RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

Laboratory Record Note Book

Name : . . . .

Year / Branch / Section : . . . .

Register No. : . . . .

College Roll No. : . . . .

Semester : . . . .

Academic Year : . . . .

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

BONAFIDE CERTIFICATE

Name : . . . .

Academic Year : . . . . Semester : . . . . Branch : . . . .

Register No.

Certified that this is the bonafide record of work done by the above student in the

...Laboratory during the year

20 - 20

Signature of Faculty in-charge

Submitted for the Practical Examination held on . . . .

Internal Examiner **External Examiner**

INDEX

Reg. No. : ____________________ Name : ________________________

Year : __________ Branch : __________ Sec : _____

Ex. No. : 1 Date : Reg. No. :**211701049**

HTML – Image Map

Aim: Program to create and use image maps:

i. To embed a map in a web page.
ii. To fix the hot spots in that map.
iii. Show all the related information when the hot spots are clicked.

Procedure:

1. Define an image maps by using a map element.

2. Use attribute id to identify the image map.

3. Define hotspots with area elements.

4. Use attribute href to specify the link's target (i.e.,the resource to which to link).

5. Use attributes shape and coords to specify the hotspot's shape and coordinates, respectively.

6. Use attribute alt to provide alternate text for the link.

7. Use the markup to create a rectangular hotspot (shape = "rect") for the coordinates specified in the coords attribute (For rectangular hotspots, the required coordinates are those of the upper-left and lower-right corners of the rectangle).
8. Use the map area to assign the shape attribute "poly" to create a hotspot in the shape of a polygon using the coordinates in attribute coords (These coordinates represent each vertex, or corner, of the polygon).
9. Use the map area to assign the shape attribute "circle" to create a circular hotspot (the coords attribute specifies the circle's center coordinates and the circle's radius, in pixels).
10. Use an image map with an img element, the img element's usemap attribute is assigned the id of a map.
11. Locate the image map within the same document so internal linking is used.

Code:

<u>Indiamap.html</u>

```
<!DOCTYPE html>
<html>
<body>
<h1>The map and area elements</h1>
```

```
<p>Click on the Hotspots to go to a new page and read more about the Places:</p>
<img src="india.jpg" alt="Workplace" usemap="#workmap" width="209"

height="242">
<map name="workmap">
<area shape="rect" coords="83,41,57,63" alt="delhi" href="delhi.html">
  <area shape="rect" coords="149,129,124,83" alt="calcutta" href="calcutta.html">
  <area shape="rect" coords="89,184,56,226" alt="tamilnadu" href="tamilnadu.html">
</map>
</body>
</html>
```

calcutta.html

```
<html>
<body bgcolor="white">
<font face="Times New Roman" size="10" color="RED">
<center><b><i>Calcutta is the wealthy city in WEST BENGAL<br> and <br>It has
Famous
"Sunderbans Forests"</i></b></center>
<a href="Indiamap.html">Home Page</a>
</font>
</body>
</html>
```

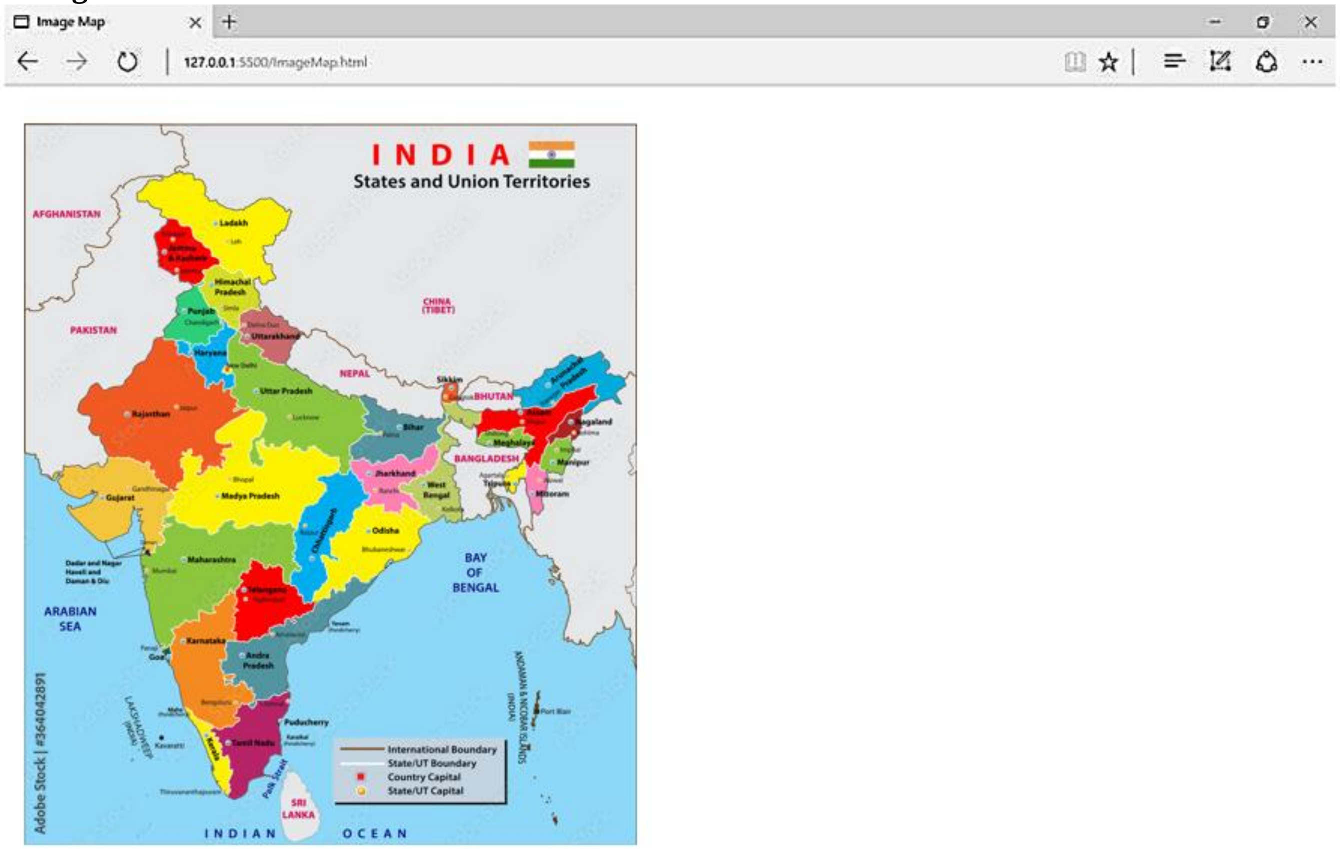
delhi.html

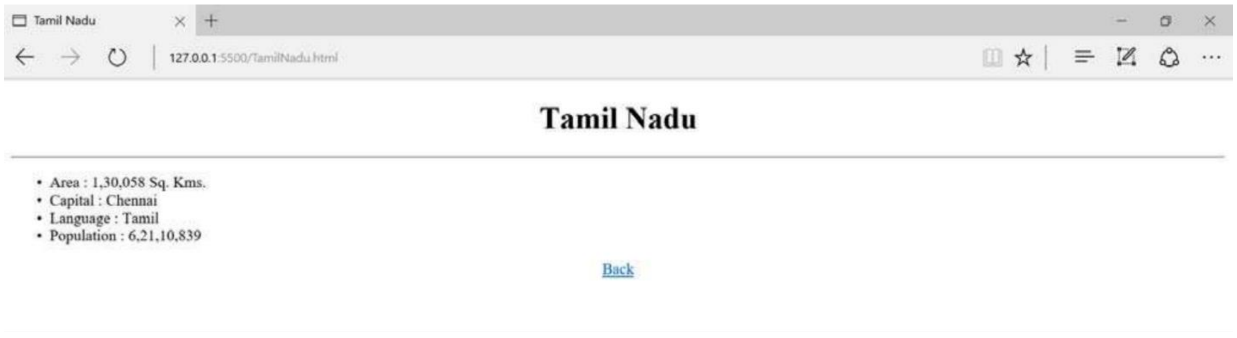
```
<html>
<body bgcolor="white">
<font face="Arial" size="10" color="RED">
<center><b><i><tt>Delhi is the capital of our INDIA<br> and <br>More IT companies
are
Camped at Delhi</tt></i></b></center>
<a href="Indiamap.html">Home Page</a>
</font>
</body>
</html>
```

tamilnadu.html

```
<html>
<body bgcolor="white">
<font face="Times New Roman" size="10" color="orange">
<center>Chennai is the capital of Tamil Nadu<br> and <br>More IT companies are
camped at
Chennai</center>
<a href="Indiamap.html">Home Page</a>
</font>
</body>
</html>
```

Design:





Tamil Nadu

- Area : 1,30,058 Sq. Kms.
- Capital : Chennai
- Language : Tamil
- Population : 6,21,10,839

Back

RESULT: Thus the given design was successfully developed and output was verified.

Ex. No. : 2 Date : Reg. No. :**211701049**

CSS

Aim:

Program to design web pages using basic elements, hyperlinks and to perform web navigation using CSS.

Procedure:

Inline Style Sheets

1. Create inline styles that declare an individual element's format using attribute style.
2. Apply inline styles to p elements to alter their font size and color.
3. Use the attribute style to specify the style for an element.
4. Create CSS property (the font-size property) followed by a colon and a value.
5. Use the two properties, font-size and color, separated by a semicolon.

Embedded Style Sheets

1. Use the style element to define the embedded style sheet.
2. Place the Styles in the head to apply matching elements in the entire document, not just to a single element.
3. Use the type attribute to specify specifies the Multipurpose Internet Mail Extension (MIME) type that describes a file's content. CSS documents use the MIME type text/css.
4. Use the body of the style sheet to declare the CSS rules for the style sheet.
5. The body of each rule is enclosed in curly braces ({ and }).
6. Declare a style class. Class declarations are preceded with a period and are applied to elements only of that class.
7. Use the property name is followed by a colon (:) and the value of that property. Multiple properties are separated by semicolons (;).

Linking External Style Sheets

1. Create a link element, which uses the rel attribute to specify a relationship between the current document and another document.
2. Declare the linked document to be a stylesheet for this document.
3. Use the type attribute to specify the MIME type as text/css.
4. Use the href attribute provides the URL for the document containing the style sheet.

Code:

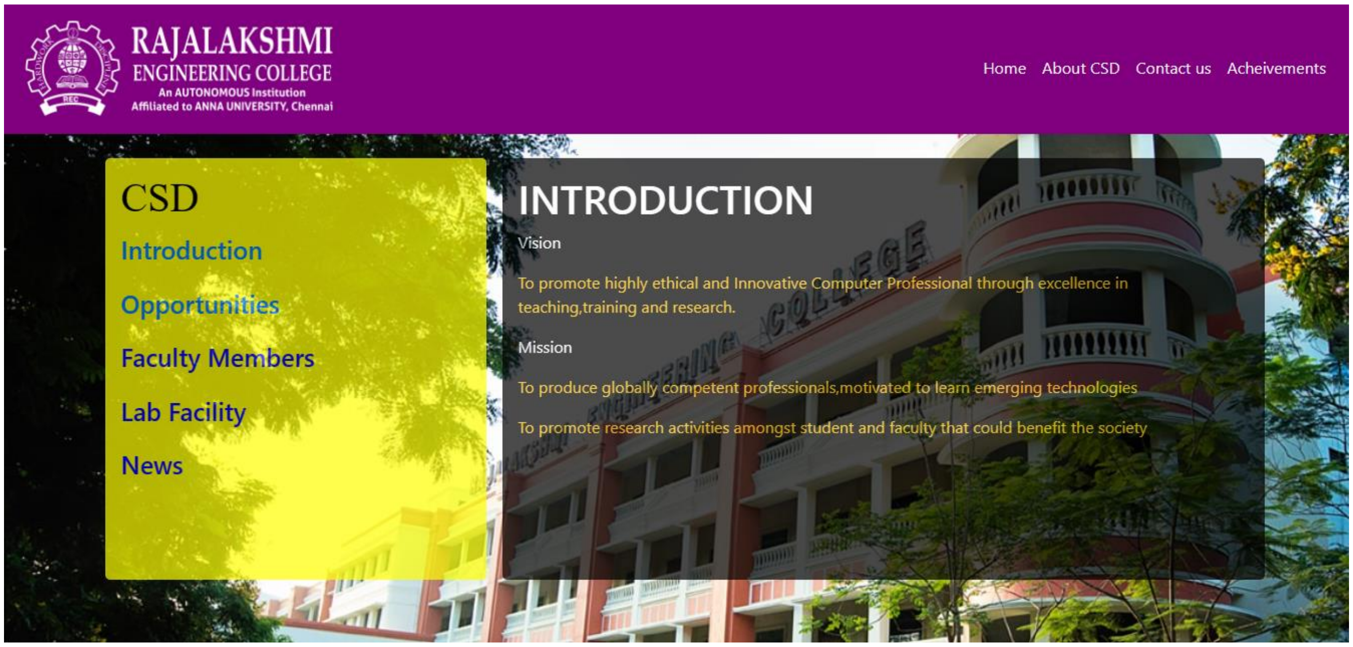
```
<!DOCTYPE html>
<head>
     <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
  <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js" integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js" integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>
  <link rel="stylesheet" type="text/css" href="website.css">
</head>
<body>
     <nav class="navbar navbar-expand-lg  navbar-background navbar-light navbar_bg">
    <a class="navbar-brand" href="#">
      <img
        src="logo-w.png"
        class="navbar-image"
        alt=""
      />
    </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup"  style="color:white;">
      <div class="navbar-nav ml-auto">
        <a class="nav-link active text-white" href="website.html">
          Home <span class="sr-only">(current)</span>
        </a>
        <a class="nav-link active text-white" href="welcome_csd.html">About CSD</a>
        <a class="nav-link active text-white" href="#">Contact us</a>
```

```
          <a class="nav-link active text-white" href="#">Acheivements</a>
        </div>
      </div>
    </nav>
    <div class="website_container">
          <div class="container">
                    <div class="row">
          <div class=" d-flex flex-row mt-4">
                              <div class="links-container col-xl-4 p-3">
                                        <h1 class="department">CSD</h1>
           <p class="content"><a href="website.html">Introduction</a></p>
           <p class="content"><a href="website1.html">Opportunities</a></p>
           <p class="content">Faculty Members</p>
           <p class="content">Lab Facility</p>
           <p class="content">News</p>
                              </div>
          <div class="col-xl-8 item-container p-3 ml-3 text-white">
           <h1 class="text-white">INTRODUCTION</h1>
           <p class="text-black">Vision</p>
           <p class="text-warning">To promote highly ethical and Innovative Computer Professional through
excellence in teaching,training and research.</p>
           <p class="text-black">Mission</p>
           <p class="text-warning">To produce globally competent professionals,motivated to learn emerging
technologies</p>
           <p class="text-warning">To promote research activities amongst student and faculty that could
benefit the society</p>
                              </div>
        </div>
                    </div>
          </div>
          </div>
</body>
</html>
```

<u>website.css</u>

```
.website_container{
    background-image:url("rec_image.jpg");
    background-size:cover;
    height:100vh;
}
.navbar-image{
    width:320px;
    height:106px;
}
.navbar_bg{
    background-color:purple;
    color:white;
}
.links-container{
    background-color:yellow;
    opacity:0.7;
    height:65vh;
    width:20vw;
    color:yellow;
    border-radius:5px;

}
.department{
    font-family:"serif";
    color:black;

}
.content{
    color:blue;
    font-size:25px;
    font-weight:600;
}
.item-container{
    background-color:black;
    opacity:0.7;
    border-radius:5px;


}
```

Design:



RAJALAKSHMI ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

Home About CSD Contact us Acheivements

CSD
Introduction
Opportunities
Faculty Members
Lab Facility
News

INTRODUCTION

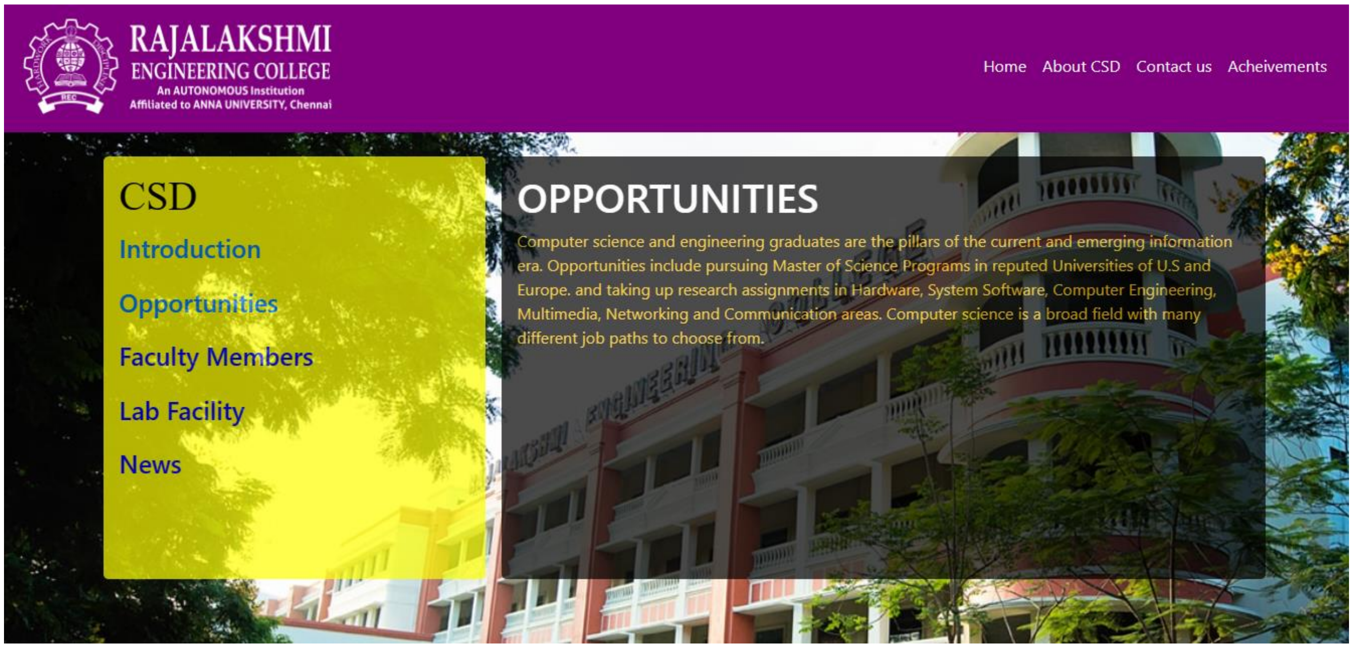
Vision

To promote highly ethical and Innovative Computer Professional through excellence in teaching,training and research.

