RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

RAJALAKSHMI
ENGINEERING COLLEGE

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 : _____

S. No.	Date	Title	Page No.	Teacher's Signature / Remarks
1		Create a web page to embed a map along with hot spot AND links		
2		Create a web page using an embedded, external, and inline CSS file		
3		Create a registration page along with validations		
4a		JSP - Library Management System		
4b		Servlet - Bank Application		
5		PHP – Employee Details to connect database and execute queries to retrieve and update data. Prepare report for single and group of employees based on the end user needs		
6		Bootstrap – Web Page		
7		Design a Web page with Navigation menu, Inline editor, Order form, Instant Search & Switchable Grid.		
8		Angular JS – Single Page Application		

CD19643 – WEB ESSENTIALS

List of Experiments:

1. Create a web page to embed a map along with hot spot, frames & links.
2. Create a web page using an embedded, external and inline CSS file.
3. Create a registration page along with validations.
4. Consider a Library Management System. Develop a JavaScript program that will validate the controls in the forms you have created for the application. State the assumptions you make (business logic you are taking into consideration). Note: Your application must access a database using Servlet / JSP.
5. Write a 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. Also, prepare the report for single and group of employees based on the end user needs.
6. Develop an Attractive web pages using Bootstrap.
7. Design a Web page with Navigation menu, Inline editor, Order form, Instant Search & Switchable Grid.
8. Design a single page application using Angular 14.

Ex. No 1 Creating a web page to embed a map along with hot spot AND links

AIM:

To create a web page to embed a map along with hot spot and links.

ALGORITHM:

1. To create a web page, use html tags and give the suitable title.
2. <map> tag is used to define an image map which is an image with clickable areas.
3. <area> tag is used to specify a area inside the image.
4. Inside the area tag shape, coordinate, href and alt values are given.
5. Image source is taken and given in <img> tag and use map attribute specifies an image asa user side image map.

PROGRAM:

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Image Map</title>
  </head>
  <body>
    <map name="mmap">
      <area shape="circle" coords="70,280,30" alt="Tamilnadu" href="Tamilnadu.html">
      <area shape="rect" coords="40,160,200,250" alt="Karnataka"
href="Karnataka.html">
    </map>
    <img src="C:\Users\STUDENT\Desktop\india-map-2019.jpg" alt="India"
usemap="#mmap" width="200" height="300">
  </body>
</html>
```

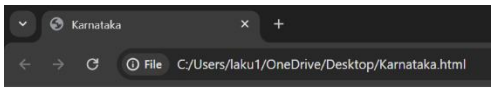
Tamilnadu.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tamilnadu</title>
  </head>
  <body>
    <h1>Tamilnadu</h1>
    <img src="C:\Users\STUDENT\Desktop\Tamilnadu.png" width="300" height="400">
  </body>
</html>
```

Karnataka.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Karnataka</title>
</head>
<body>
<h1>Karnataka</h1>
<img src="C:\Users\STUDENT\Desktop\karnataka-map.jpg" width="300"
height="400">
</body>
</html>
```

OUTPUT:



Karnataka

RESULT :

Thus the webpage has been successfully created.

Ex. No 2 Web page creation using an embedded, external, and inline CSS file.

AIM:

To create a web page using an embedded, external, and inline CSS file.

ALGORITHM:

1. Web page is created using html tags.
2. To style the sheet give inline CSS if it is needed.
3. Else create a separate CSS file which is external and link with the html file.
4. Another way of applying CSS is by using the <style> tag within html file.
5. CSS given for P tag, classes and headings.

PROGRAM:

Index.html

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="styel.css">
    <title>Rajalakshmi Engineering College</title>
  </head>
  <body>
    <div class="header">
      <h1 style="padding-top:
2%;"><U>RAJALAKSHMI</U><br><small>ENGINEERING COLLEGE</small></h1>
    </div>
    <div class = "address">
      <p>Rajalakshmi Nagar, Thandalam, Chennai - 602 105</p>
    </div>
    <div class="cont">
      <div class="content">
      <p>Welcome to the Department of CSE home page</p>
      </div>
```

```
        <div class="container">
        <ul style="list-style: none;">
            <br><br>
            <li style="color: black;"><a href="website.html"><b>CSE</b></a></li><br>
            <li><a href="intro.html">Introduction</a></li><br>
            <li><a href="opp.html">Opportunities</a></li><br>
            <li><a href="lab.html">Lab facility</a></li><br>
            <li><a href="news.html">News</a></li></ul>
        </div></div>
    </body></html>
```

Intro.html

```
<!DOCTYPE html>
<html>
    <body>
        <div class="cont">
            <div class="content">
                <h4 style="color: blueviolet;">INTRODUCTION</h4>
                <p><b>Vision</b></p>
                <p>To promote Ethical and Innovative Computer Professionals through excellence in teaching, training and research.</p>
                <p><b>Mission</b></p>
                <p>To produce globally competent professionals<br>To promote research activities<br>To impart mortal and ethical values</p>
            </div>
            <div class="container">
            <ul style="list-style: none;">
                <br><br>
                <li style="color: black;"><a href="website.html"><b>CSE</b></a></li><br>
                <li><a href="intro.html">Introduction</a></li><br>
                <li><a href="opp.html">Opportunities</a></li><br>
                <li><a href="lab.html">Lab facility</a></li><br>
                <li><a href="news.html">News</a></li><ul></div></div>
    </body></html>
```

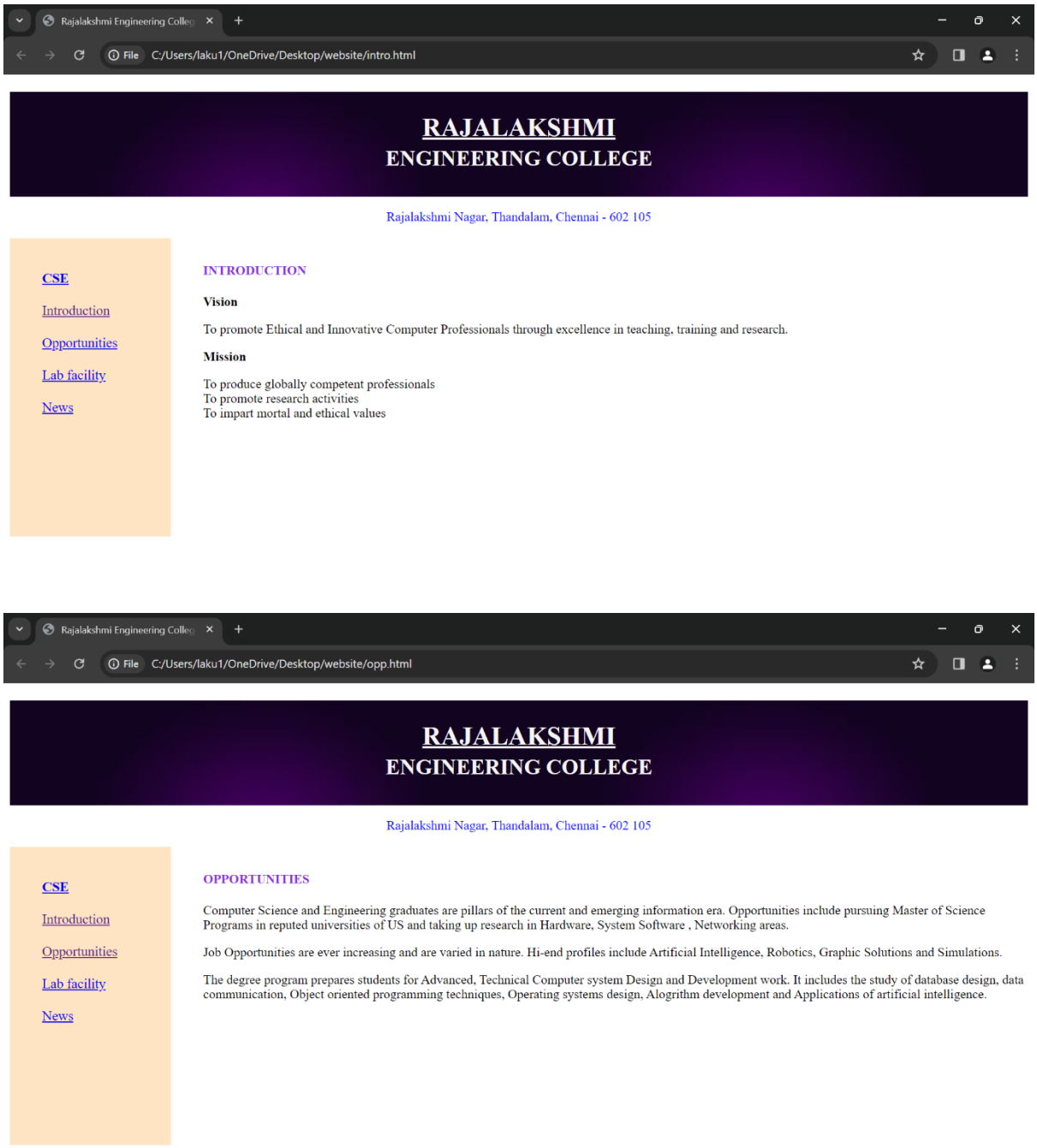
opp.html

```
<!DOCTYPE html>
<html>
  <body>
    <div class="cont">
      <div class="content">
        <h4 style="color: blueviolet;">OPPORTUNITIES</h4>
        <p>Computer Science and Engineering graduates are pillars of the current and emerging information era. Opportunities include pursuing Master of Science Programs in
          reputed universities of US and taking up research in Hardware, System Software , Networking areas.
        </p>
        <p>Job Opportunities are ever increasing and are varied in nature. Hi-end profiles include Artificial Intelligence, Robotics, Graphic Solutions and Simulations.</p>
        <p>The degree program prepares students for Advanced, Technical Computer system Design and Development work. It includes the study of database design, data communication, Object oriented programming techniques,
          Operating systems design, Alogrithm development and Applications of artificial intelligence.
        </p>
      </div>
      <div class="container">
      <ul style="list-style: none;">
        <br><br>
        <li style="color: black;"><a href="website.html"><b>CSE</b></a></li><br>
        <li><a href="intro.html">Introduction</a></li><br>
        <li><a href="opp.html">Opportunities</a></li><br>
        <li><a href="lab.html">Lab facility</a></li><br>
        <li><a href="news.html">News</a></li>
       </ul>
      </div></div>
  </body>
</html>
```

