

RAJALAKSHMI ENGINEERING COLLEGE
[AUTONOMOUS]
RAJALAKSHMI NAGAR, THANDALAM - 602105



Laboratory Record Note Book

Name :

Year / Branch / Section :

Register No :

College Roll No :

Semester :

Academic Year :

RAJALAKSHMI ENGINEERING COLLEGE
RAJALAKSHMI NAGAR, THANDALAM - 602105

BONAFIDE CERTIFICATE

Name : _____

Academic Year : _____ Semester : _____ Branch : _____

Register No :

Certified that is the bonafide record of work done by the

above student in the _____

Laboratory during the year 20 - 20

Signature of Faculty in-charge

Submitted for the practical examination held on _____

Internal Examiner

External Examiner

INDEX

Reg. No. :

Name : _____

Year : _____

Branch : _____

Sec : _____

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

EXPT NO: Create a web page to embed a map along with hot spot AND links.

DATE:

AIM:

To create a web page which includes a map and display the related information when a hot spot is clicked in the map.

PROCEDURE:

1. Create a html file with map tag.
2. Set the source attribute of the img tag to the location of the image and also set the use map attribute.
3. Specify an area with name, shape and href set to the appropriate values.
4. Repeat step 3 as many hot spots you want to put in the map.
5. Create html files for each and every hot spot the user will select.

Code:

ImageMap.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0"> <title>Interactive India Map</title>
<style>
#india-map {
width: 100%;
max-width: 1000px;
margin: 20px auto;
position: relative;
font-family: 'Arial', sans-serif;
}
```

```
#kerala-link {
position: absolute;
top: 80%;
left: 40%;
transform: translate(-50%, -50%);
font-size: 18px;
color: #ffffff;
background-color: #4caf50;
padding: 10px;
border-radius: 5px;
text-decoration: none;
font-weight: bold;
z-index: 2;
}

</style>

</head>

<body>

<div id="india-map">



<a href="https://en.wikipedia.org/wiki/Kerala" id="kerala-
link">kerala</a> <map name="indiaMap">
<area shape="rect" coords="218,790,800,952"
href="https://en.wikipedia.org/wiki/Kerala">

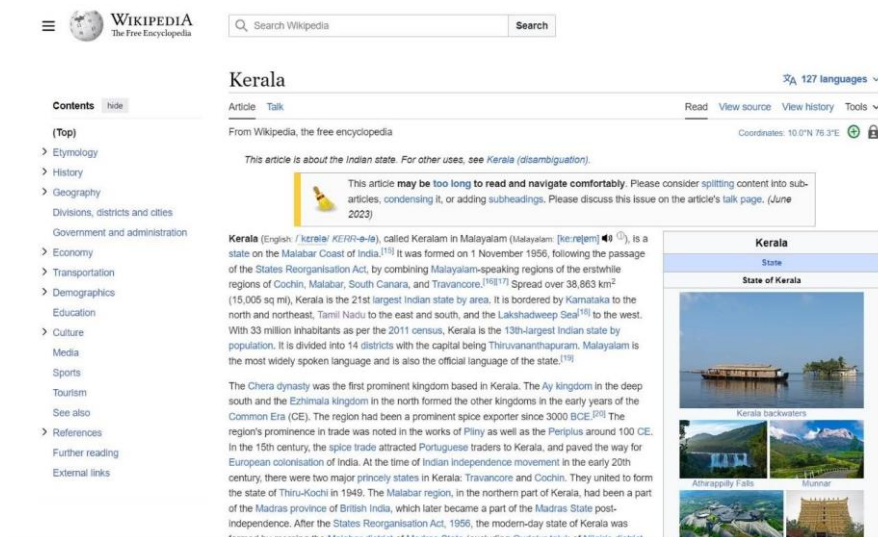
</map>

</div>

</body>

</html>
```

Output:



RESULT:

Thus the creation of a web page which includes a map and display the related in-formation when a hot spot is clicked in the map was executed successfully.

EXPT NO: Create a web page using an embedded, external, and inline CSS file.

DATE:

AIM:

To create a web page that displays college information using various style sheet.

PROCEDURE:

1. Create a web page with frame sets consisting two frames.
2. In the first frame include the links.
3. In the second frame set display the web page of the link.
4. Create a external style sheets.
5. Create a embedded style sheets.
6. Create a inline and internal style sheets and make it link to the external style sheets.

Code:

Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>The Mars</title>
<style>
body {
font-family: 'Arial', sans-serif;
```

```
margin: 0;
padding: 0;
background-color: #222;
color: #fff;
}
header {
background-color: #333;
color: #fff;
padding: 20px;
text-align: center;
}
nav {
background-color: #555;
padding: 10px;
}
nav ul {
list-style-type: none;
margin: 0;
padding: 0;
text-align: center;
}
nav ul li {
display: inline;
margin-right: 20px;
}
nav ul li a {
color: #fff;
```



```
text-decoration: none;
}
nav ul li a:hover {
text-decoration: underline;
}
main {
padding: 20px;
}
footer {
background-color: #333;
color: #fff;
padding: 10px;
text-align: center;
position: fixed;
width: 100%;
bottom: 0;
font-size: 14px;
}
</style>
</head>
<body>
<header>
<h1>THE MARS</h1>
</header>
<nav>
<ul>
```

HOME

MARS BEGINS

THE DARK MARS

THE MARS KNIGHT
RISES

</nav>

<main>

<h2 align="center">The planet mars</h2>

<p align="center">Mars is the fourth planet from the Sun. The surface of Mars is orange- red because it is covered in iron(III) oxide dust, giving it the nickname "the Red Planet".[21][22] Mars is among the brightest objects in Earth's sky and its high-contrast albedo features have made it a common subject for telescope viewing. It is classified as a terrestrial planet and is the second smallest of the Solar System's planets with a diameter of 6,779 km (4,212 mi). In terms of orbital motion, a Martian solar day (sol) is equal to 24.5 hours and a Martian solar year is equal to 1.88 Earth years (687 Earth days). Mars has two natural satellites that are small and irregular in shap.</p>

<div align="center">

<iframe width="560" height="315"

src="https://www.youtube.com/embed/D8pnmwOXhoY?si=2GGoqucO-
ER6Jfyz"

title="YouTube video player" frameborder="0" allow="accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"

allowfullscreen></iframe>r">

</div>

</main>

<footer>

© 2024 MARS. All rights reserved.

</footer>

```
</body>
```

```
</html>
```

Mars begins.html:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>THE MARS BEGINS</title>
```

```
<style>
```

```
body {
```

```
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
```

```
margin: 0;
```

```
padding: 0;
```

```
}
```

```
header {
```

```
background-color: #1d1d1d; /* Dark gray background color */
```

```
color: #fff;
```

```
padding: 20px;
```

```
text-align: center;
```

```
}
```

```
nav {
```

```
background-color: #333; /* Slightly darker gray navigation background color */
```

```
padding: 10px;
```

```
}
```

```
nav ul {
```

```
list-style-type: none;
```

```
margin: 0;
```

```
padding: 0;
```

```
text-align: center;
}
nav ul li {
display: inline;
margin-right: 20px;
}
nav ul li a {
color: #fff;
text-decoration: none;
font-weight: bold;
font-size: 16px;
}
nav ul li a:
hover {
text-decoration: underline;
}
main {
padding: 20px;
background-color: #f5f5f5; /* Light gray background color */
color: #333; /* Dark gray text color */
}
footer {
background-color: #1d1d1d; /* Dark gray footer background color */
color: #fff;
padding: 10px;
text-align: center;
position: fixed;
```

```
width: 100%;
bottom: 0;
font-size: 14px;
}
</style>
</head>
<body>
<header>
<h1 style="font-family: 'Arial', sans-serif;">THE MARS BEGINS</h1>
</header>
<nav>
<ul>
<li><a href="index2.html">HOME</a></li>
<li><a href="mars-begins.html">MARS BEGINS</a></li>
<li><a href="the-mars-knight.html">THE MARS KNIGHT</a></li>
<li><a href="the-mars-knight-rises.html">THE MARS KNIGHT
RISES</a></li>
</ul>
</nav>
<main>
<h2 align="center" style="color: #333;">Mars Begins</h2>
<p align="center">The planet Mars has been explored remotely by spacecraft. Probes sent from Earth, beginning in the late 20th century, have yielded a large increase in knowledge about the Martian system, focused primarily on understanding its geology and habitability potential.[1][2] Engineering interplanetary journeys is complicated and the exploration of .Mars has experienced a high failure rate, especially the early attempts. Roughly sixty percent of all spacecraft destined for Mars failed before completing their missions, with some failing before their observations could even begin. Some missions have been met with unexpected success, such as the twin Mars
```

Exploration Rovers, Spirit and Opportunity, which operated for years beyond their specification.</p>

<div align="center">

<iframe width="560" height="315"

src="https://www.youtube.com/embed/coZ83RM1jbk?si=0tmgK6isiBb39-62" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>

</div>

</main>

<footer>

© 2024 MARS. All rights reserved.

</footer>

</body>

</html>

Mars knight.html:

<!DOCTYPE html>

<html lang="en">

<head>

<title>TRIOLOGY - The Dark MARS </title>

<style>

body {

font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

margin: 0;

padding: 0;

background-color: #f5f5f5;

color: #333;

}