Mission

To produce globally competent professionals,motivated to learn emerging technologies

To promote research activities amongst student and faculty that could benefit the society



RAJALAKSHMI ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

Home About CSD Contact us Acheivements

CSD
Introduction
Opportunities
Faculty Members
Lab Facility
News

OPPORTUNITIES

Computer science and engineering graduates are the pillars of the current and emerging information era. Opportunities include pursuing Master of Science Programs in reputed Universities of U.S and Europe. and taking up research assignments in Hardware, System Software, Computer Engineering, Multimedia, Networking and Communication areas. Computer science is a broad field with many different job paths to choose from.

RESULT: Thus the given design was successfully developed and output was verified.

Ex.No. : 3 Date :14.2.2024 Reg. No. :211701049

FORM VALIDATION

AIM:

Create a HTML form for course registration with student_name, rollno, gender, year, department, Section, mobile_no, E-Mail_ID, Address, City, Country, pincode and validate with the following specifications.

I. Check whether all the inputs are entered or not.

II. Check whether the inputs entered should be in correct format.

After validating using JavaScript, display proper error messages in red color just next to the textbox where there is an error.

PROCEDURE:

1. The form is defined by a form element.

<form method = "post" action = "#">

2. Use the attribute method specifies how the form’s data is sent to the Web server.Using method = "post" appends form data to the browser request, which contains the protocol (i.e., HTTP) and the requested resource’s URL. Scripts located on the Web server’s computer (or on a computer accessible through the network) can access the form data sent as part of the request. For example, a script may take the form information and update an electronic mailing list. The other possible value, method = "get" appends the form data directly to the end of the URL.

3. The action attribute in the <form> tag specifies the URL of a script on the Web server’

4. Use the type of input as "text" input inserts a text box into the form. Users can type data in text boxes.

5. The input element’s size attribute specifies the number of characters visible in the text box. Optional attribute maxlength limits the number of characters input into the text box.

6. There are two types of input elements in lines

<input type = "submit" value = "Submit Your Entries" />

<input type = "reset" value = "Clear Your Entries" />

7. The "submit" input element is a button. When the user presses a "submit" button, the browser sends the data in the form to the Web server for processing. The value attribute sets the text displayed on the button (the default value is Submit Query).

8. The "reset" input element allows a user to reset all form elements to their default values. The value attribute of the "reset" input element sets the text displayed on the button (the default value is Reset).

9. The textarea element inserts a multiline text box, called a text area, into the form. The number of rows is specified with the rows attribute and the number of columns (i.e.,characters) is specified with the cols attribute. In this example, the textarea is four rows high and 36 characters wide. To display default text in the text area, place the text between the <textarea> and </textarea> tags. Default text can be specified in other input types, such as text boxes, by using the value attribute.

10. The "password" input in lines inserts a password box with the specified size. A password box allows users to enter sensitive information, such as credit card numbers and passwords, by “masking” the information input with asterisks. The actual value input is sent to the Web server, not the character that mask the input.

11. Checkboxes enable users to select from a set of options. When a user selects a checkbox, a check mark appears in the check box. Otherwise, the checkbox remains empty. Each "checkbox" input creates a new checkbox. Checkboxes can be used individually or in groups. Checkboxes that belong to a group are assigned the same name.

12. Radio buttons are similar to checkboxes, except that only one radio button in a group of radio buttons may be selected at any time. The radio buttons in a group have the same name attributes and are distinguished by their different value attributes. The attribute- value pair checked = "checked" indicates which radio button, if any, is selected initially. The checked attribute also applies to checkboxes.

13. The select element provides a drop-down list of items from which the user can select an item. The name attribute identifies the drop-down list. The option element adds items to the drop-down list. The option element’s selected attribute specifies which item initially is displayed as the selected item in the select element.

14. Use the events for processing forms - onsubmit and onreset.

15. These events fire when a form is submitted or reset, respectively.

Code:

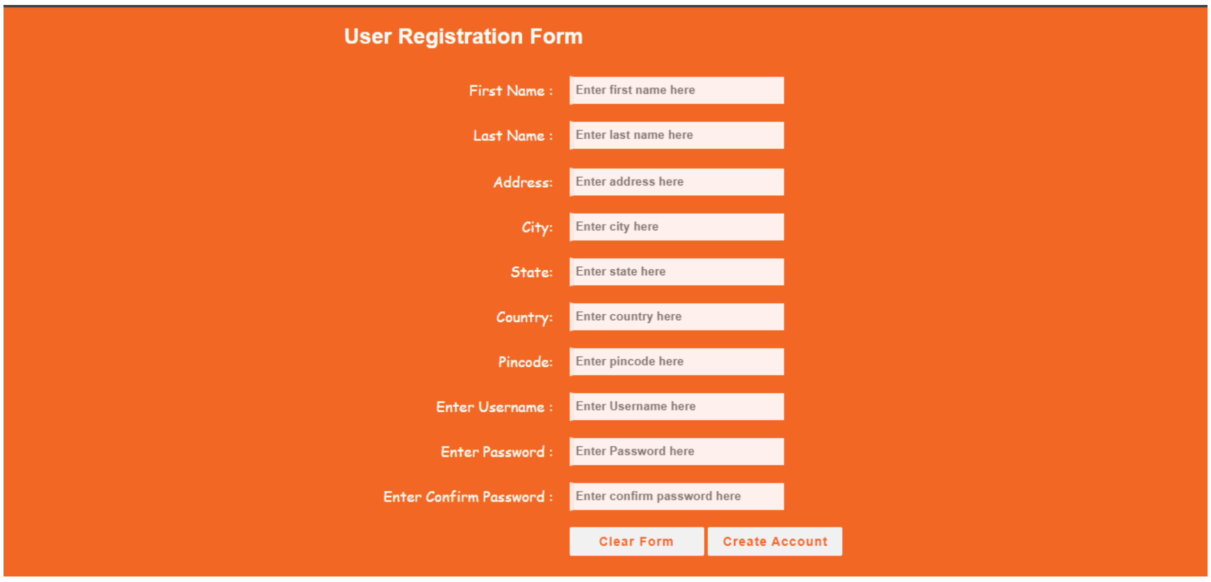
```
<!DOCTYPE html>
  <html>
  <head>
  <title>Welcome To Registration Form</title>
  <style type="text/css">
     body
    {
            margin:0px;
            background-color:#f26724;
            background-image:url(image/background.jpg);
            color:#f9fcf5;
            font-family:Arial, Helvetica, sans-serif;
    }

    #main{width:600px; height:auto; overflow:hidden; padding-bottom:20px; margin-left:auto; margin-
right:auto;
    border-radius:5px; padding-left:10px; margin-top:100px; border-top:3px double #f1f1f1;
    border-bottom:3px double #f1f1f1; padding-top:20px;
    }

    #main table{font-family:"Comic Sans MS", cursive;}
    /* css code for textbox */
    #main .tb{height:28px; width:230px; border:1px solid #f26724; color:#fd7838; font-weight:bold;
border-left:5px solid #f7f7f7; opacity:0.9;}

    #main .tb:focus{height:28px; border:1px solid #f26724; outline:none; border-left:5px solid #f7f7f7;}

    /* css code for button*/
    #main .btn{width:150px; height:32px; outline:none;  color:#f7f7f7; font-weight:bold; border:0px solid
#f26724;
    text-shadow: 0px 0.5px 0.5px #fff; border-radius: 2px; font-weight: 600; color: #f26724; letter-spacing:
1px;
    font-size:14px; background-color:#f1f1f1; -webkit-transition: 1s; -moz-transition: 1s; transition: 1s;}

    #main .btn:hover{background-color:#f26724; outline:none;  border-radius: 2px; color:#f1f1f1;
border:1px solid #f1f1f1;
    -webkit-transition: 1s; -moz-transition: 1s; transition: 1s; }
```

```
</style>
 <script>
   function registration()
        {

                var name= document.getElementById("t1").value;
                var email= document.getElementById("t2").value;
                var uname= document.getElementById("t3").value;
                var pwd= document.getElementById("t4").value;
                var cpwd= document.getElementById("t5").value;

      //email id expression code
                var pwd_expression = /^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@$%^&*-])/;
                var letters = /^[A-Za-z]+$/;
                var filter = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

                if(name=='')
                {
                        alert('First name/last name/city/country/user name/password must be filled out');
                }
                else if(!letters.test(name))
                {
                        alert('Name field required only alphabet characters');
                }
                else if(email=='')
                {
                        alert('Please enter your user email id');
                }
                else if (!filter.test(email))
                {
                        alert('Invalid email');
                }
                else if(uname=='')
                {
                        alert('Please enter the user name.');
                }
                else if(!letters.test(uname))
                {
                        alert('User name field required only alphabet characters');
                }
                else if(pwd=='')
                {



alert('Please enter Password');
                }
                else if(cpwd=='')
                {
                        alert('Enter Confirm Password');
                }
                else if(!pwd_expression.test(pwd))
                {
                        alert ('Upper case, Lower case, Special character and Numeric letter are required in
Password filed');
                }
                else if(pwd != cpwd)
                {
                        alert ('Password not Matched');
                }
                else if(document.getElementById("t5").value.length < 6)
                {
                        alert ('Password minimum length is 6');
                }
                else if(document.getElementById("t5").value.length > 12)
                {
                        alert ('Password max length is 12');
                }
                else
                {
            alert('This site says...');
            alert('First name/last name/city/country/user name/password must be filled out');

                }
        }
        function clearFunc()
        {
                document.getElementById("t1").value="";
                document.getElementById("t2").value="";
                document.getElementById("t3").value="";
```

```
                    document.getElementById("t4").value="";
                    document.getElementById("t5").value="";
          }
   </script>

</head>
<body>
          <!-- Main div code -->
          <div id="main">
          <div class="h-tag">
          <h2>User Registration Form</h2>
          </div>
          <div class="login">
          <table cellspacing="2" align="center" cellpadding="8" border="0">
          <tr>
          <td align="right">First Name :</td>
          <td><input type="text" placeholder="Enter first name here" id="t1" class="tb" /></td>
          </tr>
        <tr>
        <td align="right">Last Name :</td>
        <td><input type="text" placeholder="Enter last name here" id="t1" class="tb" /></td>
        </tr>
              <tr>
        <tr>
           <td align="right">Address:</td>
           <td><input type="text" placeholder="Enter address here" id="t1" class="tb" /></td>
        </tr>
        <tr>
           <td align="right">City:</td>
           <td><input type="text" placeholder="Enter city here" id="t1" class="tb" /></td>
        </tr>
        <tr>
           <td align="right">State:</td>
           <td><input type="text" placeholder="Enter state here" id="t1" class="tb" /></td>
        </tr>
        <tr>
           <td align="right">Country:</td>
           <td><input type="text" placeholder="Enter country here" id="t1" class="tb" /></td>
           </tr>
```

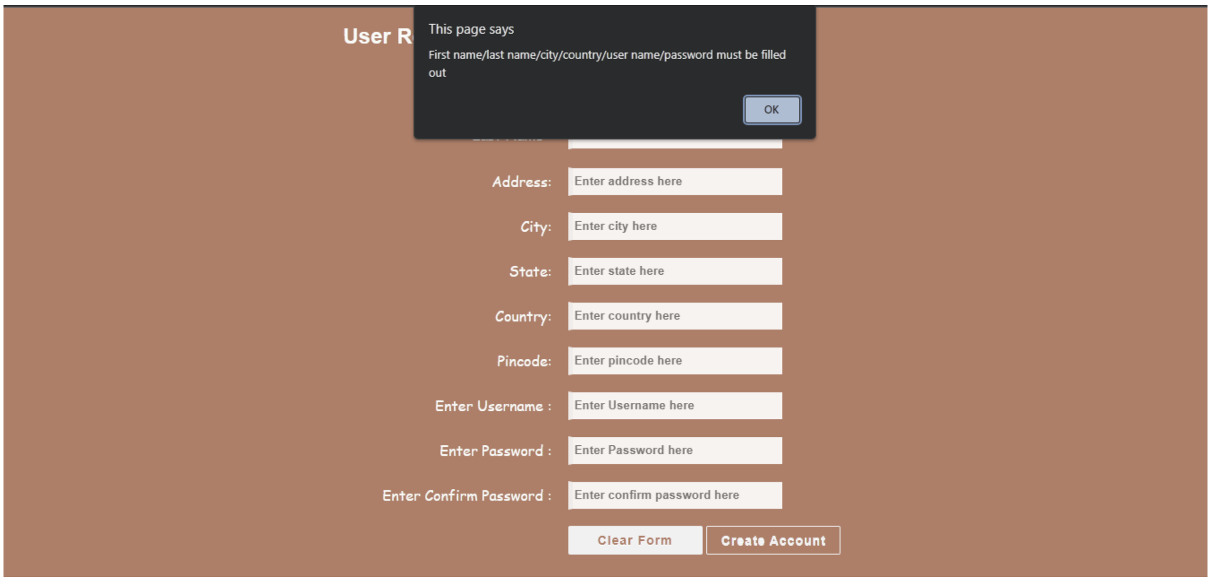
```
	<td align="right">Pincode:</td>
	<td><input type="text" placeholder="Enter pincode here" id="t2" class="tb" /></td>
	</tr>
	<tr>
	<td align="right">Enter Username :</td>
	<td><input type="text" placeholder="Enter Username here" id="t3" class="tb" /></td>
	</tr>
	<tr>
	<td align="right">Enter Password :</td>
	<td><input type="password" placeholder="Enter Password here" id="t4" class="tb" /></td>
	</tr>
	<tr>
	<td align="right">Enter Confirm Password :</td>
	<td><input type="password" placeholder="Enter confirm password here" id="t5" class="tb"
/></td>
	</tr>
	<tr>
	<td></td>
	<td>
	<input type="reset" value="Clear Form" onclick="clearFunc()" id="res" class="btn" />
	<input type="submit" value="Create Account" class="btn" onclick="registration()" /></td>
	</tr>
	</table>
	</div>
	<!-- create account box ending here.. -->
	</div>
	<!-- Main div ending here... -->
	</body>
	</html>
```

Design:



User Registration Form

First Name :	Enter first name here
Last Name :	Enter last name here
Address:	Enter address here
City:	Enter city here
State:	Enter state here
Country:	Enter country here
Pincode:	Enter pincode here
Enter Username :	Enter Username here
Enter Password :	Enter Password here
Enter Confirm Password :	Enter confirm password here



This page says

First name/last name/city/country/user name/password must be filled out

Address:	Enter address here
City:	Enter city here
State:	Enter state here
Country:	Enter country here
Pincode:	Enter pincode here
Enter Username :	Enter Username here
Enter Password :	Enter Password here
Enter Confirm Password :	Enter confirm password here

RESULT: Thus the given design was successfully developed and output was verified.

Ex.No. : 4a Date : Reg. No. :211701049

LIBRARY MANAGEMENT SYSTEM

AIM:

To create a responsive library management system using HTML, CSS and to develop a JavaScript program that will validate the controls in the form you have created for the application.

