VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering(SCOPE)

Winter Semester 2023-2024

Course Code: BCSE203E

Course Title: WEB PROGRAMMING

LAB MANUAL

[illegible]

[illegible]

Course Coordinators

Dr. MOHANKUMAR. B

Dr. PRIYANKA. N

[illegible]

Head of the Department

Head
Department of Information Security
School of Computer Science & Engineering (SCOPE)
Vellore Institute of Technology (VIT),
(Deemed to be University under section 3 of UGC Act, 1956)
Vellore - 632014, Tamil Nadu, India

Course Outcomes

Graduates of the students will have an ability to:

1. Apply various elements of HTML and CSS.
2. Design interactive web pages using JavaScript.
3. Create Dynamic Web Applications using ReactJS.
4. Deploy and host web applications in Local Servers or Cloud platforms.

Marks weightage for Lab components

BCSE203E - Web Programming

SCHEDULE & RUBRICS

S.No	Assessment Component Name	Modules to be covered	Weightage	Course Outcome (CO) Mapping	Assessment Component Type
1.	Assessment 1	Module 1, 2, 3	10	CO1	Internal Assessment
2.	Assessment 2	Module 4,5	10	CO2	
3.	Assessment 3	Module 6,7	10	CO3	
4.	Mini Project	All	20	CO4	
5.	Quiz	Module 1, 2, 3	10	CO1	
6.	Final Assessment Test	All	40	CO4	Final Assessment
Total -100					

Software's and IDEs

NETBEANS

https://www.apache.org/dyn/closer.cgi/netbeans/netbeans-installers/16/Apache-NetBeans-16-bin-windows-x64.exe

ECLIPSE

https://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/neon /2/eclipse-java-neon-2-win32-x86_64.zip

VISUAL STUDIO CODE

https://visualstudio.microsoft.com/downloads/

EDITORS

NOTEPAD++

https://notepad-plus-plus.org/downloads/

ATOM

https://atom.io/

ONLINE EDITORS

https://www.w3schools.com/tryit/

https://codepen.io/

https://jsfiddle.net/

ASSIGNMENTS

Sl. No	Title	Page No.
1	Create a web page Using Paragraphs, Text, Lists, Images and Tables.	1
2	Create a responsive website using semantics html.	4
3	Apply various types of CSS to design a web page.	7
4	Create an attractive web page by using CSS properties and different styles	12
5	Design an interactive web page using java script functions	19
6	Design a form and validate all the fields of the form elements	22
7	Implement Event Handlers to design a dynamic web page	30
8	Apply Constructor and hooks concept by using React JS	34
9	Design a simple search filter functionality to display a filtered list based on the search query entered by the user using React JS	37
10	Apply props validation in React component	39
11	Implement routing concept to simplify the navigations in web page design	43
12	Additional Exercises	48

[illegible] [illegible]
Prepared by
Dr. Sathya K
Dr. C. R. Dhivyaa

[illegible] [illegible]
Moderators
Dr. Lokesh Kumar R
Dr. Meenakshi S P

[illegible] [illegible]
Course Coordinators
Dr. Mohankumar B
Dr. Priyanka N

[illegible]
HoD

Head
Department of Information Security)
School of Computer Science & Engineering (SCOP
Vellore Institute of Technology (VIT
(Deemed to be University under section 3 of UGC Act, 19
Vellore - 632014, Tamil Nadu, Ind

Task 1

Problem Statement

Create a web page Using Paragraphs, Text, Lists, Images and Tables.

Concept to be Applied

Write a code to display the webpage with following HTML elements

- Paragraph
- Heading
- Bold & Strong Tag
- Label
- Textbox
- Small
- Italic
- Strike
- Images
- Tables

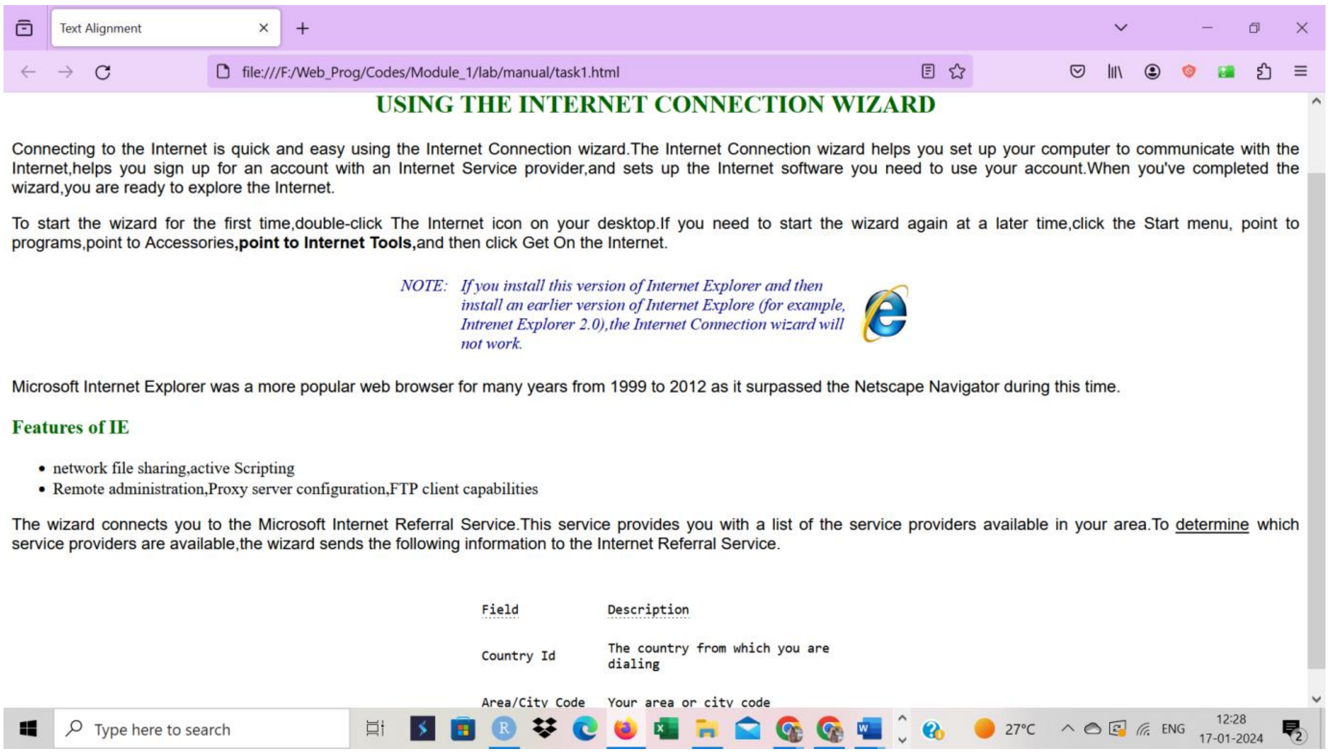
Procedure/Steps

```
<html>
<head>
<title>Text Alignment</title>
</head>
<body>
<br><br>
<h2 align="center" style="color:darkgreen"><b>USING THE INTERNET CONNECTION
WIZARD</b>
</h2>
<p align="justify" style="font-family:arial;font-size:16px">
Connecting to the Internet is quick and easy using the Internet Connection wizard. The Internet
Connection wizard helps you set up your computer to communicate with the Internet,helps
you sign up for an account with an Internet Service provider,and sets up the Internet software
you need to use your account.When you've completed the wizard,you are ready to explore the
Internet.
</p>
<p align="justify" style="font-family:arial;font-size:16px">
To start the wizard for the first time,double-click The Internet icon on your desktop. If you need
to start the wizard again at a later time,click the Start menu, point to programs, point to
```

```
Accessories<strong>,point to Internet Tools,</strong>and then click Get On the Internet.
</p>
<table border="0" align="center" cellpadding="5">
<tr>
        <td valign="top" align="justify">
          <p style="color:darkblue"><i>NOTE:</i></p>
          </td>
              <td align="center">
           <p align="left" style="color:darkblue"><i>If you install this version of Internet Explorer
and then<br>install an earlier version of Internet Explore (for example,<br>Intrenet Explorer
2.0),the Internet Connection wizard will<br> not work.</i></p>
           </td>
           <td>
                <img src="ie.jfif" alt="Internet Explore" width="50" height="60">
            </td>
</tr>
</table>
<p align="justify" style="font-family:arial;font-size:16px">Microsoft Internet Explorer was a
more popular web browser for many years from 1999 to 2012 as it surpassed the Netscape
Navigator during this time.
</p>
<h3 align="left" style="color:darkgreen"> Features of IE </h3>
<ul>
      <li>network file sharing,active Scripting</li>
      <li>Remote administration,Proxy server configuration,FTP client capabilities</li>
</ul>
<p align="justify" style="font-family:arial;font-size:16px">The wizard connects you to the
Microsoft Internet Referral Service. This service provides you with a list of the service provider
available in your area. To <u>determine</u> which service providers are available, the wizar
sends the following information to the Internet Referral Service.
</p>
<table  align="center" border="0" cellpadding="10">
<tr>
<td>
<tt style="border-bottom: 1px dashed">Field</tt>
</td>
<td>
<tt style="border-bottom: 1px dashed">Description</tt>
</td>
</tr>
```

```
<tr>
        <td><tt>Country Id</tt></td>
       <td><tt>The country from which you are<br> dialing</tt></td>
</tr>
<tr>
          <td><tt>Area/City Code</tt></td>&nbsp;  &nbsp;
               <td><tt>Your area or city code</tt></td>
</tr>
 </table>
</body>
</html>
```

Output:



USING THE INTERNET CONNECTION WIZARD

Connecting to the Internet is quick and easy using the Internet Connection wizard.The Internet Connection wizard helps you set up your computer to communicate with the Internet,helps you sign up for an account with an Internet Service provider,and sets up the Internet software you need to use your account.When you've completed the wizard,you are ready to explore the Internet.

To start the wizard for the first time,double-click The Internet icon on your desktop.If you need to start the wizard again at a later time,click the Start menu, point to programs,point to Accessories,**point to Internet Tools,**and then click Get On the Internet.

NOTE: If you install this version of Internet Explorer and then install an earlier version of Internet Explore (for example, Intrenet Explorer 2.0),the Internet Connection wizard will not work.

Microsoft Internet Explorer was a more popular web browser for many years from 1999 to 2012 as it surpassed the Netscape Navigator during this time.

Features of IE

- network file sharing,active Scripting
- Remote administration,Proxy server configuration,FTP client capabilities

The wizard connects you to the Microsoft Internet Referral Service.This service provides you with a list of the service providers available in your area.To determine which service providers are available,the wizard sends the following information to the Internet Referral Service.

Field	Description
Country Id	The country from which you are dialing
Area/City Code	Your area or city code

Task 2

Problem Statement

You volunteer at a local food bank called ABC Food Bank that collects community food donations and provides food and other services to those in need. The company has asked you to create a responsive website using semantics html.

