

# **FRONT END DESIGN TOOLS & WEB TECHNOLOGIES(IT-455)**

## **PRACTICAL FILE**



**University School of Information, Communication and  
Technology**

**Guru Gobind Singh Indraprastha University, Delhi**

**Submitted by: YASH ARYAN (06816403220)**

**Batch: B. Tech (CSE) 7<sup>th</sup> Semester**

**Submitted to: Dr. SANJAY MALIK**

# Index

Sr. No	Experiment
1	Create a webpage about yourself using basic HTML tags.
2	Write an experiment using advanced HTML tags like creating table etc.
3	Write an experiment for creating a frame
4	Write an experiment for creating forms taking input from the user.
5	Create a webpage using CSS styles.
6	Write an experiment with Java script.
7	Create a webpage using Dynamic HTML.
8	Develop an interactive form using ASP.
9	Develop an interactive form using PHP.
10	Develop a JSP page that includes a simple form with input fields for a name and email address. Upon form submission, display a 'Thank you, [Name]' message on the same page.
11	Using current front end development technologies write an experiment.
12	Using current back end development technologies write an experiment.
13	Create a Java program that reads and modifies an XML document containing a list of employees. Load the XML data, add a new employee, update the information of an existing employee, and save the modified XML back to a file.
14	Write an experiment creating a JavaBean component
15	Develop a Java servlet that is a simple online poll system.
16	Write an experiment using JSP Programming and applications.
17	Keeping program 10 as reference write an experiment using JDBC.
18	Case study using any above program.
19	Database Programming.
20	Create a website of USIC&T, GGSIPU containing all necessary and required information.

## Program No. 1

Create a webpage about yourself using basic HTML tags.

Code:

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <style>
      * {
        margin: 0;
        padding: 0;
        box-sizing: border-box;
      }
      body {
        background-color: rgb(253, 254, 255);
        display: flex;
        justify-content: center;
        align-items: center;
      }
      .full {
        background-image: url("https://img.freepik.com/premium-vector/white-abstract-background-theme_23-2148831659.jpg");
        width: 50%;
        max-width: 1000px;
        min-height: 100px;
        display: grid;
        grid-template-columns: 2fr 4fr;
      }
      .left {
        position: initial;
        margin-right: 100px;
        margin-left: 70px;
      }
      .right {
        position: initial;
      }
      .image,
      .Contact,
      .Skills,
      .Language,
      .Hobbies,
      .title,
      .Summary,
      .Experience,
      .Education,
      .project {
        margin-bottom: 30px;
      }
      .h2 {
```

```

        background-color: rgb(4, 96, 150);
    }
</style>
</head>
<body>
    <div class="full">
        <div class="left">
            <div class="image">
                
            </div>
            <div class="Contact">
                <h2>Contact</h2>
                <p><b>Email id:</b>yasharyan14032003@gmail.com</p>
                <p><b>Mobile no :</b>9958995378</p>
            </div>
            <div class="Skills">
                <h2>Skills</h2>
                <ul>
                    <li><b>Programming Languages : Python, C, C++,
JavaScript</b></li>
                    <li><b>Frontend : HTML5, CSS3, JavaScript, React</b></li>
                    <li><b>Backend : Node.js</b></li>
                </ul>
            </div>
            <div class="Language">
                <h2>Language</h2>
                <ul>
                    <li>English</li>
                    <li>Hindi</li>
                </ul>
            </div>
            <div class="Hobbies">
                <h2>Hobbies</h2>
                <ul>
                    <li>Playing Chess</li>
                    <li>Binge Watching</li>
                    <li>Programming</li>
                </ul>
            </div>
        </div>
        <div class="right">
            <div class="name">
                <h1>Yash Aryan</h1>
            </div>
            <div class="title">
                <p>Final year student</p>
            </div>
            <div class="Summary">
                <h2>Summary</h2>
                <p>Aspiring Software Developer</p>
            </div>
            <div class="Experience">

```

```


<h2>Experience</h2>
<h3>Postman - Postman Student Expert</h3>
<p>March 2021 to July 2021</p>
<ul>
  <li>Actively engaged in web development.</li>
  <li>Designing project & planning</li>
</ul>
<h3>Internshala - Student Partner</h3>
<p>April 2021 to June 2021</p>
<ul>
  <li>Actively engaged in web development.</li>
  <li>Designing project & planing</li>
  <li>Working on designing</li>
</ul>
</div>
<div class="Education">
  <h2>Education</h2>
  <table>
    <tr>
      <th>University/college</th>
      <th>Passing year</th>
      <th>percentage/cgpa</th>
    </tr>
    <tr>
      <td>USICT</td>
      <td>2024</td>
      <td>8.8</td>
    </tr>
    <tr>
      <td>12th</td>
      <td>2019</td>
      <td>95.2%</td>
    </tr>
  </table>
</div>
<div class="project">
  <ul>
    <li>
      <h2>Sorting Visualizer</h2>
      <p>This project is based on React</p>
    </li>
    <li>
      <h2>Open AI Chatbot</h2>
      <p>This project is based on Python</p>
    </li>
  </ul>
</div>
</div>
</div>
</body>
</html>

```

# Output:

index.html x +

File | C:/Users/Hp/OneDrive/Desktop/FEDT%20File/index.html



## Yash Aryan

Final year student

### Contact

Email  
id:yasharyan14032003@gmail.com  
Mobile no :9958995378

### Skills

- Programming Languages :  
Python, C, C++, JavaScript
- Frontend : HTML5, CSS3,  
JavaScript, React
- Backend : Node.js

### Language

- English
- Hindi

### Hobbies

- Playing Chess
- Binge Watching
- Programming

### Summary

Aspiring Software Developer

### Experience

**Postman - Postman Student Expert**  
March 2021 to July 2021

- Actively engaged in web development.
- Designing project & planning

**Internshala - Student Partner**  
April 2021 to June 2021

- Actively engaged in web development.
- Designing project & planing
- Working on designing

### Education

University/college	Passing year	percentage/cgpa
USICT	2024	8.8
12th	2019	95.2%

- **Sorting Visualizer**  
This project is based on React
- **Open AI Chatbot**  
This project is based on Python

## Program No. 2

**Write an experiment using advanced HTML tags like creating table etc.**

### Code:

```
<!DOCTYPE html>
<html>

<head>
  <title>HTML Table</title>
  <style>
    table {
      width: 70%;
      border: 1px solid #ccc;
      border-collapse: collapse;
      margin-left: auto;
      margin-right: auto;
      font-size: 1.8rem;
    }

    th {
      background-color: #f2f2f2;
      padding: 8px;
      text-align: center;
      font-weight: bold;
    }

    td {
      padding: 8px;
      text-align: center;
    }

    h1 {
      text-align: center;
    }
  </style>
</head>

<body>

  <h1>Time Table</h2>

  <table>
    <tr>
      <th>Day/Time</th>
      <th>9:00-11:00</th>
      <th>11:00-13:00</th>
      <th>13:30-15:30</th>
      <th>15:30-17:30</th>
    </tr>
```

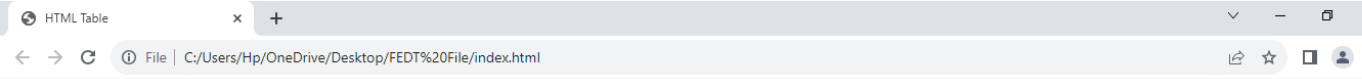
```
<tr>
  <td>Monday</td>
  <td>FEDT</td>
  <td>Software</td>
  <td>AI</td>
  <td>ACN</td>
</tr>
<tr>
  <td>Tuesday</td>
  <td>FEDT</td>
  <td>Software</td>
  <td>AI</td>
  <td>ACN</td>
</tr>
<tr>
  <td>Wednesday</td>
  <td>AI</td>
  <td>Software</td>
  <td>ACN</td>
  <td></td>
</tr>
</table>
```

```
</body>
```

```
</html>
```



# Output:



**Time Table**

Day/Time	9:00-11:00	11:00-13:00	13:30-15:30	15:30-17:30
Monday	FEDT	Software	AI	ACN
Tuesday	FEDT	Software	AI	ACN
Wednesday	AI	Software	ACN	

## Program No. 3

### Write an experiment for creating a frame

#### Code:

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

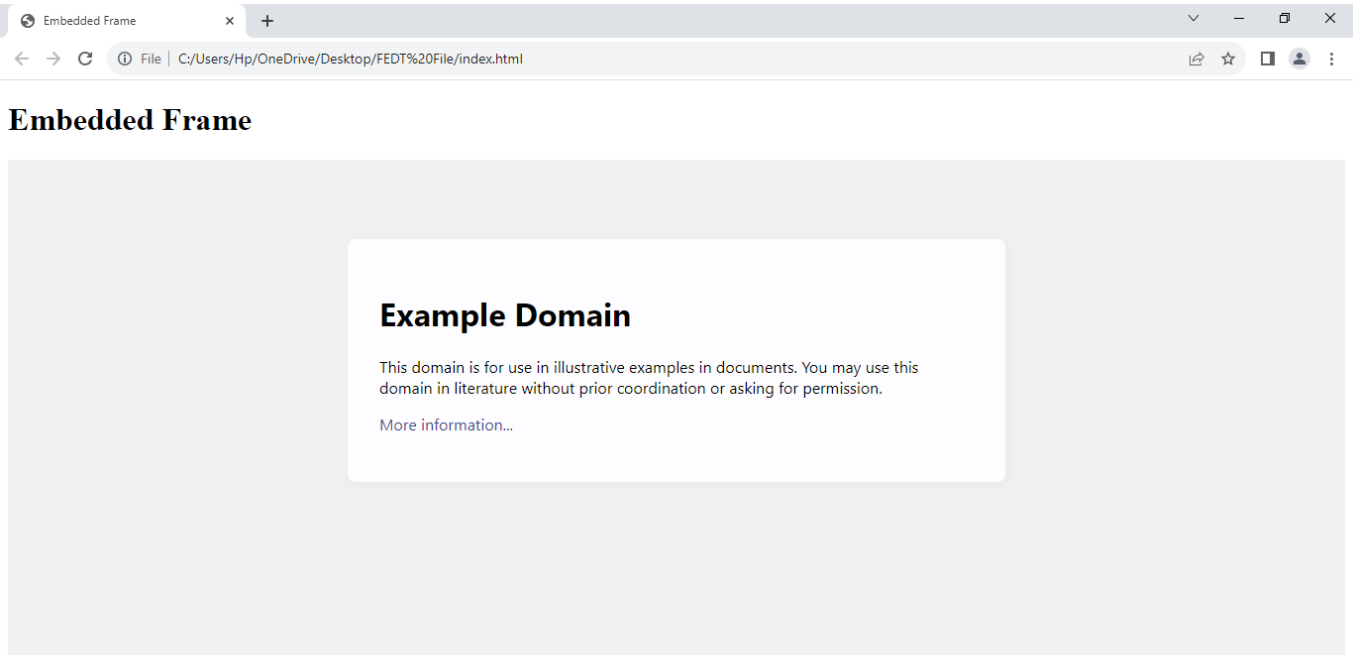
<head>
  <title>Embedded Frame</title>
  <style>
    iframe {
      width: 100%;
      height: 500px;
      border: 0;
    }
  </style>
</head>

<body>
  <h1>Embedded Frame</h1>

  <iframe src="https://www.example.com">
    <p>Your browser does not support iframes.</p>
  </iframe>
</body>

</html>
```

# Output:



## Program No. 4

**Write an experiment for creating forms taking input from the user**

### Code:

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

<head>
  <title>User Input Form</title>
  <style>
    form {
      width: 50%;
      margin: 0 auto;
    }

    label {
      display: block;
      margin-bottom: 5px;
    }

    input {
      width: 100%;
      padding: 5px;
      border: 1px solid #ccc;
    }

    button {
      padding: 5px 10px;
      border: 1px solid #ccc;
      cursor: pointer;
    }

    h1 {
      text-align: center;
    }

    #message {
      width: 100%;
    }
  </style>
</head>

<body>

  <h1>User Input Form</h1>

  <form action="#">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required>
```

```
<label for="email">Email:</label>
<input type="email" id="email" name="email" required>

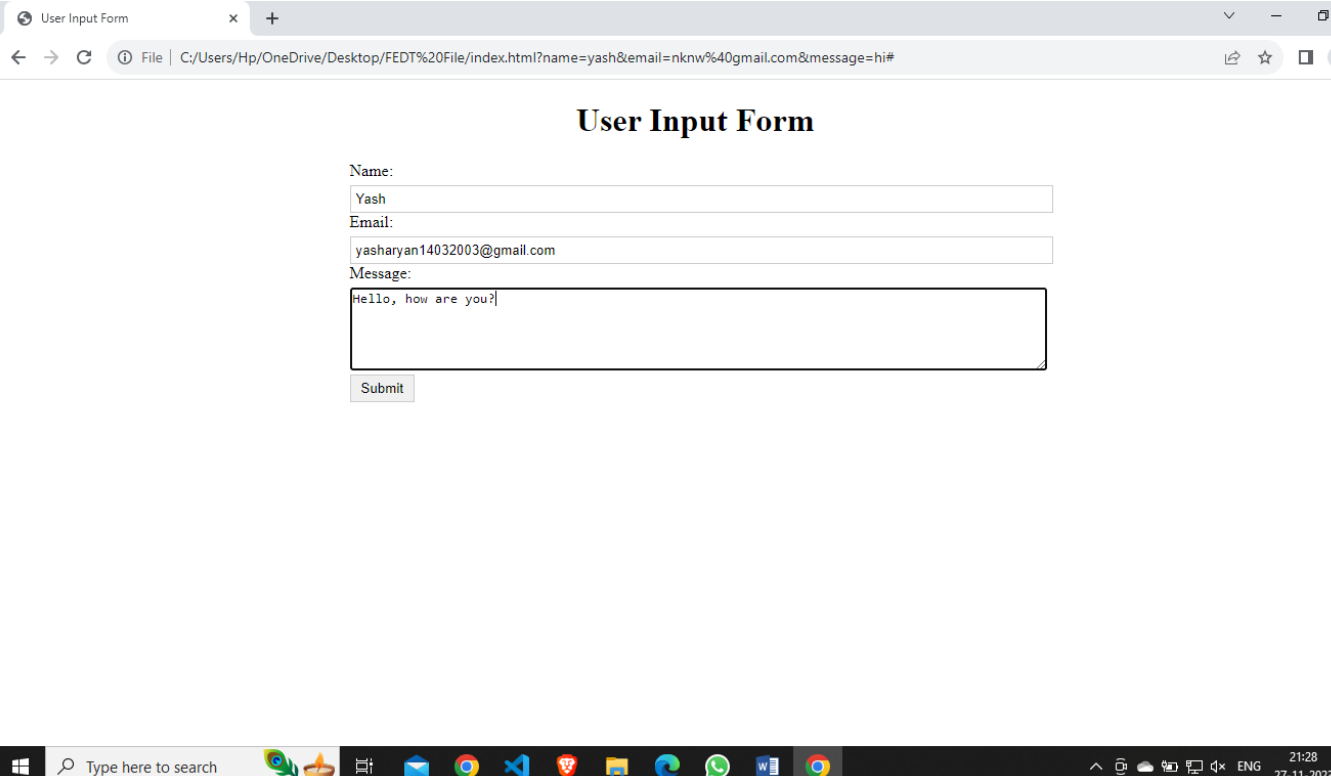
<label for="message">Message:</label>
<textarea id="message" name="message" rows="5" required></textarea>