CODE:

Lib.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Library Management System</title>
<style>
body{
background-image: url(lib\ bg.jpeg);
background-repeat: no-repeat;
background-size: cover;
}
h1{
color: rgb(55, 3, 3);
text-align: center;
padding-bottom: 10px;
font-weight: bolder;
}
.tit{
margin-top: 100px;
}
td{
width: 50%;
}
table,tr,td{
border: 1px solid black;
border-collapse: collapse;
margin-left: 380px;
padding: 10px;
text-align: center;
color: rgb(55, 3, 3);
font-size: 25px;
font-weight: bold;
}

</style>
</head>
<body>
<h1 class="tit">Library Management System</h1>
<form action="insert.jsp" method="post">
<table>
<tr>
<td><label for="AccNo">ID</label></td>
<td class="sub"><input type="number" id="AccNo" name="AccNo" style="width: 350px;"></td>
</tr>
<tr>
<td><label for="Bname">Book Name</label></td>
<td class="sub"><input type="text" id="Bname" name="Bname" style="width: 350px;"></td>
</tr>
<tr>
<td><label for="Aname">Author Name</label></td>
<td class="sub"><input type="text" id="Aname" name="Aname" style="width: 350px;"></td>
</tr>
<tr>
<td><label for="Pub">Publisher</label></td>
<td class="sub"><input type="text" id="Pub" name="Pub" style="width: 350px;"></td>
</tr>
<tr>
<td><label for="Edit">Edition</label></td>
<td class="sub"><input type="text" id="Edit" name="Edit" style="width: 350px;"></td>
</tr>
<tr>
<td><label for="Price">Price</label></td>
<td class="sub"><input type="number" id="Price" name="Price" style="width: 350px;"></td>
</tr>
<tr>
<td><input type="submit" value="submit"></td>
<td><input type="reset" value="Clear"></td>
</tr>
</table>
</form>
</body>
</html>
```

Insert.html

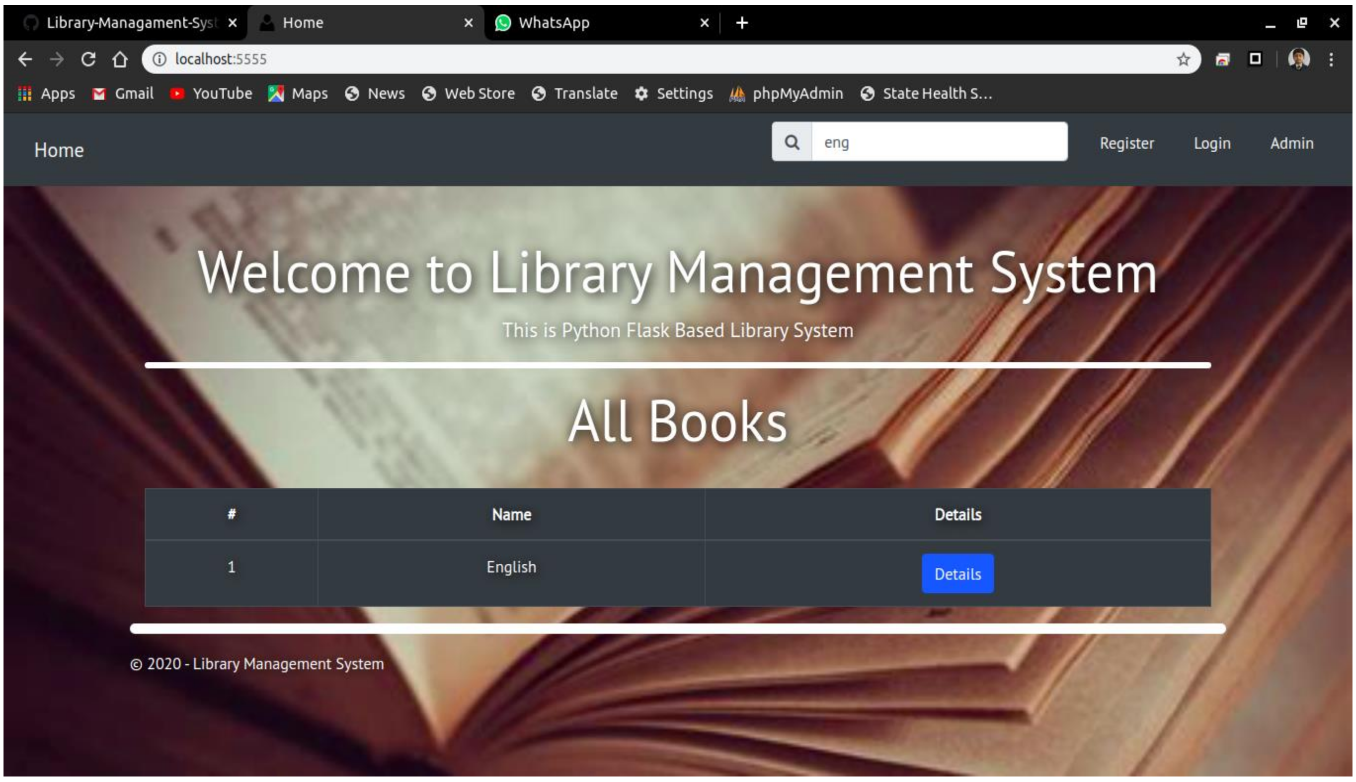
```
<%@ page import="java.sql.*, javax.naming.*, javax.sql.*, java.util.Date, java.text.SimpleDateFormat"
%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>

<head>
<title>Insert Book</title>
```

```
</head>
<body>
<script>
function showAlert(message) {
alert(message);
}
</script>
<h2>Insert Book</h2>
<%
// Retrieve form parameters
String AccountNo = request.getParameter("AccNo");
String Bookname = request.getParameter("Bname");
String authorname = request.getParameter("Aname");
String publisher = request.getParameter("Pub");
String edition = request.getParameter("Edit");
String price = request.getParameter("Price");
out.println("account: "+AccountNo);
out.println("bname: "+Bookname);
out.println("author: " +authorname);
out.println("publisher: "+publisher);
out.println("edition: "+edition);
out.println("price: " +price);

// Check if all parameters are not null and not empty
if (AccountNo != null && !AccountNo.isEmpty() &&
Bookname != null && !Bookname.isEmpty() &&
authorname != null && !authorname.isEmpty() &&
publisher != null && !publisher.isEmpty() &&
edition != null && !edition.isEmpty() &&
price != null && !price.isEmpty()) {
try {
// Get database connection
Context initContext = new InitialContext();
Context envContext = (Context)initContext.lookup("java:/comp/env");
DataSource ds = (DataSource)envContext.lookup("jdbc/LibMan");
Connection conn = ds.getConnection();
// Prepare SQL statement to insert book
String sql = "INSERT INTO libtable (Acname, Bookn, Authn, Publish, Edition, Amount) VALUES
(?, ?, ?, ?, ?, ?)";
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1, Integer.parseInt(AccountNo));
pstmt.setString(2, Bookname);

pstmt.setString(3, authorname);
pstmt.setString(4, publisher);
pstmt.setString(5, edition);
pstmt.setInt(6, Integer.parseInt(price));
// Execute the insert statement
int rowsAffected = pstmt.executeUpdate();
// Close resources
pstmt.close();
conn.close();
out.println("<script>showAlert('Book inserted successfully');</script>");
} catch (Exception e) {
e.printStackTrace();
out.println("<script>showAlert('Error inserting book');</script>");
}
} else {
out.println("<script>showAlert('One or more parameters are missing');</script>");
}
%>
</body>
</html>
```

View.html

```
<%@ page import="java.sql.*" %>
<%@ page import="javax.naming.*, javax.sql.*" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
<title>Books Information</title>
</head>
<body>
<h2>Books Information</h2>
<table border="1">
<tr>
<th>Book ID</th>
<th>Title</th>
<th>Author</th>
<th>Publication</th>
<th>Edition</th>
<th>Price</th>
</tr>
<%
try {
// Get database connection

Context initContext = new InitialContext();
Context envContext = (Context)initContext.lookup("java:/comp/env");
DataSource ds = (DataSource)envContext.lookup("jdbc/LibMan");
Connection conn = ds.getConnection();
// Create SQL statement
Statement stmt = conn.createStatement();
String sql = "SELECT Acname, Bookn , Authn, Publish, Edition, Amount FROM libtable";
ResultSet rs = stmt.executeQuery(sql);
// Iterate through the result set and display records
while (rs.next()) {
%>
<tr>
<td><%= rs.getString("Acname") %></td>
<td><%= rs.getString("Bookn") %></td>
<td><%= rs.getString("Authn") %></td>
<td><%= rs.getString("Publish") %></td>
<td><%= rs.getString("Edition") %></td>
<td><%= rs.getString("Amount") %></td>
</tr>
<%
}
rs.close();
stmt.close();
conn.close();
} catch (Exception e) {
e.printStackTrace();
}
%>
</table>
</body>
</html>
```

OUTPUT:



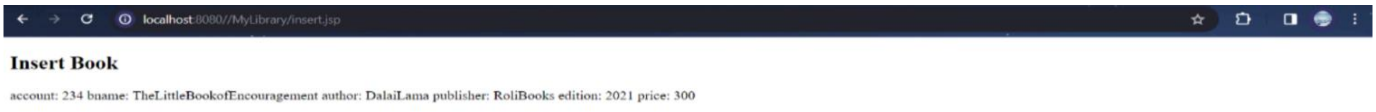
Welcome to Library Management System

This is Python Flask Based Library System

All Books

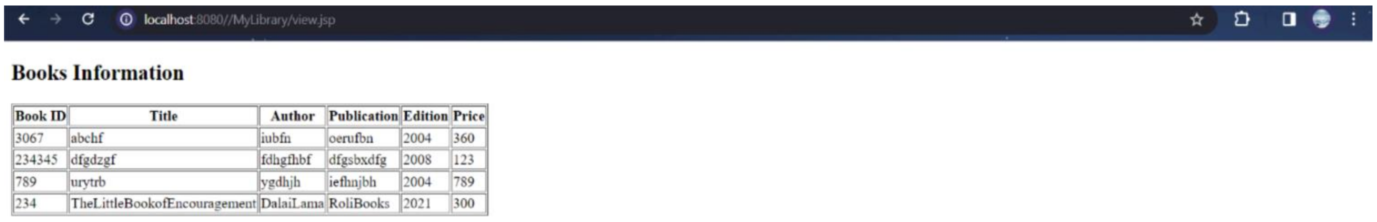
#	Name	Details
1	English	Details

© 2020 - Library Management System



Insert Book

account: 234 bname: TheLittleBookofEncouragement author: DalaiLama publisher: RoliBooks edition: 2021 price: 300



localhost:8080//MyLibrary/view.jsp

Books Information

Book ID	Title	Author	Publication	Edition	Price
3067	abchf	iubfn	oerufbn	2004	360
234345	dfgdzgf	fdhgfhbf	dfgsbxdfg	2008	123
789	urytrb	ygdhjh	iefhnjbh	2004	789
234	TheLittleBookofEncouragement	DalaiLama	RoliBooks	2021	300

RESULT: Thus the library management system has been successfully created

Ex.No. : 4b Date : Reg. No. :211701049

BANKING APPLICATION

AIM:

Program to develop a Banking application accessing a database using Servlet.

PROCEDURE:

Relations using MYSQL for a banking application given below enforcingprimary key and foreign key constraints:

CUSTOMER (CID, CNAME)

ACCOUNT (ANO, ATYPE, BALANCE, CID)

An account can be a savings account or a current account. Check ATYPE in ‘S’or ‘C’.

A customer can have both types of accounts. TRANSACTION (TID, ANO, TTYPE, TDATE, TAMOUNT)

TTYPE can be ‘D’ or ‘W’

D- Deposit; W – Withdrawal

1. Open MySQL.
2. Create a database.

```
mysql> create database banking; Query OK, 1
row affected (0.05 sec)
```

3. Connect to the database.

```
mysql> use banking;
Database changed
```

4. Create the following tables:

```
mysql> create table customer (cid integer, cname varchar(20),
-> primary key (cid));
```

index.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Banking Application</title>
</head>
<body>
<h1 align="center">Banking Application</h1>
<hr />
<a href="Customer.html">Customer Details</a>
<br />
<br />
<a href="Account.html">Account Details</a>
</body>
</html> Customer.html:
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Customer Details</title>
</head>
<body>
<h1 align="center">Customer Details</h1>
<hr />
<form action="AddCustomer" method="post">
```

```
<table>
<tr>
</tr>
<tr>
</tr>
<tr>
</tr>
</table>
<td>Customer Id. :</td>
<td><input type="text" name="cid"></td>
<td>Customer Name :</td>
<td><input type="text" name="cname"></td>
<td colspan="2" align="center"><input type="submit"value="Add
Customer"></td>
<br /> <a href="ViewCustomers">View All Customers</a>
</form>
</body>
</html> AddCustomer.java:
import java.io.IOException; import
java.io.PrintWriter; import
java.sql.Connection; import
java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.servlet.ServletException; import
javax.servlet.annotation.WebServlet; import
javax.servlet.http.HttpServlet; import
javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
* Servlet implementation class AddCustomer
*/ @WebServlet("/AddCustomer")
public class AddCustomer extends HttpServlet { private staticfinal long
serialVersionUID = 1L;Connection conn = null; PreparedStatement ps = null;
/**
        * @see HttpServlet#doPost(HttpServletRequest request,
        HttpServletResponse
        * response)
*/
protected        void        doPost(HttpServletRequest        request,
HttpServletResponseresponse)
throws ServletException, IOException {
//     TODO     Auto-generated     method     stub
response.setContentType("text/html");   PrintWriter
out = response.getWriter();out.println("<html>");
out.println("<head><title>Add Customer Details</title></head>");
out.println("<body>");
out.println("<h1 align='center'>Add Customer Details</h1>");out.println("<hr
/>");
try { Class.forName("com.mysql.cj.jdbc.Driver");
String URL = "jdbc:mysql://localhost:3306/banking";
conn = DriverManager.getConnection(URL, "root", "admin");
ps = conn.prepareStatement("insert customer values (?, ?)"); ps.setInt(1,
Integer.parseInt(request.getParameter("cid"))); ps.setString(2,
request.getParameter("cname"));
int res = ps.executeUpdate();if (res !=
0)
out.println("Customer Details Inserted
Successfully...");

else
out.println("Customer Details Insertion Failure...");ps.close();
conn.close();
} catch (Exception e) {
out.println(e);
}
out.println("<br />");
out.println("<a href='Customer.html'>Back</a>");out.println("</body></html>");
}
}
```

ViewCustomers.java:

```
import java.io.IOException; import
java.io.PrintWriter; import
java.sql.Connection; import
java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

import javax.servlet.ServletException; import

javax.servlet.annotation.WebServlet; import

javax.servlet.http.HttpServlet; import

javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/**

        * Servlet implementation class ViewCustomers

*/ @WebServlet("/ViewCustomers")

public class ViewCustomers extends HttpServlet { private staticfinal long

serialVersionUID = 1L;Connection conn = null; PreparedStatement ps = null;

ResultSet rs = null;

/**

        * @see HttpServlet#doPost(HttpServletRequest request,

        HttpServletResponse

        * response)

*/

protected void doGet(HttpServletRequest request,
HttpServletResponseresponse)

throws ServletException, IOException {

// TODO Auto-generated method stub

response.setContentType("text/html"); PrintWriterout =

response.getWriter(); out.println("<html>");

out.println("<head><title>View All Customer

Details</title></head>"); out.println("<body>");

out.println("<h1 align='center'>View All Customer Details</h1>");
out.println("<hr />");


try { Class.forName("com.mysql.cj.jdbc.Driver");
String URL = "jdbc:mysql://localhost:3306/banking";
conn = DriverManager.getConnection(URL, "root", "admin"); ps =conn.prepareStatement("select * from
customer order bycid");
rs = ps.executeQuery(); out.println("<table
border='1'>"); out.println("<tr>");
out.println("<td>Customer Id.</td>");
out.println("<td>Customer Name</td>");
out.println("<td>Edit</td>");
out.println("<td>Delete</td>"); out.println("</tr>");
while (rs.next()) {
out.println("<tr>");
out.println("<td>" + rs.getInt("cid") + "</td>"); out.println("<td>"
+ rs.getString("cname") + "</td>");out.println("<td><a href='EditCustomer?cid=" + rs.getInt("cid")
+ "'>Edit</a></td>");out.println("<td><a href='DeleteCustomer?cid=" + rs.getInt("cid")
+ "'>Delete</a></td>"); out.println("</tr>");
}
out.println("</table>");ps.close();
conn.close();
} catch (Exception e) {
out.println(e);
}
out.println("<br />");
out.println("<a href='Customer.html'>Back</a>");out.println("</body></html>");
}
}
```

EditCsutomer.java:

```
import java.io.IOException; import
java.io.PrintWriter; import
java.sql.Connection; import
java.sql.DriverManager;
import java.sql.PreparedStatement;import
java.sql.ResultSet;
import javax.servlet.ServletException; import
javax.servlet.annotation.WebServlet; import
javax.servlet.http.HttpServlet; import
javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
        * Servlet implementation class EditCustomer
*/ @WebServlet("/EditCustomer")
public class EditCustomer extends HttpServlet { private staticfinal long
serialVersionUID = 1L;Connection conn = null;


PreparedStatement ps = null;ResultSet
rs = null;
/**
        * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
        * response)
*/
protected       void       doGet(HttpServletRequest       request,
HttpServletResponseresponse)
throws ServletException, IOException {
//      TODO      Auto-generated      method      stub
response.setContentType("text/html");   PrintWriter
out = response.getWriter();out.println("<html>");
out.println("<head><title>Edit Customer Details</title></head>");
out.println("<body>");
out.println("<h1 align='center'>Edit All Customer Details</h1>");out.println("<hr />");
try { Class.forName("com.mysql.cj.jdbc.Driver");
String URL = "jdbc:mysql://localhost:3306/banking";
conn = DriverManager.getConnection(URL, "root", "admin");ps =conn.prepareStatement("select *
from customer where cid = ?"); ps.setInt(1,Integer.parseInt(request.getParameter(("cid"))));rs =
ps.executeQuery();
rs.next();
out.println("<form action='UpdateCustomer' method='post'>");
out.println("<table>");
out.println("<tr>"); out.println("<td>Customer Id.
:</td>");
out.println("<td><input type='text' name='cid' value='" +rs.getInt("cid")
+ "' readonly></td>"); out.println("</tr>");
out.println("<tr>");
out.println("<td>Customer Name :</td>"); out.println("<td><input type='text'
name='cname' value='" +rs.getString("cname") + "'></td>");out.println("</tr>");
out.println("<tr>");
out.println("<td colspan='2' align='center'><inputtype='submit'value='Update
Customer'></td>"); out.println("</tr>"); out.println("</table>");
out.println("</form>");ps.close();
conn.close();
} catch (Exception e) {
out.println(e);
}
out.println("<br />");
out.println("<a href='ViewCustomers'>Back</a>");out.println("</body></html>");
}
}
```

UpdateCustomer.java:

```
import java.io.IOException; import


java.io.PrintWriter; import
java.sql.Connection; import
java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.servlet.ServletException; import
javax.servlet.annotation.WebServlet; import
javax.servlet.http.HttpServlet; import
javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
        * Servlet implementation class UpdateCustomer
*/ @WebServlet("/UpdateCustomer")
public class UpdateCustomer extends HttpServlet { private staticfinal long
serialVersionUID = 1L;Connection conn = null; PreparedStatement ps = null;
/**
        * @see HttpServlet#doPost(HttpServletRequest request,
        HttpServletResponse response)
*/
protected void doPost(HttpServletRequest request,
HttpServletResponseresponse)
throws ServletException, IOException {


//    TODO    Auto-generated    method    stub
response.setContentType("text/html");  PrintWriter
out = response.getWriter();out.println("<html>");
out.println("<head><title>Update Customer Details</title></head>");
out.println("<body>");
out.println("<h1 align='center'>Update Customer Details</h1>");out.println("<hr
/>");
try { Class.forName("com.mysql.cj.jdbc.Driver");
String URL = "jdbc:mysql://localhost:3306/banking";
conn = DriverManager.getConnection(URL, "root", "admin"); ps =
conn.prepareStatement("update customer set cname = ?where cid = ?");ps.setString(1,
request.getParameter("cname")); ps.setInt(2,
Integer.parseInt(request.getParameter("cid")));int res = ps.executeUpdate();
if (res != 0)
out.println("Customer Details Updated
Successfully...");
else
out.println("Customer Details Updation Failure...");ps.close();
conn.close();
} catch (Exception e) {
out.println(e);
}
out.println("<br />");
out.println("<a href='ViewCustomers'>Back</a>");out.println("</body></html>");


}
}
```