Style.css

```
.header{
    text-align: center;
    background-image: url(https://img.freepik.com/free-photo/studio-background-concept-dark-gradient-purple-studio-room-background-product_1258-53875.jpg);
    color: whitesmoke;
    height: 130px;
    font-family: "Times New Roman", Times, serif;
}
.address{
    color: blue;
    text-align: center;
}
.container{
    color: purple;
    background-color: bisque;
    width: 200px;
    height: 370px;
    text-align: left;
    font-size: 17px;
}
.content{
    position: absolute;
    padding-left: 240px;
    padding-top: 10px;
}
```

OUTPUT:



RESULT:

Thus a webpage using external, inline and embedded CSS has been created successfully.

Ex. No 3 Creation of registration page along with validations

AIM:

To create a registration form with validations

ALGORITHM:

1. Create a registration form interface with HTML and CSS.
2. Get inputs from the user using the input tag.
3. Set appropriate tables above the input boxes.
4. Finally call the java script file using @media to access the file.
5. Perform validation and display alert message validation when submitted.

PROGRAM:

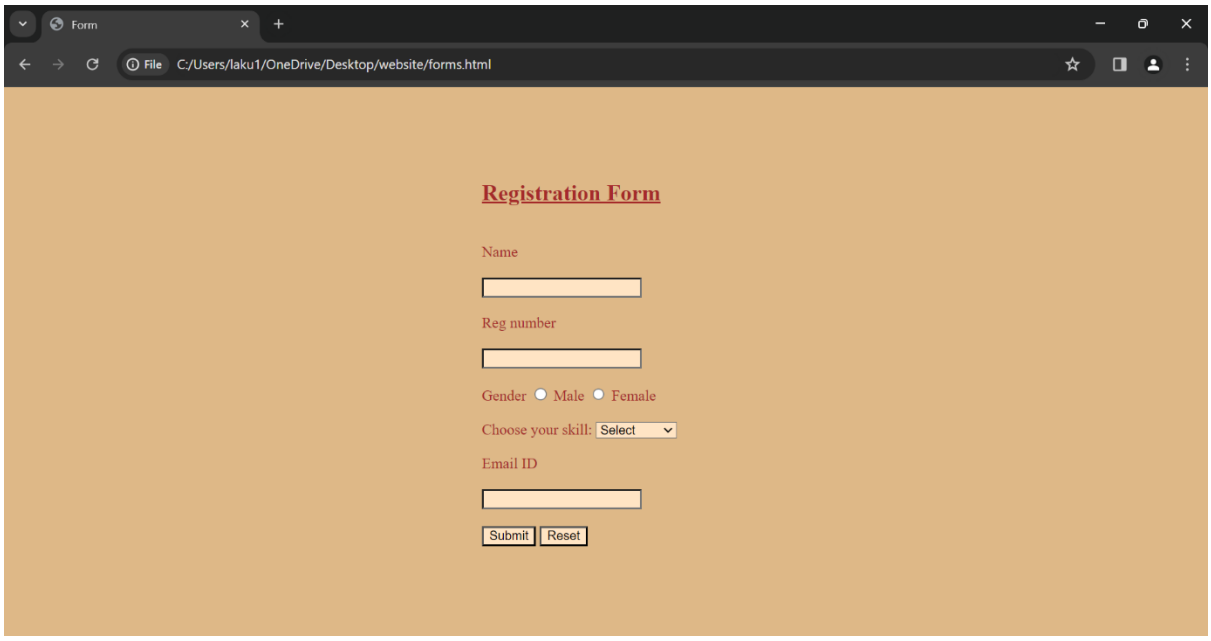
Forms.html

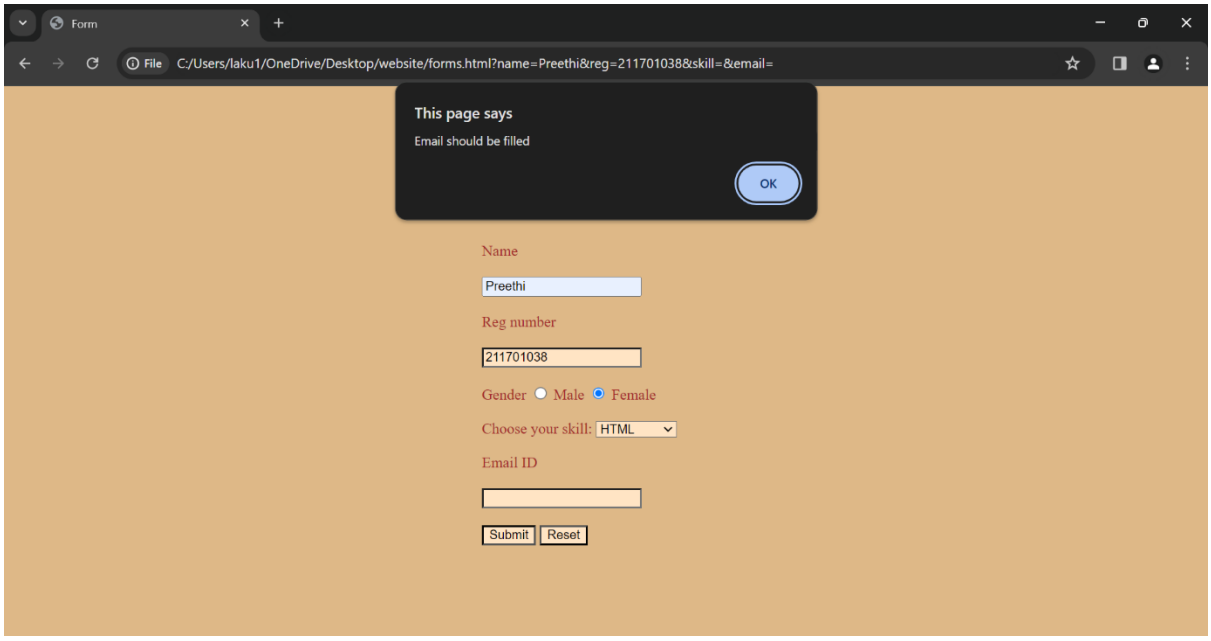
```
<!DOCTYPE html>
<html>
  <head>
    <title>Form</title>
    <script>
      function validate(){
      var name = document.getElementById('name').value;
      var reg = document.getElementById('reg').value;
      var email = document.getElementById('email').value;
      if(name==null || name==""){
        alert("Name must be filled");
      }else if(email==null || email==""){
        alert("Email should be filled");
      }
      else{
      alert("Thanks for submitting!\nName:"+name+"\nReg No:"+reg+"\nEmail:"+email);
    }  }
```

```
    </script>
    <style>
        body{
            background-color: burlywood;
            color: brown;
            padding-top: 70px;
            padding-left: 500px;
        }
        label{
            text-align: center;
            padding-top: 10px;
        }
        input{
            position: relative;
            background-color: bisque;
        }
        option ,select{
            background-color: bisque;
        }
        form{
            border: 20px;
            padding-top: 20px;
        }
    </style>
</head>
<body>
    <h2><u>Registration Form</u></h2>
    <div class="formm">
        <form>
```

```
            <label for="name">Name</label><br><br>
            <input type="text" id="name" name="name"><br><br>
            <label for="reg">Reg number</label><br><br>
            <input type="number" id="reg" name="reg"><br><br>
            <label for="gen">Gender</label>
            <input type="radio" id="m" name="m" >
            <label for="m">Male</label>
            <input type="radio" id="f" name="f">
            <label for="f">Female</label><br><br>
            <label for="skill">Choose your skill:</label>
            <select id="skill" name="skill" aria-placeholder="select">
            <option value="">Select</option>
            <option value="HTML">HTML</option>
            <option value="CSS">CSS</option>
            <option value="JavaScript">JavaScript</option>
            <option value="Angular">Angular</option>
            </select><br><br>
            <label for="email">Email ID</label><br><br>
            <input type="email" id="email" name="email"><br><br>
            <input type="submit" value="Submit" onclick="validate()">
            <input type="reset" value="Reset" style="text-align: center;">
        </form>
    </div>
  </body>
</html>
```

OUTPUT :





RESULT:

Thus a registration page with validations has been created successfully.