  <button type="submit">Submit</button>
</form>

</body>

</html>
```

### Output:

The screenshot shows a web browser window with a single tab titled 'User Input Form'. The address bar displays the file path 'C:/Users/Hp/OneDrive/Desktop/FEDT%20File/index.html?name=yash&email=nknw%40gmail.com&message=hi#'. The page content features a large heading 'User Input Form' in a bold, black, serif font. Below the heading, there are three input fields: a text field for 'Name' containing 'Yash', a text field for 'Email' containing 'yasharyan14032003@gmail.com', and a larger text area for 'Message' containing 'Hello, how are you?'. A 'Submit' button is positioned below the message field. The browser's interface includes back, forward, and refresh buttons, as well as icons for file, star, and window management. The Windows taskbar at the bottom shows the search bar, several application icons (including Chrome, VS Code, and a terminal), and the system clock indicating 21:28 on 27-11-2023.

## Program No. 5

## Create a webpage using CSS styles

**Code:**

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

<head>
  <title>Welcome to Nature's Haven</title>
  <style>
    body {
      font-family: sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f2f2f2;
    }

    header {
      background-color: #58d68d;
      padding: 20px;
      text-align: center;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.2);
    }

    h1 {
      font-size: 32px;
      color: white;
      margin-bottom: 20px;
    }

    nav {
      display: flex;
      justify-content: space-between;
      align-items: center;
      padding: 20px;
    }

    nav ul {
      list-style: none;
      padding: 0;
      margin: 0;
    }

    nav li {
      display: inline-block;
      margin-left: 20px;
    }

    nav a {
      text-decoration: none;
      color: #333;

```

```

        font-weight: bold;
    }

    main {
        padding: 20px;
    }

    section.hero {
        background-image: url('https://images.unsplash.com/photo-1549634342507-4257493f1893?ixlib=rb-1.2.1&ixid=MnwxMjM0NDhfMiw&auto=format&fit=crop&w=1000&q=80');
        background-size: cover;
        background-position: center;
        height: 300px;
        text-align: center;
        color: white;
        padding-top: 100px;
    }

    section.hero h2 {
        font-size: 48px;
        font-weight: bold;
    }

    .section-content {
        padding: 20px;
        border: 1px solid #ccc;
        margin-bottom: 20px;
    }

    footer {
        background-color: #58d68d;
        padding: 20px;
        text-align: center;
        box-shadow: 0 -2px 4px rgba(0, 0, 0, 0.2);
    }
</style>
</head>

<body>

    <header>
        <h1>Welcome to Nature's Haven</h1>
    </header>

    <nav>
        <ul>
            <li><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </nav>

    <main>

```



```
<section class="hero">
  <h2>Embark on a journey of tranquility</h2>
  <p>Escape the hustle and bustle of city life and discover the serenity of
nature's embrace.</p>
</section>

<section class="section-content">
  <h2>Explore the wonders of the natural world</h2>
  <p>From towering forests to serene lakes, embark on adventures that will
awaken your senses and reconnect you with
    the beauty of the natural world.</p>
</section>

<section class="section-content">
  <h2>Find peace and tranquility amidst nature's embrace</h2>
  <p>Let the soothing sounds of nature wash over you as you immerse yourself
in the calming atmosphere of forests,
    meadows, and gardens.</p>
</section>

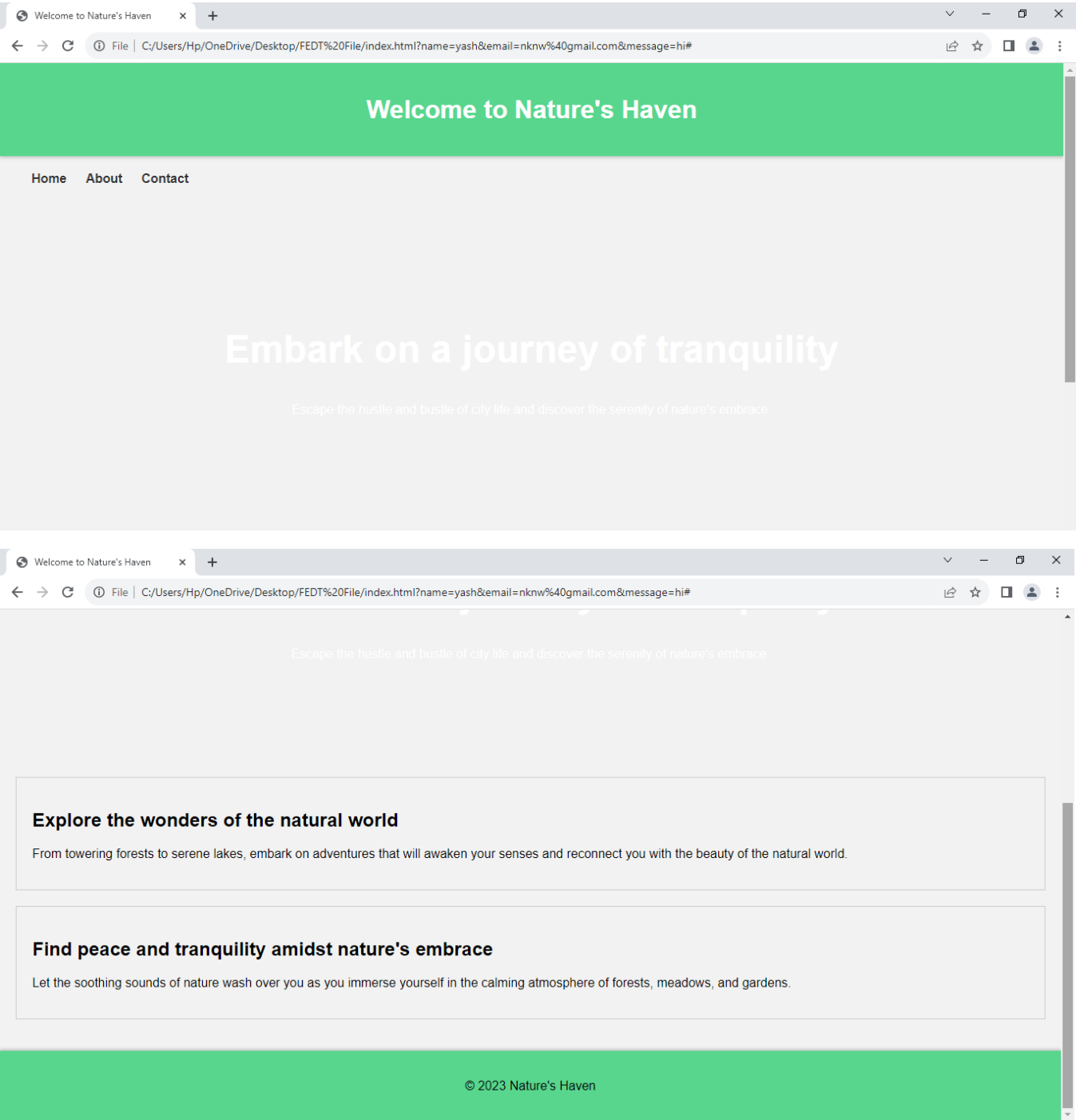
</main>

<footer>
  <p>&copy; 2023 Nature's Haven</p>
</footer>

</body>

</html>
```

# Output:



## Program No. 6

### Write an experiment with Java script

#### Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>JavaScript Calculator</title>
</head>
<body>
  <h1>JavaScript Calculator</h1>

  <p>Enter the first number:</p>
  <input type="number" id="num1" name="num1">

  <p>Enter the second number:</p>
  <input type="number" id="num2" name="num2">

  <button onclick="calculateSum()">Calculate Sum</button>

  <p id="resultText"></p>

  <script>
    function calculateSum() {
      var num1 = parseFloat(document.getElementById('num1').value);
      var num2 = parseFloat(document.getElementById('num2').value);
      var sum = num1 + num2;

      document.getElementById('resultText').innerHTML = "The sum of " + num1 + "
and " + num2 + " is " + sum;
    }
  </script>
</body>
</html>
```

# Output:

JavaScript Calculator

Enter the first number:

Enter the second number:

JavaScript Calculator

Enter the first number:

Enter the second number:

The sum of 2 and 2 is 4

## Program No. 7

### Create a webpage using Dynamic HTML

#### Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Dynamic Shopping List</title>
</head>
<body>
  <h1>To Do List</h1>

  <input type="text" id="newListItem" placeholder="Enter new item">
  <button onclick="addItem()">Add Item</button>

  <ul id="shoppingList"></ul>

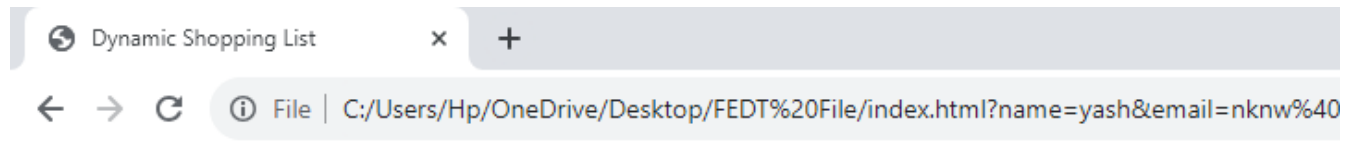
  <script>
    var shoppingList = [];

    function addItem() {
      var newListItemText = document.getElementById('newListItem').value;
      if (newListItemText) {
        shoppingList.push(newListItemText); // Add new item to the array

        // Create and append a new list item element
        var newListItemElement = document.createElement('li');
        newListItemElement.textContent = newListItemText;
        document.getElementById('shoppingList').appendChild(newListItemElement);

        // Clear the input field
        document.getElementById('newListItem').value = '';
      }
    }
  </script>
</body>
</html>
```

## Output:



## To Do List

- Eat
- Sleep
- Code
- Repeat

## Program No. 8

### Develop an interactive form using ASP

#### Code:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site.Master"
AutoEventWireup="true" CodeBehind="std_reg.aspx.cs"
Inherits="Student_Registration.std_reg" %>
<asp:Content ID="Content1" ContentPlaceHolderID="MainContent" runat="server">

    <section id="main-content">
        <section class="wrapper">
            <div class="row">
                <div class="col-lg-12">
                    <section class="panel">
                        <header class="panel-heading">
                            <div class="col-md-4 col-md-offset-4">
                                <h1>Student Registration</h1>
                            </div>
                        </header>
                        <div class="panel-body">
                            <div class="row">
                                <div class="col-md-4 col-md-offset-1">
                                    <div class="form-group">
                                        <asp:Label runat="server"
AssociatedControlID="txtStdname"><b>Student Name</b></asp:Label><br />
                                        <asp:TextBox runat="server"
required="required" Enabled="True" name="BrandName" ID="txtStdname" class="form-
control input-sm" placeholder="Student Name"></asp:TextBox>
                                    </div>
                                </div>
                                <div class="col-md-4 col-md-offset-1">
                                    <div class="form-group">
                                        <asp:Label runat="server"
AssociatedControlID="txtfname"><b>Father Name</b></asp:Label><br />
                                        <asp:TextBox runat="server"
required="required" Enabled="True" name="BrandName" ID="txtfname" class="form-
control input-sm" placeholder="Father Name"></asp:TextBox>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </section>
                </div>
            </div>
        </section>
    </div>
    <div class="row">
        <div class="col-md-4 col-md-offset-1">
            <div class="form-group">
                <asp:Label runat="server"
AssociatedControlID="txtdob"><b>DOB</b></asp:Label><br />
```

```

        <asp:TextBox runat="server"
required="required" TextMode="Date" Enabled="True" name="BrandName" ID="txtdob"
class="form-control input-sm" placeholder="DOB "></asp:TextBox>
    </div>
</div>
    <div class="col-md-4 col-md-offset-1">
        <div class="form-group">
            <asp:Label runat="server"
AssociatedControlID="txtprogram" ><b>Program</b></asp:Label><br />
            <asp:TextBox runat="server"
required="required" Enabled="True" name="BrandName" ID="txtprogram" class="form-
control input-sm" placeholder="Program"></asp:TextBox>
        </div>
    </div>
</div>
<div class="row">
    <div class="col-md-4 col-md-offset-1">
        <div class="form-group">
            <asp:Label runat="server"
AssociatedControlID="txtregion"><b>Region</b></asp:Label><br />
            <asp:DropDownList ID="txtregion"
CssClass="form-control input-sm" runat="server" >
                <asp:ListItem Text="India" />
                <asp:ListItem Text="USA" />
                <asp:ListItem Text="Canada" />
                <asp:ListItem Text="Australia" />
                <asp:ListItem Text="Spain" />
                <asp:ListItem Text="Argentina" />
            </asp:DropDownList>
        </div>
    </div>
    <div class="col-md-4 col-md-offset-1">
        <div class="form-group">
            <asp:Label runat="server"
AssociatedControlID="txtaddress"><b>Address</b></asp:Label><br />
            <asp:TextBox runat="server"
required="required" Enabled="True" name="BrandName" ID="txtaddress" class="form-
control input-sm" placeholder=" Address "></asp:TextBox>
        </div>
    </div>
</div>
</div>
<div class="row">
    <div class="col-md-4 col-md-offset-1">
        <div class="form-group">
            <asp:Label runat="server"
AssociatedControlID="txtcell"><b>Cell No</b></asp:Label><br />
            <asp:TextBox runat="server"
required="required" TextMode="Number" Enabled="True" name="BrandName" ID="txtcell"
class="form-control input-sm" placeholder="Cell Number "></asp:TextBox>
        </div>
    </div>

```



