3">WT Lab</td>
</tr>
<tr align="center">
<th>FRI</th>
<td>AP</td>
<td>AP</td>
<td>C#</td>
<td>OOAD</td>
<td colspan="3">C# Lab</td>

<tr align="center">
<th>SAT</th>
<td>OOAD</td>
<td>SCI</td>
<td>WT</td>
<td>SEM</td>
<td>AP</td>
<td>AP</td>
```

<td>C#</td>

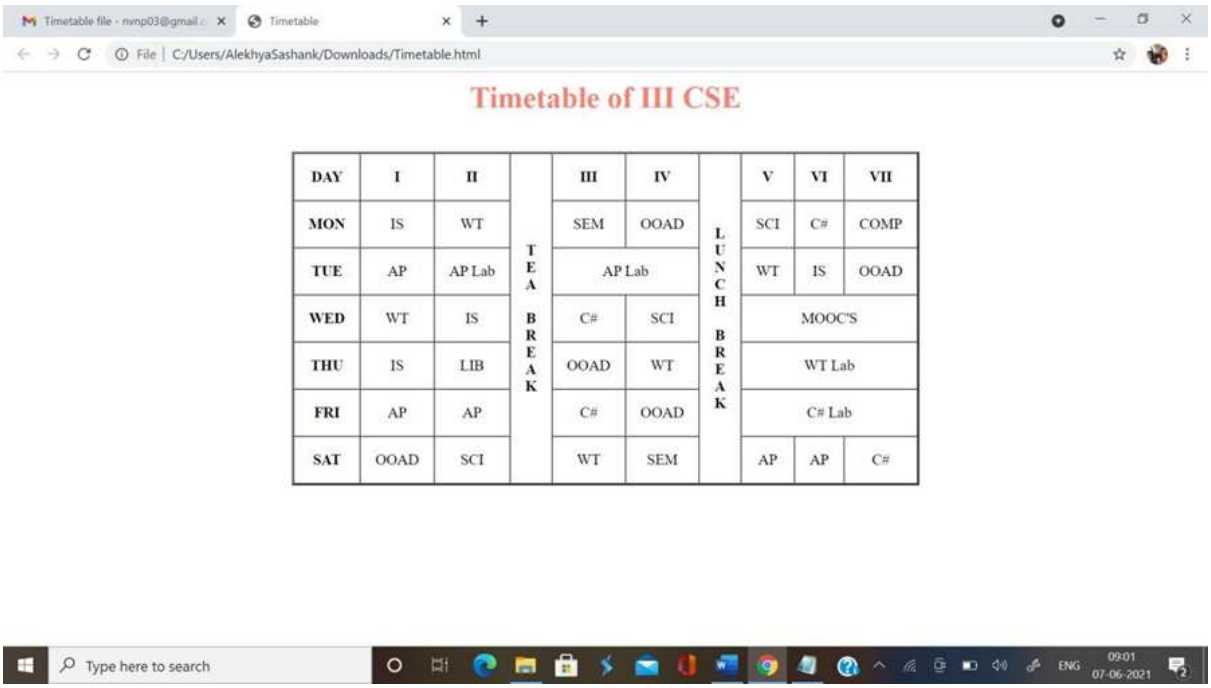
</tr>

</table>

</body>

</html>

Output:



Experiment No. 05:

Experiment Name / Title: Write a HTML program to develop a static Home Page using frames.

Objectives: Basic webpage creation using HTML and CSS

CODE:

```
<html>
<head>
<title>RGM ENGINEERING COLLEGE</title>
</head>
<frameset cols="30%,70%">
<frameset rows="25%,25%,50%">
<frame src="e:\cse546\logo.html">
<frame src="e:\cse546\home1.html">
<frame src="e:\cse546\courses.html">
</frameset>
<frameset rows="25%,25%,50%">
<frame src="e:\cse546\name.html">
<frame src="e:\cse546\table.html">
<frame src="e:\cse546\default.html" name="display">
</frameset>
</frameset>
</html>
```

Output:

Experiment No. 06:

Experiment Name / Title: Write a HTML program to develop a static Registration Form.

Objective: Basic HTML form creation for different websites

CODE:

<html>

<head>

<title>Registration</title>

</head>

<body bgcolor=lightblue>

<h1 align=center><u>Registration Form</u></h1>

*

*

* *

**

*First Name   <input type=text value=" " name="txt1">

 Last Name   <input type=text value=" " name="txt2">

 UserName   <input type=text*

*value="" name="txt3">

 Password   <input type=password value="" name="pwd1">
*

*Confirm Password   <input type=password value="" name="pwd2">

 Address   <textarea rows=3 cols=60></textarea>

*

Date of Birth   dd<select name="sel1">

<option>--</option>

<option>01</option>

<option>02</option>

<option>03</option>

<option>04</option>

<option>05</option>

<option>27</option>

<option>28</option>

<option>29</option>

<option>30</option>

<option>31</option>

</select>

mm<select name="sel2">

<option>--</option>

<option>01</option>

<option>02</option>

<option>03</option>

<option>04</option>

<option>05</option>

<option>06</option>

<option>07</option>

<option>08</option>

<option>09</option>

<option>10</option>

<option>11</option>

<option>12</option>

</select>

yyyy<select name="sel3">

<option> </option>

<option>1987</option>

<option>1988</option>

<option>1989</option>

<option>1990</option>

<option>1991</option>

<option>1992</option>

<option>1993</option>

<option>1994</option>

<option>1995</option>

<option>1996</option>

<option>1997</option>

<option>1998</option>

<option>1999</option>

<option>2000</option>

<option>2001</option>

<option>2002</option>

<option>2003</option>