Concept to be Applied

Sematic html tags such as navigations, section, article and aside and non-semantic html tags such as div, span

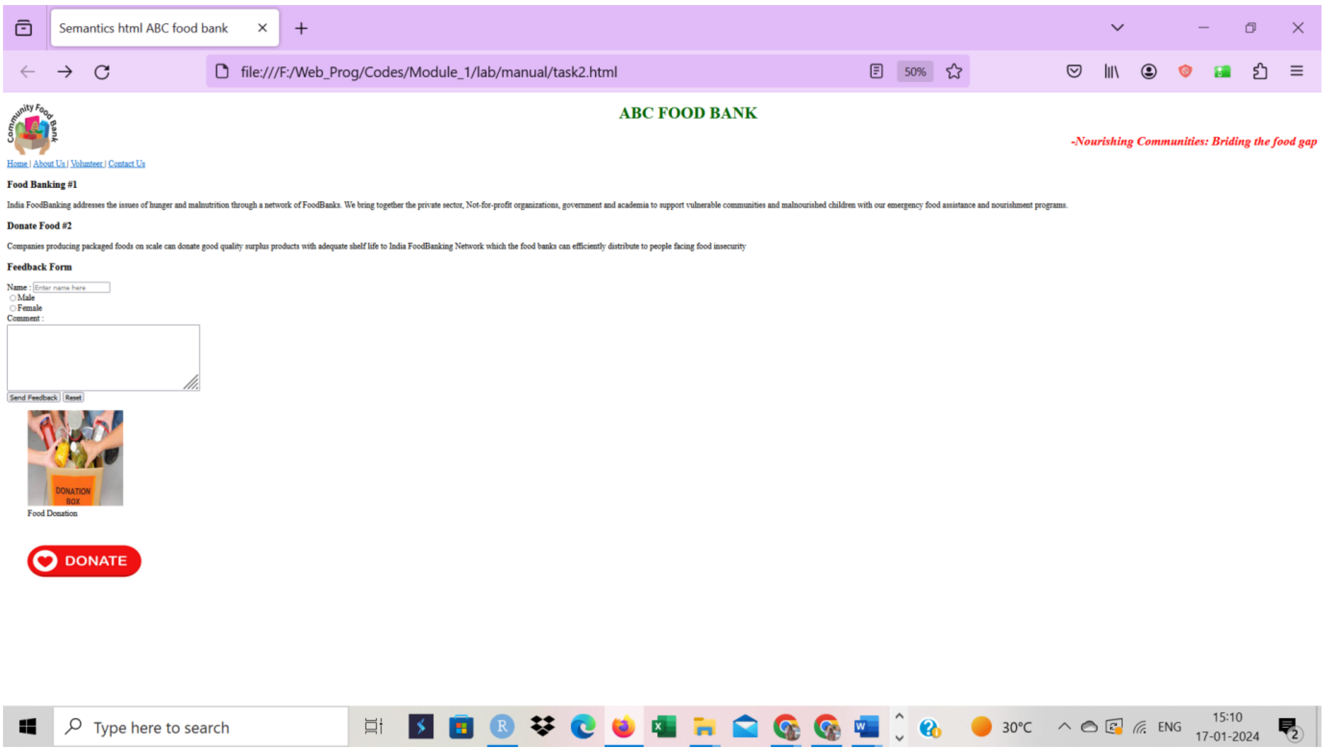
Procedure/Steps

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8" />
   <title>Semantics html ABC food bank</title>
</head>
<body>
<header>
 <hgroup>
      <img src="food bank.png" alt="food bank" width="100" height="100" align="left"/>
         <h1 align="center" style="color:darkgreen"><strong>ABC FOOD
BANK</strong></h1>
         <h2 align="right" style="color:red"><em><i>-Nourishing Communities: Briding the
food gap </em></i></h2>
 </hgroup>
 </header>
<nav>
               <a href="/home/">
                        Home
               </a>   |
               <a href="/about-us/">
                        About Us
               </a>   |
               <a href="/Volunteer/">
                         Volunteer
               </a>   |
               <a href="/Contact us/">
```

```
                    Contact Us
               </a>
</nav>
   <section>
      <article>
         <header>
            <h1>Food Banking #1</h1>
         </header>
         <section>
            India FoodBanking addresses the issues of hunger and malnutrition through a
network of FoodBanks. We bring together the private sector, Not-for-profit organizations,
government and academia to support vulnerable communities and malnourished children
with our emergency food assistance and nourishment programs.
         </section>
      </article>
      <article>
         <header>
            <h1>Donate Food #2</h1>
         </header>
         <section>
            Companies producing packaged foods on scale can donate good quality surplus
products with adequate shelf life to India FoodBanking Network which the food banks can
efficiently distribute to people facing food insecurity
         </section>
      </article>
   </section>
   <aside>
<section>
         <h3> Feedback Form</h3>
         <FORM action="1.html" method="post" enctype="text/plain">
               Name : <INPUT type="text" name="firstname" placeholder="Enter name here">
         <BR>
          <INPUT type="radio" name="gender" value="male">Male<BR>
          <INPUT type="radio" name="gender" value="female">Female<BR>
           Comment :<BR>
      <TEXTAREA rows="6" cols="50" name="commentfield"></TEXTAREA>
      <BR>
      <INPUT TYPE="submit" value="Send Feedback">
      <INPUT TYPE="reset" value="Reset">
   </FORM>
```

```
</section>
 <figure align="center">
         <img width="185" height="185"
            src="donatefood.jfif"
            alt="foobar"/>
         <figcaption align="center"><span>Food Donation</span></figcaption>
</figure>
</aside>
<footer>
    <div>
      <p><a href="https://www.w3schools.com">
      <img src="donate.jpg" alt="donate" width="300" height="132">
      </a></p>
   </div>
</footer>
</body></html>
```

<u>Output</u>



Semantics html ABC food bank

file:///F:/Web_Prog/Codes/Module_1/lab/manual/task2.html

ABC FOOD BANK

-Nourishing Communities: Briding the food gap

Home | About Us | Volunteer | Contact Us

Food Banking #1

India FoodBanking addresses the issues of hunger and malnutrition through a network of FoodBanks. We bring together the private sector, Not-for-profit organizations, government and academia to support vulnerable communities and malnourished children with our emergency food assistance and nourishment programs.

Donate Food #2

Companies producing packaged foods on scale can donate good quality surplus products with adequate shelf life to India FoodBanking Network which the food banks can efficiently distribute to people facing food insecurity

Feedback Form

Name : Enter name here
○ Male
○ Female
Comment :

Send Feedback | Reset

Food Donation

Task 3

Problem Statement

Include CSS style for the above food bank problem as mentioned in task 2 and apply with different types of CSS

Concept to be Applied

Write a script to display the webpage with following HTML elements

- Internal CSS, Internal CSS and External CSS
- Div tag
- Style

Procedure/Steps

```
<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8" />
   <title>Semantics html ABC food bank</title>
<style>
aside {
  width: 75%;
  padding-left: 15px;
  margin-left: 30px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
main {
  width: 75%;
  padding-left: 15px;
  margin-left: 30px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
section article {
  float: left;
  margin: 0 1.5%;
  width: 40%;
}
```

```
footer {
 position:fixed;
 bottom:0;
 left:0;
 width:100%;
 height: 50px;
 background-color: #D0DAEE;
}
</style>

</head>
<body>
   <header style="background-color:#33475b">
      <hgroup>
        <img src="food bank.png" alt="food bank" width="100" height="100" align="left"/>
         <h1 align="center" style="color:white"><strong>ABC FOOD BANK</strong></h1>
         <h2 align="right" style="color:red"><em><i>-Nourishing Communities: Briding the
food gap </em></i></h2>
      </hgroup>
   </header>
   <nav>
              <a href="/home/">
                        Home
              </a>   |
              <a href="/about-us/">
                        About Us
              </a>   |
              <a href="/Volunteer/">
                         Volunteer
              </a>   |
              <a href="/Contact us/">
                        Contact Us
              </a>
     </nav>
   <div>
      <article>
         <header style="background-color:#33475b">
            <h1>Food Banking #1</h1>
         </header>
         <p>
```

```
            India FoodBanking addresses the issues of hunger and malnutrition through a
network of FoodBanks. We bring together the private sector, Not-for-profit organizations,
government and academia to support vulnerable communities and malnourished children
with our emergency food assistance and nourishment programs.
          </p>
       </article>
       <article>
          <header style="background-color:#33475b">
             <h1>Donate Food #2</h1>
          </header>
          <p>
             Companies producing packaged foods on scale can donate good quality surplus
products with adequate shelf life to India FoodBanking Network which the food banks can
efficiently distribute to people facing food insecurity
          </p>
       </article>
  </div>
<div style="display:flex;align-items:center;">
<main>
  <h3>Item we required:<br></h3>
  <h5> Our pantry is in need of the following item:
  <ul>
    <li>Rice</li>
    <li>Vegetables</li>
    <li>Bread</li>
    <li>Fruits</li>
    <li>Millets</li>
   </ul>
 </main>
<aside>
       <section>
          <h3> Feedback Form</h3>
          <FORM action="1.html" method="post" enctype="text/plain">
               Name : <INPUT type="text" name="firstname" placeholder="Enter name here">
           <BR>Comment :<BR>
           <TEXTAREA rows="6" cols="50" name="commentfield"></TEXTAREA>
       <BR>
       <INPUT TYPE="submit" value="Send Feedback">
       <INPUT TYPE="reset" value="Reset">
    </FORM>
```

```
        </section>
        <figure align="center">
            <img width="185" height="185"
                src="donatefood.jfif"
                alt="foobar"/>
            <figcaption align="center"><span>Food Donation</span></figcaption>

         </figure>
    </aside>
</div>
    <footer>
       <p>&copy www.abcfoodbank.com</p>
    </footer>
</body>
</html>
```

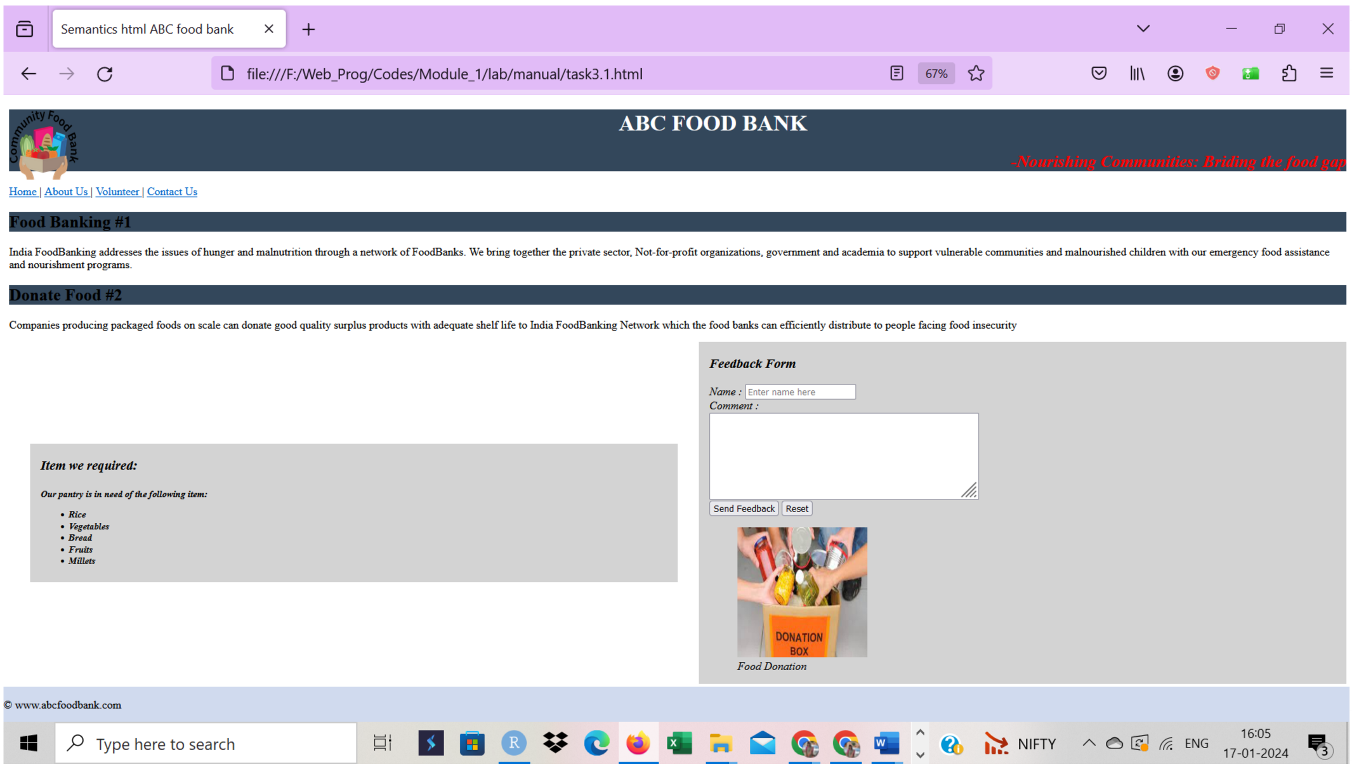
External CSS Syntax

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
```

mystyle.css

```
aside {
  width: 75%;
  padding-left: 15px;
  margin-left: 30px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
main {
  width: 75%;
  padding-left: 15px;
  margin-left: 30px;
  float: right;
  font-style: italic;
  background-color: lightgray;
}
section article {
  float: left;
  margin: 0 1.5%;
  width: 40%;
}
footer {
  position:fixed;
  bottom:0;
  left:0;
  width:100%;
  height: 50px;
  background-color: #D0DAEE;
}
```

<u>Output</u>



Semantics html ABC food bank
file:///F:/Web_Prog/Codes/Module_1/lab/manual/task3.1.html
67%

ABC FOOD BANK

-Nourishing Communities: Briding the food gap

Home | About Us | Volunteer | Contact Us

Food Banking #1

India FoodBanking addresses the issues of hunger and malnutrition through a network of FoodBanks. We bring together the private sector, Not-for-profit organizations, government and academia to support vulnerable communities and malnourished children with our emergency food assistance and nourishment programs.

