RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105

Laboratory Record Note Book

Name : . . . .

Year / Branch / Section : . . . .

Register No. : . . . .

College Roll No. : . . . .

Semester : . . . .

Academic Year : . . . .

211701010

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 : ______

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 as a user side image map.

PROGRAM:

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>MapImage</title>
  </head>
  <body>
    <map id="picture" name="picture">
      <area href="TamilNadu.html" shape="rect" coords="418,902,322,1063"
alt="TamilNadu"/>
      <area href="karnataka.html" shape="rect" coords="217,951,309,758" alt
="Karnataka"/>
      <area href="Maharashtra.html" shape="rect" coords="157,754,338,613"
alt="Maharashtra"/>
      <area href="rajasthan.html" shape="rect" coords="63,484,309,341" alt="Rajasthan"/>
      <area href="Andhra.html" shape="rect" coords="312,892,428,796"
alt="AndhraPradesh"/>
    </map>
    <img src="C:\Users\Hari\Downloads\india.jpg"  alt="india"  usemap="#picture"/>
  </body>
</html>
```

TamilNadu.html:

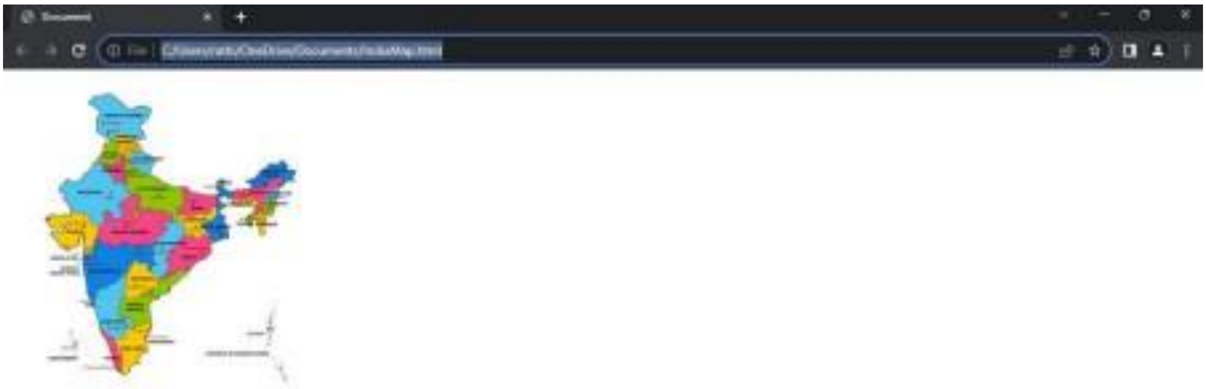
```
<!DOCTYPE html>
<html lan="en">
```

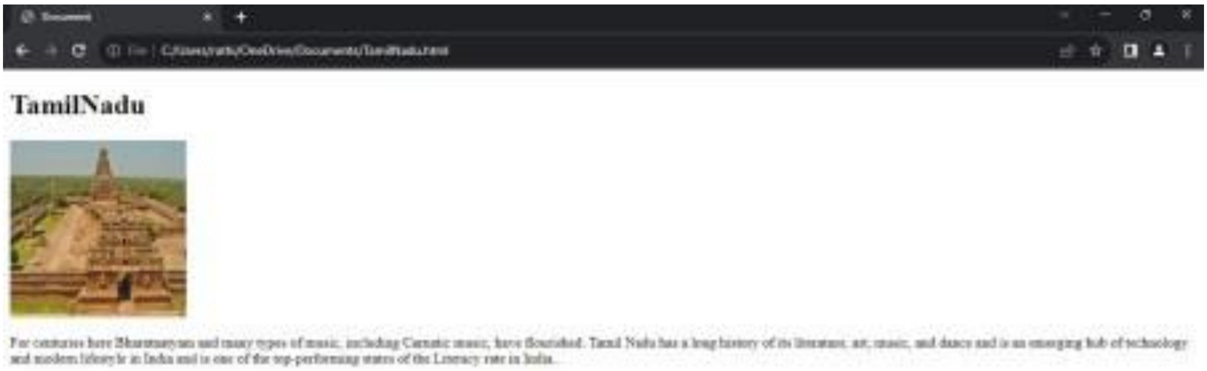
```
    <head>
        <title>TamilNaduImage</title>
    </head>
    <style>
        body {background-color: rgb(247, 250, 250);}
        h1{color:rgb(91, 5, 5)}
        p{color:rgb(45, 45, 61); text-align :justify}
        * {
          box-sizing:border-box;
}

.column {
  float: right;
  width: 30%;
  padding: 5px;
}

    </style>
    <body>
        <h1>TamilNadu</h1>
        <p>Tamil Nadu is straddled by the Western Ghats and Deccan Plateau in the west, the
Eastern Ghats in the north, the Eastern Coastal Plains lining the Bay of Bengal in the east, the
Gulf of Mannar and the Palk Strait to the south-east, the Laccadive Sea at the southern cape
of the peninsula, with the river Kaveri bisecting the state. Politically, Tamil Nadu is bound by
the Indian states of Kerala, Karnataka, and Andhra Pradesh, and the union territory of
Puducherry and shares an international maritime border with the Northern Province of Sri
Lanka at Pamban Island.

            Archaeological evidence points to Tamil Nadu being inhabited for more than 400
millennia and has more than 5,500 years of continuous cultural history. Historically, the
Tamilakam region was inhabited by Tamil-speaking Dravidian people and was ruled by
several regimes over centuries, such as the Sangam era triumverate of the Cheras, Cholas and
Pandyas, the Pallavas (3rd–9th century CE), and the later Vijayanagara Empire (14th–17th
century CE).<br><br><br> </p>
        <div class="row">
            <div class="column">
              <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/Left_side_view_Brihadee
swara.jpg/1280px-Left_side_view_Brihadeeswara.jpg" alt="Snow" style="width:90%">
            </div>
            <div class="column">
              <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/f/f4/Statue_of_Thiruvalluvar.j
pg/1024px-Statue_of_Thiruvalluvar.jpg" alt="Forest" style="width:90%">
            </div>
            <div class="column">
              <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/5/52/Hogenakkal_Falls_Close.
jpg/1024px-Hogenakkal_Falls_Close.jpg" alt="Mountains" style="width:90%">
            </div>
          </div>
    </body>
</html>
```

Karnataka.html:

```
<!DOCTYPE html>
<html lan="en">
    <head>
        <title>KarnatakaImage</title>
    </head>
    <style>
        body {background-color: rgb(247, 250, 250);}
        h1{color:rgb(91, 5, 5)}
        p{color:rgb(45, 45, 61); text-align :justify}
        * {
          box-sizing:border-box;
}

.column {
  float: right;
  width: 30%;
  padding: 5px;
}

    </style>
    <body>
        <h1>Karnataka</h1>
        <p>Karnataka is derived from the Kannada words karu and nādu, meaning "elevated
land". Karu Nadu may also be read as karu, meaning "black" and nadu, meaning "region", as
a reference to the black cotton soil found in the Bayalu Seeme region of the state. The British
used the word Carnatic, sometimes Karnatak, to describe both sides of peninsular India, south
of the Krishna.<br>With an antiquity that dates to the paleolithic,  Karnataka has been home
to some of the most powerful empires of ancient and medieval India. The philosophers and
musical bards patronised by these empires launched socio-religious and literary movements
which have endured to the present day. Karnataka has contributed significantly to both forms
of Indian classical music, the Carnatic and Hindustani traditions..<br><br><br> </p>
        <div class="row">
            <div class="column">
              <img src="https://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/7th_-
_9th_century_Hindu_and_Jain_temples%2C_Pattadakal_monuments_Karnataka_5.jpg/1024
px-7th_-
_9th_century_Hindu_and_Jain_temples%2C_Pattadakal_monuments_Karnataka_5.jpg"
alt="Snow" style="width:90%">
            </div>
            <div class="column">
              <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/Gopura_at_Murdeshwar.j
pg/1024px-Gopura_at_Murdeshwar.jpg" alt="Forest" style="width:90%">
            </div>
            <div class="column">
              <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/d/d3/DGTF2318.jpg/1024px-
DGTF2318.jpg" alt="Mountains" style="width:90%">
            </div>
          </div>
    </body>
</html>
```

OUTPUT:





TamilNadu

For centuries here Bharatanatyam and many types of music, including Carnatic music, have flourished. Tamil Nadu has a long history of its literature, art, music, and dance and is an emerging hub of technology and modern lifestyle in India and is one of the top-performing states of the Literacy rate in India.

Result:

Thus, the given program is executed successfully.

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 lang="en">
   <head>
      <title> BEC</title>
   </head><title> BEC</title>
   <style>
      .topnav {
  background-color: #cfcccc;
  overflow:auto;
}

/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  color:rgb(14, 14, 44);
  text-align: center;
  padding: 14px 16px;
  text-decoration:double;
  font-weight: bold;
  font-size: 22px;
}

/* Change the color of links on hover*/
.topnav a:hover {
  background-color: #ddd;
  color: black;
}
```

```
/* Create a right-aligned (split) link inside the navigation bar */
.topnav a.split {
  float: right;
  background-color: #76b98d;
  color: rgb(255, 255, 255);
}
.center {
  position: absolute;
  top: 100px;
  right: 570px;
  font-size: 32px;
  text-align: center;
  font-weight: bolder;
  color:rgb(14, 14, 44) ;
}
.column {
  float: right;
  width: 32%;
  padding: 2px;
  padding-top: 50px;

}
img {
  opacity: 0.8;
  border-radius: 8px;
}

/*img:hover {
  opacity: 1.0;
}*/
.hw{
  text-align: center;
  font-size:25px;
  font-weight: bold;
  color:rgb(14, 14, 44);
  overflow: auto;
  padding-bottom: 50px;
}
.cen{
  position: absolute;
  top: 850px;
  right: 100px;
  left:100px;
  font-size: 28px;
  text-align: center;
  font-weight:500;
  color:rgb(242, 242, 246) ;
}
.cent{
  position: absolute;
  top: 135px;
  right: 500px;
  font-size: 20px;
  font-weight: lighter;
  text-align: center;
  color:rgb(30, 30, 102) ;
}
.A1{
  opacity: 1.0;
}

</style>
    <body>

        <div class="topnav">
            <a href="#home">Home</a>
            <a href="about.html">About</a>
            <a href="contact.html">News</a>
            <a href="#contact">Contact</a>
            <a href="#news" class="split">Help</a>
          </div>
        <div>
          <img src="https://images.unsplash.com/photo-1562774053-
701939374585?q=80&w=1000&auto=format&fit=crop&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8Nnx8Y29sbGVnZXxlbnwwfHwwfHx8MA
%3D%3D"  alt="Mountains" style="width:100%" >
          <div class="center">Best Engineering College</div>
          <div class="cent">An Autonomous Institution, Affiliated to Anna University.</div>
          <div class="cen">College is an educational school where people go to learn and earn
an academic degree upon graduation. The requirement to enter college is different in different
countries, but in most countries, the student must have a high school diploma or equivalent
</div>
        </div>
        <div class="row">
          <div class="column">
            <img class="A1"
src="https://t4.ftcdn.net/jpg/05/39/10/47/360_F_539104776_BchIZKRhIUXDY0ZaVHxaoI
DvRa2eAG3d.jpg" alt="Snow" style="width:90%">
            <div class="hw">Students</div>

          </div>
          <div class="column">
            <img class="A1" src="https://jeppiaarcollege.org/jeppiaar/wp-
content/uploads/2018/03/INFRASTRUCTURE-ABOUTUS.jpg" alt="Forest"
style="width:90%">
            <div class="hw">Infrastructure</div>

          </div>
          <div class="column">
```

```
            <img class="A1" src="https://npsschool.in/wp-content/uploads/2021/07/Computer-
Lab.webp" alt="Mountains" style="width:90%">
            <div class="hw">Labs and Classes</div>
          </div>
        </div>
    </body>
</html>
```

About.html:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title> BEC</title>
    </head><title> BEC</title>
    <style>
        .topnav {
  background-color: #cfcccc;
  overflow:auto;
}

/* Style the links inside the navigation bar */
.topnav a {
  float: left;
  color:rgb(14, 14, 44);
  text-align: center;
  padding: 14px 16px;
  text-decoration:double;
  font-weight: bold;
  font-size: 22px;
}

/* Change the color of links on hover*/
.topnav a:hover {
  background-color: #ddd;
  color: black;
}

/* Create a right-aligned (split) link inside the navigation bar */
.topnav a.split {
  float: right;
  background-color: #76b98d;
  color: rgb(255, 255, 255);
}
.center {
  position: absolute;
  top: 100px;
  right: 570px;
  font-size: 32px;
  text-align: center;
  font-weight: bolder;
  color:rgb(14, 14, 44) ;
}
.column {
  float: right;
  width: 32%;
  padding: 2px;
  padding-top: 50px;