```

        </div>
        <div class="col-md-4 col-md-offset-1">
        <div class="form-group">
            <asp:Label runat="server"
AssociatedControlID="txtgender" ><b>Gender</b></asp:Label><br />
            <asp:RadioButtonList CssClass="form-group"
ID="txtgender" runat="server">
                <asp:ListItem Text="Male" />
                <asp:ListItem Text="Female" />
            </asp:RadioButtonList>

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

    <div class="row">
        <div class="col-md-10 col-md-offset-1">
        <div class="form-group">
            <asp:Button Text="Save" ID="btnsave"
OnClick="btnsave_Click" CssClass="btn btn-primary btn-lg" Width="220px"
runat="server" />

            <asp:Button Text="Update"
ID="btnupdate" OnClick="btnupdate_Click" CssClass="btn btn-primary btn-lg"
Width="220px" runat="server" />

            <asp:Button Text="Delete"
ID="btndlt" OnClick="btndlt_Click" CssClass="btn btn-danger btn-lg"
Width="220px" runat="server" />

            <asp:Label Text="" ForeColor="Green" Font-
Bold="true" ID="lblmessage" CssClass="label label-success" runat="server" />
        </div>
        </div>
    </div>

    </div>
    <div class="row">
        <div class="col-md-10 col-md-offset-1">
        <div class="form-group">
            <div class="table-responsive">
                <asp:GridView ID="gv" Width="100%"
AutoGenerateSelectButton="true" SelectedRowStyle-BackColor="Green"
OnSelectedIndexChanged="gv_SelectedIndexChanged" runat="server"></asp:GridView>
            </div>
        </div>
        </div>
    </div>
</div>
</section>
</div>
</div>
</section>
</section>

</asp:Content>

```

# Output:

Application name

Home

About

Contact

Student Registration

Student Registration

Student Name

Student Name

I

DOB

mm/dd/yyyy

Region

Cell No

Student Name

Father Name

Father Name

Program

Program

Address

Address

Program

Male

Female

Save

Update

Delete

## Program No. 9

### Develop an interactive form using PHP

#### Code:

*Index.php*

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="bootstrap.css">
    <title>Contact Us Form In Php</title>
</head>
<body>

    <div class="container">
        <div class="row">
            <div class="col-lg-6 m-auto">
                <div class="card mt-5">
                    <div class="card-title">
                        <h2 class="text-center py-2"> Contact Us </h2>
                        <hr>
                        <?php
                            $Msg = "";
                            if(isset($_GET['error']))
                            {
                                $Msg = " Please Fill in the Blanks ";
                                echo '<div class="alert alert-
danger">'. $Msg. '</div>';
                            }

                            if(isset($_GET['success']))
                            {
                                $Msg = " Your Message Has Been Sent ";
                                echo '<div class="alert alert-
success">'. $Msg. '</div>';
                            }

                        ?>
                    </div>
                    <div class="card-body">
                        <form action="process.php" method="post">
                            <input type="text" name="UName" placeholder="User
Name" class="form-control mb-2">
                            <input type="email" name="Email" placeholder="Email"
class="form-control mb-2">
                            <input type="text" name="Subject"
placeholder="Subject" class="form-control mb-2">
```

```

        <textarea name="msg" class="form-control mb-2"
placeholder="Write The Message"></textarea>
        <button class="btn btn-success" name="btn-send"> Send
</button>

    </form>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

### *Process.php*

```

<?php

if(isset($_POST['btn-send']))
{
    $UserName = $_POST['UName'];
    $Email = $_POST['Email'];
    $Subject = $_POST['Subject'];
    $Msg = $_POST['msg'];

    if(empty($UserName) || empty($Email) || empty($Subject) || empty($Msg))
    {
        header('location:index.php?error');
    }
    else
    {
        $to = "admin@onlineittuts.com";

        if(mail($to,$Subject,$Msg,$Email))
        {
            header("location:index.php?success");
        }
    }
}
else
{
    header("location:index.php");
}

```

# Output:

---

Contact Us

User Name

Email

Subject

Write The Message

Send

## Program No. 10

Develop a JSP page that includes a simple form with input fields for a name and email address. Upon form submission, display a 'Thank you, [Name]' message on the same page

Code:

### form.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Person Form</h1>
<form method="post" action="person.jsp">
First Name: <input type="text" name="firstName">
<br>
Last Name: <input type="text" name="lastName">
<br>
Email: <input type="text" name="email">
<br>
<Button type="submit" name="submit">Submit</Button>
</form>
</body>
</html>
```

### mrbean.java

```
package netrox;

public class mrbean {
    private String firstName;
    private String lastName;
    private String email;
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public String getEmail() {
        return email;
    }
}
```

```

    }
    public void setEmail(String email) {
        this.email = email;
    }
}

```

## person.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Person Bean Test</title>
</head>
<body>
<%
if("POST".equalsIgnoreCase(request.getMethod()) &&
request.getParameter("submit")!=null)
{
%>
<jsp:useBean id="mrbean" class="netrox.mrbean" scope="session">
<jsp:setProperty name="mrbean" property="*/>
</jsp:useBean>
<%
out.println("Thank You"+ mrbean.getFirstName()+ " "+ mrbean.getLastName());

}
%>
</body>
</html>

```

# Output:

Lab Experiments.pdf

Insert title here

localhost:8080/buzz/form.jsp

Person Form

First Name: Yash

Last Name: Aryan

Email: yasharyan14032003@gmail.

Submit

Lab Experiments.pdf

Person Bean Test

localhost:8080/buzz/person.jsp

Thank You Yash Aryan



## Program No. 11

Using current front end development technologies write an experiment

Code:

index.html:

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

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <title>Analog Clock</title>
</head>

<body>
  <div class="clock">
    <div class="hour hand" id="hour"></div>
    <div class="minute hand" id="minute"></div>
    <div class="seconds hand" id="seconds"></div>
    
  </div>

  <script src="js/script.js"></script>
</body>

</html>
```

script.js

```
let hour = document.getElementById("hour");
let minute = document.getElementById("minute");
let seconds = document.getElementById("seconds");

let set_clock = setInterval(() => {
  let date_now = new Date();

  let hr = date_now.getHours();
  let min = date_now.getMinutes();
  let sec = date_now.getSeconds();

  let calc_hr = hr * 30 + min / 2;
  let calc_min = min * 6 + sec / 10;
  let calc_sec = sec * 6;

  hour.style.transform = `rotate(${calc_hr}deg)`;
```

```
minute.style.transform = `rotate(${calc_min}deg)`;  
seconds.style.transform = `rotate(${calc_sec}deg)`;  
}, 1000);
```

## style.css

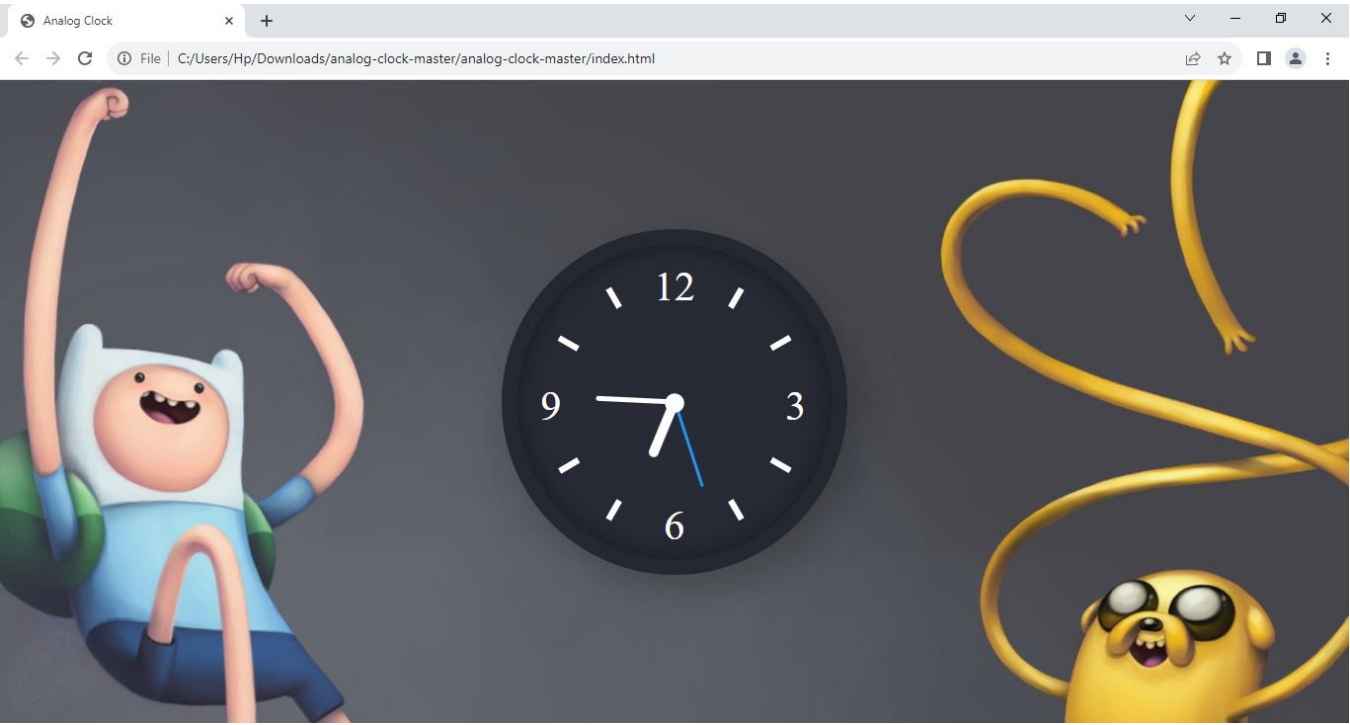
```
* {  
  padding: 0;  
  margin: 0;  
  box-sizing: border-box;  
}  
  
body {  
  height: 100vh;  
  background-image:   
url("https://media.moddb.com/images/groups/1/14/13495/213443.jpg");  
  background-size: cover;  
}  
  
.clock {  
  background: #282a35;  
  height: 320px;  
  width: 320px;  
  position: absolute;  
  transform: translate(-50%, -50%);  
  top: 50%;  
  left: 50%;  
  box-sizing: content-box;  
  border-radius: 50%;  
  border: 15px solid #242931;  
  box-shadow: 15px 15px 35px rgba(0, 0, 0, 0.2),  
    inset 0 0 30px rgba(0, 0, 0, 0.4);  
}  
  
img {  
  position: relative;  
}  
  
.hand {  
  position: absolute;  
  background-color: #ffffff;  
  margin: auto;  
  left: 0;  
  right: 0;  
  border-radius: 5px;  
  transform-origin: bottom;  
}  
  
.hour {
```

```
    height: 60px;
    width: 10px;
    top: 100px;
}

.minute {
    height: 80px;
    width: 5px;
    top: 80px;
}

.seconds {
    height: 90px;
    width: 3px;
    top: 70px;
    background-color: #2196f3;
}
```

# Output:



## Program No. 12

**Using current back end development technologies write an experiment**

**Code:**

**Demo.jsx**

```
import React, { useState, useEffect } from "react";

import { copy, linkIcon, loader, tick } from "../assets";
import { useLazyGetSummaryQuery } from "../services/article";

const Demo = () => {
  const [article, setArticle] = useState({
    url: "",
    summary: "",
  });
  const [allArticles, setAllArticles] = useState([]);
  const [copied, setCopied] = useState("");

  // RTK lazy query
  const [getSummary, { error, isFetching }] = useLazyGetSummaryQuery();

  // Load data from localStorage on mount
  useEffect(() => {
    const articlesFromLocalStorage = JSON.parse(
      localStorage.getItem("articles")
    );

    if (articlesFromLocalStorage) {
      setAllArticles(articlesFromLocalStorage);
    }
  }, []);

  const handleSubmit = async (e) => {
    e.preventDefault();

    const existingArticle = allArticles.find(
      (item) => item.url === article.url
    );

    if (existingArticle) return setArticle(existingArticle);

    const { data } = await getSummary({ articleUrl: article.url });
    if (data?.summary) {
      const newArticle = { ...article, summary: data.summary };
      const updatedAllArticles = [newArticle, ...allArticles];

      // update state and local storage
      setArticle(newArticle);
```

```

        setAllArticles(updatedAllArticles);
        localStorage.setItem("articles", JSON.stringify(updatedAllArticles));
    }
};

// copy the url and toggle the icon for user feedback
const handleCopy = (copyUrl) => {
    setCopied(copyUrl);
    navigator.clipboard.writeText(copyUrl);
    setTimeout(() => setCopied(false), 3000);
};

const handleKeyDown = (e) => {
    if (e.keyCode === 13) {
        handleSubmit(e);
    }
};

return (
    <section className='mt-16 w-full max-w-xl'>
        { /* Search */ }
        <div className='flex flex-col w-full gap-2'>
            <form
                className='relative flex justify-center items-center'
                onSubmit={handleSubmit}
            >
                <img
                    src={linkIcon}
                    alt='link-icon'
                    className='absolute left-0 my-2 ml-3 w-5'
                />

                <input
                    type='url'
                    placeholder='Paste the article link'
                    value={article.url}
                    onChange={(e) => setArticle({ ...article, url: e.target.value })}
                    onKeyDown={handleKeyDown}
                    required
                    className='url_input peer'
                />
                <button
                    type='submit'
                    className='submit_btn peer-focus:border-gray-700 peer-focus:text-gray-
700 '
                >
                    <p>↵</p>
                </button>
            </form>

            { /* Browse History */ }
            <div className='flex flex-col gap-1 max-h-60 overflow-y-auto'>
                {allArticles.reverse().map((item, index) => (
                    <div
                        key={`link-${index}`}

