

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

Date:

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

```

```

<div class="column">
  
  </div>
</div>
</body>
</html>

```

OUTPUT:




TamilNaduImage x +

← → ↻ File C:/Users/Hari/OneDrive/Desktop/tkinter_project/TamilNadu.html ☆ 📄 🖨️ 🔍 ⋮

TamilNadu

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 triumvirate of the Cheras, Cholas and Pandyas, the Pallavas (3rd-9th century CE), and the later Vijayanagara Empire (14th-17th century CE).




The first image shows a wide waterfall cascading over rocks. The second image shows a large, dark statue of Lord Venkateswara on a pedestal. The third image shows a tall, ornate temple gopuram (tower) with intricate carvings.

KarnatakaImage x +

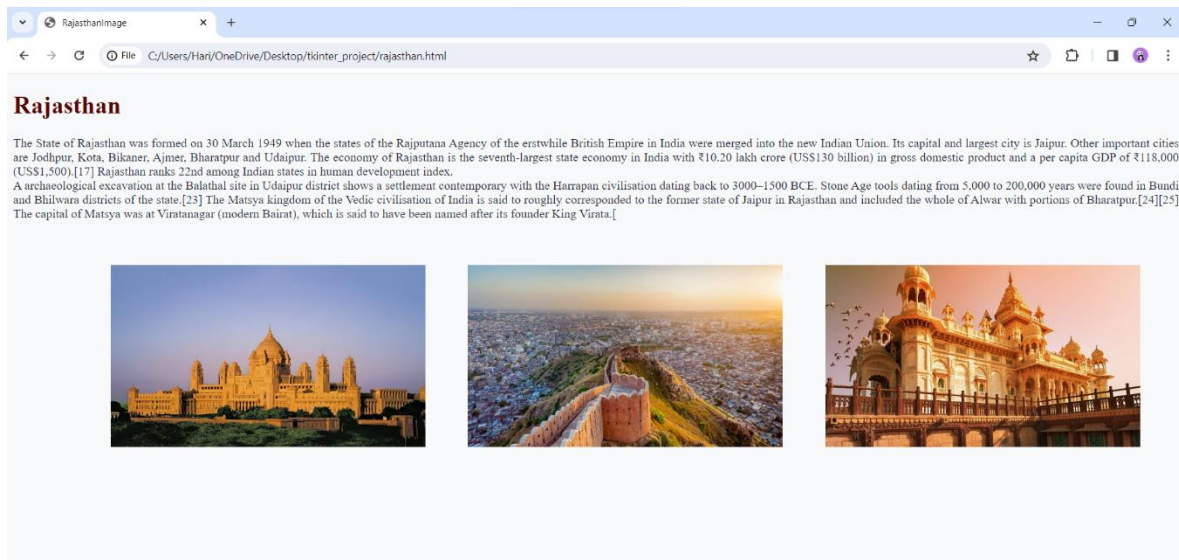
← → ↻ File C:/Users/Hari/OneDrive/Desktop/tkinter_project/karnataka.html ☆ 📄 🖨️ 🔍 ⋮

Karnataka

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 Camatic, sometimes Karnatak, to describe both sides of peninsular India, south of the Krishna. 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..



The first image shows a waterfall cascading over rocks. The second image shows a tall, white temple tower (Gopuram) with a flag on top. The third image shows ancient stone ruins, including a large archway and other structures.



RESULT:

Thus web page to embed a map has been created successfully.

Ex.NO:2 Web page creation using an embedded, external, and inline CSS file.
Date:

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;
    }
  </style>
</html>
```

```
/* 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>
    
    <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">
      
      <div class="hw">Students</div>

    </div>
    <div class="column">
      
      <div class="hw">Infrastructure</div>

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

```

```

    
    <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>
  
  <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">
    
    <div class="hw">Students</div>

  </div>
  <div class="column">
    
    <div class="hw">Infrastructure</div>

  </div>
  <div class="column">
    
    <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">
    
    <div class="hw">Students</div>

```

```

</div>
<div class="column">
  
  <div class="hw">Infrastructure</div>

</div>
<div class="column">
  
  <div class="hw">Labs and Classes</div>
</div>
</div>
</html>

```

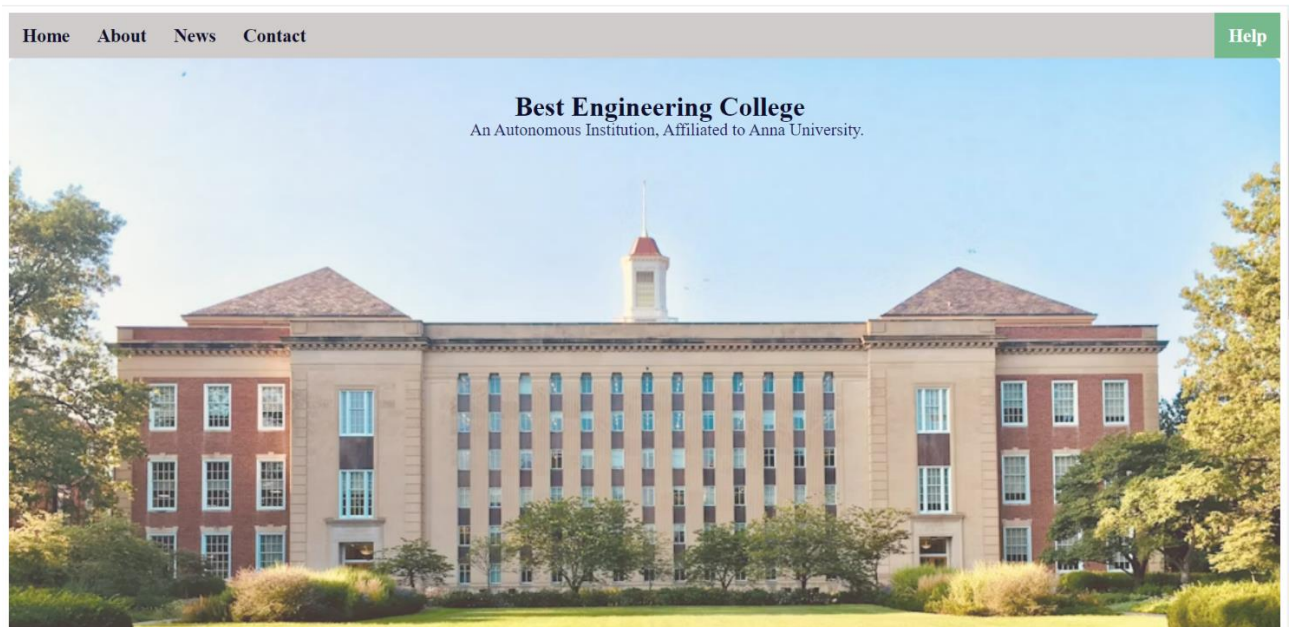
Style.css:

```

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