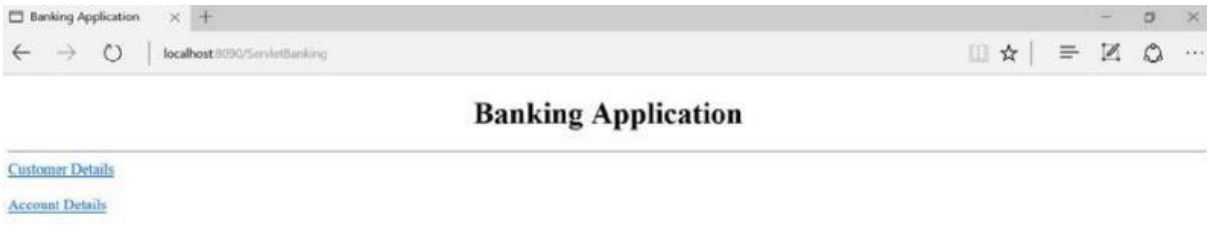
DeleteCustomer.java:

```
import java.io.IOException; import
java.io.PrintWriter; import
java.sql.Connection; import
java.sql.DriverManager;
import java.sql.PreparedStatement;
import javax.servlet.ServletException; import
javax.servlet.annotation.WebServlet; import
javax.servlet.http.HttpServlet; import
javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
/**

        * Servlet implementation class DeleteCustomer

*/ @WebServlet("/DeleteCustomer")

public class DeleteCustomer extends HttpServlet { private staticfinal long

serialVersionUID = 1L;Connection conn = null; PreparedStatement ps = null;

/**

        * @see HttpServlet#doGet(HttpServletRequest request,
        HttpServletResponseresponse)

*/

protected void doGet(HttpServletRequest request,
HttpServletResponseresponse)

throws ServletException, IOException {


//       TODO       Auto-generated       method       stub

response.setContentType("text/html"); PrintWriter  out

= response.getWriter();out.println("<html>");

out.println("<head><title>Delete Customer Details</title></head>");
out.println("<body>");

out.println("<h1 align='center'>Delete Customer Details</h1>");out.println("<hr

/>");

try { Class.forName("com.mysql.cj.jdbc.Driver");

String URL = "jdbc:mysql://localhost:3306/banking";

conn = DriverManager.getConnection(URL, "root", "admin"); ps =

conn.prepareStatement("delete from customer where cid =

?");

ps.setInt(1, Integer.parseInt(request.getParameter("cid")));int res =

ps.executeUpdate();

if (res != 0)

out.println("Customer Details Deleted uccessfully...");else

out.println("Customer Details Deletion Failure...");ps.close();

conn.close();

} catch (Exception e) {
```

```
out.println(e);
}
out.println("<br />");


out.println("<a href='ViewCustomers'>Back</a>");out.println("</body></html>");
}
}
```

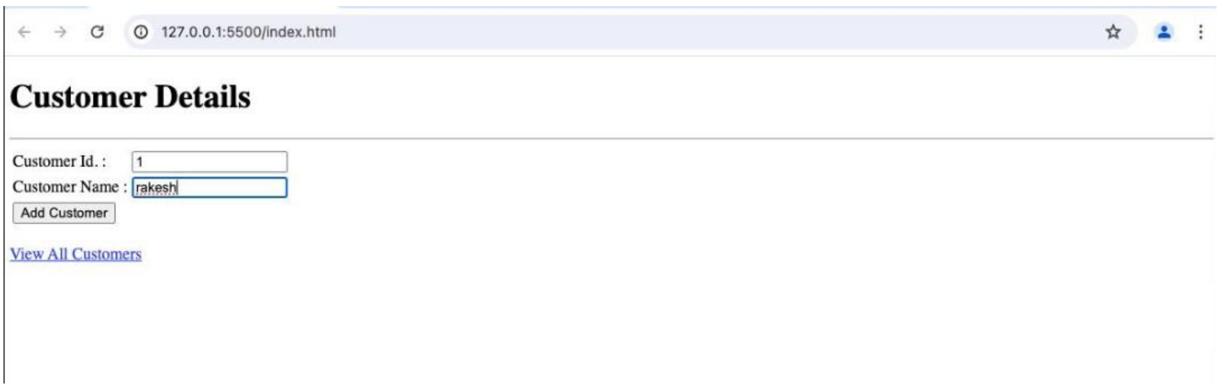
OUTPUT:



Banking Application

Customer Details

Account Details



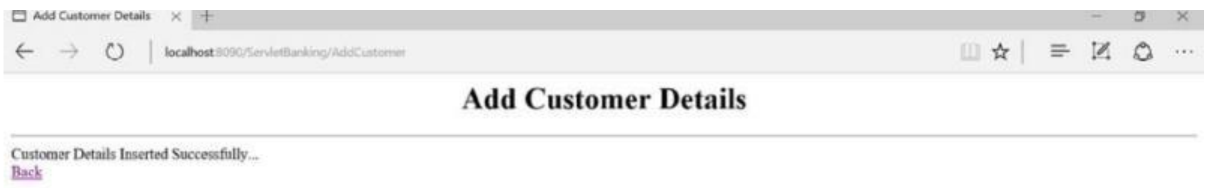
Customer Details

Customer Id. : 1

Customer Name : rakesh

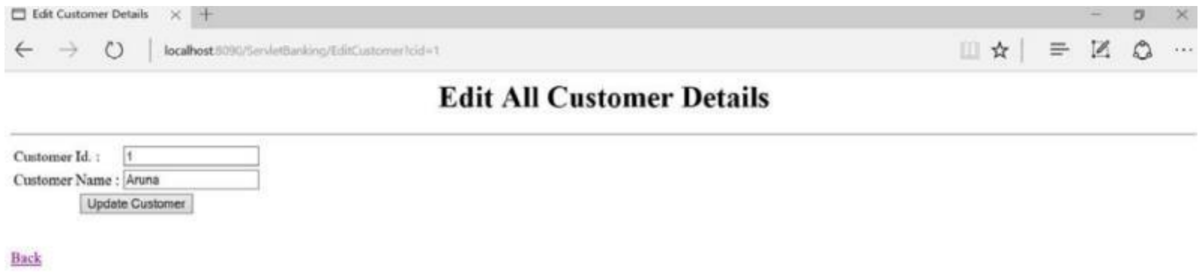
Add Customer

View All Customers



Add Customer Details

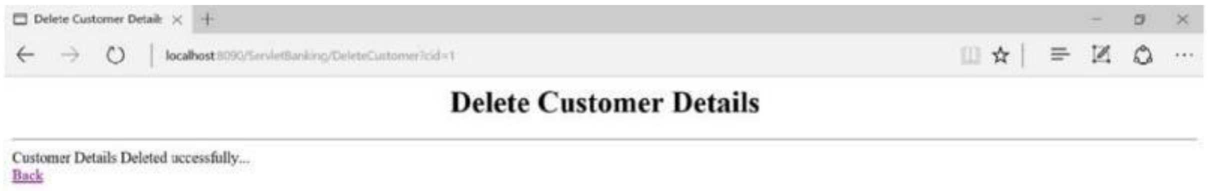
Customer Details Inserted Successfully...
Back



Edit All Customer Details

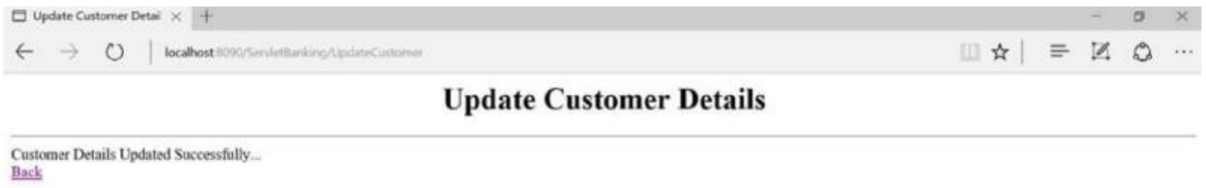
Customer Id. : 1
Customer Name : Aruna
Update Customer

Back



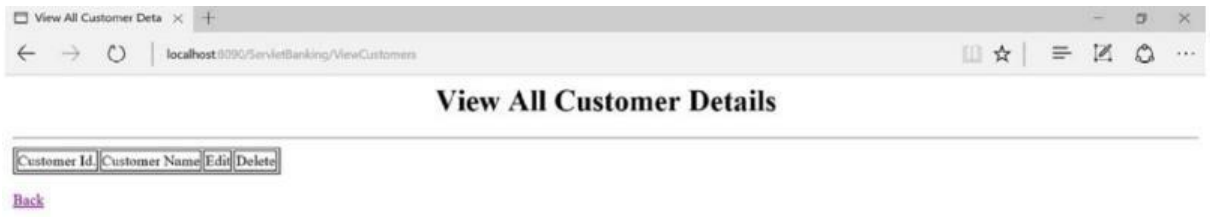
Delete Customer Details

Customer Details Deleted uccessfully...
Back



Update Customer Details

Customer Details Updated Successfully...
Back



View All Customer Details

Customer Id.	Customer Name	Edit	Delete

Back

RESULT:

Hence developed a Banking application accessing a database using Servlet isimplemented successfully.

Ex.No. : 5 Date : Reg. No. :211701049

PHP

AIM:

PHP program for Employee Details, which includes EmpID, Name, Designation, Salary, DOJ, etc., to connect with the database and execute queries to retrieve and update data.

PROCEDURE:

Relations using MYSQL for a banking application given below enforcing primary key constraints:EMPDETAILS (<u>EMPID</u>, ENAME, DESIG, DEPT, DOJ, SALARY)

1. Open MySQL.
2. Create a database.
3. Connect to the database.
4. Create a table
5. Create a connection using mysqli_connect().
6. Create a page with links for the view page and insert form.
7. Create a form to insert employee details and using mysqli_query() the query is executed.
8. Using mysqli_query() select statement is executed and the details are displayed in a tableformat
9. For editing the form , update statement is executed and updated in the database.
10. For deleting an employee details,delete statement is used

CODE:

Index.html:

```
<html>

<head>

<title>Banking Application</title>

<style>li{

list-style-type: none;

}

.hi li a:hover{ font-size:

x-large;padding: 5px;
color: red;

background-color: aquamarine;

}

a{

text-decoration: none;

}

</style>

</head>

<body>

<h1 align="center">Banking Application</h1>

<hr>

<div class="hi">

<li><a href="show.php">Retrive Employee Details</a></li>

<br><br>

<li><a href="update.html">Update Employee Details</a></li>

<br><br>

<li><a href="insert.html">Insert Employee Details</a></li>

</div>

</body>

</html>
```

Insert.html:

```
<html>

<head>

<title>Insert Details</title>

<style>

button:hover{
```

```
background-color: blueviolet;
}

</style>

</head>

<body>

<form action="insert.php" method="post">

Employee Id: <input type="number" id="eid" name="eid"><br><br> Employee

Name: <input type="text" id="ename" name="ename"><br><br>Designation:

<input type="text" id="desig" name="desig"><br><br> Department: <input

type="text" id="dept" name="dept"><br><br>

DOJ: <input type="date" id="doj" name="doj"><br><br> Salary:

<input type="number" id="sal" name="sal"><br><br>

<button type="submit">INSERT Details</button>

<br>

</form>

</body>

</html>
```

Update.html:

```
<html>

<head>

<title>Update Details</title>

<style>

button:hover{

background-color: blueviolet;

}

</style>
```

```
</head>

<body>
<form action="update.php" method="post">

Employee Id: <input type="number" id="eid" name="eid"><br><br> Employee

Name: <input type="text" id="ename" name="ename"><br><br>Designation:

<input type="text" id="desig" name="desig"><br><br>


Department: <input type="text" id="dept" name="dept"><br><br>DOJ:

<input type="date" id="doj" name="doj"><br><br>

Salary: <input type="number" id="sal" name="sal"><br><br>

<button type="submit">UPDATE Details</button>

<br>

</form>

</body>

</html>
```

Insert.php:

```
<?php

$my=new mysqli("localhost","root","","empdetails");

$Eid=$_POST['eid'];

$EName=$_POST['ename'];

$Designation=$_POST['desig'];

$Department=$_POST['dept'];

$DOJ=$_POST['doj'];

$Salary=$_POST['sal'];

$sql="insert into empdet(EMPID,ENAME,DESIG,DEPT,DOJ,SALARY)

values('$Eid','$EName','$Designation','$Department','$DOJ','$Salary')";
```

```
$final=$my->query($sql);

if($final==true)

echo "INSERT SUCESSEFULLY";
else

echo "INSERT UNSUCESSEFULLY";


?>
```

Update.php:

```
<?php

$my = new mysqli("localhost", "root", "", "empdetails");if

($my->connect_error) {

die("Connection failed: " . $my->connect_error);

}

$Eid = $_POST['eid'];

$EName = $_POST['ename'];

$Designation = $_POST['desig'];

$Department = $_POST['dept'];

$DOJ = $_POST['doj'];

$Salary = $_POST['sal'];

$sql = $my->prepare("UPDATE empdet SET ENAME=?, DESIG=?, DEPT=?, DOJ=?,
SALARY=? WHERE EMPID=?");

$sql->bind_param("sssssi", $EName, $Designation, $Department, $DOJ, $Salary, $Eid);

$final = $sql->execute();if

($final === true) {

echo "UPDATE SUCCESSFULLY";

} else {
```

```
echo "UPDATE UNSUCCESSFULLY";

}
$sql->close();

$my->close();

?>
```

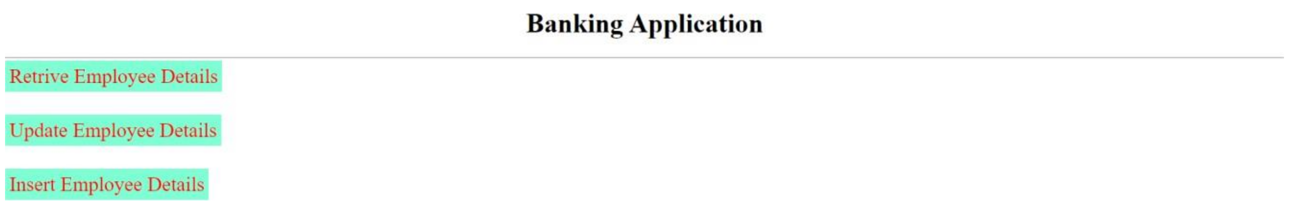
Show.php:

```
<?php

$my = new mysqli("localhost", "root", "", "empdetails");if

($my->connect_error) {

die("Connection failed: " . $my->connect_error);

}

$sql = 'SELECT * FROM empdet';

$retval = mysqli_query($my, $sql); if

(mysqli_num_rows($retval) > 0) {

while ($row = mysqli_fetch_assoc($retval)) { echo

"Employee ID : {$row['EMPID']} <br> " ."Employee

Name : {$row['ENAME']} <br> " . "Designation :

{$row['DESIG']} <br> " . "Department:

{$row['DEPT']} <br> " .

"DOJ : {$row['DOJ']} <br> " .

"Salary : {$row['SALARY']} <br> " ."

-----------------------------------------------------------<br>";

}

} else {

echo "0 results";

}
```

```
mysqli_close($my);

?>
```

DESIGN:

Main page:



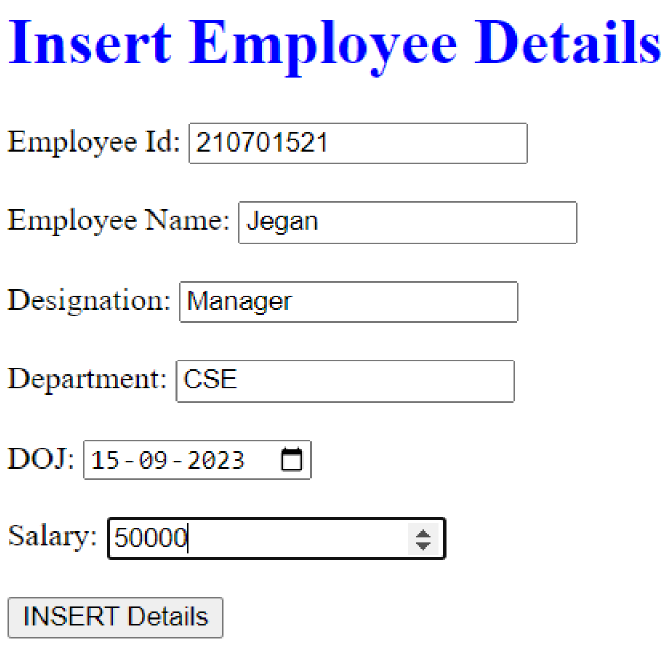
Banking Application

Retrive Employee Details

Update Employee Details

Insert Employee Details

Insert Page:



Insert Employee Details

Employee Id: 210701521

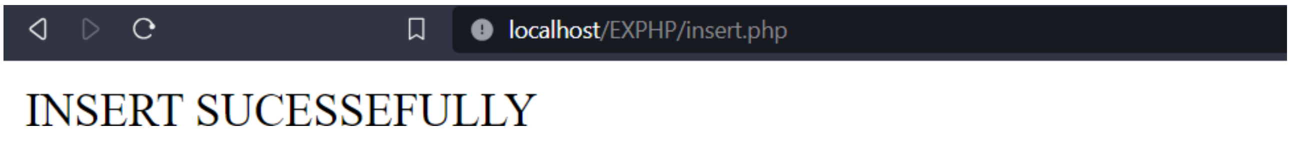
Employee Name: Jegan

Designation: Manager

Department: CSE

DOJ: 15-09-2023

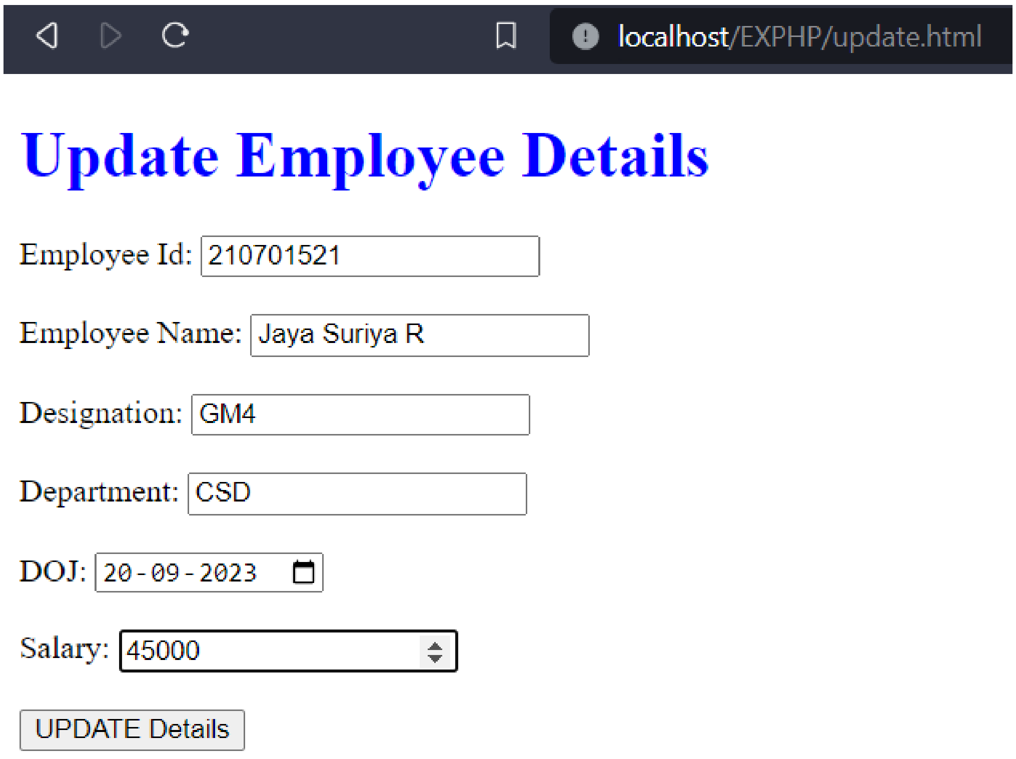
Salary: 50000



localhost/EXPHP/insert.php

INSERT SUCESSEFULLY

Update page:


localhost/EXPHP/update.html

Update Employee Details

Employee Id: 210701521

Employee Name: Jaya Suriya R

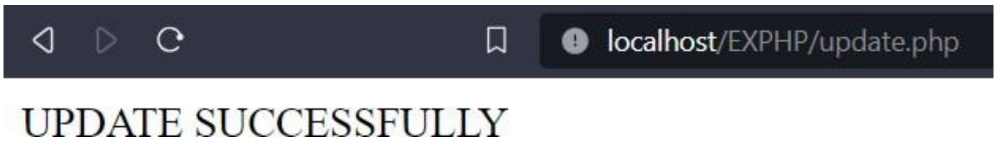
Designation: GM4

Department: CSD

DOJ: 20-09-2023

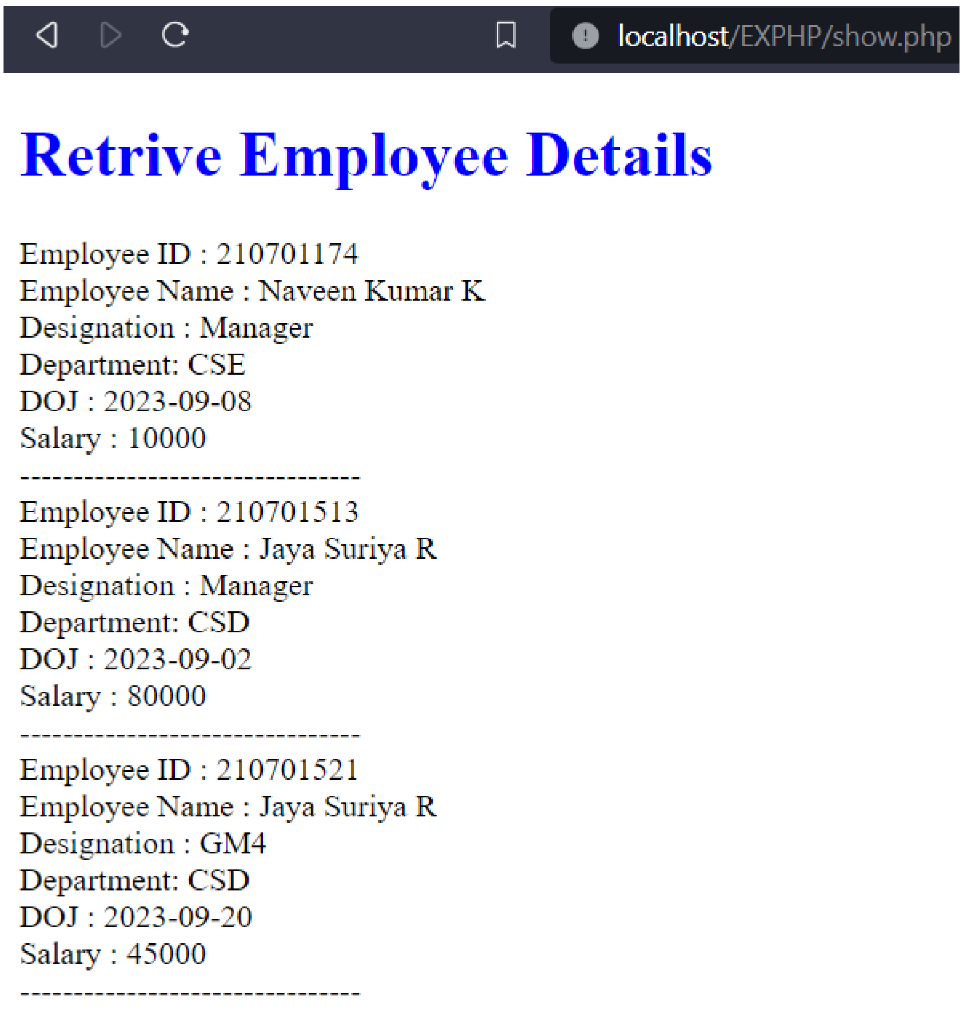
Salary: 45000

UPDATE Details


localhost/EXPHP/update.php

UPDATE SUCCESSFULLY

Retrieve Page



```
Retrive Employee Details

Employee ID : 210701174
Employee Name : Naveen Kumar K
Designation : Manager
Department: CSE
DOJ : 2023-09-08
Salary : 10000
--------------------------------
Employee ID : 210701513
Employee Name : Jaya Suriya R
Designation : Manager
Department: CSD
DOJ : 2023-09-02
Salary : 80000
--------------------------------
Employee ID : 210701521
Employee Name : Jaya Suriya R
Designation : GM4
Department: CSD
DOJ : 2023-09-20
Salary : 45000
--------------------------------
```

RESULT: Thus the above mentioned PHP Program for Employee Details was Successfully created.

Ex.No. : 6 Date : Reg. No. :211701049

BOOTSTRAP

AIM:

Program to develop an attractive web pages using Bootstrap.

PROCEDURE:

1. Ensuring you get a responsive Bootstrap website is as simple as placing the correct metatag inside the head of your web pages:

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

2. The above meta tag is quite self-explanatory in nature. We’re setting the width of the pageto the width of the device and initially scaling it to 1 — its default size.

3. Apart from this, you’re good to go: Bootstrap is responsive by default.

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

   <link href="https://fonts.googleapis.com/css?family=Saira+Extra+Condensed:
   500,700" rel="stylesheet">

   <link href="https://fonts.googleapis.com/css?family=Muli:400,400i,800,800i"rel="stylesheet">

   <link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet">

   PROGRAM:

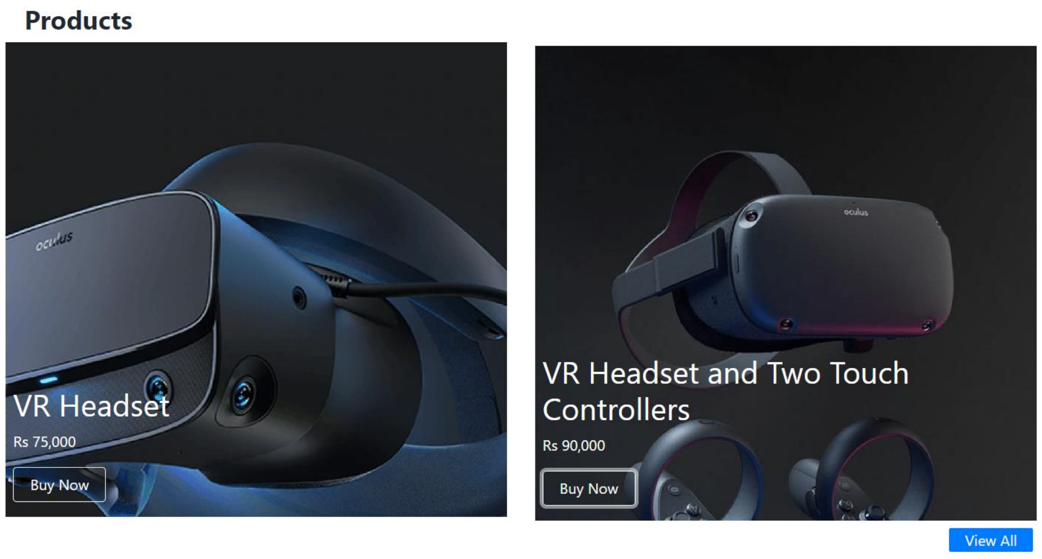
```
<!DOCTYPE html>

<html>



<head>

   <link rel="stylesheet"

href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css

" integrity="sha384-

JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGM
```

```
N5t9UJ0Z" crossorigin="anonymous" />
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+Or
CXaRkfj" crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu7
35Sk7lN" crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMM
V+rV" crossorigin="anonymous"></script>
</head>

<body>
    <div class="vr-products-web bg-white">
        <div class="container">
            <div class="row">
                <div class="col-12">
                    <h1 class="vr-web-title pt-4 pl-4">Products</h1>
                </div>
                <div class="col-12 col-md-6">
                <div class="vr-product-card-1 ml-1 d-flex flex-column justify-
content-end pl-2">
                  <h1 class="vr-product-heading">VR Headset
CSS
@import
url('https://fonts.googleapis.com/css2?family=Bree+Serif&family=Caveat:wgh
t@400;700&family=Lobster&family=Monoton&family=Open+Sans:ital,wght
@0,400;0,700;1,400;1,700&family=Playfair+Display+SC:ital,wght@0,400;0,
700;1,700&family=Playfair+Display:ital,wght@0,400;0,700;1,700&family=R
oboto:ital,wght@0,400;0,700;1,400;1,700&family=Source+Sans+Pro:ital,wgh
t@0,400;0,700;1,700&family=Work+Sans:ital,wght@0,400;0,700;1,700&disp
lay=swap');

.vr-product-card-1 {
  text-align: left;
  color: white;
  background-image: url("https://d1tgh8fmlzexmh.cloudfront.net/ccbp-
responsive-website/vr-products-headset-img.png");
  background-size: cover;
  width: 100%;
  height: 510px;
}

.vr-web-image {
  width: 100%;
```

```
}

.vr-web-title {
   color: #1f2933;
   font-size: 28px;
   font-weight: bold;
}
.vr-product-heading {
   font-size: 33px;
   font-weight: normal;
}
.vr-product-price {
   color: white;
   font-size: 16px;
}
.buy-now-button {
   width: 100px;
}
.vr-product-card-2 {
   text-align: left;
   color: white;
   background-image: url("https://d1tgh8fmlzexmh.cloudfront.net/ccbp-
responsive-website/vr-products-headset-with-controllers-img.png");
   background-size: cover;
   width: 100%;
```

```
 height: 510px;

}

.view-all-button {

    border-width: 0px;

    margin-right: auto;

    border-radius: 2px;

    width: 90px;

}
```

OUTPUT:



RESULT: Thus an attractive web pages using Bootstrap has been completed successfully

Ex.No. : 7 Date : Reg. No. :211701049

DESIGN A WEB PAGE WITH - NAVIGATION MENU

AIM:

Program to design a web page with navigation menus using Angular JS.

PROCEDURE:

1. Using Angular's directives to set and read the active variable.
2. When it changes, it causes the HTML that uses it to be updated automatically.
3. In Angular's terminology, this variable is called a model. It is available to all directives in the current scope, and can be accessed in your controllers (more on that in the nextexample).
4. JavaScript templates are with the {{var}} syntax, the framework sees such a string, it replaces it with the contents of the variable.
5. This operation is repeated every time var is changed.

PROGRAM:

index.html

```
<!DOCTYPE html>
<html>

      <head>
            <meta charset="utf-8"/>
            <title>Learn AngularJS - Navigation Menu</title>

            <link href="http://fonts.googleapis.com/css?family=Open+Sans:400,700"
                  rel="stylesheet" />

            <!-- The main CSS file -->
            <link href="style.css" rel="stylesheet" />

            <!--[if lt IE 9]>
         <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
         </script>
            <![endif]-->
      </head>

      <!-- The ng-app directive tells angular that the code below shouldbe evaluated -->

      <body ng-app>

            <!--The navigation menu will get the value of the "active" variable as a class. The
```

```
        $event.preventDefault() stops the page from jumping whena link is clicked.        -->

           <nav class="{{active}}" ng-click="$event.preventDefault()">

        <!-- When a link in the menu is clicked, we set the active variable -->

        <a href="#" class="home" ng-click="active='home'">Home</a>
        <a href="#" class="projects" ng-click="active='projects'">Projects</a>
        <a href="#" class="services" ng-click="active='services'">Services</a>
        <a href="#" class="contact" ng-click="active='contact'">Contact</a>
        </nav>

        <!-- ng-show will show an element if the value in the quotes is truthful,while ng-hide does
                 the opposite. Because the active variable is
                 not set initially, this will cause the first paragraph to bevisible. -->

           <p ng-hide="active">Please click a menu item</p>
           <p ng-show="active">You chose <b>{{active}}</b></p>

           <!-- Include AngularJS from Google's CDN -->
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js">
</script>
      </body>
</html>
```

