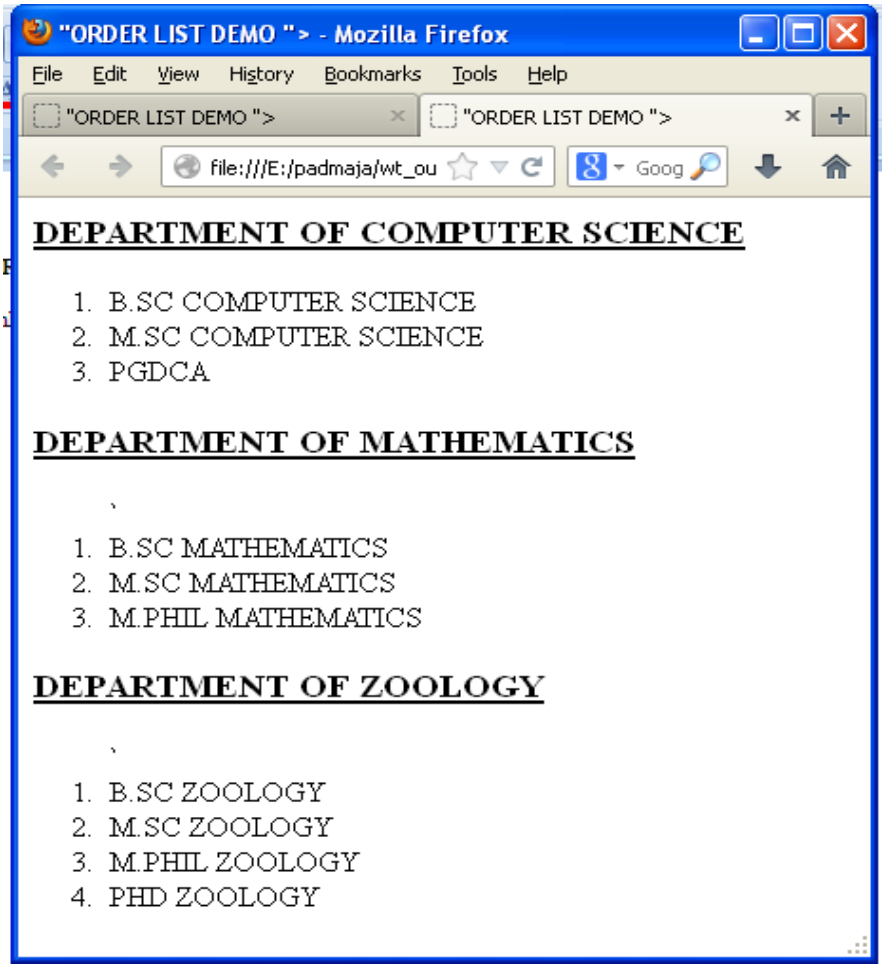
HTML Code

```
<HTML>
<HEAD>
    <TITLE>"ORDER LIST DEMO "></TITLE>
</HEAD>
<BODY>
    <H1>
        <U>DEPARTMENT OF COMPUTER SCIENCE</U>
    </H1>
    <OL TYPE="1">
        <LI>B.SC COMPUTER SCIENCE</LI>
        <LI>M.SC COMPUTER SCIENCE</LI>
        <LI>PGDCA</LI>
    </OL>
    <H1>
        <U>DEPARTMENT OF MATHEMATICS</U>
    </H1>
    <OL TYPE="4">
`       <LI>B.SC
  MATHEMATICS</LI>
 <LI>M.SC MATHEMATICS</LI>
        <LI>M.PHIL MATHEMATICS</LI>
    </OL>
    <H1>
        <U>DEPARTMENT OF ZOOLOGY</U>
    </H1>
    <OL TYPE="7">
`       <LI>B.SC
  ZOOLOGY</LI>
<LI>M.SC ZOOLOGY</LI>
        <LI>M.PHIL ZOOLOGY</LI>
        <LI>PHD ZOOLOGY</LI>
    </OL>
 </BODY>
</HTML>
```

OUTPUT:

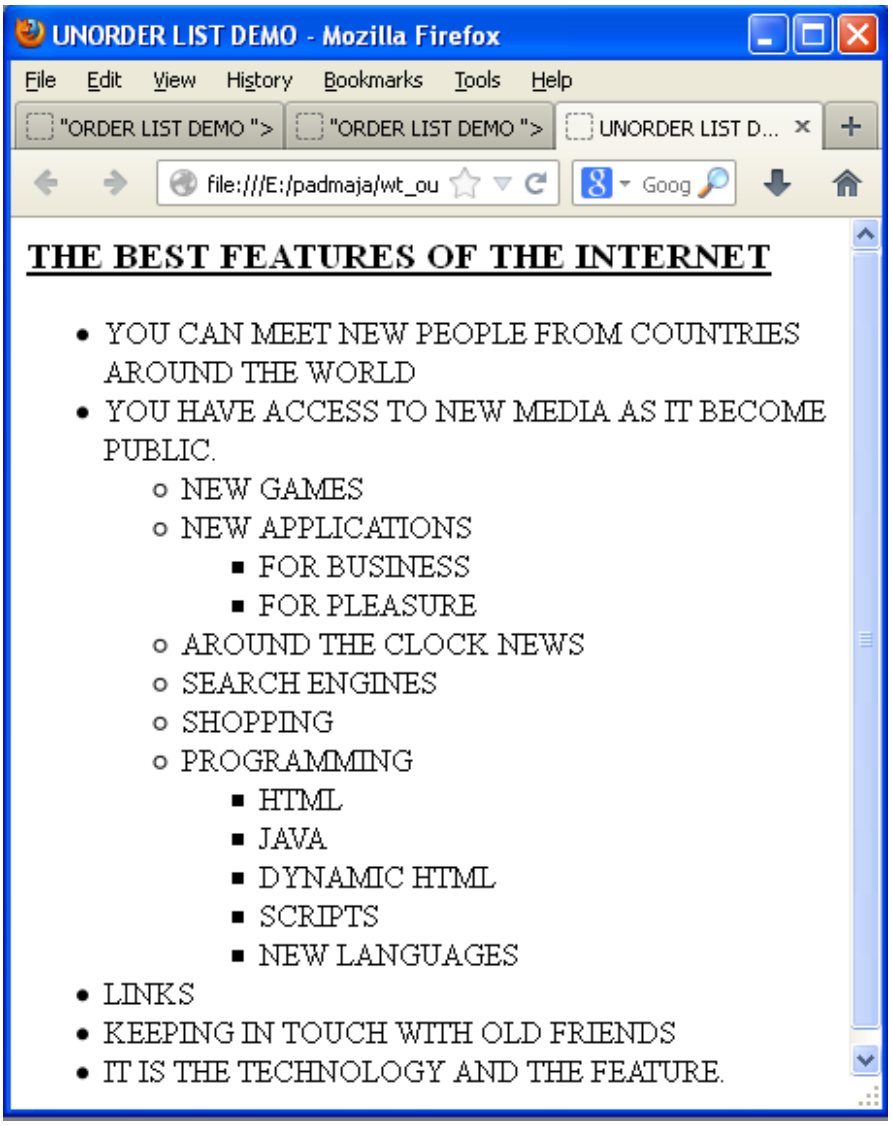

"ORDER LIST DEMO "> - Mozilla Firefox
File Edit View History Bookmarks Tools Help
"ORDER LIST DEMO ">
"ORDER LIST DEMO ">
file:///E:/padmaja/wt_ou
Goog
DEPARTMENT OF COMPUTER SCIENCE
1. B.SC COMPUTER SCIENCE
2. M.SC COMPUTER SCIENCE
3. PGDCA
DEPARTMENT OF MATHEMATICS
1. B.SC MATHEMATICS
2. M.SC MATHEMATICS
3. M.PHIL MATHEMATICS
DEPARTMENT OF ZOOLOGY
1. B.SC ZOOLOGY
2. M.SC ZOOLOGY
3. M.PHIL ZOOLOGY
4. PHD ZOOLOGY

.

HTML Code

```
<HTML>
    <BODY>
        <H3><U>THE BEST FEATURE OF THE INTERNET</U></H3>
        <UL TYPE="DISC">
            <LI>YOU CAN MEET NEW PEOPLE FROM
                COUNTRIES AROUND THE
                WORLD.</LI>
            <LI>YOU HAVE ACCESS TO NEW MEDIA AS
                IT BECOME PUBLIC.</LI>
                <UL TYPE="CIRCLE">
                    <LI>NEW GAMES</LI>
                    <LI>NEW APPLICATIONS</LI>
                        <UL TYPE="SQUARE">
                            <LI>FOR BUSINESS</LI>
                            <LI>FOR PLEASURE</LI>
                        </UL>
                    <LI>AROUND THE CLOCK NEWS</LI>
                    <LI>SEARCH ENGINES</LI>
                    <LI>SHOPPING</LI>
                    <LI>PROGRAMMING</LI>
                <UL TYPE="SQUARE">
                    <LI>HTML</LI>
                            <LI>JAVA</LI>
                            <LI>DYNAMIC HTML</LI>
                            <LI>SCRIPTS</LI>
                            <LI>NEW LANGUAGES</LI>
                        </UL>
                </UL>
<LI>LINKS</LI>
            <LI>KEEPING IN TOUCH WITH OLD FRIENDS</LI> <LI>IT IS THE
            TECHNOLOGY AND THE FEATURE.</LI>
</UL>
</BODY> </HTML>
```

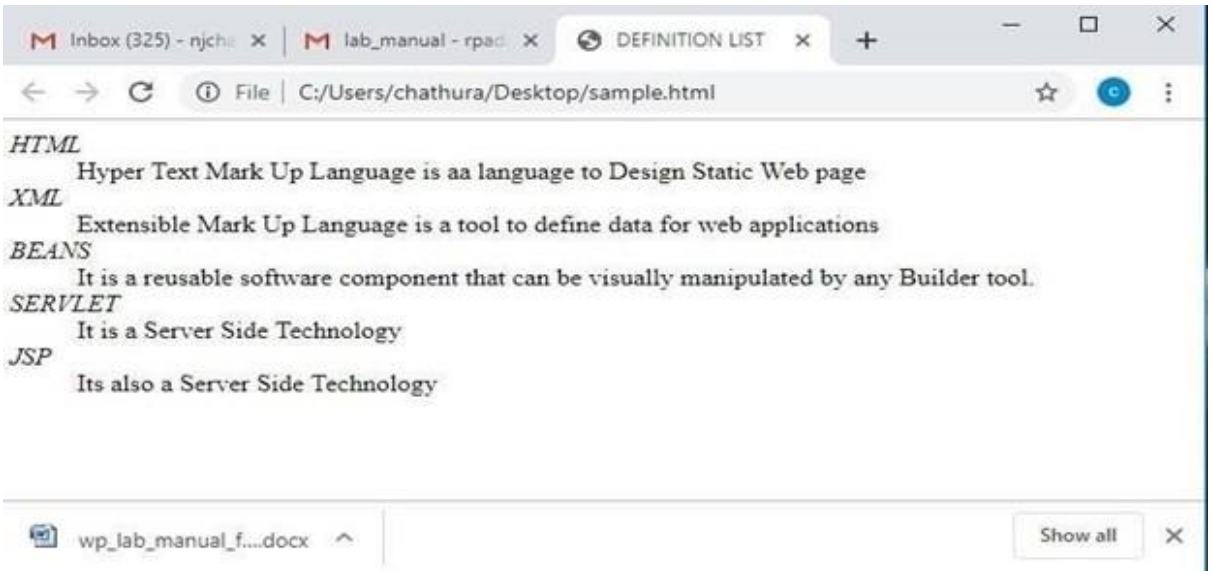
OUTPUT:


UNORDER LIST DEMO - Mozilla Firefox
File Edit View History Bookmarks Tools Help
"ORDER LIST DEMO ">
"ORDER LIST DEMO ">
UNORDER LIST D...
file:///E:/padmaja/wt_ou
Goog
THE BEST FEATURES OF THE INTERNET
YOU CAN MEET NEW PEOPLE FROM COUNTRIES AROUND THE WORLD
YOU HAVE ACCESS TO NEW MEDIA AS IT BECOME PUBLIC.
NEW GAMES
NEW APPLICATIONS
FOR BUSINESS
FOR PLEASURE
AROUND THE CLOCK NEWS
SEARCH ENGINES
SHOPPING
PROGRAMMING
HTML
JAVA
DYNAMIC HTML
SCRIPTS
NEW LANGUAGES
LINKS
KEEPING IN TOUCH WITH OLD FRIENDS
IT IS THE TECHNOLOGY AND THE FEATURE.

HTML Code

```
<HTML>
<HEAD>
        <TITLE>DEFINITION LIST</TITLE>
</HEAD>
                <BODY BGCOLOR="VIOLET" TEXT="MAROON">
                <DL>
                        <DT><I>HTML</I></DT>
                                <DD>Hyper Text Mark
                                Up Language is aa
                                language to Design
                                Static Web page</DD>
                        <DT><I>XML</I></DT>
                                <DD>Extensible Mark Up Language is a  tool
                                        to define data for web applications </DD>
                        <DT><I>BEANS</I></DT>
                                <DD>It is a reusable software component that
                                can be  visually manipulated by any Builder
                                tool.</DD>
                        <DT><I>SERVLET</I></DT>
                                <DD> It is a Server Side Technology</DD>
                        <DT><I>JSP</I></DT>
                                <DD>Its also a Server Side Technology</DD>
                </DL>
                </BODY>
</HTML>
```

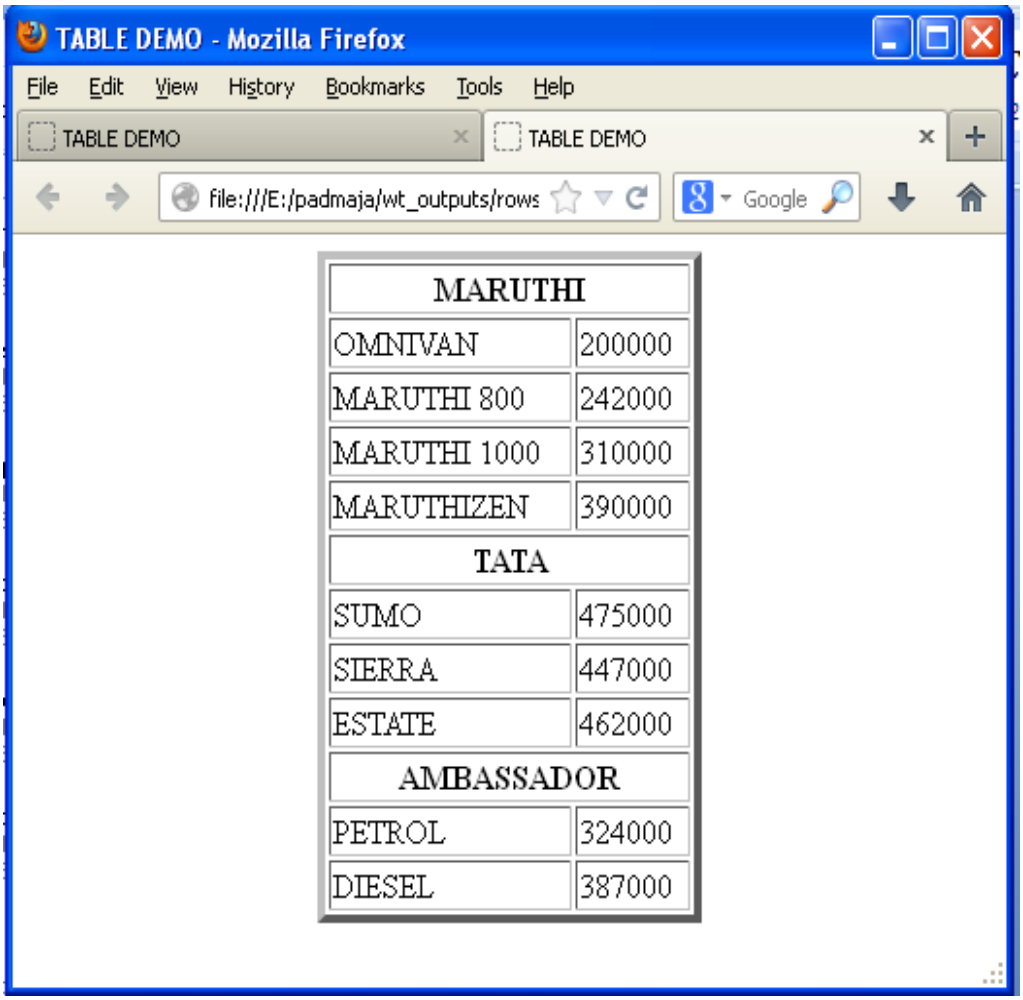
OUTPUT:


Inbox (325) - njch
lab_manual - rpad
DEFINITION LIST
File | C:/Users/chathura/Desktop/sample.html
HTML
Hyper Text Mark Up Language is aa language to Design Static Web page
XML
Extensible Mark Up Language is a tool to define data for web applications
BEANS
It is a reusable software component that can be visually manipulated by any Builder tool.
SERVLET
It is a Server Side Technology
JSP
Its also a Server Side Technology
wp_lab_manual_f....docx
Show all

HTML Code

```
<HTML>
<BODY>
        <TABLE BORDER=4 WIDTH="40%">
            <TR>
                <THCOLSPAN=4>MARUTHI</TH>
            </TR>
            <TR>
                <TD>OMNIVAN</TD>
                <TD>200000</TD>
            </TR>
            <TR>
                <TD>MARUTHI 800</TD>
                <TD>242000</TD>
            </TR>
            <TR>
                <TD>MARUTHI 1000</TD>
                <TD>310000</TD>
            </TR>
            <TR>
                <TD>MARUTHIZEN</TD>
                <TD>390000</TD>
            </TR>
            <TR>
                <THCOLSPAN=2>TATA</TH>
            </TR>
            <TR>
                <TD>SUMO</TD>
                <TD>475000</TD>
            </TR>
            <TR>
                <TD>SIERRA</TD>
                <TD>447000</TD>
            <TR>
                <TD>ESTATE</TD>
                <TD>462000</TD>
            </TR>
            <TR>
                <THCOLSPAN=2>AMBASSADOR</TH>
            </TR>
            <TR>
                <TD>PETROL</TD>
                <TD>324000</TD>
            </TR>
            <TR>
                <TD>DIESEL</TD>
                <TD>387000</TD>
            </TR>
        </TABLE>
<BODY> </HTML>
```

OUTPUT:

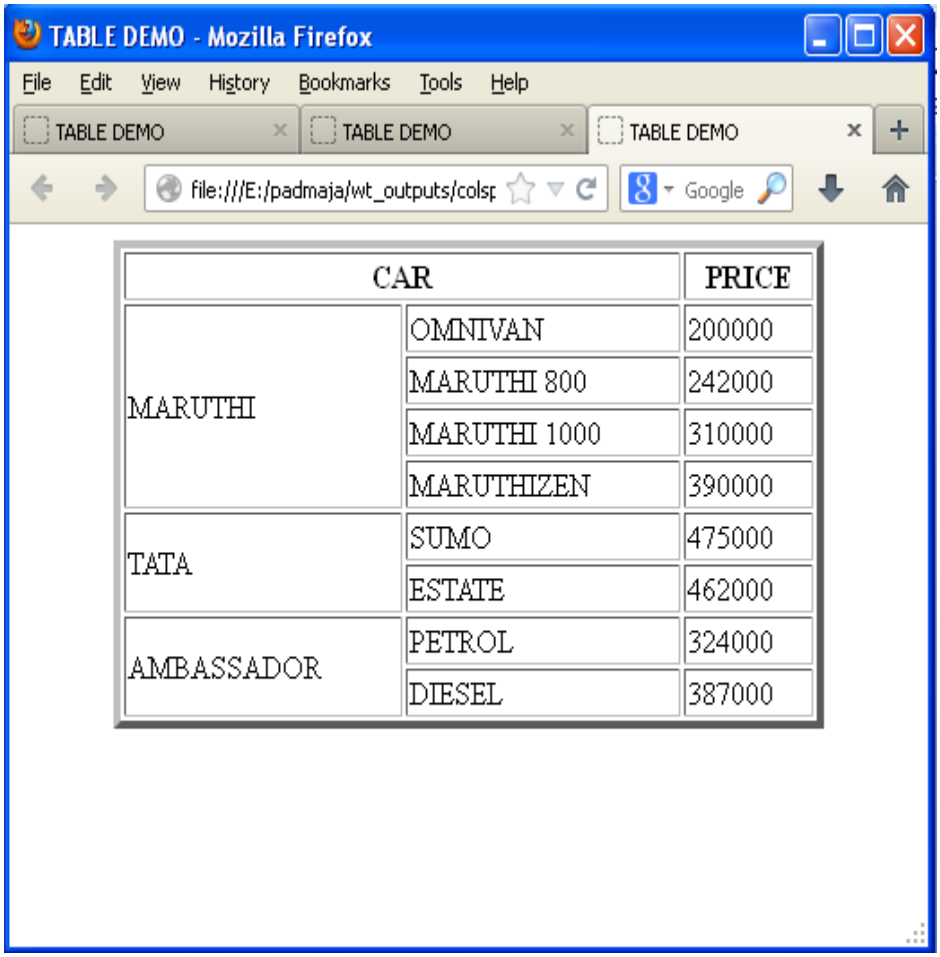


MARUTHI	
OMNIVAN	200000
MARUTHI 800	242000
MARUTHI 1000	310000
MARUTHIZEN	390000
TATA	
SUMO	475000
SIERRA	447000
ESTATE	462000
AMBASSADOR	
PETROL	324000
DIESEL	387000