```

```

      onClick={() => setArticle(item)}
      className='link_card'
    >
      <div className='copy_btn' onClick={() => handleCopy(item.url)}>
        <img
          src={copied === item.url ? tick : copy}
          alt={copied === item.url ? "tick_icon" : "copy_icon"}
          className='w-[40%] h-[40%] object-contain'
        />
      </div>
      <p className='flex-1 font-satoshi text-blue-700 font-medium text-sm
truncate'>
        {item.url}
      </p>
    </div>
  )))
</div>
</div>

{/* Display Result */}
<div className='my-10 max-w-full flex justify-center items-center'>
  {isFetching ? (
    <img src={loader} alt='loader' className='w-20 h-20 object-contain' />
  ) : error ? (
    <p className='font-inter font-bold text-black text-center'>
      Well, that wasn't supposed to happen...
    <br />
    <span className='font-satoshi font-normal text-gray-700'>
      {error?.data?.error}
    </span>
    </p>
  ) : (
    article.summary && (
      <div className='flex flex-col gap-3'>
        <h2 className='font-satoshi font-bold text-gray-600 text-xl'>
          Article <span className='blue_gradient'>Summary</span>
        </h2>
        <div className='summary_box'>
          <p className='font-inter font-medium text-sm text-gray-700'>
            {article.summary}
          </p>
        </div>
      </div>
    )
  )}
</div>
</section>
);
};

```

export default Demo;

## Hero.jsx

```

import React from "react";

import { logo } from "../assets";

const Hero = () => {
  return (
    <header className='w-full flex justify-center items-center flex-col'>
      <nav className='flex justify-between items-center w-full mb-10 pt-3'>
        <img src={logo} alt='sumz_logo' className='w-28 object-contain' />

        <button
          type='button'
          onClick={() =>
            window.open("https://github.com/YashAryanTheCoder/Text-Summarizer-using-AI", "_blank")
          }
          className='black_btn'
        >
          GitHub
        </button>
      </nav>

      <h1 className='head_text'>
        Summarize Articles with <br className='max-md:hidden' />
        <span className='orange_gradient '>OpenAI GPT-4</span>
      </h1>
      <h2 className='desc'>
        This project was created by Yash Aryan for USICT
        as an open-source article summarizer
        to transform lengthy articles into clear and concise summaries.
      </h2>

    </header>
  );
};

```

```
export default Hero;
```

## article.js

```

import { createApi, fetchBaseQuery } from '@reduxjs/toolkit/query/react'

const rapidApiKey = import.meta.env.VITE_RAPID_API_ARTICLE_KEY;

export const articleApi = createApi({
  reducerPath: 'articleApi',
  baseQuery: fetchBaseQuery({
    baseUrl: 'https://article-extractor-and-summarizer.p.rapidapi.com/',
    prepareHeaders: (headers) => {
      headers.set('X-RapidAPI-Key', rapidApiKey);
      headers.set('X-RapidAPI-Host', 'article-extractor-and-summarizer.p.rapidapi.com');
    }
  })
});

```



```

        return headers;
    },
 )),
  endpoints: (builder) => ({
    getSummary: builder.query({
      query: (params) =>
`summarize?url=${encodeURIComponent(params.articleUrl)}&length=3`,
    }),
  }),
})

export const { useLazyGetSummaryQuery } = articleApi

```

## store.js

```

import { configureStore } from "@reduxjs/toolkit";

import { articleApi } from "./article";

export const store = configureStore({
  reducer: {
    [articleApi.reducerPath]: articleApi.reducer,
  },
  middleware: (getDefaultMiddleware) =>
getDefaultMiddleware().concat(articleApi.middleware)
})

```

## App.css

```

@tailwind base;
@tailwind components;
@tailwind utilities;

.main {
  width: 100vw;
  min-height: 100vh;
  position: fixed;
  display: flex;
  justify-content: center;
  padding: 120px 24px 160px 24px;
  pointer-events: none;
}

.main:before {
  background: radial-gradient(circle, rgba(2, 0, 36, 0) 0, #fafafa 100%);
  position: absolute;
  content: "";
  z-index: 2;
  width: 100%;
  height: 100%;
  top: 0;
}

```

```

.main:after {
  content: "";
  background-image: url("/src/assets/grid.svg");
  z-index: 1;
  position: absolute;
  width: 100%;
  height: 100%;
  top: 0;
  opacity: 0.4;
  filter: invert(1);
}

.gradient {
  height: fit-content;
  z-index: 3;
  width: 100%;
  max-width: 640px;
  background-image: radial-gradient(
    at 27% 37%,
    hsla(215, 98%, 61%, 1) 0px,
    transparent 0%
  ),
  radial-gradient(at 97% 21%, hsla(125, 98%, 72%, 1) 0px, transparent 50%),
  radial-gradient(at 52% 99%, hsla(354, 98%, 61%, 1) 0px, transparent 50%),
  radial-gradient(at 10% 29%, hsla(256, 96%, 67%, 1) 0px, transparent 50%),
  radial-gradient(at 97% 96%, hsla(38, 60%, 74%, 1) 0px, transparent 50%),
  radial-gradient(at 33% 50%, hsla(222, 67%, 73%, 1) 0px, transparent 50%),
  radial-gradient(at 79% 53%, hsla(343, 68%, 79%, 1) 0px, transparent 50%);
  position: absolute;
  content: "";
  width: 100%;
  height: 100%;
  filter: blur(100px) saturate(150%);
  top: 80px;
  opacity: 0.15;
}

@media screen and (max-width: 640px) {
  .main {
    padding: 0;
  }
}

/* Tailwind Styles */

.app {
  @apply relative z-10 flex justify-center items-center flex-col max-w-7xl mx-auto sm:px-16 px-6;
}

.black_btn {
  @apply rounded-full border border-black bg-black py-1.5 px-5 text-sm text-white transition-all hover:bg-white hover:text-black;
}

```

```

}

.head_text {
  @apply mt-5 text-5xl font-extrabold leading-[1.15] text-black sm:text-6xl text-center;
}

.orange_gradient {
  @apply bg-gradient-to-r from-amber-500 via-orange-600 to-yellow-500 bg-clip-text text-transparent;
}

.desc {
  @apply mt-5 text-lg text-gray-600 sm:text-xl text-center max-w-2xl;
}

.url_input {
  @apply block w-full rounded-md border border-gray-200 bg-white py-2.5 pl-10 pr-12 text-sm shadow-lg font-satoshi font-medium focus:border-black focus:outline-none focus:ring-0;
}

.submit_btn {
  @apply hover:border-gray-700 hover:text-gray-700 absolute inset-y-0 right-0 my-1.5 mr-1.5 flex w-10 items-center justify-center rounded border border-gray-200 font-sans text-sm font-medium text-gray-400;
}

.link_card {
  @apply p-3 flex justify-start items-center flex-row bg-white border border-gray-200 gap-3 rounded-lg cursor-pointer;
}

.copy_btn {
  @apply w-7 h-7 rounded-full bg-white/10 shadow-[inset_10px_-50px_94px_0_rgb(199,199,199,0.2)] backdrop-blur flex justify-center items-center cursor-pointer;
}

.blue_gradient {
  @apply font-black bg-gradient-to-r from-blue-600 to-cyan-600 bg-clip-text text-transparent;
}

.summary_box {
  @apply rounded-xl border border-gray-200 bg-white/20 shadow-[inset_10px_-50px_94px_0_rgb(199,199,199,0.2)] backdrop-blur p-4;
}

```

## App.jsx

```
import Hero from "../components/Hero";
```

```
import Demo from "../components/Demo";

import "../App.css";

const App = () => {
  return (
    <main>
      <div className='main'>
        <div className='gradient' />
      </div>

      <div className='app'>
        <Hero />
        <Demo />
      </div>
    </main>
  );
};
```

```
export default App;
```

## main.jsx

```
import React from "react";
import ReactDOM from "react-dom/client";
import { Provider } from "react-redux";

import App from "../App";
import { store } from "../services/store";
```


```
ReactDOM.createRoot(document.getElementById("root")).render(
  <React.StrictMode>
    <Provider store={store}>
      <App />
    </Provider>
  </React.StrictMode>
);
```

## index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link
      href="https://api.fontshare.com/v2/css?f[]=satoshi@1,900,700,500,300,400&display=swap"
      rel="stylesheet"
    />
    <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=Inter:wght@100;200;300;400;500;600;700;800;900&display=swap"
    rel="stylesheet"
  />
  <title>OpenAI Article Summarizer</title>
</head>
<body>
  <div id="root"></div>
  <script type="module" src="/src/main.jsx"></script>
</body>
</html>
```

## Output:

 **Sumz**

[GitHub](#)

# Summarize Articles with OpenAI GPT-4

This project was created by Yash Aryan for USICT as an open-source article summarizer to transform lengthy articles into clear and concise summaries.

C=>


https://medium.com/@MediumStaff/list/the-new-chatbots-chatgpt-bard-and-

✓

https://medium.com/@MediumStaff/list/the-new-chatbots-chatgpt-bard-and-b...

### Article Summary

This text discusses various aspects of large language models (LLMs) and their applications. It mentions the popularity of ChatGPT, a system developed by OpenAI, and how Google has introduced its own version called Bard to compete in this space. The text emphasizes that while LLMs have their limitations and

 **Sumz**


[GitHub](#)

# Summarize Articles with OpenAI GPT-4

This project was created by Yash Aryan for USICT as an open-source article summarizer to transform lengthy articles into clear and concise summaries.

C=>

https://medium.com/@MediumStaff/list/the-new-chatbots-chatgpt-bard-and-



# Summarize Articles with OpenAI GPT-4

This project was created by Yash Aryan for USICT as an open-source article summarizer to transform lengthy articles into clear and concise summaries.

➔ <https://medium.com/@MediumStaff/list/the-new-chatbots-chatgpt-bard-and->

🔗 <https://medium.com/@MediumStaff/list/the-new-chatbots-chatgpt-bard-and-b...>

## Article Summary

This text discusses various aspects of large language models (LLMs) and their applications. It mentions the popularity of ChatGPT, a system developed by OpenAI, and how Google has introduced its own version called Bard to compete in this space. The text emphasizes that while LLMs have their limitations and

🔗 <https://medium.com/@MediumStaff/list/the-new-chatbots-chatgpt-bard-and-b...>

## Article Summary

This text discusses various aspects of large language models (LLMs) and their applications. It mentions the popularity of ChatGPT, a system developed by OpenAI, and how Google has introduced its own version called Bard to compete in this space. The text emphasizes that while LLMs have their limitations and may not be perfect, they can still be valuable tools for problem-solving and generating innovative solutions. It also highlights the use of LLMs in generating questions for events and the potential for LLMs to provide insights and echo back knowledge that humans have already discovered. The text further delves into the technical aspects of LLMs, such as the self-attention mechanism that enables training of models like GPT-3 and the technique of Reinforcement Learning From Human Feedback. It mentions the ease of accessing the power of GPT-4 through the OpenAI API and provides steps for developers to interact with the model using Python. The text also mentions that ChatGPT is trained on dialogue datasets, including demonstration data, to provide expected responses. The limitations of LLMs are also discussed, with examples given of situations where ChatGPT may not perform well, such as handling client work or solving coding problems that are not commonly encountered. The text also touches upon the issue of bias in LLMs, specifically mentioning the higher occurrence of toxic prompt completions when referring to LGBTQ+ groups. Finally, the text mentions the anticipated growth of the AI market due to the implementation and development of fundamental models and the availability of significant capital investment in this field.

## Program No. 13

**Create a Java program that reads and modifies an XML document containing a list of employees. Load the XML data, add a new employee, update the information of an existing employee, and save the modified XML back to a file**

**Code:**

### **Person.java**

```
package mygroup.myproject.model;

public class Person {
    String firstName;
    String lastName;
    int age;
    Address address;
    String[] cars;
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
}
```



```

        public Address getAddress() {
            return address;
        }
        public void setAddress(Address address) {
            this.address = address;
        }
        public String[] getCars() {
            return cars;
        }
        public void setCars(String[] cars) {
            this.cars = cars;
        }
    }
}

```

## Address.java

```

package mygroup.myproject.model;

public class Address {
    String street;
    String city;
    String zip;
    public Address() {}
    public Address(String street, String city, String zip) {
        super();
        this.street = street;
        this.city = city;
        this.zip = zip;
    }
    public String getStreet() {
        return street;
    }
    public void setStreet(String street) {
        this.street = street;
    }
    public String getCity() {

```

```

        return city;
    }
    public void setCity(String city) {
        this.city = city;
    }
    public String getZip() {
        return zip;
    }
    public void setZip(String zip) {
        this.zip = zip;
    }
}

```

## **pom.xml**

```

<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>mygroup</groupId>
    <artifactId>myproject</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <packaging>jar</packaging>

    <name>myproject</name>
    <url>http://maven.apache.org</url>

    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>
    <dependency>

```

```

        <groupId>com.fasterxml.jackson.dataformat</groupId>
        <artifactId>jackson-dataformat-xml</artifactId>
        <version>2.9.7</version>
    </dependency>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
    </dependency>
</dependencies>
</project>

```

## App.java

```

package mygroup.myproject;

import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.util.List;

import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.fasterxml.jackson.dataformat.xml.XmlMapper;

import mygroup.myproject.model.Address;
import mygroup.myproject.model.Person;

/**
 * Hello world!
 *
 */
public class App
{
    public static void main( String[] args )

```

```

{
    try {
        ObjectMapper mapper = new XmlMapper();

        InputStream inputStream = new FileInputStream(new
File("/home/parallels/demo/persons.xml"));

        TypeReference<List<Person>> typeReference = new
TypeReference<List<Person>>() {};

        List<Person> persons = mapper.readValue(inputStream,
typeReference);

        for(Person p :persons) {
            System.out.println("name is "+p.getFirstName()+" city is
"+p.getAddress().getCity()+" first car is "+p.getCars()[0]+" age is
"+p.getAge());
        }

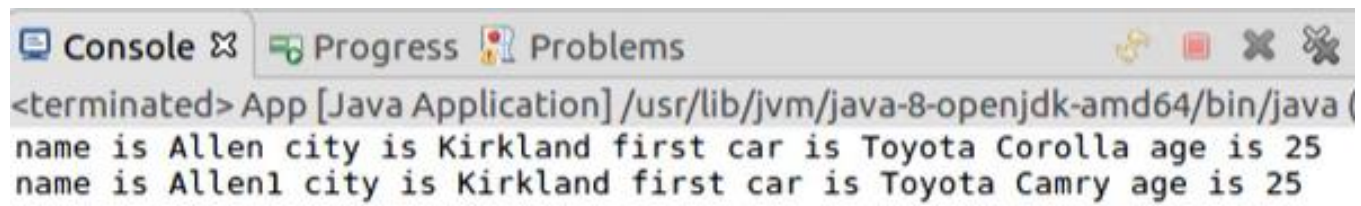
        Person person = new Person();
        person.setFirstName("Jack");
        person.setLastName("Ryan");
        person.setAge(29);
        person.setAddress(new Address("12 Street", "Kirkland", "98052"));
        person.setCars(new String[] {"Toyota Camary", "Honda Accord"});

        mapper.writeValue(new File("/home/parallels/demo/personsOut.xml"),
person);

        inputStream.close();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
}

```

## Output:



The screenshot shows a console window with three tabs: 'Console', 'Progress', and 'Problems'. The 'Console' tab is active and displays the following text:

```
<terminated> App [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (
name is Allen city is Kirkland first car is Toyota Corolla age is 25
name is Allen1 city is Kirkland first car is Toyota Camry age is 25
```

## Program No. 14

### Write an experiment creating a JavaBean component

#### Code:

##### form.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Person Form</h1>
<form method="post" action="person.jsp">
First Name: <input type="text" name="firstName">
<br>
Last Name: <input type="text" name="lastName">
<br>
Email: <input type="text" name="email">
<br>
<Button type="submit" name="submit">Submit</Button>
</form>
</body>
</html>
```

##### mrbean.java

```
package netrox;

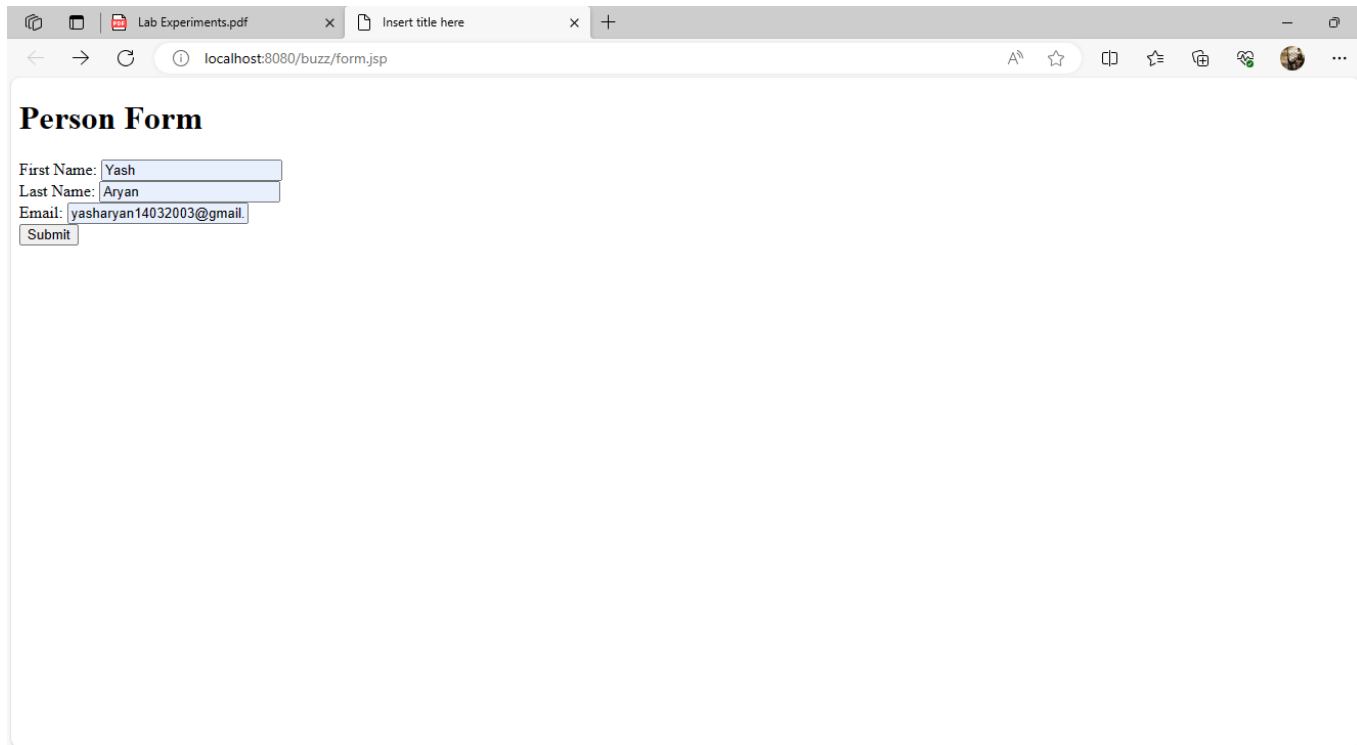
public class mrbean {
    private String firstName;
    private String lastName;
    private String email;
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
}
```

```
}
```

## person.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Person Bean Test</title>
</head>
<body>
<%
if("POST".equalsIgnoreCase(request.getMethod()) &&
request.getParameter("submit")!=null)
{
%>
<jsp:useBean id="mrbean" class="netrox.mrbean" scope="session">
<jsp:setProperty name="mrbean" property="*" />
</jsp:useBean>
<%
out.println("First Name:" + mrbean.getFirstName()+"<br>");
out.println("Last Name:" + mrbean.getLastName()+"<br>");
out.println("Email:" + mrbean.getEmail());
}
%>
</body>
</html>
```

# Output:



Lab Experiments.pdf x Insert title here x +

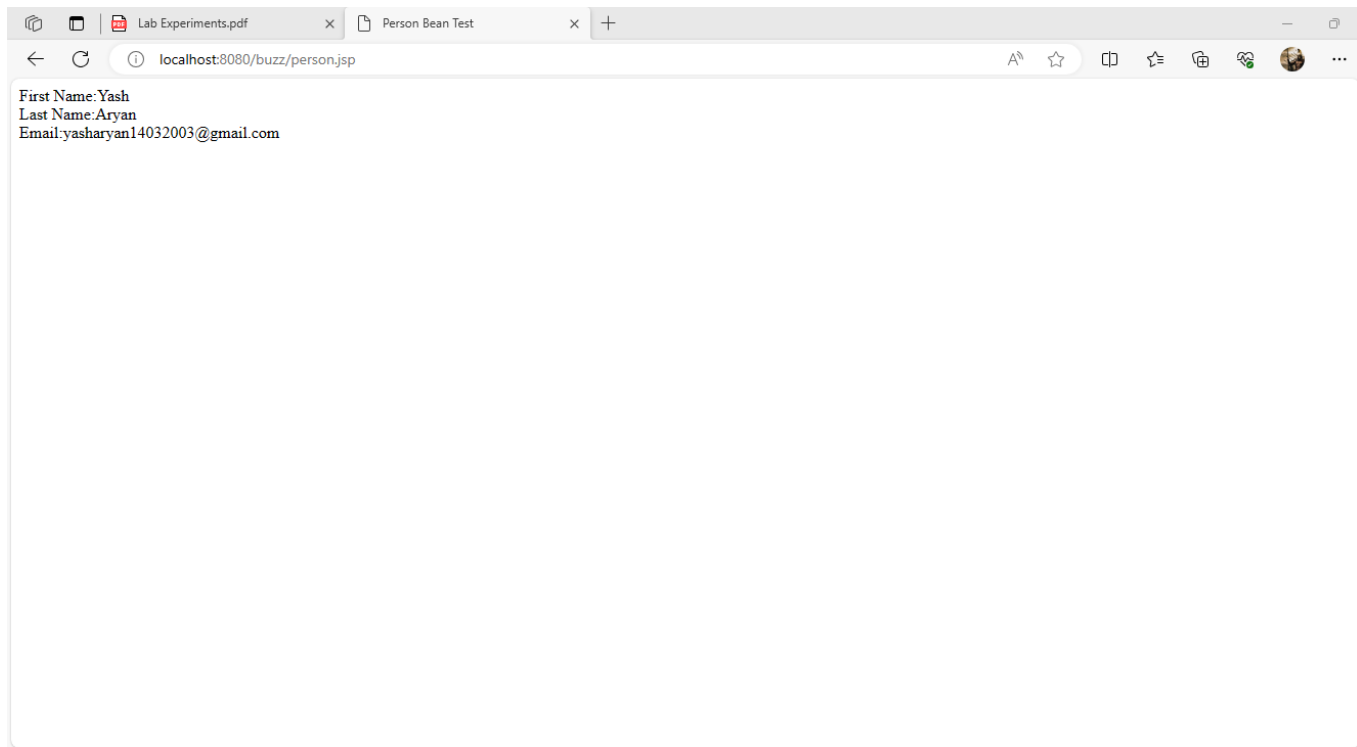
localhost:8080/buzz/form.jsp

## Person Form

First Name:

Last Name:

Email:



Lab Experiments.pdf x Person Bean Test x +

localhost:8080/buzz/person.jsp

First Name: Yash  
Last Name: Aryan  
Email: yasharyan14032003@gmail.com



## Program No. 15

**Develop a Java servlet that is a simple online poll system**

**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>Online Poll</title>
  <style>
    /* Style the radio buttons */
    input[type="radio"] {
      margin-right: 5px;
    }

    /* Style the labels */
    label {
      display: inline;
      font-size: 24px;
    }

    /* Style the submit button */
    input[type="submit"] {
      background-color: #007bff;
      color: white;
      padding: 10px 20px;
      border: none;
      cursor: pointer;
    }
    #option1{
      margin-left: 56px;
    }
    #option3{
      margin-left: 46px;
    }
  </style>
</head>
<body>
  <h1>Vote for Your Favorite Party</h1>
  <form action="VoteServlet" method="post">
    <label for="option1">BJP:</label>
    <input type="radio" name="vote" value="option1" id="option1"
required><br>

    <label for="option2">Congress:</label>
    <input type="radio" name="vote" value="option2" id="option2"
required><br>

    <label for="option3">AAP:</label>
    <input type="radio" name="vote" value="option3" id="option3"
required><br>
```

```

        <input type="submit" value="Vote">
    </form>
</body>
</html>

```

## VoteServlet.java

```

package com.yash;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet("/VoteServlet")
public class VoteServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
        throws ServletException, IOException {
        String selectedOption = request.getParameter("vote");

        // In a real application, you would store the vote in a database or
other persistent storage.
        // For simplicity, we are just printing the result to the console in
this example.
        System.out.println("Vote for " + selectedOption + " recorded.");

        // You can redirect the user to a thank you page or any other relevant
page.
        response.sendRedirect("thankyou.html");
    }
}

```

## web.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="https://jakarta.ee/xml/ns/jakartaee"
xsi:schemaLocation="https://jakarta.ee/xml/ns/jakartaee
https://jakarta.ee/xml/ns/jakartaee/web-app_6_0.xsd" id="WebApp_ID"
version="6.0">
    <servlet>
        <servlet-name>abc</servlet-name>
        <servlet-class>com.yash.VoteServlet</servlet-class>
    </servlet>

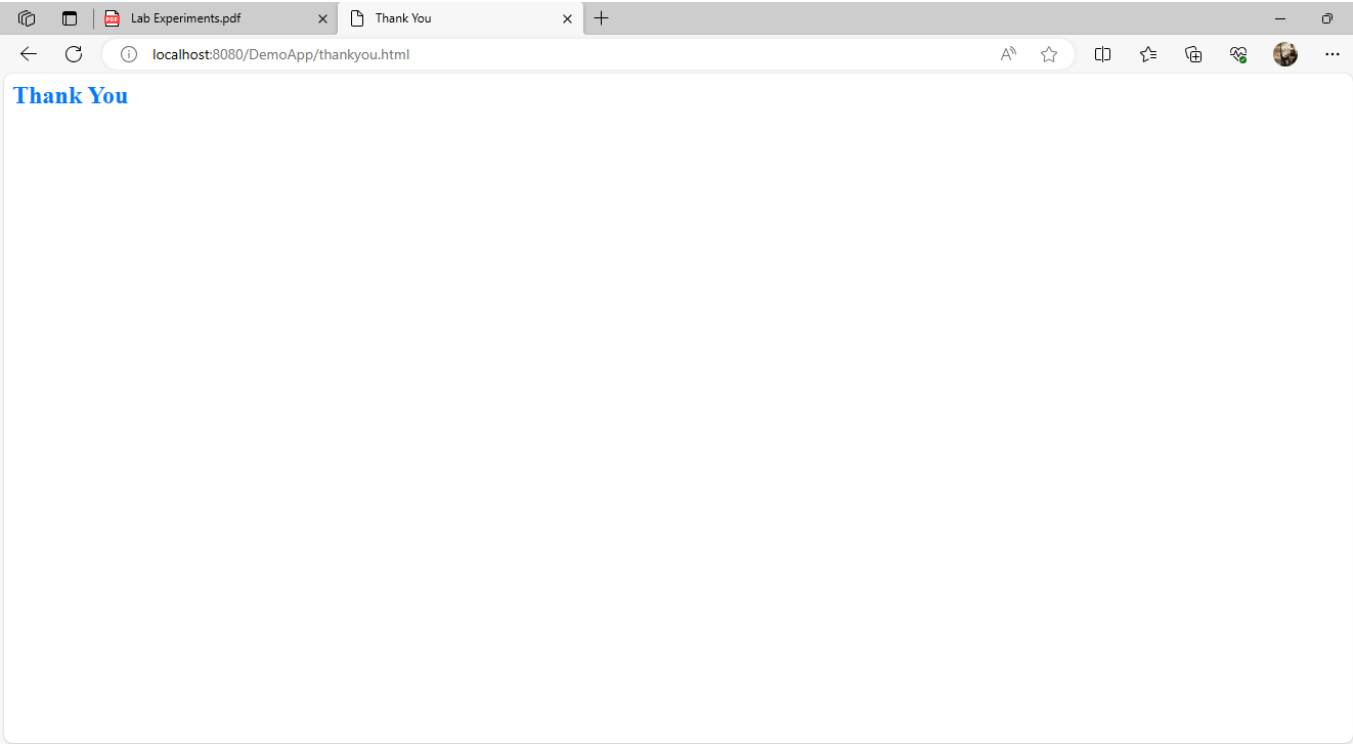
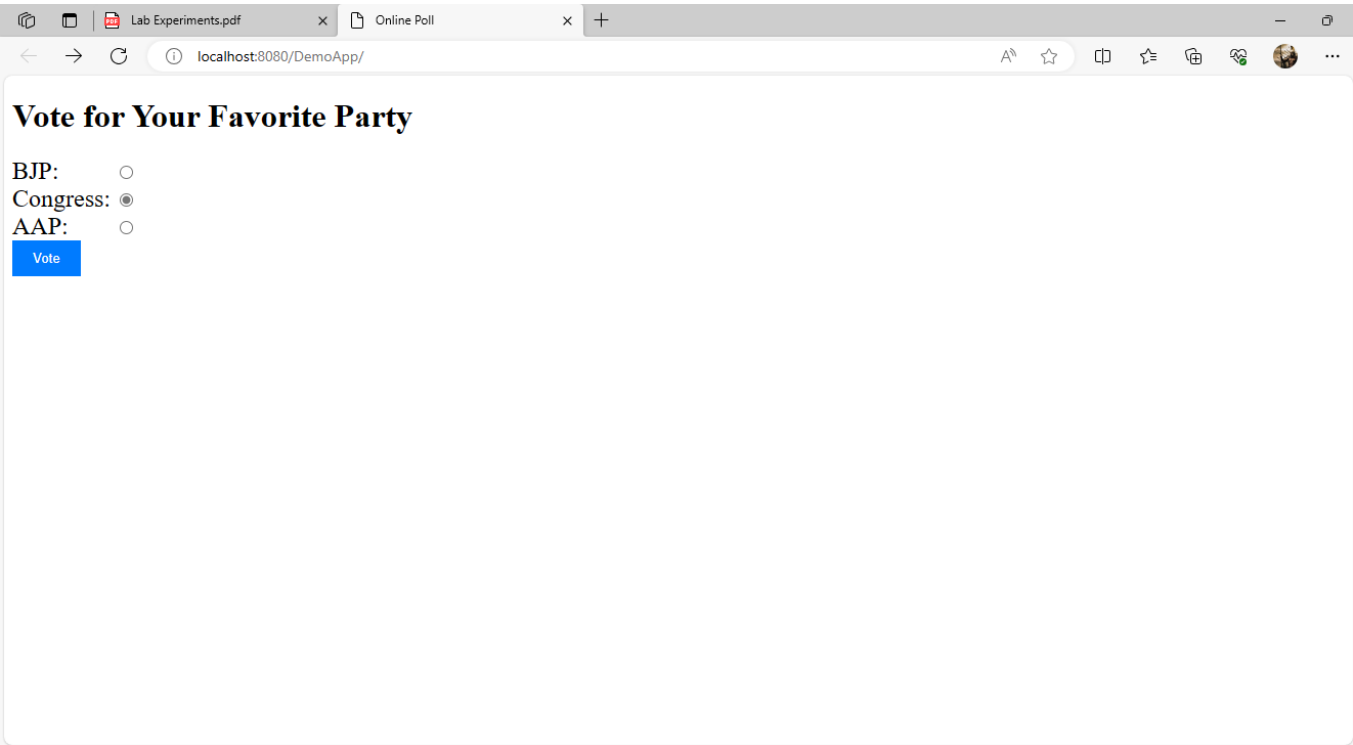
    <servlet-mapping>
        <servlet-name>abc</servlet-name>
        <url-pattern>/add</url-pattern>
    </servlet-mapping>
</web-app>

```

## thankyou.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Thank You</title>
</head>
<body>
<span style="font-size: 24px; font-weight: bold; color: #007bff;">Thank
You</span>
</body>
</html>
```