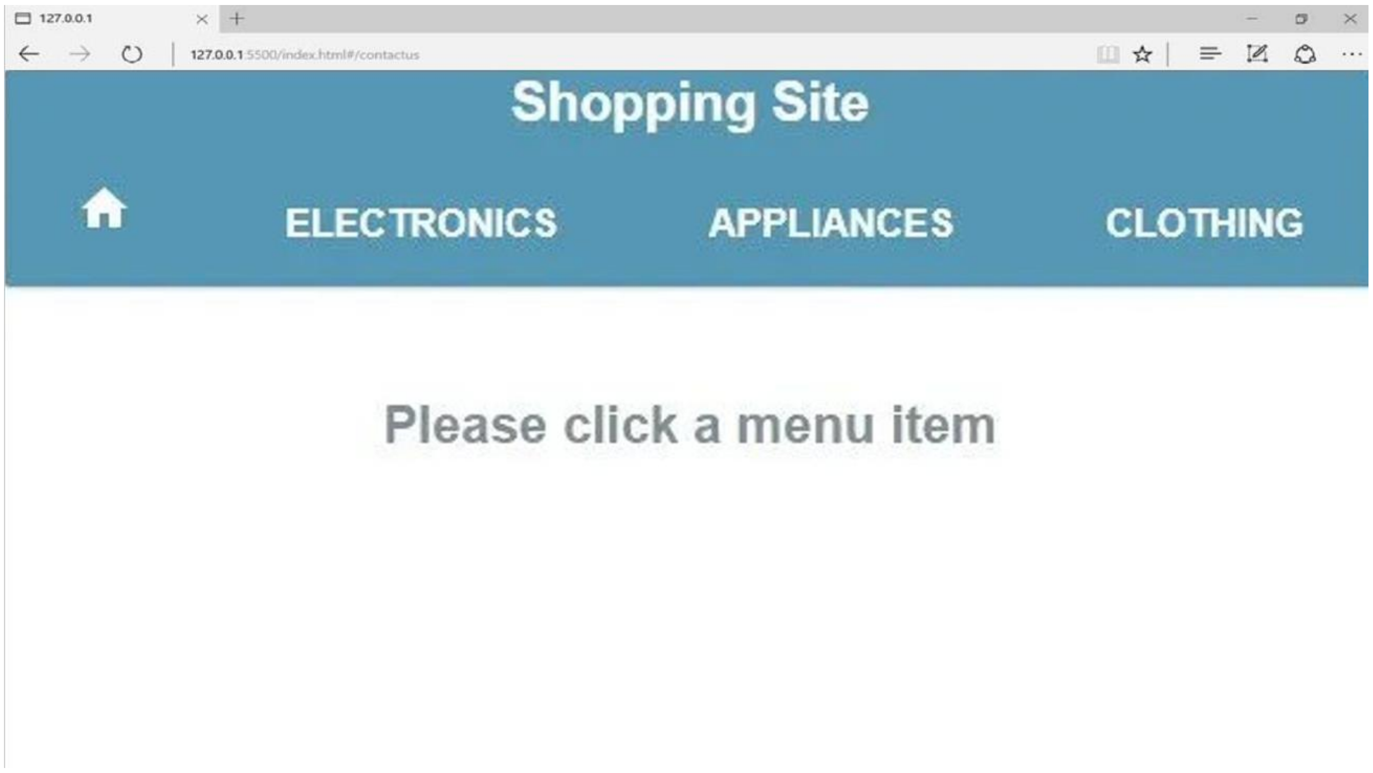
style.css

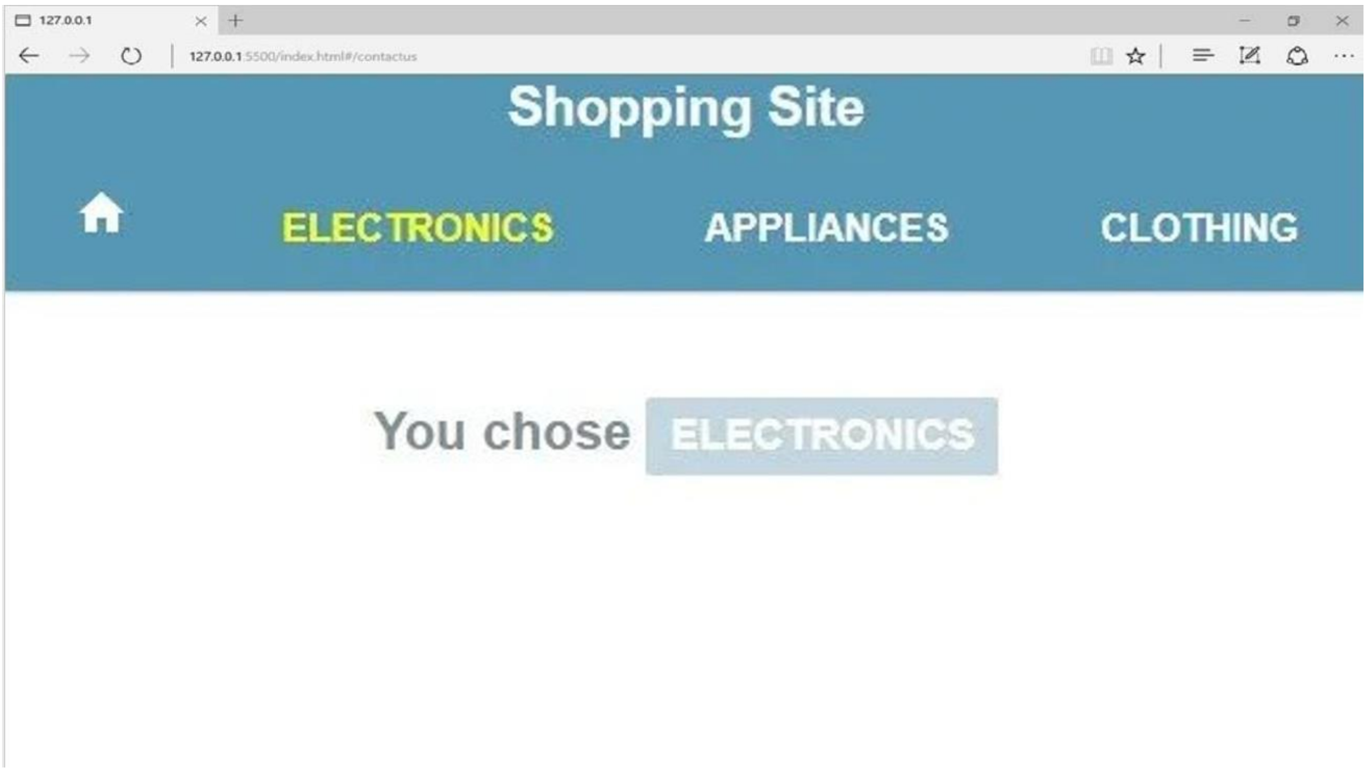
```
/* -------------------------------------------
      Simple reset
---------------------------------------------*/


*{
      margin:0;
      padding:0;
}

/* -------------------------------------------
      General Styles
---------------------------------------------*/

body{
      font:15px/1.3 'Open Sans', sans-serif;color:
      #5e5b64;
      text-align:center;
}

a, a:visited {
      outline:none;
      color:#389dc1;
}
```

```
a:hover{
        text-decoration:none;
}

section, footer, header, aside, nav{display:
        block;
}

/* -------------------------------------------
        The menu
--------------------------------------------*/

nav{
        display:inline-block;
        margin:60px auto 45px;
        background-color:#5597b4;
        box-shadow:0 1px 1px #ccc;
        border-radius:2px
```

```
nav a{
        display:inline-block;
        padding: 18px 30px;
        color:#fff !important;font-
        weight:bold;
        font-size:16px;
        text-decoration:none !important;line-
        height:1;
        text-transform: uppercase;
        background-color:transparent;

        -webkit-transition:background-color  0.25s;
        -moz-transition:background-color 0.25s;
        transition:background-color 0.25s;
  }


  nav a:first-child{
        border-radius:2px 0 0 2px;
  }

  nav a:last-child{
        border-radius:0 2px 2px 0;
  }

  nav.home .home,
  nav.projects .projects,
  nav.services .services,
  nav.contact .contact{
        background-color:#e35885;
  }

  p{
        font-size:22px; font-
        weight:bold;
        color:#7d9098;
  }

  p b{ color:#ffffff;
        display:inline-block;
        padding:5px 10px;
        background-color:#c4d7e0;
        border-radius:2px;
        text-transform:uppercase;font-
        size:18px;
  }
```

OUTPUT:





RESULT:

Hence designed a web page with navigation menus using Angular JS.

INLINE EDITOR

AIM:

Program to design a web page with inline editor using Angular JS.

Procedure:

1. Clicking a paragraph will show a tooltip with a text field.
2. Use a controller that will initialize the models and declare two methods for toggling the visibility of the tooltip.
3. Controllers are regular JavaScript functions which are executed automatically by Angular,and which are associated with your page using the ng-controller directive.
4. When the controller function is executed, it gets the special $scope object as a parameter.
5. Adding properties or functions to it makes them available to the view.
6. Using the ng-model binding on the text field tells Angular to update that variable when thevalue of the field changes (this in turn re-renders the paragraph with the value).

PROGRAM:

index.html

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8"/>
        <title>Learn AngularJS - Inline Editor</title>

        <link href="http://fonts.googleapis.com/css?family=Open+Sans:400,700"
      rel="stylesheet" />

        <!-- The main CSS file -->
        <link href="style.css" rel="stylesheet" />

        <!--[if lt IE 9]>
      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
      </script>
        <![endif]-->
    </head>

    <!-- Notice the controller directive -->
    <body ng-app ng-controller="InlineEditorController">

        <!-- When this element is clicked, hide the tooltip -->
        <div id="main" ng-click="hideTooltip()">

            <!-- This is the tooltip. It is shown only when the showtooltipvariable is
```

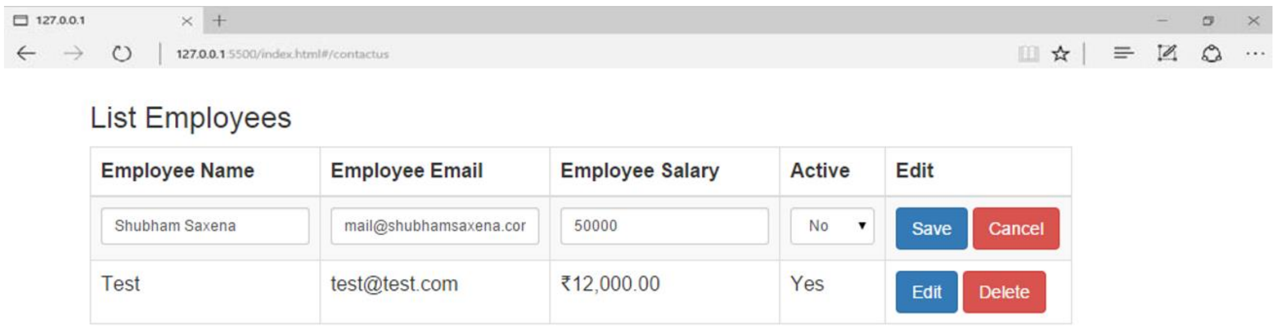
```
truthful -->

<table class="table table-striped table-bordered">
 <thead>
  <th>Employee Name</th>
  <th>Employee Email</th>
  <th>Employee Salary</th>
  <th>Active</th>
  <th>Edit</th>
 </thead>
 <tbody>
  <tr ng-repeat="employee in employees" ng-
include="getTemplate(employee)">
   <script type="text/ng-template" id="display">
    <td>{{employee.empName}}</td>
    <td>{{employee.empEmail}}</td>
    <td>{{employee.empSalary | currency:"₹"}}</td>
    <td>{{employee.active | active}}</td>
    <td>
     <button type="button" class="btn btn-primary" ng-
click="editEmployee(employee)">Edit</button>
     <button type="button" class="btn btn-danger" ng-
click="deleteEmployee(employee)">Delete</button>
    </td>
   </script>
   <script type="text/ng-template" id="edit">
    <td><input type="text" ng-model=employee.empName
class="form-control input-sm"/></td>
    <td><input type="text" ng-model=employee.empEmail
class="form-control input-sm"/></td>
    <td><input type="text" ng-model=employee.empSalary
class="form-control input-sm"/></td>
    <td>
      <select class="form-control input-sm" ng-
model=employee.active>
        <option value='1'>Yes</option>
        <option value='0'>No</option>
      </select>
    </td>
    <td>
     <button type="button" class="btn btn-primary" ng-
click="updateEmployee(employee)">Save</button>
     <button type="button" class="btn btn-danger" ng-
```

```
                    click="reset()">Cancel</button>
                       </td>
                      </script>


                     </tr>
                     </tbody>
                    <table>// When a model is changed, the view will be automatically
              // updated by by AngularJS. In this case it will hide the tooltip.

              $scope.showtooltip = false;
       }

       $scope.toggleTooltip = function(e){
              e.stopPropagation();
              $scope.showtooltip = !$scope.showtooltip;
       }
```

style.css

```
/* -------------------------------------------
       Simple reset
--------------------------------------------*/

*{
       margin:0;
       padding:0;
}

/* -------------------------------------------
       General Styles
--------------------------------------------*/

body{
       font:15px/1.3 'Open Sans', sans-serif;color:
       #5e5b64;
       text-align:center;
}

a, a:visited {
       outline:none;
       color:#389dc1;
}

a:hover{
       text-decoration:none;
}
```

```
section, footer, header, aside, nav{display:
        block;
}

/* -------------------------------------------
        The edit tooltip
--------------------------------------------*/

.tooltip{
        background-color:#5c9bb7;

        background-image:-webkit-linear-gradient(top, #5c9bb7, #5392ad);background-
        image:-moz-linear-gradient(top, #5c9bb7, #5392ad); background-image:linear-
        gradient(top, #5c9bb7, #5392ad);

        box-shadow: 0 1px 1px #ccc;
        border-radius:3px;
        width: 290px;
        padding: 10px;
        position: absolute;
        left:50%;
        margin-left:-150px;top:
        80px;
}

.tooltip:after{ content:'';
        position:absolute;
        border:6px solid #5190ac;
        border-color:#5190ac transparent transparent;width:0;
        height:0;
        bottom:-12px;
        left:50%;
        margin-left:-6px;
}

.tooltip input{
        border: none;
        width: 100%;
        line-height: 34px;
        border-radius: 3px;
        box-shadow: 0 2px 6px #bbb inset;text-
        align: center;
        font-size: 16px;
        font-family: inherit;color:
        #8d9395;
        font-weight: bold;
        outline: none;
}

p{
        font-size:22px; font-
        weight:bold;
        color:#6d8088;
        height: 30px;
        cursor:default;
}

p b{ color:#ffffff;
        display:inline-block;
        padding:5px 10px;
        background-color:#c4d7e0;
        border-radius:2px;
        text-transform:uppercase;font-
        size:18px;
}
p:before{
        content:'✎';
        display:inline-block;
        margin-right:5px; font-
        weight:normal;
        vertical-align: text-bottom;
}
#main{
        height:300px;
        position:relative;
        padding-top: 150px;
```



List Employees

Employee Name	Employee Email	Employee Salary	Active	Edit
Shubham Saxena	mail@shubhamsaxena.cor	50000	No	Save Cancel
Test	test@test.com	₹12,000.00	Yes	Edit Delete

RESULT: Hence designed a web page with inline editor using Angular JS.

ORDER FORM

AIM:

Program to design a web page with order form using Angular JS.

PROCEDURE:

1. Code an order form with a total price updated in real time, using another one of Angular's useful features - filters.

2. Filters let modify models and can be chained together using the pipe character |.

3. Use the currency filter, to turn a number into a properly formatted price, complete with a dollar sign and cents. You can easily make your own filters.

4. The ng-repeat binding (docs) is another useful feature of the framework. It lets loopthrough an array of items and generate markup for them. It is intelligently updated when an item is changed or deleted.

PROGRAM:

index.html

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8"/>
        <title>Learn AngularJS - Order Form</title>

        <link href="http://fonts.googleapis.com/css?family=Cookie|Open+Sans:400,700"
      rel="stylesheet" />

        <!-- The main CSS file -->
        <link href="style.css" rel="stylesheet" />

        <!--[if lt IE 9]>
      <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
      </script>
        <![endif]-->
    </head>

    <!-- Declare a new AngularJS app and associate the controller -->
    <body ng-app ng-controller="OrderFormController">

        <form>

            <h1>Services</h1>

            <ul>
                <!-- Loop through the services array, assign a click handler,and set or
```

```
                    click="toggleActive(service)"
                    ng-class="{active:service.active}">
                        <!-- Notice the use of the currency filter, it willformat the price
                            -->
                        {{service.name}} <span>{{service.price | currency}}</span>
                </li>
            </ul>

            <div class="total">
                <!-- Calculate the total price of all chosen services.
                    Format it as currency. -->
                Total: <span>{{total() | currency}}</span>
            </div>

        </form>

        <!-- Include AngularJS from Google's CDN -->
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js">
</script>
        <script src="script.js"></script>
    </body>
</html>
```

script.js

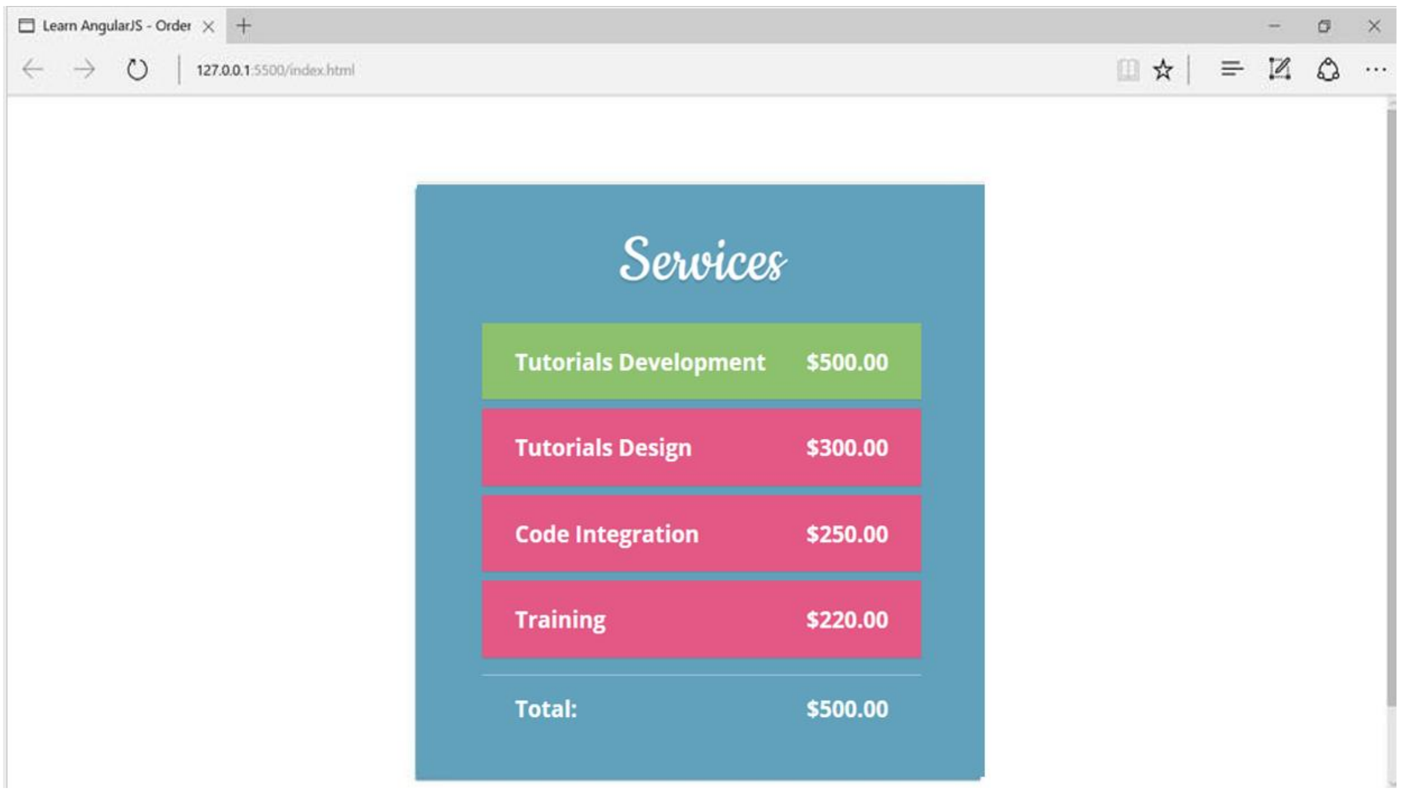
```
function OrderFormController($scope){

    // Define the model properties. The view will loop
    // through the services array and genreate a li
    // element for every one of its items.

    $scope.services = [
        {
            name: 'Web Development',
            price: 300,
            active:true
         },{
            name: 'Design',
            price: 400,
            active:false
         },{
            name: 'Integration',price:
            250, active:false

         },{ name: 'Training',
            price: 220,
            active:false

        }
    ];
    }

        $scope.toggleActive = function(s){s.active
              = !s.active;
        };

        // Helper method for calculating the total price

        $scope.total = function(){var

              total = 0;

              // Use the angular forEach helper method to
              // loop through the services array:

              angular.forEach($scope.services, function(s){if
                    (s.active){
                          total+= s.price;
                    }
              });

              return total;
        };
    }
```

style.css

```
    /* -------------------------------------------
          Simple reset
    -------------------------------------------*/

    *{
          margin:0;
          padding:0;
    }

    /* -------------------------------------------
          General Styles
    -------------------------------------------*/

    body{
          font:15px/1.3 'Open Sans', sans-serif;color:
          #5e5b64;
          text-align:center;
    }

    a, a:visited {
          outline:none;
          color:#389dc1;
    }

    a:hover{
        text-decoration:none;
}
section, footer, header, aside, nav{display:
        block;
}

/* ------------------------------------------
        The order form
-------------------------------------------*/

form{
        background-color: #61a1bc;
        border-radius: 2px;
        box-shadow: 0 1px 1px #ccc;
        width: 400px;
        padding: 35px 60px;
        margin: 80px auto;
}

form h1{
        color:#fff; font-
        size:64px;
        font-family:'Cookie', cursive;font-
        weight: normal;
        line-height:1;
        text-shadow:0 3px 0 rgba(0,0,0,0.1);
}

form ul{
        list-style:none;
        color:#fff; font-
        size:20px;
        font-weight:bold; text-
        align: left; margin:20px
        0 15px;
}

form ul li{
        padding:20px 30px;
        background-color:#e35885;
        margin-bottom:8px;
        box-shadow:0 1px 1px rgba(0,0,0,0.1);
        cursor:pointer;
}

form ul li span{
        float:right;
}

form ul li.active{
        background-color:#8ec16d;
```

```
}

div.total{
        border-top:1px solid rgba(255,255,255,0.5);
        padding:15px 30px;
        font-size:20px; font-
        weight:bold;text-
        align: left;color:#fff;
}

div.total span{
        float:right;
}
```

OUTPUT:



Services

Tutorials Development	$500.00
Tutorials Design	$300.00
Code Integration	$250.00
Training	$220.00
Total:	$500.00

RESULT: Hence designed a web page with order form using Angular JS.