HTML Code

```
<HTML>
    <BODY>
        <TABLE BORDER=4
            ALIGN="CENTER"
            WIDTH="60%">
            <TRCOLSPAN=2>
                <THCOLSPAN=2>CAR</TH>
                <TH>PRICE</TH>
            </TR>
            <TR>
                <TD ROWSPAN=4>MARUTHI</TD>
                <TD>OMNIVAN</TD>
            <TD>200000</TD>
        </TR>
        <TR>
            <TD>MARUTHI 800</TD>
            <TD>242000</TD>
        </TR>
        <TR>
            <TD>MARUTHI 1000</TD>
            <TD>310000</TD>
        </TR>
        <TR>
            <TD>MARUTHIZEN</TD>
            <TD>390000</TD>
            </TR>
            <TR>
                <TD ROWSPAN=2>TATA</TD>
                <TD>SUMO</TD>
            <TD>475000</TD>
        </TR>
        <TR>
            <TD>ESTATE</TD>
            <TD>462000</TD>
            </TR>
            <TR>
                <TD
                ROWSPAN=2>AMBA
                SSADOR</TD>
                <TD>PETROL</TD>
            <TD>324000</TD>
        </TR>
        <TR>
            <TD>DIESEL</TD>
            <TD>387000</TD>
            </TR>
        </TABLE>
</BODY> </HTML>
```

OUTPUT:



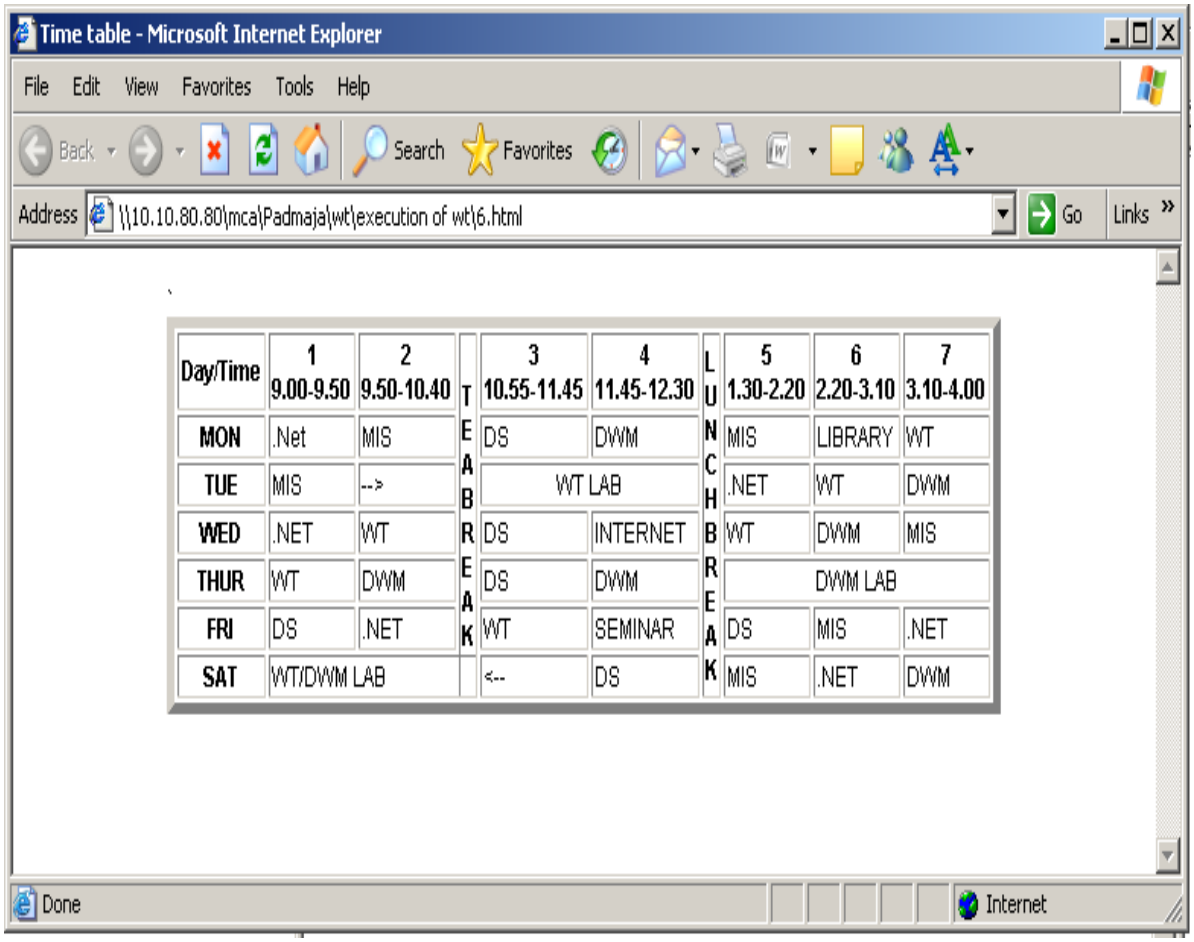
CAR		PRICE
MARUTHI	OMNIVAN	200000
	MARUTHI 800	242000
	MARUTHI 1000	310000
	MARUTHIZEN	390000
TATA	SUMO	475000
	ESTATE	462000
AMBASSADOR	PETROL	324000
	DIESEL	387000

HTML Code

```
<HTML>

        <HEAD>
                <TITLE>TIME TABLE</TITLE>
        </HEAD>
        <BODY>
                <TABLE BORDER="5" WIDTH="75%"
                        ALIGN="CENTER">
                <TR>
                        <TH>DAY/TIME</TH>
                        <TH>1</BR>9.00-9.50</TH>
                        <TH>2</BR>9.50-10.40</TH>
                        <TH ROWSPAN="7">T</BR>E
                                </BR>A</BR>B</BR>R</BR>E
                                </BR>A</BR>K</BR></TH>
                        <TH>3</BR>10.55-11.45</TH>
                        <TH>4</BR>11.45-12.30</TH>
                        <TH ROWSPAN="7">L</BR>U
                                </BR>N</BR>C</BR>H</BR>B
                                </BR>R</BR>E</BR>A</BR>K</BR></TH>
            <TH>5</BR>1.30-2.20</TH>
            <TH>6</BR>2.20-3.10</TH>
                                `
                        <TH>7</BR>
                    3.10-4.00</TH>
                </TR>
                <TR>
                        <TH>MON</TH>
                        <TD>.NET</TD>
                        <TD>MIS</TD>
                        <TD>DS</TD>
                        <TD>DWM</TD>
                        <TD>MIS</TD>
                        <TD>LIBRARY</TD>
                        <TD>WT</TD>
                </TR>
                <TR>
                        <TH>TUE</TH>
                        <TD>MIS</TD>
                        <TD>--></TD>
                        <TD COLSPAN = 2
                                ALIGN =
                                "CENTER"> WT
                                LAB</TD>
                        <TD>.NET</TD>
                        <TD>WT</TD>
                        <TD>DWM</TD>
                </TR>
                <TR>
```

```
                    <TH>WED</TH>
                    <TD>.NET</TD>

            <TD>DS</TD>
            <TD>INTERNET</TD>
            <TD>WT</TD>
            <TD>DWM</TD>
            <TD>MIS</TD>
        </TR>
        <TR>
            <TH>THUR</TH>
            <TD>WT</TD>
            <TD>DWM</TD>
            <TD>DS</TD>
            <TD>DWM</TD>
                <TD COLSPAN="3" ALIGN = "CENTER">
                    DWM LAB</TD>
        </TR>
        <TR>
            <TH>FRI</TH>
            <TD>DS</TD>
            <TD>.NET</TD>
            <TD>WT</TD>
            <TD>SEMINAR</TD>
            <TD>DS</TD>
            <TD>MIS</TD>
            <TD>.NET</TD>
        </TR>
        <TR>
            <TH>SAT</TH>
                <TD COLSPAN="3">WT/DWM LAB</TD>
            <TD><--</TD>
            <TD>DS</TD>
            <TD>MIS</TD>
            <TD>.NET</TD>
            <TD>DWM</TD>
        </TR>
        </TABLE>
        </BODY>


</HTML>
```

OUTPUT:



Day/Time	1 9.00-9.50	2 9.50-10.40	TEA BREAK	3 10.55-11.45	4 11.45-12.30	LUNCH BREAK	5 1.30-2.20	6 2.20-3.10	7 3.10-4.00
MON	.Net	MIS		DS	DWM		MIS	LIBRARY	WT
TUE	MIS	-->		WT LAB			.NET	WT	DWM
WED	.NET	WT		DS	INTERNET		WT	DWM	MIS
THUR	WT	DWM		DS	DWM		DWM LAB		
FRI	DS	.NET		WT	SEMINAR		DS	MIS	.NET
SAT	WT/DWM LAB			<--	DS		MIS	.NET	DWM

FRAMESET AND FRAME TAG ILLUSTRATION:

FramesetDemo.html

```
<html>
    <head>
        <title> Frames demo</title>
    </head>
    <frameset rows = "50%, 50%">
        <frameset cols = "50%,50%">
            <frame name = "html" src = "html.html"/>
            <frame name = "xml" src = "xml.html"/>
        </frameset>
        <frameset cols = "40%,30%,30%">
            <frame name = "bean" src = "bean.html"/>
            <frame name = "jsp" src = "jsp.html"/>
            <frame name = "servlet" src = "servlet.html"/>
        </frameset>
    </frameset>
</html>
```

html.html

```
<html>
  <head>
        <title>FrameDemo</title>
  </head>
  <body>
        <h3>HTML is a <b>Markup</b> language for describing web documents</h3>
        <ul type = "disc">
            <li> HTML stands for Hyper Text Markup Language</li>
            <li> A Markup Language is a set of Markup tags</li>
            <li> HTML is a universal documentation Language</li>
            <li> HTML is a commonly  used language for designing static web
            page</li>
         </ul>
    </body>
</html>
```

xml.html

```
<html>
<head>
  <title>FrameDemo</title>
</head>
<body>
  <h3><b>What is XML ?</b></h3>
  <ul type = "square">
        <li> XML stands for Extensible Markup Language</li>
        <li> XML was designed to describe data, not to display data</li>
        <li> XML tags are not predefined </li>
        <li> DTD and XSD are the languages to define XML document schema</li>
 </ul>
</body>
</html>
```

bean.html

```
<html>
    <head>
        <title>FrameDemo</title>
    </head>
    <body>
        <h2><b>What is Java Bean ?</b></h2>
            <p> Java Bean is a reusable software component
                which can be visually manipulated by any buider tools.
            </p>
    </body>
</html>
```

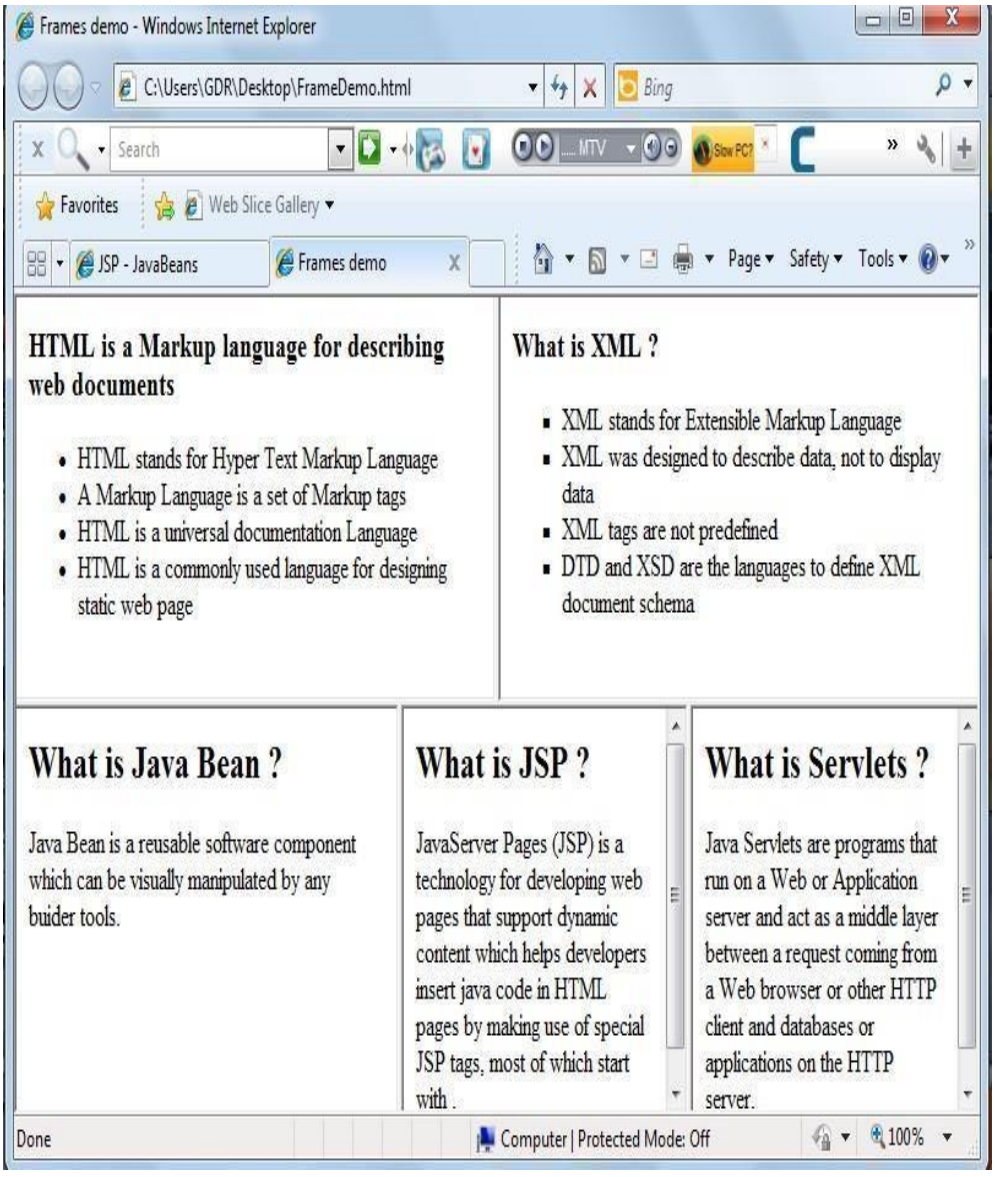
Servlet.html

```
<html>
  <head>
    <title>FrameDemo</title>
  </head>
  <body>
    <h2><b>What is Servlets ?</b></h2>

        <p> Java Servlets are programs that run on a Web or Application
            server and act as a middle layer between a request coming
            from a Web browser or other HTTP client and databases or
            applications on the HTTP server.
        </p>
  </body>
</html>
```

jsp.html

```
<html>
    <head>
        <title>FrameDemo</title>
    </head>
    <body>
        <h2><b>What is JSP ?</b></h2>
            <p>JavaServer Pages (JSP) is a technology for
                developing web pages that support
                dynamic content which helps
                developers insert java code in HTML
                pages by making use of special JSP
                tags, most of which start with <% and
                end with %>.
            </p>
        </body>
</html>
```

OUTPUT:

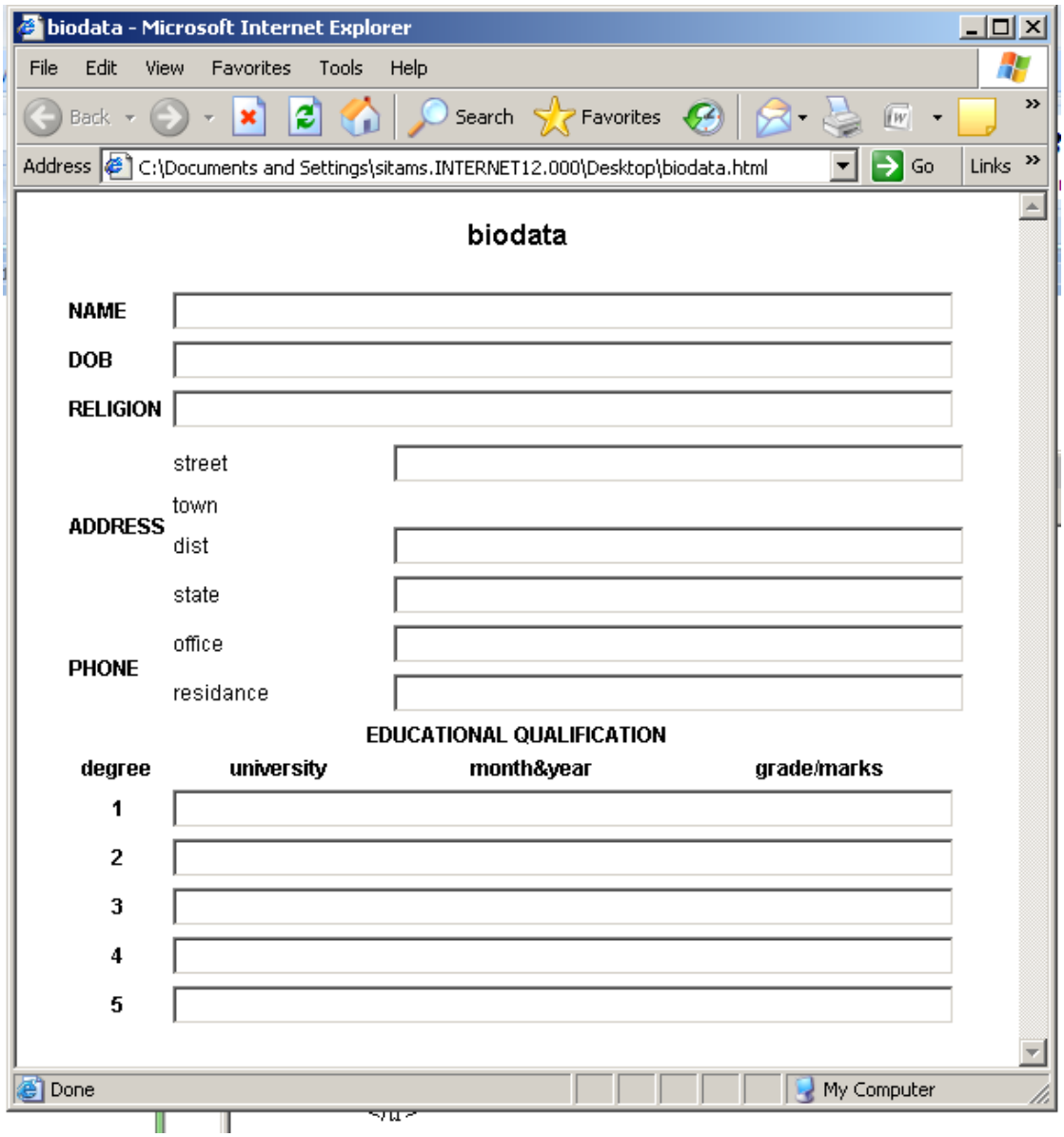

Frames demo - Windows Internet Explorer
C:\Users\GDR\Desktop\FrameDemo.html
Bing
Search
Favorites
Web Slice Gallery
JSP - JavaBeans
Frames demo
Page
Safety
Tools
HTML is a Markup language for describing web documents
HTML stands for Hyper Text Markup Language
A Markup Language is a set of Markup tags
HTML is a universal documentation Language
HTML is a commonly used language for designing static web page
What is XML ?
XML stands for Extensible Markup Language
XML was designed to describe data, not to display data
XML tags are not predefined
DTD and XSD are the languages to define XML document schema
What is Java Bean ?
Java Bean is a reusable software component which can be visually manipulated by any buider tools.
What is JSP ?
JavaServer Pages (JSP) is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with .
What is Servlets ?
Java Servlets are programs that run on a Web or Application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server.
Done
Computer | Protected Mode: Off
100%