<option>2004</option>

<option>2005</option>

<option>2006</option>

<option>2007</option>

<option>2008</option>

<option>2009</option>

<option>2010</option>

<option>2011</option>

<option>2012</option>

<option>2013</option>

<option>2014</option>

<option>2015</option>

<option>2016</option>

<option>2017</option>

*</select>

 Sex *

<input name="rb1" type="radio" value="radiobutton">Male

<input name="rb1" type="radio" value="radiobutton">Female

*

*

*Martial Status *

<input name="rb2" type="radio" value="radiobutton">Single

<input name="rb2" type="radio" value="radiobutton">Married

*

*

*Mobile Number   <input type=text name="txt4">

 Branch *

<input name="rb3" type="radio" value="radiobutton">CSE

<input name="rb3" type="radio" value="radiobutton">IT

<input name="rb3" type="radio" value="radiobutton">ECE

<input name="rb3" type="radio" value="radiobutton">EEE

<input name="rb3" type="radio" value="radiobutton">MECH

*

*

*Languages Known *

<input name="cb1" type="checkbox" value="checkbox">English

<input name="cb1" type="checkbox" value="checkbox">Telugu

<input name="cb1" type="checkbox" value="checkbox">Hindi

<input name="cb1" type="checkbox" value="checkbox">Kannada

<input name="cb1" type="checkbox" value="checkbox">Tamil

*

*

* *

*<input type=submit value="SUBMIT" name="btn1"> *

<input type=reset value="CANCEL" name="btn1">

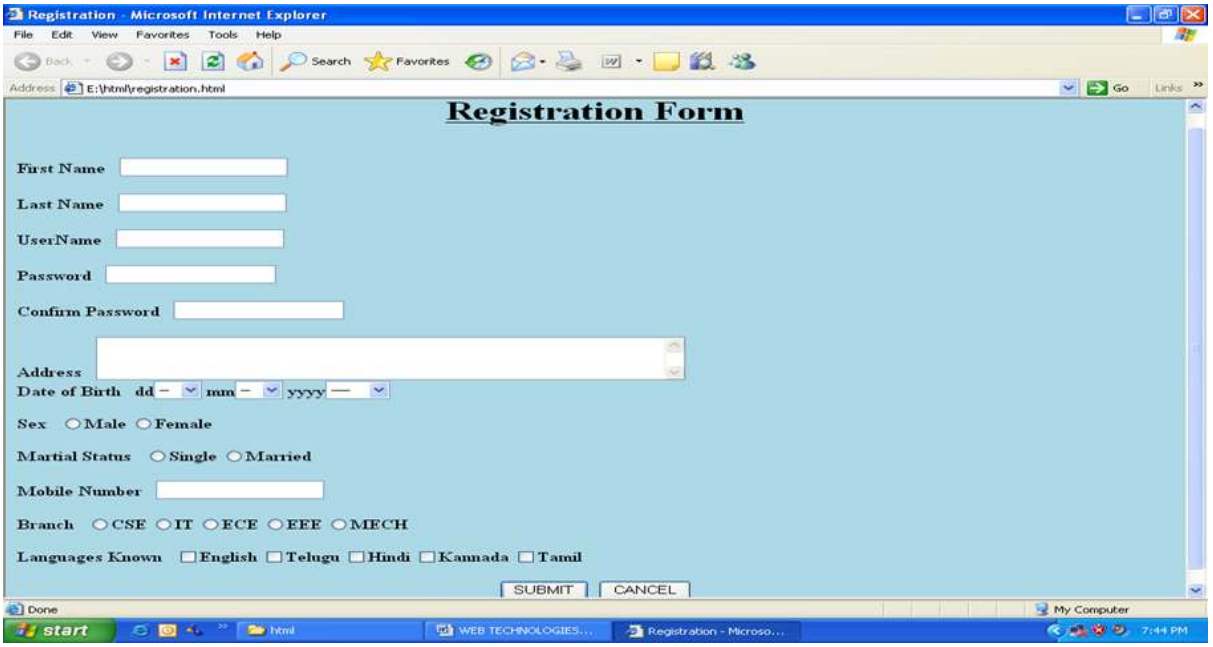
* *

**

</body>

</html>

Output:



Experiment No. 07:

Experiment Name / Title: Write a HTML program to develop a static Login Page.

Objective: Basic HTML form creation for different websites

CODE:

<html>

<head>

<title>login</title>

</head>

<body>

*

*

<h1 align=center><u>LOGIN</u></h1>

*

*

<h4>

* *

*username <input type=text>
 password <input type=password>

*

</h4>

*<input type=submit value=submit>   *

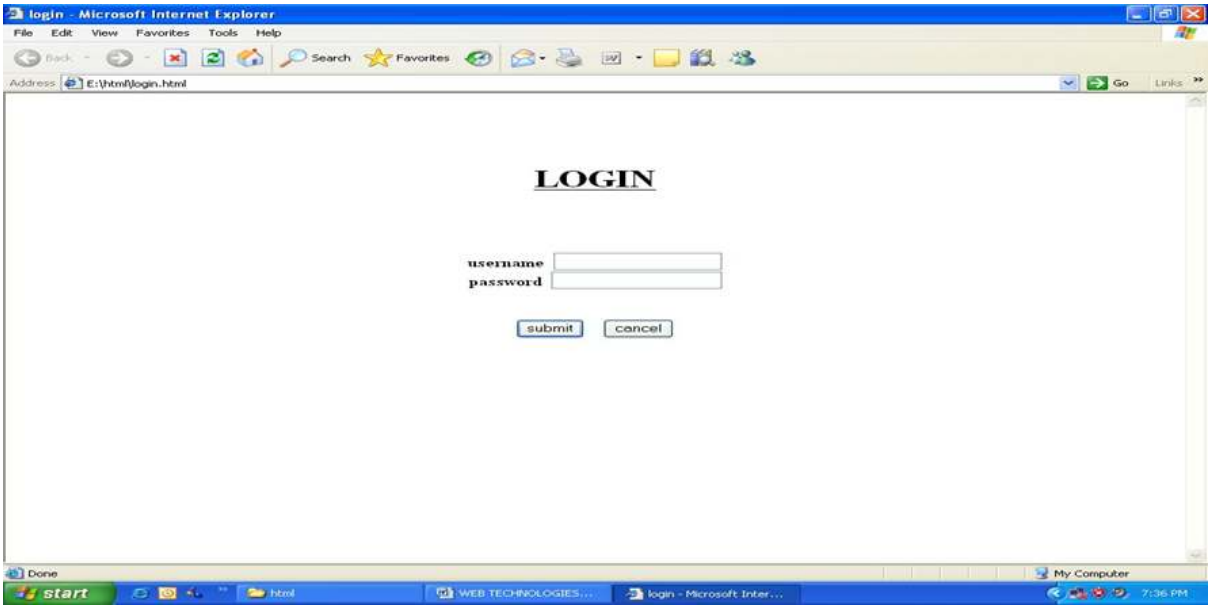
<input type=reset value=cancel>

* *

</body>

</html>

Output:



Experiment No. 08:

Experiment Name / Title: Write a HTML program to develop a static Web Page for Shopping Cart

Objective: Creating different types of component for a website

Code:

```
<html>
<head>
<title>Cart</title>
</head>
<body>
<center>
<table border=0 width=50 height=10 cellpadding=10 cellspacing=10>
<tr>
<th>Book name</th>
<th>Price</th>
<th>Quantity</th>
<th>Amount</th>
</tr>
<tr>
<td>JAVA 2</td>
<td>$35.5</td>
<td>2</td>
<td>$70</td>
</tr>
<tr>
```

<td>XML Bible</td>

<td>$40.5</td>

<td>1</td>

<td>$40.5</td>

</tr>

</table>

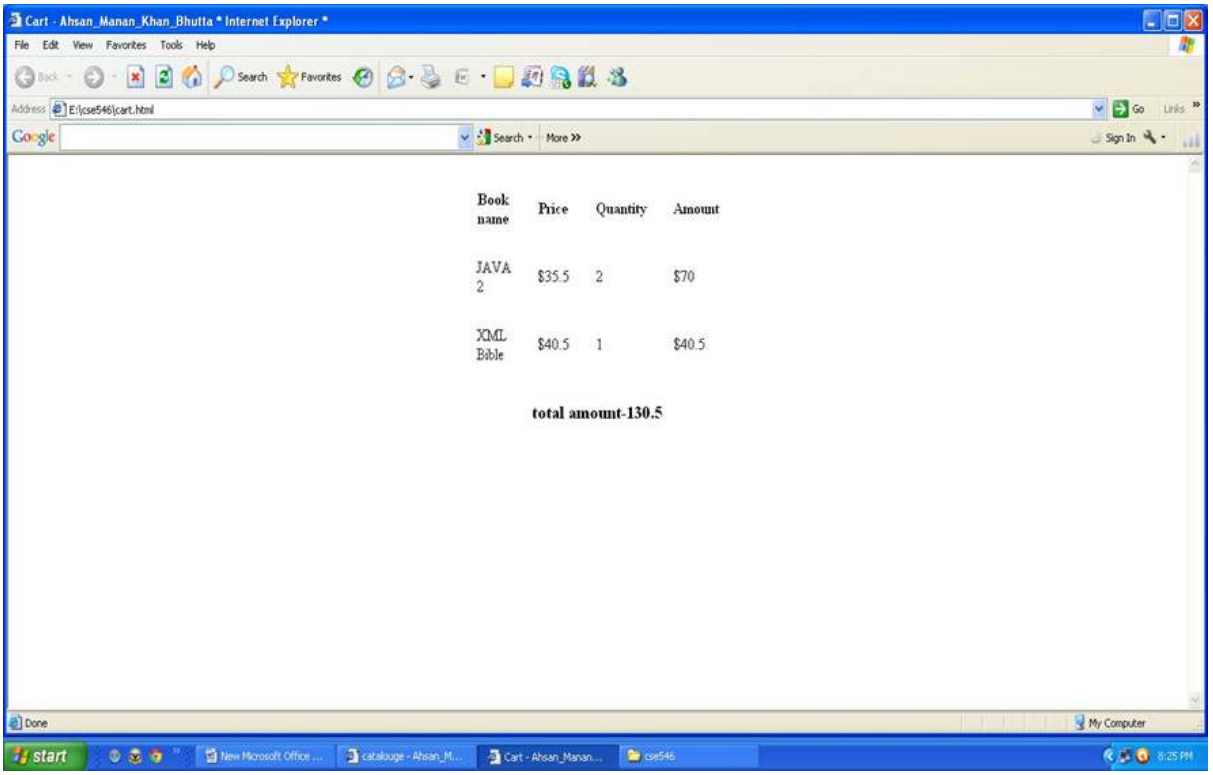
<h3>total amount-130.5</h3>

* *

</body>

</html>

Output:



Experiment No. 09:

Experiment Name / Title: 9. Write HTML for demonstration of cascading style sheets.

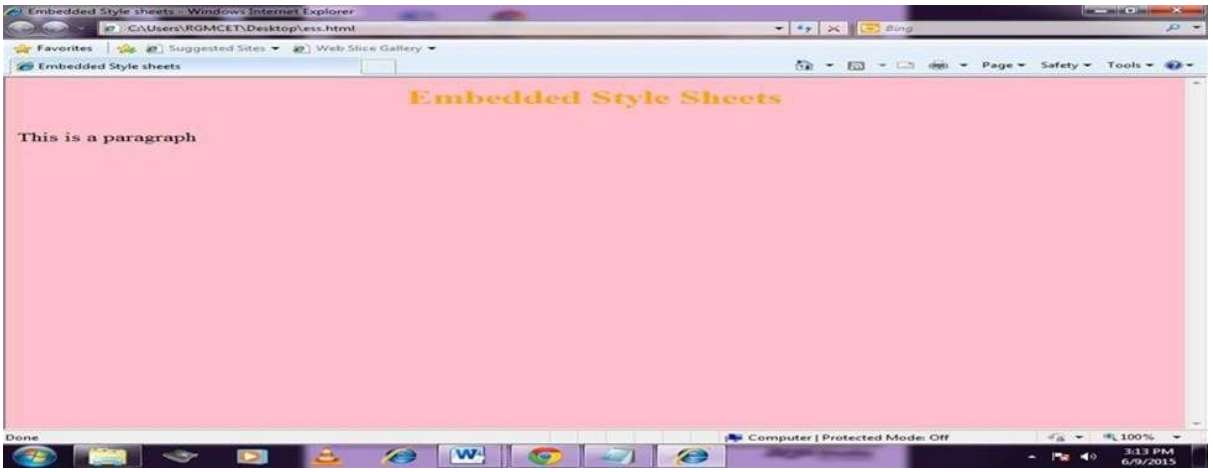
Objective: Understanding Cascading Style Sheet

- **Embedded stylesheets.**
- **External stylesheets.**
- **Inline styles.**

Embedded stylesheets:

```
<html>
<head>
<title>Embedded Style sheets</title>
<style type=”text/css”>
body{background-color: pink;}

h1{color:orange; text-align: center;
}
 </style>
</head>
<body>

p {
font-family: "Times New Roman";
font-size: 20px;
}
 </html>
```

Output:



External Stylesheets:

extern.css:

body {background-color: #d0e4fe;} h1 {

color: orange; text-align: center;

}

p {

font-family: "Times New Roman"; font-size: 20px;

}

extern.html:

<html>

<head>

<title>External Style Sheets</title>

<link rel=”stylesheet” type=”text/css” href=”extern.css”>

</head>

<body>

*<h1>External Style Sheets</h1>
*

<p>This is a paragraph

</body>

</html>

Output:

Inline styles:

<html>

<head>

<title>HTML Tables</table>

</head>

<body bgcolor="pink">

* *

*<h1>Creating HTML Tables</h1>
*

<table border=”2” cellpadding=”4” cellspacing=”4”>

<tr>

<th colspan=”2” style=”background-color:red”> WebSites</th>

```
</tr>

<tr>

<th style=”background- color:blue”>MailSites</th>

<th style=”background- color:green”>JobSites</th>

 </tr>

<tr>

</tr>

<tr>

</tr>

</table>

</center>

</body>

</html>
```

Output:

Experiment No. 10:

Experiment Name / Title: Responsive Web Design and Bootstrap.

OBJECTIVES:

i. **Responsive Web design**

ii. **Bootstrap**

Responsive Web design: Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge a website, to make it look good on all devices (desktops, tablets, and phones). A responsive web design will automatically adjust for different screen sizes and viewports.

To create a responsive website, add the following <meta> tag to all your web pages:

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

Responsive Web Design – Frameworks – Bootstrap

Bootstrap is a CSS framework which uses HTML, CSS and jQuery to make responsive web pages. Bootstrap is an open source toolkit for developing with HTML, CSS, and JS and it's a world's most popular front-end component library. Bootstrap has a light but powerful plugin built on jQuery.

Bootstrap Starter template

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-tofit=no">

<link rel="stylesheet" href="css/bootstrap.min.css" />

<link rel="stylesheet" href="css/main.css" />

<title>Bootstrap</title>

</head>

<body>

<h1>Welcome</h1>

<script src="js/jquery.min.js"></script>

<script src="js/popper.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

Bootstrap Elements

Containers: Containers are the most basic layout element in Bootstrap and are required when using our default grid system.

* *

<!-- Content here -->

* *

Grid System: Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content.

* *

* *

* One of three columns *

* One of three columns *

* One of three columns *

* *

* *

Bootstrap Component

Alert:

* A simple alert *

alert-secondary | alert-success | alert-danger | alert-warning | alert-info| alert-light| alert-dark

* *

Holy guacamole!

You should check in on some of those fields below.

<button type="button" class="close" data-dismiss="alert" aria-label="Close">

×

</button>

* *

Button :

<button type="button" class="btn btn-outline-primary">Primary</button>

btn-lg | btn-sm | btn-block

Collapse

* *

**

Link with href

**

* *

* *

* [your_content] *

* *

EXERCISES: Building a responsive Web Page – Design a responsive web page using HTML, CSS & Bootstrap**.**

Experiment No. 11:

Experiment Name / Title:Write a javascript program to validate the USER LOGIN page.

Objective:

- Learn basic javascript and different approaches to use javascript in web development project setup.
- Apply javascript to validate user login page.

Code:

<html>

<head>

<title>Login Validation</title>

<script language="javascript"> function formValidator()

{

var username=document.getElementById('uname'); var password=document.getElementById('pwd'); if(isEmpty(username)&&isEmpty(password))

{

alert("enter something"); document.form1.uname.focus();

}

if(!isEmpty(username)&&isEmpty(password)&&isAlphabet(username))

```
{
alert("Please enter password"); document.form1.pwd.focus();
}
if(!isEmpty(username)&&!isEmpty(password)&&isAlphabet(username))
{
return true;
}
else
{
if(!isEmpty(username)&&!isEmpty(password)&&!isAlphabet(username))
{
alert("Please Enter only alphabets for username"); document.form1.uname.focus();
}
}
return false;
}
function isEmpty(elem)
{
if(elem.value.length==0)
{
return true;
}
return false;
}
function isAlphabet(elem)
```