```

OUTPUT:



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



Labs and Classes



Infrastructure



Students

About:

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.



Labs and Classes



Infrastructure



Students

News:

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.



Labs and Classes



Infrastructure



Students

RESULT:

Thus the web page with inline, embedded and external CSS has been created successfully.

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

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">


</div>

<form class="registration-form">
<p>Registration 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>Pass Word:</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:

File | C:/Users/Hari/OneDrive/Desktop/tkinter_project/reg.html

Registration Form

First Name:
Tinuu

Last Name:
A V

Department:
CSD

Email Id:
tinuvtinu@gmail.com

PassWord:

☒ I agree to the terms and conditions.

Register

Regform x +

File | C:/Users/DELL/OneDrive/Documents/form.html

This page says
Your registration is successful
OK

Last Name:
Kumar

Department:
CSD

Email Id:
21101040@rajalekshmi.edu.in

PassWord:

☒ I agree to the terms and conditions.

Register

Type here to search

Video Game Storyb... form.html - Visual S... Regform - Google ...

11:00 13-02-2024

RESULT:

Thus the registration form has been created successfully.

Ex. No:4 Creation of Library Management System
Date:

AIM:

To create a library management system that access a database using JSP.

ALGORITHM:

- 1.Create a index.html file through which book details can be added , viewed and deleted.
- 2.Write insert.jsp code to insert items in table.
- 3.View.jsp file is created to write code to view the books in the table.
- 4.Delete.jsp file is created to write the jsp code to delete the required book from the table.
- 5.Database connection is established .
- 6.Tomcat server is installed and started
- 7.url is opened and operations are performed.

PROGRAM:

Index.html:

```
<html>
  <head>
  </head>
  <style>
    .btn,img{
      position:absolute;
    }
    .btn{

      display:inline-block;
    }
  </style>
  <body>
    <h1><center>LIBRARY MANAGEMENT</center></h1>
    
    <div class="btn">
      <button style="width: 150px;height:50px;"><a href="view.jsp">View
Books</a></button>
      <button style="width:150px;height: 50px;"><a href="library.html">Add
Books</a></button>
      <button style="width:150px;height: 50px;"><a href="delete.jsp">Delete
Books</a></button>
    </div>

  </body>
```

```
</html>
```

Library.html:

```
<!DOCTYPE html>
<html lan="en">
<head>
  <title>Book Details</title>
</head>
<style>
  body{
    background-color: rgb(217, 238, 245);
  }
  table {
    font-family: arial, sans-serif;
    border-collapse: collapse;
    table-layout:auto;
    width:fit-content;
    margin-left: auto;
    margin-right: auto;
  }

  td, th {
    border: 1px solid #dddddd;
    text-align:left;
    padding: 8px;
    font-weight:600;
    color:rgb(82, 0, 0);
  }

  tr:nth-child(even) {
    background-color: #f3f0f5;
  }
  tr:nth-child(odd) {
    background-color: #e3dce6;
  }
  h1{
    text-align: center;
    color:rgb(35, 35, 247);
  }
</style>

<body>
  <h1>Book Details</h1>
  <fieldset >
    <form name="frmContact" class="needs-validation " method="post" action="insert.jsp">
<table>
  <tr>
    <td>Account No.</td>
    <td><input type="number" name="txtAcc" id="txtAcc"></td>
  </tr>
```

```

<tr>
  <td>Book Title</td>
  <td><input type="text" name="txtTitle" id="txtTitle"></td>
</tr>
<tr>
  <td>Author</td>
  <td><input type="text" name="txtAuthor" id="txtAuthor"></td>
</tr>
<tr>
  <td>Publisher</td>
  <td><input type="text" name="txtPublisher" id="txtPublisher"></td>
</tr>
<tr>
  <td>Edition</td>
  <td><input type="number" name="txtEdition" id="txtEdition"></td>
</tr>
<tr>
  <td>Price</td>
  <td><input type="number" name="txtPrice" id="txtPrice"></td>
</tr>
</table>
<center><input type="submit" name="Submit" id="Submit" value="SUBMIT"></center>
</form>
</fieldset>
</body>
</html>

```

Insert.jsp:

```

<%@ page import="java.sql.*, javax.naming.*, javax.sql.*" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
  <title>Insert Book</title>
</head>
<body>
  <script>
    function showAlert(message) {
      alert(message);
    }
  </script>
<h2>Insert Book</h2>
<%
  // Retrieve form parameters
  String account = request.getParameter("txtAcc");
  String title = request.getParameter("txtTitle");
  String author = request.getParameter("txtAuthor");
  String publisher = request.getParameter("txtPublisher");
  String edition = request.getParameter("txtEdition");