BIODATA FORM USING FORM TAG:

```
<HTML>
    <HEAD>
                    <TITLE>BIODATA</TITLE>
    </HEAD>
    <BODY>
        <FORM NAME="BIODATA" METHOD="POST">
        <H2 ALIGN=CENTER>BIODATA</H2>
        <TABLE ALIGN=CENTER WIDTH="30%">
            <TR>
                <TH>NAME</TH>
                <TD COLSPAN="5"><INPUT TYPE="TEXT"  S IZE=70>
            </TR>
            <TR>
                <TH>D.O.B</TH>
                <TD COLSPAN="5"><INPUT TYPE="TEXT" SIZE=70>
            </TR>
            <TR>
                <TH>RELIGION</TH>
                <TD COLSPAN="5"><INPUT TYPE="TEXT" SIZE=70>
            </TR>
            <TR>
                <TH ROWSPAN="5">ADDRESS</TH>
            </TR>
            <TR>
                <TD>STREET</TD>
                <TD COLSPAN="4"><INPUT TYPE="TEXT" SIZE=50>
            </TR>
            <TR>
                <TD>TOWN</TD>
                <TD COLSPAN="4”><INPUT TYPE="TEXT" SIZE=50>
                </TR>
            <TR>
                <TD>DIST</TD>
                <TD COLSPAN="4"><INPUT TYPE="TEXT"  SIZE=50>
            </TR>
            <TR>
                <TD>STATE</TD>
                <TD COLSPAN="4"><INPUT TYPE="TEXT" SIZE=50>
            </TR>
            <TR>
                <TH ROWSPAN="2">PHONE</TH>
                <TD>OFFICE</TD>
                <TD COLSPAN="4"><INPUT TYPE="TEXT" SIZE=50>
            </TR>
            <TR>
                <TD>RESIDANCE</TD>
```

```
            <TD COLSPAN="4"><INPUT TYPE="TEXT" SIZE=50>
        </TR>

        <TR>
            <TH COLSPAN="6">EDUCATIONAL QUALIFICATION</TH>

        </TR>
        <TR>
            <TH>DEGREE</TH>
            <TH>UNIVERSITY</TH>
            <TH>MONTH&YEAR</TH>
            <TH>GRADE/MARKS</TH>
        </TR>
        <TR>
            <TH>1</TH>
            <TD COLSPAN="4"><INPUT TYPE="TEXT"  SIZE=70>
        </TR>
        <TR>
            <TH>2</TH>
            <TD COLSPAN="4"><INPUT TYPE="TEXT" SIZE=70>
        </TR>
        <TR>
            <TH>3</TH>
            <TD COLSPAN="4"><INPUT TYPE="TEXT"  SIZE=70>
        </TR>
        <TR>
            <TH>4</TH>
            <TD COLSPAN="4"><INPUT TYPE="TEXT" SIZE=70>
        </TR>
        <TR>
            <TH>5</TH>
            <TD COLSPAN="4"><INPUT TYPE="TEXT"  SIZE=70>
                    </TR>
      </TABLE>
    </FORM>
  </BODY>
</HTML>
```

OUTPUT:


biodata - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\sitams.INTERNET12.000\Desktop\biodata.html
Go Links
biodata
NAME
DOB
RELIGION
ADDRESS
street
town
dist
state
PHONE
office
residance
EDUCATIONAL QUALIFICATION
degree
university
month&year
grade/marks
1
2
3
4
5
Done
My Computer

CASCADING STYLE SHEET ILLUSTRATIONS:

Task 9.1

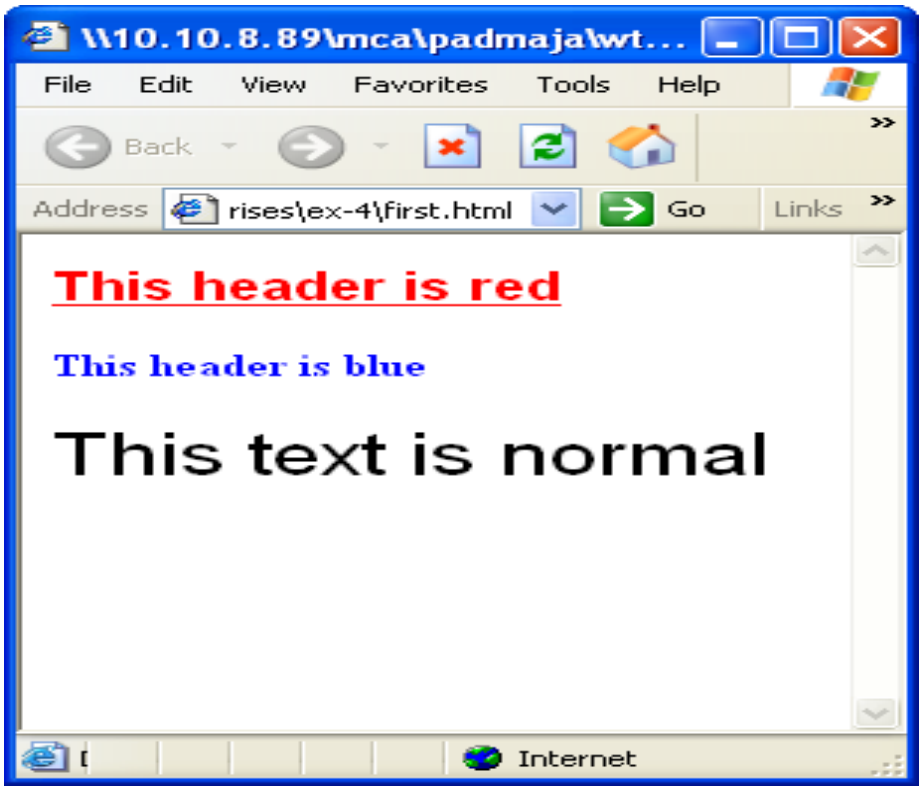
<!—USE OF DIFFERENT FONT,STYLES AND COLORS→

```
<html>
	<head>
		<link rel="stylesheet" type="text/css" href="test1.css"/>
	</head>
	<body>
		<h1>This header is red</h1>
		<h2>This header is blue</h2>
		<p>This text is normal</p>
	</body>
</html>
```

Test.css

```
h1{color:red;font-size:22px;font-family:arial;text-
decoration:underline} h2{color:blue;font-size:16px}
p{font-family:arial;font-size:30px}
```

OUTPUT:

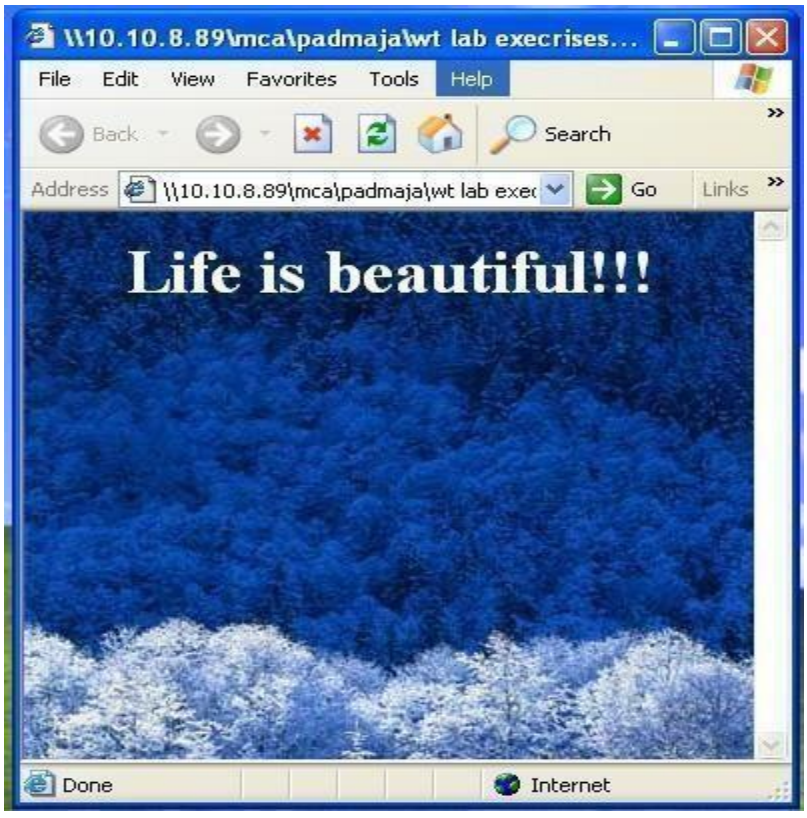


Task 9.2

<!—SETTING BACKGROUND IMAGE ..>

```
<html>
        <head>
                <style
                type="te
                xt/css">
                body
                {
                        background-
                        image:url("winter.jp
                        g"); background-
                        repeat:no-repeat
                }
                h1
                {
                    color:white;
                    font-size:35px;
                }
                </style>
        </head>
        <body>
                <center><h1>Life is beautiful!!!</h1></center>
        </body>
</html>
```

OUTPUT:



Task 9.3

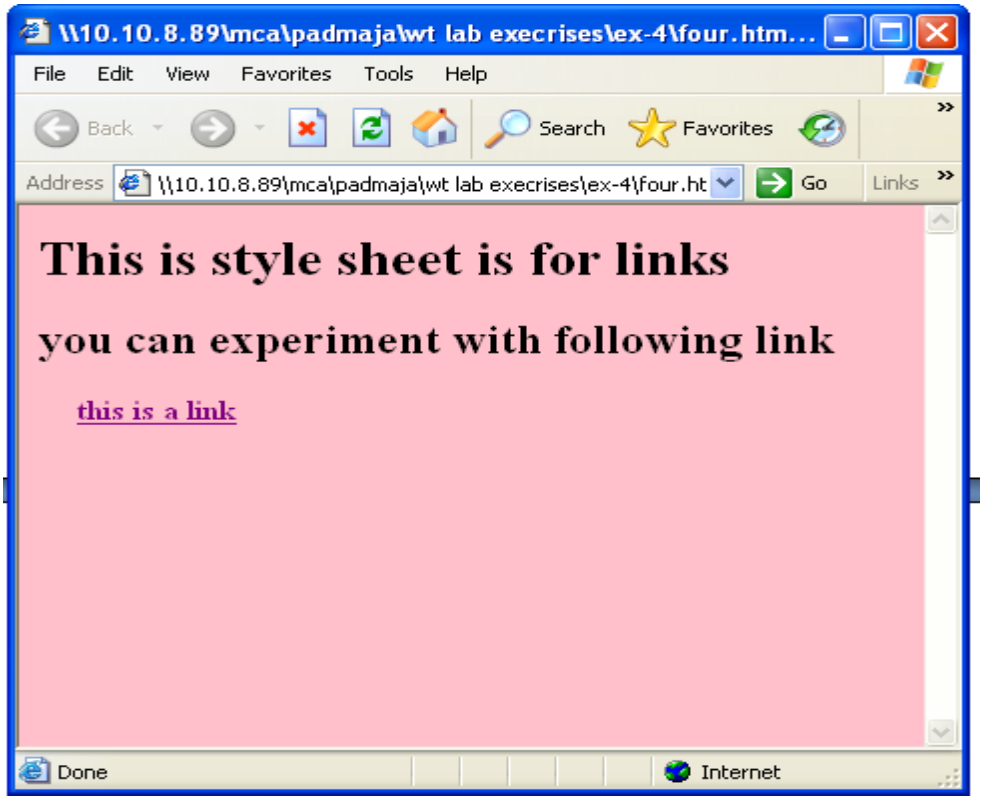
<!—DEFINING STYLES FOR LINKS..>

```
<html>
    <head>
        <link rel="stylesheet" type="text/css" href="test4.css"/>
    </head>
    <body>
        <h1>This is style sheet is for links</h1>
        <h2>you can experiment with following link</p>
        <p><a href="http://www.google.co.in">this is a link</a></p>
    </body>
</html>
```

test4.css

```
body{backgr
ound-
color:pink}
h1{color:blac
k;font-
size:22px}
p{font-
size:12}
a:link{color:b
lue}
a:visited{colo
r:purple}
a.hover{color:red;text-decoration:underline}
a.active{color:green}
```

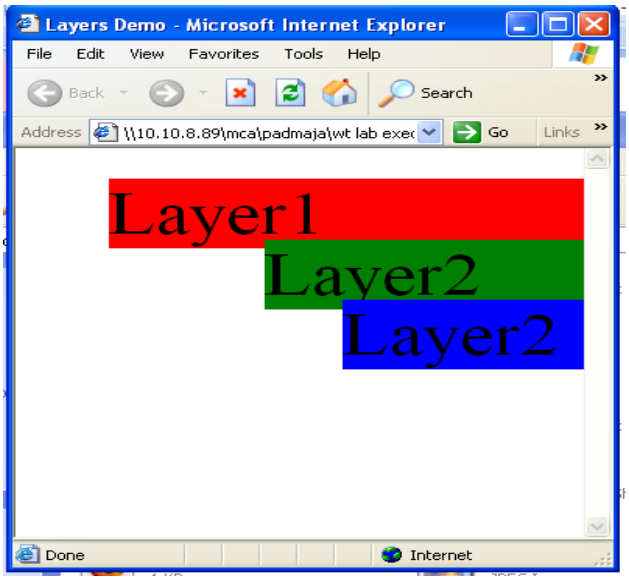
OUTPUT:



Task 9.4

```
<!— WORKING WITH LAYERS ..>
<html>
<head>
<title>Layers Demo</title>
</head>
<body>
<div style="position:relative;
      font-size:50px;
      left:50;
      top:10;
      Background-color:red;
      z-index:1;">Layer1</div>
<div style="position:relative;
      font-size:50px;
      left:150;
      top:3;
      Background-color:green;
      z-index:2;">Layer2</div>
<div style="position:relative;
      font-size:50px;
      left:200;
      top:-5;
      Background-color:blue;
      z-index:3;">Layer2</div>
</body>
</html>
```

OUTPUT:

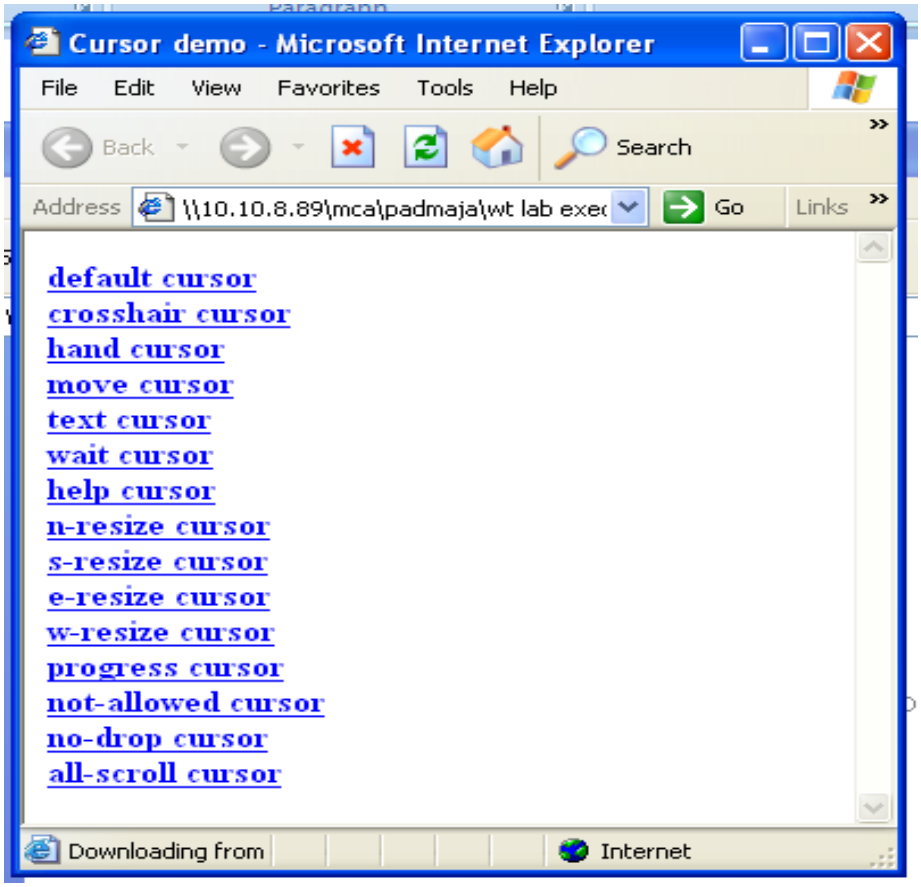


Task 9.5

<!— CUSTOMIZED CURSOR ..>

```
<html>
	<head>
		<title>Cursor demo</title>
		<style type="text/css">
		.link1{cursor:default}
		.link2{cursor:crosshair}
		.link3{cursor:hand}
		.link4{cursor:move}
		.link5{cursor:text}
		.link6{cursor:wait}
		.link7{cursor:help}
		.link8{cursor:n-resize}
		.link9{cursor:s-resize}
		.link10{cursor:e-resize}
		.link11{cursor:w-resize}
		.link12{cursor:progress}
		.link13{cursor:not-allowed}
		.link14{cursor:no-drop}
		.link15{cursor:all-scroll}
		</style>
	</head>
	<body>
		<b><a href="test.html" class="link1">default cursor</a><br/>
		<b><a href="test.html" class="link2">crosshair cursor</a> <br/>
		<b><a href="test.html" class="link3">hand cursor</a><br/>
		<b><a href="test.html" class="link4">move cursor</a><br/>
		<b><a href="test.html" class="link5">text cursor</a><br/>
		<b><a href="test.html" class="link6">wait cursor</a><br/>
		<b><a href="test.html" class="link7">help cursor</a><br/>
		<b><a href="test.html" class="link8">n-resize cursor</a><br/>
		<b><a href="test.html" class="link9">s-resize cursor</a><br/>
		<b><a href="test.html" class="link10">e-resize cursor</a><br/>
		<b><a href="test.html" class="link11">w-resize cursor</a><br/>
		<b><a href="test.html" class="link12">progress cursor</a><br/>
		<b><a href="test.html" class="link13">not-allowed cursor</a><br/>
		<b><a href="test.html" class="link14">no-drop cursor</a> <br/>
		<b><a href="test.html" class="link15">all-scroll cursor</a><br/>
	</body>
</html>
```

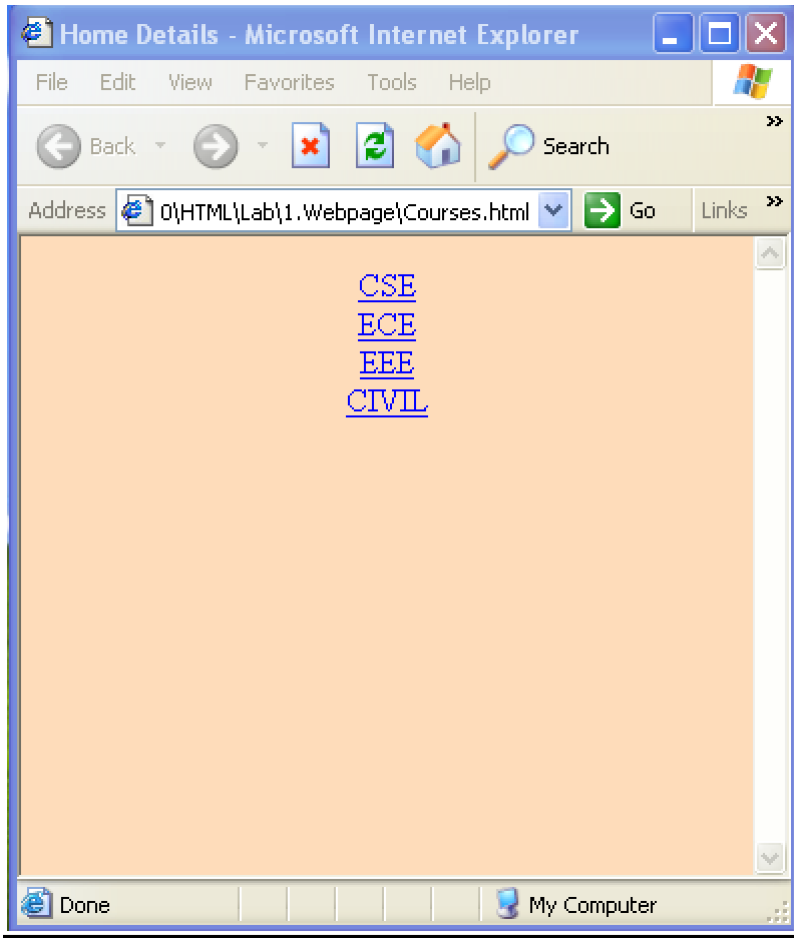
OUTPUT:


Cursor demo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search
Address \\10.10.8.89\mca\padmaja\wt lab exec Go Links
default cursor
crosshair cursor
hand cursor
move cursor
text cursor
wait cursor
help cursor
n-resize cursor
s-resize cursor
e-resize cursor
w-resize cursor
progress cursor
not-allowed cursor
no-drop cursor
all-scroll cursor
Downloading from
Internet

courses.html

```
<html>
    <head>
        <title>Home Details</title>
    </head>
    <body bgcolor="#fedcba">
        <center>
            <a href="Cse.html" target="Bottom Right">CSE</a><br>
            <a href="Ece.html" target="Bottom Right">ECE</a><br>
            <a href="Eee.html" target="Bottom Right">EEE</a><br>
      <a href="Civil.html" target="Bottom Right">CIVIL</a>
        </center>
    </body>
</html>
```

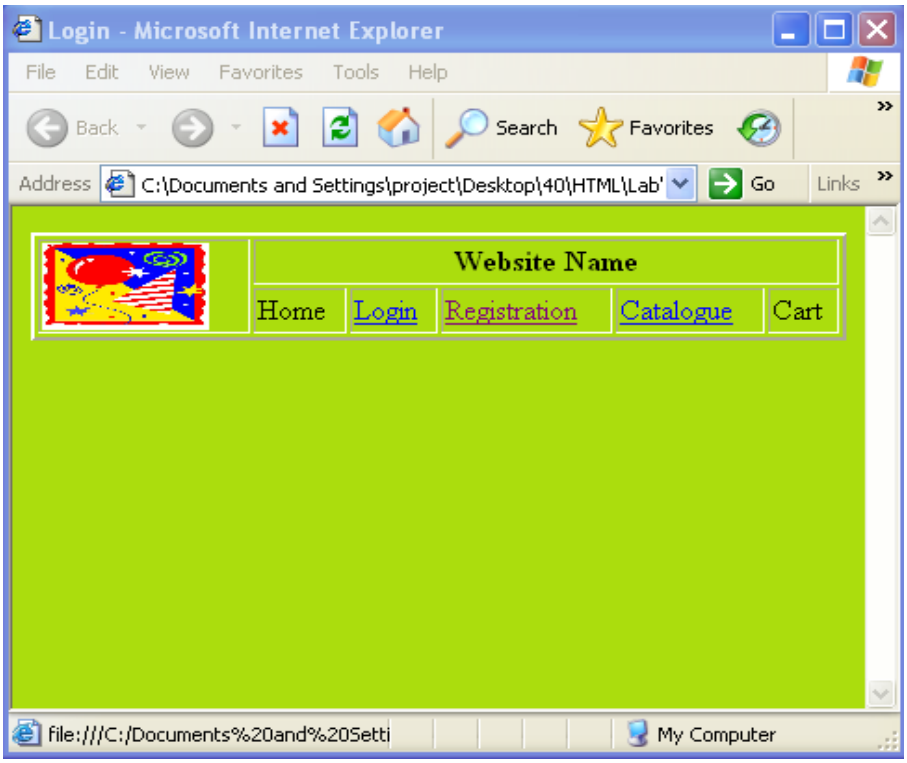
OUTPUT:



menu.html

```
<html>
	<head>
		<title>Login</title>
	 </head>
	<body bgcolor="#abdd0d">
		<table border="2" width="100%">
		<tr>
		    <td rowspan="2"><imgsrc="3.jpg" width=90 height=50></td>
		    <thcolspan="5">Website Name</th>
		 </tr>
		  <tr>
		      <td>Home</td>
		      <td><a href="Login.html" target="Bottom Right">Login</a></td>
		      <td><a href="Registration.html" target="Bottom
		          Right">Registration</a></td>
		      <td><a href="Catalogue.html" target="Bottom
		          Right">Catalogue</a></td>
		      <td>Cart</td>
		 </tr>
		</table>
	</body>
</html>
```

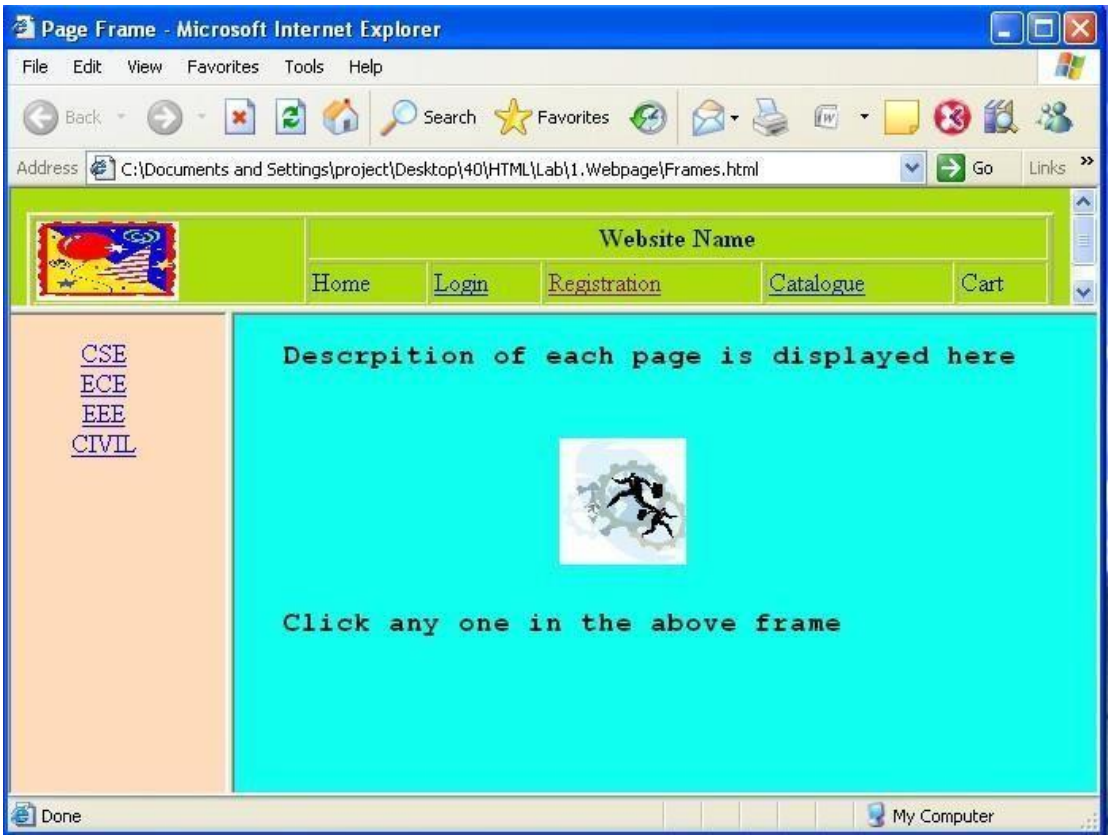
OUTPUT:



Frames.html

```
<html>
    <head>
        <title>Page Frame</title>
    </head>
    <frameset rows="20%,80%">
            <frame src="menu.html" name="Top">
            <frameset cols="20%,80%">
                <frame src="Courses.html" name="Bottom Left">
                <frame src="Description.html" name="Bottom Right">
            </frameset>
    </frameset>
</html>
```

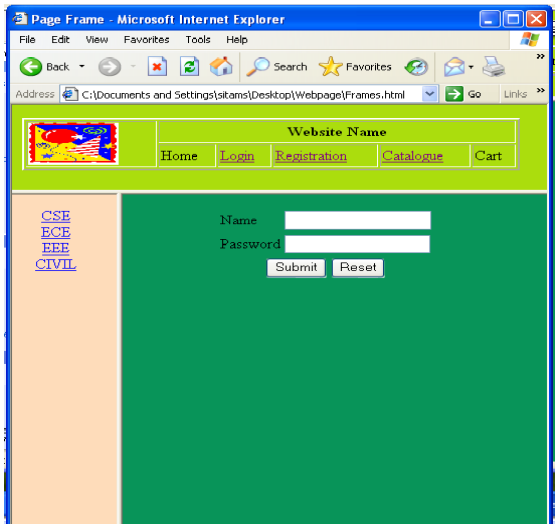
OUTPUT:



LOGIN.HTML

```
<html>
    <head>
        <title>Login</title>
    </head>
    <body bgcolor="#089459">
      <form name="login" Action:\cgi-bin\mycgi-pl"          method="post">
          <table align=center>
              <tr>
                  <td>Name</td>
                  <td><input type="text" name="tname"></td>
              </tr>
              <tr>
                  <td>Password</td>
                  <td><input type="Password" name="pass"></td>
              </tr>
              <tr>
                  <td colspan="2" align=center><input type="Submit"
                              name="Submit"
                  <input type="Reset" name="Reset" value="Reset">
              </tr>
          </table>
      </form>
  </body>
</html>
```

OUTPUT:



REGISTRATION FORM (with validation):

Registration.html

```
<html>
    <head>
        <script>
            function name_validate()
            {
            var
            name=document.forms[0].ele
            ments[0].value;
            name_re=/^[A-Z][A-Za-
            z]{6,}/g;
            if(!name.match(name_re))
            alert("please enter valid name");
            }

            function pwd_validate()
            {
            var
            passwd=document.forms[0].ele
            ments[1].value; pwd=/^[A-
            Z]\w{6,}/g;
            if(!passwd.matc
            h(pwd))
            alert("please
            enter valid
            password");
            }
            function mail_validate()
            {
            email=document.forms[0].elements[2].value;
            email_reg=/^[A-Za-z][A-Za-z0-9]+@[A-Za-z0-9,.-
            ]+\.[a-zA-Z]{3}/; if(!email.match(email_reg))
            alert("please enter valid mail id");
            }

            function ph_validate()
            {
            phone=document.forms[0].elements[
            3].value; ph_re=/\d{10}/;
            if(!phone.match(ph_re))
            alert("please enter valid phone number");
            }

        </script>
</head>
<body bgcolor="pink">
```

```
<form method="post">
<h2 align="center">Registration Form</h2>
<table border="0" width=100% align="center">
<tr>
<th>Name</th>
<td><input type="text" name="uname"
onBlur="name_validate()">
</td>
</tr>
<tr>
<th>Password</th>
<td><input type="password" name="pwd"
onBlur="pwd_validate()"></td>
</tr>
<tr>
<th>Email_ID</th>
<td><input type="text" name="eml"
onBlur="mail_validate()"></td>
</tr>
<tr>
<th>Phone Number</th>
<td><input type="text" name="pno"
onBlur="ph_validate()"></td>
</tr>
<tr>
<th>Gender</th>
<td><input type="radio" name="gender">Female
<input type="radio"
name="gender">Male</td>
</tr>
<tr>
<th>DOB</th>
<th>Date</th>
<td><select name="date">
<option>1</option>
<option>2</option>
<option>3</option>
<option>4</option>
<option>5</option>
<option>6</option>
<option>7</option>
<option>8</option>
<option>9</option>
<option>10</option>
<option>11</option>
<option>12</option>
<option>13</option>
<option>14</option>
<option>15</option>
<option>16</option>
```

```
                    <option>17</option>
                    <option>18</option>
                    <option>19</option>
                    <option>20</option>
                    <option>21</option>
                    <option>22</option>
                    <option>23</option>
                    <option>24</option>
                    <option>25</option>
                    <option>26</option>
                    <option>27</option>
                    <option>28</option>
                    <option>29</option>
                    <option>30</option>
                    <option>31</option>
                </select></td>
                <th>Month</th>
                <td><select name="month">
                    <option>jan</option>
                    <option>feb</option>
                    <option>mar</option>
                    <option>apr</option>
                    <option>may</option>
                    <option>jun</option>
                    <option>jul</option>
                    <option>aug</option>
                    <option>sep</option>
                    <option>oct</option>
                    <option>nov</option>
                    <option>dec</option>
                    </select></td>
                <th>Year</th>
                    <td><select name="year">
                    <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>
                </select></td>
            </tr>
```

```
<tr>
        <th>Languages Known</th>
        <td><input type="checkbox">Telugu</td>
        <td><input type="checkbox">Tamil</td>
        <td><input type="checkbox">English</td>
    </tr>
    <tr>
        <th>Address</th>
        <td><textarea name="address"
            rows="6"
            cols="10"></textarea></td>
    </tr>
    <tr>
        <td><input type="submit" value="Submit"></td>
        <td><input type="Reset" value="Reset"></td>
</tr>  </table></form></body></html>
```

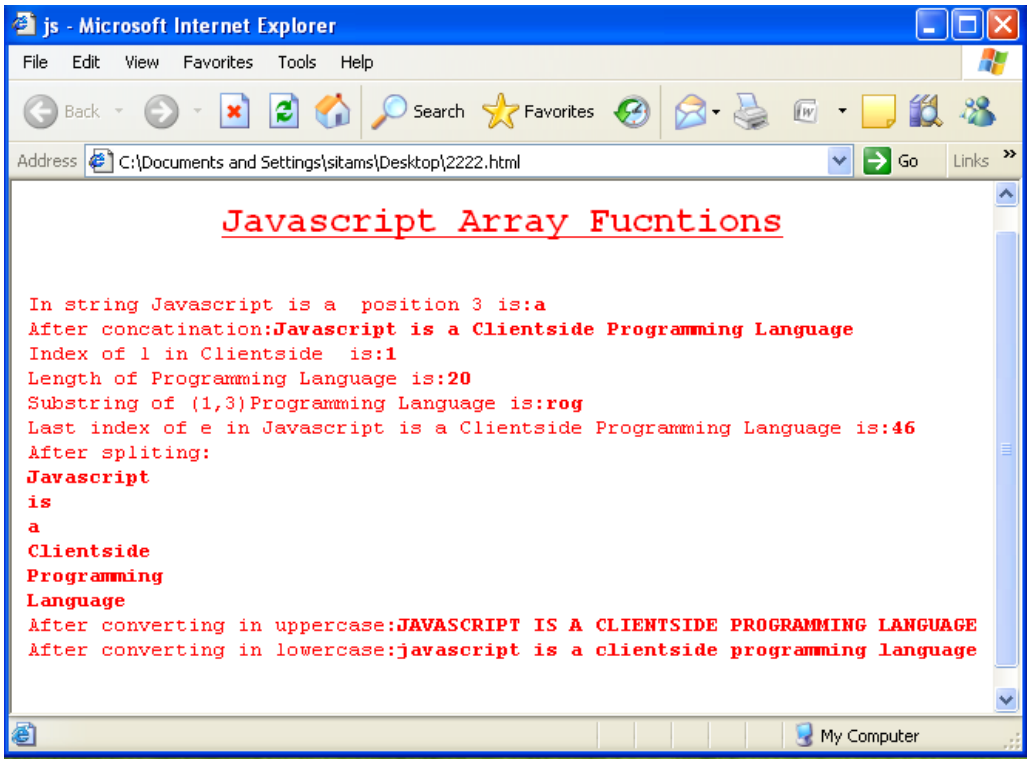
OUTPUT:


Page Frame - Windows Internet Explorer
E:\padmaja\wt_outputs\home.html
Live Search
File Edit View Favorites Tools Help
Page Frame
Page Tools
Website Name
Home Login Registration Catalogue Cart
CSE
ECE
EEE
CIVIL
Registration Form
Name asfd
Windows Internet Explorer
please enter valid name
OK
Month jan Year 2000
Languages Known
Telugu
Tamil
English
Address
Submit
Reset
My Computer
100%

JAVASCRIPT STRING MANIPULATION FUNCTIONS:

```
<html>
    <head>
        <title>String Manipulation</title>
    </head>
    <body text="red">
        <pre>
<h2 align="center"><u>Javascript Array Fucntions</u></h2>
        <script>
            var
            str=prompt("Enetr
            the string1"); var
            str2=prompt("Enetr
            the string2"); var
            str3=prompt("Enter
            the string3");
            var pos=prompt("Enter the position u want to
            display in string1 "); var res=str.charAt(pos);
            document.writeln("<br/>"+"In string "+str+" position
                    "+pos+" is:"+"<b>"+res+"</b>");
            document.writeln("AfterConcatination:"+"<b>"+str.conca
              t(str2,str3)+"</ b>");
            var i=str2.indexOf("l");
            document.writeln("<br>Index of l in "+str2+"
            is:<b>"+i+"</b>"); document.writeln("<br>Length of
            "+str3+" is:<b>"+str3.length+"</b>"); var
            st=str.concat(str2,str3);
            document.writeln("<br>Substring of (1,3)"+str3+"
                                    is:<b>"+str3.substr(1,3)+"</b>"
            ); document.writeln("<br>Last index of e in "+st+"
                                    is:<b>"+st.lastIndexOf("e")+"</b
            >"); document.writeln("<br>After spliting:");
            var stt=st.split(" ");
            for(var i=0;i<stt.length;i++)
            document.writeln("<br><b>"+stt[i]+"</b>");
            document.writeln("<br>After converting in
                                uppercase:<b>"+st.toUpperCase()+"<
            /b>"); document.writeln("<br>After converting in
                    lowercase:<b>"+st.toLowerCase()+"</b>");
        </script>
        </pre>
    </body>
</html>
```

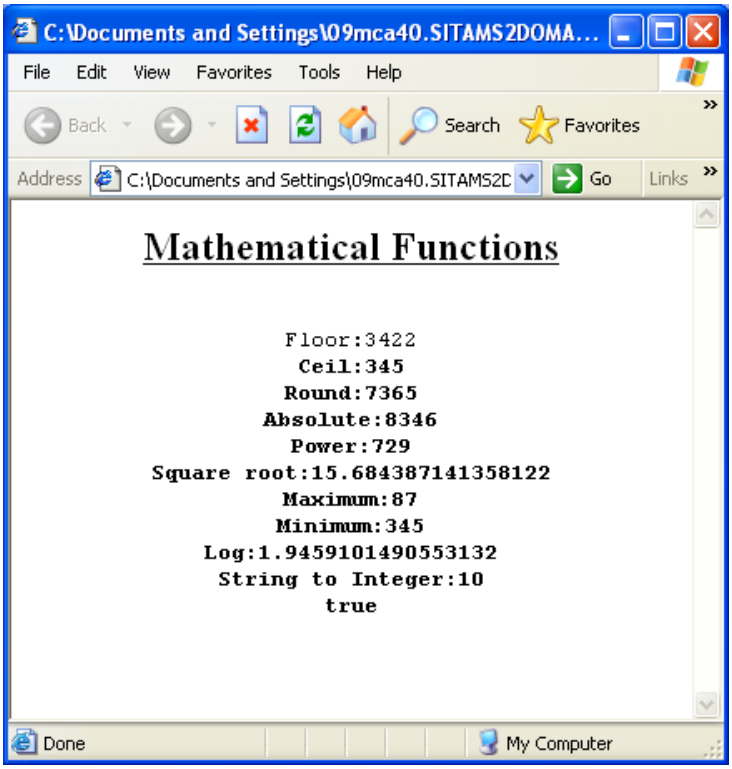
OUTPUT:


js - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\sitams\Desktop\2222.html
Go Links
Javascript Array Fucntions
In string Javascript is a position 3 is:a
After concatination:Javascript is a Clientside Programming Language
Index of l in Clientside is:1
Length of Programming Language is:20
Substring of (1,3)Programming Language is:rog
Last index of e in Javascript is a Clientside Programming Language is:46
After spliting:
Javascript
is
a
Clientside
Programming
Language
After converting in uppercase:JAVASCRIPT IS A CLIENTSIDE PROGRAMMING LANGUAGE
After converting in lowercase:javascript is a clientside programming language
My Computer

JAVASCRIPT MATHEMATICAL FUNCTIONS:

```
<html>
<body>
    <center>
        <h2><u>Mathematical Functions</u></h2>
    <pre>
        <script>
            document.writeln("<br/>"+"Floor:"+Mat
            h.floor(3422.74));
            document.writeln("<b>"+"Ceil:"+Math.c
            eil(344.45));
            document.writeln("<b>"+"Round:"+Mat
            h.round(7364.87));
            document.writeln("<b>"+"Absolute:"+M
            ath.abs(-8346));
            document.writeln("<b>"+"Power:"+Mat
            h.pow(3,6));
            document.writeln("<b>"+"Square
            root:"+Math.sqrt(246));
            document.writeln("<b>"+"Maximum:"+
            Math.max(74,87));
            document.writeln("<b>"+"Minimum:"+M
            ath.min(345,958));
            document.writeln("<b>"+"Log:"+Math.l
            og(7));
            var res=parseInt(1010,2);
            document.writeln("<b>"+"String
            to Integer:"+res); var
            res1=isNaN()
            document.writeln(res1);
        </script>
    </pre>
</center> </body> </html>
```

OUTPUT:

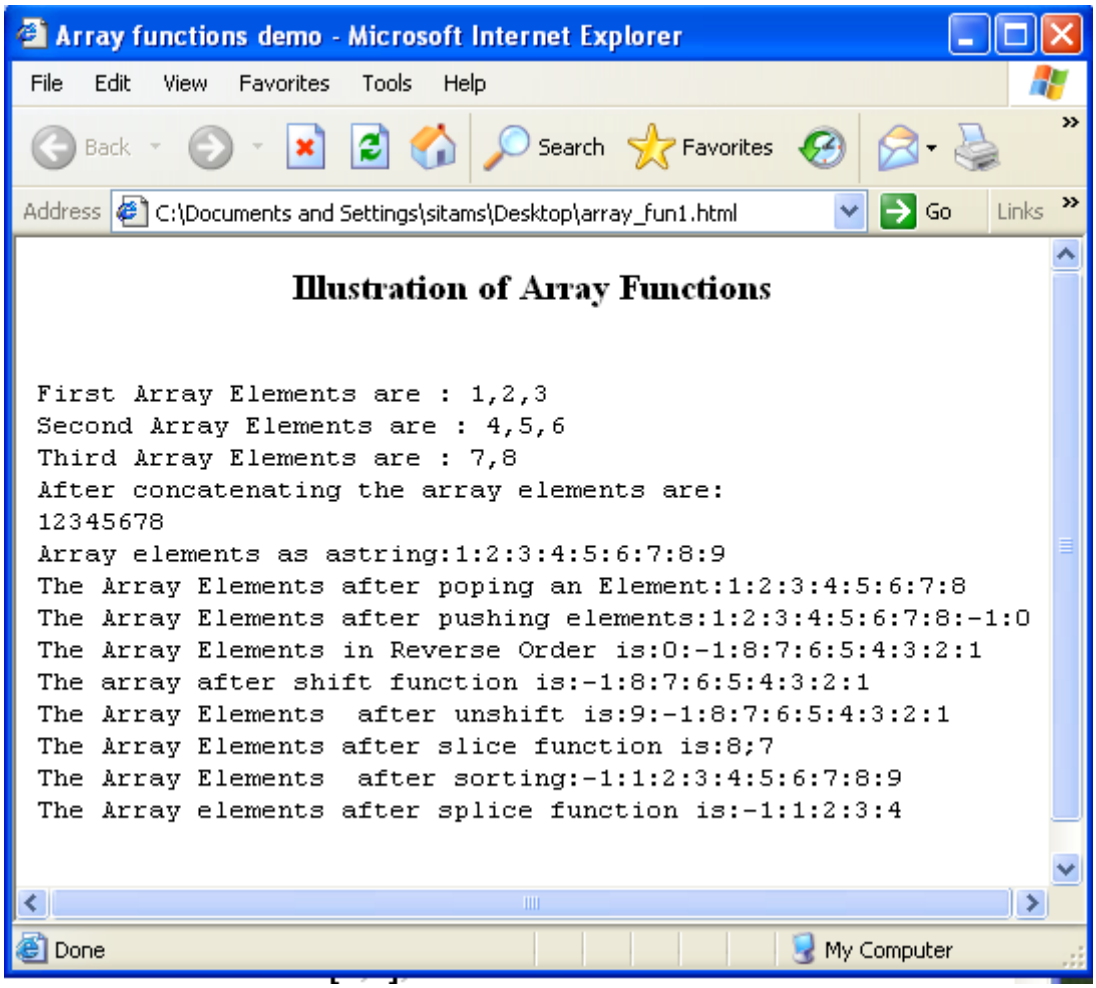

C:\Documents and Settings\09mca40.SITAMS2DOMA...
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\09mca40.SITAMS2D
Go Links
Mathematical Functions
Floor:3422
Ceil:345
Round:7365
Absolute:8346
Power:729
Square root:15.684387141358122
Maximum:87
Minimum:345
Log:1.9459101490553132
String to Integer:10
true
Done
My Computer

JAVASCRIPT ARRAY FUNCTIONS:

```
<html>
<head>
<title>Array functions demo</title>
</head>
<body>
<h3 align = "center" > Illustration of Array Functions </h3>
<pre>
<script>
//concat function: combines array elements var first=[1,2,3];
var second=[4,5,6]; var third=[7,8];
var res=first.concat(second,third); document.writeln("<br/>"+"First Array Elements
are : "
+first.join(",")); document.writeln("Second Array Elements are : "
+second.join(",")); document.writeln("Third Array Elements are : "
+third.join(","));
document.writeln("After concatenating the array elements are:"); for(var i = 0;i
<res.length;i++)
document.write(res[i]);
//Join function: combines array elements as string var a=[1,2,3,4,5,6,7,8,9];
var str=a.join(":"); document.writeln("<br/>"+"Array elements as
a string:"+str);//1:2:3:4:5:6:7:8:9
//pop function:Delete elements at the end of array a.pop();
var str1=a.join(":");
document.writeln("The Array Elements after poping an Element:" +
str1);//1:2:3:4:5:6:7:8
//push functon: Inserts an element at the end of array a.push(-1,0);
var str2=a.join(":");
document.writeln("The Array Elements after
pushing elements:" +str2);//-1:0:1:2:3:4:5:6:7:8
//reverse function: reverse the array a.reverse();
var str3=a.join(":");
document.writeln("The Array Elements in Reverse Order
is:"+str3);//8:7:6:5:4:3:2:1:0:-1
//shift function: Remove an element at the front of array a.shift();
var str4=a.join(":");
document.writeln("The array after shift function is:"+str4);//7:6:5:4:3:2:0:-1
//unshift function: Insert an element at the front of array
a.unshift(9);
var str5=a.join(":");
document.writeln("The Array Elements after
unshift is:"+str5);//9:7:6:5:4:3:2:1:0:-1
```

```
//slice function: Extract elements
var b=a.slice(2,4); var str6=b.join(";");
document.writeln("The Array Elements after slice function is:"+str6);//9:7:4:3:2:1:0:-
1
//sort function:sort elements in dictionary order a.sort();
var str7=a.join(":");
document.writeln("The Array Elements after sorting:"+str7);//-1:0:1:2:3:4:7:9
//splice function:insert elements into at specified position a.splice(5,10);
var str8=a.join(":");
document.writeln("The Array elements after splice function is:"+str8);
</script>
</pre>
</body>
</html>
```

OUTPUT:

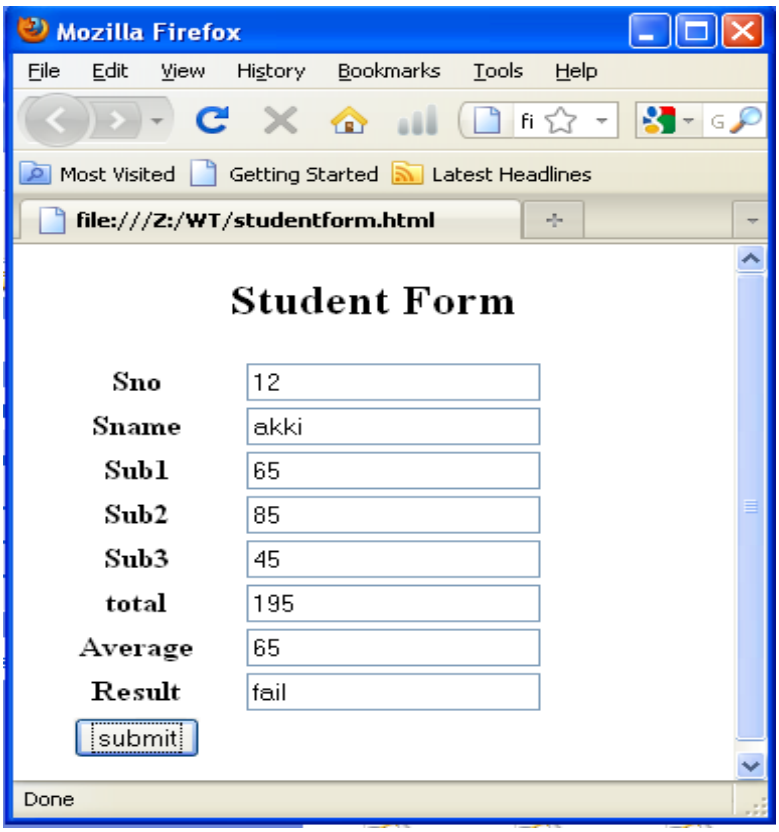

Array functions demo - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address C:\Documents and Settings\sitams\Desktop\array_fun1.html
Go Links
Illustration of Array Functions
First Array Elements are : 1,2,3
Second Array Elements are : 4,5,6
Third Array Elements are : 7,8
After concatenating the array elements are:
12345678
Array elements as astring:1:2:3:4:5:6:7:8:9
The Array Elements after poping an Element:1:2:3:4:5:6:7:8
The Array Elements after pushing elements:1:2:3:4:5:6:7:8:-1:0
The Array Elements in Reverse Order is:0:-1:8:7:6:5:4:3:2:1
The array after shift function is:-1:8:7:6:5:4:3:2:1
The Array Elements after unshift is:9:-1:8:7:6:5:4:3:2:1
The Array Elements after slice function is:8;7
The Array Elements after sorting:-1:1:2:3:4:5:6:7:8:9
The Array elements after splice function is:-1:1:2:3:4
Done
My Computer

A PROGRAM TO ILLUSTRATE JAVASCRIPT EVENTS:

```
<html>
    <head>
    <script>
    function compute()
    {
    var m1=document.forms[0].elements[2].value; var s1=parseInt(m1);
    var m2=document.forms[0].elements[3].value; var s2=parseInt(m2);
    var m3=document.forms[0].elements[4].value; var s3=parseInt(m3);
    var tot=s1+s2+s3; var avg=tot/3; var res;
    if((s1>=50)&&(s2>=50)&&(s3>=50))
    res="pass"; else res="fail";
    document.forms[0].elements[5].value=tot; document.forms[0].elements[6].value=avg;
    document.forms[0].elements[7].value=res;
    }
    </script>
    </head>
    <body>
    <form method="post">
    <center>
    <h2>Student Form</h2>
    <table border="0" width=100% align="center">
    <tr>
    <th>Sno</th>
    <td><input type="text" name="sno"></td>
    </tr>
    <tr>
    <th>Sname</th>
    <td><input type="text" name="sname"></td>
    </tr>
    <tr>
    <th>Sub1</th>
    <td><input type="text" name="s1"></td>
    </tr>
    <tr>
    <th>Sub2</th>
    <td><input type="text" name="s2"></td>
    </tr>
    <tr>
    <th>Sub3</th>
    <td><input type="text" name="s3"></td>
```

```
</tr>
<tr>
<th>total</th>
<td><input type="text" name="tot"></td>
</tr>
<tr>
<th>Average</th>
<td><input type="text" name="avg"></td>
</tr>
<tr>
<th>Result</th>
<td><input type="text" name="res"></td>
</tr>
<tr>
<th align="center">
<input type="button" value="submit" onClick="compute()">
</th>
</tr>
</table>
</center>
</form>
</body>
</html>
```

OUTPUT:

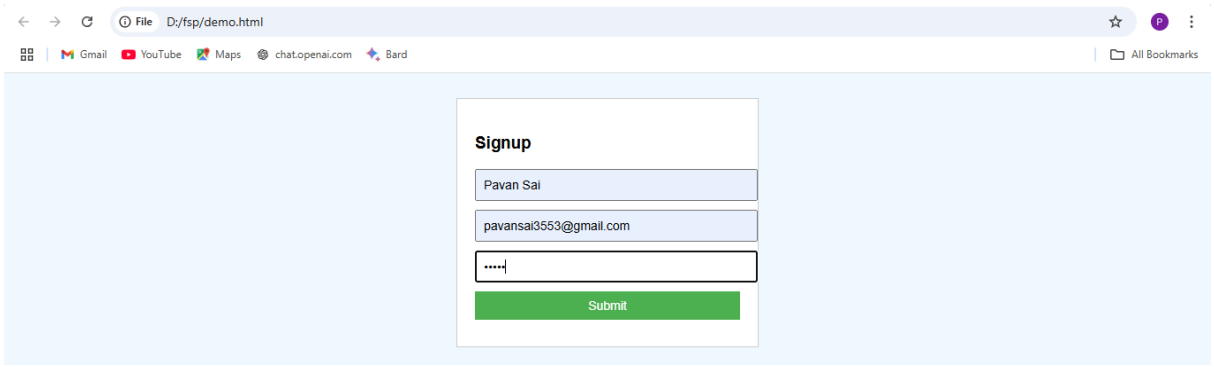

Mozilla Firefox
File Edit View History Bookmarks Tools Help
Most Visited Getting Started Latest Headlines
file:///Z:/WT/studentform.html
Student Form
Sno 12
Sname akki
Sub1 65
Sub2 85
Sub3 45
total 195
Average 65
Result fail
submit
Done

```
<!DOCTYPE html>
<html>
<head>
  <title>Basic Form</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f0f8ff;
      padding: 20px;
    }
    form {
      background-color: white;
      padding: 20px;
      border: 1px solid #ccc;
      max-width: 300px;
      margin: auto;
    }
    input, button {
      display: block;
      width: 100%;
      margin: 10px 0;
      padding: 8px;
      font-size: 14px;
    }
    button {
      background-color: #4CAF50;
      color: white;
      border: none;
    }
    .error {
      color: red;
      font-size: 13px;
      text-align: center;
    }
  </style>
</head>
<body>

  <form id="basicForm">
    <h3>Signup</h3>
    <input type="text" id="name" placeholder="Name">
    <input type="email" id="email" placeholder="Email">
    <input type="password" id="password" placeholder="Password">
```

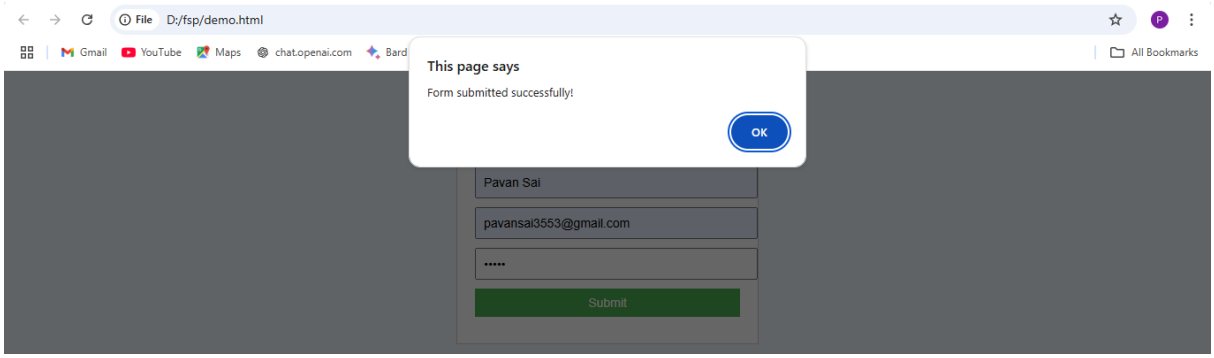
```
    <button type="submit">Submit</button>
    <div class="error" id="errorMsg"></div>
  </form>

  <script>
    document.getElementById("basicForm").onsubmit = function(e) {
      e.preventDefault();
      const name = document.getElementById("name").value.trim();
      const email = document.getElementById("email").value.trim();
      const pass = document.getElementById("password").value.trim();
      const error = document.getElementById("errorMsg");

      if (!name || !email || !pass) {
        error.textContent = "All fields are required.";
      } else {
        error.textContent = "";
        alert("Form submitted successfully!");
        this.reset();
      }
    };
  </script>
 </body>
</html>
```

OUTPUT:

ENTERING DETAILS FOR “SIGNUP”



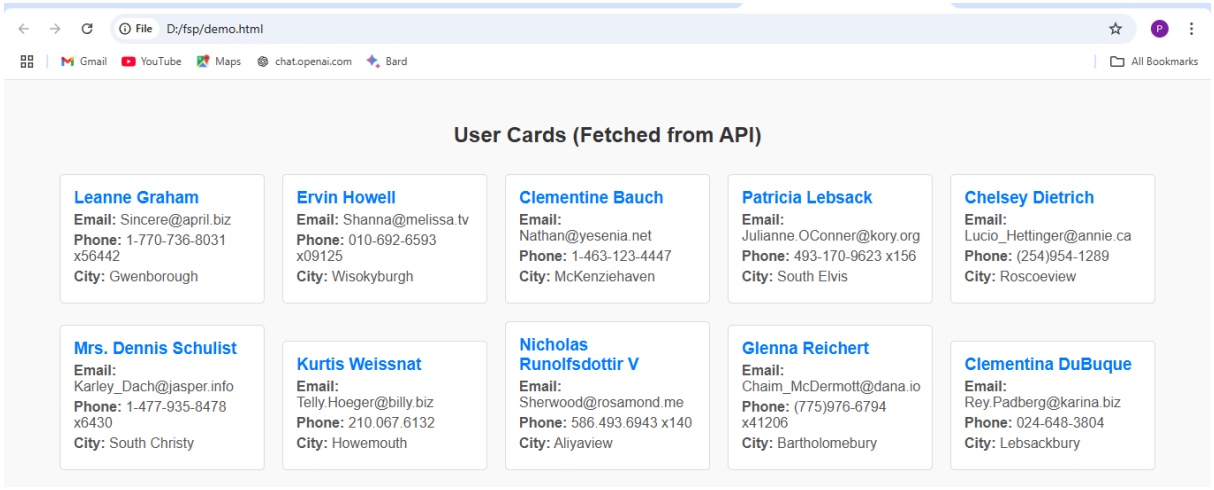
AFTER CLICKING SUBMIT, FORM SUBMITTED SUCCESSFULLY!



```
<!DOCTYPE html>
<html>
<head>
  <title>Fetch API</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f9f9f9;
      padding: 20px;
      text-align: center;
    }

    h2 {
      color: #333;
    }

    .card {
      background-color: #fff;
      padding: 15px;
      margin: 10px;
      border: 1px solid #ddd;
      border-radius: 5px;
      width: 200px;
      display: inline-block;
      text-align: left;
    }

    .card h3 {
      color: #007BFF;
      margin: 0;
    }

    .card p {
      margin: 5px 0;
      color: #555;
    }
  </style>
</head>
<body>

  <h2>User Cards (Fetched from API)</h2>
  <div id="cardsContainer"></div>
```

```
  <script>
    fetch('https://jsonplaceholder.typicode.com/users')
      .then(response => response.json())
      .then(users => {
        const container = document.getElementById('cardsContainer');
        users.forEach(user => {
          const card = document.createElement('div');
          card.className = 'card';
          card.innerHTML = `
            <h3>${user.name}</h3>
            <p><strong>Email:</strong> ${user.email}</p>
            <p><strong>Phone:</strong> ${user.phone}</p>
            <p><strong>City:</strong> ${user.address.city}</p>
          `;
          container.appendChild(card);
        });
      })
      .catch(error => {
        document.getElementById('cardsContainer').innerHTML = `<p
style="color:red;">Error fetching data: ${error}</p>`;
      });
  </script>
</body>
</html>
```

OUTPUT:


File D:/fsp/demo.html
Gmail YouTube Maps chat.openai.com Bard
All Bookmarks
User Cards (Fetched from API)
Leanne Graham
Email: Sincere@april.biz
Phone: 1-770-736-8031 x56442
City: Gwenborough
Ervin Howell
Email: Shanna@melissa.tv
Phone: 010-692-6593 x09125
City: Wisokyburgh
Clementine Bauch
Email: Nathan@yesenia.net
Phone: 1-463-123-4447
City: McKenziehaven
Patricia Lebsack
Email: Julianne.OConner@kory.org
Phone: 493-170-9623 x156
City: South Elvis
Chelsey Dietrich
Email: Lucio_Hettinger@annie.ca
Phone: (254)954-1289
City: Roscoeview
Mrs. Dennis Schulist
Email: Karley_Dach@jasper.info
Phone: 1-477-935-8478 x6430
City: South Christy
Kurtis Weissnat
Email: Telly.Hoeger@billy.biz
Phone: 210.067.6132
City: Howemouth
Nicholas Runolfsdottir V
Email: Sherwood@rosamond.me
Phone: 586.493.6943 x140
City: Aliyaview
Glenna Reichert
Email: Chaim_McDermott@dana.io
Phone: (775)976-6794 x41206
City: Bartholomebury
Clementina DuBuque
Email: Rey.Padberg@karina.biz
Phone: 024-648-3804
City: Lebsackbury

index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Static Server</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <header>
        <h1>Welcome to My Simple Static Server</h1>
    </header>

    <section>
        <p>This is a basic page served using Node.js without Express.</p>
        <p>Enjoy exploring!</p>
    </section>

    <footer>
        <p>Created by PS</p>
    </footer>
</body>
</html>
```

style.css

```
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f4f4f4;
}

header {
    background-color: #333;
    color: white;
    padding: 10px 0;
    text-align: center;
}

section {
    padding: 20px;
```

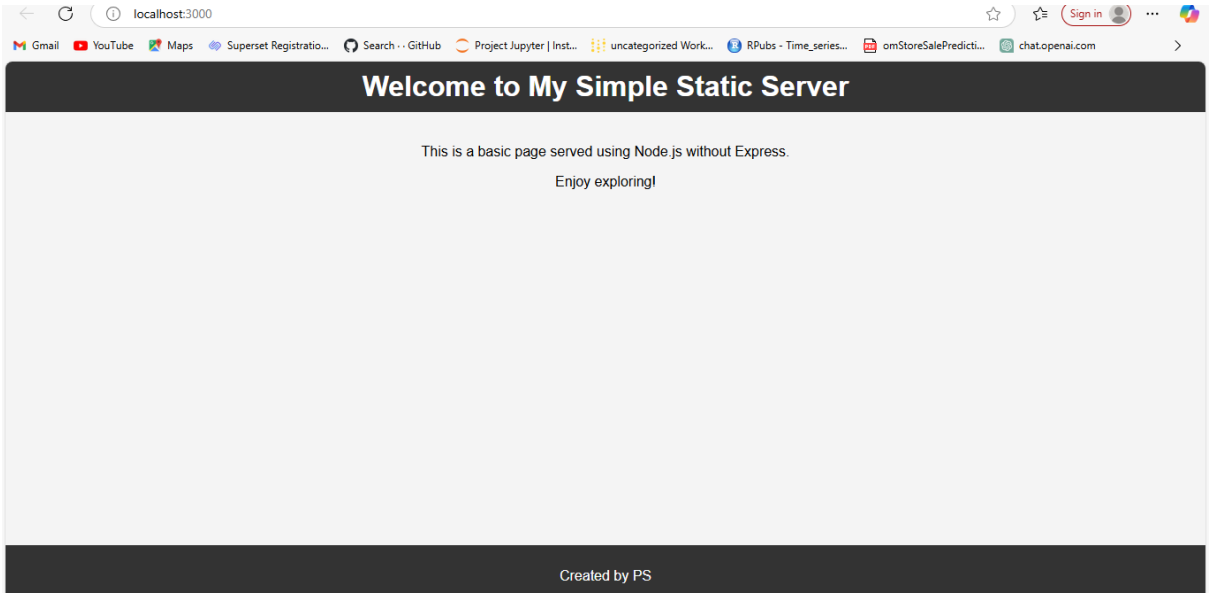
```
    text-align: center;
}

footer {
    background-color: #333;
    color: white;
    text-align: center;
    padding: 10px 0;
    position: fixed;
    width: 100%;
    bottom: 0;
}

h1 {
    margin: 0;
}
```

server.js

```
const http = require('http');
const fs = require('fs');
const path = require('path');

// Create a server
http.createServer((req, res) => {
    // Set the file path based on the URL
    let filePath = '.' + req.url;
    if (filePath == './') filePath = './index.html'; // Default to index.html

    // Get the file extension
    const extname = path.extname(filePath);
    let contentType = 'text/html'; // Default to HTML

    // Determine the content type based on the file extension
    if (extname == '.css') {
        contentType = 'text/css';
    }

    // Read and serve the file
    fs.readFile(filePath, (err, data) => {
        if (err) {
            res.writeHead(404, { 'Content-Type': 'text/plain' });
            res.end('404 Not Found');
        } else {
```

```
            res.writeHead(200, { 'Content-Type': contentType });
            res.end(data);
        }
    });
}).listen(3000, () => {
    console.log('Server running at http://localhost:3000');
});
```

OUTPUT:


localhost:3000
Sign in
Gmail
YouTube
Maps
Superset Registratio...
Search · · GitHub
Project Jupyter | Inst...
uncategorized Work...
RPubs - Time_series...
omStoreSalePredicti...
chat.openai.com
Welcome to My Simple Static Server
This is a basic page served using Node.js without Express.
Enjoy exploring!
Created by PS

server.js

```
const express = require('express');
const fs = require('fs');
const exphbs = require('express-handlebars');

const app = express();

app.engine('hbs', exphbs.create({ extname: 'hbs', defaultLayout: false }).engine);
app.set('view engine', 'hbs');
app.use(express.urlencoded({ extended: true }));

app.get('/', (req, res) => res.render('form'));

app.post('/submit', (req, res) => {
  fs.writeFileSync('data.json', JSON.stringify(req.body));
  res.redirect('/data');
});

app.get('/data', (req, res) =>
  res.render('data', { data: JSON.parse(fs.readFileSync('data.json')) })
);

app.listen(3000, () => console.log('http://localhost:3000'));
```

views/form.hbs (the form page)

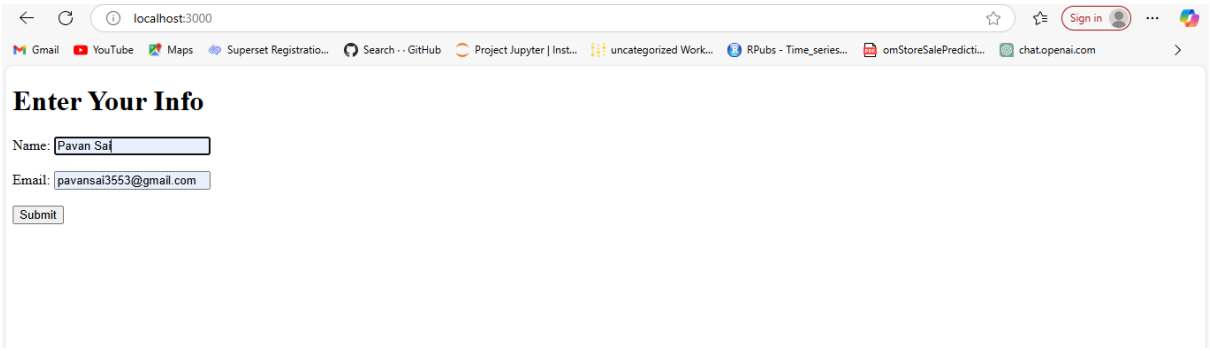
```
<!DOCTYPE html>
<html>
<head>
    <title>Form Page</title>
</head>
<body>
    <h1>Enter Your Info</h1>
    <form action="/submit" method="POST">
        <label>Name: <input type="text" name="name" required></label><br><br>
        <label>Email: <input type="email" name="email" required></label><br><br>
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

views/data.hbs (the redirect page that shows saved data)

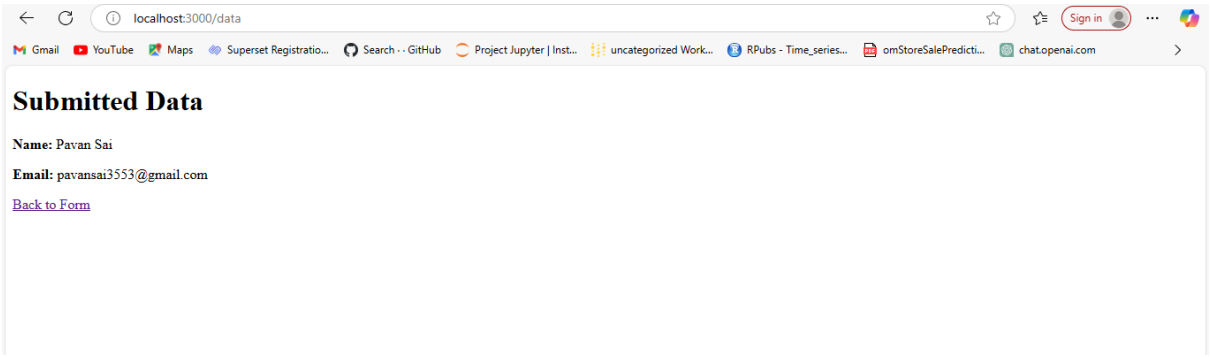
```
<!DOCTYPE html>
<html>
<head>
    <title>Data Page</title>
</head>
<body>
    <h1>Submitted Data</h1>
    <p><strong>Name:</strong> {{data.name}}</p>
    <p><strong>Email:</strong> {{data.email}}</p>
    <a href="/">Back to Form</a>
</body>
</html>
```

OUTPUT:

ENTERING DETAILS



AFTER SUBMITTING FORM, REDIRECT TO ANOTHER PAGE AND DISPLAY THE SUBMITTED DATA



server.js (Main server file)

```
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');
const Student = require('./models/student');

const app = express();
mongoose.connect('mongodb://localhost/studentDB');

app.use(bodyParser.urlencoded({ extended: true }));
app.set('view engine', 'ejs');

// Routes
app.route('/')
  .get(async (req, res) => res.render('index', { students: await Student.find() }));

app.route('/add')
  .get((req, res) => res.render('form'))
  .post(async (req, res) => {
    await new Student(req.body).save();
    res.redirect('/');
  });

app.route('/edit/:id')
  .get(async (req, res) => res.render('edit', { student: await Student.findById(req.params.id)
}))
```

```
  .post(async (req, res) => {
    await Student.findByIdAndUpdate(req.params.id, req.body);
    res.redirect('/');
  });

app.get('/delete/:id', async (req, res) => {
  await Student.findByIdAndDelete(req.params.id);
  res.redirect('/');
});

app.listen(3000, () => console.log('Server running on http://localhost:3000'));
```

models/student.js (MongoDB Schema for Student)

```
const mongoose = require('mongoose');

const studentSchema = new mongoose.Schema({
  name: String,
  age: Number,
  grade: String
});

module.exports = mongoose.model('Student', studentSchema);
```

views/form.ejs (Form to Add Student)

```
<!DOCTYPE html>
<html>
<head>
  <title>Add Student</title>
</head>
<body>
  <h1>Add Student</h1>
  <form action="/add" method="POST">
    <input type="text" name="name" placeholder="Name" required><br>
    <input type="number" name="age" placeholder="Age" required><br>
    <input type="text" name="grade" placeholder="Grade" required><br>
    <button type="submit">Submit</button>
  </form>
  <a href="/">Back to Students List</a>
</body>
</html>
```

views/edit.ejs (Form to Edit Student)

```
<!DOCTYPE html>
<html>
<head>
  <title>Edit Student</title>
</head>
<body>
  <h1>Edit Student</h1>
```

```
  <form action="/edit/<%= student._id %>" method="POST">
    <input type="text" name="name" value="<%= student.name %>" required><br>
    <input type="number" name="age" value="<%= student.age %>" required><br>
    <input type="text" name="grade" value="<%= student.grade %>" required><br>
    <button type="submit">Update</button>
  </form>
  <a href="/">Back to Students List</a>
</body>
</html>
```

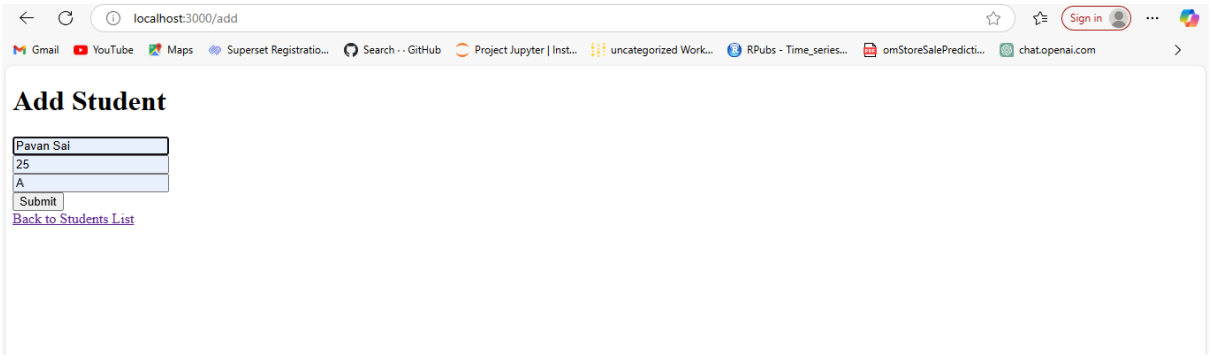
views/index.ejs (List of Students)

```
<!DOCTYPE html>
<html>
<head>
  <title>Student List</title>
</head>
<body>
  <h1>Student List</h1>
  <a href="/add">Add Student</a><br><br>
  <table border="1">
    <tr>
      <th>Name</th>
      <th>Age</th>
      <th>Grade</th>
      <th>Actions</th>
    </tr>
    <% students.forEach(student => { %>
```

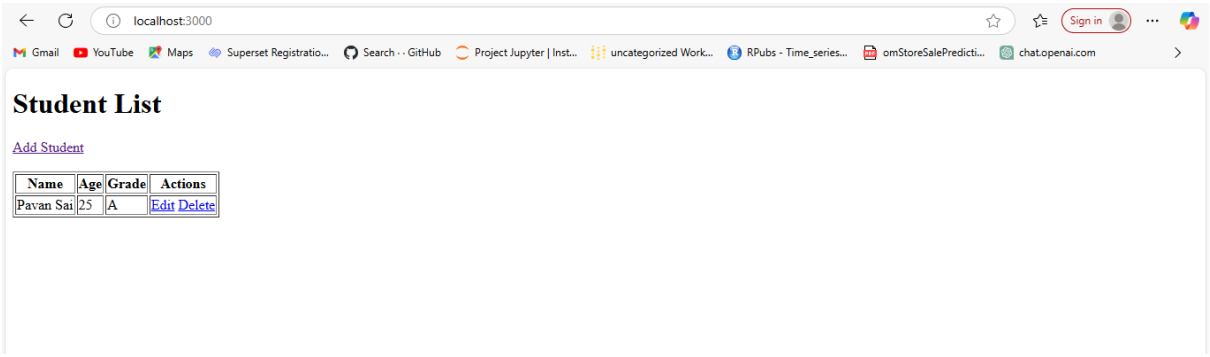
```
    <tr>
      <td><%= student.name %></td>
      <td><%= student.age %></td>
      <td><%= student.grade %></td>
      <td>
        <a href="/edit/<%= student._id %>">Edit</a>
        <a href="/delete/<%= student._id %>">Delete</a>
      </td>
    </tr>
    <% }); %>
  </table>
</body>
</html>
```

OUTPUT:

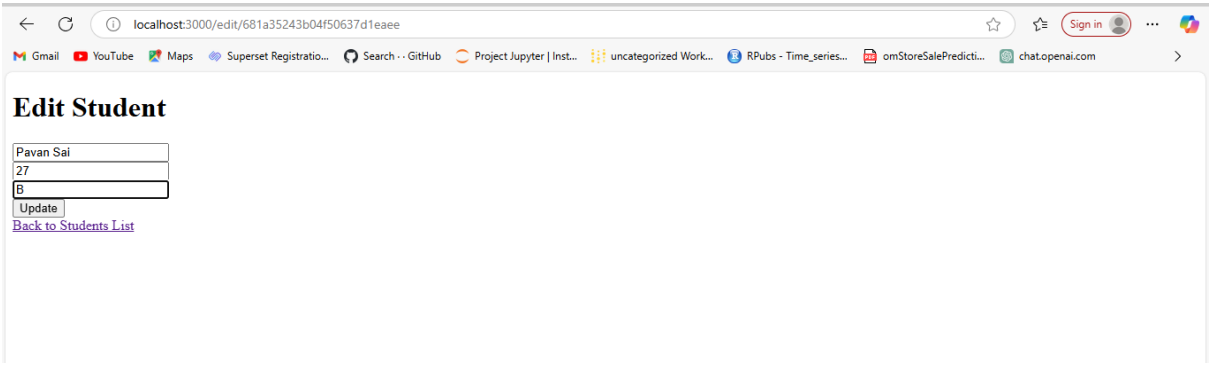
ADD STUDENT DATA



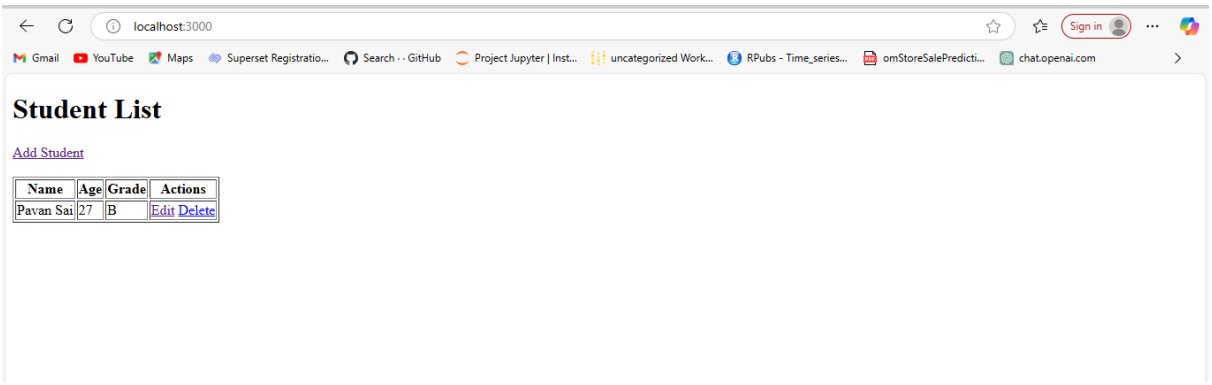
VIEWING STUDENT LIST



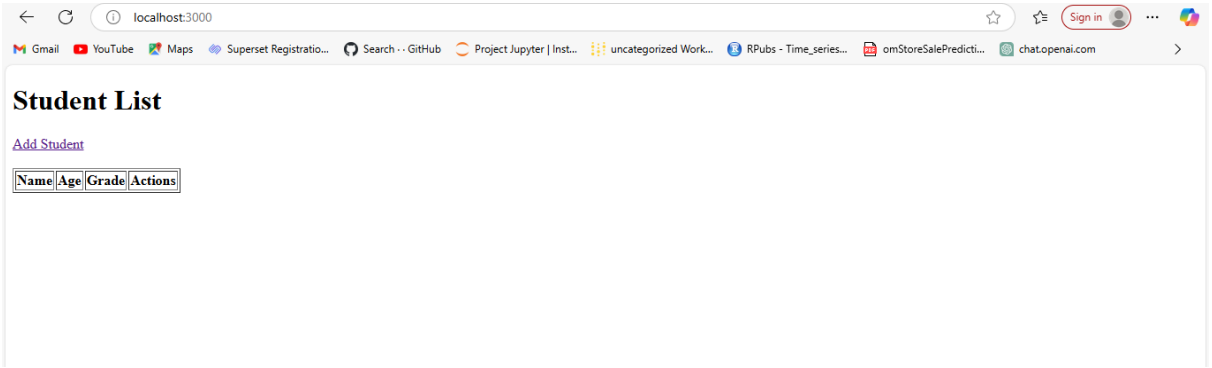
UPDATE STUDENT DATA



AFTER UPDATION VIEWING STUDENT LIST



AFTER DELETION VIEWING STUDENT LIST



index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>React Count App</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

App.js

```
import React, { useState } from "react";

function App() {
  const [count, setCount] = useState(0);

  return (
    <div className="app">
      <h1>Simple Count App</h1>
      <div className="counter">{count}</div>
      <div className="buttons">
        <button onClick={() => setCount(count - 1)}>-</button>
        <button onClick={() => setCount(0)}>Reset</button>
        <button onClick={() => setCount(count + 1)}>+</button>
      </div>
    </div>
  );
}
export default App;
```

App.css

```
body {
  margin: 0;
  padding: 0;
  font-family: Arial, sans-serif;
  background-color: #f7f9fc;
  display: flex;
  justify-content: center;
  align-items: center;
```

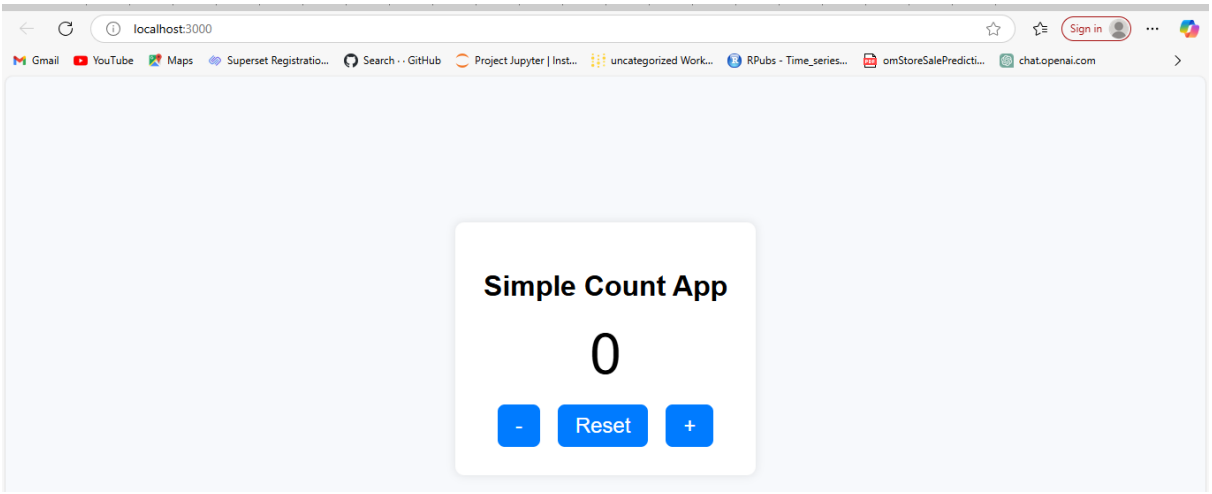
```
  height: 100vh;
}

.app {
  text-align: center;
  background: white;
  padding: 2rem;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

.counter {
  font-size: 4rem;
  margin: 20px 0;
}

.buttons button {
  font-size: 1.5rem;
  padding: 10px 20px;
  margin: 0 10px;
  border: none;
  border-radius: 8px;
  cursor: pointer;
  background-color: #007bff;
  color: white;
  transition: background-color 0.2s ease;
}

.buttons button:hover {
  background-color: #0056b3;
}
```

index.js

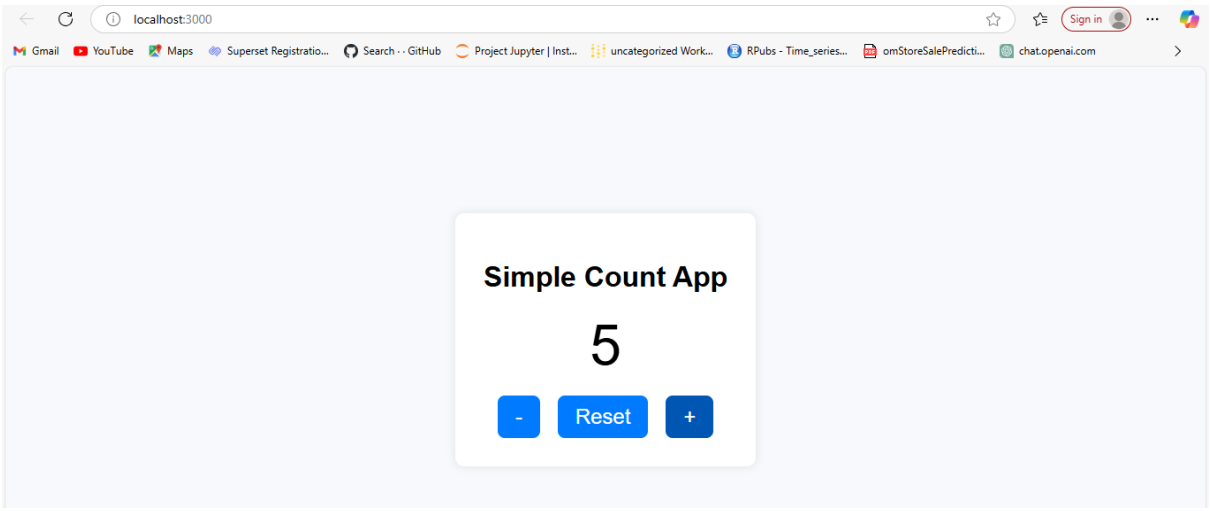
```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App.js";
import "./App.css";
const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

OUTPUT:

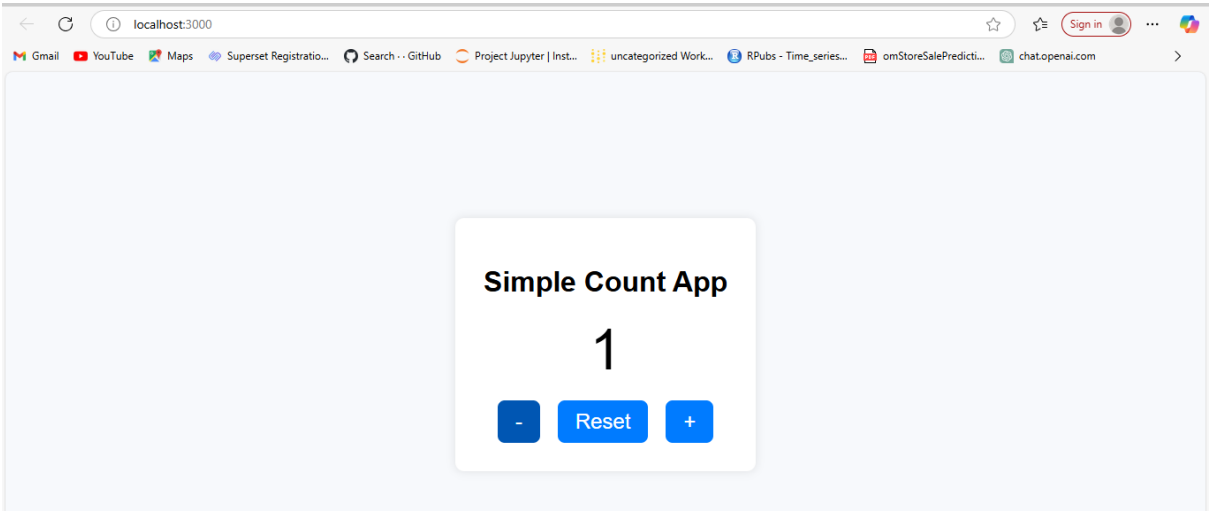
VIEWING COUNT APP



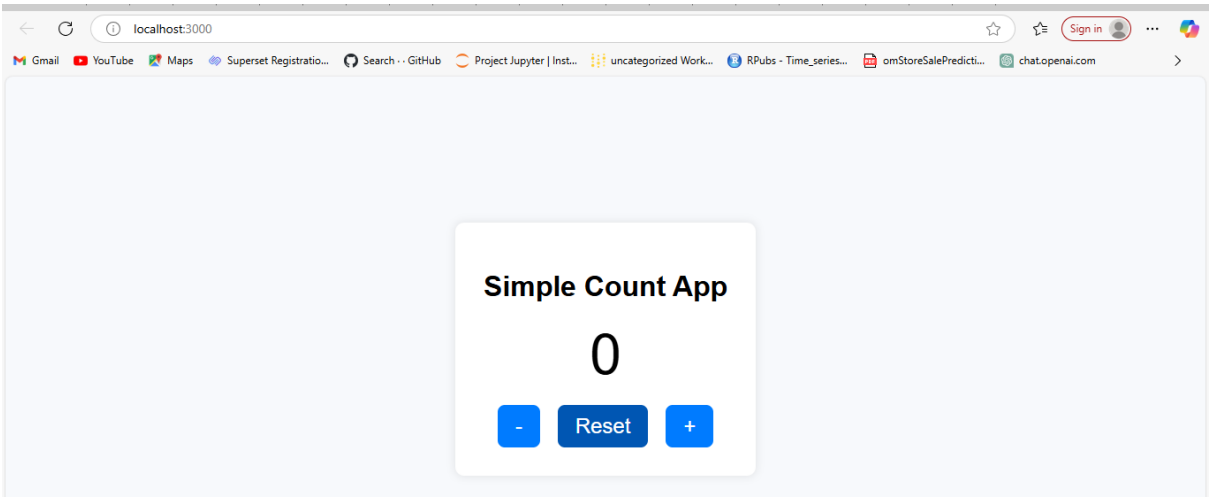
INCREASED COUNT VALUE FROM "0" TO "5"



DECREASED COUNT VALUE FROM “5” TO “1”



AFTER RESETING THE COUNT APP



Frontend Code

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Todo App</title>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

App.js

```
import { useEffect, useState } from 'react';
import { v4 as uuid } from 'uuid';
export default function App() {
  const [todos, setTodos] = useState([]);
  const [task, setTask] = useState('');

  // Fetch todos from the backend when the component mounts
```

```
useEffect(() => {
  fetch('http://localhost:5000/todos')
    .then((res) => res.json())
    .then(setTodos);
}, []);


const add = async () => {
  if (!task) return;


  // Post the new task to the backend
  await fetch('http://localhost:5000/todos', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ id: uuid(), task }),
  });


  // Reset the input field and update the todo list
  setTask('');
  const res = await fetch('http://localhost:5000/todos');
  setTodos(await res.json());
};
const del = async (id) => {
  // Delete the task from the backend
```

```
    await fetch(`http://localhost:5000/todos/${id}`, { method: 'DELETE' });
    setTodos(todos.filter((t) => t.id !== id));
  };
  return (
    <div style={{ textAlign: 'center', marginTop: 30 }}>
      <h2>Todo App</h2>
      <input
        value={task}
        onChange={(e) => setTask(e.target.value)}
        placeholder="Enter task"
      />
      <button onClick={add}>Add</button>
      <ul>
        {todos.map((t) => (
          <li key={t.id}>
            {t.task}
            <button onClick={() => del(t.id)}>✕</button>
          </li>
        ))}
      </ul>
    </div>
  );
}
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client'; // Import the new API for React 18
import App from './App';
import './style.css';
const root = ReactDOM.createRoot(document.getElementById('root')); // Create the root
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

style.css

```
/* Basic styling for the app */
body {
    font-family: Arial, sans-serif;
    background-color: #f0f4f8;
    margin: 0;
    padding: 0;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
    color: #333;
  }
```

```
/* Container for the Todo app */
div {
  background-color: white;
  border-radius: 10px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  padding: 20px;
  width: 300px;
  text-align: center;
}

/* Heading styling */
h2 {
  color: #4CAF50;
  font-size: 2rem;
  margin-bottom: 15px;
}

/* Styling for input field */
input {
  padding: 8px;
  width: 80%;
  margin-bottom: 15px;
  border: 1px solid #ccc;
  border-radius: 5px;
  font-size: 1rem;
}
```

```
/* Add button styling */
button {
  padding: 8px 16px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  font-size: 1rem;
  transition: background-color 0.3s ease;
}

/* Button hover effect */
button:hover {
  background-color: #45a049;
}

/* List styling */
ul {
  list-style-type: none;
  padding: 0;
  margin: 0;
}

/* Styling for each list item */
li {
  background-color: #f9f9f9;
```

```
  border: 1px solid #ddd;
  padding: 10px;
  margin-bottom: 10px;
  border-radius: 5px;
  display: flex;
  justify-content: space-between;
  align-items: center;
  font-size: 1rem;
}

/* Delete button (✕) */
button {
  background-color: red;
  padding: 5px 10px;
  font-size: 1.2rem;
  border-radius: 50%;
  border: none;
  color: white;
  cursor: pointer;
}

/* Delete button hover effect */
button:hover {
  background-color: darkred;
}
```

Backend code

server.js

```
// server.js
const express = require('express');
const fs = require('fs');
const cors = require('cors');
const app = express();
const FILE = 'todos.json';

app.use(cors());
app.use(express.json());

app.get('/todos', (_, res) =>
  fs.readFile(FILE, 'utf8', (_, data) => res.json(JSON.parse(data || '[]')))
);

app.post('/todos', (req, res) =>
  fs.readFile(FILE, 'utf8', (_, data) => {
    const todos = JSON.parse(data || '[]');
    todos.push(req.body);
    fs.writeFile(FILE, JSON.stringify(todos), () => res.sendStatus(200));
  })
);

app.delete('/todos/:id', (req, res) =>
  fs.readFile(FILE, 'utf8', (_, data) => {
```

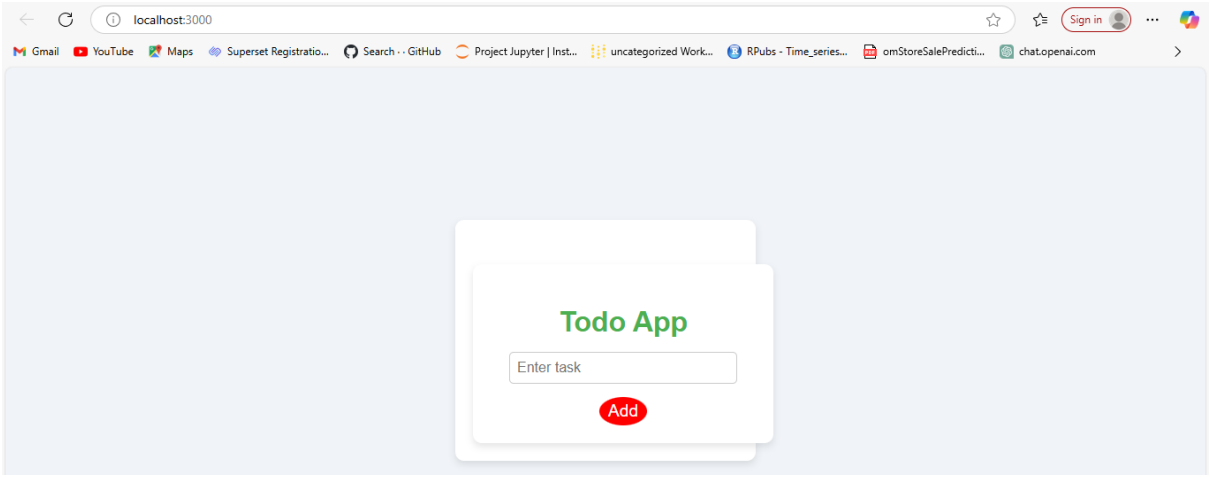
```
    const todos = JSON.parse(data || '[]').filter(t => t.id !== req.params.id);
    fs.writeFile(FILE, JSON.stringify(todos), () => res.sendStatus(200));
  })
);

app.listen(5000, () => console.log('Server on http://localhost:5000'));
```

todos.json

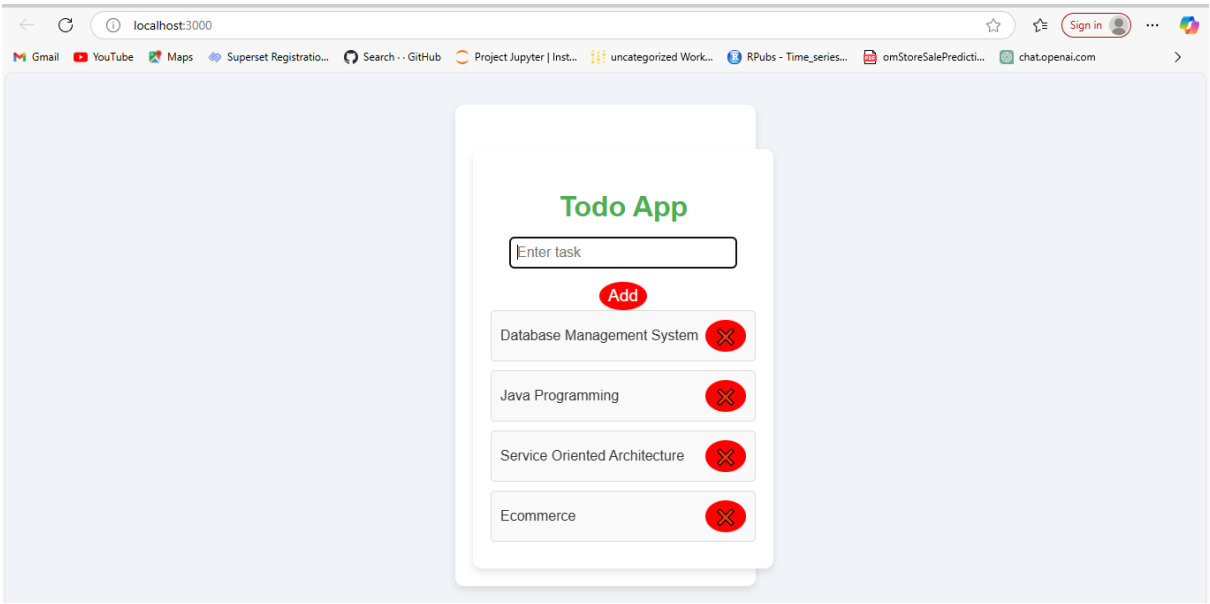
```
[ ]
```

OUTPUT:

VIEWING TODO APP



VIEWING AFTER ADDED TASKS IN TODO APP



Frontend

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>Auth App</title>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

App.js

```
import { useState } from 'react';
import axios from 'axios';

// Set the base URL and ensure credentials (cookies) are sent with requests
axios.defaults.baseURL = 'http://localhost:5000';
axios.defaults.withCredentials = true;

export default function App() {
  const [email, setEmail] = useState('');
  const [pw, setPw] = useState('');
  const [msg, setMsg] = useState('');

  const signup = () => {
    if (!email || !pw) {
      setMsg('Please enter both email and password');
      return;
```

```
    }
    axios.post('/signup', { email, password: pw }).then(() => setMsg('Signed Up'));
  };

  const login = () => {
    if (!email || !pw) {
      setMsg('Please enter both email and password');
      return;
    }
    axios.post('/login', { email, password: pw }).then(() => setMsg('Logged In'));
  };

  const profile = () => {
    if (!email || !pw) {
      setMsg('Please log in first');
      return;
    }
    axios.get('/profile').then(res => setMsg(`User: ${res.data.id}`)).catch(() => setMsg('Please
log in first'));
  };

  return (
    <div>
      <input placeholder="Email" onChange={e => setEmail(e.target.value)} />
      <input placeholder="Password" type="password" onChange={e =>
setPw(e.target.value)} />
      <button onClick={signup}>Signup</button>
      <button onClick={login}>Login</button>
      <button onClick={profile}>Profile</button>
      <p>{msg}</p>
    </div>
  );
}
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import App from './App';
import './style.css';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);
```

style.css

```
body {
    font-family: sans-serif;
    background: #f4f4f4;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }

  div {
    background: white;
    padding: 2rem;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }

  input {
    display: block;
    margin: 10px 0;
    padding: 0.5rem;
    width: 200px;
  }
```

```
button {
  margin-right: 10px;
  padding: 0.5rem 1rem;
  background: #007bff;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover {
  background: #0056b3;
}

p {
  margin-top: 15px;
  font-weight: bold;
}
```

Backend

Server.js

```
const express = require('express');
const mongoose = require('mongoose');
const bcrypt = require('bcryptjs');
const cookie = require('cookie-parser');
const cors = require('cors');
const app = express();

// Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/auth')
  .then(() => console.log('MongoDB connected'))
  .catch(err => console.error('MongoDB error:', err));
```

```
const User = mongoose.model('User', new mongoose.Schema({
  email: String,
  password: String
}));

// Middleware
app.use(cors({ origin: 'http://localhost:3000', credentials: true }));
app.use(express.json(), cookie());

// Helper function to validate email
const isValidEmail = (email) => /^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-
Z]{2,6}$/.test(email);

// Sign Up Route
app.post('/signup', async (req, res) => {
  console.log('Received Signup request:', req.body); // Debugging line

  const { email, password } = req.body;

  // Check for empty fields
  if (!email || !password) {
    return res.status(400).send('Email and Password are required');
  }

  // Validate email format
  if (!isValidEmail(email)) {
    return res.status(400).send('Invalid email format');
  }

  const hash = await bcrypt.hash(password, 10);
  await User.create({ email, password: hash });
  res.send('OK');
});
```

```
// Login Route
app.post('/login', async (req, res) => {
  console.log('Received Login request:', req.body); // Debugging line

  const { email, password } = req.body;

  // Check for empty fields
  if (!email || !password) {
    return res.status(400).send('Email and Password are required');
  }

  const user = await User.findOne({ email });

  if (!user || !(await bcrypt.compare(password, user.password))) {
    return res.status(401).send('Invalid credentials');
  }

  res.cookie('user', user._id, { httpOnly: true, secure: false }).send('Logged in');
});

// Profile Route
app.get('/profile', async (req, res) => {
  console.log('Received Profile request with cookies:', req.cookies); // Debugging line

  const userId = req.cookies.user;

  // Check if user is logged in
  if (!userId) {
    return res.status(401).send('Unauthorized');
  }

  const user = await User.findById(userId);
  if (!user) return res.status(401).send('Unauthorized');
```

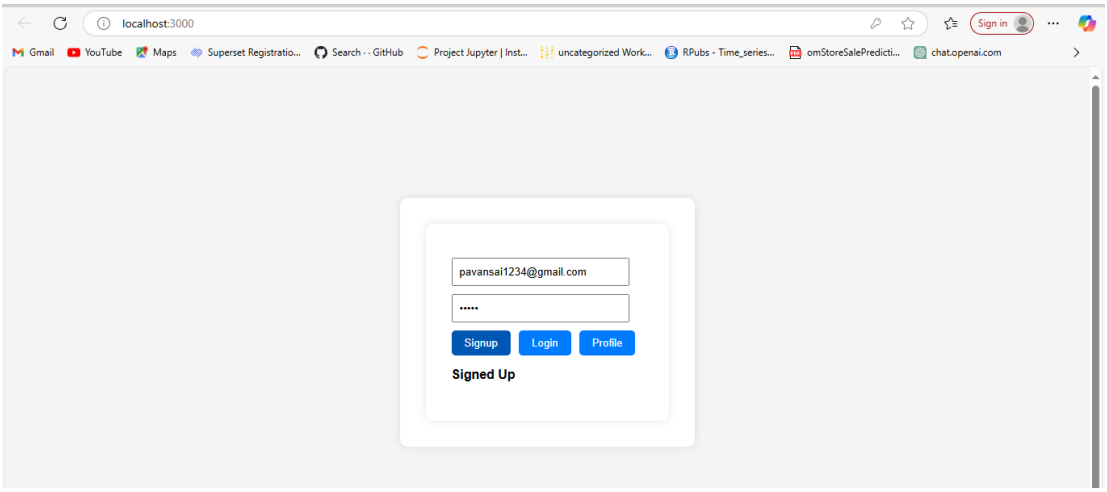
```
  res.json({ id: user._id });
});

// Start server
app.listen(5000, () => console.log('Server running on http://localhost:5000'));
```

User.js

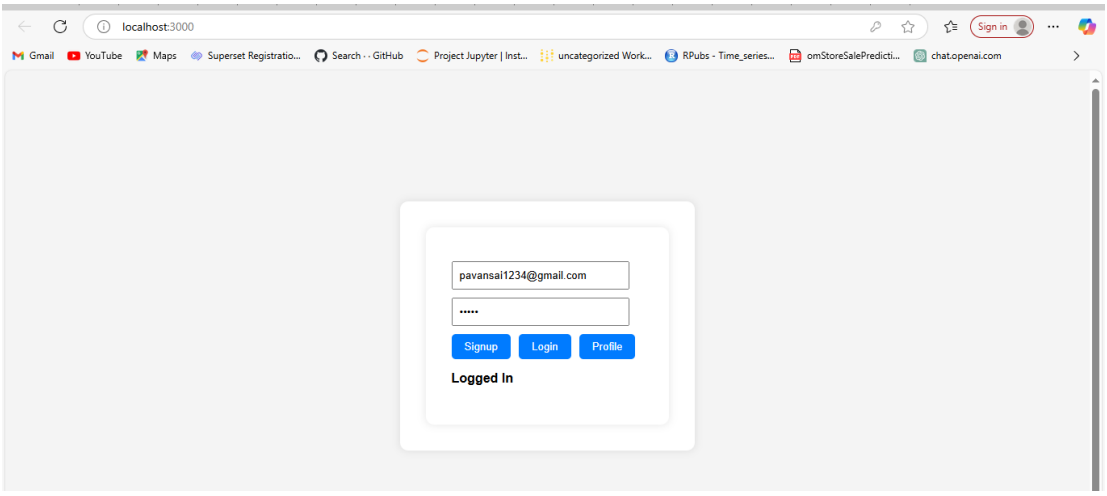
```
const mongoose = require('mongoose');

// Define User schema
const userSchema = new mongoose.Schema({
  email: {
    type: String,
    required: true,
    unique: true,  // Ensure email is unique in the database
    lowercase: true,
    trim: true,
  },
  password: {
    type: String,
    required: true,
  },
});

// Create and export the User model
const User = mongoose.model('User', userSchema);

module.exports = User;
```

OUTPUT:

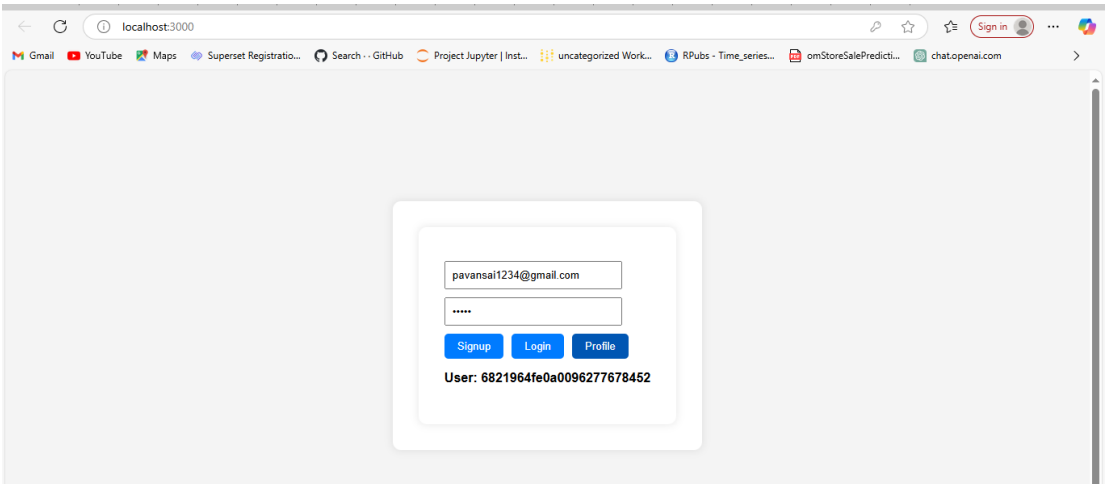
SUCCESFULLY SIGNED UP



AFTER LOGIN



VIEWING USER ID BY CLICKING THE PROFILE



index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Create Event</title>
</head>
<body>
  <h1>Create a New Event</h1>
  <form action="/events" method="POST">
    <label>
      Title:<br />
      <input type="text" name="title" required />
    </label>
    <br /><br />
    <label>
      Date:<br />
      <input type="date" name="date" required />
    </label>
    <br /><br />
    <label>
      Location:<br />
      <input type="text" name="location" required />
    </label>
    <br /><br />
    <label>
      Description:<br />
      <textarea name="description" rows="4" cols="30"></textarea>
    </label>
    <br /><br />
```

```
    <button type="submit">Add Event</button>
  </form>

  <br />
  <p><a href="events.html">View All Events</a></p>
</body>
</html>
```

events.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <title>Event List</title>
  <style>
    table, th, td { border: 1px solid black; border-collapse: collapse; padding: 8px; }
    th { background: #eee; }
    #updateForm { display: none; margin-top: 20px; }
  </style>
</head>
<body>
  <h1>Events</h1>
  <table id="eventsTable">
    <thead>
      <tr>
        <th>Title</th><th>Date</th><th>Location</th><th>Description</th><th>Actions</th>
      </tr>
    </thead>
    <tbody></tbody>
  </table>

  <form id="updateForm">
    <h2>Update Event</h2>
```

```
  <input type="hidden" name="id" />
  <label>Title:<br /><input name="title" required /></label><br />
  <label>Date:<br /><input type="date" name="date" required /></label><br />
  <label>Location:<br /><input name="location" required /></label><br />
  <label>Description:<br /><textarea name="description"></textarea></label><br /><br />
  <button type="submit">Save</button>
  <button type="button" id="cancelUpdate">Cancel</button>
</form>

<script>
  const tableBody = document.querySelector('#eventsTable tbody');
  const updateForm = document.getElementById('updateForm');

  // Fetch and show events in table
  async function loadEvents() {
    const res = await fetch('/events');
    const events = await res.json();

    tableBody.innerHTML = '';
    for (const e of events) {
      const tr = document.createElement('tr');
      tr.innerHTML = `
        <td>${e.title}</td>
        <td>${e.date}</td>
        <td>${e.location}</td>
        <td>${e.description}</td>
        <td>
          <button onclick="showUpdateForm('${e._id}', '${e.title}', '${e.date}', '${e.location}',
\`${e.description}\`)">Update</button>
          <button onclick="deleteEvent('${e._id}')">Delete</button>
        </td>
      `;
      tableBody.appendChild(tr);
    }
```

```
}

// Show the update form with event data
function showUpdateForm(id, title, date, location, description) {
  updateForm.style.display = 'block';
  updateForm.id.value = id;
  updateForm.title.value = title;
  updateForm.date.value = date;
  updateForm.location.value = location;
  updateForm.description.value = description;
}

// Hide update form
document.getElementById('cancelUpdate').onclick = () => {
  updateForm.style.display = 'none';
};

// Handle update form submit
updateForm.onsubmit = async (e) => {
  e.preventDefault();
  const id = updateForm.id.value;
  const data = new URLSearchParams(new FormData(updateForm));

  await fetch('/events/update/' + id, {
    method: 'POST',
    body: data,
    headers: { 'Content-Type': 'application/x-www-form-urlencoded' }
  });
  updateForm.style.display = 'none';
  loadEvents();
};

// Delete event
async function deleteEvent(id) {
```

```
    if (confirm('Delete this event?')) {
      await fetch('/events/delete/' + id, { method: 'POST' });
      loadEvents();
    }
  }

  loadEvents();
 </script>
</body>
</html>
```

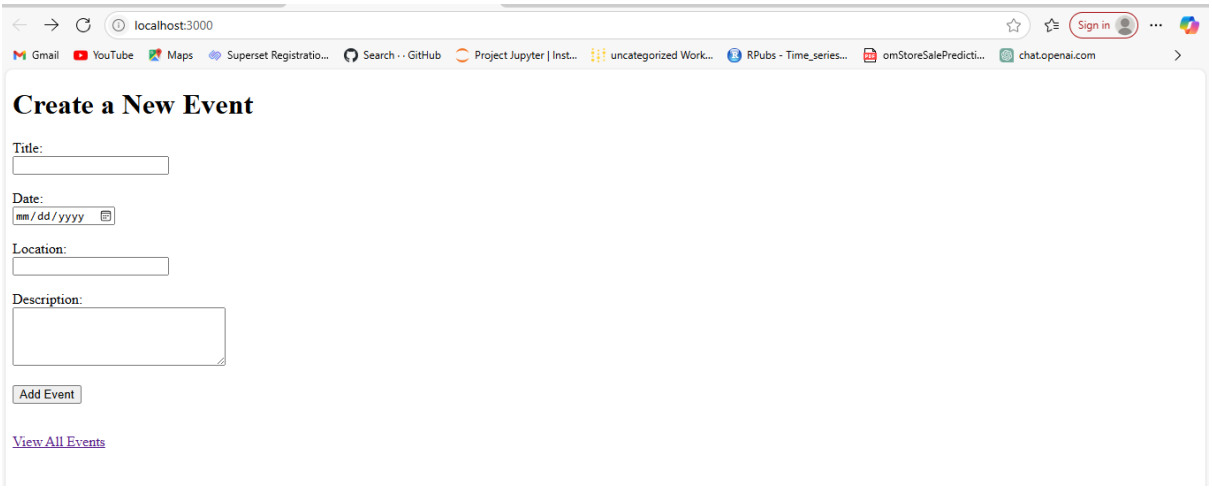
server.js

```
const express = require('express')
const mongoose = require('mongoose')
const bodyParser = require('body-parser')

mongoose.connect('mongodb://localhost:27017/eventsdb')
 .then(() => console.log('MongoDB connected'))
 .catch(err => console.error('MongoDB connection error:', err))

const Event = mongoose.model('Event', new mongoose.Schema({
 title: String,
 date: String,
 location: String,
 description: String,
}))

const app = express()
app.use(bodyParser.urlencoded({ extended: true }))
app.use(express.static('public'))

app.get('/', (req, res) => {
 res.sendFile(__dirname + '/public/index.html')
```

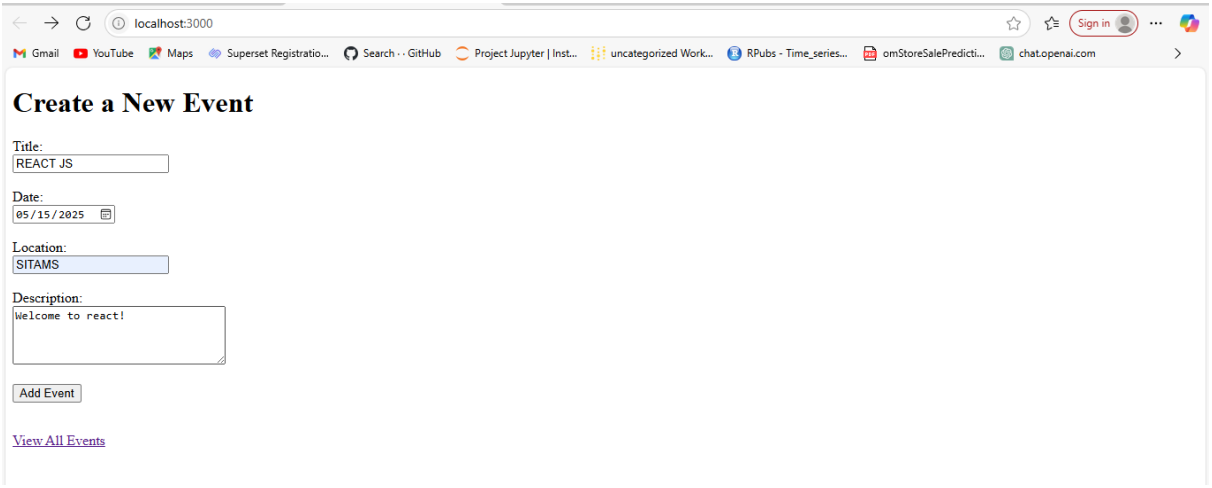
```
})

app.get('/test', (req, res) => res.send('Test OK'))

app.post('/events', async (req, res) => {
  await new Event(req.body).save()
  res.redirect('/events')
})

app.get('/events', async (req, res) => {
  const events = await Event.find()
  res.send(events)
})

app.post('/events/update/:id', async (req, res) => {
  await Event.findByIdAndUpdate(req.params.id, req.body)
  res.redirect('/events')
})

app.post('/events/delete/:id', async (req, res) => {
  await Event.findByIdAndDelete(req.params.id)
  res.redirect('/events')
})

app.listen(3000, () => console.log('Server running on http://localhost:3000'))
```

OUTPUT:

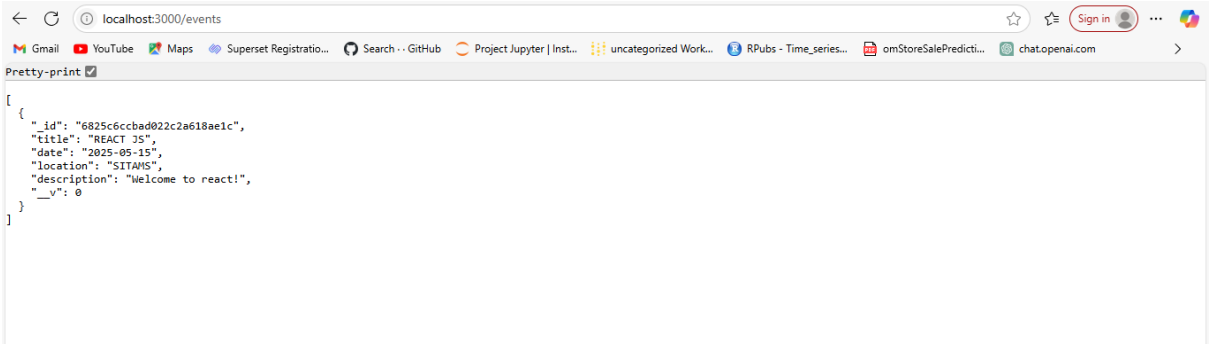
CREATE A NEW EVENT



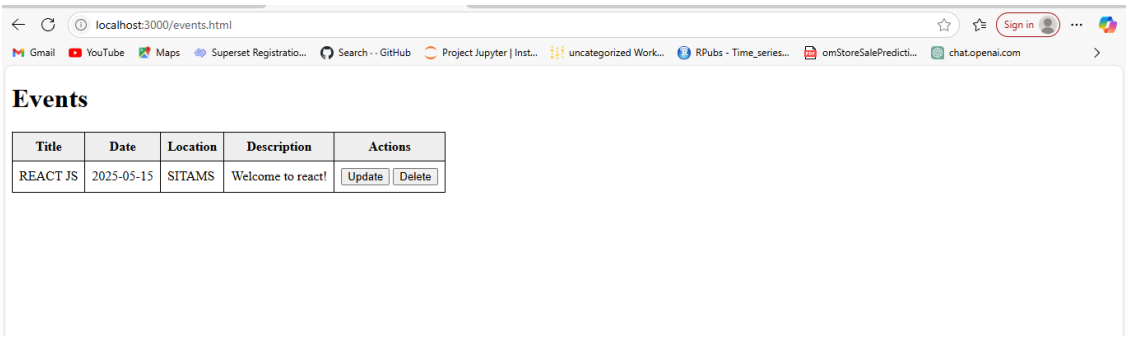
AFTER ENTERING DATA CLICK “ADD EVENT”



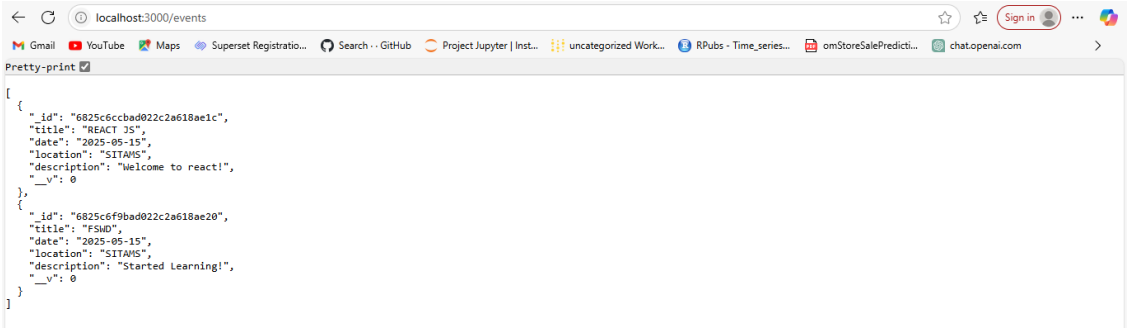
AFTER CLICKING THE ADD EVENT, VIEWING THE EVENT IN “JSON FORMAT”



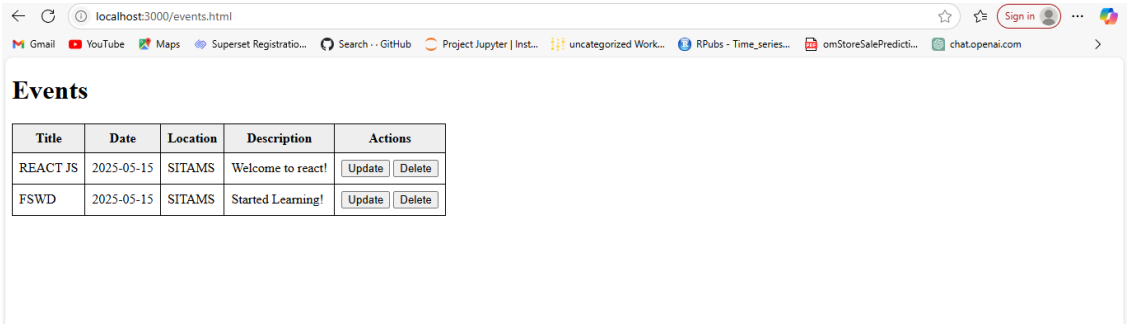
VIEWING THE EVENTS LIST



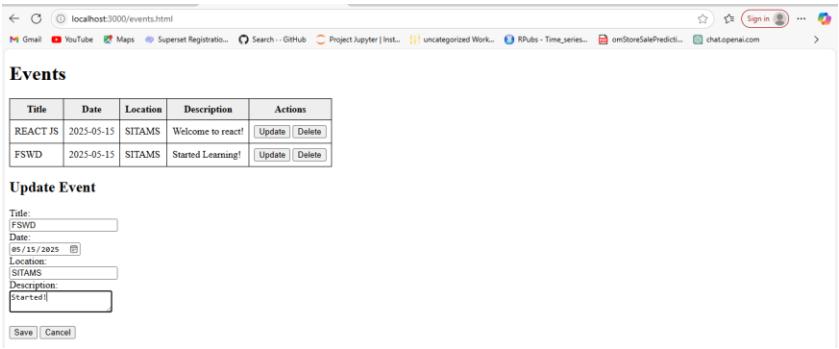
AFTER ADDING ANOTHER DATA VIEWING THE MULTIPLE EVENTS IN "JSON FORMAT"



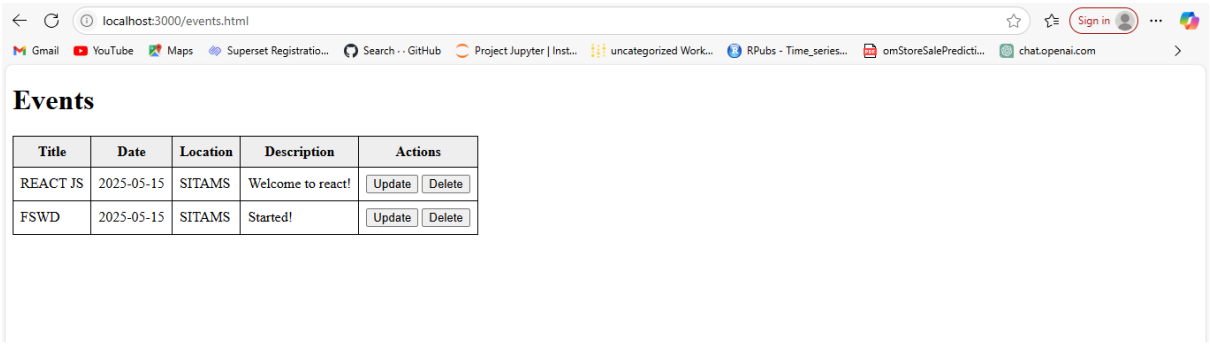
VIEWING ALL EVENTS LIST



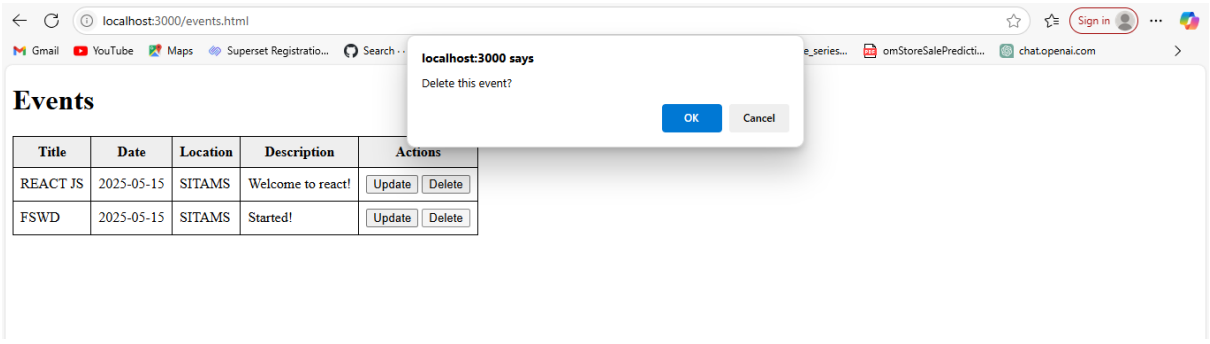
UPDATING EVENT IN THE EVENTS LIST



AFTER UPDATION VIEWING EVENTS LIST



DELETING THE EVENT IN THE EVENTLIST



AFTER DELETION VIEWING THE EVENTS LIST