Donate Food #2

Companies producing packaged foods on scale can donate good quality surplus products with adequate shelf life to India FoodBanking Network which the food banks can efficiently distribute to people facing food insecurity

Item we required:

Our pantry is in need of the following item:

- *Rice*
- *Vegetables*
- *Bread*
- *Fruits*
- *Millets*

Feedback Form

Name : Enter name here
Comment :

Send Feedback | Reset

Food Donation

© www.abcfoodbank.com

Task 4

Problem Statement

Create a web application for ABC Food Bank using HTML and CSS properties, apply witl different styles

Concept to be Applied

Write a script to display the webpage with following HTML elements

- Internal CSS
- CSS Selectors
- Class selectors
- Pseudo class selector
- Div tag
- Style
- Table
- Heading

Procedure/Steps

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>ABC Food Bank</title>
<style>
/* Global styles */ body {
font-family: 'Arial', sans-serif; margin: 0;
padding: 0;
background-color: #f5f5f5; color: #333;
}

/* Navbar styles */
.navbar {
background-color: #333; overflow: hidden;
}

.navbar a {
```

```
float: left; display: block; color: white;
text-align: center; padding: 14px 16px; text-decoration: none;
}

.navbar a:hover { background-color: #ddd; color: black;
}

/* Banner styles */
.banner {
text-align: center;

margin: 20px 0; padding: 40px;
background-color: #007BFF; color: white;
}

/* Account info styles */
.account-info {
max-width: 600px; margin: 20px auto; padding: 20px; background-color: white;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

/* Transactions styles */
.transactions {
max-width: 600px; margin: 20px auto; padding: 20px; background-color: white;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.transactions table { width: 100%;
border-collapse: collapse;
}

.transactions table, .transactions table th, .transactions table td { border: 1px solid #ddd;
padding: 8px; text-align: left;
}

.transactions table th { background-color: #f2f2f2;
}

/* Footer styles */
.footer {
```

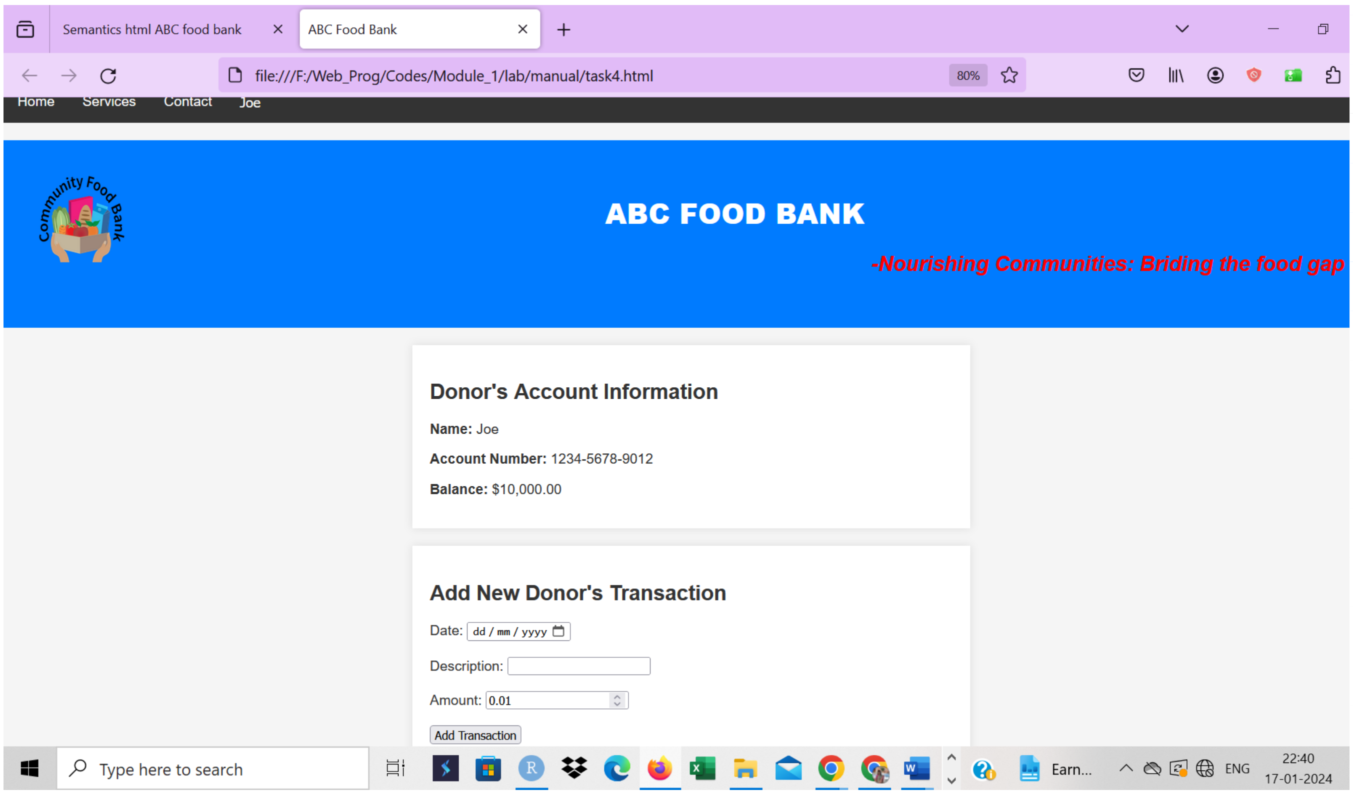
```
padding: 20px; text-align: center;
background-color: grey; color: white;
margin-top: 20px;
}

/* Dropdown Profile */
.dropdown {
float: left; overflow: hidden;
}

.dropdown .dropbtn { font-size: 16px; border: none; outline: none; color: white; padding: 14px
16px;
background-color: inherit; cursor: pointer;
}

.navbar a:hover, .dropdown:hover .dropbtn { background-color: #ddd;
color: black;
}

.dropdown-content { display: none; position: absolute;
background-color: #f9f9f9; min-width: 160px;
box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2); z-index: 1;
}

.dropdown-content a { float: none; color: black; padding: 12px 16px;
text-decoration: none; display: block;
text-align: left;
}

.dropdown-content a:hover { background-color: #ddd;
}

.dropdown:hover .dropdown-content { display: block;
}

/* Form Styles */

.transaction-form { max-width: 600px; margin: 20px auto; padding: 20px;
background-color: white;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
}

/* News Section Styles */
.news {
max-width: 600px; margin: 20px auto; padding: 20px; background-color: white;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

/* Testimonials Styles */
.testimonials {
max-width: 600px; margin: 20px auto; padding: 20px; background-color: white;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

</style>
</head>

<body>
<!-- Navigation Bar -->
<div class="navbar">
<a href="#home">Home</a>
<a href="#services">Services</a>
<a href="#contact">Contact</a>
<!-- Profile Dropdown -->
<div class="dropdown">
<button class="dropbtn">Joe</button>
<div class="dropdown-content">
<a href="#">Profile</a>
<a href="#">Settings</a>
<a href="#">Logout</a>
</div>
</div>
</div>

<!-- Banner Section -->
<div class="banner">
 <img src="food bank.png" alt="food bank" width="100" height="100" align="left"/>
 <h1 align="center" style="color:white"><strong>ABC FOOD BANK</strong></h1>
 <h2 align="right" style="color:red"><em><i>-Nourishing Communities: Briding the food gap
</em></i></h2>
```

```
</div>

<!-- Account Information -->
<div class="account-info">
<h2>Donor's Account Information</h2>
<p><strong>Name:</strong> Joe</p>
<p><strong>Account Number:</strong> 1234-5678-9012</p>
<p><strong>Balance:</strong> $10,000.00</p>
</div>

<!-- Add Transactions Form -->
<div class="transaction-form">
<h2>Add New Donor's Transaction</h2>
<form action="#">
<label for="date">Date:</label>
<input type="date" id="date" name="date"><br><br>
<label for="desc">Description:</label>
<input type="text" id="desc" name="desc"><br><br>
<label for="amount">Amount:</label>
<input type="number" id="amount" name="amount" step="0.01"><br><br>
<input type="submit" value="Add Transaction">
</form>
</div>

<!-- Recent Transactions -->
<div class="transactions">
<h2>Recent Transactions</h2>
<table>
<thead>
<tr>
<th>Date</th>
<th>Description</th>
<th>Amount</th>
</tr></thead>
<tbody><tr>
<td>2023-08-30</td>
<td>ATM Withdrawal</td>
<td>-$200.00</td></tr>
<tr>
```

```
<td>2023-08-29</td>
<td>Deposit</td>
<td>+$1,000.00</td></tr>
<tr>
<td>2023-08-28</td>
<td>Online Purchase</td>
<td>-$50.00</td></tr>
</tbody>
</table>
</div>