```
header {
background-color: #333;
color: #fff;
padding: 20px;
text-align: center;
font-size: 24px;
}

nav {
background-color: #555;
padding: 10px;
}

nav ul {
list-style-type: none;
margin: 0;
padding: 0;
text-align: center;
}

nav ul li {
display: inline;
margin-right: 20px;
}

nav ul li a {
color: #fff;
text-decoration: none;
font-size: 16px;
}

nav ul li a:hover {
```

```
text-decoration: underline;
```

```
}
```

```
main {
```

```
padding: 20px;
```

```
}
```

```
footer {
```

```
background-color: #333;
```

```
color: #fff;
```

```
padding: 10px;
```

```
text-align: center;
```

```
position: fixed;
```

```
width: 100%;
```

```
bottom: 0;
```

```
font-size: 14px;
```

```
}
```

```
h2 {
```

```
color: #333;
```

```
}
```

```
p {
```

```
color: #555;
```

```
font-size: 16px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<header>
```

```
<h1>MARS</h1>
```


</header>

<nav>

HOME

MARS BEGINS

THE MARS KNIGHT

THE MARS KNIGHT
RISES

</nav>

<main>

<h2 align="center">The Dark Mars</h2>

<p align="center">Mars is approximately half the diameter of Earth, with a surface area only slightly less than the total area of Earth's dry land.[2] Mars is less dense than Earth, having about 15% of Earth's volume and 11% of Earth's mass, resulting in about 38% of Earth's surface gravity. Mars is the only presently known example of a desert planet, a rocky planet with a surface akin to that of Earth's hot deserts. The red-orange appearance of the Martian surface is caused by ferric oxide, or rust.[40] It can look like butterscotch;[41] other common surface colors include golden, brown, tan, and greenish, depending on the minerals present.[. </p>

<div align="center">

<iframe width="560" height="315"

src="https://www.youtube.com/embed/IvjCvaJFsbs?si=toqAAUNunW4hn7WD
"title="YouTube video player" frameborder="0" allow="accelerometer;
autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-
share"allowfullscreen></iframe>

</div>

</main>

<footer>

© 2024 MARS. All rights reserved.

```
</footer>
```

```
</body>
```

```
</html>
```

Mars knight rises.html:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title>Mars</title>
```

```
<style>
```

```
body {
```

```
font-family: Arial, sans-serif;
```

```
margin: 0;
```

```
padding: 0;
```

```
}
```

```
header {
```

```
background-color: #0a0202;
```

```
color: #fff;
```

```
padding: 20px;
```

```
text-align: center;
```

```
}
```

```
nav {
```

```
background-color: #353434;
```

```
padding: 10px;
```

```
}
```

```
nav ul {
```

```
list-style-type: none;
```

```
margin: 0;
padding: 0;
text-align: center;
}
nav ul li {
display: inline;
margin-right: 20px;
}
nav ul li a {
color: #fff;
text-decoration: none;
}
nav ul li a:hover {
text-decoration: underline;
}
main {
padding: 20px;
}
footer {
background-color: #0a0202;
color: #fff;
padding: 10px;
text-align: center;
position: fixed;
width: 100%;
bottom: 0;
}
```

```
</style>
</head>
<body>
<header>
<h1>MARS</h1>
</header>
<nav>
<ul>
<li><a href="index2.html">HOME</a></li>
<li><a href="mars begins.html">MARS BEGINS</a></li>
<li><a href="the-mars-knight.html">THE MARS KNIGHT</a></li>
<li><a href="the-mars-knight rises.html">THE MARS KNIGHT
RISES</a></li>
</ul>
</nav>
<main>
<h2 align="center">The mars knight rises</h2>

<p align="center">Mars can appear bright or faint in our sky. 2024 is mostly a
faint year. Around February, Mars becomes visible in the east before dawn. It's
faint and far across the solar system from Earth..</p>
<div align="center">
<iframe width="560" height="315"
src="https://www.youtube.com/embed/Vy_RPd0rbII?si=f0KP6n-63vFgnn3X"
title="YouTubevideo player" frameborder="0" allow="accelerometer; autoplay;
clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
allowfullscreen></iframe>
</div>
</main>
<footer>
```

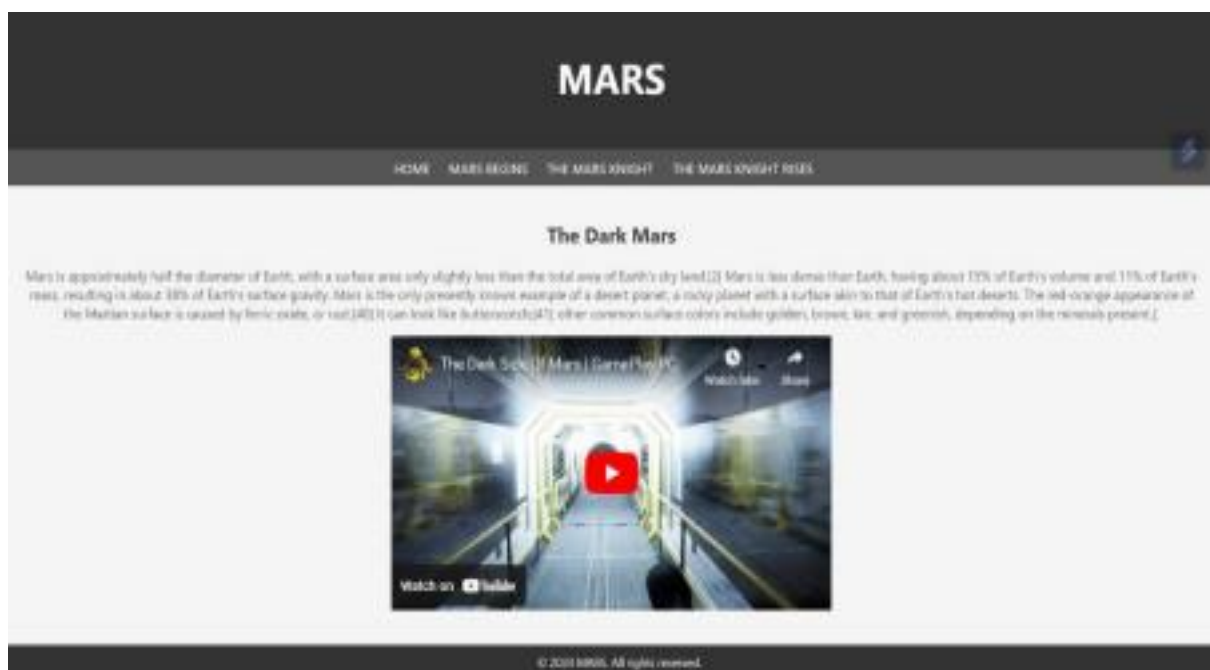
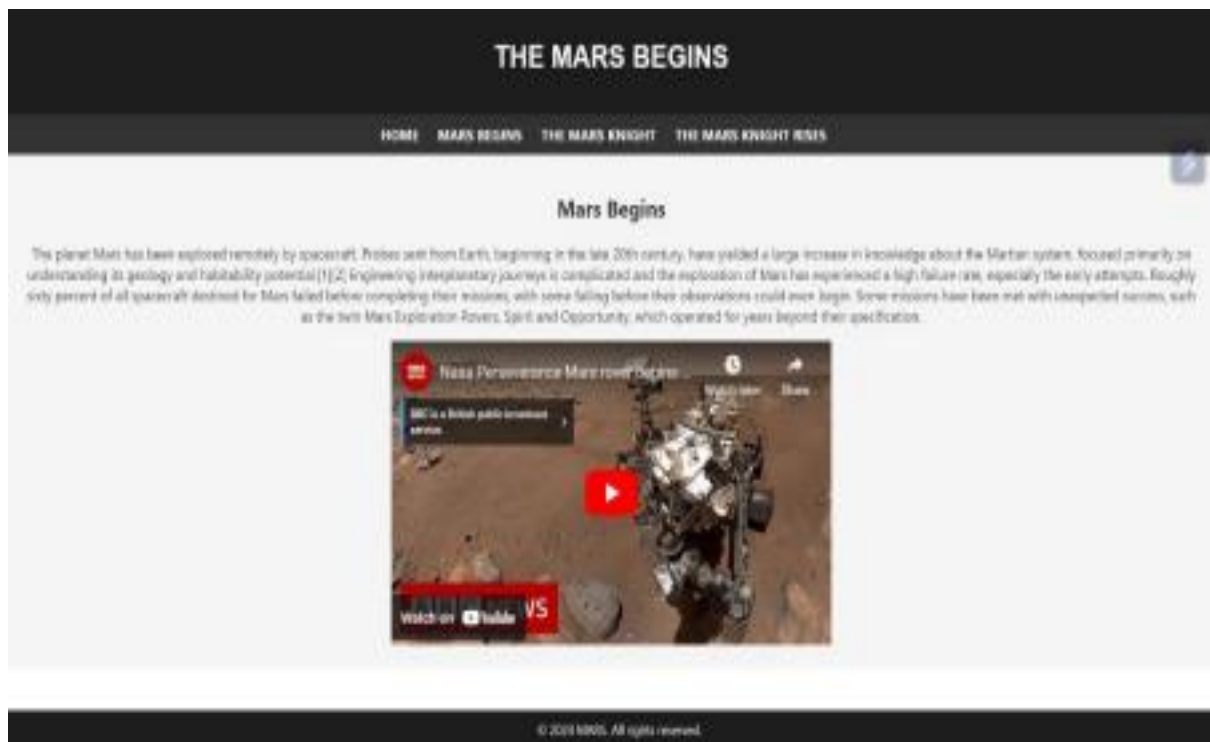
</footer>

</body>

</html>

Output:





RESULT:

Thus the creation of a web page that displays college information using various style sheet was successfully executed and verified.