}
img {
  opacity: 0.8;
  border-radius: 8px;
}

/*img:hover {
  opacity: 1.0;
}*/
.hw{
  text-align: center;
  font-size:25px;
  font-weight: bold;
  color:rgb(14, 14, 44);
  overflow: auto;
  padding-bottom: 50px;
}
.cen{
  position: absolute;
  top: 850px;
  right: 100px;
  left:100px;
  font-size: 28px;
  text-align: center;
  font-weight:500;
  color:rgb(242, 242, 246) ;
}
.cent{
  position: absolute;
  top: 135px;
  right: 500px;
  font-size: 20px;
  font-weight: lighter;
  text-align: center;
  color:rgb(30, 30, 102) ;
}
.A1{
  opacity: 1.0;
}

</style>
```

```
    <body>

        <div class="topnav">
            <a href="#home">Home</a>
            <a href="about.html">About</a>
            <a href="contact.html">News</a>
            <a href="#contact">Contact</a>
            <a href="#news" class="split">Help</a>
          </div>
        <div>
          <img src="https://images.unsplash.com/photo-1562774053-
701939374585?q=80&w=1000&auto=format&fit=crop&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8Nnx8Y29sbGVnZXxlbnwwfHwwfHx8MA
%3D%3D"  alt="Mountains" style="width:100%" >
          <div class="center">Best Engineering College</div>
          <div class="cent">An Autonomous Institution, Affiliated to Anna University.</div>
          <div class="cen">College is an educational school where people go to learn and earn
an academic degree upon graduation. The requirement to enter college is different in different
countries, but in most countries, the student must have a high school diploma or equivalent
</div>
        </div>
        <div class="row">
          <div class="column">
            <img class="A1"
src="https://t4.ftcdn.net/jpg/05/39/10/47/360_F_539104776_BchIZKRhIUXDY0ZaVHxaoI
DvRa2eAG3d.jpg" alt="Snow" style="width:90%">
            <div class="hw">Students</div>

          </div>
          <div class="column">
            <img class="A1" src="https://jeppiaarcollege.org/jeppiaar/wp-
content/uploads/2018/03/INFRASTRUCTURE-ABOUTUS.jpg" alt="Forest"
style="width:90%">
            <div class="hw">Infrastructure</div>

          </div>
          <div class="column">
            <img class="A1" src="https://npsschool.in/wp-content/uploads/2021/07/Computer-
Lab.webp" alt="Mountains" style="width:90%">
            <div class="hw">Labs and Classes</div>
          </div>
        </div>
    </body>
</html>
```

News.html:

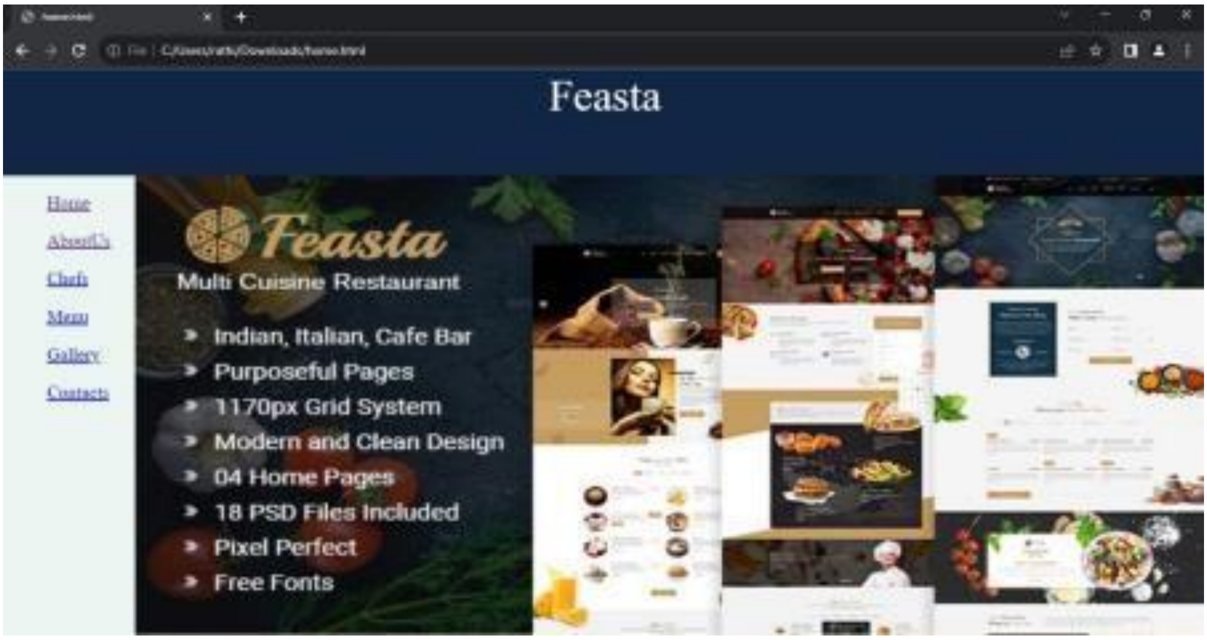
```
<!DOCTYPE html>
<html lang="en">
    <head>
```

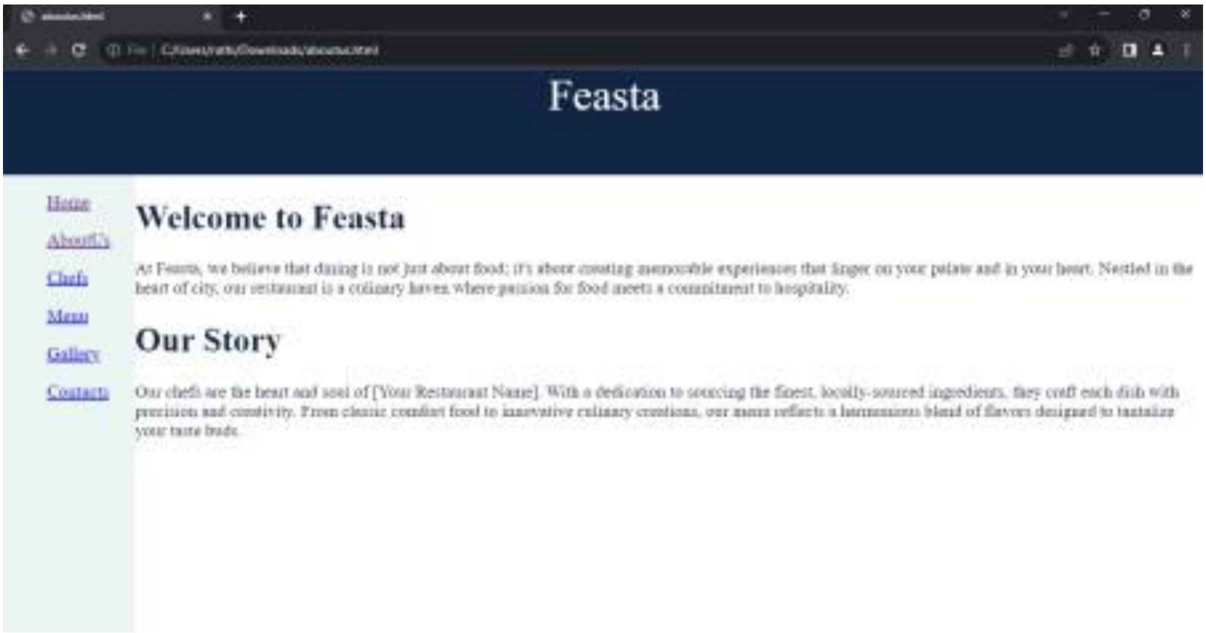
```
        <title>News</title>
    </head>
    <style>
        body{
            background-color: white;
        }
        p{
            padding-inline-start: 30px;
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            color:rgb(14, 14, 44);
            font-size: medium;
            text-align: justify;
        }
        .hw{
  text-align: center;
  font-size:25px;
  font-weight: bold;
  color:rgb(14, 14, 44);
  overflow: auto;
  padding-bottom: 50px;
}
.column {
  float: right;
  width: 32%;
  padding: 2px;
  padding-top: 50px;


}
    </style>
    <body>
        <h1>News:</h1>
        <p>There has also been a recent trend to rename or create government secondary
schools as "colleges". In the state of Victoria, some state high schools are referred to as
secondary colleges, although the pre-eminent government secondary school for boys in
Melbourne is still named Melbourne High School. In Western Australia, South Australia and
the Northern Territory, "college" is used in the name of all state high schools built since the
late 1990s, and also some older ones. In New South Wales, some high schools, especially
multi-campus schools resulting from mergers, are known as "secondary colleges". In
Queensland some newer schools which accept primary and high school students are styled
state college, but state schools offering only secondary education are called "State High
School". In Tasmania and the Australian Capital Territory, "college" refers to the final two
years of high school (years 11 and 12), and the institutions which provide this. In this context,
"college" is a system independent of the other years of high school. Here, the expression is a
shorter version of matriculation college.</p>
    </body>
    <div class="row">
        <div class="column">
          <img class="A1" src="https://files.delhievents.com/images/events/2018/february/R-K-
Films-Media-Festival-2018-Creative.jpg" alt="Snow" style="width:90%">
          <div class="hw">Students</div>
        </div>
        <div class="column">
          <img class="A1"
src="https://i.pinimg.com/originals/e1/c2/d0/e1c2d08481d6a8b2954078d3bb7d8ef4.jpg"
alt="Forest" style="width:90%">
          <div class="hw">Infrastructure</div>

        </div>
        <div class="column">
          <img class="A1" src="https://pbs.twimg.com/media/DMZaNJJW4AA25Ak.jpg"
alt="Mountains" style="width:90%">
          <div class="hw">Labs and Classes</div>
        </div>
      </div>
</html>
```

Style.css:

```
.Content{margin-left:0px;
Display:flex;
Flex direction:row;}
```

OUTPUT:





Feasta

Home

AboutUs

Chefs

Menu

Gallery

Contacts

Welcome to Feasta

At Feasta, we believe that dining is not just about food; it's about creating memorable experiences that linger on your palate and in your heart. Nestled in the heart of city, our restaurant is a culinary haven where passion for food meets a commitment to hospitality.

Our Story

Our chefs are the heart and soul of [Your Restaurant Name]. With a dedication to sourcing the finest, locally-sourced ingredients, they craft each dish with precision and creativity. From classic comfort food to innovative culinary creations, our menu reflects a harmonious blend of flavors designed to tantalize your taste buds.

Result:

Thus, the given program is executed successfully.

Ex.No:3 Creation of registration page along with validations.

AIM:

To create a registration form with validations.

ALGORITHM:

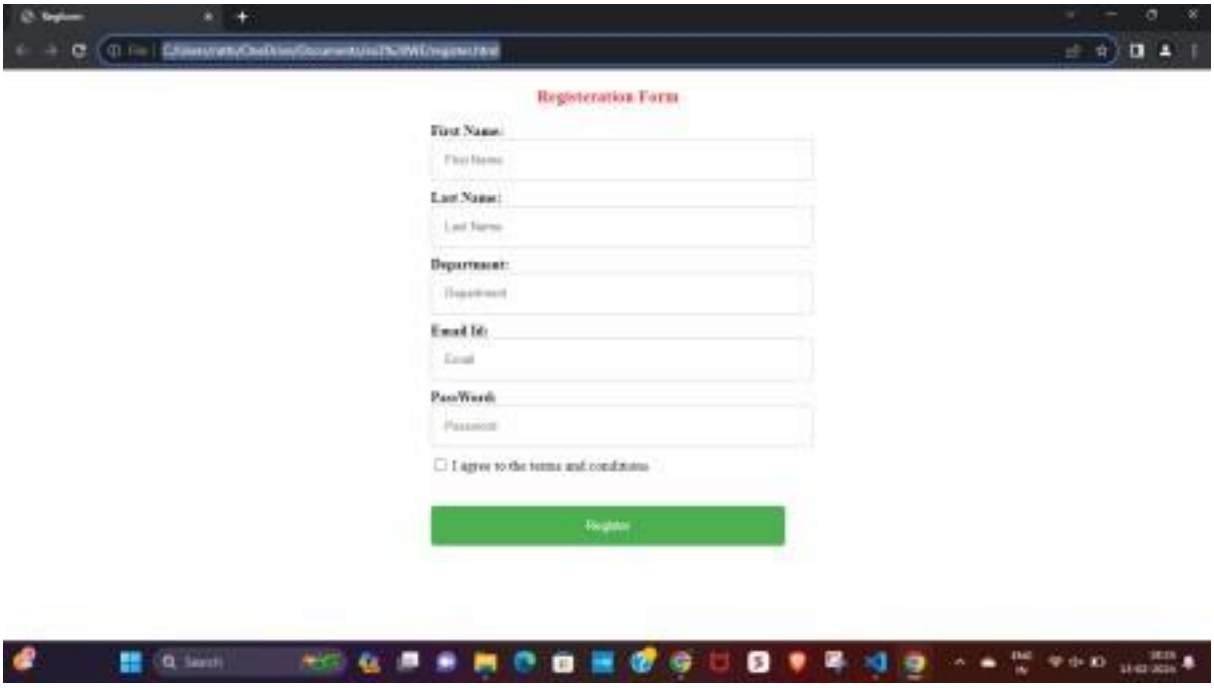
1. Create a registration form interface with HTML and CSS.
2. Get inputs from the user using the input tag.
3. Set appropriate tables above the input boxes.
4. Finally call the java script file using @media to access the file.
5. Perform validation and display alert message validation when submitted.

PROGRAM:

Reg.html:

```
<!DOCTYPE html>
<html lan="en">
    <head>
        <title>Regform</title>
    </head>
    <style>

.registration-form {
  width: 100%;
  max-width: 400px;
  margin: 0 auto;
  top:100px;

}
.registration-form input[type="text"], .registration-form input[type="email"], .registration-form input[type="password"] {
  width: 100%;
  padding: 15px;
  margin-bottom: 10px;
  border-radius: 5px;
  border: 1px solid #ccc;
}
.registration-form input[type="submit"] {
  width: 100%;
  padding: 15px;
  background-color: #4CAF50;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
@media (max-width: 600px) {
  .registration-form input[type="text"], .registration-form input[type="email"], .registration-
form input[type="password"] {
    width: 100%;
  }
}
p{
  text-align: center;
  font-weight:bold ;
  font-size: larger;
  color:blue;
}

</style>
<body>
  <div class="back">
    <img src="https://images.unsplash.com/photo-1562774053-
701939374585?q=80&w=1000&auto=format&fit=crop&ixlib=rb-
4.0.3&ixid=M3wxMjA3fDB8MHxzZWFyY2h8Nnx8Y29sbGVnZXxlbnwwfHwwfHx8MA
%3D%3D"  alt="Mountains" style="width:100%" opacity="0.5">

  </div>

<form  class="registration-form">
  <p>Registeration Form</p>
  <label for="first"><b>First Name:</b></label>
  <input type="text" name="first" placeholder="First Name" required>
  <label for="last"><b>Last Name:</b></label>
  <input type="text" name="last" placeholder="Last Name" required>
  <label for="dept"><b>Department:</b></label>
  <input type="text" name="dept" placeholder="Department" required>
  <label for="email"><b>Email Id:</b></label>
  <input type="email" name="email" placeholder="Email" required>
  <label for="pass"><b>PassWord:</b></label>
  <input type="password" name="pass" placeholder="Password" required>
  <input type="checkbox" required> I agree to the terms and conditions.
  <br><br><br>
  <input type="submit" value="Register">

</form>
</body>
</html>
```

OUTPUT:



Registeration Form

First Name:

First Name

Last Name:

Last Name

Department:

Department

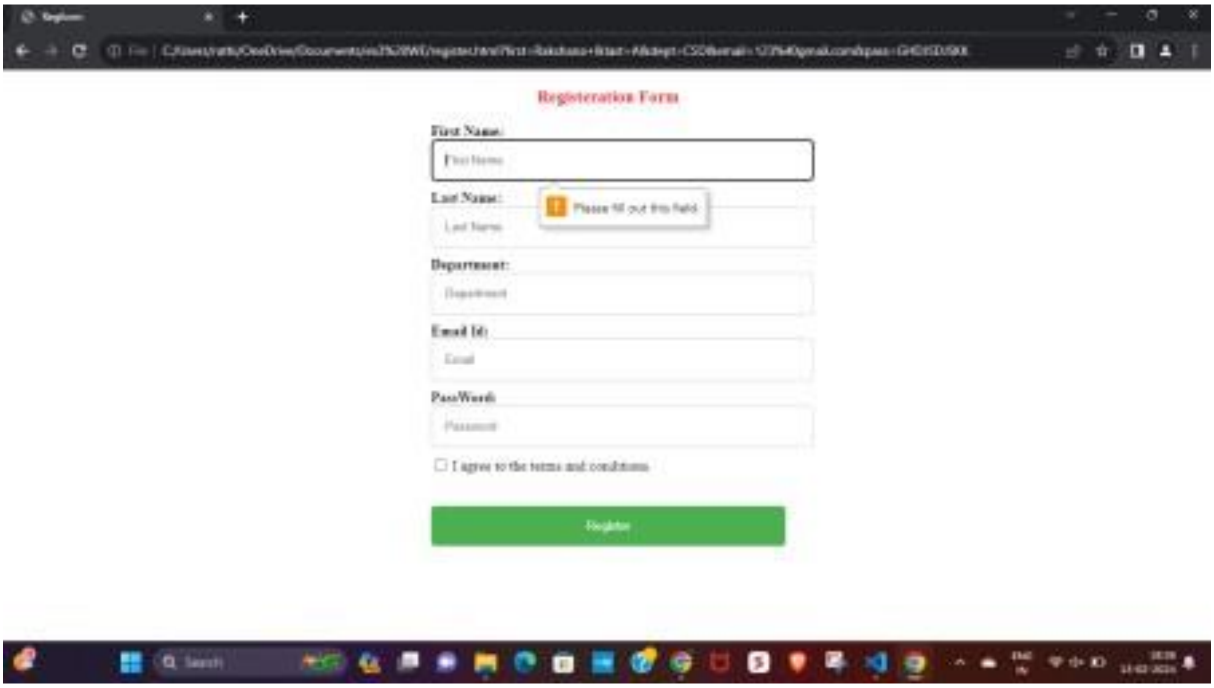
Email Id:

Email

PassWord:

Password

□ I agree to the terms and conditions



Registeration Form

First Name:

First Name

Please fill out this field.

Last Name:

Last Name

Department:

Department

Email Id:

Email

PassWord:

Password

□ I agree to the terms and conditions

Result:

Thus, the given program is executed successfully.

Ex.No:4 JSP –Library Management System

AIM:

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:

add-author.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">


<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Add Authors</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
    <div th:insert="comman/header :: header"></div>
    <div class="container my-5">
        <h3>Add Authors</h3>

        <div class="card-body">
            <div class="col-md-10">
                <form action="#" th:action="@{/add-author}" th:object="${author}"
                    method="post">
                    <div class="row">

                        <div class="form-group col-md-8">
                            <label for="name" class="col-form-label">Author
Name</label> <input
                                type="text" th:field="*{name}"
class="form-control" id="name"
                                placeholder="Author Name" />

                        </div>

                        <div class="form-group col-md-8">
                            <label for="description" class="col-form-
label">Description</label>
                            <input type="text" th:field="*{description}"
class="form-control"
                                id="description" placeholder="Description"
/>

                        </div>

                        <div class="col-md-6">
                            <input type="submit" class="btn btn-success"
value=" Submit ">

                        </div>
                    </div>
                </form>
```

```
                    </div>
               </div>
          </div>


          <div th:insert="comman/footer :: footer"></div>

</body>

</html>
```

Update-category.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Update Category</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
          href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
          href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
          <div th:insert="comman/header :: header"></div>
          <div class="container my-5">
               <h3>Update Category</h3>
               <div class="card">
                    <div class="card-body">
                         <div class="col-md-10">
                              <form action="#"
                                   th:action="@{/update-category/{id}(id=${category.id})}"
                                   th:object="${category}" method="post">
                                   <div class="row">

                                        <div class="form-group col-md-8">
                                             <label for="name" class="col-form-
label">Category Name</label> <input
                                                  type="text" th:field="*{name}"
class="form-control" id="name"
                                                  placeholder="Category Name" />

                                        </div>

                                        <div  class="col-md-6">
                                             <input type="submit" class="btn btn-
success" value=" Submit ">

                                        </div>
```

```
                </div>
                          </div>
                    </form>
```

```
                        </div>
                </div>
        </div>

        <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

Update-book.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Update Book</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
        href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
        <div th:insert="comman/header :: header"></div>
        <div class="container my-5">
                <h3>Update Book</h3>
                <div class="card">
                        <div class="card-body">
                                <div class="col-md-10">
                                        <form action="#" th:action="@{/update-
book/{id}(id=${book.id})}"
                                                th:object="${book}" method="post">
                                                <div class="row">

                                                        <div class="form-group col-md-8">
                                                                <label for="isbn" class="col-form-
label">ISBN</label> <input
                                                                        type="text" th:field="*{isbn}"
class="form-control" id="isbn"
                                                                        placeholder="ISBN" /> <span
                                                                        th:if="${#fields.hasErrors('isbn')}"
th:errors="*{isbn}"></span>

                                                        </div>



label">Book Name</label> <input

class="form-control" id="name"
```

```
<div class="form-group col-md-
8">
        <label

                for="nam

                e"
                                                        class="col-form- type="text"

                                                        th:field="*{name}"

                                                        placeholder="Book Name" />


                                              </div>
```

```
							<div class="form-group col-md-8">
								<label for="serialName" class="col-form-
label">Serial
									Name </label> <input type="text"
th:field="*{serialName}"
									class="form-control"
id="serialName" placeholder="Serial Name" />

							</div>

							<div class="form-group col-md-8">
								<label for="description" class="col-form-
label">Description</label>
								<input type="text" th:field="*{description}"
									class="form-control"
id="description" placeholder="Description" />

							</div>

							<div class="col-md-6">
								<input type="submit" class="btn btn-
success" value=" Submit ">

							</div>

						</div>
					</form>
				</div>
			</div>
		</div>
	</div>

	<div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

Update-publisher.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Update Publisher</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
	href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
	href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
	<div th:insert="comman/header :: header"></div>
```

```
<div class="container my-5">
        <h3>Update Publisher</h3>
        <div class="card">
            <div class="card-body">
                <div class="col-md-10">
                    <form action="#"
                        th:action="@{/update-publisher/{id}(id=${publisher.id})}"
                        th:object="${publisher}" method="post">
                        <div class="row">

                            <div class="form-group col-md-8">
                                <label for="name" class="col-form-
label">Publisher Name</label>
                                <input type="text" th:field="*{name}"
class="form-control"
                                    id="name" placeholder="Publisher
Name" />

                            </div>

                            <div  class="col-md-6">
                                <input type="submit" class="btn btn-
success" value=" Submit ">

                            </div>

                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

    <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

Update-author.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Update Author</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
    <div th:insert="comman/header :: header"></div>
    <div class="container my-5">
```

```
<h3>Update Author</h3>
<div class="card">
```

```
                        <div class="card-body">
                              <div class="col-md-10">
                                    <form action="#"
                                          th:action="@{/update-author/{id}(id=${author.id})}"
                                          th:object="${author}" method="post">
                                          <div class="row">

                                                <div class="form-group col-md-8">
                                                      <label for="name" class="col-form-
label">Author Name</label> <input
                                                            type="text" th:field="*{name}"
class="form-control" id="name"
                                                            placeholder="Author Name" />

                                                </div>


                                                <div class="form-group col-md-8">
                                                      <label for="description" class="col-form-
label">Description</label>
                                                      <input type="text" th:field="*{description}"
                                                            class="form-control"
id="description" placeholder="Description" />

                                                </div>

                                                <div  class="col-md-6">
                                                      <input type="submit" class="btn btn-
success" value=" Submit ">

                                                </div>

                                          </div>
                                    </form>
                              </div>
                        </div>
                  </div>
            </div>

            <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

List-author.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>All Authors</title>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
```

```
href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
```

```
</head>
<body>
    <div th:insert="comman/header :: header"></div>
    <div class="container my-2">

        <div th:switch="${authors}" class="container my-5">
            <p class="my-5">
                <a href="/addAuthor" class="btn btn-dark"> <i
                    class="fas fa-user-plus ml-2"> Add Authors
</i></a>&nbsp;&nbsp;<a
                    href="/export/all-author" class="btn btn-info"> <i
                    class="fas fa-file-csv ml-2"> Export CSV </i></a>
            </p>

            <div class="col-md-12">
                <h2 th:case="null">No record found !!</h2>
                <div th:case="*">
                    <table class="table table-bordered">
                        <thead>
                            <tr>
                                <th>Author Name</th>
                                <th>Description</th>
                                <th>Edit</th>
                                <th>Delete</th>
                            </tr>
                        </thead>
                        <tbody>

                            <tr th:each="author : ${authors}">
                                <td th:text="${author.name}"></td>
                                <td th:text="${author.description}"></td>
                                <td><a
th:href="@{/updateAuthor/{id}(id=${author.id})}"