<!-- News & Updates -->
<div class="news">
<h2>Mission</h2>
<article>
<h3>Our Vision</h3>
<p>The vision of IFBN is to end hunger, achieve food security and improve nutrition.</p>
</article>
<article>
<h3>Our Values</h3>
<p>We stand for Collaboration, Integrity, Volunteerism & Food Safety</p>
</article>
</div>
<!-- Footer -->
<div class="footer">
<p>&copy; ABC Food Bank Management. All rights reserved.</p>
</div></body></html>
```

Output



Semantics html ABC food bank
ABC Food Bank
file:///F:/Web_Prog/Codes/Module_1/lab/manual/task4.html
80%
Home
Services
Contact
Joe
Type here to search
Earn...
ENG
22:40
17-01-2024

ABC FOOD BANK

-Nourishing Communities: Briding the food gap

Donor's Account Information

Name: Joe

Account Number: 1234-5678-9012

Balance: $10,000.00

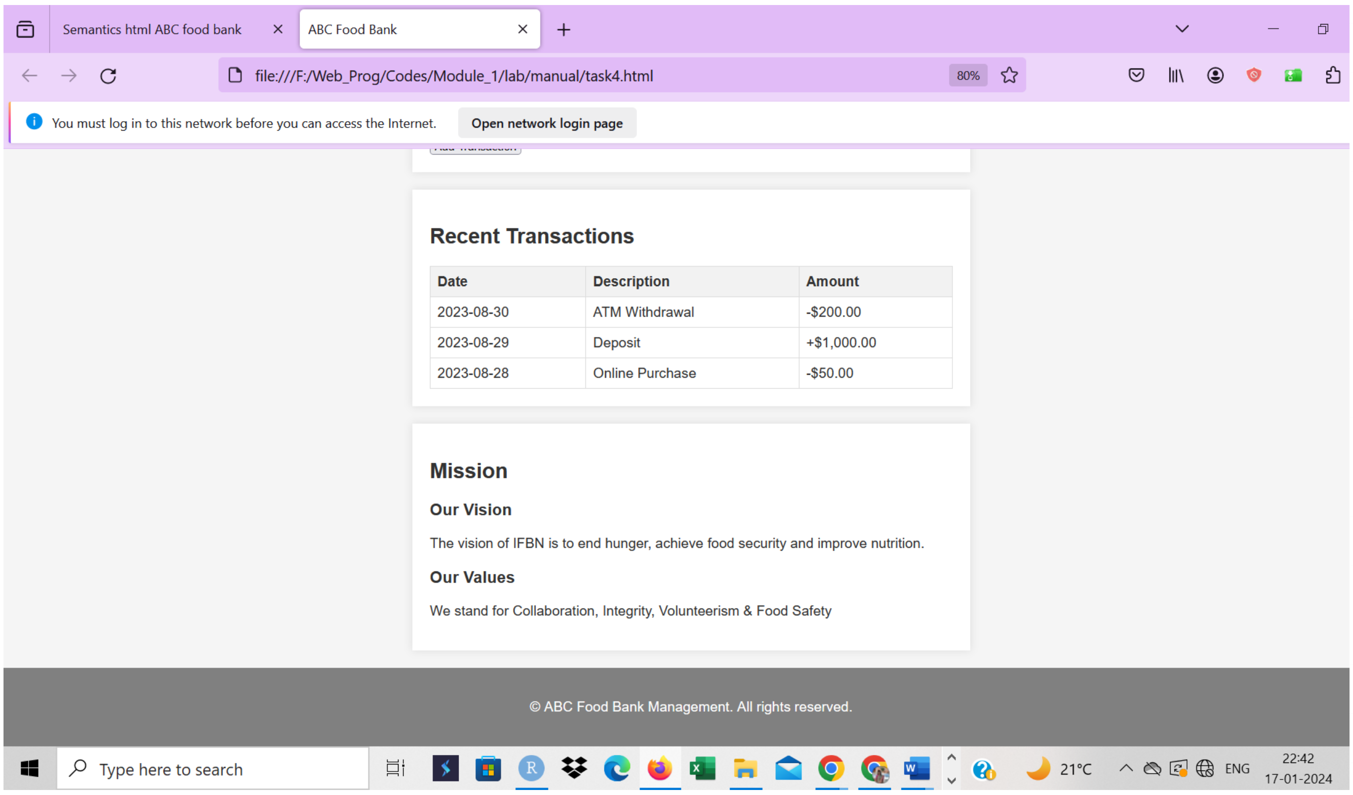
Add New Donor's Transaction

Date: dd / mm / yyyy

Description:

Amount: 0.01

Add Transaction



Semantics html ABC food bank
ABC Food Bank
file:///F:/Web_Prog/Codes/Module_1/lab/manual/task4.html
80%
You must log in to this network before you can access the Internet.
Open network login page
Type here to search
21°C
ENG
22:42
17-01-2024

Recent Transactions

Date	Description	Amount
2023-08-30	ATM Withdrawal	-$200.00
2023-08-29	Deposit	+$1,000.00
2023-08-28	Online Purchase	-$50.00

Mission

Our Vision

The vision of IFBN is to end hunger, achieve food security and improve nutrition.

Our Values

We stand for Collaboration, Integrity, Volunteerism & Food Safety

© ABC Food Bank Management. All rights reserved.

Task 5

Problem Statement

Job 1:

Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10 and outputs html text that displays the resulting values in an html table format.

Job2:

Write a script with a function solve Quadratic() taking three parameters a, b and c. Allow the function to find the real roots of the quadratic equation. Use appropriate forms to capture the values from the user. Use Math object to find the square root of a number.

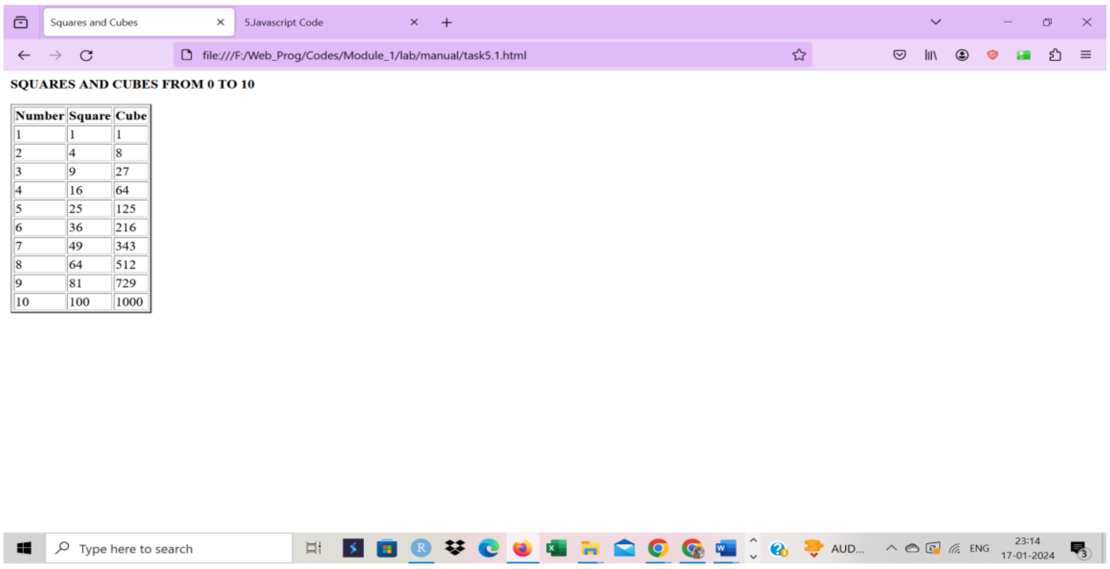
Concept to be Applied

Basics of JavaScript- alert, prompt , for loop, if- else, function, on-submit

Job1

```
<html>
<head><title> Squares and Cubes </title></head>
<script>
   document.write('<p><b>SQUARES AND CUBES FROM 0 TO 10</b></p>');
   document.write('<table border="2" cellspacing="2">');
   document.write('<th> Number </th> <th> Square </th> <th> Cube </th>');
   for(var i=1;i<=10;i++)
      document.write("<tr><td>"+ i +"</td><td>"+ i*i + "</td><td>"+ i*i*i +"</td></tr>");
   document.write("</table>");
</script>
</html>
```

Output



SQUARES AND CUBES FROM 0 TO 10

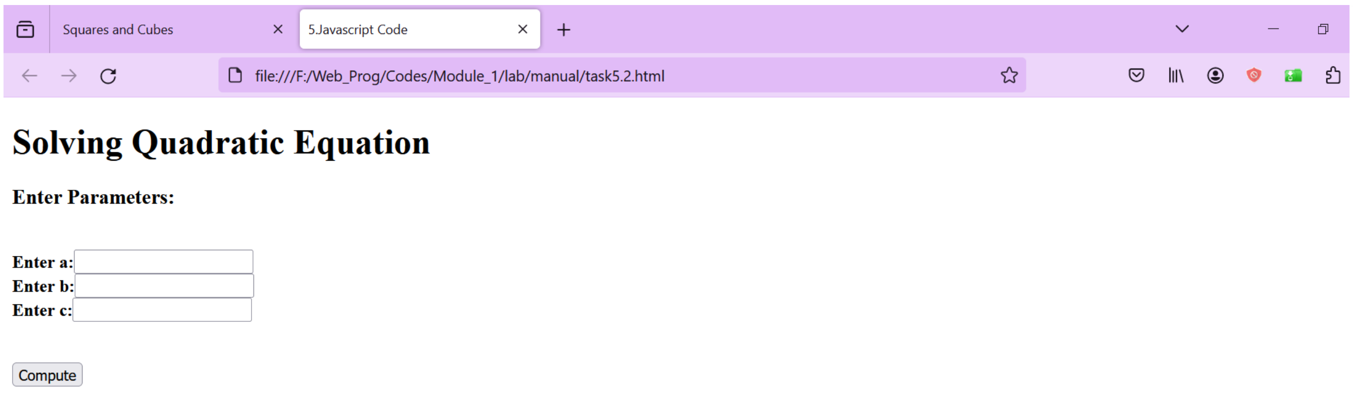
Number	Square	Cube
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

Job 2

```
<html xmlns="http://www.w3.org/1999/xhtml">
        <head>
        <title>5.Javascript Code</title>
        <script type="text/javascript">
        function solveQuadratic()
         {
           a=document.forms.form1.param1.value;
           b=document.forms.form1.param2.value;
           c=document.forms.form1.param3.value;
          var det=b*b-4*a*c;
         if(det<0)
         alert("Sorry!! Computing Roots is impossible");
         else
            {
              var x1=(-b+Math.sqrt(det))/(2*a);
              var x2=(-b-Math.sqrt(det))/(2*a);
             alert("The roots of are "+x1+" and "+x2); }
            }
         </script>
        </head>
       <body>
       <form name="form1" onSubmit="return solveQuadratic();" "method="get">
       <h1>Solving Quadratic Equation</h1>
       <h3> Enter Parameters:</h3><br>
```