# Output:



## Program No. 16

Write an experiment using JSP Programming and applications.

Code:

### form.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Person Form</title>
</head>
<body>
<h1>Person Form</h1>
<form method="post" action="person.jsp">
First Name: <input type="text" name="firstName">
<br>
Last Name: <input type="text" name="lastName">
<br>
Email: <input type="text" name="email">
<br>
<Button type="submit" name="submit">Submit</Button>
</form>
</body>
</html>
```

### mrbean.java

```
package netrox;

public class mrbean {
    private String firstName;
    private String lastName;
    private String email;
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
}
```

```
}
```

## person.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Person Bean Test</title>
</head>
<body>
<%
if("POST".equalsIgnoreCase(request.getMethod()) &&
request.getParameter("submit")!=null)
{
%>
<jsp:useBean id="mrbean" class="netrox.mrbean" scope="session">
<jsp:setProperty name="mrbean" property="*/>
</jsp:useBean>
<%
out.println("Welcome"+ mrbean.getFirstName()+ " "+ mrbean.getLastName()+" I am
glad to see you");

}
%>
</body>
</html>
```

# Output:

Lab Experiments.pdf

Person Form

localhost:8080/buzz/form.jsp

Person Form

First Name: Yash

Last Name: Aryan

Email: yasharyan14032003@gmail.

Submit

Lab Experiments.pdf

Person Bean Test

localhost:8080/buzz/person.jsp

WelcomeYash Aryan I am glad to see you

## Problem No. 17

Keeping program 10 as reference write an experiment using JDBC.

Code:

### memberRegister.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <form action="Register" method="post">
        <table>
            <tr>
                <td>User Name</td>
                <td><input type="text" name="uname"></td>
            </tr>
            <tr>
                <td>Password</td>
                <td><input type="password" name="password"></td>
            </tr>
            <tr>
                <td>Email</td>
                <td><input type="text" name="email"></td>
            </tr>
            <tr>
                <td>Phone</td>
                <td><input type="text" name="phone"></td>
            </tr>
            <tr>
                <td>Submit</td>
                <td><input type="submit" value="register"></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

### Register.java

```
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/**
 * Servlet implementation class Register
```



```

    */
@WebServlet("/Register")
public class Register extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Register() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
    HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at:
    ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
    HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        String uname=request.getParameter("uname");
        String password=request.getParameter("password");
        String email=request.getParameter("email");
        String phone=request.getParameter("phone");
        Member member=new Member(uname, password, email, phone);
        RegisterDao rdao=new RegisterDao();
        String result=rdao.insert(member);
        response.getWriter().println(result);
    }
}

```

## RegisterDao.java

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class RegisterDao {
    private String dburl = "jdbc:mysql://localhost:3306/userdb";
    private String dbuname = "root";
    private String dbpassword = "yash";
    private String dbdriver = "com.mysql.cj.jdbc.Driver";

    public void loadDriver(String dbDriver) {
        try {
            Class.forName(dbDriver);

```

```

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}

public Connection getConnection() throws SQLException {
    Connection con = DriverManager.getConnection(dburl, dbuname,
dbpassword);
    return con;
}

public String insert(Member member) {
    loadDriver(dbdriver);
    Connection con = null;
    try {
        con = getConnection();
        if (con != null) {
            String sql = "insert into member values(?,?,?,?)";
            try (PreparedStatement ps = con.prepareStatement(sql)) {
                ps.setString(1, member.getUsername());
                ps.setString(2, member.getPassword());
                ps.setString(3, member.getEmail());
                ps.setString(4, member.getPhone());
                ps.executeUpdate();
                return "Thank you "+member.getUsername();
            } catch (SQLException e) {
                e.printStackTrace();
                return "Data Not Entered Successfully - PreparedStatement
Error: " + e.getMessage();
            }
        } else {
            return "Database Connection Error - Connection is null";
        }
    } catch (SQLException e) {
        e.printStackTrace();
        return "Database Connection Error - Exception: " + e.getMessage();
    } finally {
        try {
            if (con != null) {
                con.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}
}

```

## Member.java

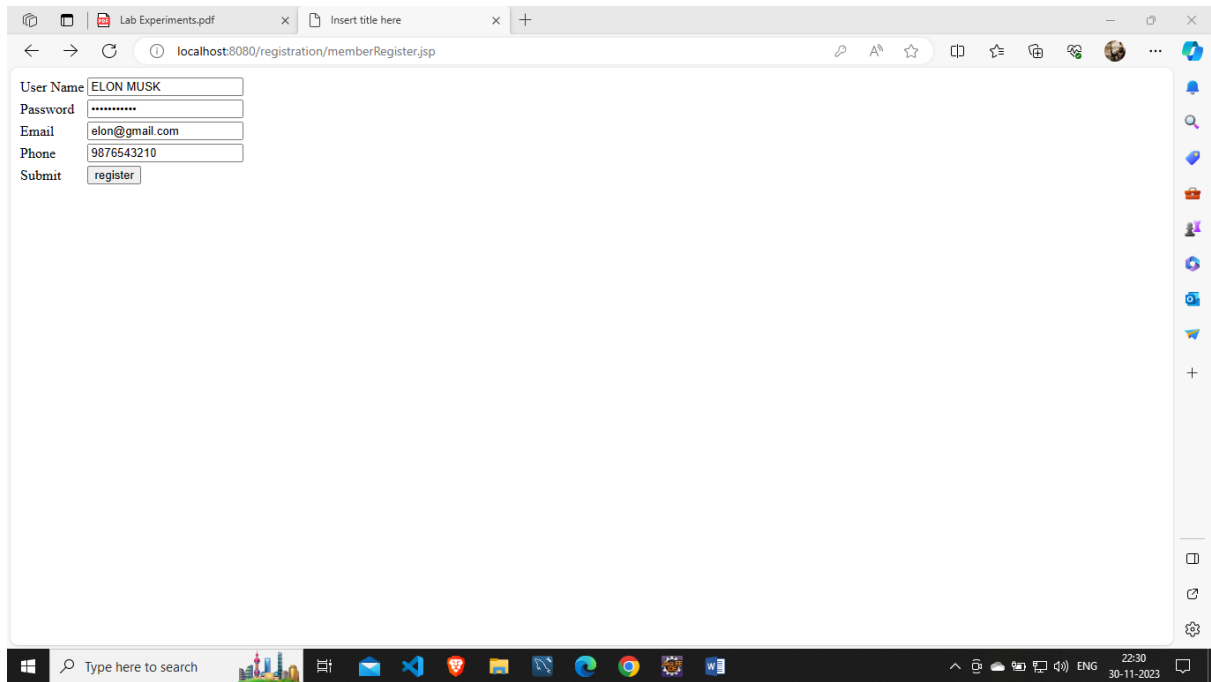
```

public class Member
{
    private String uname,password,email,phone;
    public Member() {
        super();
    }
}

```

```
    public Member(String uname, String password, String email, String phone)
{
    super();
    this.uname = uname;
    this.password = password;
    this.email = email;
    this.phone = phone;
}
    public String getUname() {
        return uname;
    }
    public void setUname(String uname) {
        this.uname = uname;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```

# Output:



Lab Experiments.pdf x Insert title here x +

localhost:8080/registration/memberRegister.jsp

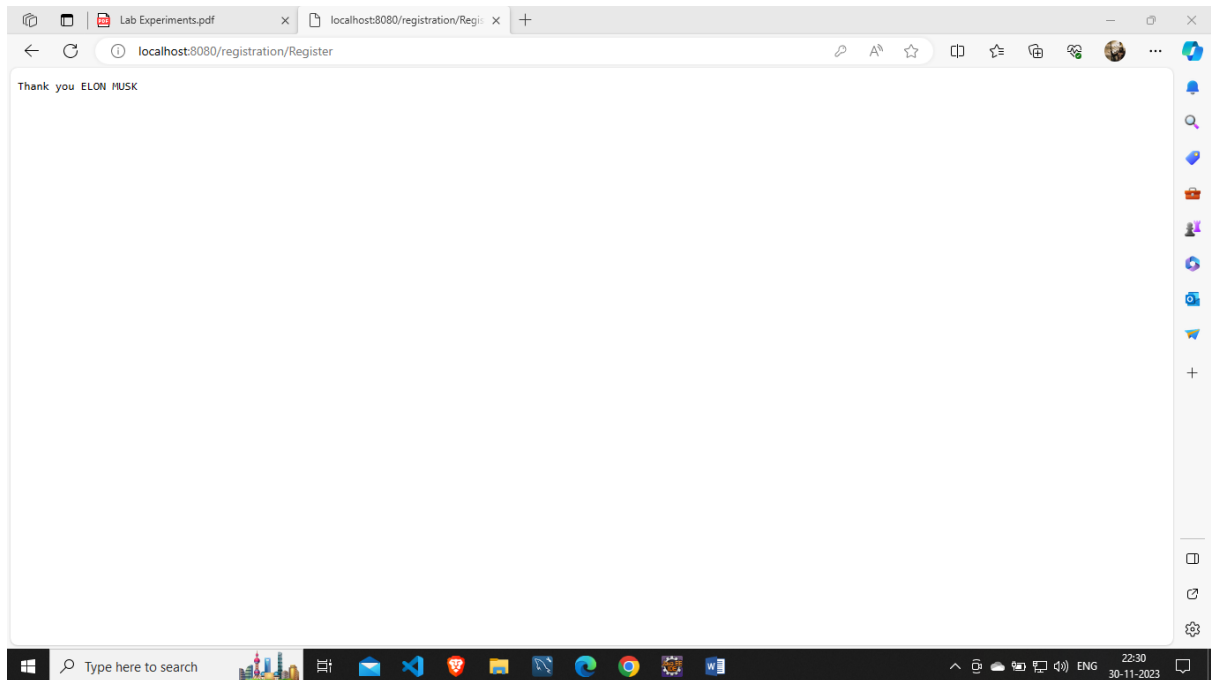
User Name

Password

Email

Phone

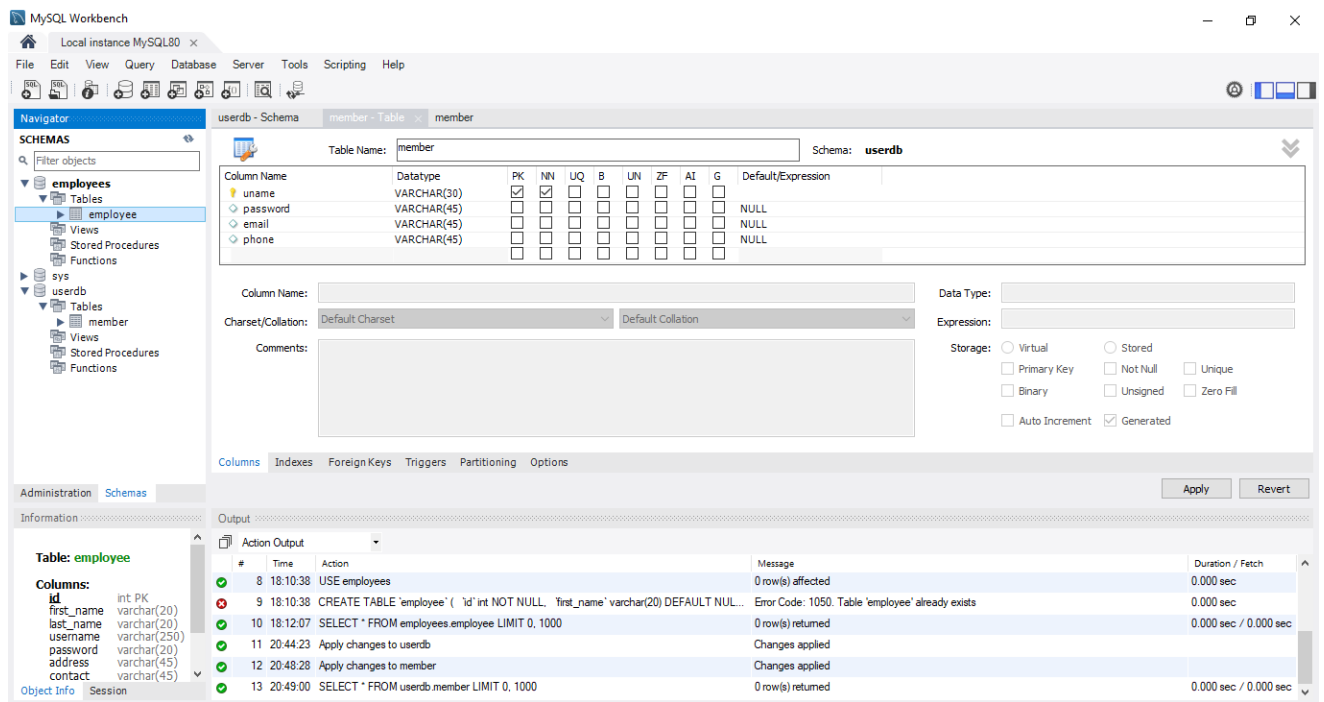
Submit



Lab Experiments.pdf x localhost:8080/registration/Regi x +

localhost:8080/registration/Register

Thank you ELON MUSK



**Table: employee**

**Columns:**

id	int PK
first_name	varchar(20)
last_name	varchar(20)
username	varchar(250)
password	varchar(20)
address	varchar(45)
contact	varchar(45)