EXPT NO: Create a registration page along with validations.

DATE:

Aim:

To create a visually appealing registration form with validation for email addresses.

Procedure:

1. Layout Design:

Design the layout of the registration form, including input fields for username, email, and password, along with a submit button.

2. Styling:

Apply CSS to style the form elements, providing appropriate spacing, alignment, and background color to enhance visual appeal.

3. Email Validation:

Implement JavaScript to validate the email address entered by the user. Ensure that the email follows the standard format and contains the "@" symbol.

4. Error Handling:

Display error messages if the email entered by the user is invalid. These messages should provide clear guidance on how to correct the input.

5. Submission Handling:

Handle form submission events, ensuring that the form data is submitted only if all fields are filled correctly, including a valid email address. If any field is invalid, prevent form submission and prompt the user to correct the errors.

Code:

Index.html:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-
scale=1.0"> <title>UserValidation</title>
  <link rel="stylesheet" href="style.css">

  <link rel="preconnect" href="https://fonts.googleapis.com">

  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

  <link
href="https://fonts.googleapis.com/css2?family=Protest+Guerrilla&display=
swap" rel="stylesheet">

  <style>

  body {

    font-family: 'Protest Guerrilla', sans-serif;

    color: #333; /* Updated text color */

    background-color: #f5f5f5; /* Updated background color */

  }

  .container {

    margin: 50px;

  }

  #form {

    width: 80%;

    margin: 10vh auto;

    background-color: #fff;
```



```
border-radius: 10px;
padding: 30px;
box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2);
}
h1 {
text-align: center;
color: #008793; /* Updated heading color */
font-size: 28px; /* Updated heading font size */
}
label {
color: #555; /* Updated label color
*/ font-size: 18px; /* Updated label font
size */ }
input {
border-radius: 5px;
font-size: 16px;
margin-top: 5px;
padding: 10px;
border: 1px solid #ddd;
}
button {
background-color: #4CAF50;
color: #fff;
cursor: pointer;
width: 100%;
font-size: 20px;
margin: 20px 0;
border-radius: 5px;
```

```
border: 1px solid #4CAF50;
padding: 10px;
transition: background-color 0.3s
ease; }

button:hover {
background-color: #45a049;
}

</style>

</head>

<body>

<div class="container">

<form action="message.html" id="form">

<h1>FORM VALIDATION</h1>

<div class="inp">

<label for="firstname">Firstname</label>

<input type="text" id="firstname"
name="firstname"> </div>

<div class="inp">

<label for="lastname">LastName</label>

<input type="text" id="lastname"
name="lastname"> </div>

<div class="inp">

<label for="email">Email</label>

<input type="email" id="email" name="email">

</div>

<div class="inp">

<label for="username">Username</label>

<input type="text" id="username"
name="username"> </div>

<div class="inp">
```

```
<label for="Password">Password</label>
<input type="password" id="Password"
name="Password"> </div>
<button onclick="return validate()"
type="submit">Register</button> </form>
<script src="index.js"></script>

</div>

</body>

</html>
```

Index.js:

```
function validate() {
    var fname = document.getElementById("firstname");
    var lname = document.getElementById("lastname");
    var email = document.getElementById("email");
    var uname = document.getElementById("username");
    var pass = document.getElementById("password");

    if (fname.value === "" || lname.value === "" || email.value === "" ||
    uname.value === "" || pass.value === "") {
        // Customize the alert message style
        alertMessage("No blank spaces are allowed", "red", "18px");
        return false;
    } else {
        return true;
    }
}

function alertMessage(message, color, fontSize) {
    var alertDiv = document.createElement("div");
    alertDiv.style.color = color || "red";
    alertDiv.style.fontSize = fontSize || "16px";
```

```
alertDiv.innerHTML = message;
```

```
// Append the alert message to the body or another suitable  
container document.body.appendChild(alertDiv);  
// Automatically remove the alert message after 3 seconds (adjust  
as needed)  
setTimeout(function () { document.body.removeChild(alertDiv); },  
3000);  
}
```

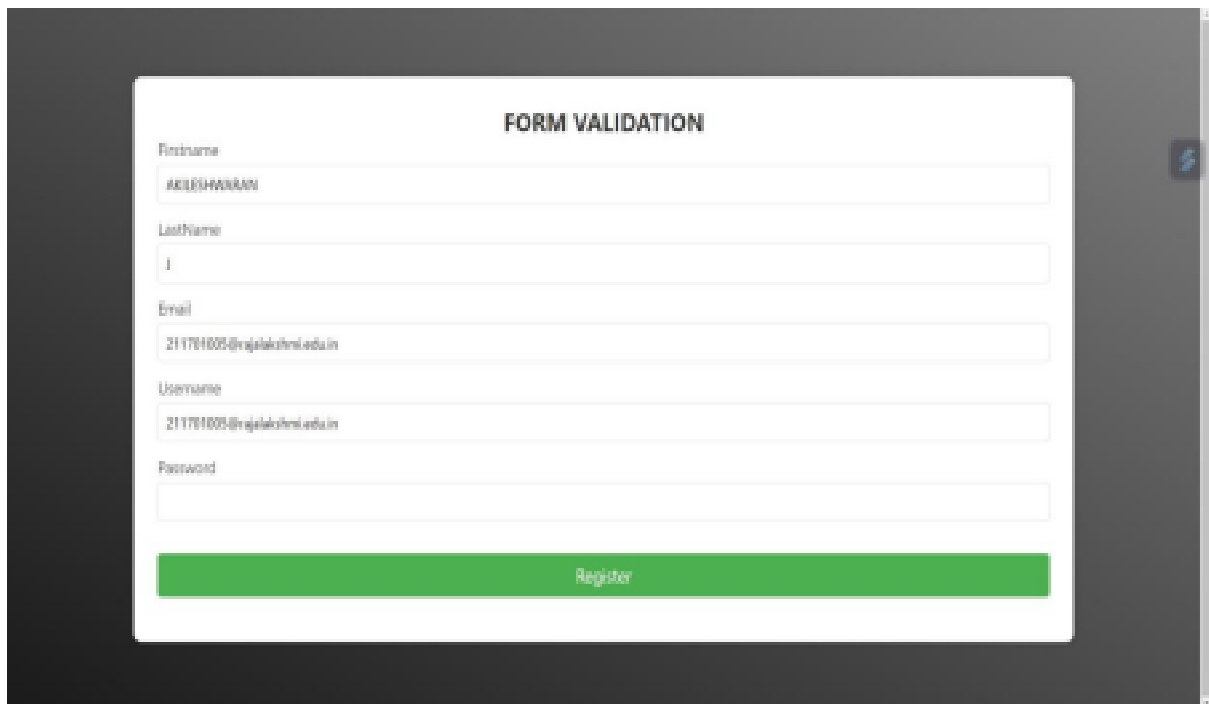
Style.css:

```
* {  
  margin: 0;  
  padding: 0;  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif; /* Updated  
font stack */ }  
  
body {  
  background-image: linear-gradient(to right top, #1a1a1a, #333333,  
#4d4d4d, #666666, #808080);  
  background-attachment: fixed;  
}  
  
#form {  
  width: 80%;  
  margin: 10vh auto;  
  background-color: #fff;  
  border-radius: 10px;  
  padding: 30px;  
  box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.2); /* Added box shadow  
for depth */ }  
  
@media(max-width: 700px) {
```

```
#form {  
width: 90%;  
  
margin-bottom: 50px;  
  
}  
  
#form h1 {  
text-align: center;  
  
color: #333; /* Adjusted heading color */  
  
}  
  
#form button {  
  
background-color: #4CAF50; /* Updated button  
color */ color: #fff;  
cursor: pointer;  
  
width: 100%;  
  
font-size: 20px;  
  
margin: 20px 0;  
  
border-radius: 5px;  
  
border: 1px solid #4CAF50;  
  
padding: 10px;  
  
transition: background-color 0.3s ease;  
  
}  
  
#form button:hover {  
  
background-color: #45a049; /* Adjusted hover  
color */ }  
  
inp {  
  
display: flex;  
flex-direction: column;  
  
margin-bottom: 15px;  
  
}
```

```
inp input {  
  border-radius: 5px;  
  font-size: 16px;  
  margin-top: 5px;  
  padding: 10px;  
  border: 1px solid #ddd; /* Added border for input fields */  
}
```

Output:



FORM VALIDATION

Firstname
ABHIRAM

Last Name
i

Email
211704003@rajalakshmi.edu.in

Username
211704003@rajalakshmi.edu.in

Password

Register

Result:

A visually appealing registration form with email validation ensures user inputs adhere to standard email format, enhancing data accuracy and user experience.

EXPT NO:

JSP - LIBRARY MANAGEMENT SYSTEM

DATE:

Aim:

To develop a JavaScript program that validates the controls in the forms of the Library Management System application, ensuring data integrity and user input correctness.

Procedure:

1. Identify forms requiring validation in the Library Management System.
2. Define validation rules for each form field.
3. Develop JavaScript functions for validation based on defined rules.
4. Integrate validation functions with form submission processes.
5. Prevent form submission if data fails validation, display error messages.
6. Thoroughly test and debug the validation functionality for usability and reliability.

Code:

Index.html:

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

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