INSTANT SEARCH

AIM:

Program to design a web page with instant search using Angular JS.

PROCEDURE:

1. To filter a list of items by typing into a text field.
2. First have to turn the application into a module.
3. Modules are a way of organizing JavaScript applications into self-contained componentsthat can be combined in new and interesting ways.
4. Angular relies on this technique for code isolation and requires that your applicationfollows it before you can create a filter.
5. There are only two things that you need to do to turn your app into a module:
    1. Use the angular.module("name",[]) function call in your JS. This will instantiate and return a new module;
    2. Pass the name of the module as the value of the ng-app directive.
6. Creating a filter then is as simple as calling the filter() method on the module object returned by angular.module("name", []).
7. Filters follow the Angular.js philosophy - every piece of code that you write should be self-contained, testable and reusable.
8. Use this filter in all your views and even combine it with others through chaining.

PROGRAM:

index.html

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="utf-8"/>
        <title>Learn AngularJS - Instant Search</title>

        <link href="http://fonts.googleapis.com/css?family=Cookie|Open+Sans:400,700"
    rel="stylesheet" />

        <!-- The main CSS file -->
        <link href="style.css" rel="stylesheet" />

        <!--[if lt IE 9]>
    <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
```

```
            </script>
               <![endif]-->
         </head>

         <!-- Initialize a new AngularJS app and associate it with a module named"instantSearch"--
            >
         <body ng-app="instantSearch" ng-controller="InstantSearchController">

               <div class="bar">
                     <!-- Create a binding between the searchString model and thetext field -->
                     <input type="text" ng-model="searchString"
                     placeholder="Enter your search terms" />
               </div><!-- Render a li element for every entry in the items array. Noticethe custom search filter
               "searchFor". It takes the value of the searchString model as an argument. -->
<li ng-repeat="i in items | searchFor:searchString">
      <a href="{{i.url}}"><img ng-src="{{i.image}}" /></a>
      <p>{{i.title}}</p>
                     </li>
               </ul>

               <!-- Include AngularJS from Google's CDN -->
            <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/
            angular.min.js"></script>
               <script src="script.js"></script>
         </body>
   </html>
```

script.js

```
   // Define a new module for our app
   var app = angular.module("instantSearch", []);
   / Create the instant search filter

   app.filter('searchFor', function(){

         // All filters must return a function. The first parameter
         // is the data that is to be filtered, and the second is an
         // argument that may be passed with a colon (searchFor:searchString)return

         function(arr, searchString){

               if(!searchString){
                     return arr;
               }

               var result = [];

               searchString = searchString.toLowerCase();

               // Using the forEach helper method to loop through the array
               angular.forEach(arr, function(item){
```

```
                        if(item.title.toLowerCase().indexOf(searchString) !== -1){
                                result.push(item);
                        }

                });

                return result;
        };

});

// The controller

function InstantSearchController($scope){

        // The data model. These items would normally be requested via AJAX,
        // but are hardcoded here for simplicity. See the next example for
        // tips on using AJAX.

        $scope.items = [
                {
                        url: 'http://tutorialzine.com/2013/07/50-must-have-plugins-for-extending-
twitter-bootstrap/',
                        title: '50 Must-have plugins for extending Twitter Bootstrap',image:
                        'http://cdn.tutorialzine.com/wp-
content/uploads/2013/07/featured_4-100x100.jpg'
                },



                {
                        url:  'http://tutorialzine.com/2013/08/simple-registration-system-
```

```
php-mysql/',
                title: 'Making a Super Simple Registration System With PHP and MySQ
L',
                image: 'http://cdn.tutorialzine.com/wp-


content/uploads/2013/08/simple_registration_system-100x100.jpg'
            },
            {
                url: 'http://tutorialzine.com/2013/08/slideout-footer-css/', title: 'Create a slide-out
                footer with this neat z-index trick',image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/08/slide-out-footer-100x100.jpg'
            },
            {
                url: 'http://tutorialzine.com/2013/06/digital-clock/', title: 'How to Make a
                Digital Clock with jQuery and CSS3',image:
                'http://cdn.tutorialzine.com/wp-
content/uploads/2013/06/digital_clock-100x100.jpg'
            },
            {
                url: 'http://tutorialzine.com/2013/05/diagonal-fade-gallery/',title: 'Smooth
                Diagonal Fade Gallery with CSS3 Transitions', image:
                'http://cdn.tutorialzine.com/wp-
content/uploads/2013/05/featured-100x100.jpg'
            },
            {
                url: 'http://tutorialzine.com/2013/05/mini-ajax-file-upload-form/',title: 'Mini AJAX
                File Upload Form',
                image: 'http://cdn.tutorialzine.com/wp-
content/uploads/2013/05/ajax-file-upload-form-100x100.jpg'
            },
            {
                url:  'http://tutorialzine.com/2013/04/services-chooser-backbone-
js/',
                title: 'Your First Backbone.js App – Service Chooser',image:
                'http://cdn.tutorialzine.com/wp-
```

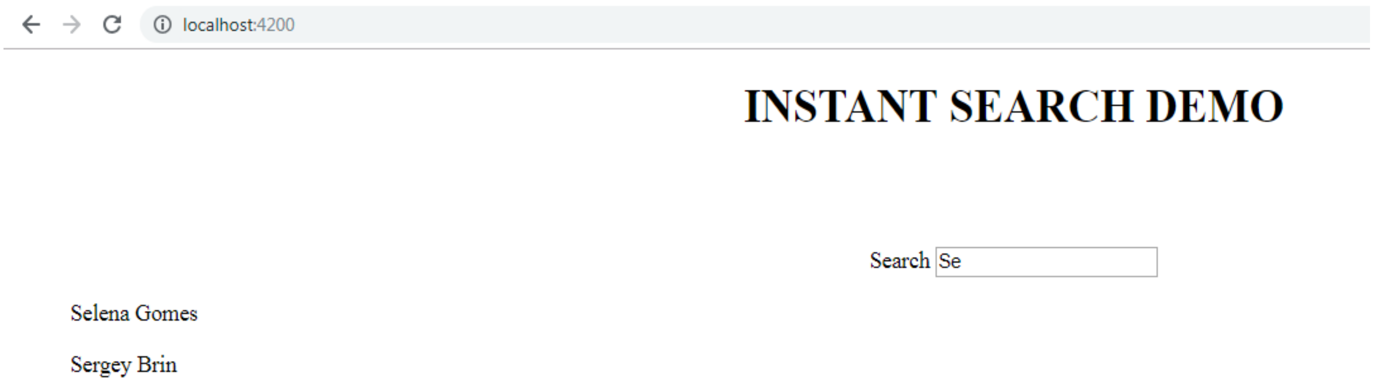
```
content/uploads/2013/04/service_chooser_form-100x100.jpg'
                }
            ];

    }
```

style.css

```
    /*
        Simple reset
                                        */

    *{
        margin:0;
        padding:0;
    }

    /*
        General Styles
                                        */

    body{
        font:15px/1.3 'Open Sans', sans-serif;color:
        #5e5b64;
        text-align:center;
    }

    a, a:visited {
        outline:none;
        color:#389dc1;
    }

    a:hover{
        text-decoration:none;
    }

    section, footer, header, aside, nav{display:
        block;
    }

    /*
        The search input
                                        */
```

```
            margin: 80px auto
            20px;
            position:relative;
    w }
    i
    d  .bar input{
    t       background:#fff no-repeat 13px 13px;
    h
    :
            background- image:url(data:image/png);
    4       borde
    0       r:
    0       none;
    p       width
    x       :
    ;       100%
    p       ;
    a       line-height: 19px;
    d padding: 11px 0;
    d  }
    i  ul{
    n
    g
    :
    1
    4
    border-bottom: 1px solid #ddd; padding:
    p10px;
    x
    overflow: hidden;
    ;



 ul li img{
    width:60px;
    height:60px;
    float:left;
    border:none;
 }

 ul li p{
    margin-left:      75px;
    font-weight:       bold;
    padding-top:      12px;
    color:#6e7a7f;
 }
```

OUTPUT:


localhost:4200

INSTANT SEARCH DEMO

Search Se

Selena Gomes

Sergey Brin

RESULT: Hence designed a web page with instant search using Angular JS.

SWITCHABLE GRID

AIM:

Program to design a web page with Switchable grid using Angular JS.

PROCEDURE:

1. Write a service that communicates with Instagram's API and returns an array with themost popular photos at the moment.

2. Include one additional Angular.js file in the page:

```
<script   src="http://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular-resource.min.js">
</script>
```

3. This includes the ngResource module for easily working with AJAX APIs (the module is exposed as the $resource variable in the code).

4. This file is automatically included in the editor.

PROGRAM:

```
index.html
<!DOCTYPE html>
<html>

        <head>
                <meta charset="utf-8"/>
                <title>Learn AngularJS - Switchable Grid</title>

                <link href="http://fonts.googleapis.com/css?family=Cookie|Open+Sans:400,700"
            rel="stylesheet" />

                <!-- The main CSS file -->
                <link href="style.css" rel="stylesheet" />

                <!--[if lt IE 9]>
            <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js">
            </script>
                <![endif]-->
        </head>

        <body ng-app="switchableGrid"  ng-controller="SwitchableGridController">

                <div class="bar">

                        <!-- These two buttons switch the layout varaible,which causes
                                  the correct UL to be shown. -->

                        <a href="#" class="list-icon" ng-class =
                        "{active: layout == 'list'}" ng-click="layout = 'list'"></a>
                        <a href="#" class="grid-icon" ng-class =
                        "{active: layout == 'grid'}" ng-click="layout = 'grid'"></a>
                </div>

                <!-- We have two layouts. We choose which one to show depending onthe
            "layout" binding -->

                <ul ng-show="layout == 'grid'" class="grid">
                        <!-- A view with big photos and no text -->
                        <li ng-repeat="p in pics">
                                <a href="{{p.link}}" target="_blank">
                                   <img  ng-src="{{p.images.low_resolution.url}}"  /></a>
                        </li>
                </ul>

                <ul ng-show="layout == 'list'" class="list">
                        <!-- A compact view smaller photos and titles -->
                        <li ng-repeat="p in pics">
                                <a href="{{p.link}}" target="_blank">
                                   <img ng-src="{{p.images.thumbnail.url}}" /></a>
                                <p>{{p.caption.text}}</p>
                        </li>
                </ul>
                <!-- Include AngularJS from Google's CDN and the resource module -->
     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js"></script>
    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular-
    resource.min.js"></script>
                <script src="script.js"></script>
        </body>
</html>
```

script.js

```
    // Define a new module. This time we declare a dependency on
    // the ngResource module, so we can work with the Instagram APIvar app =

    angular.module("switchableGrid", ['ngResource']);

    // Create and register the new "instagram" service
    app.factory('instagram', function($resource){

            return {
                    fetchPopular: function(callback){

    // The ngResource module gives us the $resource service. It makes working with
    // AJAX easy. Here I am using a client_id of a test app. Replace it with yours.

    var api = $resource('https://api.instagram.com/v1/media/popular?client_id=:clie
    nt_id&callback=JSON_CALLBACK',{
                client_id: '642176ece1e7445e99244cec26f4de1f'
    },{
    // This creates an action which we've chosen to name "fetch". It issues
    // an JSONP request to the URL of the resource. JSONP requires that the
    // callback=JSON_CALLBACK part is added to the URL.
            fetch:{method:'JSONP'}
            });

            api.fetch(function(response){
});// Call the supplied callback function callback(response.data);
    });
    }// The controller. Notice that I've included our instagram service which we
        // defined below. It will be available inside the function automatically.function

        SwitchableGridController($scope, instagram){

            // Default layout of the app. Clicking the buttons in the toolbar
            // changes this value.

            $scope.layout = 'grid';

            $scope.pics = [];

            // Use the instagram service and fetch a list of the popular pics
            instagram.fetchPopular(function(data){
        }
```

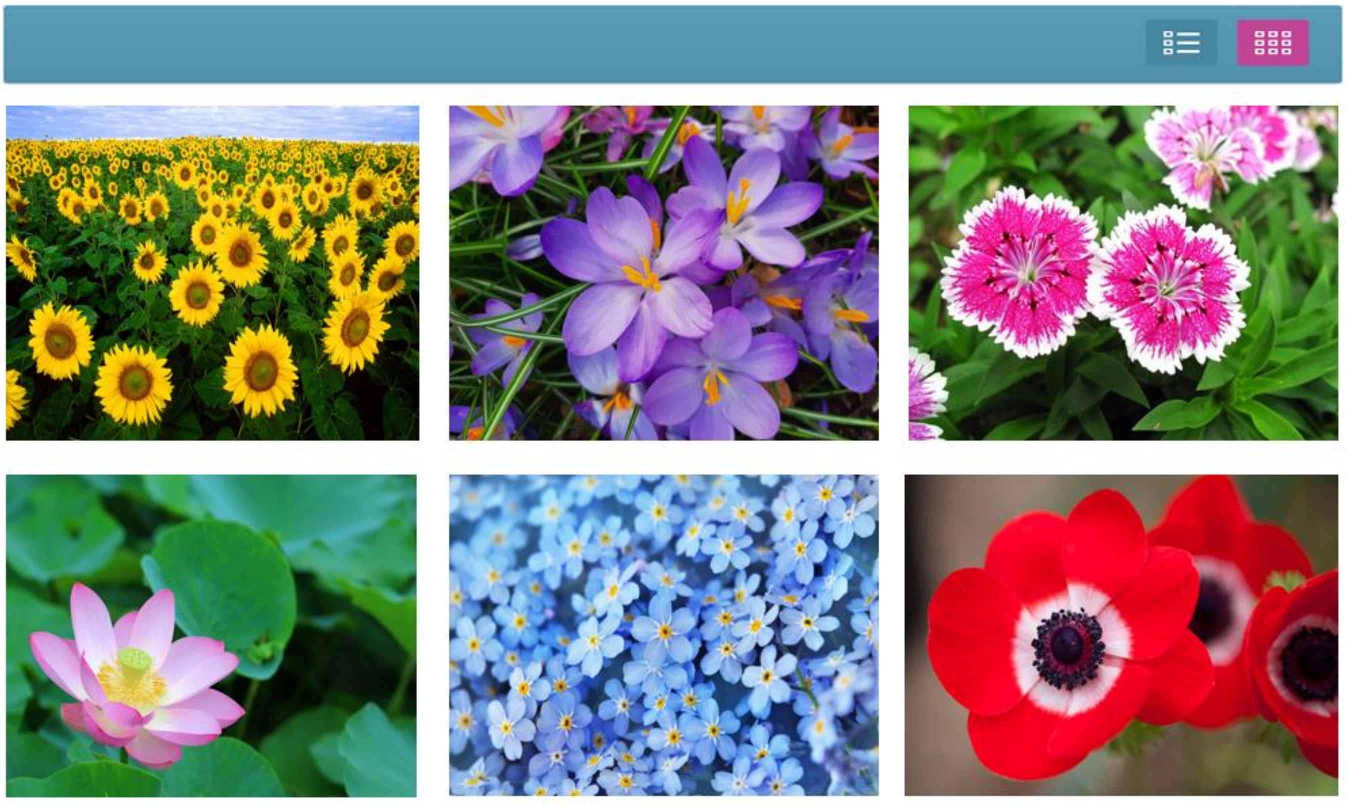
style.css

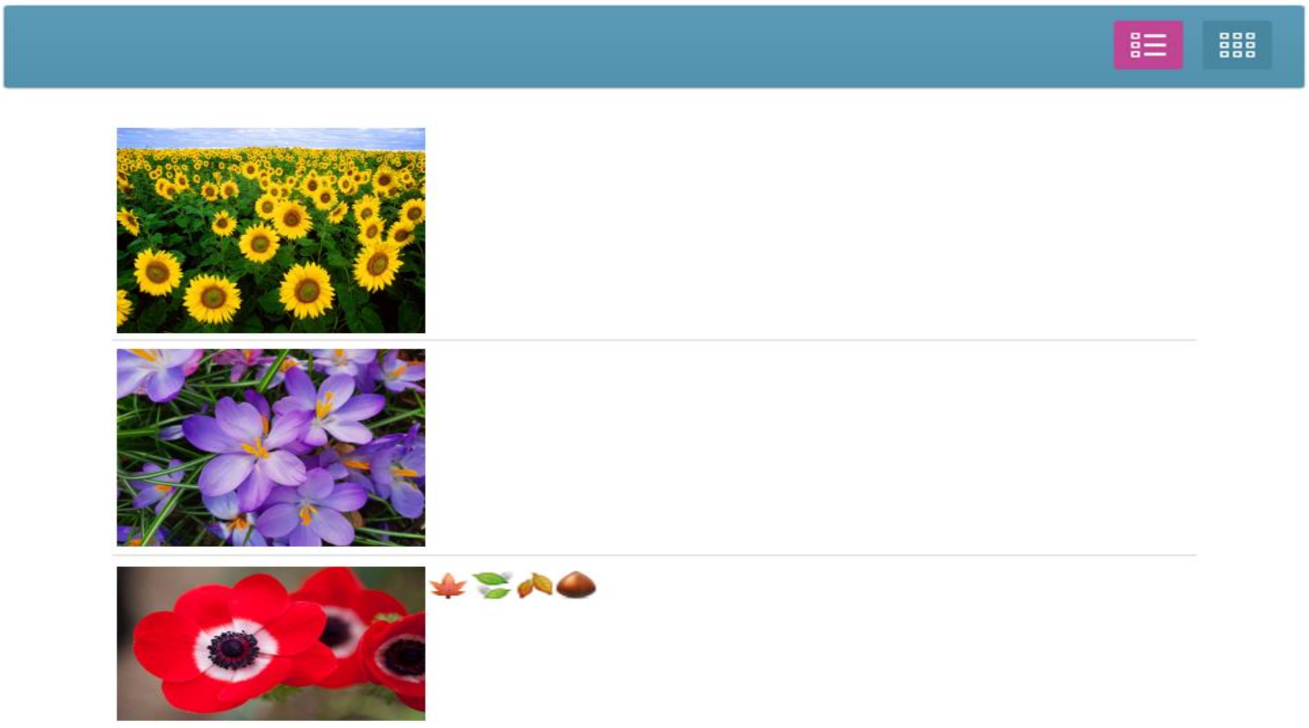
```
        /*                                  Simple reset
                                           */