Ex. No 4a) **JSP –Library Management System**

AIM:

To write a program to develop a library Management System application accessing a database using JSP.

ALGORITHM:

Relations using MYSQL given below enforcing primary key and foreign key constraints:
BOOK (ACCNO, TITLE, AUTHOR, PUBLISHER, EDITION, PRICE)

MEMBER (MID, MNAME, BRANCH)FINE (MID, FINE_DATE, AMOUNT)

1. Open MySQL.
2. Create a database.

mysql> create database library; Query OK, 1 row affected (0.02 sec)

3. Connect to the database.

mysql> use library; Database changed

4. Create the following tables:

mysql> create table book (accno integer, title varchar(20),

-> author varchar(20), publisher varchar(20),

-> edition integer, price integer, primary key (accno)); Query OK, 0 rows affected (0.09 sec)

mysql> create table member (mid integer, mname varchar(20),

-> branch varchar(20), primary key (mid)); Query OK, 0 rows affected (0.06 sec)

mysql> create table fine (mid integer references member(mid),

-> fine_date date, amount integer); Query OK, 0 rows affected (0.06 sec)

PROGRAM:

Index.html

```
<!DOCTYPE html>
<html>
  <head><title>Library Management System</title>
    <meta charset="UTF-8">
    <style>
      table,th,td,tr{
        border: 1px solid black;
        border-collapse: collapse ;
        color:#B03A2E;
        background-color: #FADBD8;
        text-align: center;
      }
      th{
        width: 30%;
      }
      td{
        width: 40%;
      }
      .tables{
        padding-left: 30%;
        height: 60%;
        width: 60%;
      }
    </style></head><body>
    <p style="text-align: center;">Books Data</p><br>
    <div class="tables">
    <form action="action.php" method="post">
    <table style="width: 500px; height: 200px;">
      <tr>
        <td><label for="AccNo">Account Number</label></td>
```

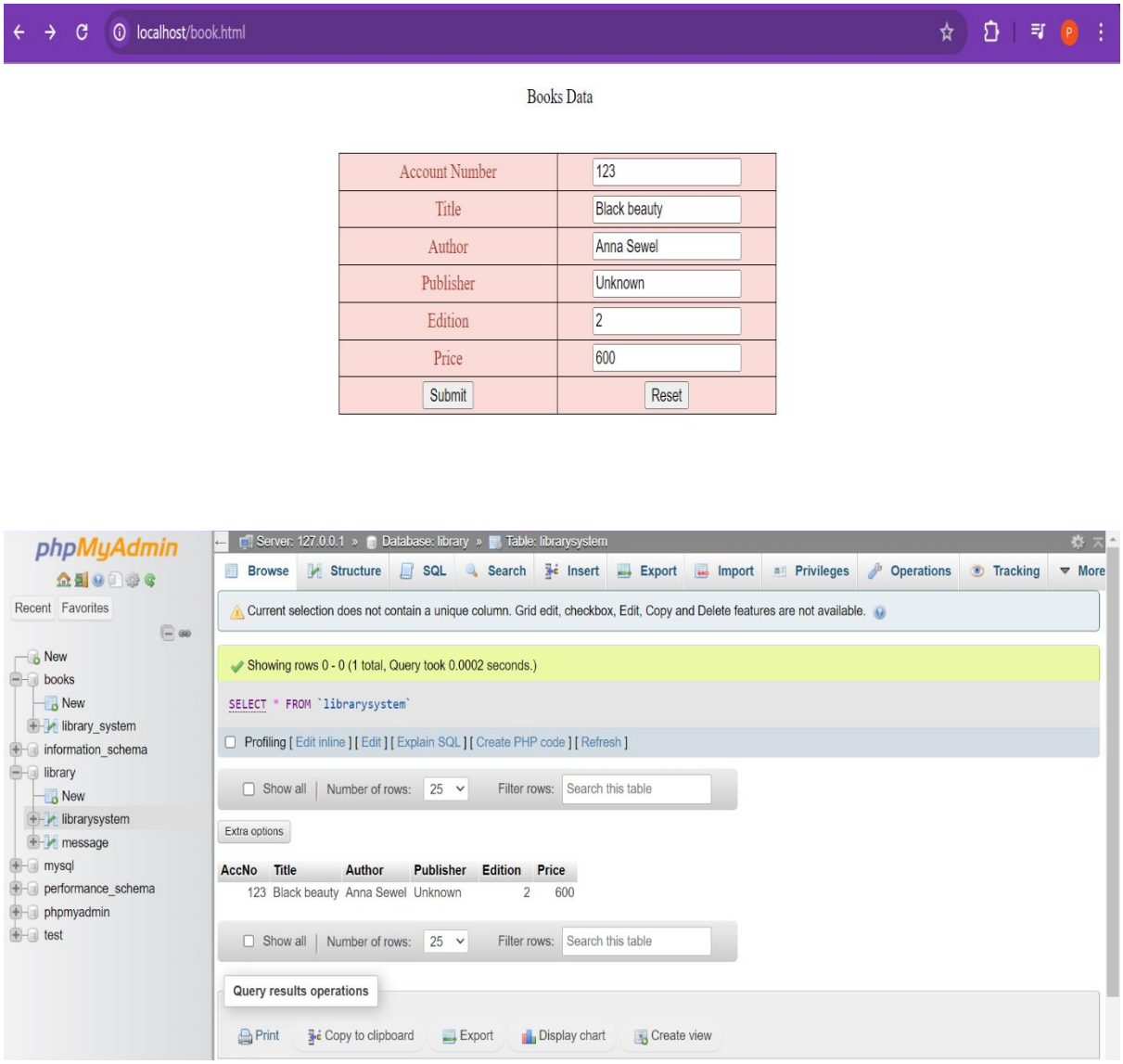
```
            <td><input type="number" name="AccNo" id="AccNo"></td>
        </tr><tr>
            <td><label for="Title">Title</label></td>
            <td><input type="text" name="Title" id="Title"></td>
        </tr><tr>
            <td><label for="Authot">Author</label></td>
            <td><input type="text" name="Author" id="Author"></td>
        </tr><tr>
            <td><label for="Publisher">Publisher</label></td>
            <td><input type="text" name="Publisher" id="Publisher"></td>
        </tr><tr>
            <td><label for="Edition">Edition</label></td>
            <td><input type="text" name="Edition" id="Edition"></td>
        </tr><tr>
            <td><label for="Price">Price</label></td>
            <td><input type="number" name="Price" id="Price"></td>
        </tr><tr>
        <td><input type="submit" value="Submit"></td>
        <td><input type="reset" value="Reset" style="text-align: center;"></td></tr>
    </table>
</form></div>
    </body></html>
```

Action File

```
<?php
$AccNo = $_POST['AccNo'];
$Title = $_POST['Title'];
$Author = $_POST['Author'];
$Publisher = $_POST['Publisher'];
$Edition = $_POST['Edition'];
$Price = $_POST['Price'];

$host = "localhost";
```

```
$dbname = "library";
$username = "root";
$password = "";

$conn = mysqli_connect(hostname: $host, username: $username, password: $password,
database: $dbname);

if(mysqli_connect_errno()){
   die("Connection error: ".mysqli_connect_error());
}

$sql = "INSERT INTO
librarysystem(AccNo,Title,Author,Publisher,Edition,Price)VALUES(?,?,?,?,?,?)";
$stmt = mysqli_stmt_init($conn);

if ( ! mysqli_stmt_prepare($stmt, $sql)) {

   die(mysqli_error($conn));
}

mysqli_stmt_bind_param($stmt,"isssii",$AccNo,$Title,$Author,$Publisher,$Edition,$Price);

mysqli_stmt_execute($stmt);

echo "Record saved.";
```

OUTPUT:



RESULT:

Thus a library management system has be created successfully.