```
    <b>Enter a:</b><input type="text" name="param1" size="20"><br>
    <b>Enter b:</b><input type="text" name="param2" size="20"><br>
    <b>Enter c:</b><input type="text" name="param3" size="20"><br><br><br>
   <input type="submit" value="Compute"><br></form> </body>    </html>
```

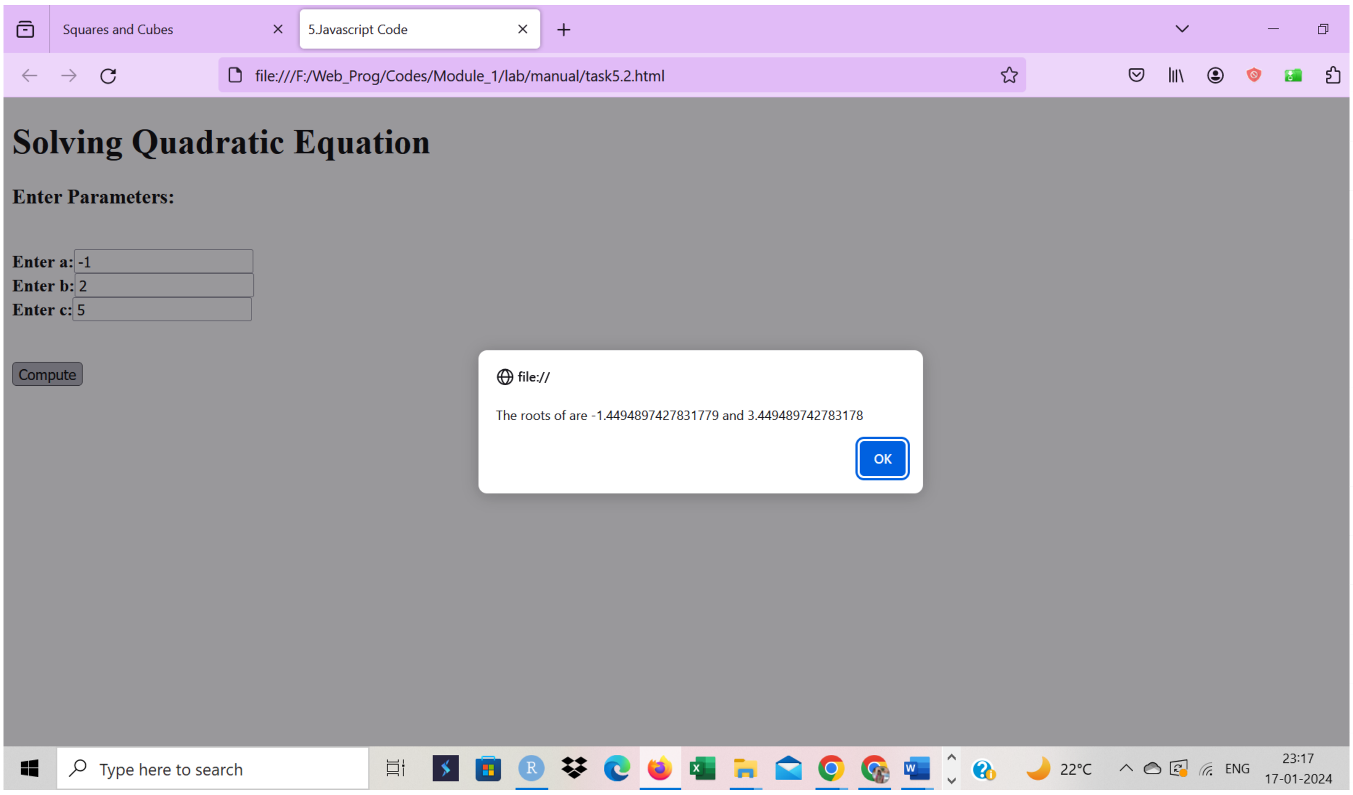
Output



Solving Quadratic Equation

Enter Parameters:

Enter a:
Enter b:
Enter c:



Solving Quadratic Equation

Enter Parameters:

Enter a: -1
Enter b: 2
Enter c: 5

file://

The roots of are -1.4494897427831779 and 3.449489742783178

Task 6

Problem:

Design a volunteer registration form for ABC food bank website using HTML, CSS and apply validation.

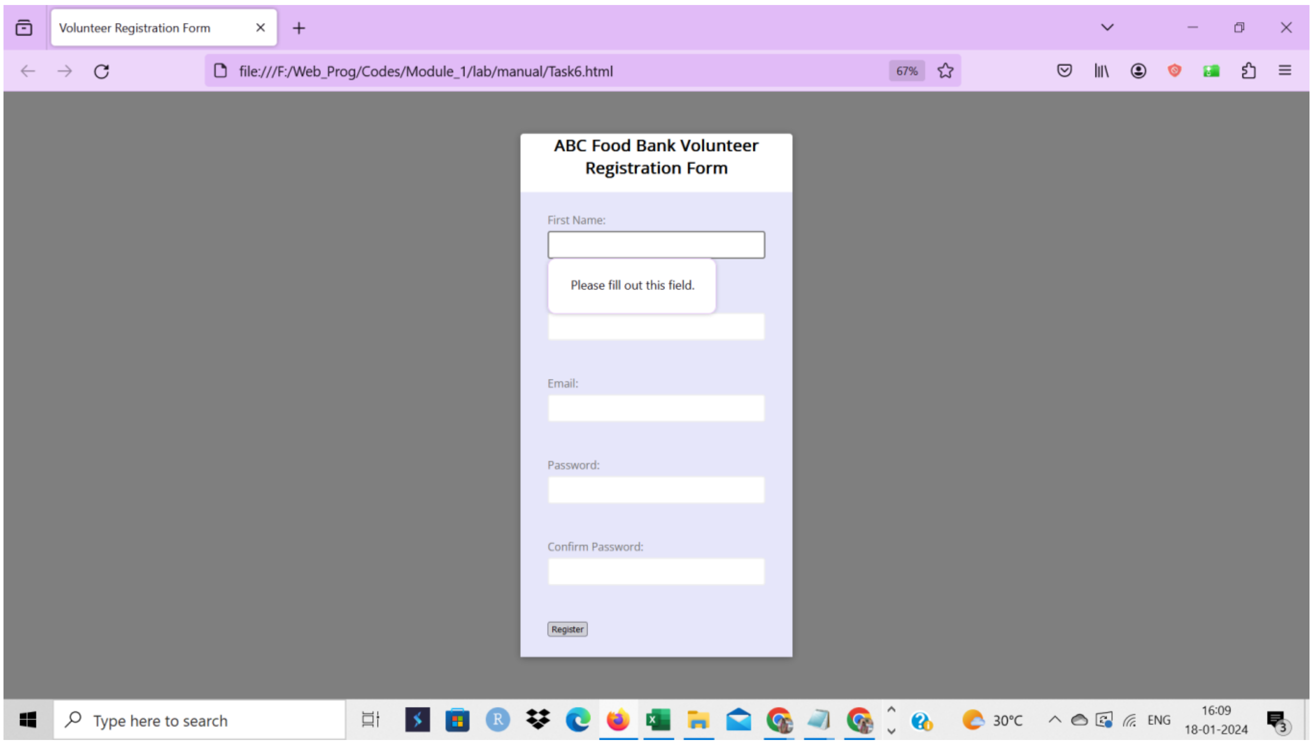
Procedures:

Job1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Volunteer Registration Form</title>
  <link rel="stylesheet" href="style.css" />
  <script>
    function validateForm() {
      var firstName = document.getElementById('firstName').value;
      var lastName = document.getElementById('lastName').value;
      var email = document.getElementById('email').value;
      var password = document.getElementById('password').value;
      var confirmPassword = document.getElementById('confirmPassword').value;
      // Simple validation for demonstration purposes
      if (firstName === '' || lastName === '' || email === '' || password === '' ||
confirmPassword === '') {
        alert('All fields must be filled out');
        return false;
      }
      if (password !== confirmPassword) {
        alert('Passwords do not match');
        return false;
      }
      // Additional validation logic can be added based on requirements
      return true; // Form is valid
    }
  </script>
</head>
<body>
<div class="container">
  <h2>ABC Food Bank Volunteer Registration Form</h2>
```

```
    <form id="form" onsubmit="return validateForm()" class="form" style="background
color:#E6E6FA">
   <div class="form-control">
    <label for="firstName">First Name:</label>
    <input type="text" id="firstName" name="firstName" required><br>
  </div>
    <div class="form-control">
    <label for="lastName">Last Name:</label>
    <input type="text" id="lastName" name="lastName" required><br>
  </div>
   <div class="form-control">
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br>
   </div>
   <div class="form-control">
    <label for="password">Password:</label>
    <input type="password" id="password" name="password" required><br>
   </div>
   <div class="form-control">
    <label for="confirmPassword">Confirm Password:</label>
    <input type="password" id="confirmPassword" name="confirmPassword" required><br>
   </div>
    <input type="submit" value="Register">
  </form></div></body></html>
```

Output


Volunteer Registration Form
file:///F:/Web_Prog/Codes/Module_1/lab/manual/Task6.html
67%

ABC Food Bank Volunteer Registration Form

First Name:

Please fill out this field.

Email:

Password:

Confirm Password:

Register

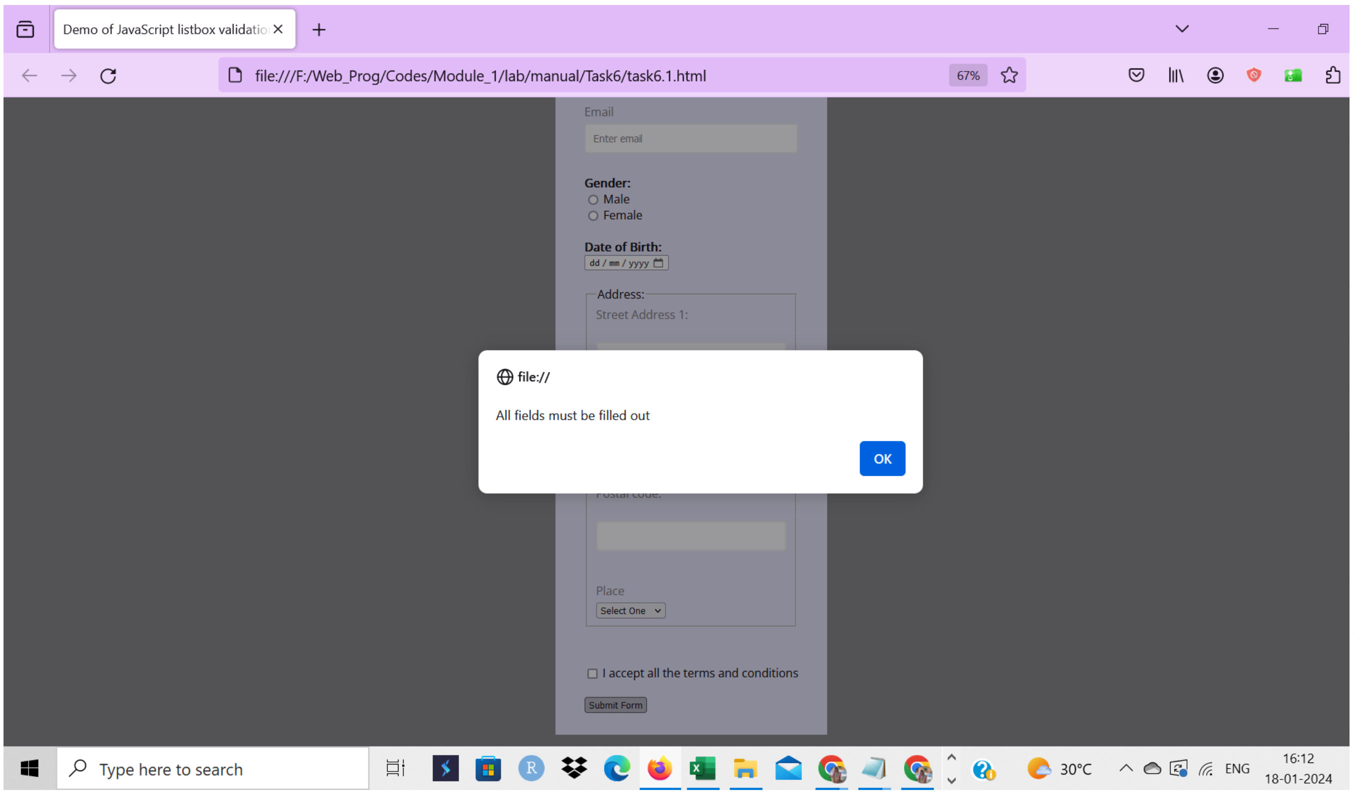
Type here to search
30°C
ENG
16:09
18-01-2024

Job2:

```
<html><head>
<title>Demo of JavaScript listbox validation in a form</title>
<link rel="stylesheet" href="style.css" />
<script type="text/javascript">

function data_check()
{
var firstName = document.getElementById('uname').value;
var email = document.getElementById('email').value;
if (firstName === '' || email === '') {
                alert('All fields must be filled out');
                return false;
            }
var str=document.getElementById("l1").value;
if(str.length <=0){
alert("Please select one option ");
}else{
document.forms['drop_list'].submit();// working
const male = document.querySelector('.male');
  const female = document.querySelector('.female');
  if(!male.checked && !female.checked){
    alert('Please choose your Gender: Male or Female ')
  }else{
    alert('Form submitted!')
  }
}
 var checkbox = document.getElementById("accept");
 if (!checkbox.checked){
  document.getElementById("error").innerHTML = "You must accept the terms and condition
by checking the Checkbox";
  return false;
 }
}
</script>
</head>
<body>
<form     name="drop_list"     method="post"     id='f1'class="form"     style="background
color:#E6E6FA">
        <div class="form-control">
          <label for="uname">Username</label>
          <input type="text" id="uname" placeholder="Enter username" />
          <small>Error message</small>
        </div>
        <div class="form-control">
          <label for="email">Email</label>
          <input type="text" id="email" placeholder="Enter email" />
          <small>Error message</small>
        </div>
       <div class="formdivider">
         <label for="class"><b>Gender:</b></label><br>
         <input type="radio" class="male" name="gender" value="Male"> Male <br>
         <input type="radio" class="female" name="gender" value="Female"> Female <br>
       </div><br>
       <div class=""form-control">
        <label for="class"><b>Date of Birth:</b></label><br>
        <input type="date"name="dt">
        </div><br>
<div class="form-control">
<fieldset>
         <legend>Address:</legend>
```

```
          <label for="saddr">Street Address 1:</label><br>
          <input type="text" id="fname" name="saddr"><br>
          <label for="city">City:</label><br>
          <input type="text" id="city" name="city"><br><br>
          <label for="postal">Postal code:</label><br>
          <input type="text" id="pc" name="Postal"><br><br>
          <label for="place">Place</label>
         <select id="l1" name="Category">
            <option value=''>Select One</option>
            <option value="chennai">Chennai</option>
            <option value="cbe">Coimbatore</option>
            <option value="mad">Madurai</option>
            <option value="tri">Trichy</option>
        </select>
        </fieldset>
     </div><br>
      <div class="formdivider">
      <input type="checkbox" id="accept" required name="terms"> I accept all the terms anc
conditions<br>
      </div>
        <br>
<input onclick="data_check()"; value="Submit Form" type="button">
</form>
</body>
</html>
```