```

```

String price = request.getParameter("txtPrice");
out.println("account: "+account);
out.println("bnmae: "+title);
// Check if all parameters are not null and not empty
if (account != null && !account.isEmpty() &&
    title != null && !title.isEmpty() &&
    author != null && !author.isEmpty() &&
    publisher!= null && !publisher.isEmpty() &&
    edition != null && !edition.isEmpty() &&
    price != null && !price.isEmpty()) {
    try {

        // Get database connection
        Context initContext = new InitialContext();
        Context envContext = (Context)initContext.lookup("java:/comp/env");
        DataSource ds = (DataSource)envContext.lookup("jdbc/libdb");
        Connection conn = ds.getConnection();

        // Prepare SQL statement to insert book
        String sql = "INSERT INTO book
(fldAccount,fldTitle,fldAuthor,fldPublisher,fldEdition,fldPrice) VALUES (?, ?, ?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, Integer.parseInt(account));
        pstmt.setString(2, title);
        pstmt.setString(3, author);
        pstmt.setString(4,publisher);
        pstmt.setInt(5, Integer.parseInt(edition));
        pstmt.setInt(6, Integer.parseInt(price));

        // Execute the insert statement
        int rowsAffected = pstmt.executeUpdate();

        // Close resources
        pstmt.close();
        conn.close();

        out.println("<script>showAlert('Book inserted successfully');</script>");

    } catch (Exception e) {
        e.printStackTrace();
        out.println("<script>showAlert('Error inserting book');</script>");
    }
} else {
    out.println("<script>showAlert('One or more parameters are missing');</script>");
}
%>
</body>
</html>

```

View.jsp:

```

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

<!DOCTYPE html>
<html>
<head>
    <title>Books Information</title>
</head>
<body>

<h2>Books Information</h2>

<table border="1">
    <tr>
        <th>Account</th>
        <th>Title</th>
        <th>Author</th>
        <th>Publisher</th>
        <th>Edition</th>
        <th>Price</th>
    </tr>
<%
try {
    // Get database connection
    Context initContext = new InitialContext();
    Context envContext = (Context)initContext.lookup("java:/comp/env");
    DataSource ds = (DataSource)envContext.lookup("jdbc/libdb");
    Connection conn = ds.getConnection();

    // Create SQL statement
    Statement stmt = conn.createStatement();
    String sql = "SELECT fldAccount,fldTitle,fldAuthor,fldPublisher,fldEdition,fldPrice
FROM book";
    ResultSet rs = stmt.executeQuery(sql);

    // Iterate through the result set and display records
    while (rs.next()) {
%>
        <tr>
            <td><%= rs.getString("fldAccount") %></td>
            <td><%= rs.getString("fldTitle") %></td>
            <td><%= rs.getString("fldAuthor") %></td>
            <td><%= rs.getString("fldPublisher") %></td>
            <td><%= rs.getString("fldEdition") %></td>
            <td><%= rs.getString("fldPrice") %></td>
        </tr>
<%

```



```

    }
    rs.close();
    stmt.close();
    conn.close();
} catch (Exception e) {
    e.printStackTrace();
}
%>
</table>
</body>
</html>

```

Delete.jsp:

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ page import="java.sql.*" %>

<%
    // Database connection parameters
    String url = "jdbc:mysql://localhost:3306/db_library"; // Replace with your database URL
    String username = "root"; // Replace with your database username
    String password = ""; // Replace with your database password

    // Get the book ID and account ID from the request parameter
    String bookIdToDelete = request.getParameter("fldAccount");

    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        // Establish the database connection
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url, root, "");

        // Prepare the SQL statement to delete the book
        String sql = "DELETE FROM book WHERE fldAccount=?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, bookIdToDelete);

        // Execute the delete operation
        int rowsAffected = pstmt.executeUpdate();

        if (rowsAffected > 0) {
            // Book successfully deleted

        } else {
            // Book not found or deletion failed
        }

        out.println("Book deleted successfully.");
    } catch (Exception e) {
        e.printStackTrace();
    }
%>

```

```

        out.println("Failed to delete book.");
    }
} catch (Exception e) {
    // Handle any exceptions
    e.printStackTrace();
    out.println("An error occurred: " + e.getMessage());
} finally {
    // Close resources
    try {
        if (pstmt != null) pstmt.close();
        if (conn != null) conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
}
%>

```

OUTPUT:



Book Details

Account No.

Book Title

Author

Publisher

Edition

Price

SUBMIT

localhost:8080/libraryManage/insert.jsp

localhost:8080 says
Book inserted successfully

OK

localhost:8080/libraryManage/insert.jsp

localhost:8080 says
Book Deleted successfully

OK

Books Information

Account	Title	Author	Publisher	Edition	Price
23	dsgsfq	dgfdhg	hvjghg	2023	400
23	dsgsfq	dgfdhg	hvjghg	2023	400
23	dstd	sdsd	dax	2010	211
21	asa	asdx	dax	2019	180
21	asa	asdx	dax	2019	180
43	JHD	.kjh	.kjh	9876	988

RESULT:

Thus the library management system has been created successfully.

Ex.No:5**PHP program for Bank Details****Date:****AIM:**

To write a PHP program for Bank Details, which include customer id, customer name, customer place, customer phone, account type, account status to connect with the database and execute queries to retrieve and update data.

ALGORITHM:

- 1.Create a config file to establish database connection.
- 2.Add.html file is created to add employee details.
- 3.Edit.php, delete.php , index.php, add.php files are created.
- 4.These files are created add writtend code for their respective operations.
- 5.Apache server is connected and database table is created to store the employee details.
- 6.Open the url link and perform the operations.

PROGRAM:**Config.php:**

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

Add.html:

```
<html>
<head>
<title>Add Bank Details</title>
</head>
<body>
<h1 align="center">Add Bank Account 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>Account.No : </td>
<td><input type="text" name="AccId"></td>
</tr>
<tr>
<td>Customer Id : </td>
<td><input type="text" name="CusId"></td>
</tr>
<tr>
<td>Customer Name : </td>
<td><input type="text" name="Name"></td>
</tr>
<tr>
<td>Customer Place : </td>
<td><input type="text" name="Place"></td>
</tr>
<tr>
<td>Customer Phone : </td>
<td><input type="text" name="Phone"></td>
</tr>
<tr>
<td>Account Type : </td>
<td><input type="text" name="AccType"></td>
</tr>
<tr>
<td>Account Status : </td>
<td><input type="text" name="AccStatus"></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit"
name="Submit" value="Add"></td>
```