```
  <title>Library Management System</title>
```

```
  <link rel="stylesheet" href="styles.css">
```

```
</head>
<body>
  <div class="container">
    <h2>Library Management System</h2>
    <form id="bookForm">
      <label for="title">Book Title:</label>
      <input type="text" id="title" name="title" required><br>
      <label for="author">Author:</label>
      <input type="text" id="author" name="author" required><br>
      <label for="year">Publication Year:</label>
      <input type="number" id="year" name="year" required><br>
      <button type="submit">Add Book</button>
    </form>
    <table id="bookTable">
      <thead>
        <tr>
          <th>Book Title</th>
          <th>Author</th>
          <th>Year</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
        <!-- Book rows will be added here dynamically -->
      </tbody>
    </table>
  </div>
```



```
<script>

    const bookForm = document.getElementById('bookForm');
    const bookTable =
document.getElementById('bookTable').getElementsByTagName('tbody')[0];
    bookForm.addEventListener('submit', function(event) {
        event.preventDefault();
        const title = document.getElementById('title').value;
        const author = document.getElementById('author').value;
        const year = document.getElementById('year').value;
        addBook(title, author, year);
        bookForm.reset();
    });
    function addBook(title, author, year) {
        const row = bookTable.insertRow(-1);
        const titleCell = row.insertCell(0);
        const authorCell = row.insertCell(1);
        const yearCell = row.insertCell(2);
        const actionCell = row.insertCell(3);
        titleCell.textContent = title;
        authorCell.textContent = author;
        yearCell.textContent = year;
        actionCell.innerHTML = '<button
onclick="editBook(this)">Edit</button>';
    }
    function editBook(button) {
        const row = button.parentElement.parentElement;
        const title = row.cells[0].textContent;
        const author = row.cells[1].textContent;
```

```
const year = row.cells[2].textContent;
const newTitle = prompt('Enter new title:', title);
const newAuthor = prompt('Enter new author:', author);
const newYear = prompt('Enter new year:', year);
if (newTitle && newAuthor && newYear) {
    row.cells[0].textContent = newTitle;
    row.cells[1].textContent = newAuthor;
    row.cells[2].textContent = newYear;
}
}
</script>
</body>
</html>
```

Style.css:

```
body {
    font-family: 'Arial', sans-serif;
    background-color: #e8f5e9;
    margin: 0;
    padding: 0;
}
.container {
    max-width: 800px;
    margin: 40px auto;
    padding: 30px;
    background-color: #ffffff;
    border-radius: 10px;
    box-shadow: 0 0 15px rgba(0, 0, 0, 0.2);
}
```

```
}  
h2 {  
    text-align: center;  
    color: #2e7d32;  
    font-size: 28px;  
    margin-bottom: 20px;  
}  
form {  
    margin-bottom: 30px;  
}  
form label {  
    display: block;  
    margin-bottom: 8px;  
    font-size: 16px;  
}  
form input[type="text"],  
form input[type="number"] {  
    width: calc(100% - 16px);  
    padding: 10px;  
    margin-bottom: 15px;  
    font-size: 16px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
}  
form button[type="submit"] {  
    padding: 10px 25px;  
    background-color: #43a047;
```

```
    color: #fff;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
    transition: background-color 0.3s;
}
form button[type="submit"]:hover {
    background-color: #388e3c;
}
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
}
th, td {
    padding: 12px;
    border-bottom: 1px solid #ddd;
    text-align: left;
}
th {
    background-color: #f1f8e9;
    color: #2e7d32;
}
tr:hover {
    background-color: #f1f1f1;
}
```

```
button {  
    padding: 5px 10px;  
    background-color: #0288d1;  
    color: white;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
}  
  
button:hover {  
    background-color: #0277bd;  
}
```

Output:

Library Management System

Book Title:

Author:

Publication Year:

Add Book

Library Management System



Book Title:

Author:

Publication Year:

Add Book

Book Title	Author	Year	Action
apple	kumar	2024	Edit

Result:

Developed JavaScript program to validate Library Management System forms, ensuring data integrity and input correctness.

EXPT NO: PHP – Employee Details

DATE:

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'];
$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 empdetails 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>
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 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>
<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>

<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. :

Name :

Designation :

Department :

DOJ :

Salary :

Add Employee Details

[Home](#)

Employee Id. :

Name :

Designation :

Department :

DOJ :

Salary :



Add Employee Details

Data added successfully.

[View Result](#)

Result:

Hence, 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 is executed successfully.

EXPT NO: **Servlet - Bank Application**

DATE:

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));
```

index.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Banking Application</title>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
<h1 align="center" class="title">Banking Application</h1>
<hr />
<div class="container">
  <a href="Customer.html" class="link">Customer Details</a>
  <a href="Account.html" class="link">Account Details</a>
</div>
</body>
</html>
```

Style.css:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 0;
}
.container {
  display: flex;
  justify-content: center;
```

```
    margin-top: 50px;
}
.title {
    color: #007bff;
}
.link {
    background-color: #007bff;
    color: white;
    text-decoration: none;
    padding: 10px 20px;
    margin: 10px;
    border-radius: 5px;
    transition: background-color 0.3s ease;
}
.link:hover {
    background-color: #0056b3;
}
```

Customer.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Customer Details</title>
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
<h1 align="center" class="title">Customer Details</h1>
```

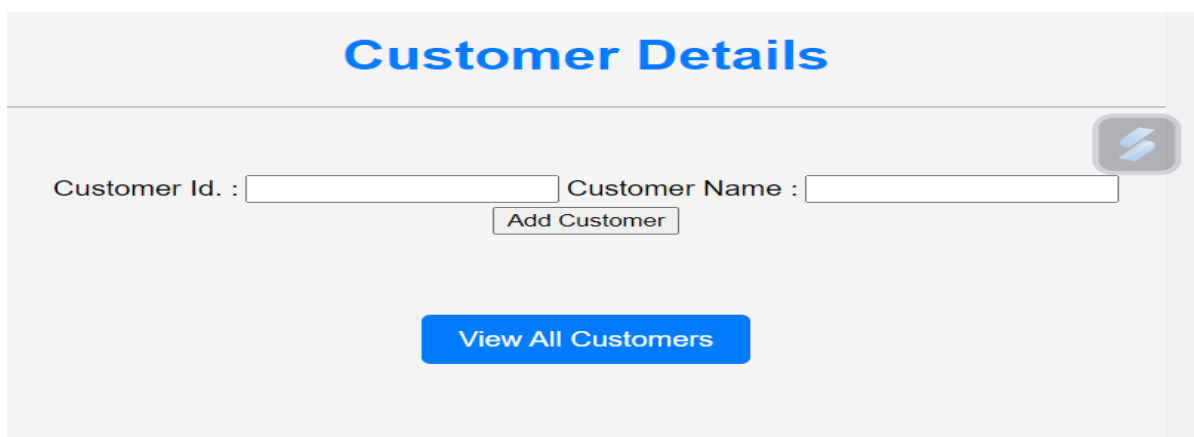
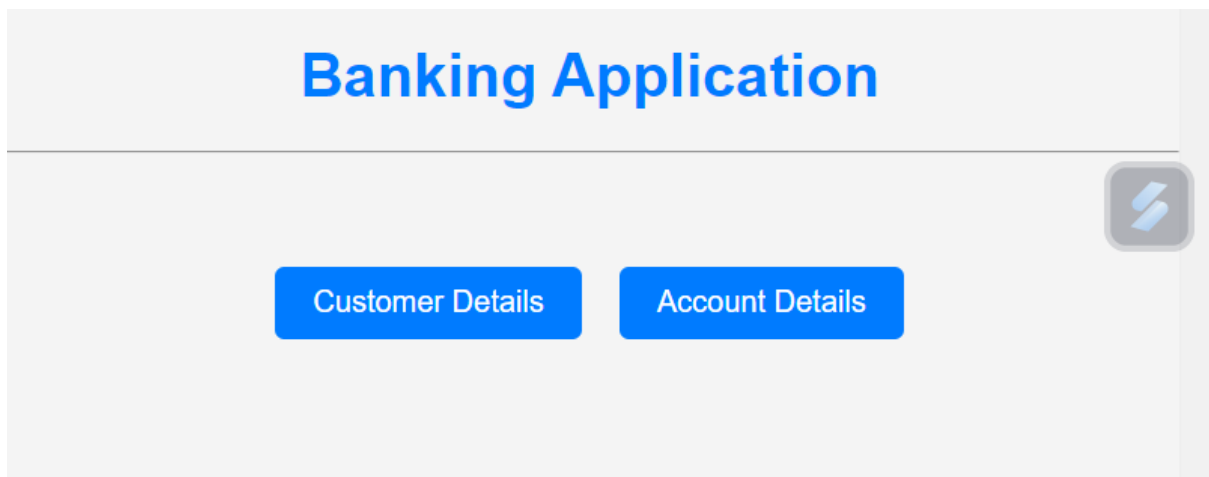
```
<hr />
<div class="container">
  <form action="AddCustomer" method="post">
    <table align="center">
      <tr>
        <td>Customer Id. :</td>
        <td><input type="text" name="cid"></td>
      <tr>
        <td>Customer Name :</td>
        <td><input type="text" name="cname"></td>
      </tr>
      <tr>
        <td colspan="4" align="center"><input type="submit" value="Add
Customer" class="button"></td>
      </tr>
    </table>
  </form>
</div>
<div class="container">
  <a href="ViewCustomers" class="link">View All Customers</a>
</div>
</body>
</html>
```

Account.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Account Details</title>
```

```
<link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
<h1 align="center" class="title">Account Details</h1>
<hr />
<div class="container">
  <p>No account details available</p>
</div>
</body>
</html>
```

Output:



Customer Details

Customer Id. :

Customer Name :

Account Details

No account details available

Result:

Hence developed a Banking application accessing a database using Servlet is implemented successfully.