Output



All fields must be filled out

Style.css

```
@import url('https://fonts.googleapis.com/css?family=Open+Sans&display=swap');
:root {
  --success-color: #2ecc71;
  --error-color: #e74c3c;
}
* {
  box-sizing: border-box;
}
body {
  background: linear-gradient(to bottom, rgba(0,0,0,.5), rgba(0,0,0,.5)), url("./img1.jpg");
  background-position: center;
  font-family: 'Open Sans', sans-serif;
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  margin: 0;
}
.container {
  background-color: #fff;
 border-radius: 5px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.3);
  width: 400px;
}
h2 {
  text-align: center;
  margin: 0 0 20px;
}
.form {
  padding: 30px 40px;
}
.form-control {
  margin-bottom: 10px;
  padding-bottom: 20px;
  position: relative;
}
.form-control label {
  color: #777;
  display: block;
  margin-bottom: 5px;
}
.form-control input {
  border: 2px solid #f0f0f0;
  border-radius: 4px;
  display: block;
  width: 100%;
  padding: 10px;
  font-size: 14px;
}
.form-control input:focus {
  outline: 0;
  border-color: #777;
}
.form-control.success input {
  border-color: var(--success-color);
}
.form-control.error input {
  border-color: var(--error-color);
}
```

```
.form-control small {
  color: var(--error-color);
  position: absolute;
  bottom: 0;
  left: 0;
  visibility: hidden;
}
.form-control.error small {
  visibility: visible;
}
.formdivider small {
  color: var(--error-color);
  position: absolute;
  bottom: 0;
  left: 0;
  visibility: hidden;
}
.formdivider.error small {
  visibility: visible;
}
.form button {
  cursor: pointer;
  background-color: #3498db;
  border: 2px solid #3498db;
  border-radius: 4px;
  color: #fff;
  display: block;
  font-size: 16px;
  padding: 10px;
  margin-top: 20px;
  width: 100%;
}
```

Task 7

Problem:

Implement the following dynamic interactions on a web page using HTML and JavaScript:

i. Create a 'click' event handler to toggle the background color of area 'A' between 'silver' and 'gold'.

ii. Implement a 'mouseover' event handler for area 'B' that cycles its background color through "blue", "lime", "red", "yellow", "green", and back to "blue" with each mouse movement.

iii. Use 'mouseenter' and 'mouseleave' event handlers for area 'C' to change its background color to "red" when the mouse enters and to "green" when it leaves.

iv. Apply 'dblclick' event handlers for area 'D' to modify its background color to "pink"

Procedures:

HTML Structure:

- The HTML document begins with the <!DOCTYPE html> declaration.
- Inside the <head> section, there is a <script> tag containing JavaScript functions and event listeners.
- The <style> tag includes CSS rules for styling.

JavaScript Functions:

- registerAllEvents(): This function is called when the body of the HTML document is loaded. It registers various event listeners for the specified div elements.
- handleDivA(evt): Toggles the background color of div A between silver and gold on a 'click' event.
- handleDivB(evt): Changes the background color of div B in a cyclic pattern ("blue" -> "lime" -> "red" -> "yellow" -> "green" -> "blue") on a 'mousemove' event.
- handleDivC_1(evt): Changes the background color of div C to red on a 'mouseenter' event.
- handleDivC_2(evt): Changes the background color of div C to green on a 'mouseleave' event.
- handleDivD_1(evt): Changes the background color of div D to pink on a 'dblclick' event.

CSS Styling:

- The CSS styles define a grid layout with two rows and two columns, with a small gap between the cells.

- The background color of the grid container is set to AliceBlue, and each cell has a blue background with a font size of 3em.

Body Section:

- The <body> tag has an onload attribute set to registerAllEvents(), ensuring that the JavaScript functions are executed when the body is loaded.
- Inside the body, there is a <div> element with the class "container" containing four child div elements (divA, divB, divC, and divD), each representing a cell in the grid.

Event Listeners:

- Event listeners are added using addEventListener in the registerAllEvents() function, associating specific event types with corresponding event handler functions.

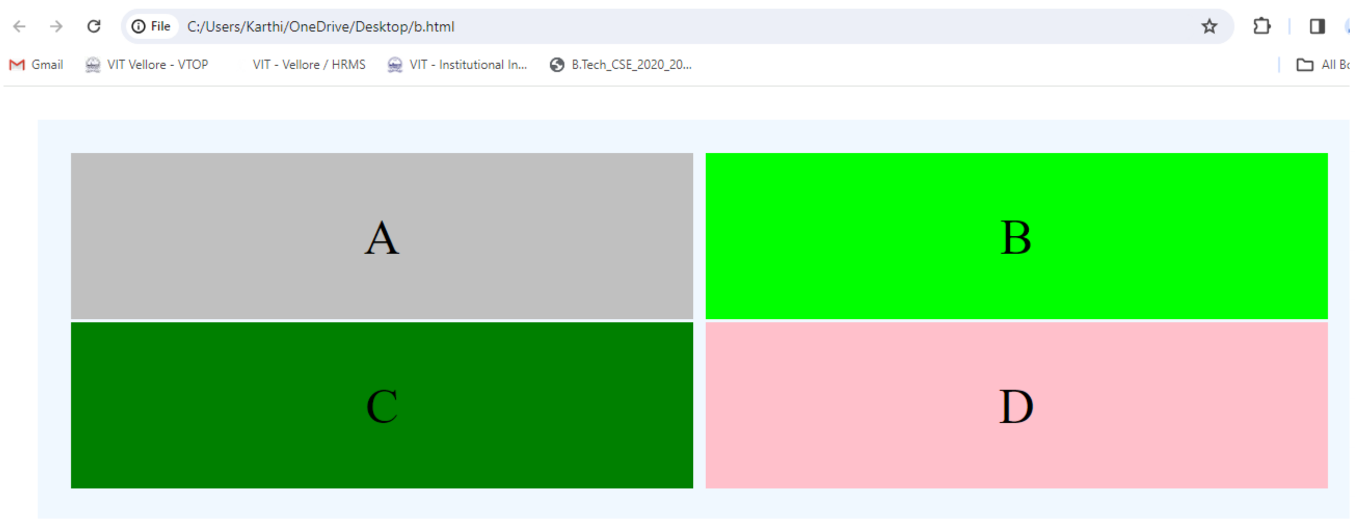
Event Handling:

- Event handler functions (handleDivA, handleDivB, handleDivC_1, handleDivC_2, and handleDivD_1) modify the background color of the respective div elements based on the specified conditions.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <script>
 function registerAllEvents() {
    // register different event types
    document.getElementById("divA").addEventListener('click', handleDivA);
    document.getElementById("divB").addEventListener('mousemove', handleDivB);
    // two event handler for the same element
    document.getElementById("divC").addEventListener('mouseenter', handleDivC_1);
    document.getElementById("divC").addEventListener('mouseleave', handleDivC_2);
  document.getElementById("divD").addEventListener('dblclick', handleDivD_1);
  }
  function handleDivA(evt) {
    // toggle between silver and gold
    const elem = evt.target;
    if (elem.style.backgroundColor !== 'silver') {
      elem.setAttribute("style", "background-color: silver");
    } else {
      elem.setAttribute("style", "background-color: gold");
    }
  }
  function handleDivB(evt) {
    // rotate in a palette of colors
    const elem = evt.target;
    switch (elem.style.backgroundColor) {
      case "blue":
        elem.setAttribute("style", "background-color: lime");
        break;
      case "lime":
        elem.setAttribute("style", "background-color: red");
        break;
      case "red":
        elem.setAttribute("style", "background-color: yellow");
        break;
      case "yellow":
        elem.setAttribute("style", "background-color: green");
        break;
      default:
        elem.setAttribute("style", "background-color: blue");
        break;
    }
  }
  function handleDivC_1(evt) {
    // switch to a certain color
    evt.target.setAttribute("style", "background-color: red");
  }
  function handleDivC_2(evt) {
    // switch to a certain color
    evt.target.setAttribute("style", "background-color: green");
  }
 function handleDivD_1(evt) {
    // switch to a certain color
    evt.target.setAttribute("style", "background-color: pink");
  }

  </script>
  <style>
   .container {
      display: grid;
      grid-template-columns: 49% 49%;
      grid-template-rows:    10em 10em;
      gap: 1%;
      background-color: AliceBlue;
      margin: 2em;
      padding: 2em;
    }
    .cell {
      display: flex;
      align-items: center;
      justify-content: center;
      background-color: blue;
      font-size: 3em;
    }
  </style>
</head>
<body id="body" onload="registerAllEvents()">
  <div class="container">
    <div id="divA" class="cell">A</div>
    <div id="divB" class="cell">B</div>
    <div id="divC" class="cell">C</div>
    <div id="divD" class="cell">D</div>
</body>
</html>
```

Output:


File C:/Users/Karthi/OneDrive/Desktop/b.html
Gmail
VIT Vellore - VTOP
VIT - Vellore / HRMS
VIT - Institutional In...
B.Tech_CSE_2020_20...
A
B
C
D

Task 8

Problem:

Create a simple counter that increments dynamically on-screen as the user clicks on the respective button by applying functional component state with hooks. Also, design a decrement counter by utilizing the concepts of class component life cycle and constructor to build a react application.

Procedures:

Functional component (Increment Counter)

1. Create a functional component named Counter
2. Utilize the useState hook for managing the state of the counter.
3. Display the current count on the screen.
4. Implement two buttons, enabling users to increment and decrement the count.

Counter.js

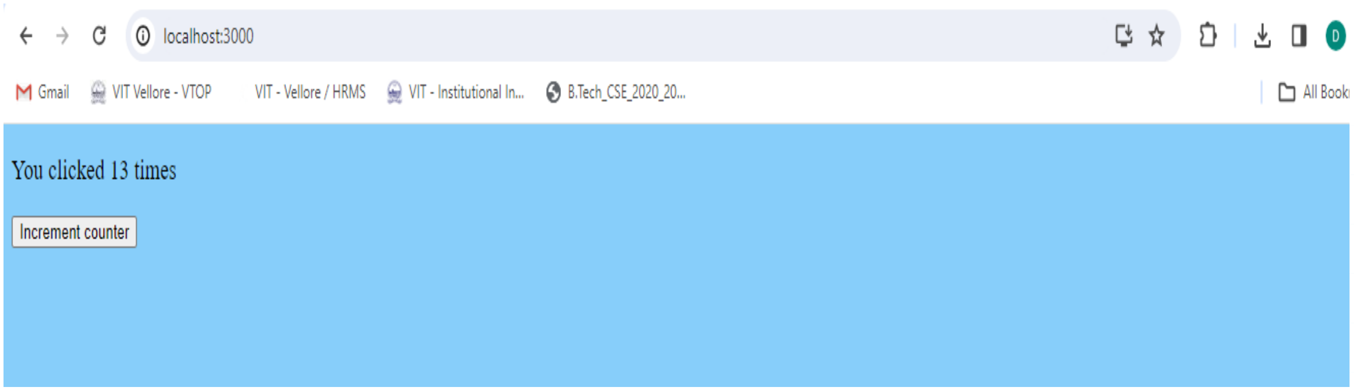
```
import React, { useState, useEffect } from 'react';
function Counter() {
 const [count, setCount] = useState(0);
 const incrementCount = () => setCount(count + 1);
 useEffect(() => {
  document.title = `You clicked ${count} times`
 });
 return (
  <div>
   <p>You clicked {count} times</p>
   <button onClick={incrementCount}>Increment counter </button>
  </div>
 )
}
export default Counter;
```

App.js

```
import Counter from './Counter'
function App() {
  return (
    <div classname="App">
  <Counter/>
```