```
{
var alphaExp=/^[a-z A-Z]+$/; if(elem.value.match(alphaExp))
{
return true;
}
}
</script>
</head>
<body bgColor=megastar>
<h1 align=center>USER LOGIN VALIDATION</h1>
<br><br>
<form name="form1" onSubmit="return formValidator()">
<center>
<table border=0 colsSpacing=4>
<tr>
<td>Username:</td>
<td><input type=text value="" name="uname"></td>
</tr>
<tr>
<td>Password:</td>
<td><input type=password value="" name="pwd"></td>
</tr>
<tr>
<td><input type=submit value="SUBMIT" name="btn1"></td>
<td><input type=reset value="CANCEL" name="btn2"></td>
```

</tr>

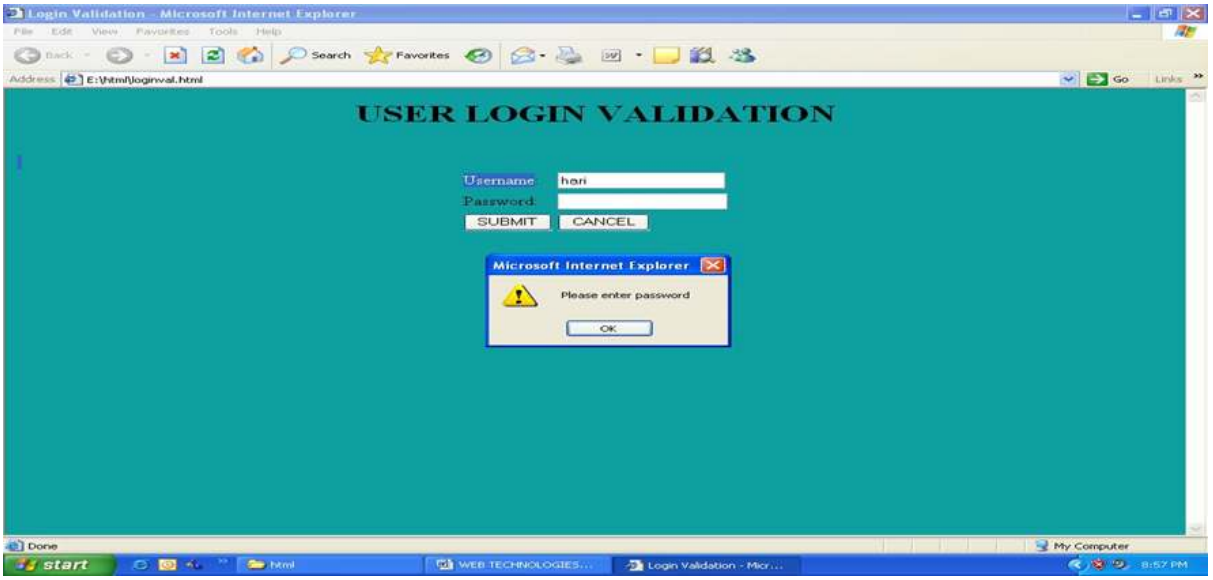
</table>

</form>

</body>

</html>

Output:



Experiment No. 12:

Experiment Name / Title: Write a javascript program for validating REGISTRATION FORM.

Objective:

- Apply javascript to validate Registration forms
- Different rules for handling different types of data in a HTML form.

Code:

```
<html>

<head>

<title>JavaScript sample registration from validation </title>

<script type='text/javascript'> function formValidation()

{

var uid = document.form1.userid; var passid = document.form1.passid;

var uname = document.form1.username; var uadd = document.form1.address;

var uzip = document.form1.zip;

var uemail = document.form1.email; var umsex = document.form1.msex; var ufsex =
document.form1.fsex; if(userid_validation(uid,5,12))

{

if(userid_validation(passid,7,12))

{

if(allLetter(uname))

{

if(alphanumeric(uadd))

{

if(allnumeric(uzip))

{
```

```
if(ValidateEmail(uemail))
{
if(validsex(umsex,ufsex))
{
}
}
}
}
}
}
}
return false;
} function userid_validation(uid,mx,my)
{
var uid_len = uid.value.length;
if (uid_len == 0 || uid_len >= my || uid_len < mx)
{
alert("It should not be empty / length be between "+mx+" to "+my); uid.focus();
return false;
}
return true;
}
function allLetter(uname)
{
var letters = /^[A-Za-z]+$/; if(uname.value.match(letters))
```

```
{
return true;
}
else
{
alert('Please input alphabet characters only'); uname.focus();
return false;
}
}
function alphanumeric(uadd)
{
var letters = /^[0-9a-zA-Z]+$/; if(uadd.value.match(letters))
{
return true;
}
else
{
alert('Please input alphanumeric characters only'); uadd.focus();
return false;
}
}
function allnumeric(uzip)
{
var numbers = /^[0-9]+$/; if(uzip.value.match(numbers))
{
```

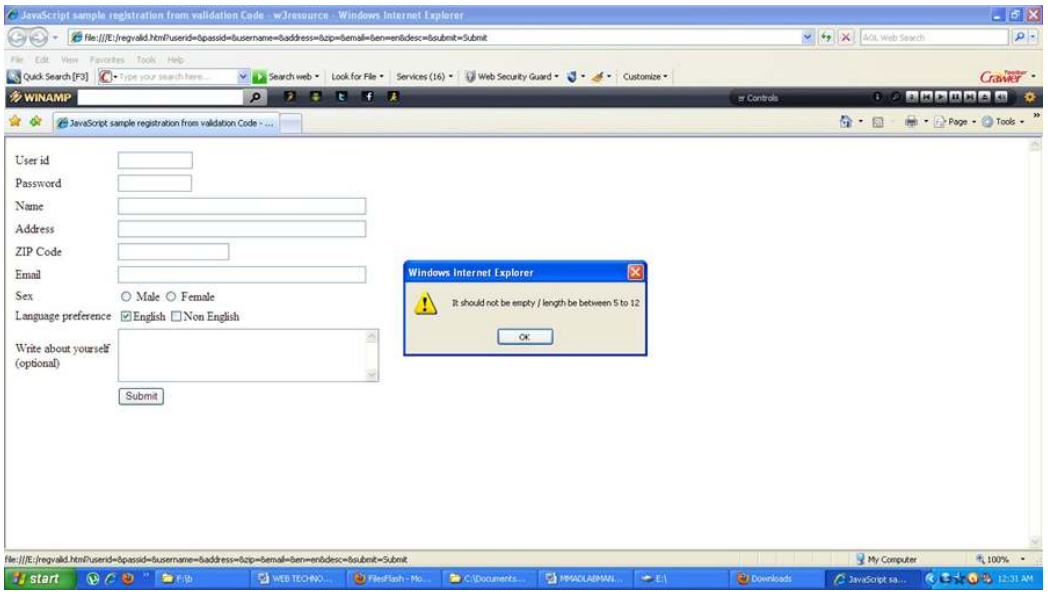
```
return true;