```

</tr>
</table>
</form>
</body>
</html>

```

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 tbl_bank ORDER BY AccId DESC");
?>
<html>
<head>
<title>Homepage</title>
</head>
<body>
<h1 align="center">Add Bank Details</h1>
<hr />
<a href="add.html">Add New Data</a><br/><br/>
<table width='100%' border=0>
<tr bgcolor='#CCCCCC'>
<td>AccId</td>
<td>CusId</td>
<td>Name</td>
<td>Place</td>
<td>Phone</td>
<td>AccType</td>
<td>AccStatus</td>
<td>Edit / Delete</td>
</tr>
<?php
while($res = mysqli_fetch_array($result)) {

```

```

echo "<tr>";
echo "<td>".$res['AccId']."</td>";
echo "<td>".$res['CusId']."</td>";
echo "<td>".$res['Name']."</td>";
echo "<td>".$res['Place']."</td>";
echo "<td>".$res['Phone']."</td>";
echo "<td>".$res['AccType']."</td>";
echo "<td>".$res['AccStatus']."</td>";
echo "<td><a href='edit.php?AccId=$res[AccId]'>Edit</a>";
echo " | <a href='delete.php?AccId=$res[AccId]'>Delete</a></td>";
echo "</tr>";
}
?>
</table>
</body>
</html>

```

Edit.php:

```

<?php
// including the database connection file
include_once("config.php");
if(isset($_POST['update']))
{
    $AccId = $_POST['AccId'];
    $CusId = $_POST['CusId'];
    $Name = $_POST['Name'];
    $Place = $_POST['Place'];
    $Phone = $_POST['Phone'];
    $AccType = $_POST['AccType'];
    $AccStatus = $_POST['AccStatus'];
//updating the table
$result = mysqli_query($mysqli, "UPDATE tbl_bank SET AccId='$AccId',
CusId='$CusId',Name='$Name',Place='$Place',Phone=$Phone WHERE
AccId=$AccId");

```



```

//redirectig to the display page. In our case, it is index.php
header("Location: index.php");
}
?>
<?php
echo "<h1 align='center'>Edit Employee Details</h1>";
echo "<hr />";
//getting id from url
$AccId = $_GET['AccId'];
//selecting data associated with this particular eid
$result = mysqli_query($mysqli, "SELECT * FROM tbl_bank WHERE AccId=$AccId");
while($res = mysqli_fetch_array($result))
{
    $AccId = $res['AccId'];
    $CusId = $res['CusId'];
    $Name = $res['Name'];
    $Place = $res['Place'];
    $Phone = $res['Phone'];
    $AccType = $res['AccType'];
    $AccStatus = $res['AccStatus'];
}
?>
<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>AccountId : </td>

```

```
<td><input type="text" name="AccId" value="<?php echo $AccId;?>"></td>
</tr>
<tr>
<td>
<td>Customer ID : </td>
<td><input type="text" name="CusId" value="<?php echo $CusId;?>"></td>
</tr>
<tr>
<td>
<td>Name : </td>
<td><input type="text" name="Name" value="<?php echo $Name;?>"></td>
</tr>
<tr>
<td>
<td>Place : </td>
<td><input type="text" name="Place" value="<?php echo $Place;?>"></td>
</tr>
<tr>
<td>
<td>Phone : </td>
<td><input type="text" name="Phone" value="<?php echo $Phone;?>"></td>
</tr>
<tr>
<td>
<td>Account Type : </td>
<td><input type="text" name="AccType" value="<?php echo $AccType;?>"></td>
</tr>
<tr>
<td>
<td>Account Status : </td>
<td><input type="text" name="AccStatus" value="<?php echo $AccStatus;?>"></td>
</tr>
<tr>
<td><input type="hidden" name="AccId" value="<?php echo $_GET['AccId'];?>"></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
```

```
$AccId = $_GET['AccId'];
```

```
//deleting the row from table
```

```
$result = mysqli_query($mysqli, "DELETE FROM tbl_bank WHERE AccId=$AccId");
```

```
//redirecting to the display page (index.php in our case)
```

```
header("Location:index.php");
```

```
?>
```

OUTPUT:

The image shows two screenshots of a web browser. The first screenshot displays a form titled "Add Bank Account Details" at the URL "localhost/bank/add.html". The form includes input fields for "Account.No :", "Customer Id :", "Customer Name :", "Customer Place :", "Customer Phone :", "Account Type :", and "Account Status :", followed by an "Add" button. The second screenshot shows the same browser at "localhost/bank/add.php", displaying the title "Add Bank Details" and a green message "Data added successfully." with a link to "View Result".

211701004

← → ↻ 🌐 localhost/bank/index.php ☆ 📄 📥 📱 📶 ⋮

Add Bank Details

[Add New Data](#)

AccId	CusId	Name	Place	Phone	AccType	AccStatus	Edit / Delete
3	34	sddfg	afrefc	1098768765	sav	dead	Edit Delete
2	33	Tinu	Delhi	7890654332	sav	Active	Edit Delete
1	11	Gokul	Chennai	2873465481	savings	active	Edit Delete

← → ↻ 🌐 localhost/bank/edit.php?AccId=3 ☆ 📄 📥 📱 📶 ⋮

Edit Employee Details

[Home](#)

AccountId :

Customer ID :

Name :

Place :

Phone :

Account Type :

Account Status :

RESULT:

Thus PHP program for Bank Account details have been created successfully.

Ex.No:6**PHP program for Employee Details****Date:****AIM:**

To 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.

ALGORITHM:

- 1.Create a config file to establish database connection.
- 2.Add.html file is created to add employee details.
- 3.Edit.php, delete.php , index.php, add.php files are created.
- 4.These files are created add writtend code for their respective operations.
- 5.Apache server is connected and database table is created to store the employee details.
- 6.Open the url link and perform the operations.