Ex. No 4b) **Servlet – Bank Application**

AIM:

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

ALGORITHM:

Relations using MYSQL for a banking application given below enforcing primary 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));

Query OK, 0 rows affected (0.08 sec)

mysql> create table account (ano integer, atype varchar(1),

-> balance integer, cid integer references customer (cid),

-> check (atype in ('S', 'C')), primary key (ano)); Query OK, 0 rows affected (0.06 sec)

mysql> create table transaction (tid integer,

-> ano integer references account(ano),

-> ttype varchar(1), adate date, tamount integer,

-> check (ttype in ('D', 'W')), primary key (tid)); Query OK, 0 rows affected (0.06 sec)

PROGRAM:

Index.html

```
<html>
  <head>
    <title>TODO supply a title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo"
crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"
integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86dIHNDz0W1"
crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"
integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM"
crossorigin="anonymous"></script>
  </head>
  <body>
    <div class="container">
      <div class="row">
        <div><form action="servlet1" method="POST">
            <div class="form-group">
              <h2>Welcome to Ebank</h2></div>
            <div class="form-group">
              <label>Enter the Account Number</label>
              <input type="text" name="accno" class="form-control"
placeholder="AccNo"></div>
            <div class="form-group">
              <label>Pin No</label>
```

```
                    <input type="password" name="pinno" class="form-control"
placeholder="PinNo"></div>
                  <div class="form-group">
                    <button type="submit" class="btn btn-success">Submit</button>
                </div></form></div>
        </div></div></body></html>
```

Servlet1.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 java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/servlet1")
public class servlet1 extends HttpServlet
{
   PrintWriter out = null;
   Connection con;
   PreparedStatement pst;
   ResultSet rs;
   @Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
```

```
        throws ServletException, IOException
    {
      try
      {
        String result;
        Class.forName("com.mysql.jdbc.Driver");
        con = DriverManager.getConnection("jdbc:mysql://localhost/bar", "root", "");
        ServletContext context = getServletContext();
        context.setAttribute("accno", "");
        String accno = request.getParameter("accno");
        String pinno = request.getParameter("pinno");
        pst = con.prepareStatement("select * from login where accno = ? and pinno = ?");
        pst.setString(1, accno);
        pst.setString(2, pinno);
        rs = pst.executeQuery();
        boolean row = false;
        row = rs.next();
if(row == true)
      {
            result = rs.getString(2);
            context.setAttribute("accno", result);
            RequestDispatcher dispatcher =
getServletContext().getRequestDispatcher("/secondservlet");
            if(dispatcher == null)
            {
            }
            dispatcher.forward(request, response);
            con.close();
    }else
      {
            out = response.getWriter();
            response.setContentType("text/html");
```

```
            out.println("<html>");
            out.println("<body bgcolor=pink>");
            out.println("Please check the Accno and Balance");
            out.println("</body");
            out.println("</html");
            out.close();
        }} catch (ClassNotFoundException ex) {
          ex.printStackTrace();
       } catch (SQLException ex) {
           ex.printStackTrace();
    }}}
```

Servlet2.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/secondservlet")
public class secondservlet extends HttpServlet {
@Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
          throws ServletException, IOException {
PrintWriter  out = response.getWriter();
     response.setContentType("text/html");
ServletContext context = getServletContext();
      Object obj = context.getAttribute("accno");
      String value = obj.toString();
out.println("<html>");
      out.println("<body bgcolor=pink>");
```

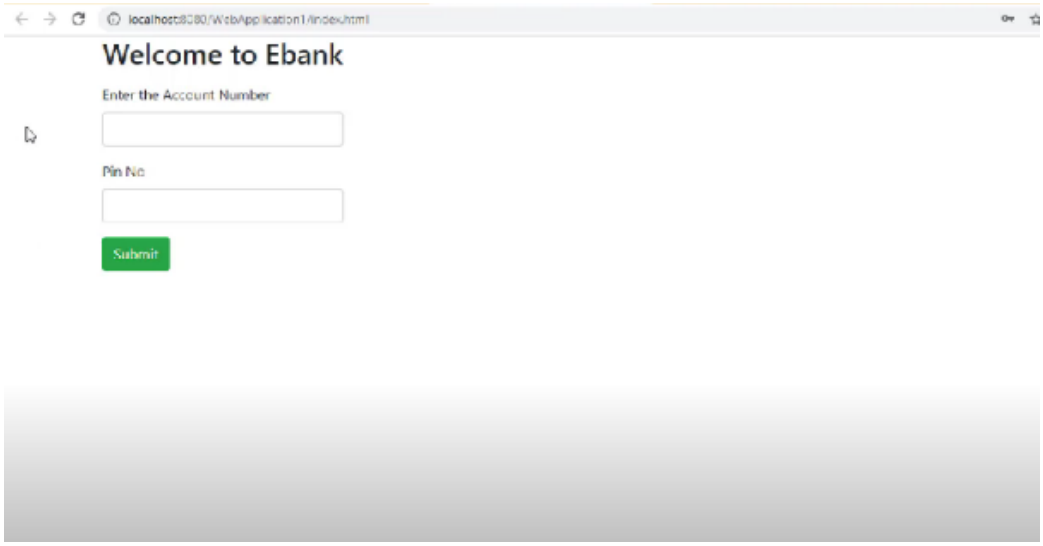
```
        out.println("<center>");
        out.println("<h2>Ebank</h2>");
        out.println("<center>");
        out.println("<Form method=post action=servlet3>");
         out.println("<b> Click the Deposit button</b>");
         out.println("<table>");
         out.println("<tr>");
         out.println("<td>");
         out.println("Account Number : " + value);
          out.println("</td>");
         out.println("</tr>");
          out.println("<tr>");
         out.println("<td>");
         out.println("Deposit Amount : </td>  <td> <input type=text name=amount value=0>");
          out.println("</td>");
         out.println("</tr>");
         out.println("</table>");
          out.println("<input type=submit value=deposit>");
          out.println("</br>");
          out.println("</Form>");
          out.println("</body>");
          out.println("</html>");
    }}
```

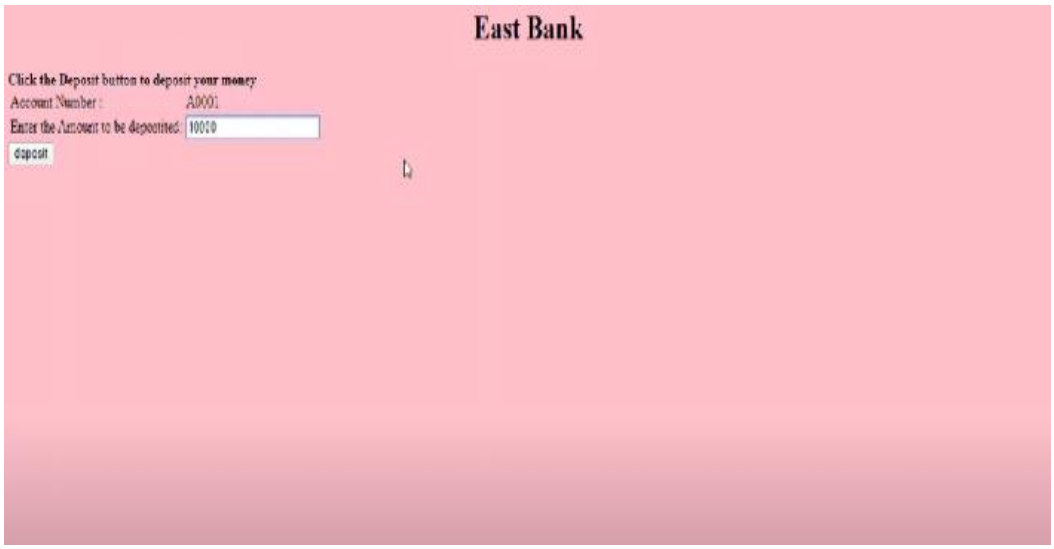
Servlet3.java

```
import java.io.IOException;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/servlet3")
public class servlet3 extends HttpServlet {@Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {try
      {Connection con;
         PreparedStatement pst;
         ResultSet rs;
         PrintWriter  out = response.getWriter();
         response.setContentType("text/html");
         Class.forName("com.mysql.jdbc.Driver");
         con = DriverManager.getConnection("jdbc:mysql://localhost/bar", "root", "");
         ServletContext context = getServletContext();
         Object obj = context.getAttribute("accno");
         String accno = obj.toString();
         DateTimeFormatter df = DateTimeFormatter.ofPattern("yyy/MM/dd");
         LocalDateTime now = LocalDateTime.now();
         String date = df.format(now);
         String amount = request.getParameter("amount");
         pst = con.prepareStatement("insert into
account_holder(accnum,date,mdeposit)values(?,?,?)");
         pst.setString(1, accno); pst.setString(2, date);pst.setString(3, amount);
         int rows = pst.executeUpdate();
if(rows==1) {out.println("Your Transaction have been done");
         }else
         {out.println("Your Transaction failed");
         }}
      catch (ClassNotFoundException ex)
      {ex.printStackTrace();}
      catch (SQLException ex) {
         ex.printStackTrace();}}}
```

OUTPUT:





RESULT:

Thus a bank application using Java servlet has been created successfully.