}

else

{

alert('Please input numeric characters only'); uzip.focus();

return true;

}

}


function ValidateEmail(uemail)

{

var mailformat = /^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
if(uemail.value.match(mailformat))

{

return true;

}

else

{

alert("You have entered an invalid email address!"); uemail.focus();

return false;

}

} function validsex(umsex,ufsex)

{ x=0;


if(umsex.checked)
```

```
{ x++;
} if(ufsex.checked)
{ x++;
}
if(x==0)
{
alert('Select Male/Female'); umsex.focus();
return false;
}
else
{
return true;
}
}
</script>
</head>
<body>
<form name='form1' onsubmit='return formValidation()' >
<table width="500" cellpadding="3" style="border-collapse: collapse;">
<tr>
<td>User id </td>
<td><input type="text" name="userid" size="12" /></td>
</tr>
<tr>
<td>Password</td>
```

```
<td><input type="password" name="passid" size="12" /></td>
</tr>
<tr>
<td>Name</td>
<td><input type="text" name="username" size="50" /></td>
</tr>
<tr>
<td>Address</td>
<td><input type="text" name="address" size="50" /></td>
</tr>
<tr>
<td>ZIP Code </td>
<td><input type="text" name="zip" /></td>
</tr>
<tr>
<td>Email</td>
<td><input type="text" name="email" size="50" /></td>
</tr>
<tr>
<td>Sex</td>
<td><input type="radio" name="msex" value="Male" /> Male
<input type="radio" name="fsex" value="Female" /> Female</td>
</tr>
<tr>
<td>Language preference</td>
```

```
<td><input type="checkbox" name="en" value="en" checked />English
<input type="checkbox" name="nonen" value="noen" />Non English</td>
</tr>
<tr>
<td>Write about yourself<br> (optional)</td>
<td><textarea name="desc" rows="4" cols="40"></textarea></td>
</tr>
<tr>
<td> </td>
<td><input type="submit" name="submit" value="Submit" /></td>
<td> </td>
</tr>
</table>
</form>
</body>
</html>
```

Output:



Experiment No. 13: Basic JavaScript DOM Manipulation

Experiment Name / Title: Write a javascript program for validating REGISTRATION FORM.

Objective:

- To understand how to select an HTML element and modify its text content using JavaScript.
- To learn how to dynamically add and remove elements from the DOM using JavaScript.
- To understand how to change the style of HTML elements using JavaScript.

Changing Text Content

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>DOM Manipulation - Changing Text Content</title>

</head>

<body>

    <h1 id="mainHeading">Hello, World!</h1>

    <button onclick="changeText()">Change Text</button>


    <script>

        function changeText() {
```

```
        const heading = document.getElementById('mainHeading');
        heading.textContent = 'Text Changed!';
    }
  </script>
</body>
</html>
```

Output

When you open this HTML file in a browser, you'll see:

1. A heading that says "Hello, World!"
2. A button labeled "Change Text"
3. When you click the button, the heading will change to "Text Changed!"

Explanation

This experiment demonstrates how to:

1. Select an HTML element using document.getElementById()
2. Modify the text content of an element using the textContent property
3. Use an onclick event to trigger a JavaScript function

Adding and Removing Elements

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>DOM Manipulation - Adding and Removing Elements</title>
</head>
```

```
<body>
    <h1>Todo List</h1>
    <input type="text" id="todoInput" placeholder="Enter a new todo">
    <button onclick="addTodo()">Add Todo</button>
    <ul id="todoList"></ul>

    <script>
        function addTodo() {
            const todoInput = document.getElementById('todoInput');
            const todoList = document.getElementById('todoList');

            if (todoInput.value !== '') {
                const li = document.createElement('li');
                li.textContent = todoInput.value;

                const deleteButton = document.createElement('button');
                deleteButton.textContent = 'Delete';
                deleteButton.onclick = function() {
                    todoList.removeChild(li);
                };