EXPT NO: BOOTSTRAP – WEB PAGE

DATE:

Aim:

Program to develop an attractive web pages using Bootstrap.

Procedure:

1. File Setup:

- Create `index.html`, `styles.css`, and `scripts.js` files.
- Ensure they are in the same directory for easy access.

2. Bootstrap Integration:

- Link Bootstrap CSS in the HTML file using the CDN (Content Delivery Network)

```
<link  
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"  
rel="stylesheet">
```

3. HTML Structure:

- Set up the basic structure of `index.html`.
- Organize sections such as "About Me," "Portfolio," and "Contact."

4. Content Integration:

- Add relevant content to each section.
- Include text, images, and links to showcase your work and skills.

5. Customization and Styling:

- Customize the appearance using custom CSS in `styles.css`.
- Adjust Bootstrap classes and add additional styles as needed for a personalized look.

PROGRAM:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Portfolio</title>
  <!-- Bootstrap CSS -->
  <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
  <!-- Font Awesome -->
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css" rel="stylesheet">
  <!-- Custom CSS -->
  <link href="styles.css" rel="stylesheet">
<style>
  /* Custom CSS */
  body {
    font-family: Arial, sans-serif;
    color: #333;
  }
  h1, h2, h3, h4, h5, h6 {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    color: #333;
  }
  p {
    font-size: 16px;
```



```
}  
.navbar {  
  background-color: #343a40 !important;  
}  
.navbar-brand, .nav-link {  
  color: #fff !important;  
}  
.navbar-toggler-icon {  
  color: #fff !important;  
}  
#about, #portfolio, #contact {  
  padding: 50px 0;  
}  
.card-title {  
  font-size: 20px;  
  color: #333;  
}  
.card-text {  
  font-size: 16px;  
  color: #666;  
}  
.btn-primary {  
  background-color: #007bff;  
  border-color: #007bff;  
}  
.btn-primary:hover {  
  background-color: #0056b3;
```

```
        border-color: #0056b3;
    }
    .text-primary {
        color: #007bff !important;
    }
</style>
</head>
<body>
<!-- Navigation -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
        <a class="navbar-brand" href="#">My Portfolio</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav ml-auto">
                <li class="nav-item">
                    <a class="nav-link" href="#about">About</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#portfolio">Portfolio</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="#contact">Contact</a>
                </li>
            </ul>
        </div>
    </div>
</nav>
```

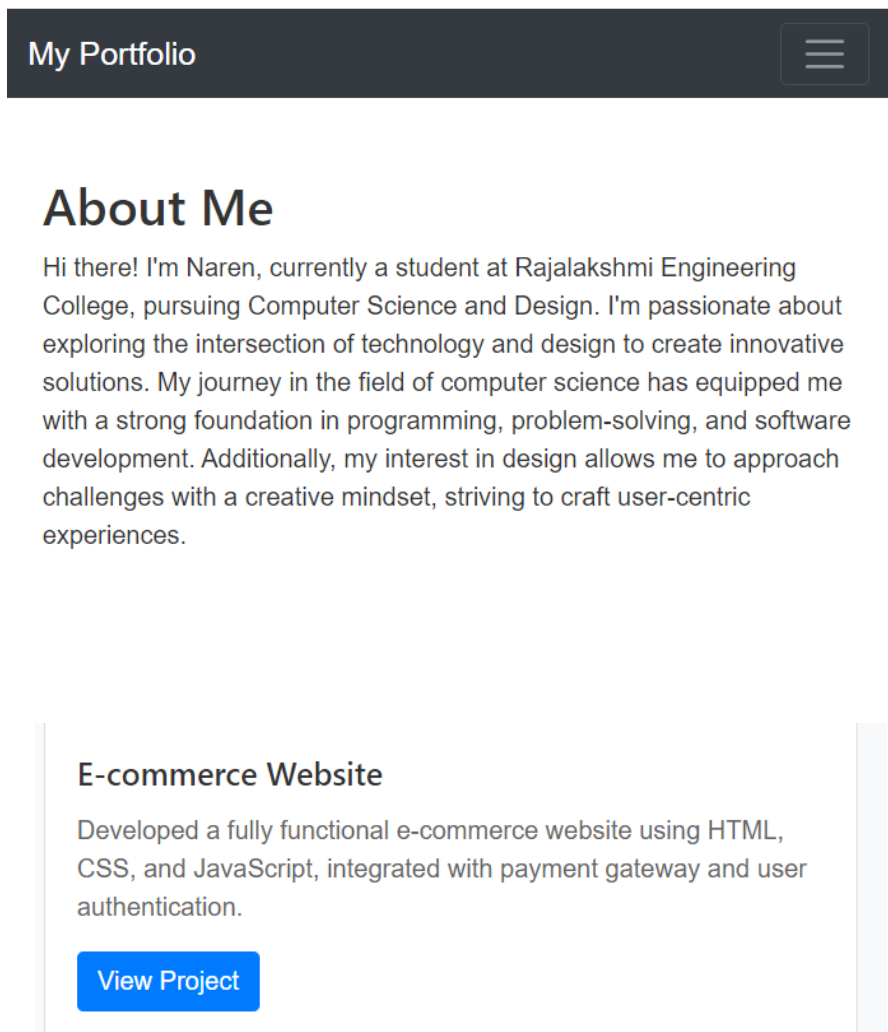
```
</ul>
</div>
</div>
</nav>
<!-- About Section -->
<section id="about" class="py-5">
  <div class="container">
    <div class="row">
      <div class="col-lg-6">
        <h2>About Me</h2>
        <p>Hi there! I'm Naren, currently a student at Rajalakshmi Engineering
        College, pursuing Computer Science and Design. I'm passionate about
        exploring the intersection of technology and design to create innovative
        solutions. My journey in the field of computer science has equipped me with a
        strong foundation in programming, problem-solving, and software development.
        Additionally, my interest in design allows me to approach challenges with a
        creative mindset, striving to craft user-centric experiences.</p>
      </div>
      <div class="col-lg-6">
        
      </div>
    </div>
  </div>
</section>
<!-- Portfolio Section -->
<section id="portfolio" class="bg-light py-5">
  <div class="container">
    <h2>Portfolio</h2>
    <div class="row">
```

```
<div class="col-md-4">
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">E-commerce Website</h5>
      <p class="card-text">Developed a fully functional e-commerce website
using HTML, CSS, and JavaScript, integrated with payment gateway and user
authentication.</p>
      <a href="#" class="btn btn-primary">View Project</a>
    </div>
  </div>
</div>
<div class="col-md-4">
  <div class="card">
    
    <div class="card-body">
      <h5 class="card-title">Portfolio Website</h5>
      <p class="card-text">Designed and built a responsive portfolio website
to showcase personal projects, skills, and experiences using Bootstrap and
custom CSS.</p>
      <a href="#" class="btn btn-primary">View Project</a>
    </div>
  </div>
</div>
<div class="col-md-4">
  <div class="card">
    
```

```
<div class="card-body">
  <h5 class="card-title">Blog Website</h5>
  <p class="card-text">Created a dynamic blog website using WordPress,
customized themes, and plugins to enhance functionality and user
experience.</p>
  <a href="#" class="btn btn-primary">View Project</a>
</div>
</div>
</div>
</div>
</div>
</section>
<!-- Contact Section -->
<section id="contact" class="py-5">
  <div class="container">
    <h2>Contact Me</h2>
    <div class="row">
      <div class="col-md-6">
        <form>
          <div class="form-group">
            <label for="name">Name</label>
            <input type="text" class="form-control" id="name">
          </div>
          <div class="form-group">
            <label for="email">Email address</label>
            <input type="email" class="form-control" id="email">
          </div>
          <div class="form-group">
```

```
<label for="message">Message</label>
<textarea class="form-control" id="message" rows="3"></textarea>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>
<div class="col-md-6">
  <div class="card bg-primary text-white">
    <div class="card-body">
      <h5 class="card-title">Get In Touch</h5>
```

OUTPUT:



Portfolio Website

Designed and built a responsive portfolio website to showcase personal projects, skills, and experiences using Bootstrap and custom CSS.

[View Project](#)

Contact Me

Name

Email address

Message

Submit

Get In Touch

RESULT:

Hence developed an attractive web pages using Bootstrap.

EXPT NO: 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 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>
```

```
    <link rel="stylesheet" href="style.css">
```

```
    <script src="https://code.angularjs.org/1.2.13/angular.js"></script>
```

```
    <script src="//cdnjs.cloudflare.com/ajax/libs/angular-material-  
icons/0.7.1/angular-material-icons.min.js"></script>
```

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



```
</head>
```

```
<body>
```

```
<!-- Adding the ng-app declaration to initialize AngularJS -->
```

```
<div id="main" ng-app="navApp">
```

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

```
    <h2>My Shopping Site</h2>
```

```
    <a href="#" class="home" ng-click="active='home'">
```

```
      <ng-md-icon icon="home" style="fill:white"></ng-md-icon>
```

```
    </a>
```

```
    <a href="#" class="electronics" ng-click="active='electronics'">
```

```
      Electronics</a>
```

```
    <a href="#" class="appliances" ng-click="active='appliances'">
```

```
      Appliances</a>
```

```
    <a href="#" class="clothing" ng-click="active='clothing'">
```

```
      Clothing</a>
```

```
  </nav>
```

```
  <p ng-hide="active">Please click a menu item</p>
```

```
  <p ng-show="active">You chose <b>{ { active} }</b></p>
```

```
</div>
```

```
</body>
```

```
</html>
```

STYLES.CSS:

```
*{  
  
    margin:0;  
  
    padding:0;  
  
}  
  
body{  
  
    font: 18px/1.5 'Arial', sans-serif;  
  
    color: #333;  
  
    text-align:center;  
  
    background-color: #f8f9fa;  
  
}  
  
a, a:visited {  
  
    outline:none;  
  
    color:#007bff; /* Blue color for links */  
  
}  
  
a:hover{  
  
    text-decoration:none;  
  
}  
  
a img{  
  
    display:inline-block;  
  
}  
  
section, footer, header, aside, nav{  
  
    display: block;
```

```
}  
  
nav{  
  
    display:inline-block;  
  
    margin:60px auto 45px;  
  
    background-color:#343a40; /* Dark background color for the navigation */  
  
    box-shadow:0 2px 4px rgba(0, 0, 0, 0.1);  
  
    border-radius:5px;  
  
}  
  
nav h2{  
  
    color:#fff; /* White color for the heading */  
  
    font-size: 24px;  
  
    padding: 15px 0;  
  
}  
  
nav a{  
  
    display:inline-block;  
  
    padding: 18px 30px;  
  
    color:#fff;  
  
    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:5px 0 0 5px;

}

nav a:last-child{

    border-radius:0 5px 5px 0;

}

nav.home .home,

nav.electronics .electronics,

nav.appliances .appliances,

nav.clothing .clothing{

    color:#ffc107; /* Yellow color for active link */

}

p{

    font-size:20px; /* Reduced font size */

    font-weight:bold;

    color:#6c757d; /* Dark gray color for paragraphs */

}
```

OUTPUT:

My Shopping Site

[Electronics](#) [Appliances](#) [Clothing](#)

Please click a menu item

You chose {{active}}

RESULT:

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

DESIGN A WEB PAGE WITH – 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:

```
<!DOCTYPE html>
<html lang="en" ng-app="inlineEditorApp">
<head>
  <meta charset="UTF-8">
  <title>Bootstrap & AngularJS Inline Editor</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
  <style>
    .center {
```

```
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
  }
  .title-container {
    display: flex;
    align-items: center;
  }
  .title {
    margin: 0;
    color: #333; /* Change title text color */
    font-size: 36px; /* Increase title font size */
  }
  .edit-input {
    width: 300px; /* Adjust input width */
    margin-right: 10px; /* Add margin to the right of the input */
  }
  .edit-btn {
    color: #007bff; /* Change button color */
    font-size: 20px; /* Change button icon size */
  }
</style>
</head>
<body>
  <div ng-controller="InlineEditorController as ctrl" class="container center">
    <div class="title-container">
```

```

<h1 class="title" ng-hide="ctrl.editingTitle">{{ ctrl.title }}</h1>

<div ng-show="ctrl.editingTitle">

  <input type="text" class="form-control edit-input" ng-model="ctrl.title"
id="title">

    <button class="btn btn-outline-secondary edit-btn" type="button" ng-
click="ctrl.saveField('title')"><i class="fas fa-check"></i></button>

  </div>

</div>

<button class="btn btn-outline-secondary mt-3 edit-btn" type="button" ng-
click="ctrl.editField('title')"><i class="fas fa-pencil-alt"></i> Edit
Title</button>

</div>

<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></scri
pt>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></s
cript>

<script>

angular.module('inlineEditorApp', [])

.controller('InlineEditorController', function() {

  var ctrl = this;

  ctrl.title = "Click to edit me";

  ctrl.editingTitle = false;

  ctrl.editField = function(field) {

    ctrl.editingTitle = true;

    // Set a timeout to focus the input after it's shown

    setTimeout(function() {

      document.getElementById(field).focus();

```



```
    });  
  };  
  ctrl.saveField = function(field) {  
    ctrl.editingTitle = false;  
  };  
});  
</script>  
</body>  
</html>
```

OUTPUT:

Click to edit me  Edit Title

Akileshwaran J  Edit Title

RESULT:

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

DESIGN A WEB PAGE WITH – 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:

```
<!DOCTYPE html>

<html lang="en" ng-app="orderFormApp">
<head>
  <meta charset="UTF-8">
  <title>Order Form</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
>
  <style>
    body, html {
      height: 100%;
      font-family: Arial, sans-serif; /* Change font */
    }
    .container {
      display: flex;
```

```
    justify-content: center;
    align-items: center;
    height: 100%;
}

.order-form {
    max-width: 400px;
    width: 100%;
    background-color: #f8f9fa; /* Change background color */
    padding: 20px;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1); /* Add shadow */
}

.form-group {
    margin-bottom: 20px;
}

.form-label {
    font-weight: bold;
    font-size: 16px; /* Increase label font size */
    color: #333; /* Change label text color */
}

.form-control {
    font-size: 16px; /* Increase input font size */
}

.btn-primary {
    font-size: 18px; /* Increase button font size */
}

.list-group-item {
```

```

        font-size: 14px; /* Decrease list item font size */
    }
    .mt-3 {
        margin-top: 15px; /* Increase margin */
    }
</style>
</head>
<body>
    <div ng-controller="OrderFormController as formCtrl" class="container">
        <form class="order-form">
            <div class="form-group">
                <label for="products" class="form-label">Products:</label>
                <select multiple class="form-control" id="products" ng-
model="formCtrl.selectedProducts" ng-options="product as product.name for
product in formCtrl.products"></select>
            </div>
            <div class="form-group">
                <label for="quantity" class="form-label">Quantity:</label>
                <input type="number" class="form-control" id="quantity" ng-
model="formCtrl.quantity">
            </div>
            <button type="button" class="btn btn-primary" ng-
click="formCtrl.addToCart()">Add to Cart</button>
            <div class="form-group mt-3">
                <label class="form-label">Cart:</label>
                <ul class="list-group">
                    <li class="list-group-item" ng-repeat="item in formCtrl.cart">{{
item.product.name }} - Quantity: {{ item.quantity }} - Total: ${{ item.total
}}</li>

```

```

    </ul>

    <p class="mt-3">Grand Total: ${{ formCtrl.grandTotal }}</p>
</div>

</form>

</div>

<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></scri
pt>

<script>

angular.module('orderFormApp', [])

.controller('OrderFormController', function() {

    var formCtrl = this;

    formCtrl.products = [

        { name: 'Web Essentials', cost: 100 },

        { name: 'Web Hosting', cost: 50 },

        { name: 'Domain Registration', cost: 20 }

    ];


    formCtrl.selectedProducts = [];

    formCtrl.quantity = "";

    formCtrl.cart = [];

    formCtrl.grandTotal = 0;


    // Function to add item to cart

    formCtrl.addToCart = function() {

        if (formCtrl.selectedProducts.length && formCtrl.quantity) {

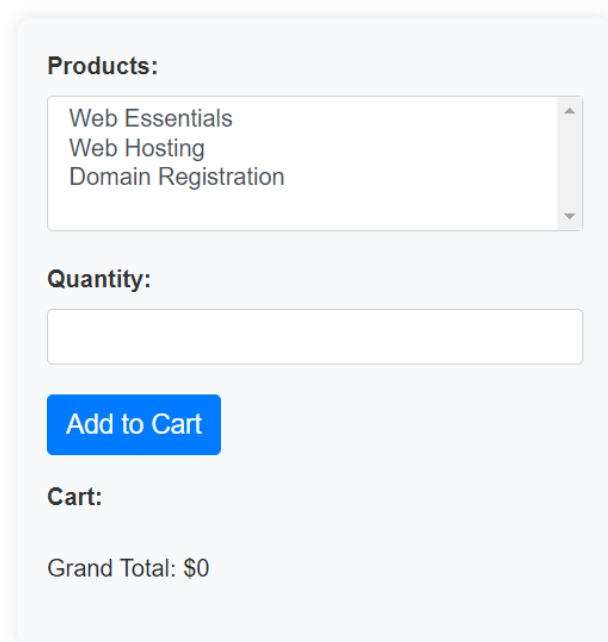
            angular.forEach(formCtrl.selectedProducts, function(product) {

                var total = product.cost * formCtrl.quantity;

```

```
        formCtrl.cart.push({ product: product, quantity: formCtrl.quantity, total:
total });
        formCtrl.grandTotal += total;
    });
    // Clear input fields after adding to cart
    formCtrl.selectedProducts = [];
    formCtrl.quantity = "";
    } else {
        console.log("Please select product(s) and enter a quantity.");
    }
};
});
</script>
</body>
</html>
```

OUTPUT:



The screenshot shows a web form with a light gray background and rounded corners. It contains the following elements:

- Products:** A label above a white dropdown menu with a vertical scrollbar. The menu is open, showing three options: "Web Essentials", "Web Hosting", and "Domain Registration".
- Quantity:** A label above a white text input field.
- Add to Cart:** A blue button with white text.
- Cart:** A label above the text "Grand Total: \$0".

Products:

Web Essentials
Web Hosting
Domain Registration

Quantity:

Add to Cart

Cart:

Web Hosting - Quantity: 13 - Total: \$650

Grand Total: \$650

RESULT:

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

DESIGN A WEB PAGE WITH – 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", [])`.

PROGRAM:

```
<!DOCTYPE html>

<html lang="en" ng-app="instantSearchApp">
<head>
  <meta charset="UTF-8">
  <title>Instant Search - Engineering Departments</title>
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
>
  <style>
    body {
      font-family: Arial, sans-serif; /* Change font */
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="row">
      <div class="col">
        <div class="input-group">
          <input type="text" class="form-control" />
          <div class="input-group-append">
            <button class="btn btn-outline-secondary" type="button">Search</button>
          </div>
        </div>
      </div>
      <div class="col">
        <div class="list-group">
          <div class="list-group-item">Engineering Department</div>
          <div class="list-group-item">Mechanical Engineering</div>
          <div class="list-group-item">Electrical Engineering</div>
          <div class="list-group-item">Computer Engineering</div>
          <div class="list-group-item">Civil Engineering</div>
          <div class="list-group-item">Chemical Engineering</div>
          <div class="list-group-item">Aeronautical Engineering</div>
          <div class="list-group-item">Marine Engineering</div>
          <div class="list-group-item">Metallurgical Engineering</div>
          <div class="list-group-item">Biomedical Engineering</div>
          <div class="list-group-item">Environmental Engineering</div>
          <div class="list-group-item">Food Engineering</div>
          <div class="list-group-item">Textile Engineering</div>
          <div class="list-group-item">Leather Engineering</div>
          <div class="list-group-item">Paper Engineering</div>
          <div class="list-group-item">Glass Engineering</div>
          <div class="list-group-item">Rubber Engineering</div>
          <div class="list-group-item">Plastic Engineering</div>
          <div class="list-group-item">Ceramic Engineering</div>
          <div class="list-group-item">Composite Engineering</div>
          <div class="list-group-item">Nanotechnology Engineering</div>
          <div class="list-group-item">Biotechnology Engineering</div>
          <div class="list-group-item">Environmental Engineering</div>
          <div class="list-group-item">Food Engineering</div>
          <div class="list-group-item">Textile Engineering</div>
          <div class="list-group-item">Leather Engineering</div>
          <div class="list-group-item">Paper Engineering</div>
          <div class="list-group-item">Glass Engineering</div>
          <div class="list-group-item">Rubber Engineering</div>
          <div class="list-group-item">Plastic Engineering</div>
          <div class="list-group-item">Ceramic Engineering</div>
          <div class="list-group-item">Composite Engineering</div>
          <div class="list-group-item">Nanotechnology Engineering</div>
          <div class="list-group-item">Biotechnology Engineering</div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```



```

h1 {
  color: #007bff; /* Change title color */
  font-size: 28px; /* Increase title font size */
}

.form-control {
  font-size: 16px; /* Increase input font size */
}

.list-group-item {
  font-size: 18px; /* Increase list item font size */
}
</style>
</head>
<body>
  <div ng-controller="SearchController as searchCtrl" class="container mt-5">
    <h1>Instant Search - Engineering Departments</h1>
    <input type="text" class="form-control mt-3" placeholder="Search..." ng-
model="searchCtrl.query">
    <ul class="list-group mt-3">
      <li class="list-group-item" ng-repeat="department in searchCtrl.departments
| filter:searchCtrl.query">{{ department }}</li>
    </ul>
  </div>
<script
src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></scri
pt>
<script>
  angular.module('instantSearchApp', [])
    .controller('SearchController', function() {
      var searchCtrl = this;

```

```
searchCtrl.departments = [  
  'Computer Science and Engineering',  
  'Computer Science and Design',  
  'Electrical and Electronics Engineering',  
  'Mechanical Engineering',  
  'Civil Engineering',  
  'Electronics and Communication Engineering',  
  'Chemical Engineering',  
  'Biomedical Engineering',  
  'Aerospace Engineering',  
  'Environmental Engineering',  
  'Materials Science and Engineering'  
];  
});  
</script>  
</body>  
</html>
```

OUTPUT:

Instant Search - Engineering Departments

Computer Science and Engineering
Computer Science and Design
Electronics and Communication Engineering

Instant Search - Engineering Departments

Computer Science and Engineering

Computer Science and Design

Electrical and Electronics Engineering

Mechanical Engineering

Civil Engineering

Electronics and Communication Engineering

Chemical Engineering

Biomedical Engineering

Aerospace Engineering

Environmental Engineering

Materials Science and Engineering

RESULT:

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

DESIGN A WEB PAGE WITH – SWITCHABLE GRID

Aim:

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

Procedure:

1. Create the HTML structure for the switchable grid interface. Include buttons for switching between grid and list views and define containers for displaying the grid and list.
2. Set up an AngularJS module and controller to manage the functionality of the switchable grid. Inject the AngularJS library into the HTML file.
3. Initialize data and view: Define a sample list of items and initialize the default view to be displayed (in this case, the grid view).
4. Create a function in the AngularJS controller to switch between grid and list views. This function will change the value of a variable representing the current view.
5. Use AngularJS directives such as ng-show and ng-repeat to bind the data to the grid and list views. Show or hide each view based on the current view variable, and iterate over the list of items to display them in the respective views.

PROGRAM:

```
<!DOCTYPE html>

<html lang="en" ng-app="switchableViewApp">
<head>
  <meta charset="UTF-8">
  <title>Switchable Grid</title>
  <link
                                                                    rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f8f9fa;
```

```
}  
.container {  
  margin-top: 50px;  
}  
h1 {  
  color: #007bff;  
  font-size: 36px;  
  text-align: center;  
}  
.btn-group button {  
  background-color: #28a745;  
  color: white;  
  border: none;  
  margin-right: 10px;  
}  
.btn-group button:hover {  
  background-color: #218838;  
}  
.grid-view, .list-view {  
  margin-top: 20px;  
}  
.card {  
  background-color: #fff;  
  border: 1px solid #ced4da;  
  border-radius: 5px;  
  padding: 15px;  
}
```

```

.card-title {
  color: #343a40;
  font-size: 20px;
  margin-bottom: 10px;
}
.card-text {
  color: #6c757d;
  font-size: 16px;
}
.item-img {
  max-width: 100%;
  height: auto;
}
</style>
</head>
<body>
<div ng-controller="SwitchableGridController as gridCtrl" class="container">
  <h1>Switchable Grid</h1>
  <div class="btn-group mb-3">
    <button class="btn btn-primary" ng-
click="gridCtrl.switchView('grid')">Grid View</button>
    <button class="btn btn-primary" ng-click="gridCtrl.switchView('list')">List
View</button>
  </div>
  <div class="grid-view" ng-show="gridCtrl.currentView === 'grid'">
    <div class="row">
      <div class="col-md-4 mb-3" ng-repeat="item in gridCtrl.items">
        <div class="card">

```

```

```

```
<div class="card-body">
```

```
<h5 class="card-title">{{ item.name }}</h5>
```

```
<p class="card-text">{{ item.description }}</p>
```

```
<p class="card-text">Price: ${{ item.price }}</p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class="list-view" ng-show="gridCtrl.currentView === 'list'">
```

```
<ul class="list-group">
```

```
<li class="list-group-item" ng-repeat="item in gridCtrl.items" style="color: #343a40;">{{ item.name }} - ${{ item.price }}</li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.2/angular.min.js"></script>
```

```
<script>
```

```
angular.module('switchableViewApp', [])
```

```
.controller('SwitchableGridController', function() {
```

```
var gridCtrl = this;
```

```
gridCtrl.items = [
```

```
{ name: 'Product 1', description: 'Description of Product 1', price: 20 },
```

```
{ name: 'Product 2', description: 'Description of Product 2', price: 30 },
```

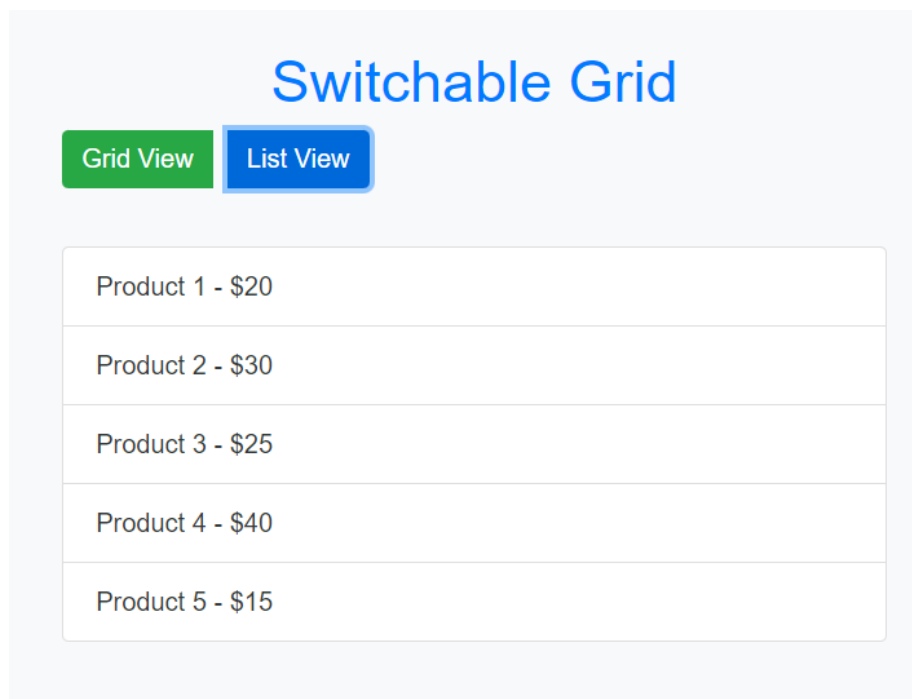
```
{ name: 'Product 3', description: 'Description of Product 3', price: 25 },
```

```

    { name: 'Product 4', description: 'Description of Product 4', price: 40 },
    { name: 'Product 5', description: 'Description of Product 5', price: 15 }
  ];
  gridCtrl.currentView = 'grid';
  gridCtrl.switchView = function(view) {
    gridCtrl.currentView = view;
  };
});
</script>
</body>
</html>

```

OUTPUT:



RESULT:

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

EXPT NO: SINGLE PAGE APPLICATION

DATE:

AIM:

Program to develop an single page application using angular js.

PROCEDURE:

1. Make a single page application and don't want any page refreshes, use Angular's routing capabilities.
2. Include angular-route script after the main angular script. 8. Specify that the module depends on ngRoute module to be able to use it.
3. The next thing is to distinguish common HTML for every page. This HTML will be layout of the website.
4. Then specify the place where HTML of each page will be placed in our layout. There is a ng-view directive for that.
5. 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.
6. Configure the routes. Use \$routeProvider service from the ngRoute module.
7. For each route, specify templateUrl and controller.
8. 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:
9. Build controllers for every route (already specified their names in routeProvider).

PROGRAM:

```
<!DOCTYPE html>
<html ng-app="myApp">
<head>
  <meta charset="UTF-8">
  <title>Single Page Application</title>
  <style>
```

```
body {  
    font-family: 'Helvetica Neue', Helvetica, Arial, sans-serif;  
    margin: 0;  
    padding: 0;  
    background-color: #f0f0f0;  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    align-items: center;  
    height: 100vh;  
}  
#navbar {  
    background-color: #2c3e50;  
    overflow: hidden;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    width: 100%;  
    position: fixed;  
    top: 0;  
    z-index: 1000;  
}  
#navbar a {  
    display: block;  
    color: white;  
    text-align: center;  
    padding: 14px 20px;
```

```
    text-decoration: none;
    font-size: 18px;
}
#navbar a:hover {
    background-color: #34495e;
    color: #ecf0f1;
}
#navbar a.active {
    background-color: #2980b9;
    color: white;
}
.content {
    padding: 20px;
    text-align: center;
    margin-top: 80px; /* Adjusted margin to make space for navbar */
}
h1 {
    margin-top: 0;
    font-size: 36px;
    color: #2c3e50;
}
h3 {
    font-size: 24px;
    color: #7f8c8d;
}
img {
    max-width: 100%;
```

```
        height: auto;

        border: 2px solid #2c3e50;

        border-radius: 10px;

    }

</style>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular.min.js"></sc
ript>

<script src="https://cdnjs.cloudflare.com/ajax/libs/angular.js/1.4.7/angular-
route.min.js"></script>

</head>

<body>

    <div id="navbar">

        <a href="#/" class="nav-link active">Home</a>

        <a href="#/services" class="nav-link">Services</a>

        <a href="#/aboutus" class="nav-link">About Us</a>

    </div>

    <div ng-view class="content"></div>

    <script type="text/ng-template" id="pages/home.html">

        <h1>Welcome Home</h1>

        <h3>{{message}}</h3>

    </script>

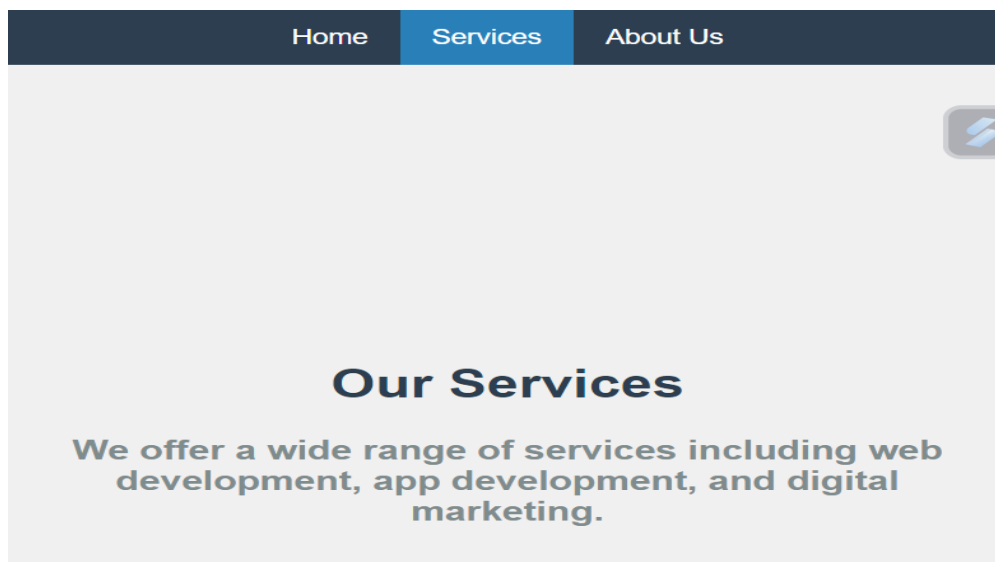
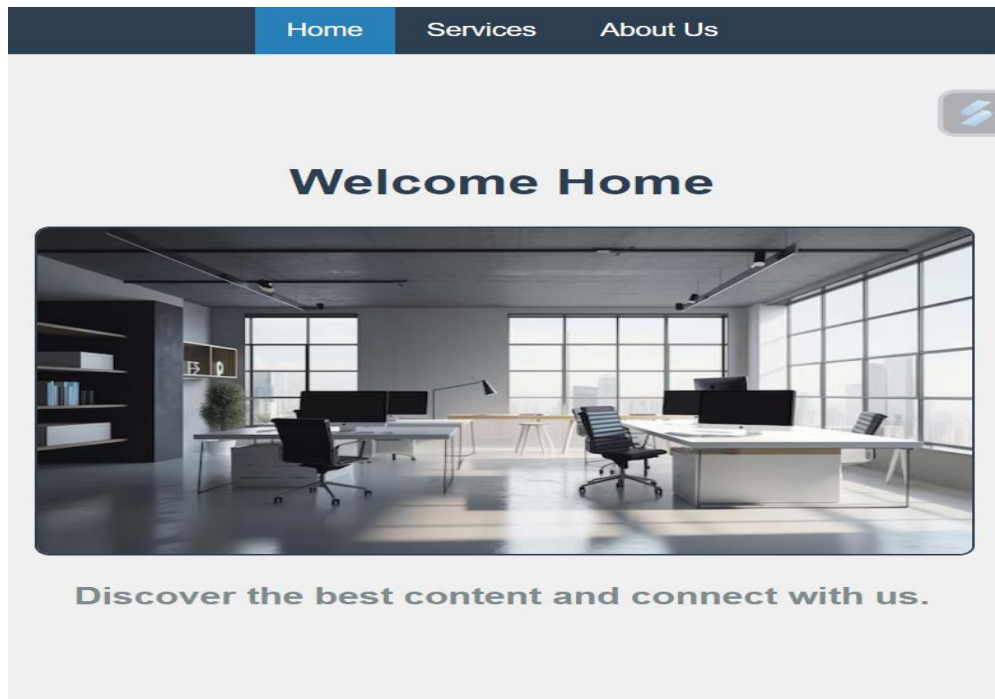
    <script type="text/ng-template" id="pages/services.html">

        <h1>Our Services</h1>
```

```
<h3>{{message}}</h3>
</script>
<script type="text/ng-template" id="pages/aboutus.html">
  <h1>About Us</h1>
  <h3>{{message}}</h3>
</script>
<script>
  var app = angular.module('myApp', ['ngRoute']);
  app.config(function($routeProvider) {
    $routeProvider
      .when('/', {
        templateUrl: 'pages/home.html',
        controller: 'HomeController'
      })
      .when('/services', {
        templateUrl: 'pages/services.html',
        controller: 'ServicesController'
      })
      .when('/aboutus', {
        templateUrl: 'pages/aboutus.html',
        controller: 'AboutUsController'
      })
      .otherwise({redirectTo: '/'});
  });
  app.controller('HomeController', function($scope) {
    $scope.message = 'Discover the best content and connect with us.';
  });
```

```
    app.controller('ServicesController', function($scope) {  
        $scope.message = 'We offer a wide range of services including web  
development, app development, and digital marketing.';  
    });  
    app.controller('AboutUsController', function($scope) {  
        $scope.message = 'We are a team of dedicated professionals committed  
to delivering top-notch solutions.';  
    });  
    // Update active class on navigation links  
    app.run(function($rootScope, $location) {  
        $rootScope.$on('$routeChangeSuccess', function() {  
            var path = $location.path();  
            angular.forEach(document.querySelectorAll('#navbar a'),  
function(link) {  
                if (link.getAttribute('href').substr(1) === path) {  
                    link.classList.add('active');  
                } else {  
                    link.classList.remove('active');  
                }  
            });  
        });  
    });  
    </script>  
</body>  
</html>
```

OUTPUT:



RESULT:

Hence developed a single page application using Angular JS.