Exp No 5 **PHP – Employee Details**

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 and foreign key constraints:

EMPDETAILS (EMPID, ENAME, DESIG, DEPT, DOJ, SALARY)

1. Open MySQL.

2. Create a database.

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

3. Connect to the database.

```
mysql> use rec;
Database changed
```

4. Create the following tables:

```
mysql> create table empdetails(empid int primary key,
   -> ename varchar(20), desig varchar(20), dept varchar(20),
   -> doj date, salary int);
Query OK, 0 rows affected (0.08 sec)
```

PROGRAM:

config.php

```
<?php
        $databaseHost = 'localhost';
        $databaseName = 'rec';
        $databaseUsername = 'root';
        $databasePassword = 'admin';
        $mysqli = mysqli_connect($databaseHost, $databaseUsername,
           $databasePassword, $databaseName);
?>
```

index.php

```
<?php
//including the database connection file
include_once("config.php");
//fetching data in descending order (lastest entry first)
$result=mysqli_query($mysqli, "SELECT * FROM empdetails ORDER BY empid DESC");
?>
<html>
<head>
        <title>Homepage</title>
</head>
<body>
        <h1 align="center">Employee Details</h1>
        <hr />
        <a href="add.html">Add New Data</a><br/><br/>
        <table width='100%' border=0>
                <tr bgcolor='#CCCCCC'>
                <td>Employee Id.</td>
                <td>Name</td>
                <td>Designation</td>
                <td>Department</td>
                <td>DOJ</td>
                <td>Salary</td>
                <td>Edit / Delete</td>

        </tr>
        <?php
                while($res = mysqli_fetch_array($result)) {echo
                        "<tr>";
                        echo "<td>".$res['empid']."</td>";
                        echo "<td>".$res['ename']."</td>";
                        echo "<td>".$res['desig']."</td>";
                        echo "<td>".$res['dept']."</td>";
                        echo "<td>".$res['doj']."</td>";
                        echo "<td>".$res['salary']."</td>";
                        echo "<td><a href='edit.php?empid=$res[empid]'>Edit</a>";
                        echo " | <a href='delete.php?empid=$res[empid]'>Delete</a></td>";echo "</tr>";
                }
        ?>
        </table>
</body>
</html>
```

add.html

```
<html>

<head>
	<title>Add Employee Details</title>
</head>

<body>
	<h1 align="center">Add Employee Details</h1>
	<hr />
	<a href="index.php">Home</a>
	<br /><br />

	<form action="add.php" method="post" name="form1">
		<table width="25%" border="0">
			<tr>
				<td>Employee Id. : </td>
				<td><input type="text" name="empid"></td>
			</tr>
			<tr>
				<td>Name : </td>
				<td><input type="text" name="ename"></td>
			</tr>
			<tr>
				<td>Designation : </td>
				<td><input type="text" name="desig"></td>
			</tr>
			<tr>
				<td>Department</td>
				<td><input type="text" name="dept"></td>
			</tr>

			 <tr>
				<td>DOJ</td>
				<td><input type="text" name="doj"></td
```

```
                </tr>
                <tr>
                    <td>Salary</td>
                    <td><input type="text" name="salary"></td>
                </tr>
                <tr>
                    <td colspan="2" align="center"><input type="submit"
                       name="Submit" value="Add"></td>
                </tr>
            </table>
        </form>
    </body>

    </html>
```

add.php

```
    <html>
    <head>
        <title>Add Employee Details</title>
    </head>
    <body>
    <?php
    //including the database connection file
        include_once("config.php");
        $empid = $_POST['empid'];
        $ename = $_POST['ename'];
        $desig = $_POST['desig'];
        $dept = $_POST['dept'];
        $doj = $_POST['doj'];
        $salary = $_POST['salary'];

            if(isset($_POST['Submit'])) {
                //insert data to database
            $result = mysqli_query($mysqli, "INSERT INTO empdetails values ($empid,
                '$ename','$desig','$dept','$doj',$salary)");
            //display success message
            echo "<h1 align='center'>Add Employee Details</h1>";echo
            "<hr />";
            echo "<font color='green'>Data added successfully.";echo
            "<br/><a href='index.php'>View Result</a>";
        }
        ?>
    </body>
    </html>
```

edit.php

```
    <?php
        // including the database connection file
        include_once("config.php");
        if(isset($_POST['update']))
        {
            $empid = $_POST['empid'];
            $ename = $_POST['ename'];
```

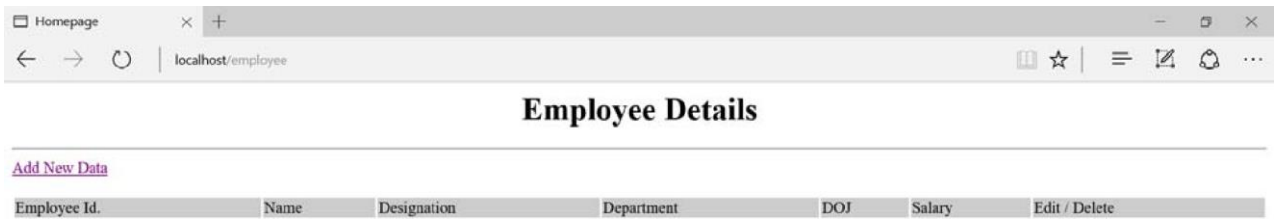
```
            $desig = $_POST['desig'];
            $dept = $_POST['dept'];
            $doj = $_POST['doj'];
            $salary = $_POST['salary'];

            //updating the table
            $result = mysqli_query($mysqli, "UPDATE empdetails SET ename='$ename',
                  desig='$desig',dept='$dept',doj='$doj',salary=$salary WHERE empid=$empid");

            //redirectig to the display page. In our case, it is index.phpheader("Location:
            index.php");
      }
?>

<?php
      echo "<h1 align='center'>Edit Employee Details</h1>";echo "<hr
      />";
      //getting id from url
      $empid = $_GET['empid'];
      //selecting data associated with this particular eid
      $result = mysqli_query($mysqli, "SELECT * FROM empdetails WHERE
         empid=$empid");
      while($res = mysqli_fetch_array($result))
      {
            $empid = $res['empid'];
            $ename = $res['ename'];
            $desig = $res['desig'];
            $dept = $res['dept'];
            $doj = $res['doj'];
            $salary = $res['salary'];
      }
?>
<html>
<head>
      <title>Edit Employee Details</title>
</head>
<body>
      <a href="index.php">Home</a>
      <br/><br/>

      <form name="empform" method="post" action="edit.php">
      <table border="0">
      <tr>
      <td>Name : </td>
      <td><input type="text" name="ename" value="<?php echo $ename;?>"></td>
      </tr>
      <tr>
      <tr>
      <td>Designation : </td>
      <td><input type="text" name="desig" value="<?php echo $desig;?>"></td>
      </tr>
      <tr>
      <td>Department : </td>
      <td><input type="text" name="dept" value="<?php echo $dept;?>"></td>
      </tr>
```

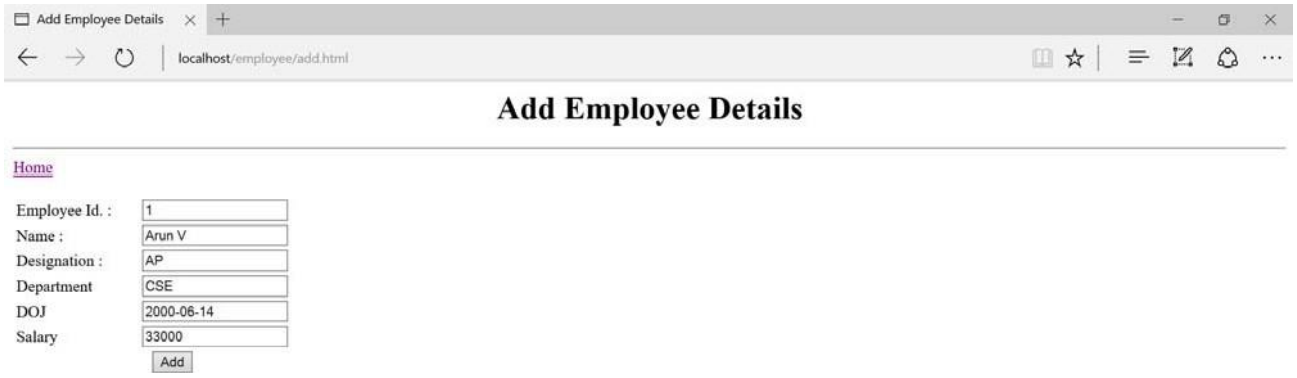
```
        <td>DOJ : </td>
        <td><input type="text" name="doj" value="<?php echo $doj;?>"></td>
        </tr>
        <tr>
        <td>Salary</td>
        <td><input type="text" name="salary" value="<?php echo $salary;?>"></td>
        </tr>
        <tr>
    <td><input type="hidden" name="empid" value=<?php echo $_GET['empid'];?>></td>
        <td><input type="submit" name="update" value="Update"></td>
        </tr>
        </table>
        </form>
    </body>
    </html>
```

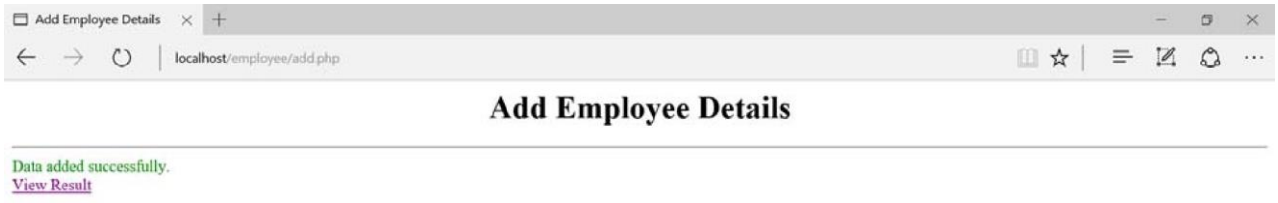
delete.php

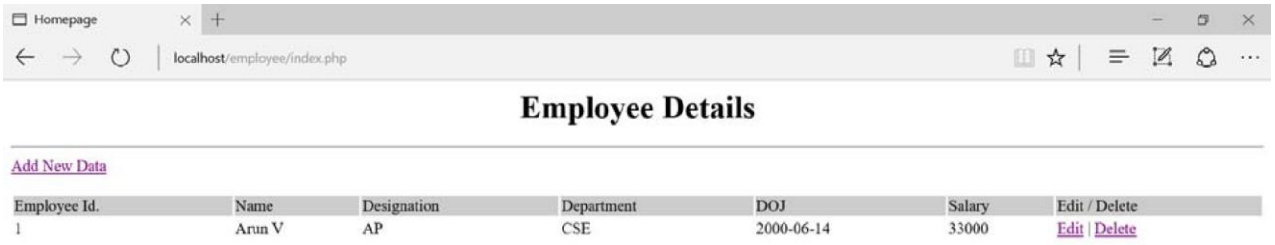
```
    <?php
    //including the database connection file
    include("config.php");
    //getting id of the data from url
    $empid = $_GET['empid'];
    //deleting the row from table
    $result = mysqli_query($mysqli, "DELETE FROM empdetails WHERE empid=$empid");
    //redirecting to the display page (index.php in our case)
    header("Location:index.php");
    ?>
```