                li.appendChild(deleteButton);
                todoList.appendChild(li);
                todoInput.value = '';
            }
```

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

Output

When you open this HTML file in a browser, you'll see:

1. A heading "Todo List"
2. An input field to enter new todos
3. An "Add Todo" button
4. When you enter text and click the button, a new list item appears with the entered text and a "Delete" button
5. Clicking the "Delete" button removes the corresponding list item

Modifying Element Styles

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>DOM Manipulation - Modifying Element Styles</title>
  <style>
    .box {
      width: 100px;
      height: 100px;
      background-color: blue;
      margin: 10px;
    }
```

```
    </style>
</head>
<body>
    <h1>Style Changer</h1>
    <div id="colorBox" class="box"></div>
    <button onclick="changeColor()">Change Color</button>
    <button onclick="changeSize()">Change Size</button>

    <script>
        function changeColor() {
            const box = document.getElementById('colorBox');
            const randomColor = Math.floor(Math.random()*16777215).toString(16);
            box.style.backgroundColor = "#" + randomColor;
        }

        function changeSize() {
            const box = document.getElementById('colorBox');
            const newSize = Math.floor(Math.random() * (200 - 50 + 1)) + 50;
            box.style.width = newSize + 'px';
            box.style.height = newSize + 'px';
        }
    </script>
</body>
</html>
```

Output

When you open this HTML file in a browser, you'll see:

1. A heading "Style Changer"
2. A blue square box
3. Two buttons: "Change Color" and "Change Size"
4. Clicking "Change Color" will change the box to a random color
5. Clicking "Change Size" will change the box to a random size between 50px and 200px

Example of Complex Engineering Problems

1. Interactive Photo Gallery

Scenario

A local photography club wants to showcase their best photos online. They need a simple, interactive website where visitors can view and interact with their photo collection.

Problem Description

The photography club currently lacks an online presence to display their work. They need a solution that allows easy browsing of photos, provides information about each image, and offers a simple way for visitors to interact with the content.

Requirements

1. HTML:

    - Create a responsive layout for the gallery
    - Use semantic HTML5 elements for proper structure
    - Include a header with the gallery name and a brief description
    - Implement a grid or flexbox layout for displaying photos

2. CSS:

    - Style the gallery with a clean, modern design
    - Ensure responsive design for various screen sizes
    - Implement hover effects on photos
    - Create a cohesive color scheme that complements the photos

3. JavaScript:

    - Implement a lightbox feature for viewing full-size images
    - Create a filtering system by photo category (e.g., nature, urban, portrait)
    - Add a "like" button for each photo with a counter
    - Implement a simple search functionality to find photos by title or description

Challenges

1. Optimizing image loading for faster page performance
2. Implementing smooth transitions and animations for a polished user experience
3. Ensuring cross-browser compatibility
4. Managing a potentially large number of images efficiently
5. Creating an intuitive and accessible user interface for diverse visitors

Expected Outcomes

Upon completion of this project, students should deliver:

1. A fully functional, responsive photo gallery website
2. Clean, well-commented HTML, CSS, and JavaScript code
3. A user-friendly interface for browsing, filtering, and interacting with photos
4. Improved skills in front-end web development, including:
5. Responsive design techniques
6. DOM manipulation with JavaScript
7. Event handling and user interaction
8. Basic performance optimization

Evaluation Criteria

The project will be evaluated based on:

1. Code quality and organization
2. Visual design and user experience
3. Functionality and feature implementation
4. Responsiveness and cross-browser compatibility
5. Creativity and problem-solving approach

2. Personal Finance Tracker

Scenario

Many young adults struggle with managing their finances effectively. They need a simple, user-friendly tool to help them track their income and expenses, set budgets, and visualize their spending habits.

Problem Description

Create a web-based personal finance tracker that allows users to log their financial transactions, categorize them, set budget goals, and view reports on their spending patterns. The application should be easy to use and provide valuable insights into the user's financial habits.

Requirements

1. HTML:
    - Create a clean, intuitive layout for the application
    - Implement forms for adding income and expenses
    - Design a dashboard to display financial summaries
    - Use tables or lists to show transaction history
2. CSS:
    - Style the application with a professional, finance-appropriate theme
    - Ensure the design is responsive for mobile and desktop use
    - Use CSS Grid or Flexbox for layout management
    - Implement transitions for a smooth user experience
3. JavaScript:
    - Develop functions to add, edit, and delete transactions
    - Implement data validation for input forms
    - Create a function to calculate and display account balance
    - Generate and display simple charts or graphs for spending analysis
    - Implement local storage to save user data between sessions

Challenges

1. Ensuring accurate calculations for financial data
2. Creating an intuitive user interface for effortless data entry
3. Generating meaningful visual representations of financial data
4. Implementing proper form validation to prevent errors
5. Managing and organizing potentially large sets of transaction data

Expected Outcomes

Upon completion of this project, students should deliver:

1. A functional personal finance tracking web application
2. Well-structured and commented HTML, CSS, and JavaScript code
3. An intuitive interface for entering and managing financial data
4. Basic data visualization of spending patterns
5. Improved skills in:
    - Form handling and data validation
    - Local storage management
    - Basic data analysis and visualization
    - Creating practical, real-world applications

Evaluation Criteria

The project will be evaluated based on:

1. Functionality and accuracy of financial calculations
2. User experience and interface design
3. Code quality, including organization and comments
4. Effective use of local storage for data persistence
5. Implementation of data visualization
6. Responsiveness and cross-browser compatibility