PROGRAM:**Config.php:**

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

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>

```

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 tbl_employ 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>".$res['eDesig']."</td>";
echo "<td>".$res['eDept']."</td>";
echo "<td>".$res['eDoj']."</td>";
echo "<td>".$res['eSalary']."</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>

```

Edit.php:

```

<?php
// including the database connection file
include_once("config.php");
if(isset($_POST['update']))
{
$empid = $_POST['empid'];
$name = $_POST['ename'];
$desig = $_POST['desig'];
$dept = $_POST['dept'];
$doj = $_POST['doj'];
$salary = $_POST['salary'];
//updating the table
$result = mysqli_query($mysqli, "UPDATE tbl_employ SET eName='$name',
eDesig='$desig',eDept='$dept',eDoj='$doj',eSalary=$salary WHERE
empId=$empid");

//redirectig to the display page. In our case, it is index.php
header("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 tbl_employ WHERE
empId=$empid");
while($res = mysqli_fetch_array($result))
{
$empid = $res['empId'];
$name = $res['eName'];
$desig = $res['eDesig'];
$dept = $res['eDept'];
$doj = $res['eDoj'];
$salary = $res['eSalary'];
}
?>
<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 $name;?>"></td>
</tr>
<tr>
<td>
<td>
<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>
<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 tbl_employ WHERE empId=$empid");
//redirecting to the display page (index.php in our case)
header("Location:index.php");
?>
```

Add.php:

```
<html>
<head>
<title>Add Employee Details</title>
</head>
<body>
<?php
//including the database connection file
include_once("config.php");
$empid = $_POST['empid'];
$name = $_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 tbl_employ values ($empid,

$name', $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>
```

OUTPUT:

localhost/employ/add.html

Add Employee Details

[Home](#)

Employee Id. :

Name :

Designation :

Department :

DOJ :

Salary :

localhost/employ/add.php

Add Employee Details

Data added successfully.

[View Result](#)

localhost/employ/add.php

Employee Details

[Add New Data](#)

Employee Id.	Name	Designation	Department	DOJ	Salary	Edit / Delete
1	Varsh	tester	csd	2003-05-06	50000	Edit Delete

localhost/employ/add.php

Employee Details

[Add New Data](#)

Employee Id.	Name	Designation	Department	DOJ	Salary	Edit / Delete
1	Varsh	tester	csd	2003-05-06	50000	Edit Delete

RESULT:

Thus PHP program for employee details have been created successfully.

Ex. No:6

BOOTSTRAP – WEB PAGE

Date:

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 page to 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);
  }

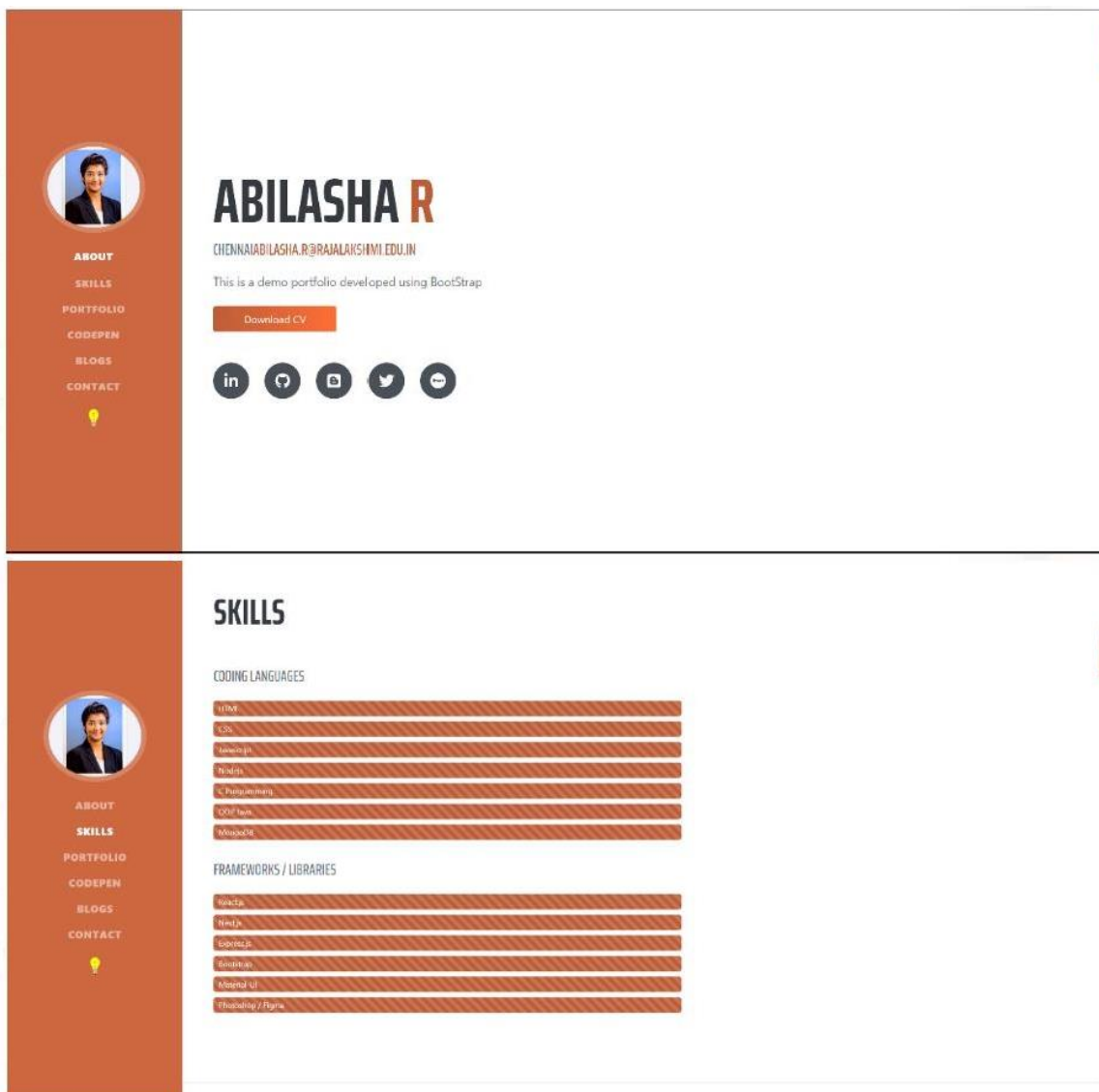
  #sideNav .navbar-collapse {
    display: flex;
    align-items: flex-start;
    flex-grow: 0;
    width: 100%;
  }
```

```

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:



Result:

Thus website using bootstrap has been created successfully.