OUTPUT:




Add Employee Details
localhost/employee/add.html
Add Employee Details
Home
Employee Id. :
1
Name :
Arun V
Designation :
AP
Department
CSE
DOJ
2000-06-14
Salary
33000
Add


Add Employee Details
localhost/employee/add.php
Add Employee Details
Data added successfully.
View Result


Homepage
localhost/employee/index.php
Employee Details
Add New Data
Employee Id.
Name
Designation
Department
DOJ
Salary
Edit / Delete
1
Arun V
AP
CSE
2000-06-14
33000
Edit | Delete

Ex no 6 **BOOTSTRAP – WEB PAGE**

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 lang="en">
  <head>
    <meta charset="utf-8" />
    <meta
      name="viewport"
      content="width=device-width, initial-scale=1, shrink-to-fit=no"
    />
    <meta
      name="description"
      content="A portfolio."
    />
    <meta name="author" content="pf" />
    <title>My protfolio</title>
    <link rel="icon" type="image/x-icon" href="./favicon.ico" />
    <link rel="stylesheet" href="./bootstrap.min.css" />
    <link rel="stylesheet" href="./main.css" />
    <script>
      function computeDarkTheme(isEnable) {
        if (isEnable) {
          document.body.classList.add("theme-dark");
          document.querySelector("#theme-img").src = "./pic_bulboff.gif";
          localStorage.setItem("theme", "dark");
        } else {
```

```
          document.body.classList.remove("theme-dark");
          document.querySelector("#theme-img").src = "./pic_bulbon.gif";
          localStorage.setItem("theme", "light");
        }
      }

      window.onload = () => {
        let theme = localStorage.getItem("theme");
        let time = new Date().getHours();

        if (theme === "dark") {
          computeDarkTheme(true);
        } else if ((theme == null && time >= 18) || time <= 5) {
          computeDarkTheme(true);
        } else {
          computeDarkTheme(false);
        }
      };
    </script>
  </head>
  <body data-spy="scroll" data-target=".navbar" data-offset="50">
    <div id="root"></div>
  </body>
</html>
```

CSS:

```
@import url('https://fonts.googleapis.com/css?family=Saira+Extra+Condensed:500,700');
/* Globals */
:root {
  --primary: #bd5d38;
  --gradient: linear-gradient(90deg, #bd5d38, #ff6e36);
}
html {
  scroll-behavior: smooth;
}
h1,
h2,
h3,
h4,
h5,
h6,
.h1,
.h2,
.h3,
.h4,
.h5,
.h6 {
  margin-bottom: 0.5rem;
  font-family: 'Saira Extra Condensed', sans-serif;
  font-weight: 700;
  line-height: 1.2;
  color: #343a40;
  text-transform: uppercase;
}
```

```
h1,
.h1 {
  font-size: 6rem;
}
h2,
.h2 {
  font-size: 3.5rem;
}
h3,
.h3 {
  font-size: 2rem;
}
h4,
.h4 {
  font-size: 1.5rem;
}
h5,
.h5 {
  font-size: 1.25rem;
}
h6,
.h6 {
  font-size: 1rem;
}
body {
  position: relative;
  padding-top: 3.375rem;
  color: #6c757d;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen',
    'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
    sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}
.btn-custom,
.btn-primary {
  border: none;
  background: var(--gradient) !important;
}
p.lead {
  font-size: 1.15rem;
  font-weight: 400;
}
.text-primary {
  color: var(--primary) !important;
}
#sideNav .navbar-nav .nav-link {
  font-weight: 800;
  letter-spacing: 0.05rem;
  text-transform: uppercase;
  transition: 0.3s;
}
```

```
#sideNav .navbar-toggler:focus {
  outline-color: #d48a6e;
}
.social-icons .social-icon {
  display: inline-flex;
  align-items: center;
  justify-content: center;
  height: 3.5rem;
  width: 3.5rem;
  background-color: #495057;
  color: #fff;
  border-radius: 100%;
  font-size: 1.5rem;
  margin-right: 1.5rem;
}
.social-icons .social-icon:last-child {
  margin-right: 0;
}
.social-icons .social-icon:hover {
  background-color: #bd5d38;
}
.dev-icons {
  font-size: 3rem;
}
section.resume-section {
  display: flex;
  align-items: center;
  padding-left: 1rem;
  padding-right: 1rem;
  padding-top: 5rem;
  padding-bottom: 5rem;
  max-width: 75rem;
}
section.resume-section .resume-section-content {
  width: 100%;
}
/* Break */
#sideNav {
  background-size: cover;
  background-position: center center;
}
#sideNav::before {
  content: '';
  position: absolute;

  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  z-index: -1;
  opacity: 0.98;
}
.cards {
  display: grid;
```

```
  grid-gap: 1em;
  grid-template-columns: repeat(auto-fill, minmax(320px, 1fr));
  align-items: start;
}
.cards .card .title {
  font-size: 24px;
  text-transform: uppercase;
}
.cards .card .card-thumbnail {
  position: relative;
  width: 100%;
  padding-top: 52.26%;
  background-size: cover;
  cursor: pointer;
  transition: all 3s ease;
}
.cards .card .card-thumbnail .overlay {
  position: absolute;
  height: 100%;
  width: 100%;
  align-items: center;
  justify-content: center;
  background-color: #f5f5f5;
}
.cards .card .card-thumbnail .overlay img {
  transform: translateX(-12%);
}
.cards .card .card-thumbnail:hover {
  background-position: bottom center;
}
.cards .btn-grp {
  width: 100%;
  display: grid;
  grid-template-columns: repeat(2, 1fr);
  grid-gap: 1em;
}

#skills .progress {
  height: 1.5rem;
}
#skills .progress .progress-bar {
  background-color: var(--primary) !important;
}
/* Contact */
#contact {
  padding-bottom: 0 !important;
}
#contact .contact-map iframe {
  border-radius: 10px;
  overflow: hidden;
  width: 100%;
  height: 200px;
}
```

```
#contact .contact-direct {
  margin-bottom: 15px;
}
#contact .contact-direct .item {
  display: flex;
  align-items: center;
}
#contact .contact-direct .item p {
  border-left: 1px solid lightgrey;
  padding-left: 20px;
  margin: 0;
}
#contact .contact-direct .item span {
  color: #ff6e36;
  font-size: 24px;
  padding: 20px;
}
::-webkit-scrollbar {
  width: 12px;
}
::-webkit-scrollbar-thumb {
  background: linear-gradient(to bottom, transparent, rgba(203, 100, 61, 0.92));
  border-radius: 6px;
}

/* iframe {
  display: none !important;
} */

/* Media query */
@media (max-width: 992px) {
  #sideNav {
    flex-direction: row-reverse;
  }

  #sideNav .navbar-collapse {
    /* display: block; */
    position: absolute;
    top: 0;
    left: 0;
    padding: 0px 1rem;
    padding-top: 18px;
    background: #cb643d;
    box-shadow: rgb(0 0 0 / 40%) 10px 0px 0px 100vw;
  }

  #sideNav .nav-img img {
    top: 0;
    left: 0;
    width: 50px;
    margin: 0px !important;
  }
```

```
  #sideNav .navbar-brand {
    padding: 0px !important;
    margin: 0px !important;
  }

  #sideNav .navbar-collapse .navbar-nav {
    margin: 15px 0;
    padding: 0 15px;
    /* background: purple; */
  }

  /* #sideNav .navbar-collapse .navbar-nav {
    margin: 15px 0;
    padding: 0 15px;
    background: white;
    border-top: 1px solid rgba(255, 255, 255, 0.25);
  }
  #sideNav .navbar-collapse .navbar-nav .nav-link {
    font-weight: 600;
    color: var(--primary);
    padding: 16px 0;
  }
  #sideNav .navbar-collapse .navbar-nav .nav-link.active {
    font-weight: 700;
  }
  #about > div > h1 {
    font-size: 5rem;
  } */
}
@media (max-width: 768px) {
  /* section.resume-section {
    padding-top: 3rem;
    padding-left: 10px;
    padding-right: 10px;
  } */
  #about > div > h1 {
    font-size: 4rem;
    line-height: 1;
  }
}
@media screen and (max-width: 520px) {
  /* Show 16:9 in mobile devices */
  #about > div > h1 .text-primary {
  font-size: 5rem;
    display: block;
  }
  .codepen-cards .codepen-card {
    padding-top: calc(16 / 9 * 100%);
  }
}
@media screen and (max-width: 400px) {
  .social-icons .social-icon {
    margin: 0px 5px;
  }
```