                                    class="btn btn-warning"><i
class="fas fa-user-edit ml-2"></i></a></td>
                                <td><a th:href="@{/remove-
author/{id}(id=${author.id})}"
                                    class="btn btn-danger"><i class="fas
fa-user-times ml-2"></i></a></td>
                            </tr>
                        </tbody>

                    </table>
                    <div th:if="${authors.totalPages > 0}" class="pagination"
                        th:each="pageNumber : ${pageNumbers}">
                        <a th:href="@{/authors(size=${authors.size},
page=${pageNumber})}"
                            th:text=${pageNumber}
                            th:class="${pageNumber==authors.number + 1} ?
active"></a>

             <   v
             /   >
             d       </div>
             i   </div>
```

</div
>

</div>

```
<div th:insert="comman/footer :: footer">Abilasha</div>
```

```
</body>

</html>
```

List-categories.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>All Categories</title>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
        href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">

</head>
<body>
        <div th:insert="comman/header :: header"></div>
        <div class="container my-2">

                <div th:switch="${categories}" class="container my-5">
                        <p class="my-5">
                                <a href="/addCategory" class="btn btn-dark"> <i
                                        class="fas fa-user-plus ml-2"> Add Category
</i></a>&nbsp;&nbsp;<a
                                        href="/export/all-category" class="btn btn-info"> <i
                                        class="fas fa-file-csv ml-2"> Export CSV </i></a>
                        </p>

                        <div class="col-md-12">
                                <h2 th:case="null">No record found !!</h2>
                                <div th:case="*">
                                        <table class="table table-bordered">
                                                <thead>
                                                        <tr>
                                                                <th>Category Name</th>
                                                                <th>Edit</th>
                                                                <th>Delete</th>
                                                        </tr>
                                                </thead>
                                                <tbody>

                                                        <tr th:each="category : ${categories}">
                                                                <td th:text="${category.name}"></td>
                                                                <td><a
th:href="@{/updateCategory/{id}(id=${category.id})}"            f         user-times ml-

class="fas fa-user-edit ml-2"></i></a></td>                     a         2"></i></a></td>

                                                                -
category/{id}(id=${category.id})}"
```

<td><a

class="
btn btn-
warning
"><i

th:href=
"@{/re
move-

class="
btn btn-
danger"
><i
class="f
as

```
                            </tr>
                        </tbody>
                    </table>

                </div>
            </div>
        </div>
    </div>

    <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

List-books.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>All Books</title>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">

</head>
<body>
    <div th:insert="comman/header :: header"></div>
    <div class="container my-2">

        <div th:switch="${books}" class="container my-5">
            <p class="my-5">
                <a href="/add" class="btn btn-dark"> <i
                    class="fas fa-user-plus ml-2"> Add Book
</i></a>&nbsp;&nbsp;<a
                    href="/export/all-book" class="btn btn-info"> <i
                    class="fas fa-file-csv ml-2"> Export CSV </i></a>
            </p>

            <div class="col-md-12">
                <h2 th:case="null">No record found !!</h2>
                <div th:case="*">

                    <table class="table table-bordered">
                        <thead>
                            <tr>

                                                                                    </tr>
```

```
<th>ISBN</th>
<th>Book Name</th>
<th>Serial Name</th>
<th>Description</th>
<th>Detail</th>
<th>Edit</th>
<th>Delete</th>
```

```
                        </thead>
                        <tbody>
                            <tr th:each="book : ${books}">
                                <td th:text="${book.isbn}"></td>
                                <td th:text="${book.name}"></td>
                                <td th:text="${book.serialName}"></td>
                                <td th:text="${book.description}"></td>
                                <td><a
th:href="@{/book/{id}(id=${book.id})}"

                                    class="btn btn-info"><i class="fas
fa-marker ml-2"></i></a></td>
                                <td><a
th:href="@{/update/{id}(id=${book.id})}"
                                    class="btn btn-warning"><i
class="fas fa-user-edit ml-2"></i></a></td>

                                <td><a th:href="@{/remove-
book/{id}(id=${book.id})}"
                                    class="btn btn-danger"><i class="fas
fa-user-times ml-2"></i></a></td>
                            </tr>
                        </tbody>
                    </table>

                    <div th:if="${books.totalPages > 0}" class="pagination"
                        th:each="pageNumber : ${pageNumbers}">
                        <a th:href="@{/books(size=${books.size},
page=${pageNumber})}"
                            th:text=${pageNumber}
                            th:class="${pageNumber==books.number + 1} ?
active"></a>

                    </div>
                </div>
            </div>
        </div>
    </div>

    <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

List-publishers.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>All Publishers</title>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
```

```
href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
```

```
</head>
```

```
<body>
    <div th:insert="comman/header :: header"></div>
    <div class="container my-2">

        <div th:switch="${publishers}" class="container my-5">
            <p class="my-5">
                <a href="/addPublisher" class="btn btn-dark"> <i
                    class="fas fa-user-plus ml-2"> Add Publisher </i></a>
&nbsp;&nbsp;<a
                    href="/export/all-publisher" class="btn btn-info"> <i
                    class="fas fa-file-csv ml-2"> Export CSV </i></a>
            </p>

            <div class="col-md-12">
                <h2 th:case="null">No record found !!</h2>
                <div th:case="*">
                    <table class="table table-bordered">
                        <thead>
                            <tr>
                                <th>Publisher Name</th>
                                <th>Edit</th>
                                <th>Delete</th>
                            </tr>
                        </thead>
                        <tbody>

                            <tr th:each="publisher : ${publishers}">
                                <td th:text="${publisher.name}"></td>
                                <td><a

    th:href="@{/updatePublisher/{id}(id=${publisher.id})}"
                                    class="btn btn-warning"><i
class="fas fa-user-edit ml-2"></i></a></td>
                                <td><a
                                    th:href="@{/remove-
publisher/{id}(id=${publisher.id})}"
                                    class="btn btn-danger"><i class="fas

fa-user-times ml-2"></i></a></td>
                            </tr>
                        </tbody>
                    </table>

                </div>
            </div>
        </div>
    </div>


    <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

Add-category.html:

```
<!DOCTYPE html>
```

```
<html xmlns:th="http://www.thymeleaf.org">
```

```
<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Add Category</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
    <div th:insert="comman/header :: header"></div>
    <div class="container my-5">
        <h3>Add Category</h3>
        <div class="card">
            <div class="card-body">
                <div class="col-md-10">
                    <form action="#" th:action="@{/add-category}"
                        th:object="${category}" method="post">
                        <div class="row">

                            <div class="form-group col-md-8">
                                <label for="name" class="col-form-label">Category Name</label> <input
                                    type="text" th:field="*{name}" class="form-control" id="name"
                                    placeholder="Category Name" />

                            </div>

                            <div  class="col-md-6">
                                <input type="submit" class="btn btn-success" value=" Submit ">

                            </div>

                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

    <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

Login.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
<meta charset="utf-8">
```

```
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Login</title>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
    <div th:insert="comman/header :: header"></div>
    <br>
    <br>
    <br>
    <br>
    <div class="container">
        <div class="row">
            <div class="col-md-6 col-md-offset-3">

                <h1>Admin Login Page</h1>
                <form th:action="@{/login}" method="post">

                    <!-- error message -->
                    <div th:if="${param.error}">
                        <div class="alert alert-danger">Invalid username or
                            password.</div>
                    </div>

                    <!-- logout message -->
                    <div th:if="${param.logout}">
                        <div class="alert alert-info">You have been logged
out.</div>

                    </div>

                    <div class="form-group">
                        <label for="username"> Username </label> : <input
type="text"

                            class="form-control" id="username"
name="username"

                            placeholder="Enter Email ID"
autofocus="autofocus">

                    </div>

                    <div class="form-group">
                        <label for="password">Password</label>: <input
type="password"

                            id="password" name="password" class="form-
control"

                            placeholder="Enter Password" />

                    </div>


                    id="login-submit"
```

```
<div class="form-group"><div class="row">
<div class="col-sm-6 col-sm-offset-3">
<input type="submit" name="login-submit"
```

```
                                    class="form-control btn btn-success"
value="Log In" />
                            </div>
                        </div>
                    </div>
                </form>


            </div>
        </div>
    </div>

    <div th:insert="comman/footer :: footer"></div>
</body>
</html>
```

Add-book.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Add Book</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
    <div th:insert="comman/header :: header"></div>
    <div class="container my-5">
        <h3>Add Book</h3>
        <div class="card">
            <div class="card-body">
                <div class="col-md-10">
                    <form action="#" th:action="@{/add-book}" th:object="${book}"
                        method="post">
                        <div class="row">

                            <div class="form-group col-md-8">
                                <label for="isbn" class="col-form-
label">ISBN</label> <input
                                    type="text" th:field="*{isbn}"
class="form-control" id="isbn"
                                    placeholder="ISBN" /> <span
                                    th:if="${#fields.hasErrors('isbn')}"
th:errors="*{isbn}"></span>

                            </div>

                            label">Book Name</label> <input
```

```
<div class="form-group col-md-
8">
        <label

                for="nam

                e"

                class="co

                l-form-

                type="te

                xt"

                th:field="

                *{name}

                "
```

```
class="form-control" id="name"
                placeholder="Book Name" />
</div>

<div class="form-group col-md-8">
        <label for="serialName" class="col-form-
label">SerialName
        </label> <input type="text"
th:field="*{serialName}"
                class="form-control"
id="serialName" placeholder="Serial Name" />

</div>

<div class="form-group col-md-8">
        <label for="description" class="col-form-
label">Description</label>

        <input type="text" th:field="*{description}"
                class="form-control"
id="description" placeholder="Description" />

</div>


<div class="form-group col-md-8">
        <label for="category" class="col-form-
label">Category</label> <select
                th:field="*{categories}"
class="form-control">
                <option th:each="category:
${categories}"
                        th:value="${category.id}"
th:text="${category.name}"></option>
        </select>

</div>

<div class="form-group col-md-8">
        <label for="author" class="col-form-
label">Author</label> <select
                th:field="*{authors}" class="form-
control">
                <option th:each="author: ${authors}"
th:value="${author.id}"

        th:text="${author.name}"></option>
        </select>
</div>


${publishers}"


label">Publisher</label>
```

```
class="form-control">
```

"

```
<div class="form-group
col-md-8">
        <label
        for="publisher"
        class="col-form-

        <select
        th:field="*{publis
        hers}"

                <option

                        th:

                        ea

                        ch

                        ="

                        pu

                        bli

                        sh

                        er:

                        th:

                        va

                        lu

                        e=

                        "$

                        {p

                        ub

                        lis

                        he

                        r.i

                        d}
```

```
th:text="${publisher.name}"></option>
                                                                </select>
                                                        </div>


                                                        <div class="col-md-6">
                                                                <input type="submit" class="btn btn-
success" value=" Submit ">

                                                        </div>

                                                </div>
                                        </form>
                                </div>
                        </div>
                </div>
        </div>

        <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

Add-publisher.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
<meta charset="utf-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Add Publisher</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
        href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
        href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
</head>

<body>
        <div th:insert="comman/header :: header"></div>
        <div class="container my-5">
                <h3>Add Publisher</h3>
                <div class="card">
                        <div class="card-body">
                                <div class="col-md-10">
                                        <form action="#" th:action="@{/add-publisher}"
                                                th:object="${publisher}" method="post">
                                                <div class="row">

                                                        Name" />


label">Publisher Name</label>

class="form-control"
```

```
<div class="form-group col-md-8">
        <label for="name" class="col-form-

        <input

                type="text" th:field="*{name}" id="name"

placeholder="Publisher

</div>
```

```
                            <div  class="col-md-6">
                                <input type="submit" class="btn btn-success" value=" Submit ">

                            </div>

                        </div>
                    </form>
                </div>
            </div>
        </div>
    </div>

    <div th:insert="comman/footer :: footer"></div>
</body>

</html>\
```

List-book.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>All Books</title>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">

</head>
<body>
    <div th:insert="comman/header :: header"></div>
    <div class="container my-2">
        <div class="card">
            <div class="card-body">
                <div class="container my-5">
                    <div  class="col-md-12">
                        <div>
                            <table class="table table-striped table-responsive-md">
                                <thead>
                                    <tr>







                                    </tr>
```

```
<tbody>     <th>ISBN</th>
            <th>Category Name</th>
            <th>Book Name</th>
            <th>Author Name</th>
            <th>Publisher Name</th>
            <th>Serial Name</th>
            <th>Description</th>
```