        *{
            margin:0;
            padding:0
            ;
        }/*    General Styles  */

        body{
            font:15px/1.3 'Open Sans', sans-serif;color:
            #5e5b64;
            text-align:center;
        }

        a, a:visited {
            outline:none;
            color:#389dc1;
        }

        a:hover{
            text-decoration:none;
        }

        section, footer, header, aside, nav{display:
```

```
        block;
    }
  ---/*---------------------------------------        The search input
--------------------------------------------      */

                                                  }
    bar a{
        background:#4987a1 center center no-repeat;
        width:32px;
        height:32px;
        display:inline-block;
        text-decoration:none !important;
        margin-right:5px;
        border-radius:2px;
    }
    .bar a.active{
        background-color:#c14694;
    }

    .bar a.list-icon{
        background-
    image:url(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf 8/
    );
    bar input{
        background:#fff no-repeat 13px 13px;

        border: none;
        width: 100%;
        line-height: 19px;
        padding: 11px 0;

        border-radius: 2px;
        box-shadow: 0 2px 8px #c4c4c4 inset;
        text-align: left;
        font-size: 14px;
        font-family: inherit;
        color: #738289;
        font-weight: bold;
        outline: none; text-
        indent: 40px;
    }

  ---/*---------------------------------------        List layout
--------------------------------------------      */

    ul.list{
        list-style: none;
        width: 500px;
        margin: 0 auto;
        text-align: left;
    }
```

```
ul.list li{
        border-bottom: 1px solid #ddd;
        padding: 10px;
        overflow: hidden;
}

ul.list li img{
        width:120px;
        height:120px
        ;float:left;
        border:none;
}

ul.list li p{
        margin-left: 135px;
        font-weight: bold;
        color:#6e7a7f;
}

/*                                                      Grid layout
                                                     */

ul.grid{
        list-style: none;
        width: 570px;
        margin: 0 auto;
        text-align: left;
}

ul.grid li{
        padding: 2px;
        float:left;
}

ul.grid li img{
        width:280px;
        height:280px;
        display:block;
        border:none;
}
```

OUTPUT:





RESULT: Hence designed a web page with Switchable grid using Angular JS.

Ex.No. : 8 Date : Reg. No. :211701049

SINGLE PAGE APPLICATION

AIM:

Program to develop an attractive web pages using Bootstrap.

PROCEDURE:

1. Define a simple controller:
2. After created module and controller, use them in our HTML.
3. Include angular script and app.js that we built.
4. Specify module in ng-app attribute and controller in ng-controller attribute.
5. Start working on adding single page application support.
6. Make a single page application and don't want any page refreshes, use Angular's routing capabilities.
7. Include angular-route script after the main angular script.
8. Specify that the module depends on ngRoute module to be able to use it.
9. The next thing is to distinguish common HTML for every page. This HTML will be layout of the website.
10. Then specify the place where HTML of each page will be placed in our layout. There is a ng-view directive for that.
11. ng-view is an Angular directive that will include the template of the current route (for example, /blog or /about) in the main layout file.
12. Configure the routes. Use $routeProvider service from the ngRoute module.
13. For each route, specify templateUrl and controller.
14. If user will try to go to the route that does not exist, handle this by using otherwise function. In our case, we will redirect user to the "/" route:
15. Build controllers for every route (already specified their names in routeProvider).

PROGRAM:

index.html

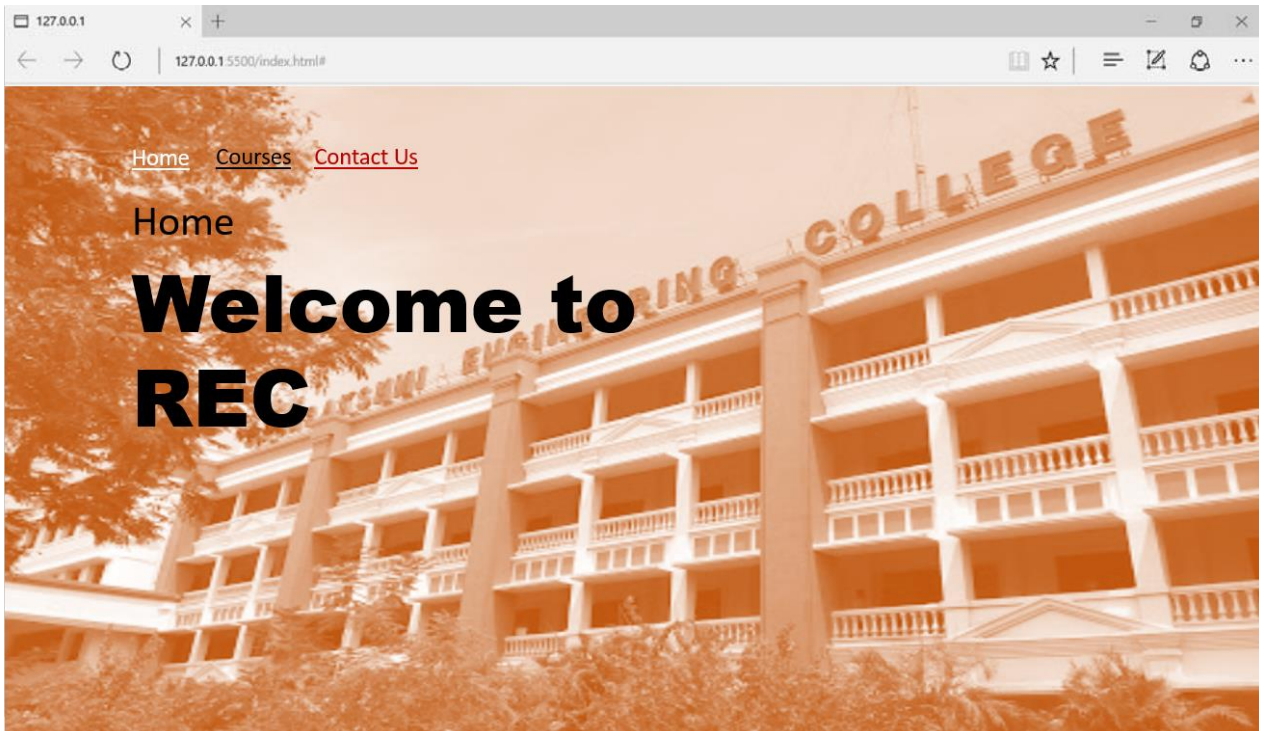
```
<!doctype html>
<html ng-app="myApp">
   <head>
      <script
         src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.
         7/ angular.min.js"></script>
      <script
         src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular
         - route.min.js"></script>
   </head>
   <body>
      <script type="text/ng-template" id="pages/home.html">
         <h1>Home</h1>
         <h3>{{message}}</h3>
      </script>
      <script type="text/ng-template" id="pages/courses.html">
         <h1>Courses</h1>
         <h3>{{message}}</h3>
      </script>
      <script type="text/ng-template" id="pages/contactus.html">
         <h1>Contact Us</h1>
         <h3>{{message}}</h3>
      </script>

      <a href="#/">Home</a>
      <a href="#/courses">Courses</a>
      <a href="#/contactus">Contact Us</a>

      <div ng-view></div>

      <script src="app.js"></script>
   </body>
</html>
```

app.js

```
var app = angular.module('myApp', ['ngRoute']);

app.config(function($routeProvider) {
   $routeProvider

   .when('/', {
      templateUrl : 'pages/home.html',
      controller   : 'HomeController'
```

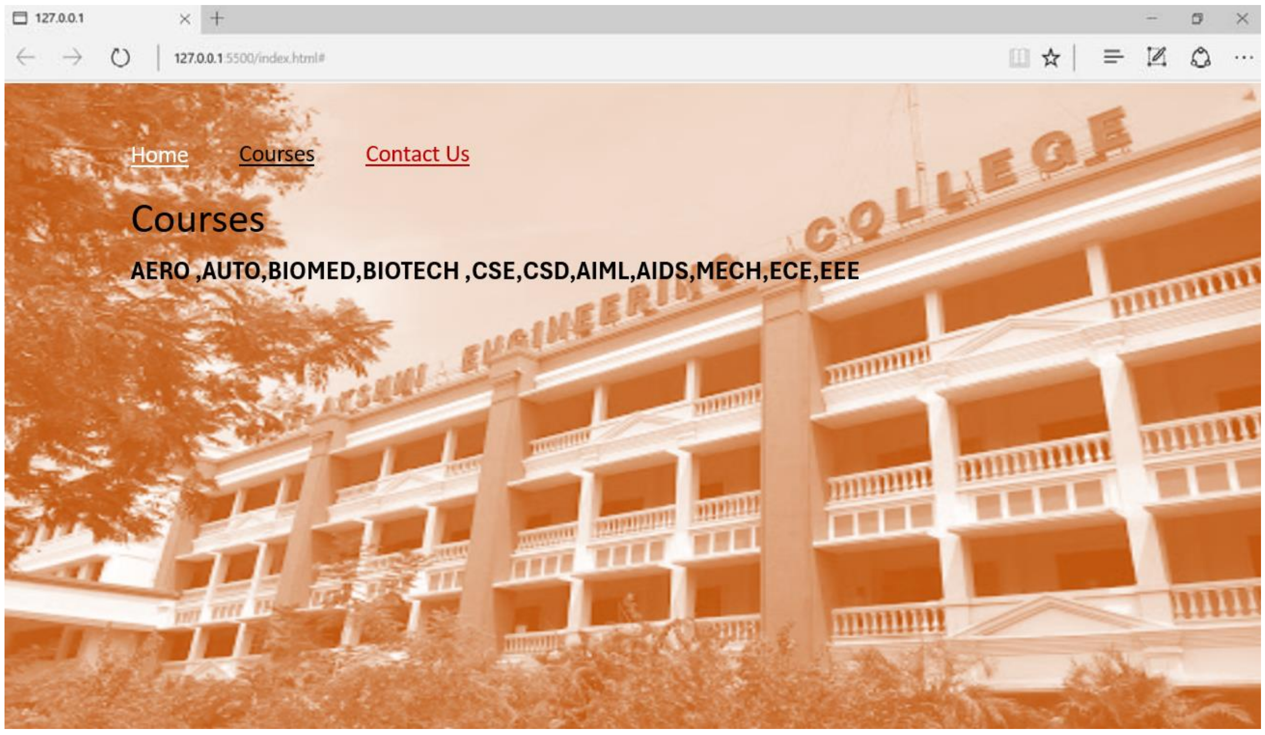
```
        })

        .when('/courses', {
            templateUrl : 'pages/courses.html',
            controller   : 'CoursesController'
        })

        .when('/contactus', {
            templateUrl :
            'pages/contactus.html',

            controller   : 'ContactUsController'

        })
        .otherwise({redirectTo: '/'});
    });

    app.controller('HomeController',  function($scope) {
        $scope.message = 'Welcome to REC';
    });

    app.controller('CoursesController',  function($scope)  {
        $scope.message = 'AERO, AUTO, BIOMED, BIOTECH, CHEMICAL, CIVIL,
                CSE, CSBC,ECE, EEE, FT, IT, MCT, MECH';
    });

    app.controller('ContactUsController',  function($scope)  {
        $scope.message = 'Rajalakshmi Nagar, Thandalam, Chennai - 602 105';
    });
```

OUTPUT:



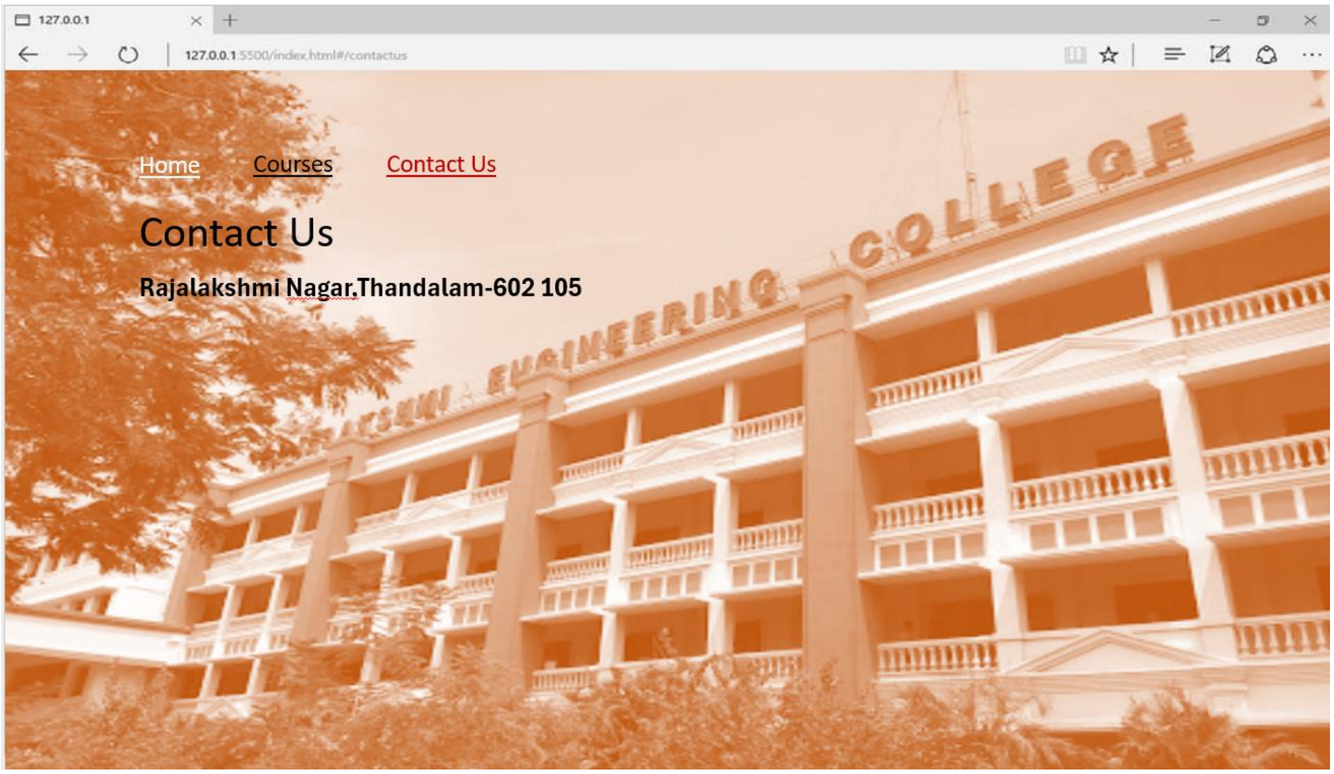
Home

Welcome to REC



Courses

AERO ,AUTO,BIOMED,BIOTECH ,CSE,CSD,AIML,AIDS,MECH,ECE,EEE



Contact Us

Rajalakshmi Nagar,Thandalam-602 105

RESULT: Thus an attractive web pages using Bootstrap has been successfully created.