```
}

@media (max-width: 720px) {
  .social-icons .social-icon {
    margin: 5px;
  }
}

@media (min-width: 768px) {
  section.resume-section {
    min-height: 100vh;
  }
}
@media (min-width: 992px) {
  section.resume-section {
  padding-left: 3rem;
    padding-right: 3rem;
    padding-top: 5rem;
    padding-bottom: 5rem;
  }
}
@media (min-width: 992px) {
  body {
    padding-top: 0;
    padding-left: 17rem;
  }
}
@media (min-width: 992px) {
  #sideNav {
    text-align: center;
    position: fixed;
    top: 0;
    left: 0;
    display: flex;
    flex-direction: column;
    width: 17rem;
    height: 100vh;
  }

  #sideNav .navbar-brand {
    display: flex;
    margin: auto auto 0;
    padding: 0.5rem;
  }
  #sideNav .navbar-brand .img-profile {
    max-width: 10rem;
    max-height: 10rem;
    border: 0.5rem solid rgba(255, 255, 255, 0.2);
  }
  #sideNav .navbar-collapse {
    display: flex;
    align-items: flex-start;
    flex-grow: 0;
    width: 100%;
```

```
    margin-bottom: auto;
   }
   #sideNav .navbar-collapse .navbar-nav {
    flex-direction: column;
    width: 100%;
   }
   #sideNav .navbar-collapse .navbar-nav .nav-item {
    display: block;
   }
   #sideNav .navbar-collapse .navbar-nav .nav-item .nav-link {
    display: block;
   }
  }
```

OUTPUT:

Ex no 7 **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

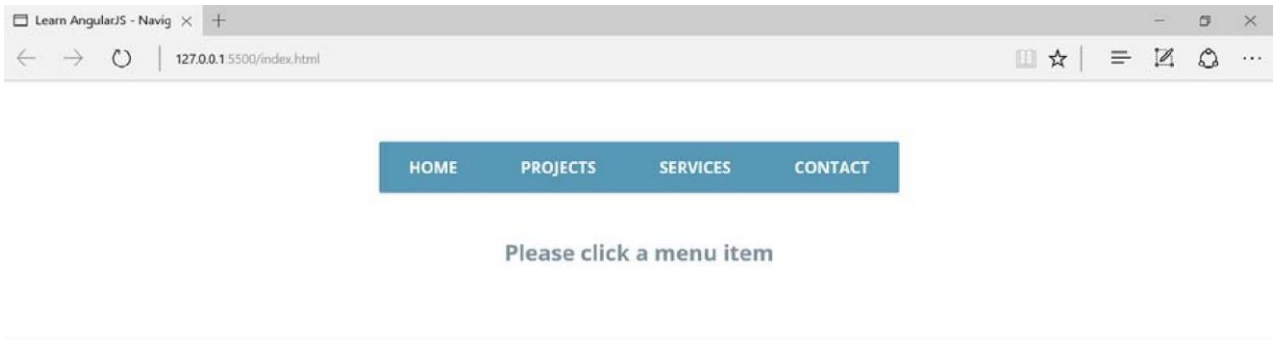
```
<!DOCTYPE html>
<html>

    <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>
```

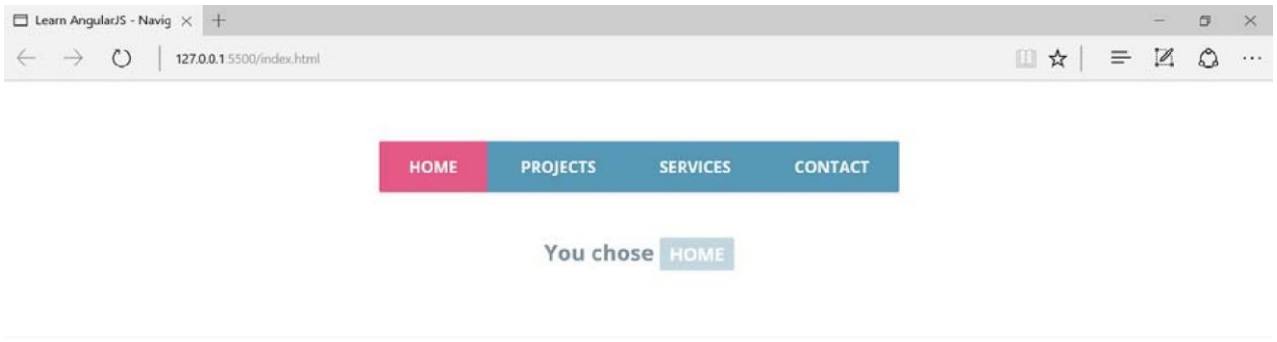
style.css

```
/*----------------------------------------------------
	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:


Learn AngularJS - Navig
127.0.0.1:5500/index.html
HOME
PROJECTS
SERVICES
CONTACT
Please click a menu item


Learn AngularJS - Navig
127.0.0.1:5500/index.html
HOME
PROJECTS
SERVICES
CONTACT
You chose HOME

Ex no 7b **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 -->
```

```
                <div class="tooltip" ng-click="$event.stopPropagation()"ng-
                show="showtooltip">

                    <!-- ng-model binds the contents of the text field with the"value" model.
                       Any changes to the text field will automatically update the value, and
                       all other bindings onthe page that depend on it.  -->

                    <input type="text" ng-model="value" />
                </div>

                <!-- Call a method defined in the InlineEditorController thattoggles the
                showtooltip varaible -->
                <p ng-click="toggleTooltip($event)">{{value}}</p>

            </div>

            <!-- 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

```
    // The controller is a regular JavaScript function. It is called
    // once when AngularJS runs into the ng-controller declaration.function

    InlineEditorController($scope){

        // $scope is a special object that makes
        // its properties available to the view as
        // variables. Here we set some default values:

        $scope.showtooltip = false;
        $scope.value = 'Edit me.';

        // Some helper functions that will be
        // available in the angular declarations

        $scope.hideTooltip = function(){

            // 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;
    }
```

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
                   remove the "active" css class if needed -->
                <li ng-repeat="service in services"ng-
```

```
                    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;
        };
    }
```

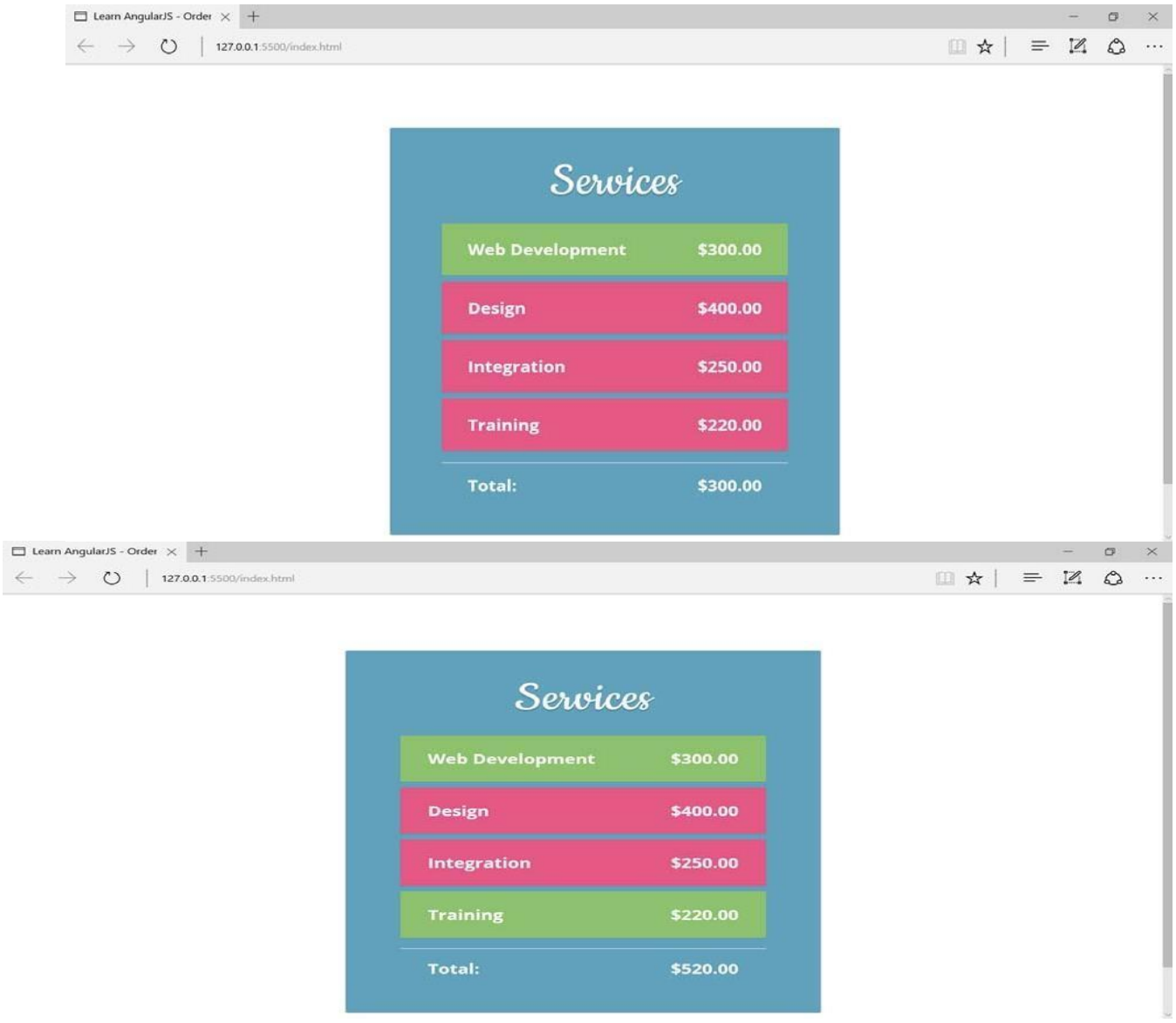
style.css