Ex.No:7

Design a Web page with - Navigation Menu

Date:

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 next example).
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 should be 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 when a 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 be visible. -->

    <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;
}
```

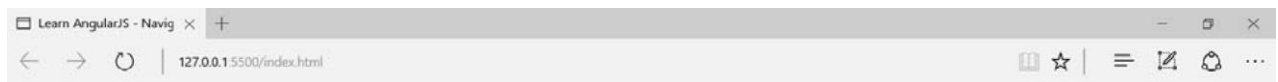
```
nav a:last-child{
    border-radius:0 2px 2px 0;
}
```

```
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 the value 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 on the page that depend on it. -->
```

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

```
    <!-- Call a method defined in the InlineEditorController that toggles the  
    showtooltip variable -->  
    <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 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: '\2192';
    display: inline-block;
    margin-right: 5px; font-
    weight: normal;
    vertical-align: text-bottom;
}

```

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



 Edit me.



Edit me.

 Edit me.

```
}
```



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 loop through 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 will format 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 generate 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;
}
```

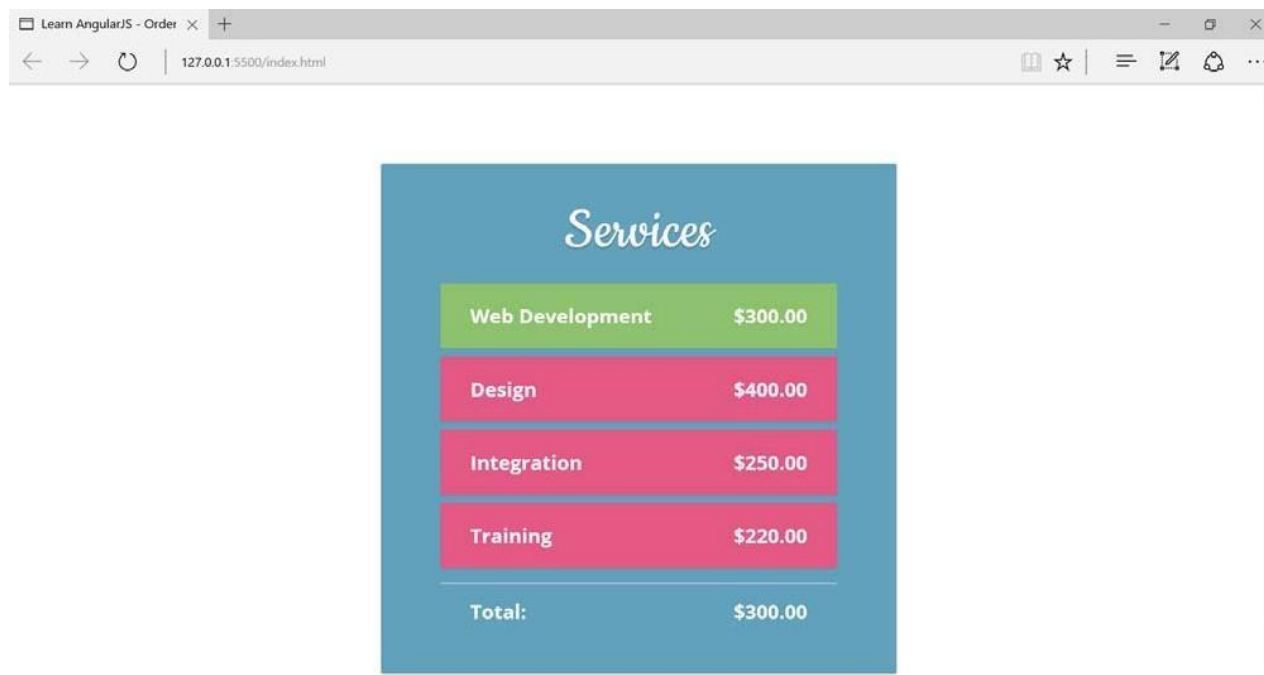
```
form ul li.active{
    background-color:#8ec16d;
}
```

```
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:



<i>Services</i>	
Web Development	\$300.00
Design	\$400.00
Integration	\$250.00
Training	\$220.00
<hr/>	
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 components that can be combined in new and interesting ways.