```
    </div>
  );}

export default App;
```

Output:



You clicked 13 times

Class Component

1. Create a class component named Counter.
2. In the constructor, initialize the state to manage the counter.
3. Display the current count on the screen.
4. Implement two buttons for incrementing and decrementing the count.
5. Incorporate the componentDidMount and componentDidUpdate life cycle methods tc log messages in the console, indicating when the component has mounted and when i has updated.

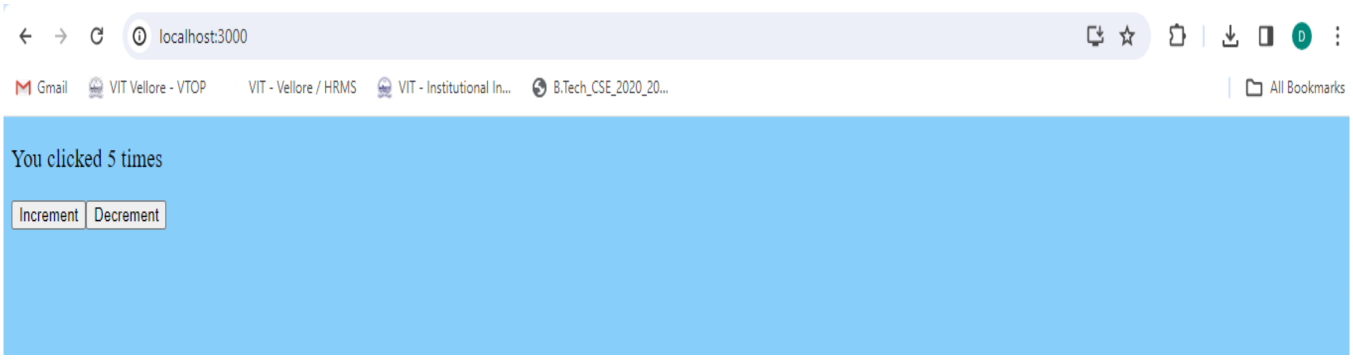
Counter.js

```
import React from 'react';
class Counter extends React.Component {
  constructor(count) {
    super(count);
    this.state = { count: 0 };
    this.incrementCount = this.incrementCount.bind(this);
    this.decrementCount = this.decrementCount.bind(this);
  }
  incrementCount() {
    this.setState({ count: this.state.count + 1 });
  }
  decrementCount() {
    this.setState({ count: this.state.count - 1 });
  }
  componentDidMount() {
    document.title = `You clicked ${this.state.count} times`;
  }
  componentDidUpdate() {
    document.title = `You clicked ${this.state.count} times`;
  }
  render() {
    return (
      <div>
        <p>You clicked {this.state.count} times</p>
        <button onClick={this.incrementCount}>Increment</button>
        <button onClick={this.decrementCount}>Decrement</button>
      </div>
    );
  }
}

export default Counter;
```

App.js

```
import Counter from './Counter'
function App() {
    return (
        <div classname="App">
    <Counter/>
      </div>
    );}

export default App;
```

Output:


localhost:3000

You clicked 5 times

Task 9

Problem:

Implement react concepts to design a simple search filter functionality to display a filtered list based on the search query entered by the user.

Procedures:

- Create a new app using the following command.
  npx create-react-app project name.
- Navigate to the project folder in command prompt.
  cd sample
- Create a new component named "Search.js" which is responsible for handling the search functionality.
- Import the Search component in the main App component and run the program.
- To run the react application give the command in command prompt as
  npm start
- Open browser and specify the url as
  localhost:3000

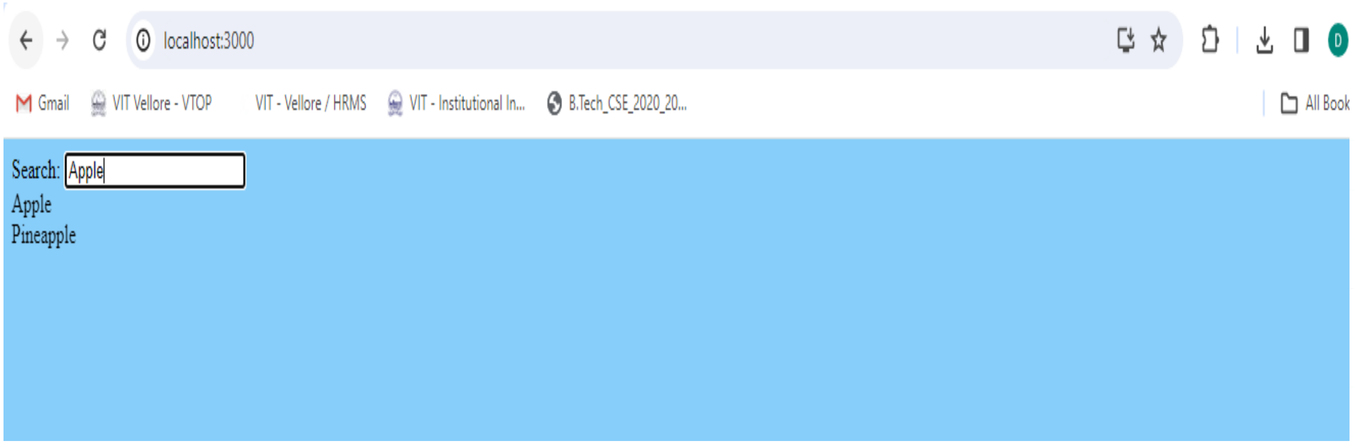
Search.js

```
import React, { useState } from "react";
function Search() {
 const list = [
  "Banana",
  "Apple",
  "Orange",
  "Mango",
  "Pineapple",
  "Watermelon"
 ];
 const [filterList, setFilterList] = useState(list);
 const handleSearch = (event) => {
  if (event.target.value === "") {
   setFilterList(list);
   return;
  }
  const filteredValues = list.filter(
   (item) =>
    item.toLowerCase().indexOf(event.target.value.toLowerCase()) !== -1
    );
    setFilterList(filteredValues);
  };
  return (
    <div className="app">
      <div>
        Search: <input name="query" type="text" onChange={handleSearch} />
      </div>
      {filterList &&
        filterList.map((item, index) => (
          <div key={index}>{item}</div> //Display each item
        ))}
    </div>
  );
}
export default Search;
```

App.js

```
import Search from './Search'
function App() {
    return (
        <div classname="App">
    <Search/>

      </div>
    );}
export default App;
```

Output:



```
localhost:3000
Search: Apple
Apple
Pineapple
```

Task 10

Problem:

Apply props validation using PropTypes to ensure that all the required props are present. Validate the following data types for the respective props:

a. userName (string): Ensure it is a string.
b. userAge (number): Ensure it is a number.
c. userSkills (array): Ensure it is an array.
d. isActive (boolean): Ensure it is a boolean.
e. updateUser (function): Ensure it is a function.

Procedures:

- Create a new app using the following command.
  ```
  npx create-react-app project name.
  ```
- Navigate to the project folder in command prompt.
  ```
  cd project name
  ```
- Import Necessary Dependencies:
  - Import React and Component from the 'react' library.
  - Import PropTypes for prop type validation.
- Create a Class Component:
  - Define a class component named User that extends React.Component.
- Implement the Render Method:
  - Inside the class, implement the render method to define the component's structure and UI.
- Structure the Component:
  - Within the render method, structure the component to include a title, a table, and rows for displaying user details.
- Use Props in the Component:
  - Utilize this.props to access the props passed to the component and display them in the appropriate table cells.
- PropTypes Validation:
  - Below the class definition, use User.propTypes to define the PropTypes validation for each prop type.
  - Ensure that propArray is an array and is required.
  - Ensure that propBool is a boolean and is required.
  - Allow propFunc to be a function.
  - Allow propNumber to be a number.
  - Allow propString to be a string.
- Default Props:

    - After PropTypes, use User.defaultProps to set default values for each prop.
    - Provide default values for propArray, propBool, propFunc, propNumber, and propString.
- Export the Component:
    - Export the User component as the default export of the module.
- Import the User component in the main App component and run the program.
- To run the react application give the command in command prompt as

  npm start
- Open browser and specify the url as

  localhost:3000

User.js

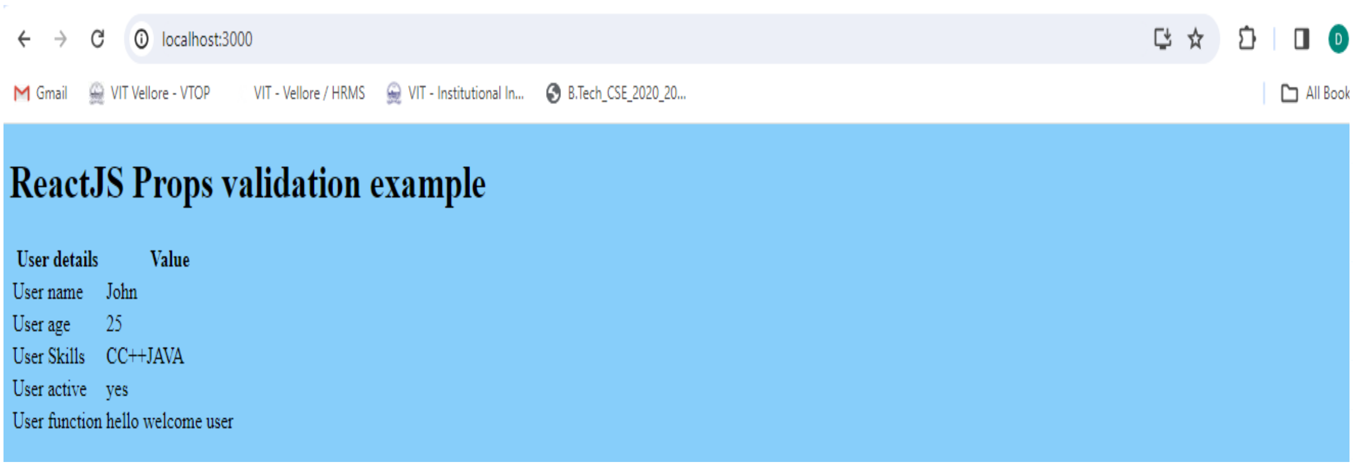
```
import React, { Component } from 'react';
import PropTypes from 'prop-types';
class User extends React.Component {
 render() {
 return (
 <div>
  <h1>ReactJS Props validation example</h1>
   <table>
     <tr>
     <th>User details</th>
      <th>Value</th>
      </tr>
      <tr>
     <td>User name</td>
     <td>{this.props.propString}</td>
  </tr>
  <tr>
     <td>User age</td>
      <td>{this.props.propNumber}</td>
  </tr>
   <tr>
   <td>User Skills</td>
   <td>{this.props.propArray}</td>
  </tr>
  <tr>
  <td>User active</td>
  <td>{this.props.propBool ? "yes" : "no"}</td>
 </tr>
```

```
  <tr>
  <td>User function</td>
  <td>{this.props.propFunc("hello welcome user")}</td>
  </tr>
  </table>
 </div>
      );
  }
}

User.propTypes = {
   propArray: PropTypes.array.isRequired,
   propBool: PropTypes.bool.isRequired,
   propFunc: PropTypes.func,
   propNumber: PropTypes.number,
   propString: PropTypes.string,
}
User.defaultProps = {
   propArray: ["C","C++","JAVA"],
   propBool: true,
   propFunc: function(x){return x},
   propNumber: 25,
   propString: "John",
}
export default User;
```

App.js

```
import User from './User'
function App() {
   return (
      <div classname="App">
   <User/>

     </div>
   );}

export default App;
```

Output:



localhost:3000

Gmail VIT Vellore - VTOP VIT - Vellore / HRMS VIT - Institutional In... B.Tech_CSE_2020_20... All Book

```
ReactJS Props validation example

 User details      Value
User name     John
User age      25
User Skills   CC++JAVA
User active   yes
User function hello welcome user
```

Task 11

Problem:

Create a React application with four distinct web pages: Home, About, Product and Contact. Utilize React Router to establish the necessary routes for navigation between these pages. Additionally, implement a navigation bar that provides users with the ability to switch effortlessly between the Home, About, Product and Contact pages.

Procedures:

1. Install router by using the following command.