```
    /*-------------------------------------------
        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:



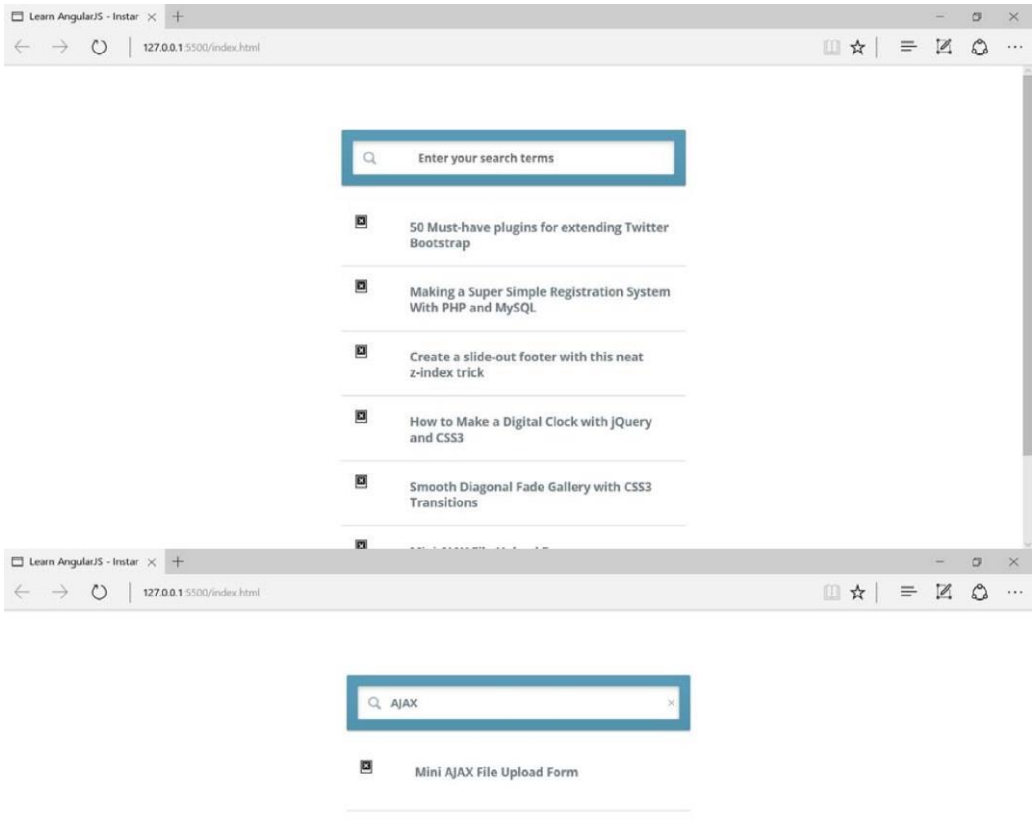
RESULT:

Thus an order form using Angular JS has been created successfully.

```
        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;
}

ul{
        list-style: none;
        width: 428px;
        margin: 0 auto; text-
        align: left
}
```

```
ul li{
        border-bottom: 1px solid #ddd;
        padding: 10px;
        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:



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){

        // Assigning the pics array will cause the view
        // to be automatically redrawn by Angular.
        $scope.pics = 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;
}
```

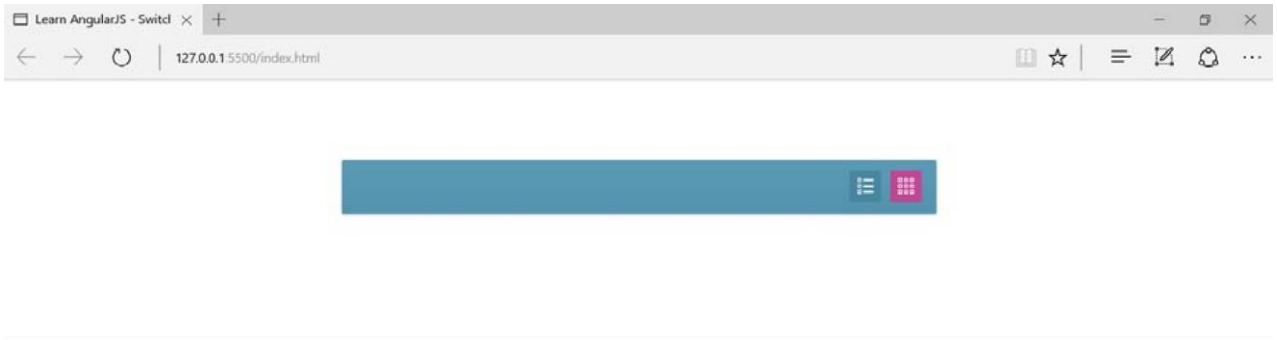
```
/*                    The search input
                      */

.bar{
    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: 2px;
    width: 580px;
    padding: 10px;
    margin: 80px auto 25px;
    position:relative;
    text-align:right;
    line-height: 1;
}
.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(
8/9hAAA
AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvb
S5hZG9iZS54bX
AAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3
prYzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFh
NUCBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgIj4
gPHJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbn
MjIj4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRv
YmUuY29tL
3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21t
LyIgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZ
VJlZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBXaW5kb3dzIiB4bXBNT
TpJbnN0YW5jZU
```

```
lEPSJ4bXAuaWlkOkYzNkFCQ0ZBMTBCRTExRTM5NDk4RDFEM0E5RkQ1NEZCIiB4bXB
NTTpEb2N1bWVud
ElEPSJ4bXAuZGlkOkYzNkFCQ0ZCMTBCRTExRTM5NDk4RDFEM0E5RkQ1NEZCIj4gPH
htcE1NOkRlcml2
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6RjM2QUJDRjgxMEJFMTFFMzk0
OThEMUQzQTl
GRDU0RkIiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6RjM2QUJDRjkxMEJFMTFFMz
k0OThEMUQzQT
lGRDU0RkIiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmRmOlJERj4gPC94OnhtcG1ldGE+ID
w/eHBhY
2tldCBlbmQ9InIiPz7h1bLqAAAAWUlEQVR42mL8////BwYGBn4GCACxBRlIAIxAA/4jaXoPEk
MyjJ+A
/g9MDJQBRhYg8RFqMwg8RJIUINYLFDmBUi+ADQAF1n8ofk9yIAy6WPg4GgtDMRYAAgw
AdLYwLAoIwPgAAAAASUVORK5CYII=);
}

.bar a.grid-icon{
    background-
image:url(
8/9hAAA
AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvb
S5hZG9iZS54bX
AAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3
prYzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFh
NUCBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgIj4
gPHJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbn
MjIj4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRv
YmUuY29tL
3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21t
LyIgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZ
VJlZiMiIHh
tcDpCcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBXaW5kb3dzIiB4bXBNT
TpJbnN0YW5jZU
lEPSJ4bXAuaWlkOjBEQkMyQzE0MTBCRjExRTNBMDlGRTYyOTlBNDdCN0I4IiB4bXBN
TTpEb2N1bWVud
ElEPSJ4bXAuZGlkOjBEQkMyQzE1MTBCRjExRTNBMDlGRTYyOTlBNDdCN0I4Ij4gPHhtc
E1NOkRlcml2
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6MERCQzJDMTIxMEJGMTFF
M0EwOUZFNjI5OUE
0N0I3QjgiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6MERCQzJDMTMxMEJGMTF
FM0EwOUZFNjI5OU
E0N0I3QjgiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmRmOlJERj4gPC94OnhtcG1ldGE+I
Dw/eHBhY
2tldCBlbmQ9InIiPz4MjPshAAAAXklEQVR42mL4////h/8I8B6IGaCYKHFGEMnAwCDIAAHv
gZgRyiZK
nImBQsACxB+hNoDAQyQ5osQZIT4gH1DsBZABH6AB8x/JaQzEig++WPiII7Rxio/GwmCIB
YAAAwAwVIzMp1R0aQAAAABJRU5ErkJggg==);
}
```

```
.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:

Thus the exercise has been completed successfully.

Ex no 8 **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 Courses Contact Us

Home

Welcome to REC

Home Courses Contact Us

Courses

AERO, AUTO, BIOMED, BIOTECH, CHEMICAL, CIVIL, CSE, CSBC, ECE, EEE, FT, IT, MCT, MECH

Home Courses Contact Us

Contact Us

Rajalakshmi Nagar, Thandalam, Chennai - 602 105

RESULT:

Thus a single page application has been created successfully.