4. Angular relies on this technique for code isolation and requires that your application follows 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 the text 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. Notice the 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} }"></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 MySQL
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
AAAAAADw/eHBhY2tldCBiZWdpbj0i77u/liBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3pr
YzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFhNU
CBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgIj4gP
HJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMj
Ij4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYm
uUuY29tL
3hhcC8xLjAvLiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21tLy
IgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJlZ3VJ
lZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBxZW5kb3dzLiB4bXBNTTpJ
bnN0YW5jZU
IEPSJ4bXAuaWlkOkU5NEY0RTIFMTA4NzExRTM5RTEzQkFBQzMyRjkyQzVBliB4bXBNTT
pEb2N1bWVud
EIEPSJ4bXAuZGlkOkU5NEY0RTIGMTA4NzExRTM5RTEzQkFBQzMyRjkyQzVBliB4bXBNTT
NOKRlcm12
ZWRGcm9tIHNOUmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6RTk0RjRFOUMxMDg3MTFFMzI
FMTNCQUFDMzJ
GOTJDNUeIiHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6RTk0RjRFOUQxMDg3MTFF
MzIIFMTNCQUFDMz
JGOTJDNUeIiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmlpZD0iQWRvYmUgUGhvdG9zaG9w
IENTNSBxZW5kb3dzLiB4bXBNTTpJbnN0YW5jZU
2tldCBiZWdpbj0i77u/liBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+I
Dw/eHBhY2tldCBiZWdpbj0i77u/liBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkIj8+I
iFYtM6uofYaiEW2SRJtEi9YxIklp07ZkWsWu0v/wNByve7vm5ee8M+85zz1jbt9Os+WiGkYdYxjC
Ox5
wgFeXUHmtBSzpcCGa+5BJTCjEP+0nKWAT8xqe4ArPGEEVC1hHEbs2oBwdXkM7mj/JLZrad4
37sCGHOF
UtcziutuYu2v8XUFF/4f6vMK/YgAH1HxkBYV60AR31gxkBYd6xAeF3VzMCwvzOBpypX8V4y
uFRzX2d2
gD/15yjH4fYQEnzkj4fae5rJulF2sMXVrAsaTWtRFu4Osb+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

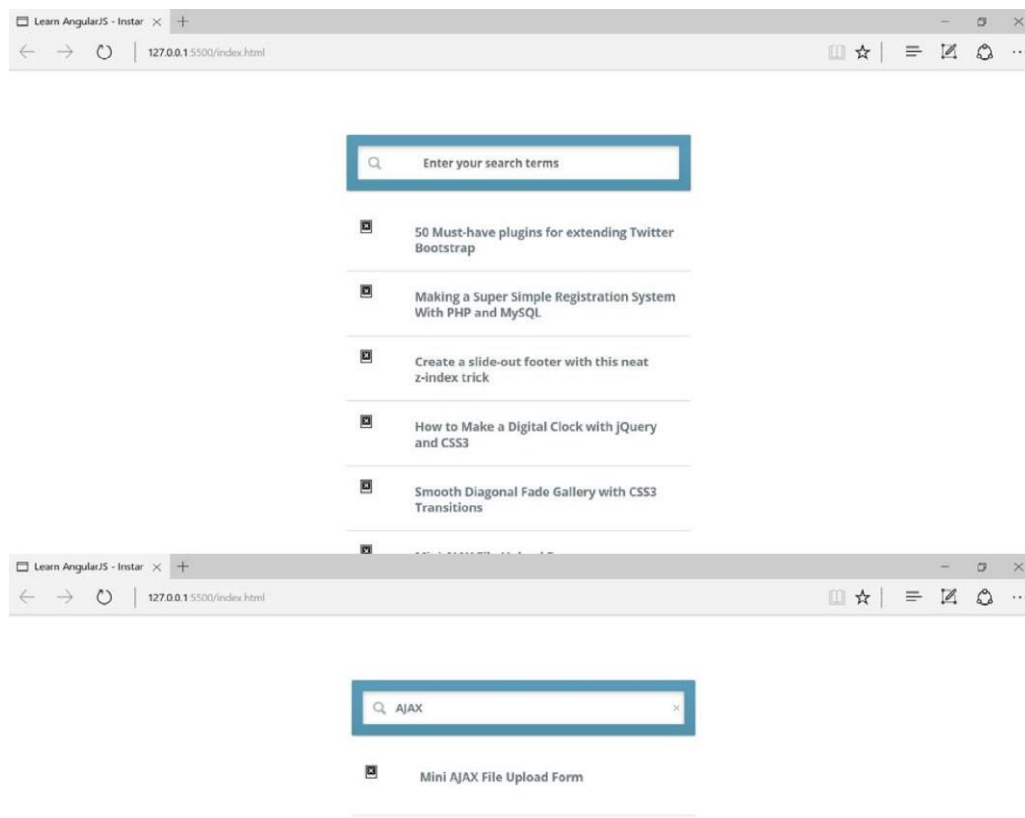
}