```
npm install react-router-dom@6 –save
```

2. Create four components Home.js, About.js, Product.js and Contact.js along with App.js, which is already present.
3. **Home.js**

```
function Home(){
  return(
     <>
     <h1>Home Page</h1>
     <p>Flipkart offers a vast array of products ranging from electronics, appliances, and fashion to books, furniture, and more. The platform has gained widespread recognition for its user-friendly interface, a diverse product catalog, and competitive pricing. Additionally, Flipkart has introduced innovative features such as cash-on-delivery, easy returns, and various payment options, contributing to its widespread adoption.</p>
     </>
  )
  }
  export default Home;
```

4. **About.js**

```
function About(){
return(
   <>
   <h1>About Page</h1>
   <p>The Flipkart Group
   The Flipkart Group is one of India's leading digital commerce entities and includes group companies Flipkart, Myntra, Flipkart Wholesale, Flipkart Health+, Cleartrip and ANS Commerce.</p>
   </>
)
}
```

```
export default About
```

5. **Product.js**

```
function Product(){
   return(
      <>
      <h1>List of Products</h1>
      <ol>
         <li>Mobile</li>
         <li>Fashion</li>
         <li>Electronins</li>
         <li>Home and Furniture</li>
         <li>Appliances</li>
         <li>Travel</li>
         <li>Beauty,Toys and more</li>
      </ol>
      </>

   )
   }
   export default Product
```

6. **Contact.js**

```
function Contact(){
   return(
      <>
      <h1>Contact Page</h1>
      <p>Flipkart Internet Private Limited,
              Buildings Alyssa, Begonia &
              Clove Embassy Tech Village,
              Outer Ring Road, Devarabeesanahalli Village,
              Bengaluru, 560103,
         Karnataka, India
         CIN : U51109KA2012PTC066107
         Telephone: 044-45614700</p></>
   )
   }
   export default Contact
```

7. **For Routing, open the index.js file and import all the three component files in it.** Also import the following line

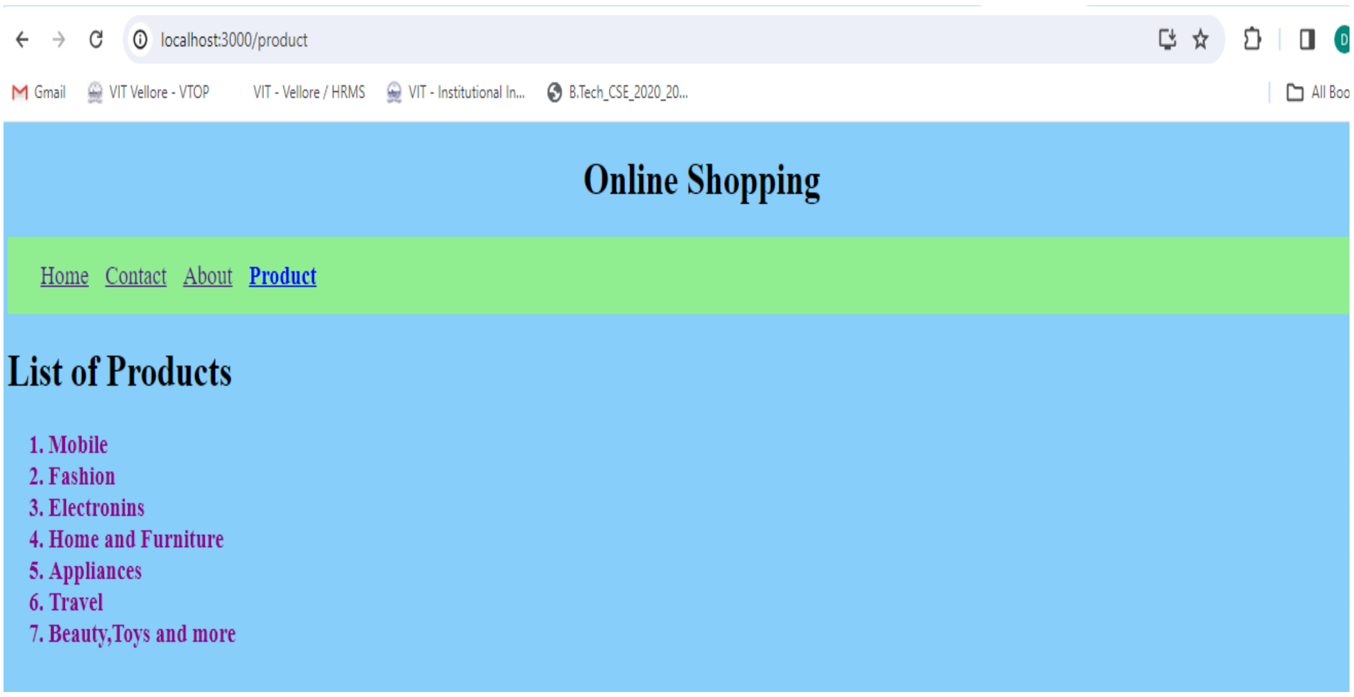
```
import {BrowserRouter} from "react-router-dom";
```
Add the following line inside render method of index.js
```
root.render(
  <React.StrictMode>
    <BrowserRouter>
    <App />
    </BrowserRouter>
  </React.StrictMode>
);
```
8. **Create Navpage.js to include link component**
```
import { NavLink } from 'react-router-dom'
function Navpage()
{
    return(
        <nav>
            <NavLink to='/home'>Home</NavLink>
            <NavLink to='/contact'>Contact</NavLink>
            <NavLink to='/about'>About</NavLink>
            <NavLink to='/product'>Product</NavLink>
        </nav>
    )
}
export default Navpage;
```
9. **Include style in index.css**
```
nav{
  background-color:lightgreen;
  padding: 16px 32px;
}
nav a
{
  margin-right:16px;
  font-size: larger;
}
nav a.active{
  color: blue;
  font-weight: bold;
}
body{
  background-color: lightskyblue;
}
```

```
p
{
  font-size: larger;
}
li{
  font-size: larger;
  color: darkmagenta;
  font-weight: bold;
}
```

10. Include Navpage and Route path in App.js

```
import {Routes,Route} from 'react-router-dom'
import Home from './Home'
import Contact from './Contact'
import Navpage from './Navpage'
import About from './About'
import Product from './Product'
function App() {
   return(
<>
<center><h1>Online Shopping</h1></center>
<Navpage />
 <Routes>
  <Route path='/home' element={<Home/>} />
  <Route path='/contact' element={<Contact/>} />
  <Route path='/about' element={<About/>} />
  <Route path='/product' element={<Product/>} />
</Routes>
</>
)
}export default App;
```

Output:



localhost:3000/product

Gmail VIT Vellore - VTOP VIT - Vellore / HRMS VIT - Institutional In... B.Tech_CSE_2020_20... All Boo

Online Shopping

Home Contact About Product

List of Products

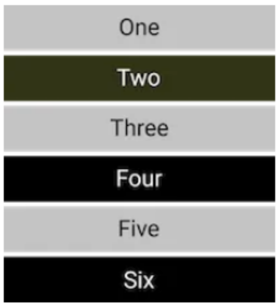
1. Mobile
2. Fashion
3. Electronins
4. Home and Furniture
5. Appliances
6. Travel
7. Beauty,Toys and more

Additional Exercises

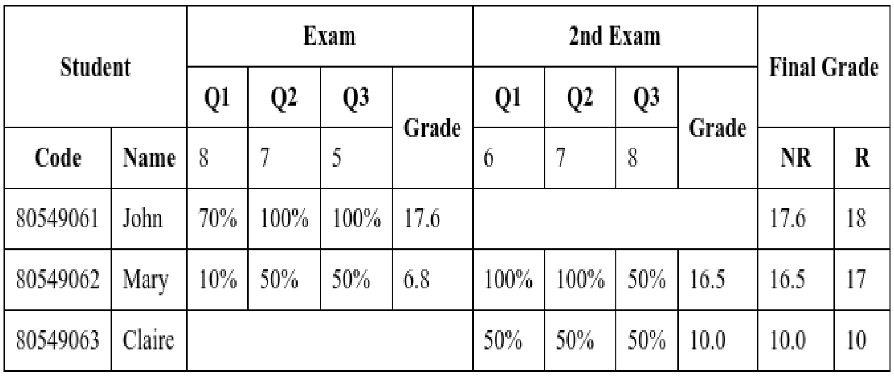
1. Write the HTML code with JavaScript to design the sales web app. The web page sells bags of coffee. You sell regular coffee at Rs.10/bag, you sell decaffeinated coffee at Rs-8/bag, and you sell mocha coffee for Rs-11/bag. If the user buys more than Rs-100, they get Rs-15 discount. Use the prompt to ask the user how many bags of regular coffee s/he wants. Then use the prompt to ask about decaf, and then about mocha. Using the input, calculate the total amount the user will pay. If the total is over Rs-100, subtract Rs-15 from their total. Finally write out how much the total will be. Use HTML layouts for design.

2. Write the JavaScript code to perform the following. Use a prompt () function to collect the student name. Use a switch statement to execute a dialog box displaying the phrase “Welcome &lt;the entered name&gt;“to the website. Use another prompt () function to collect the current CGPA as input by the student. If the CGPA entered is above 9.0, throw a message as “Selected for the Second Round”. If the CGPA is below 9.0, throw the message as“Enter the First round selection process by providing the E-Mail’. In the third prompt box get the Mobile number and validate with a JavaScript Regular expression with 10 digits only and give a welcome message with the Customized message of your choice.

3. Create a list of items using ul and li tags . Alternative colors for the backgrounds. There is space between each item.

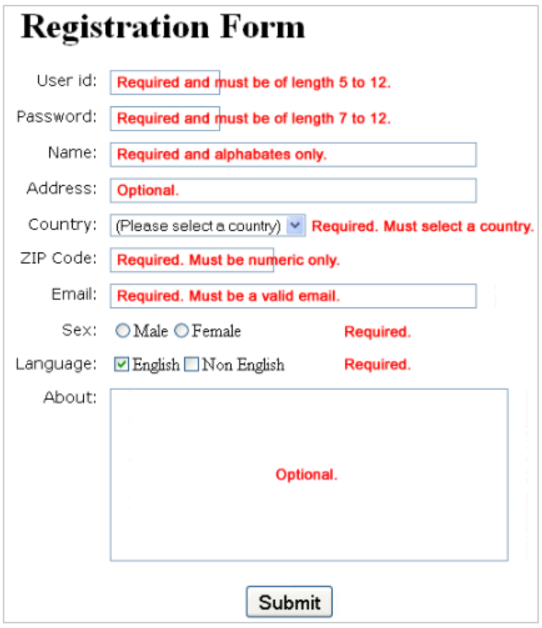


4. Create a document representing the following table



Student		Exam				2nd Exam				Final Grade	
		Q1	Q2	Q3	Grade	Q1	Q2	Q3	Grade		
Code	Name	8	7	5		6	7	8		NR	R
80549061	John	70%	100%	100%	17.6					17.6	18
80549062	Mary	10%	50%	50%	6.8	100%	100%	50%	16.5	16.5	17
80549063	Claire					50%	50%	50%	10.0	10.0	10

5. Design a web form and add validation code as given below



Registration Form

User id:	Required and must be of length 5 to 12.	
Password:	Required and must be of length 7 to 12.	
Name:	Required and alphabates only.	
Address:	Optional.	
Country:	(Please select a country)	Required. Must select a country.
ZIP Code:	Required. Must be numeric only.	
Email:	Required. Must be a valid email.	
Sex:	○ Male ○ Female	Required.
Language:	[✓ English] ☐ Non English	Required.
About:	Optional.	

Submit

6. Build a component with an input field and a button. The component should display a message based on the user input. Use state to manage the input value.
7. Create a custom hook for handling form input. This hook should manage the state of each form field and provide functions to update the values. Use this custom hook in a form component.
8. Build a component that fetches data from an API using the useEffect hook. Display the fetched data on the page.
9. Develop a web page for online medical store using different components such as home, contact, product and service. Apply routing concept to allow the user for performing navigation among different components.
10. Build a website for toy store and create components such as home, Toys category and contact. Apply different types of CSS for each component to improve the presentation of web pages.

*****