## **Program No. 18**

**Case study using any above program**

# **Case Study on Program No. 12 (Text Summarizer using AI)**

# TABLE OF CONTENT

CONTENT	Page No.
Index	2
Abstract	3
Problem Statement	4
Functional Requirements	5
Non Functional Requirements	6
Design & Implementation	7
Components	8
Workflow	9
Future Work	10
Conclusion	11
References	12

# Abstract

This summarizer uses article extractor and summarizer API for extracting the article from the link and summarizing it. This project was made with React, Vite.js and Rapid API. When a user pastes a link to an article in the search bar, it extracts the article from that link, summarizes the article and presents the summary to the user within five seconds. The search history of user is also stored, which can be further used by copying the link.

A link to the source code is also available in the upper right corner of the web page. The user can clone the repository and run the project on his local machine by using “npm run dev” command in the terminal. To build the project locally, user must use “npm run build” command in the terminal.

This project is deployed on netlify and can be accessed by following the link: <https://frabjous-tarsier-14495c.netlify.app/>



# Problem Statement

In the ever-evolving digital landscape, where information is constantly being generated and disseminated at an unprecedented rate, the need for effective tools to manage and process this vast data has become increasingly crucial. Among the many technological advancements that address this challenge, text summarizers stand out as indispensable tools that empower users to navigate the information overload and extract the essence of lengthy texts.

The need for text summarizers stems from the sheer volume of information that we encounter daily. From news articles and research papers to emails and social media feeds, we are constantly bombarded with text-based content. Devoting time to read and comprehend each piece of information in its entirety is often impractical, if not impossible. Text summarizers alleviate this burden by providing a quick overview of the key points and main ideas, allowing us to make informed decisions about whether to delve deeper into the original text.

Moreover, text summarizers play a vital role in enhancing comprehension and retention of information. By condensing complex concepts into a more succinct format, these tools help us grasp the core message more effectively and retain the information for longer periods. This is particularly beneficial for students, professionals, or anyone who needs to learn and retain information quickly.

As we move deeper into the digital age, the need for text summarizers will continue to grow. These tools have the potential to revolutionize the way we interact with information, enabling us to navigate the information overload more effectively, enhance our understanding of complex topics, and make informed decisions based on a comprehensive grasp of the available data.

# Functional Requirements

## 1. URL Input:

- The application must allow users to input URL for summarization.

## 2. User Interaction:

- Users must be able to interact with the web application.

## 3. Summary Display:

- The application must display the generated summary to the user.

## 4. Integration with Rapid API:

- The application must effectively integrate with the Rapid API for utilizing external services or data sources related to text summarization.
- API calls should be handled securely and efficiently.

## 5. Responsive Design:

- The user interface must be responsive and accessible on various devices, including desktops, tablets, and mobile phones.
- It must provide an optimal user experience across different screen sizes.

# **Non-functional Requirements**

## **1. Performance:**

- The summarization process must be fast and responsive, providing quick results to users.
- The application must handle simultaneous requests efficiently, ensuring good performance under varying loads.

## **2. Security:**

- User input and data must be handled securely to prevent vulnerabilities such as cross-site scripting or injection attacks.
- API calls to Rapid API must be secured using appropriate authentication mechanisms.

## **3. Reliability:**

- The application must be reliable and available for users at all times.
- It must include error handling and recovery mechanisms to ensure a smooth user experience.

## **4. User Experience (UX):**

- The user interface must be intuitive and user-friendly.
- It must provide clear instructions, feedback, and visual cues to guide users through the summarization process.

## **5. Compatibility:**

- The application must be compatible with modern web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
- It must adhere to web standards and best practices for cross-browser compatibility.

# Design and Implementation

- **Tech Stack Used**
  - **Front-end:** ReactJs, Vite
  - **Back-end:** Firebase
  - **API:** Article Extractor and Summarizer API
  - **Deployment Platform:** Netlify
  - **Additional**
    - Rapid API
    - VS Code Extension: ES7+ React/Redux/React-Native snippets
- **Hardware and Software Interfaces**
  - **Hardware**
    - Fast internet enabled mobile or computer device
  - **Software**

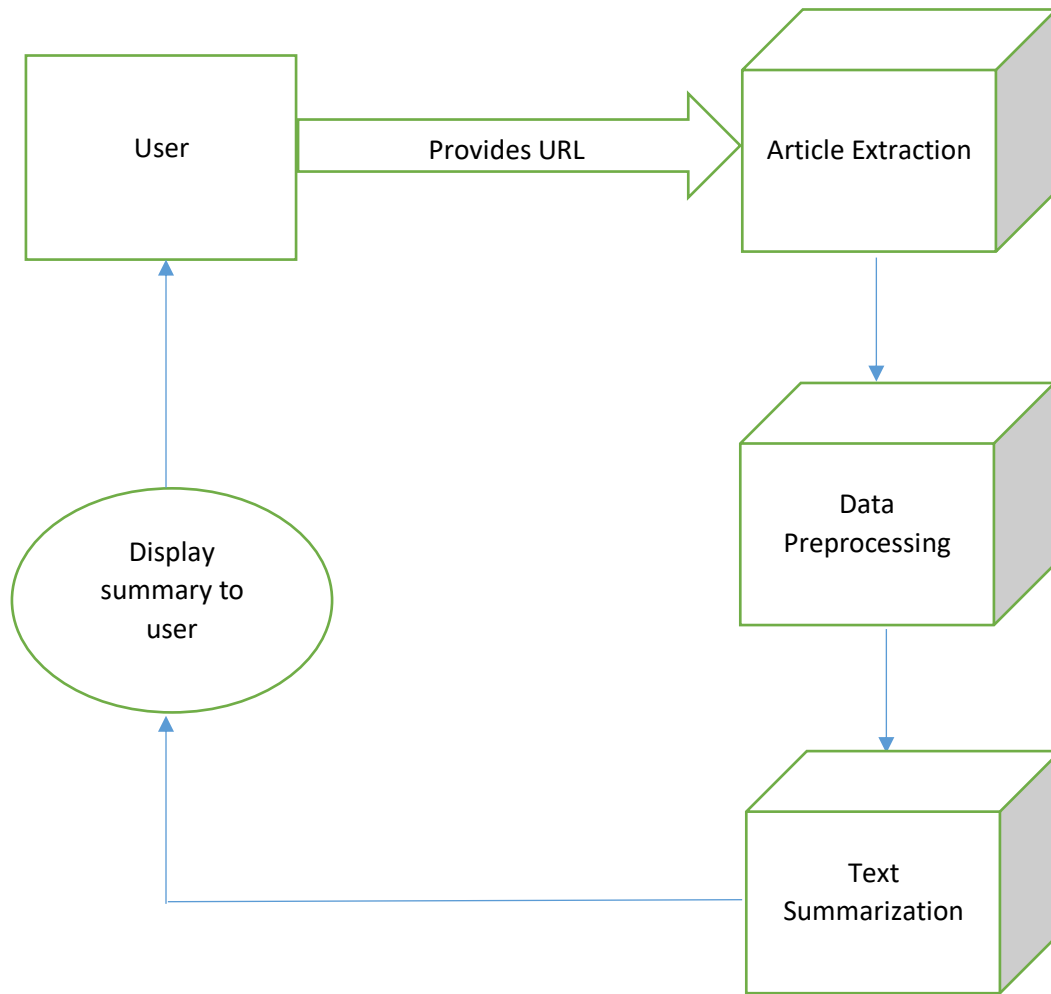
Software Used	Description
ReactJs	ReactJs is a free and open-source front-end JavaScript library for building user interfaces based on components.
Vite	<p>Vite is a frontend build tool that significantly improves the frontend development experience. It consists of two major parts:</p> <ul style="list-style-type: none"><li>• A dev server that serves your source files over native ES modules, with rich built-in features and astonishingly fast Hot Module Replacement (HMR).</li><li>• A build command that bundles your code with Rollup, pre-configured to output highly optimized static assets for production.</li></ul>

# Components

- **Architecture for Front-End:** Achieved using ReactJs and Vite. ReactJs is a free and open-source front-end JavaScript library for building user interfaces based on components. Vite is a frontend build tool that significantly improves the frontend development experience. It consists of two major parts:
  - A dev server that serves your source files over native ES modules, with rich built-in features and astonishingly fast Hot Module Replacement (HMR).
  - A build command that bundles your code with Rollup, pre-configured to output highly optimized static assets for production.
- **Article extracting and summarizing Services:** Achieved by Article Extractor and Summarizer API. It extracts news/article body from a URL and uses GPT to summarize (and optionally translate) the article content. Useful for text mining purposes. Leverages powerful and flexible web scraping engine ([ScrapeNinja.net](https://scrapeninja.net)) with high-quality rotating proxies under the hood.

# Workflow

## Web Application Workflow Overview



- User provides URL of an article to be summarized
- Extracts the article to be summarized
- Uses GPT to summarize the article
- Displays summary of article to the user

# Future Work

- **Advanced summarization techniques**
  - Implement more sophisticated text summarization algorithms, such as deep learning-based models, to generate more comprehensive and informative summaries.
- **Summarization length options**
  - Allow users to select the desired length of the summary, catering to different needs and preferences.
- **Topic selection**
  - Provide the ability to specify the main topic of the text to be summarized, enabling more focused and relevant summarization.
- **Sentiment analysis**
  - Integrate sentiment analysis to identify the overall emotional tone of the text and incorporate it into the summary.
- **Keywords extraction**
  - Identify and display the most important keywords from the text to provide a quick overview of the key points.
- **Supporting multiple languages**
  - Enable the summarization of text in different languages to broaden the app's usability.
- **Summarizing from various sources**
  - Allow users to paste text directly, input URLs to web pages, or upload documents for summarization.

# CONCLUSION

This summarizer uses article extractor and summarizer API for extracting the article from the link and summarizing it. This project was made with React, Vite.js and Rapid API. When a user pastes a link to an article in the search bar, it extracts the article from that link, summarizes the article and presents the summary to the user within five seconds. The search history of user is also stored, which can be further used by copying the link.

This project is deployed on netlify and can be accessed by following the link: <https://frabjous-tarsier-14495c.netlify.app/>



# REFERENCES

- [1] Jon Duckett, "HTML and CSS: Design and Build Web Sites", 2011
- [2] Chris Aquino, "Front-End Web Development: The Big Nerd Ranch Guide", 2016
- [3] Chong Lip Phang, "Mastering Front-End Web Development", 2020
- [4] Deitel & Goldberg, "Internet and world wide web-How to program", Pearson Education
- [5] Hans Bergsten, "Java Server Pages", SPD O'Reilly
- [6] Chris Bates, "Web programming, building internet applications", 2<sup>nd</sup> edition, Wiley

## Problem No. 19

### Database Programming

#### Code:

#### memberRegister.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <form action="Register" method="post">
        <table>
            <tr>
                <td>User Name</td>
                <td><input type="text" name="uname"></td>
            </tr>
            <tr>
                <td>Password</td>
                <td><input type="password" name="password"></td>
            </tr>
            <tr>
                <td>Email</td>
                <td><input type="text" name="email"></td>
            </tr>
            <tr>
                <td>Phone</td>
                <td><input type="text" name="phone"></td>
            </tr>
            <tr>
                <td>Submit</td>
                <td><input type="submit" value="register"></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

#### Register.java

```
import java.io.IOException;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
/**
 * Servlet implementation class Register
 */
@WebServlet("/Register")
```

```

public class Register extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Register() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
    HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at:
    ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
    HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        String uname=request.getParameter("uname");
        String password=request.getParameter("password");
        String email=request.getParameter("email");
        String phone=request.getParameter("phone");
        Member member=new Member(uname, password, email, phone);
        RegisterDao rdao=new RegisterDao();
        String result=rdao.insert(member);
        response.getWriter().println(result);

    }
}

```

## RegisterDao.java

```

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class RegisterDao {
    private String dburl = "jdbc:mysql://localhost:3306/userdb";
    private String dbuname = "root";
    private String dbpassword = "yash";
    private String dbdriver = "com.mysql.cj.jdbc.Driver";

    public void loadDriver(String dbDriver) {
        try {
            Class.forName(dbDriver);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
}

public Connection getConnection() throws SQLException {
    Connection con = DriverManager.getConnection(dburl, dbuname,
dbpassword);
    return con;
}

public String insert(Member member) {
    loadDriver(dbdriver);
    Connection con = null;
    try {
        con = getConnection();
        if (con != null) {
            String sql = "insert into member values(?,?,?,?)";
            try (PreparedStatement ps = con.prepareStatement(sql)) {
                ps.setString(1, member.getUsername());
                ps.setString(2, member.getPassword());
                ps.setString(3, member.getEmail());
                ps.setString(4, member.getPhone());
                ps.executeUpdate();
                return "Data Entered Successfully";
            } catch (SQLException e) {
                e.printStackTrace();
                return "Data Not Entered Successfully - PreparedStatement
Error: " + e.getMessage();
            }
        } else {
            return "Database Connection Error - Connection is null";
        }
    } catch (SQLException e) {
        e.printStackTrace();
        return "Database Connection Error - Exception: " + e.getMessage();
    } finally {
        try {
            if (con != null) {
                con.close();
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
}
}
}