```
                                    <tr>
                                        <td
th:text="${book.isbn}"></td>
                                        <td
th:text="${book.categories[0].name}"></td>
                                        <td
th:text="${book.name}"></td>
                                        <td
th:text="${book.authors[0].name}"></td>
                                        <td
th:text="${book.publishers[0].name}"></td>
                                        <td
th:text="${book.serialName}"></td>
                                        <td
th:text="${book.description}"></td>
                                    </tr>
                                </tbody>
                            </table>
                        </div>
                    </div>

                </div>
            </div>
        </div>
    </div>

    <div th:insert="comman/footer :: footer"></div>
</body>

</html>
```

Footer.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Footer</title>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">

</head>
<body>



    <!-- Footer -->
    <footer class="page-footer font-small blue pt-4">
        <!-- Copyright -->
        <div class="footer-copyright text-center py-3">
            Done by R Abilasha
        </div>
```

<!-- Copyright -->

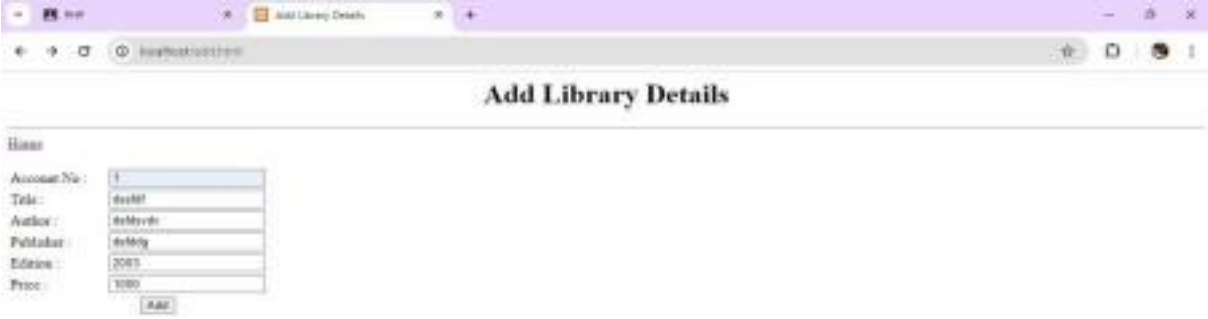
```
    </footer>
    <!-- Footer -->
</body>
</html>
```

Header.html:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="UTF-8">
<meta http-equiv="x-ua-compatible" content="ie=edge">
<title>Header</title>

<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet"
    href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
<link rel="stylesheet"
    href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">

</head>

<body>

    <nav th:fragment="header"
        class="navbar navbar-expand-sm bg-info navbar-light">

        <a class="navbar-brand" href="#">Library Management System</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
            data-target="#navbarNav">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav mr-auto">
                <li class="nav-item" sec:authorize="isAuthenticated()"><a
                    class="nav-link" href="/books">Books</a></li>
                <li class="nav-item" sec:authorize="isAuthenticated()"><a
                    class="nav-link"  href="/authors">Authors</a></li>
                <li class="nav-item" sec:authorize="isAuthenticated()"><a
                    class="nav-link" href="/publishers">Publisher</a></li>
                <li class="nav-item" sec:authorize="isAuthenticated()"><a
                    class="nav-link" href="/categories">Category</a></li>
            </ul>
            <ul class="navbar-nav">
                <li class="nav-item" sec:authorize="isAuthenticated()"><a
                    class="nav-link" th:href="@{/logout}">Logout</a></li>
         </ht

         ml>   </div>
    </nav>



</body>
```

</ul>

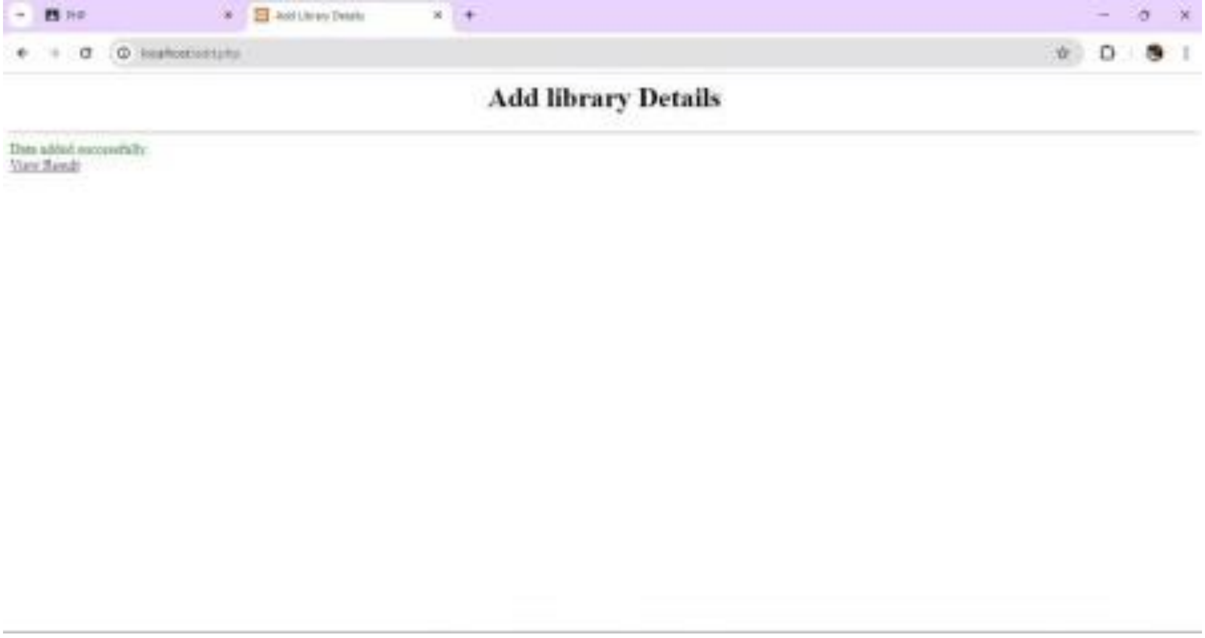
OUTPUT:



Add Library Details

Home

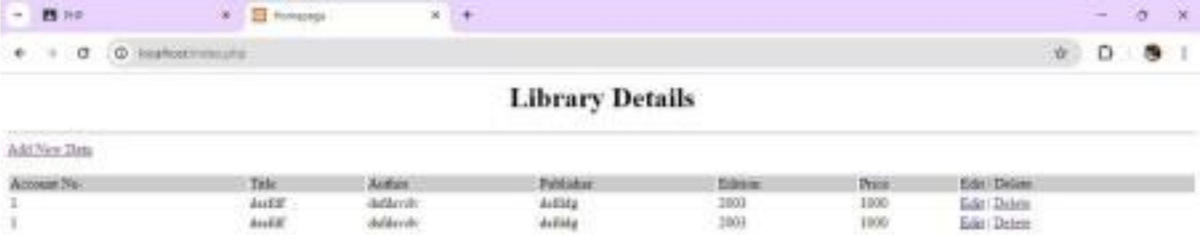
Accession No :	1
Title :	dssfdf
Author :	dsfdsvdv
Publisher :	dsfdsfg
Edition :	2003
Price :	1000



Add library Details

Data added successfully

View Result



Library Details

Add New Data

Account No.	Title	Author	Publisher	Edition	Price	Edit | Delete
1	[illegible]	[illegible]	[illegible]	2003	1000	Edit | Delete
1	[illegible]	[illegible]	[illegible]	2003	1000	Edit | Delete

RESULT:

Thus, the library management system has been created successfully.

PHP - Bank Application

Aim:

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

Procedure:

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)
```

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>
                                <td>Customer Id. :</td>
                                <td><input type="text" name="cid"></td>
                        </tr>
                        <tr>
                                <td>Customer Name :</td>
                                <td><input type="text" name="cname"></td>
                        </tr>
                        <tr>
                                <td colspan="2" align="center"><input type="submit"
                                        value="Add Customer"></td>
                        </tr>
                </table>
                <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 static
        final 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 static
    final 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"); PrintWriter
        out = 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 static
          final 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 static
        final 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>");
```

```
        }

}
```

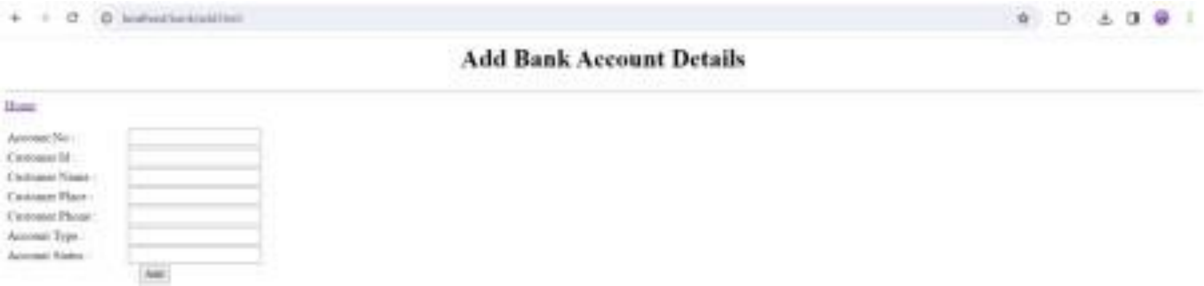
DeleteCustomer.java

```
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 static
        final 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);
                }
```

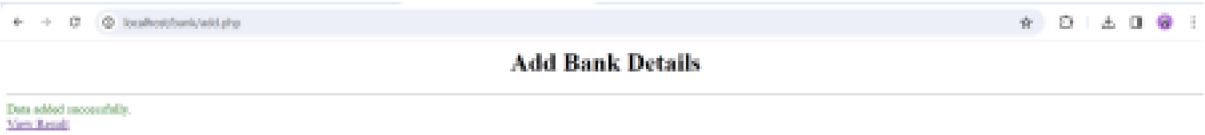
```
out.println("<br />");
```

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

OUTPUT:

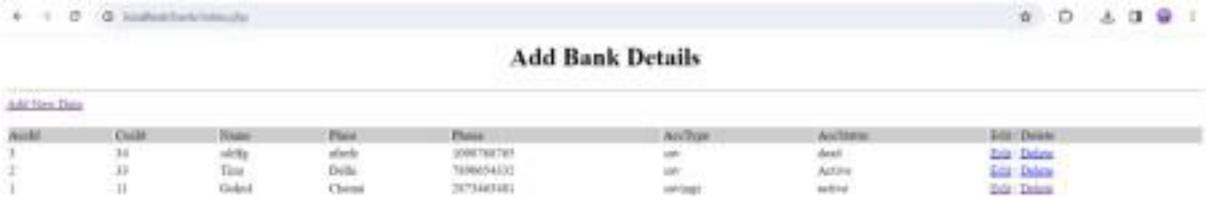


Add Bank Account Details



Add Bank Details

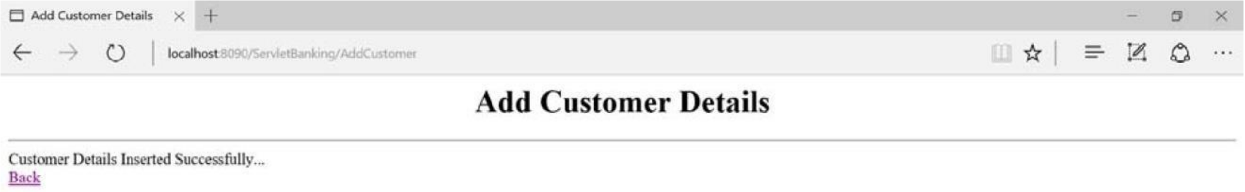
Data added successfully.



Add Bank Details

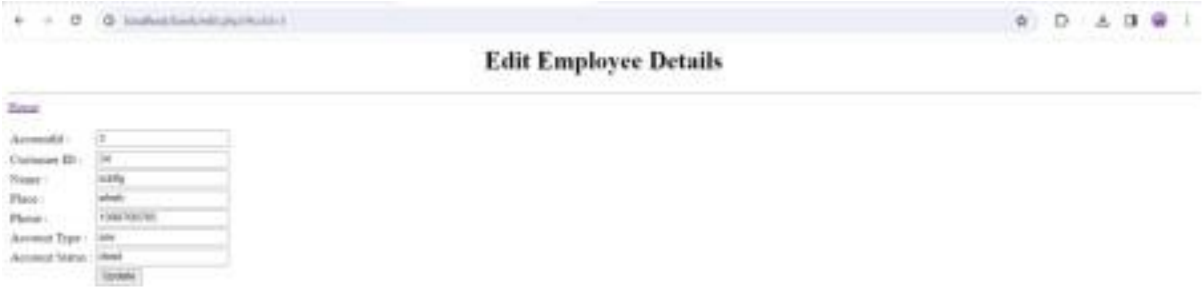
Add New Data

AccId	CusId	Name	Place	Phone	AccType	AccStatus	Edit Delete
3	34	sdcfg	afsefe	1098768785	sav	dead	Edit Delete
2	33	Tina	Delhi	7898654332	sav	Active	Edit Delete
1	11	Gokul	Chennai	2873445481	savings	active	Edit Delete



Add Customer Details

Customer Details Inserted Successfully...
Back



Edit Employee Details

Home

Accountid :	3
Customer ID :	34
Name :	sdcfg
Place :	afsefe
Phone :	1098768785
Account Type :	sav
Account Status :	dead
	Update

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>
```

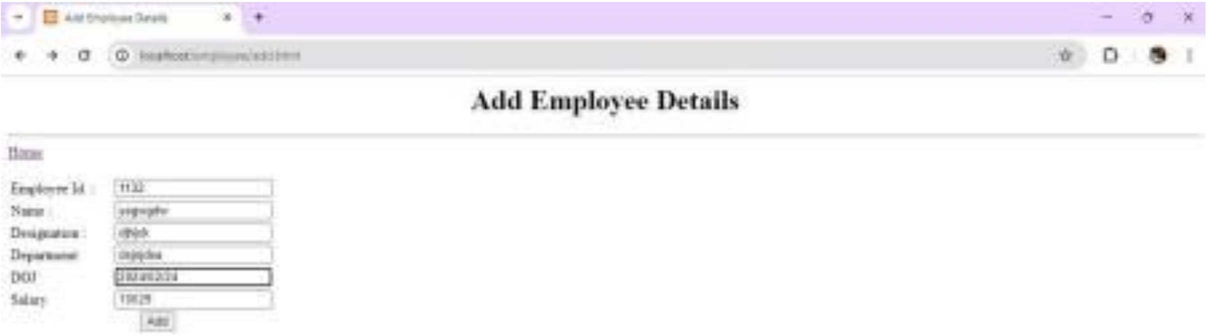
</tr>

```
        <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");
?>
```

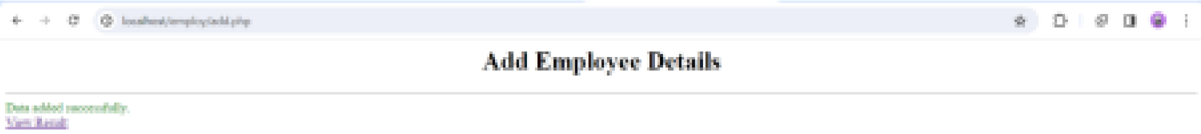
OUTPUT:



Add Employee Details

Home

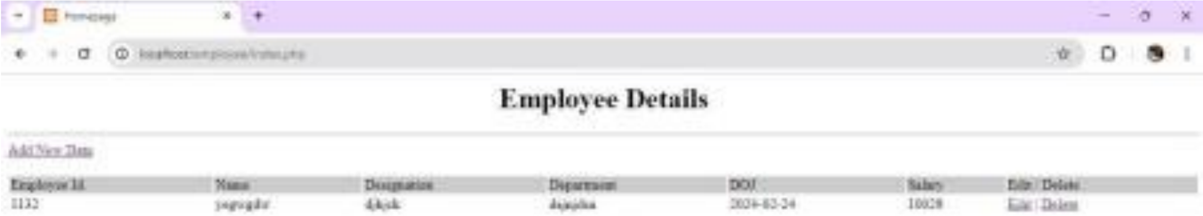
Employee Id :	1132
Name :	[illegible]
Designation :	[illegible]
Department	[illegible]
DOJ	[illegible]
Salary	[illegible]
	Add



Add Employee Details

Data added successfully.

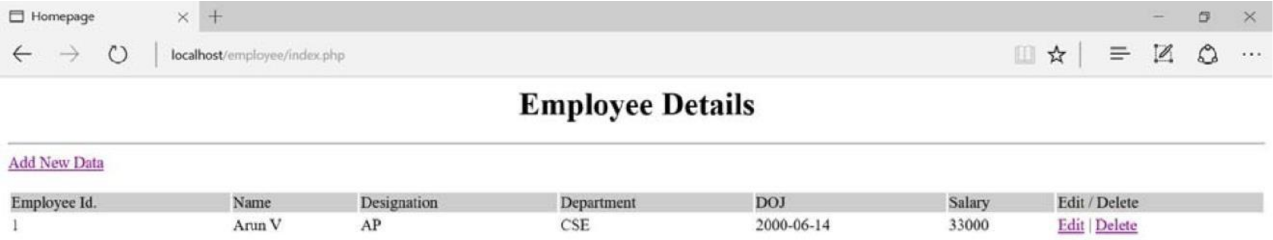
View Result



Employee Details

Add New Data

Employee Id	Name	Designation	Department	DOJ	Salary	Edit / Delete
1132	yagrgjdir	djkjsk	dsjajdsa	2024-02-24	10028	Edit / Delete



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

Result:

Thus, the PHP program for employee details has been executed Successfully.

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="Abilasha R portfolio."
       />
       <meta name="author" content="abilasha" />
       <title>Abilasha R- 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 {
```

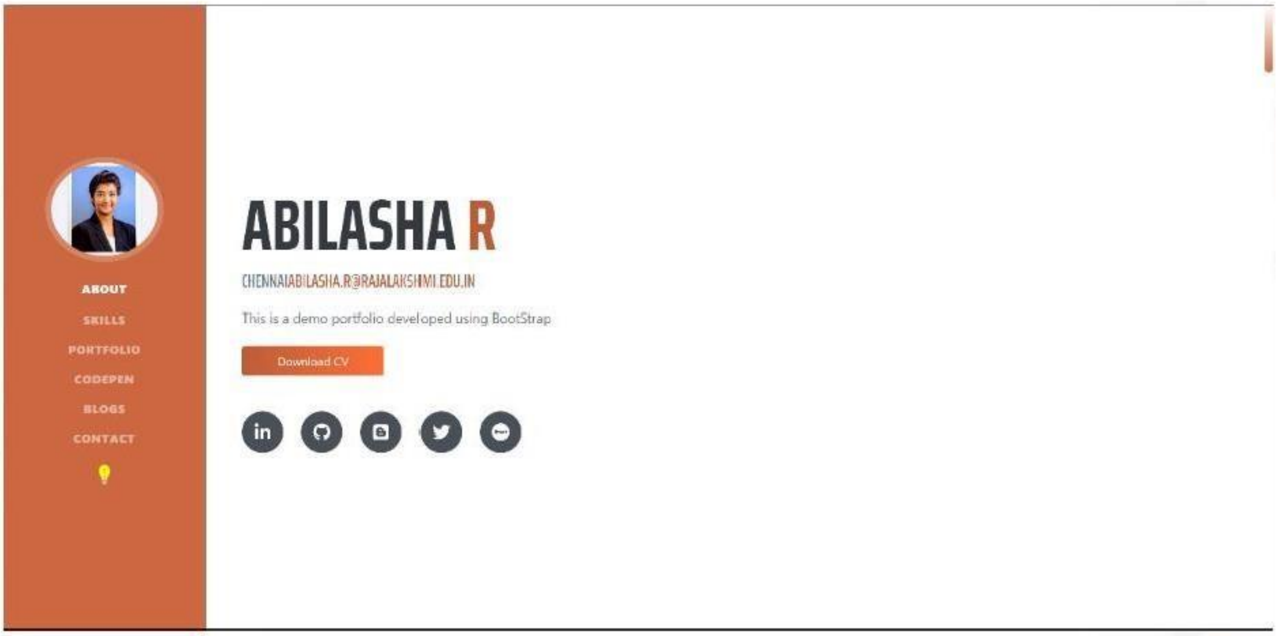
```
  margin: 0px 5px;
}
```

```
}

@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;
  }
}
```

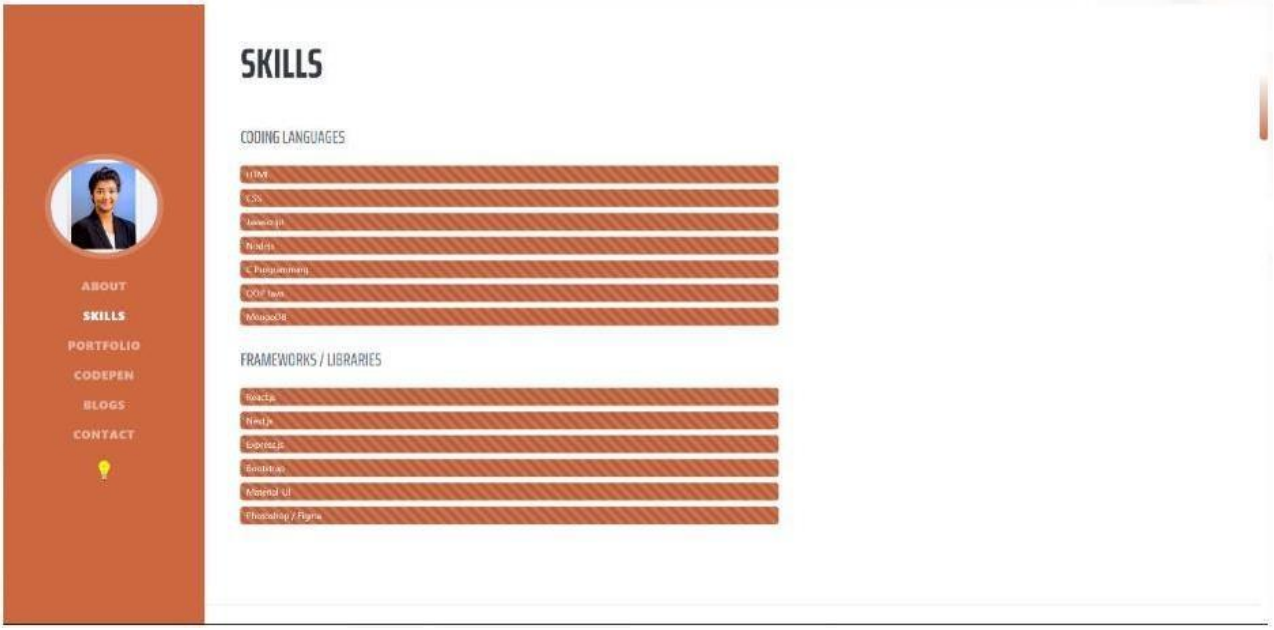
OUTPUT:



ABILASHA R

CHENNAIABILASHA.R@RAJALAKSHMI.EDU.IN

This is a demo portfolio developed using BootStrap



SKILLS

CODING LANGUAGES

HTML
CSS
JavaScript
Node.js
C Programming
OOP Java
MongoDB

FRAMEWORKS / LIBRARIES

React.js
Next.js
Express.js
Bootstrap
Material UI
Photoshop / Figma

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

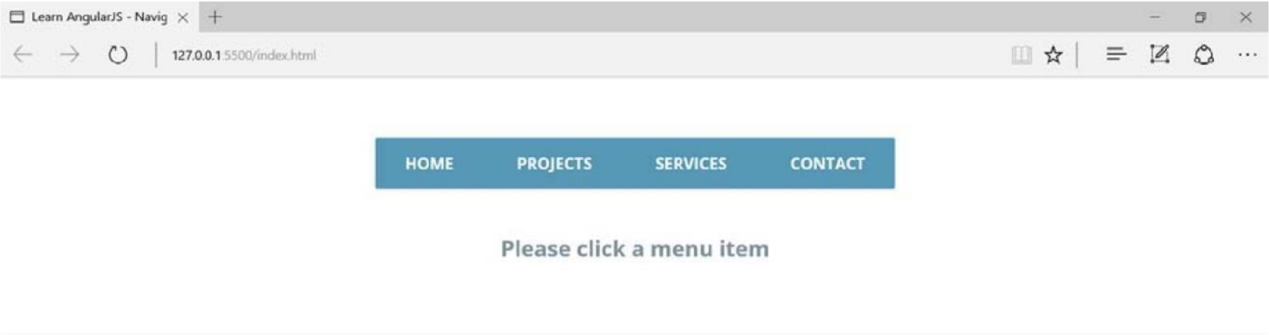
```
<!DOCTYPE html>
<html>

        <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

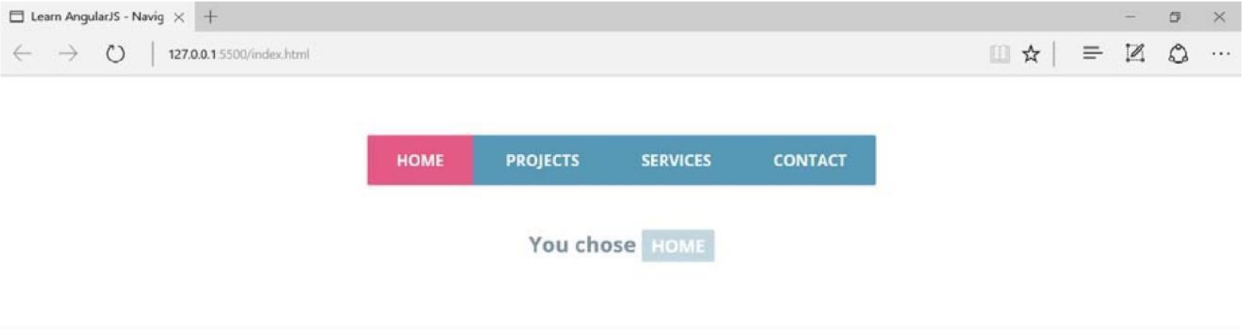
```
/* -------------------------------------------
        Simple reset
---------------------------------------------*/

*{
        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:



Please click a menu item



You chose HOME

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;
}
```

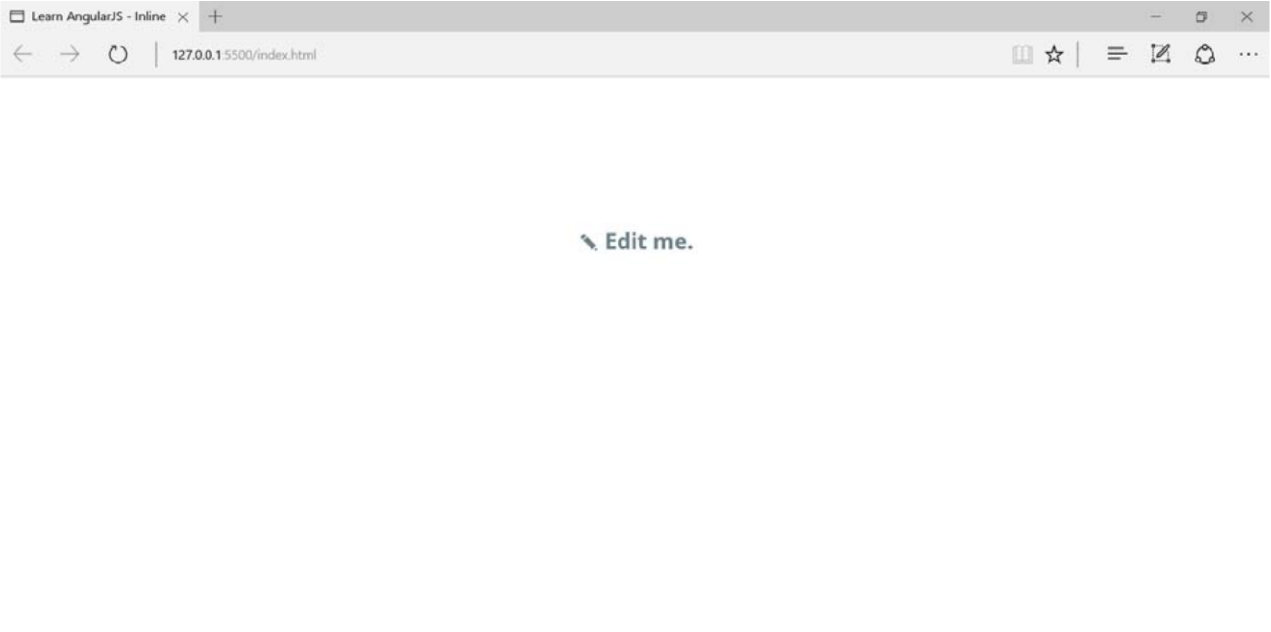
```
}
```

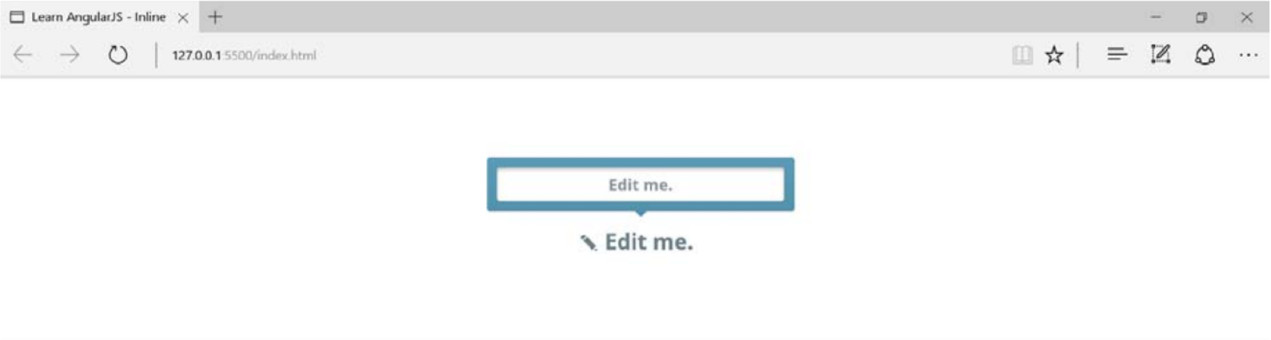
style.css

```
/* -------------------------------------------
        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;
```

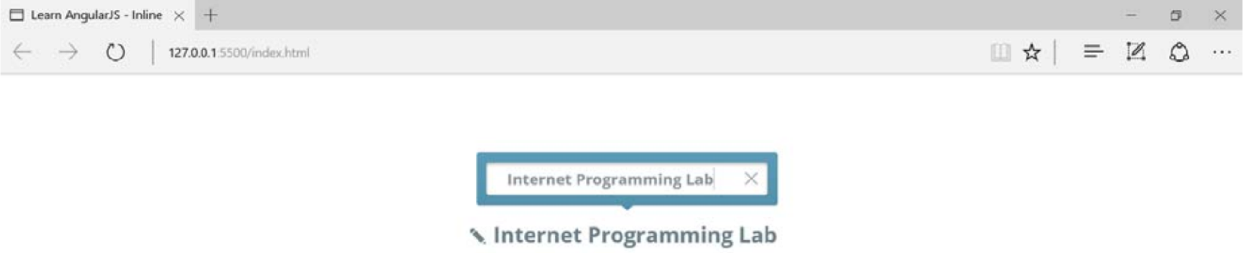
```
}
```

```
#main{
	height:300px;
	position:relative;
	padding-top: 150px;
```



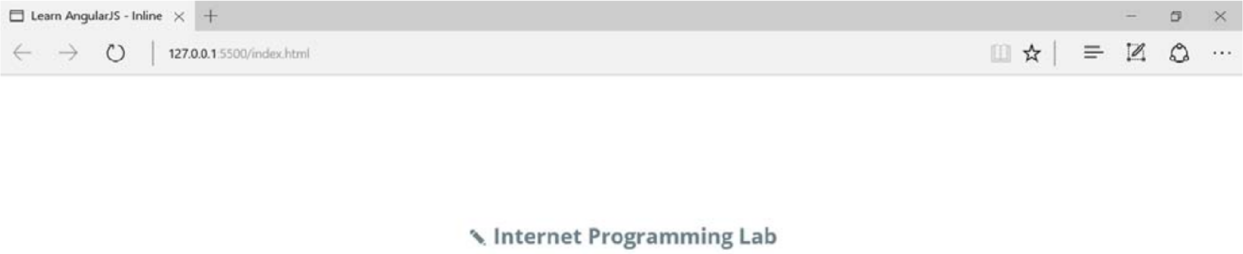


```
}
```



Internet Programming Lab

✎ Internet Programming Lab



✎ Internet Programming Lab

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

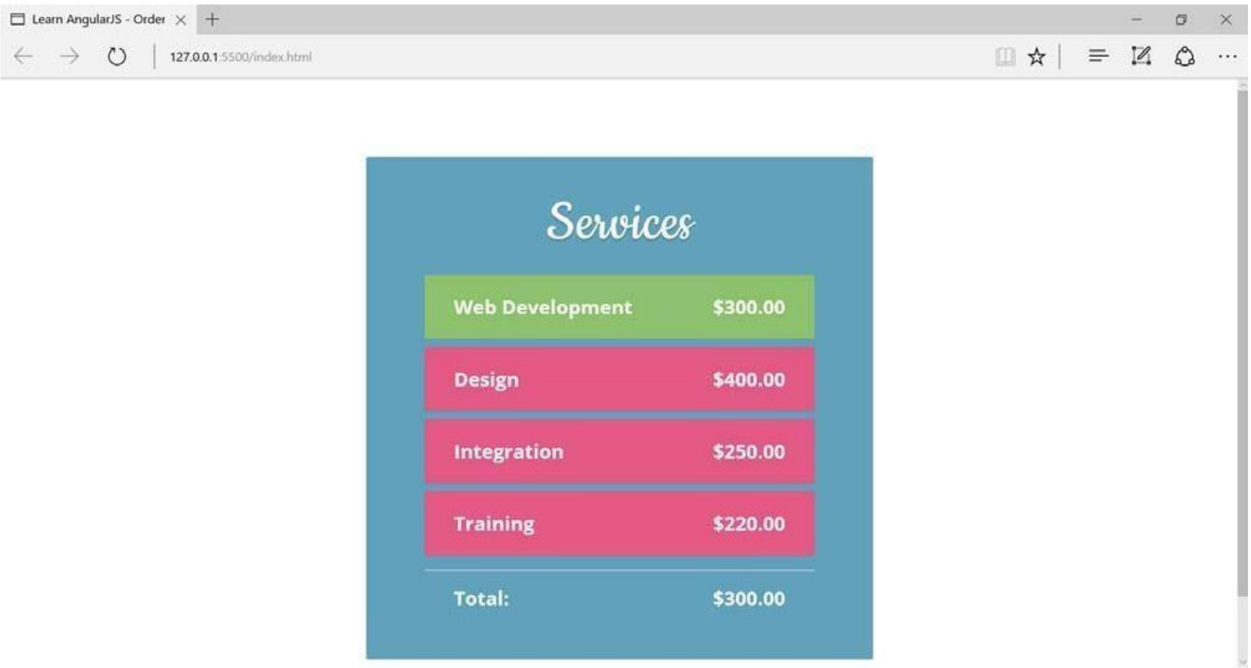
```
    /* -------------------------------------------
        Simple reset
    --------------------------------------------*/

    *{
        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:



Services

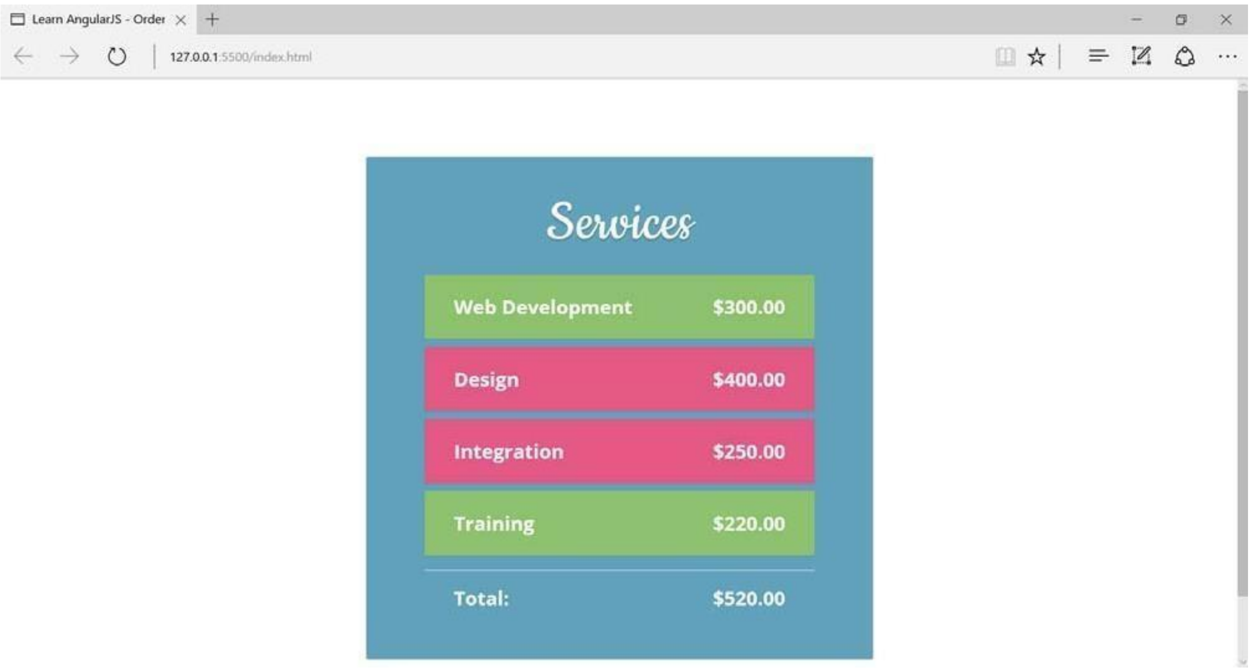
Web Development $300.00

Design $400.00

Integration $250.00

Training $220.00

Total: $300.00



Services

Web Development	$300.00
Design	$400.00
Integration	$250.00
Training	$220.00
Total:	$520.00

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>

        <ul>
            <!-- 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
-------------------------------------------*/

.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;
```

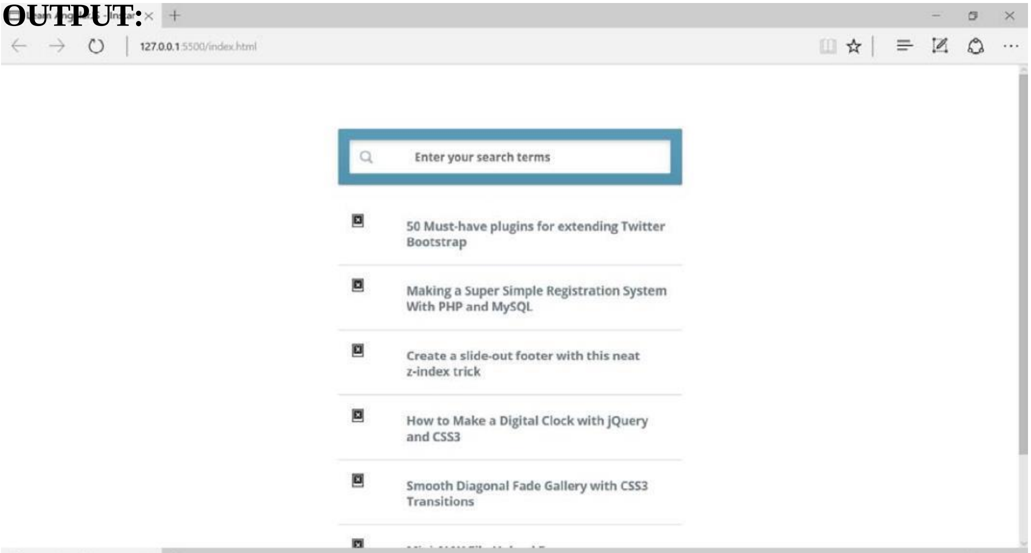
```
        width: 400px;
        padding: 14px;
        margin: 80px auto 20px;
        position:relative;
}

.bar input{
        background:#fff no-repeat 13px 13px;

        background-
image:url(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/9
hAAA
AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvbS5h
ZG9iZS54bX
AAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3pr
YzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNU
CBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgIj4gP
HJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMj
Ij4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvY
mUuY29tL
3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21tLy
IgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZVJ
lZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBXaW5kb3dzIiB4bXBNTTpJ
bnN0YW5jZU
lEPSJ4bXAuaWlkOkU5NEY0RTlFMTA4NzExRTM5RTEzQkFBQzMyRjkyQzVBIiB4bXBNTT
pEb2N1bWVud
ElEPSJ4bXAuZGlkOkU5NEY0RTlGMTA4NzExRTM5RTEzQkFBQzMyRjkyQzVBIj4gPHhtcE1
NOkRlcml2
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6RTk0RjRFOUMxMDg3MTFFMzl
FMTNCQUFDMzJ
GOTJDNUEiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6RTk0RjRFOUQxMDg3MTFF
MzlFMTNCQUFDMz
JGOTJDNUEiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmRmOlJERj4gPC94OnhtcG1ldGE+I
Dw/eHBhY
2tldCBlbmQ9InIiPz4DjA/RAAABK0lEQVR42pTSQUdEURjG8dOY0TqmPkGmRcqYD9CmzZA
WJRHVRIa0
iFYtM6uofYaiEW2SRJtEi9YxIklp07ZkWswu0v/wnByve7vm5ee8M+85zz1jbt9Os+WiGkYdYxjC
Ox5
wgFeXUHmtBSzpcCGa+5BJTCjEP+0nKWAT8xqe4ArPGEEVC1hHEbs2oBwdXkM7mj/JLZrad4
37sCGHOf
UtcziutuYu2v8XUFF/4f6vMK/YgAH1HxkBYV60AR31gxkBYd6xAeF3VzMCwvzOBpypX8V4y
uFRzX2d2
gD/l5yjH4fYQEnzkj4fae5rJulF2sMXVrAsaTWttRFu4Osb+1jEDT71/ZveyhouTch2fINQL9hKefKj
uYFfuznXWzXMTabyrvfyIV3M4vhXgAEAUMs7K0J9UJAAAAAASUVORK5CYII=);

        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;
}

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:



Enter your search terms

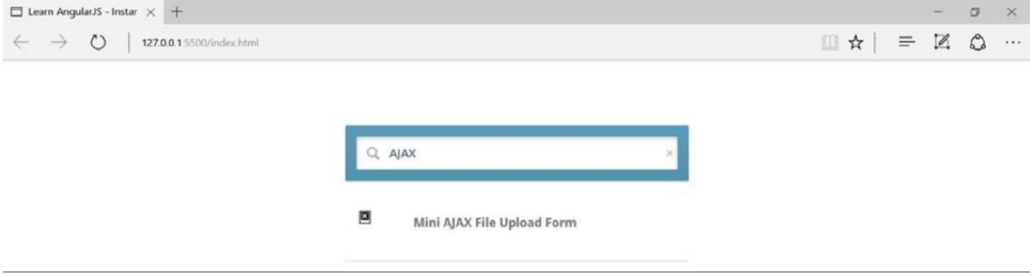
50 Must-have plugins for extending Twitter Bootstrap

Making a Super Simple Registration System With PHP and MySQL

Create a slide-out footer with this neat z-index trick

How to Make a Digital Clock with jQuery and CSS3

Smooth Diagonal Fade Gallery with CSS3 Transitions



AJAX

Mini AJAX File Upload Form

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(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf
      8/9hAAA
      AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvb
      S5hZG9iZS54bX
      AAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3
      prYzlkIj8+I
      Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFh
      NUCBDb3Jl
      IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgIj4
      gPHJkZjpSREY
      geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbn
      MjIj4gPH
      JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRv
      YmUuY29tL
      3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21t
      LyIgeG1s
      bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZ
```

211701010

VJlZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBXaW5kb3dzIiB4bXBNT
TpJbnN0YW5jZU

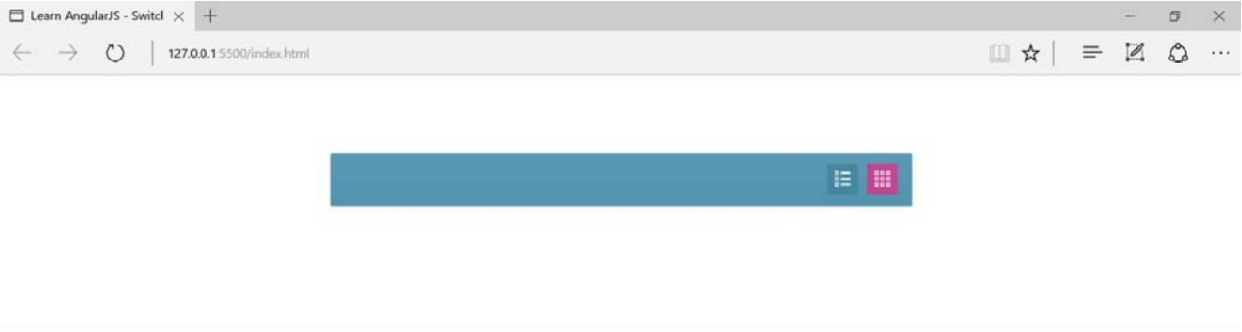
```
lEPSJ4bXAuaWlkOkYzNkFCQ0ZBMTBCRTExRTM5NDk4RDFEM0E5RkQ1NEZCIiB4bXB
NTTpEb2N1bWVud
ElEPSJ4bXAuZGlkOkYzNkFCQ0ZCMTBCRTExRTM5NDk4RDFEM0E5RkQ1NEZCIj4gPH
htcE1NOkRlcml2
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6RjM2QUJDRjgxMEJFMTFFMzk0
OThEMUQzQTl
GRDU0RkIiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6RjM2QUJDRjkxMEJFMTFFMz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

.bar a.grid-icon{
	background-
image:url(data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf
8/9hAAA
AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvb
S5hZG9iZS54bX
AAAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3
prYzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFh
NUCBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgIj4
gPHJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbn
MjIj4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRv
YmUuY29tL
3hhcC8xLjAvIiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21t
LyIgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJjZ
VJlZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBXaW5kb3dzIiB4bXBNT
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:



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>
```

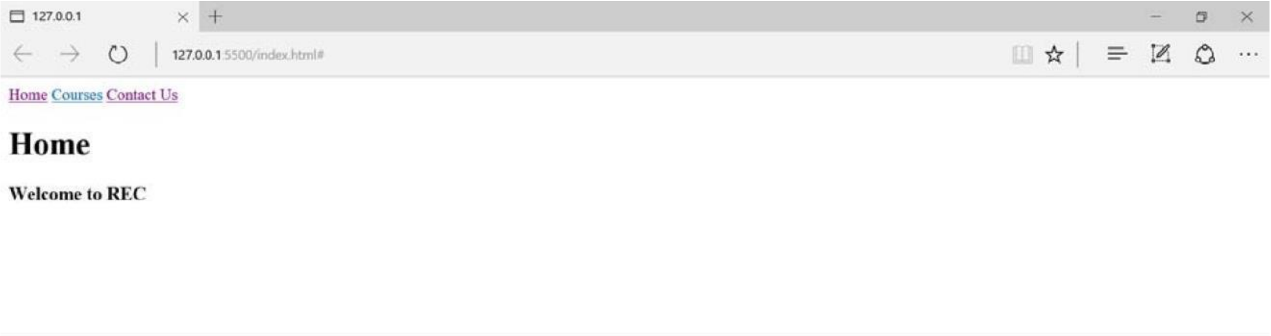
app.js

```
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';
});
```

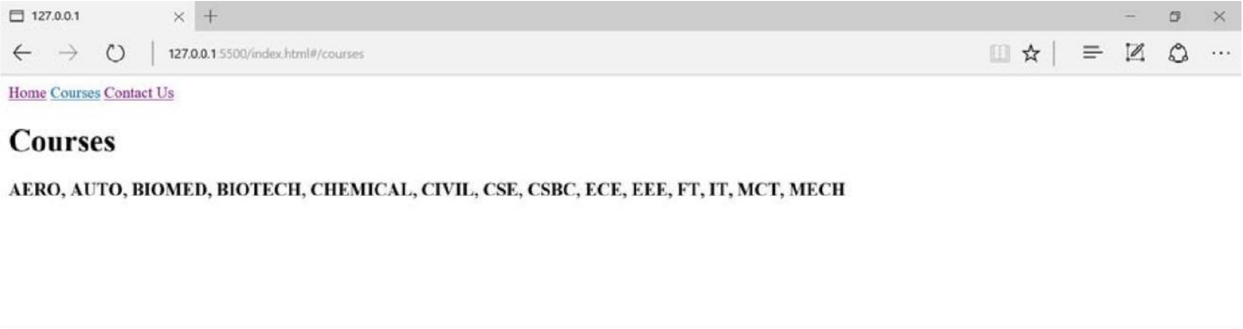
OUTPUT:



Home Courses Contact Us

Home

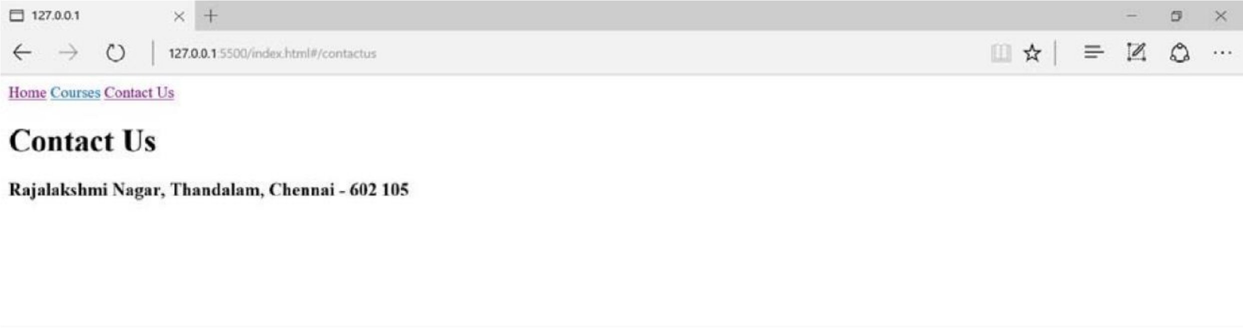
Welcome to REC



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