```
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:



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 the most 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 variable, 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 on the
    "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">
          </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">
          </a>
        <p>{{ p.caption.text }}</p>
      </li>
    </ul>
  </body>
</html>
```

```
        <!-- 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=:client_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,iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAAAAAf
8/9hAAA
AGXRFWHRTb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvb
S5hZG9iZS54bX
AAAAAADw/eHBhY2tldCBiZWdpbj0i77u/LiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3
prYzklJ8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWVRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFh
NUCBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgICAg
gPHJkZjpSREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbn
MjJ4gPH
JkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRv
YmUuY29tL
3hhcC8xLjAvLiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLnNvbS94YXAvMS4wL21t
LyIgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJzZ
VJlZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBxW5kb3dzLiB4bXBNT
TpJbnN0YXZ5JZU
```

IEPSJ4bXAuaWlkOkYzNkFCQ0ZBMTBCRTExRTM5NDk4RDFEM0E5RkQ1NEZCiiB4bXB
NTTpEb2N1bWVud
ELEPSJ4bXAuZGlkOkYzNkFCQ0ZCMTBCRTExRTM5NDk4RDFEM0E5RkQ1NEZCIj4gPH
htcE1NOkRlcm12
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6RjM2QUJDRjgxeMEJFMtFFMzk0
OThEMUQzQTl
GRDU0RkIiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6RjM2QUJDRjkxeMEJFMtFFMz
k0OThEMUQzQT
IGRDU0RkIiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmlRmOlJERj4gPC94OnhtcG1ldGE+ID
w/eHBhY
2tldCB1bmQ9InIiPz7h1bLqAAAAWU1EQVR42mL8////BwYGBn4GCACxBRlIAIxAA/4jaXoPEk
MyJ+A
/g9MDJQBRhYg8RFqMwg8RJIUINYLFDMBUi+ADQAF1n8ofk9yIAy6WPg4GgtDMRYAAgw
AdLYwLAoIwPgAAAAASUVORK5CYII=);
}

.bar a.grid-icon{
background-
image:url(data:image/png;base64,iVBORw0KGgoAAAANSUUhEUgAAABAAAAAQCAyAAAAf
8/9hAAA
AGXRFWHRtb2Z0d2FyZQBBZG9iZSBJbWFnZVJlYWR5ccllPAAAAyBpVFh0WE1MOmNvb
S5hZG9iZS54bX
AAAAAADw/eHBhY2tldCBiZWdpbj0i77u/IiBpZD0iVzVNME1wQ2VoaUh6cmVTek5UY3
prYzlkIj8+I
Dx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iIHg6eG1wdGs9IkFkb2JlIFh
NUCBDb3Jl
IDUuMC1jMDYwIDYxLjEzNDc3NywgMjAxMC8wMi8xMi0xNzozMjowMCAgICAgICAgICAgICAg
gPHJkZjpsREY
geG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5Lz4yLzIyLXJkZi1zeW50YXgtbn
Mjlk4gPH
JkZjpsREYXNjcmlwdGlviByZGY6YWJvdXQ9IiIgeG1sbnM6eG1wPSJodHRwOi8vb3NlYWRv
YmUuY29tL
3hhcC8xLjAvLiB4bWxuczp4bXBNTT0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wL21t
LyIgeG1s
bnM6c3RSZWY9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC9zVHlwZS9SZXNvdXJkZ
VJlZiMiIHh
tcDpDcmVhdG9yVG9vbD0iQWRvYmUgUGhvdG9zaG9wIENTNSBxW5kb3dzLiB4bXBNT
TpJbnN0YXV5ZU
IEPSJ4bXAuaWlkOjBEQkMyQzE0MTBCRjExRTNBMDIGRTYyOTIBNDdCN0I4IiB4bXBNT
TTpEb2N1bWVud
ELEPSJ4bXAuZGlkOjBEQkMyQzE1MTBCRjExRTNBMDIGRTYyOTIBNDdCN0I4Ij4gPHhtc
E1NOkRlcm12
ZWRGcm9tIHN0UmVmOmluc3RhbmNlSUQ9InhtcC5paWQ6MERCQzJDMTIxMEJGMtFF
M0EwOUZFNjI5OU
0N0I3QjgiIHN0UmVmOmRvY3VtZW50SUQ9InhtcC5kaWQ6MERCQzJDMTMxMEJGMtFF
FM0EwOUZFNjI5OU
E0N0I3QjgiLz4gPC9yZGY6RGVzY3JpcHRpb24+IDwvcmlRmOlJERj4gPC94OnhtcG1ldGE+I
Dw/eHBhY
2tldCB1bmQ9InIiPz4MjPshAAAAXklEQVR42mL4////h/8I8B6IGaCYKHFGEMnAwCDIAAHv
gZgRyiZK
nImBQsACxB+hNoDAQyQ5osQZIT4gH1DsBZABH6AB8x/JaQzEig++WPiI7Rxio/GwmCIB
YAAAwAwVIzMp1R0aQAAAABJRU5ErkJggg==);
}

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

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

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

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

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

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

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

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

```
----- /* Grid layout
----- */
```

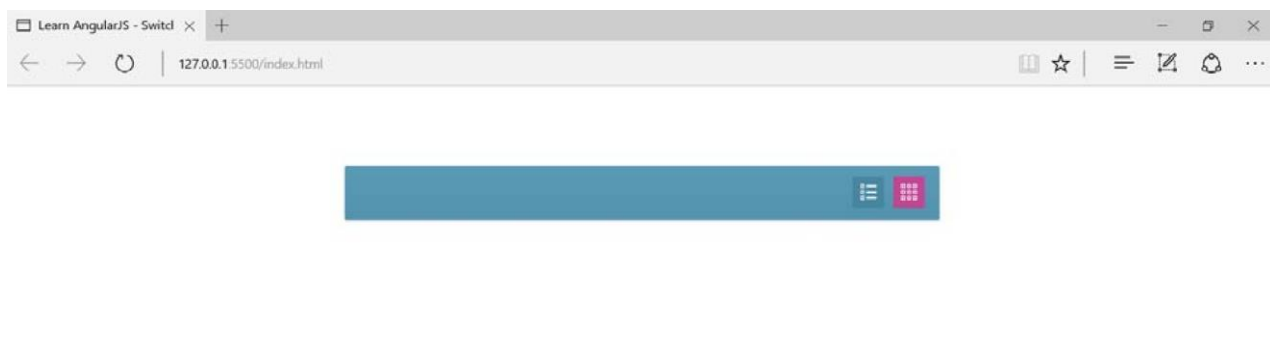
```
ul.grid{
    list-style: none;
    width: 570px;
```

```
        margin: 0 auto;
        text-align: left;
    }

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

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

OUTPUT:



Result:

Thus Web page with Navigation menu, Inline editor, Order form, Instant Search & Switchable Grid has been created successfully.

Ex.No:8

Single Page Application

Date:

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'
    })
  })
```

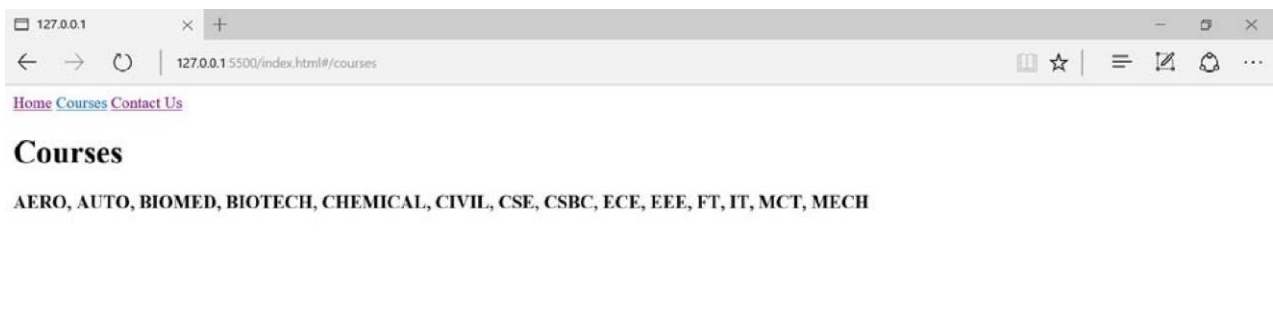
```
.otherwise({redirectTo: '/'});
});

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:





Result:

Thus single page application has been created successfully