```

## Member.java

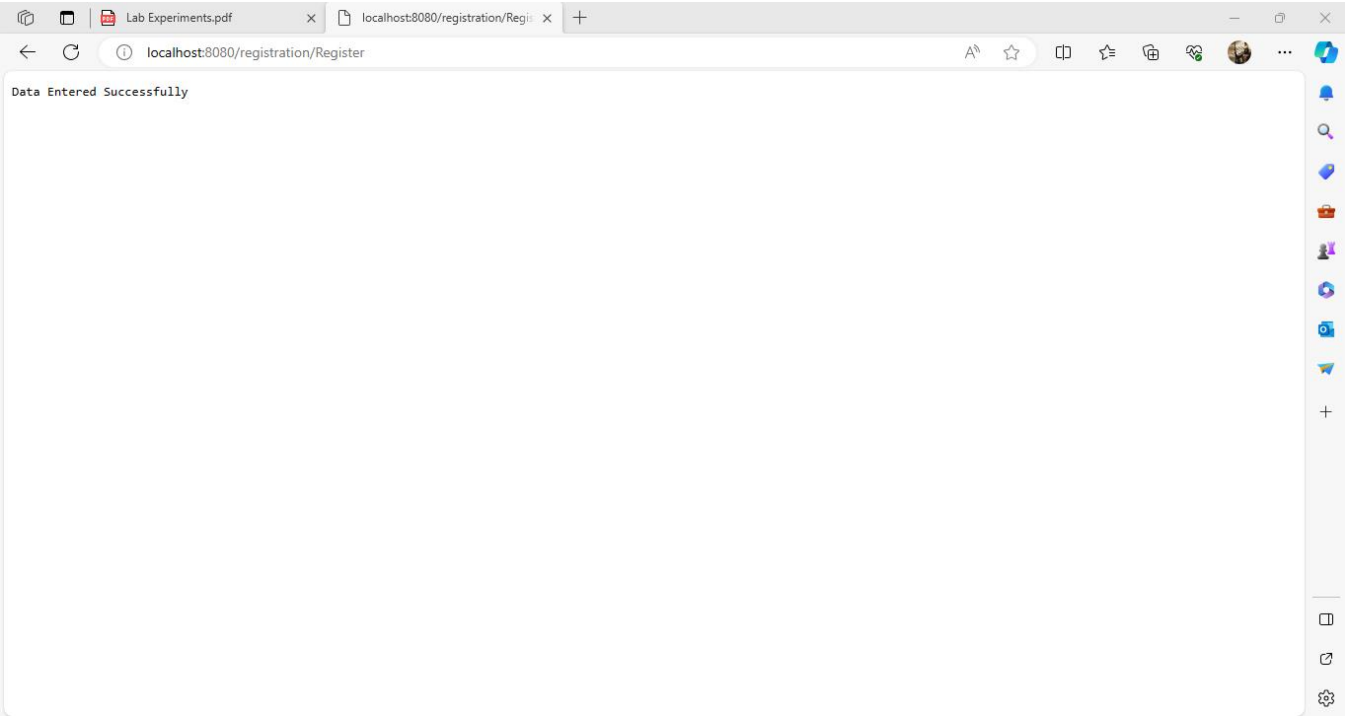
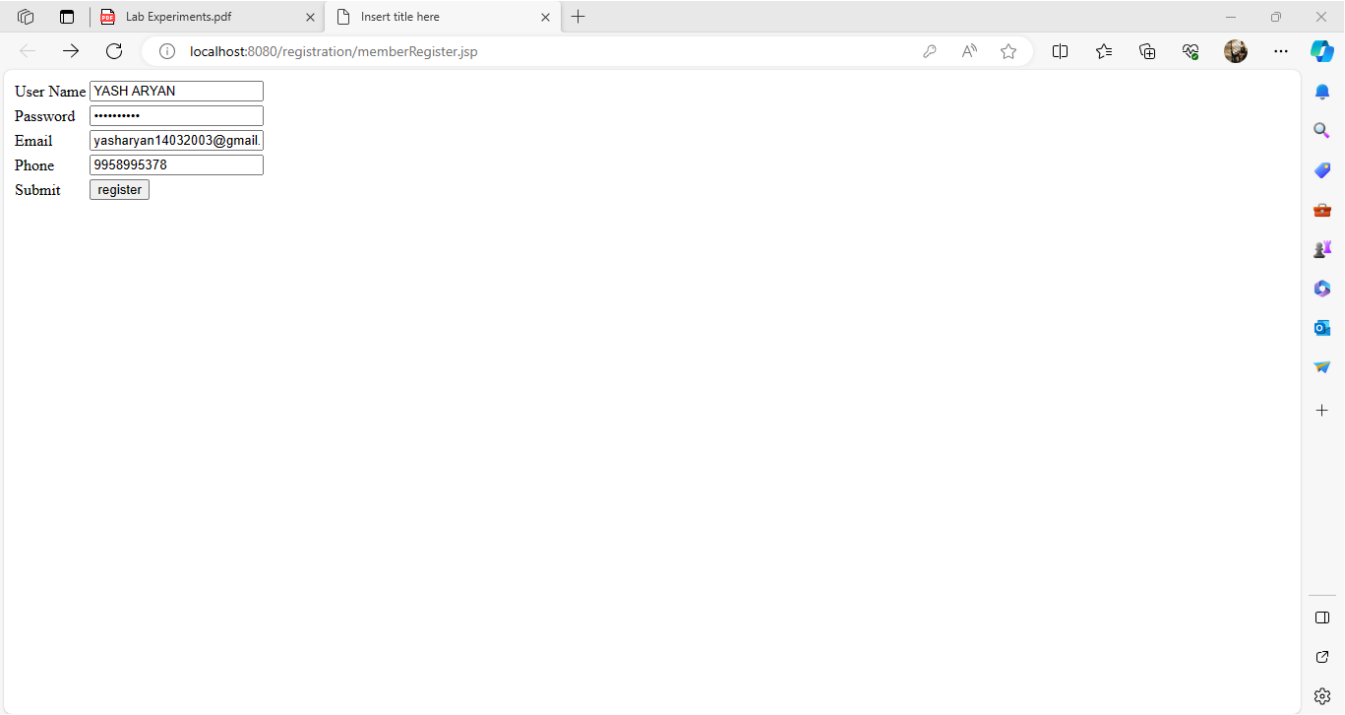
```

public class Member
{
    private String uname,password,email,phone;
    public Member() {
        super();
    }
    public Member(String uname, String password, String email, String phone)
    {
        super();
    }
}

```

```
        this.username = username;
        this.password = password;
        this.email = email;
        this.phone = phone;
    }
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```

# Output:



MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

employees

Tables

employee

Views

Stored Procedures

Functions

sys

userdb

Tables

member

Views

Stored Procedures

Functions

Administration Schemas

Information

Table: employee

Columns:

id int PK

first\_name varchar(20)

last\_name varchar(20)

username varchar(250)

password varchar(20)

address varchar(45)

contact varchar(45)

Object Info Session

userdb - Schema member - Table member

Table Name: member Schema: userdb

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
uname	VARCHAR(30)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
password	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
email	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
phone	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name: Charset/Collation: Default Charset Default Collation Comments:

Data Type: Expression:

Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☐ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☒ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
8	18:10:38	USE employees	0 row(s) affected	0.000 sec
9	18:10:38	CREATE TABLE 'employee' ( 'id' int NOT NULL, 'first_name' varchar(20) DEFAULT NUL...	Error Code: 1050. Table 'employee' already exists	0.000 sec
10	18:12:07	SELECT * FROM employees.employee LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
11	20:44:23	Apply changes to userdb	Changes applied	
12	20:48:28	Apply changes to member	Changes applied	
13	20:49:00	SELECT * FROM userdb.member LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

Table: employee

Columns:

id	int PK
first_name	varchar(20)
last_name	varchar(20)
username	varchar(250)
password	varchar(20)
address	varchar(45)
contact	varchar(45)

## Program No. 20

**Create a website of USICT, GGSIPU containing all necessary and required information**

### Code:

```
<!DOCTYPE html>
<html>
<head>
  <title>USICT GGSIPU</title>
  <meta name="viewport" content="width=device-width">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" type="text/css" href="css/all.css">
  <link rel="stylesheet" type="text/css" href="css/all.min.css">
  <link rel="stylesheet" type="text/css" href="css/lightbox.css">
  <link rel="stylesheet" type="text/css" href="css/flexslider.css">
  <link rel="stylesheet" type="text/css" href="css/owl.carousel.css">
  <link rel="stylesheet" type="text/css" href="css/owl.theme.default.css">
  <link rel="stylesheet" type="text/css" href="css/jquery.rateyo.css"/>
  <link rel="stylesheet" type="text/css" href="inner-page-style.css">
  <link rel="stylesheet" type="text/css" href="style.css">
  <link href="https://fonts.googleapis.com/css?family=Raleway:400,500,600,700"
rel="stylesheet">
</head>
<body>
  <div id="page" class="site" itemscope
itemtype="http://schema.org/LocalBusiness">
    <header class="site-header">
      <div class="top-header">
        <div class="container">
          <div class="top-header-left">
            <div class="top-header-block">
              <a href="ipu.ac.in" itemprop="email"><i class="fas
fa-envelope"></i> ipu.ac.in</a>
            </div>
            <div class="top-header-block">
              <a href="tel: +91-11-25302 801"
itemprop="telephone"><i class="fas fa-phone"></i> +91-11-25302 801</a>
            </div>
          </div>
          <div class="top-header-right">
            <div class="social-block">
              <ul class="social-list">
                <li><a href=""><i class="fab fa-
viber"></i></a></li>
                <li><a href=""><i class="fab fa-google-plus-
g"></i></a></li>
                <li><a href=""><i class="fab fa-facebook-
square"></i></a></li>
                <li><a href=""><i class="fab fa-facebook-
messenger"></i></a></li>
```



```

                <li><a href=""><i class="fab fa-
twitter"></i></a></li>

                <li><a href=""><i class="fab fa-
skype"></i></a></li>

            </ul>
        </div>
        <div class="login-block">
            <a href="">Student </a>
            <a href="">Faculty Login</a>
        </div>
    </div>
</div>
<!-- Top header Close -->
<div class="main-header">
    <div class="container">
        <div class="logo-wrap" itemprop="logo">
            
            <!-- <h1>Education</h1> -->
        </div>
        <div class="nav-wrap">
            <nav class="nav-desktop">
                <ul class="menu-list">
                    <li><a href="#">Home</a></li>
                    <li class="menu-parent">Courses
                        <ul class="sub-menu">
                            <li><a href="#">M.Tech</a></li>
                            <li><a href="#">M.B.A</a></li>
                            <li class="menu-parent">B.Tech
                                <ul class="sub-menu">
                                    <li><a href="">CSE</a></li>
                                    <li><a href="">IT</a></li>
                                </ul>
                            </li>
                        </ul>
                    </li>
                    <li class="menu-parent">Administration
                        <ul class="sub-menu">
                            <li><a href="#">VC</a></li>
                            <li><a href="#">Dean</a></li>
                        </ul>
                    </li>
                    <li><a href="">About</a></li>
                    <li><a href="">Gallery</a></li>
                    <li><a href="">Contact</a></li>
                </ul>
            </nav>
            <div id="bar">
                <i class="fas fa-bars"></i>
            </div>
            <div id="close">
                <i class="fas fa-times"></i>
            </div>
        </div>
    </div>

```



```

        <p><span>Duration:</span> 4 Years</p>
        <p><span>Class Time:</span> 9:30am-5:30pm </p>
        <p><span>Fee:</span> 4,00,000</p>
    </section>
</div>
</div>
</div>

    <div class="box-wrap" itemprop="event" itemscope
itemtype=" http://schema.org/Course">
        <div class="img-wrap" itemprop="image"></div>
        <a href="#" class="learn-desining-banner"
itemprop="name">B.Tech ECE >>></a>
        <div class="box-body" itemprop="description">
            <p>Study ECE at USICT, believe me you will never
regret it. We are one of the best colleges in Delhi</p>
            <section itemprop="time">
                <p><span>Duration:</span> 4 Years</p>
                <p><span>Class Time:</span> 9:30am-5:30pm </p>
                <p><span>Fee:</span> 4,00,000</p>
            </section>
        </div>
    </div>
</div>
</div>
</div>
</div>
<!-- Learn courses End -->
<section class="whyUs-section">
    <div class="container">
        <div class="featured-points">
            <ul>
                <li><i class="fas fa-book"></i> free books for
students</li>
                <li><i class="fas fa-money-check-alt"></i> affordable
fees</li>
                <li><i class="fas fa-chalkboard-teacher"></i> experienced
teachers</li>
                <li><i class="fas fa-book"></i> free books for
faculties</li>
            </ul>
        </div>
        <div class="whyus-wrap">
            <h1>why us?</h1>
            <p itemprop="description">The University School of
Information, Communication & Technology has been established to design and
implement courses with a twin objective of generating effective professionals and
to keep pace with the R & D activities of this fast emerging and changing field of
Information & Communication Technology. The school aims at inculcating essential
skills as demanded by the global software industry, through the interactive
learning process. This includes team-building skills, audio-visual presentations
and personality development programmes. These enhance analytic and communication
skills, besides inculcating the virtues of self-study.</p>

```

```
        <a href="#" class="read-more-btn">read more</a>
    </div>
</div>
</section>
</div>
<script type="text/javascript" src="js/jquery-3.3.1.min.js"></script>
<script type="text/javascript" src="js/lightbox.js"></script>
<script type="text/javascript" src="js/all.js"></script>
<script type="text/javascript" src="js/isotope.pkgd.min.js"></script>
<script type="text/javascript" src="js/owl.carousel.js"></script>
<script type="text/javascript" src="js/jquery.flexslider.js"></script>
<script type="text/javascript" src="js/jquery.rateyo.js"></script>
<script type="text/javascript" src="js/custom.js"></script>
</body>
</html>
```

# Output:

