

PRANVEER SINGH INSTITUTE OF TECHNOLOGY, KANPUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Even Semester 2022-23



B. Tech.- Third Year

Semester- VI

Lab File

WEB TECHNOLOGY

(KCS652)

Submitted To :

Faculty Name :_____

Designation :_____

Submitted By :

Name :_____

Roll No. :_____

Section :_____

INDEX

INDEX				
Lab No.	Objective	Date	Marks	Sign.
1	Write HTML/Java scripts to display your CV in navigator, your Institute website, Department Website and Tutorial website for specific subject.			
2	Write an HTML program to design an entry form of student details			
3	Write program using Java script for Web Page to display browsers information.			
4	A super class Detail has been defined to store the details of a customer. Define a subclass Bill to compute the monthly telephone charge of the customer as per the chart given below: Number of Calls Rate 1 – 100 Only Rental charge 101 – 200 60 paisa per call + rental charge 201 – 300 80 paisa per call + rental charge Above 300 1 rupee per call + rental charge			
5	Write a Java applet/AWT to display the Application Program screen i.e. calculator and other.			
6	Write a Java applet to display the Application Program screen i.e. Colour mixer pallet			
7	Write a program using TCP/IP socket between client and server and perform two-way communication			
8	Write a program to illustrate JDBC connectivity and perform CRUD operation on a table student/employee (at least 5 attributes)			
9	Write a program to illustrate Batch Transaction with prepared statement			
10	Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page.			
11	To Develop a student Marks sheet by using Servlet and HTML with database Oracle.			
12	Design and implement a simple servlet for an entry form of student details and send it to store at database server like SQL, Oracle or MS Access.			

13	Write a JSP which insert the details of the 3 or 4 users who register with the web site by using registration form. Authenticate the user when he submits the login form using the user name and password from the database			
14	Store 5 Students (name, branch, rollno, age) objects in list. Perform any sorting technique so as to display list in ascending order of rollno and display in descending order of age.			
15	Perform union (AUB), intersection and difference (A-B) operations using set collection.			
16	Create a Map that consists of Country-Capital pair to store information about various countries along with their capital name. Display the entries in sorted order of Country and Capital.			

Program 1

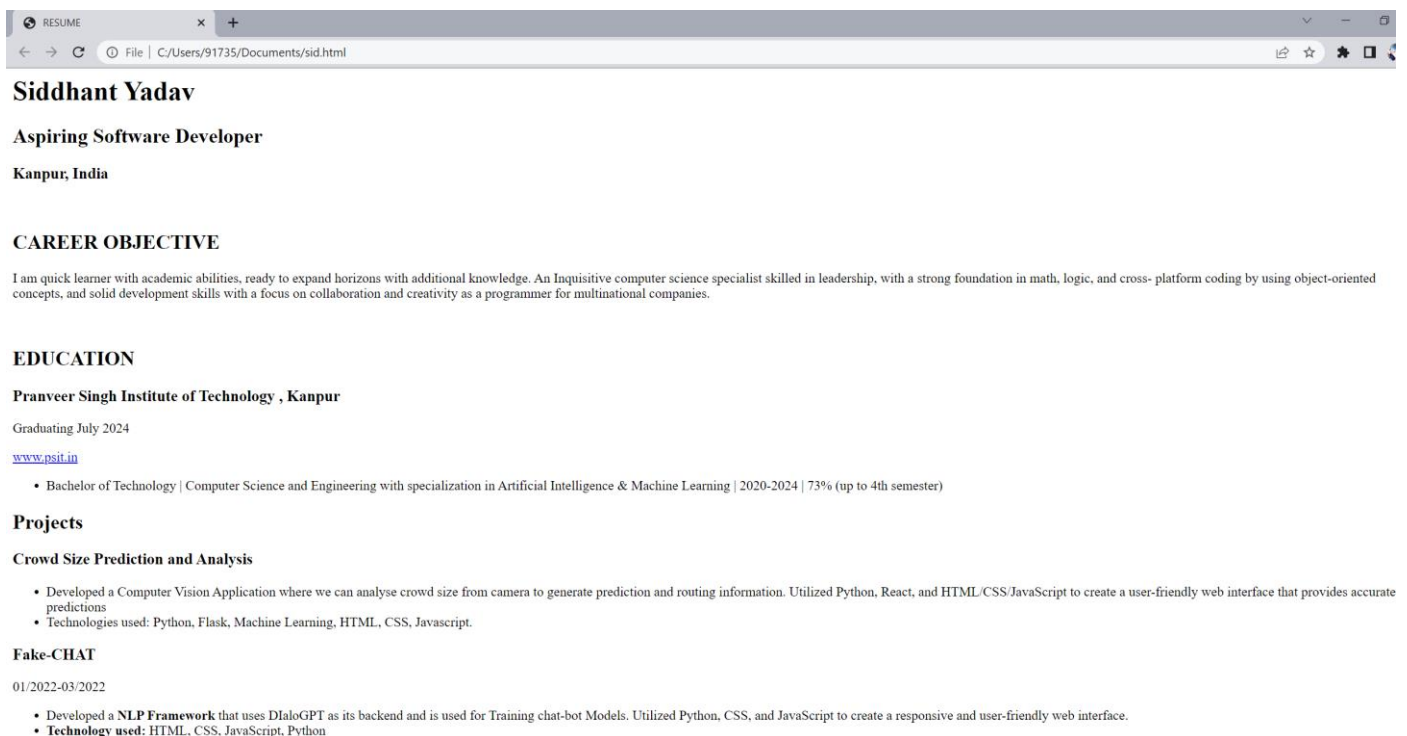
Objective: Write HTML/Java scripts to display your CV in navigator, your Institute website, Department Website and Tutorial website for specific subject.

Code:

```
<html>
<head>
  <title>RESUME </title>
</head>
<body> <!-- BEGIN DIV FOR OVERALL BOX -->
  <div id="resume"> <!-- THIS DIV CENTERS OUR HEADING -->
    <h1>Siddhant Yadav</h1>
    <h2>Aspiring Software Developer</h2>
    <h3>Kanpur, India</h3> <br /> <!-- END CENTERING DIV -->
  </div>
  <h2> CAREER OBJECTIVE</h2>
  <p> I am quick learner with academic abilities, ready to expand horizons with additional knowledge. An Inquisitive computer science specialist skilled in leadership, with a strong foundation in math, logic, and cross- platform coding by using object-oriented concepts, and solid development. </p> <br />
  <h2>EDUCATION</h2>
  <h3>Pranveer Singh Institute of Technology , Kanpur </h3>
  <p> Graduating July 2024</p>
  <a href="/">www.psit.in</a>
  <ul>
    <li> Bachelor of Technology | Computer Science and Engineering with specialization in Artificial Intelligence & Machine Learning | 2020-2024 | 73% (up to 4th semester)</li>
  </ul>
  <h2>Projects</h2>
  <h3>Crowd Size Prediction and Analysis</h3>
  <ul>
    <li> Developed a Computer Vision Application where we can analyse crowd size from camera to generate prediction and routing information. Utilized Python, React, and HTML/CSS/JavaScript to create a user-friendly web interface that provides accurate predictions</li>
    <li>Technologies used: Python, Flask, Machine Learning, HTML, CSS, Javascript.</li>
```

```
</ul>
<h3>Data Models</h3>
<p> 01/2022-03/2022 </p>
<ul>
  <li> Developed a <strong>NLP Framework</strong> that uses DialoGPT as its backend
and is used for Training chat-bot Models.
    Utilized Python,
    CSS, and JavaScript to create a responsive and user-friendly web interface.</li>
  <li><strong>Technology used: </strong> HTML, CSS, JavaScript, Python</li>
</ul> <br />
<h2>Courses</h2>
<ul>
  <li>Programming Foundations with Javascript, HTML and CSS </li>
  <p><a href="/">www.webdevelopmentbootcamp</a></p>
  <li>Supervised machine learning: Regression and Classification </li>
  <p><a href="/">www.machinelearning</a></p>
  <li>Programming for Everybody </li>
  <p><a href="/">www.programmingforeverybody</a></p>
  <li>Technical Support Fundamentals </li>
  <p><a href="/">www.technicalsupportfundamentals</a></p>
</ul>
</body>
</html>
```

Output:-



PROGRAM-2

Objective: Write an HTML program to design an entry form of student details

Code:

```
<html>
<head>
<script type="text/javascript" src="validate.js" ></script>
</head>
<style>
    td{
        font-size: 17px;
    }
</style>
<body bgcolor="aqua" border="5" style="padding: 6rem;">
<form action="register.jsp" name="StudentRegistration" method="post">
<table cellpadding="2" width="50%" border="10" align="center" cellspacing="2">
<tr>
<td colspan=2>
<center><font size=6><b>Student Registration Form</b></font></center>
</td>
</tr>
<tr>
<td>Name</td>
<td><input type="text" name="textnames" id="textname" size="35"></td>
</tr>
<tr>
<td>Father Name</td>
<td><input type="text" name="fathername" id="fathername"
size="35"></td>
</tr>
<tr>
<td>Postal Address</td>
<td><input type="text" name="paddress" id="paddress" size="35"></td>
</tr>
<tr>
<td>Personal Address</td>
<td><input type="text" name="personaladdress"
id="personaladdress" size="35"></td>
</tr>
<tr>
<td>Sex</td>
<td><input type="radio" name="sex" value=
```

```
"male" size="10">Male
<input type="radio" name="sex" value="Female" size="10">Female</td>
</tr>
<tr>
<td>City</td>
<td><select name="City">
<option value="-1" selected>select..</option>
<option value="New Delhi">NEW DELHI</option>
<option value="Mumbai">MUMBAI</option>
<option value="Goa">GOA</option>
<option value="Patna">PATNA</option>
</select></td>
</tr>
<tr>
<td>Course</td>
<td><select name="Course">
<option value="-1" selected>select..</option>
<option value="B.Tech">B.TECH</option>
<option value="MCA">MCA</option>
<option value="MBA">MBA</option>
<option value="BCA">BCA</option>
</select></td>
</tr>
<tr>
<td>District</td>
<td><select name="District">
<option value="-1" selected>select..</option>
<option value="Nalanda">NALANDA</option>
<option value="UP">UP</option>
<option value="Goa">GOA</option>
<option value="Patna">PATNA</option>
</select></td>
</tr>
<tr>
<td>State</td>
<td><select Name="State">
<option value="-1" selected>select..</option>
<option value="New Delhi">NEW DELHI</option>
<option value="Mumbai">MUMBAI</option>
<option value="Goa">GOA</option>
<option value="Bihar">BIHAR</option>
</select></td>
</tr>
<tr>
<td>PinCode</td>
```

```

<td><input type="text" name="pincode" id="pincode" size="35"></td>
</tr>
<tr>
<td>EmailId</td>
<td><input type="text" name="emailid" id="emailid" size="35"></td>
</tr>
<tr>
<td>DOB</td>
<td><input type="text" name="dob" id="dob" size="35"></td>
</tr>
<tr>
<td>MobileNo</td>
<td><input type="text" name="mobilenos" id="mobilenos" size="35"></td>
</tr>
<tr>
<td><input type="reset"></td>
<td colspan="2"><input type="submit" value="Submit Form" /></td>
</tr>
</table>
</form>
</body>
</html>

```

Output:-

Student Registration Form	
Name	<input type="text"/>
Father Name	<input type="text"/>
Postal Address	<input type="text"/>
Personal Address	<input type="text"/>
Sex	<input type="radio"/> Male <input type="radio"/> Female
City	<input type="text" value="select..."/>
Course	<input type="text" value="select..."/>
District	<input type="text" value="select..."/>
State	<input type="text" value="select..."/>
PinCode	<input type="text"/>
EmailId	<input type="text"/>
DOB	<input type="text"/>
MobileNo	<input type="text"/>
Reset	Submit Form

PROGRAM- 3

Objective: Write program using Java script for Web Page to display browsers information.

Code:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<title>Browser Information</title>
<script language=javascript>
function show()
{
document.write("Name "+navigator.appName+"<br>");
document.write("Version "+navigator.appVersion "<br>");
document.write("Codename "+navigator.appCodeName "<br>");
document.write("Cookie enable"+navigator.cookieEnabled "<br>");
document.write("Java Enable"+navigator.javaEnabled "<br>");
document.write("Mime type"+navigator.mimeTypes "<br>");
document.write("Platform"+navigator.platform "<br>");
document.write("Plug ins"+navigator.plugins "<br>");
document.write("System Language"+navigator.systemLanguage "<br>");
document.write("User language"+navigator.userAgent "<br>");
document.write("User Agent"+navigator.userAgent "<br>");
}
</script>
</head>
<body>
<form id="form1">
<div>
<input id="Button1" type="button" value="Click me" onclick="show()" />
</div>
</form>
</body>
</html>
```

Output: -

Name Netscape

Version 5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36

Codename Mozilla

Cookie enabletrue

Java Enablefunction javaEnabled() { [native code] }

Mime type[object MimeTypeArray]

PlatformWin32

Plug ins[object PluginArray]

System Languageundefined

User languageMozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36

User AgentMozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36

PROGRAM- 4

Objective: A super class Detail has been defined to store the details of a customer. Define a sub class Bill to compute the monthly telephone charge of the customer as per the chart given below:

NUMBER OF CALLS	RATE
1-100	only rental Charge
101-200	60 paisa per call + rental charge
201-300	80 paisa per call + rental charge
Above 300	1 rupee per call + rental charge

Code:-

```
import java.util.Scanner;
class Detail {
    protected String name;
    protected String address;
    protected String telno;
    protected float rent;

    public Detail(String name, String address, String telno, float rent) {
        this.name = name;
        this.address = address;
        this.telno = telno;
        this.rent = rent;
    }

    public void show() {
        System.out.println("Name: " + name);
        System.out.println("Address: " + address);
        System.out.println("Telephone number: " + telno);
        System.out.println("Monthly rental charge: " + rent);
    }
}
class Bill extends Detail {
    private int n;
    private float amt;
    public Bill(String name, String address, String telno, float rent, int n) {
        super(name, address, telno, rent);
        this.n = n;
        amt = 0.0f;
    }

    public void cal() {
        if (n <= 100) {
            amt = rent;
        } else if (n <= 200) {
            amt = rent + (0.60f * (n - 100));
        } else if (n <= 300) {amt = rent + (0.80f * (n - 200)) + 60.0f;}
        else {
            amt = rent + (1.0f * (n - 300)) + 140.0f;
        }
    }
    public void show() {
```

```

        super.show();
        System.out.println("Number of calls: " + n);
        System.out.println("Amount to be paid: " + amt);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter customer name: ");
        String name = scanner.nextLine();
        System.out.print("Enter customer address: ");
        String address = scanner.nextLine();
        System.out.print("Enter customer telephone number: ");
        String telno = scanner.nextLine();
        System.out.print("Enter monthly rental charge: ");
        float rent = scanner.nextFloat();
        System.out.print("Enter number of calls: ");
        int n = scanner.nextInt();
        Bill b = new Bill(name, address, telno, rent, n);
        b.cal();
        b.show();
        scanner.close();
    }
}

```

Output :

```

PS C:\Users\harsh\OneDrive\Desktop\Java> javac Main.java
PS C:\Users\harsh\OneDrive\Desktop\Java> java Main
Enter customer name: John Doe
Enter customer address: 123 Main St, Anytown USA
Enter customer telephone number: 555-555-1234
Enter monthly rental charge: 75.0
Enter number of calls: 250
Name: John Doe
Address: 123 Main St, Anytown USA
Telephone number: 555-555-1234
Monthly rental charge: 75.0
Number of calls: 250
Amount to be paid: 175.0
PS C:\Users\harsh\OneDrive\Desktop\Java>

```

PROGRAM- 5

Objective: Write a Java applet/AWT to display the ApplicationProgram screen i.e. calculator and other.

Code:-

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*
<applet code="Cal" width=300 height=300>
</applet>
*/
public class Cal extends Applet
    implements ActionListener {
    String msg = " ";
    int v1, v2, result;
    TextField t1;
    Button b[] = new Button[10];
    Button add, sub, mul, div, clear, mod, EQ;
    char OP;

    public void init() {
        Color k = new Color(120, 89, 90);
        setBackground(k);
        t1 = new TextField(10);
        GridLayout gl = new GridLayout(4, 5);
        setLayout(gl);
        for (int i = 0; i < 10; i++) {

            b[i] = new Button("" + i);
        }
        add = new Button("add");
        sub = new Button("sub");
        mul = new Button("mul");
        div = new Button("div");
        mod = new Button("mod");
        clear = new Button("clear");
        EQ = new Button("EQ");
        t1.addActionListener(this);
        add(t1);
        for (int i = 0; i < 10; i++) {
            add(b[i]);
        }
        add(add);
        add(sub);
        add(mul);
        add(div);
        add(mod);
        add(clear);
        add(EQ);
        for (int i = 0; i < 10; i++) {
            b[i].addActionListener(this);
        }
    }
}
```

```

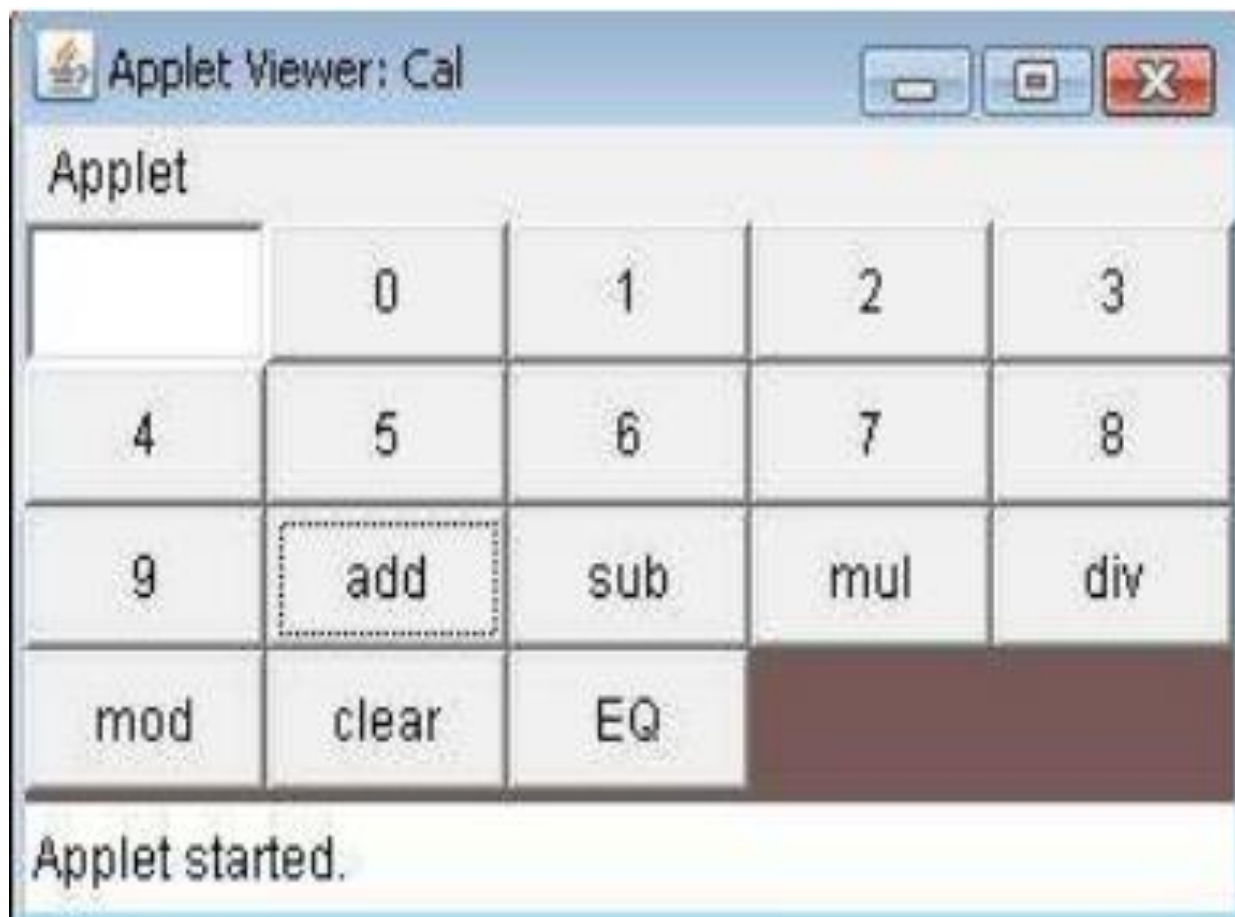
        add.addActionListener(this);
        sub.addActionListener(this);
        mul.addActionListener(this);
        div.addActionListener(this);
        mod.addActionListener(this);
        clear.addActionListener(this);
        EQ.addActionListener(this);
    }

    public void actionPerformed(ActionEvent ae) {
        String str = ae.getActionCommand();
        char ch = str.charAt(0);
        if (Character.isDigit(ch))
            t1.setText(t1.getText() + str);
        else if (str.equals("add")) {
            v1 = Integer.parseInt(t1.getText());
            OP = '+';

            t1.setText("");
        } else if (str.equals("sub")) {
            v1 = Integer.parseInt(t1.getText());
            OP = '-';
            t1.setText("");
        } else if (str.equals("mul")) {
            v1 = Integer.parseInt(t1.getText());
            OP = '*';
            t1.setText("");
        } else if (str.equals("div")) {
            v1 = Integer.parseInt(t1.getText());
            OP = '/';
            t1.setText("");
        } else if (str.equals("mod")) {
            v1 = Integer.parseInt(t1.getText());
            OP = '%';
            t1.setText("");
        }
        if (str.equals("EQ")) {
            v2 = Integer.parseInt(t1.getText());
            if (OP == '+')
                result = v1 + v2;
            else if (OP == '-')
                result = v1 - v2;
            else if (OP == '*')
                result = v1 * v2;
            else if (OP == '/')
                result = v1 / v2;
            else if (OP == '%')
                result = v1 % v2;
            t1.setText("" + result);
        }
        if (str.equals("clear")) {
            t1.setText("");
        }
    }
}

```

Output:-



PROGRAM- 6

Objective: Write a Java applet to display the Application Program screen i.e. Colour mixer pallet

Code:-

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class ColorMixer extends Applet implements AdjustmentListener {
    Scrollbar redSlider, greenSlider, blueSlider;
    Label redLabel, greenLabel, blueLabel;
    Panel colorPanel;

    public void init() {
        setLayout(new BorderLayout());

        // Create sliders
        redSlider = new Scrollbar(Scrollbar.HORIZONTAL, 0, 1, 0, 255);
        greenSlider = new Scrollbar(Scrollbar.HORIZONTAL, 0, 1, 0, 255);
        blueSlider = new Scrollbar(Scrollbar.HORIZONTAL, 0, 1, 0, 255);

        // Add listeners
        redSlider.addAdjustmentListener(this);
        greenSlider.addAdjustmentListener(this);
        blueSlider.addAdjustmentListener(this);

        // Create labels
        redLabel = new Label("Red: 0");
        greenLabel = new Label("Green: 0");
        blueLabel = new Label("Blue: 0");

        // Create panel for sliders and labels
        Panel controlPanel = new Panel();
        controlPanel.setLayout(new GridLayout(3,2));
        controlPanel.add(redLabel);
        controlPanel.add(redSlider);
        controlPanel.add(greenLabel);
        controlPanel.add(greenSlider);
        controlPanel.add(blueLabel);
        controlPanel.add(blueSlider);

        // Create panel for displaying color
        colorPanel = new Panel();
        colorPanel.setBackground(Color.black);

        // Add panels to applet
        add(controlPanel, BorderLayout.NORTH);
        add(colorPanel, BorderLayout.CENTER);
    }

    public void adjustmentValueChanged(AdjustmentEvent e) {
        int redValue = redSlider.getValue();
        int greenValue = greenSlider.getValue();
        int blueValue = blueSlider.getValue();
    }
}
```

```
redLabel.setText("Red: " + redValue);
greenLabel.setText("Green: " + greenValue);
blueLabel.setText("Blue: " + blueValue);

colorPanel.setBackground(new Color(redValue, greenValue, blueValue));
}
}
```


PROGRAM- 7

Objective: Write a program using TCP/IP socket between client and server and perform two-way communication

Code:-

Server.java

```
import java.net.*;
import java.io.*;
public class Server {
    public static void main(String[] args) {
        try {
            // Create server socket
            ServerSocket serverSocket = new ServerSocket(5000);
            System.out.println("Server started...");

            // Accept client connection
            Socket clientSocket = serverSocket.accept();
            System.out.println("Client connected...");

            // Create input and output streams
            InputStream inputStream = clientSocket.getInputStream();
            OutputStream outputStream = clientSocket.getOutputStream();

            // Create input and output readers
            BufferedReader inputReader = new BufferedReader(new InputStreamReader(inputStream));
            PrintWriter outputWriter = new PrintWriter(outputStream, true);

            // Read and write data
            String inputLine, outputLine;
            while ((inputLine = inputReader.readLine()) != null) {
                System.out.println("Client: " + inputLine);
                outputLine = "Server received: " + inputLine;
                outputWriter.println(outputLine);
                if (inputLine.equals("Bye")) {
                    break;
                }
            }
            // Close everything
            inputReader.close();
            outputWriter.close();
            clientSocket.close();
            serverSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Client.java

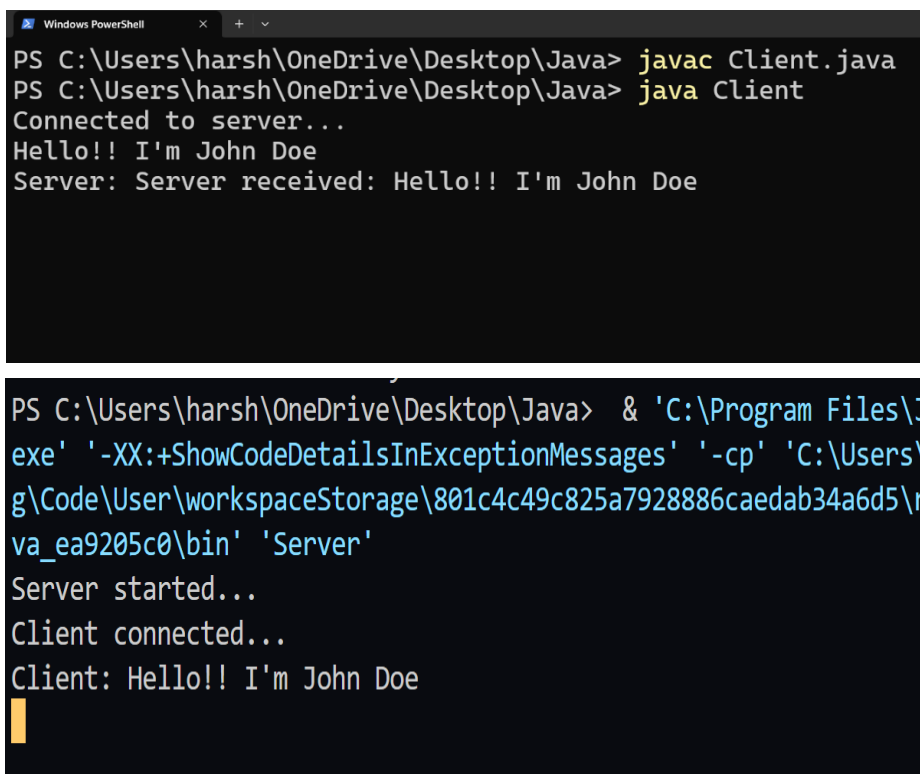
```
import java.net.*;
import java.io.*;
```

```

public class Client {
    public static void main(String[] args) {
        try {
            // Create client socket
            Socket clientSocket = new Socket("localhost", 5000);
            System.out.println("Connected to server...");
            // Create input and output streams
            InputStream inputStream = clientSocket.getInputStream();
            OutputStream outputStream = clientSocket.getOutputStream();
            // Create input and output readers
            BufferedReader inputReader = new BufferedReader(new InputStreamReader(inputStream));
            PrintWriter outputWriter = new PrintWriter(outputStream, true);
            // Read and write data
            BufferedReader userInputReader = new BufferedReader(new InputStreamReader(System.in));
            String userInput, serverResponse;
            while ((userInput = userInputReader.readLine()) != null) {
                outputWriter.println(userInput);
                if ((serverResponse = inputReader.readLine()) != null) {
                    System.out.println("Server: " + serverResponse);
                }
                if (userInput.equals("Bye")) {
                    break;
                }
            }
            // Close everything
            userInputReader.close();
            inputReader.close();
            outputWriter.close();
            clientSocket.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```

Output:-



The first screenshot shows the compilation and execution of the client program. The user runs 'javac Client.java' and 'java Client'. The output shows 'Connected to server...' followed by the user input 'Hello!! I'm John Doe' and the server response 'Server: Server received: Hello!! I'm John Doe'.

The second screenshot shows the execution of a server program. The user runs a command to start the server. The output shows 'Server started...', 'Client connected...', and 'Client: Hello!! I'm John Doe'.

```

PS C:\Users\harsh\OneDrive\Desktop\Java> javac Client.java
PS C:\Users\harsh\OneDrive\Desktop\Java> java Client
Connected to server...
Hello!! I'm John Doe
Server: Server received: Hello!! I'm John Doe

PS C:\Users\harsh\OneDrive\Desktop\Java> & 'C:\Program Files\Java\jdk-9.0.4\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\harsh\OneDrive\Desktop\Java\Code\User\workspaceStorage\801c4c49c825a7928886caedab34a6d5\workspace\va_ea9205c0\bin' 'Server'
Server started...
Client connected...
Client: Hello!! I'm John Doe

```

Program 8

Objective: Write a program to illustrate JDBC connectivity and perform CRUD operation on a table student/employee (at least 5 attributes).

Code:

```
package webtech;
import java.sql.*;
import java.io.*;
import oracle.jdbc.driver.DBConversion;

public class Jdbc {
    static boolean db_created=false;
    public static void main(String args[])
    { Connection con;
      Statement state;
      ResultSet rs;
      int no;
      int sal;
      String name;
      int ch;
      try
      {
          Class.forName("oracle.jdbc.driver.OracleDriver");
          System.out.println("Driver Loaded");

con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","DIVYA","divya");
do
    { System.out.println("\n");
      System.out.println("Menu:");
      System.out.println("1.Create table");
      System.out.println("2.Insert Record into the Table");
      System.out.println("3.Update The Existing Record.");
      System.out.println("4.Display all the Records from the Table");
      System.out.println("5.Delete");
      System.out.println("6.Exit");
      System.out.println("Enter your choice: ");

      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      ch=Integer.parseInt(br.readLine());

      String sql;

      switch(ch)
      { case 1:
        Statement stmt=con.createStatement();
```

```

String ct="create table employee(emp_id number(5),emp_name varchar2(30),emp_sal
number(8,2))";
boolean b=stmt.execute(ct);
if(b==false)    System.out.println("table not created");
else            {db_created=true;}
    stmt.close();
break;
case 2:
    if(db_created==true){
        state=con.createStatement();
        int i,n;
        System.out.println("enter the no of values you want to insert");
        n=Integer.parseInt(br.readLine());
        for(i=0;i<n;i++)
        {
            System.out.println("Enter Employee Number: ");
            no=Integer.parseInt(br.readLine());
            System.out.println("Enter Employee Name: ");
            name=br.readLine();
            System.out.println("Enter Employee Salary: ");
            sal=Integer.parseInt(br.readLine());
            sql="insert into employee values(?,?,?)";
            PreparedStatement p=con.prepareStatement(sql);
            p.setInt(1,no);
            p.setString(2,name);
            p.setInt(3,sal);
            p.executeUpdate();
            System.out.println("Record Added");
            p.close();
        }
    }
    else
        System.out.println("create table first");
    break;

case 3:
    if(db_created==true){
        state=con.createStatement();
        System.out.println("Enter Employee Number for the record you wish to Update: ");
        no = Integer.parseInt(br.readLine());
        System.out.println("Enter new Name: ");
        name = br.readLine();
        System.out.println("Enter new Salary: ");
        sal = Integer.parseInt(br.readLine());
        sql="update employee set Name=?, Salary=? where Code=?";
        PreparedStatement p=con.prepareStatement(sql);
        p.setString(1,name);
        p.setInt(2,sal);
        p.setInt(3,no);
        p.executeUpdate();
    }

```

```

        System.out.println("Record Updated");
        p.close();
    }
    else
        System.out.println("create table first");
    break;

    case 4:
        if(db_created==true){
            state=con.createStatement();
            sql="select * from employee";
            rs=state.executeQuery(sql);
            while(rs.next())
            {
                System.out.println("\n");
                System.out.print("\t" +rs.getInt(1));
                System.out.print("\t" +rs.getString(2));
                System.out.print("\t" +rs.getInt(3));
            }
        }
        else
            System.out.println("create table first");
        break;

    case 5:
        if(db_created==true){
            state=con.createStatement();
            sql="delete from employee where emp_id='1'";
            rs=state.executeQuery(sql);
            state.close();
        }
        else
            System.out.println("create table first");
        break;
    case 6:
        System.exit(1);
        break;
    default:
        System.out.println("Invalid Choice");
        break;
    }
} while(ch!=6);
con.close();
} catch(Exception e)
{
    System.out.println(e);
}
}
}

```

Output:

```
7 10 7 (SELECT COUNT(*) FROM Employee) 5)
+---+-----+-----+-----+-----+
| ID | NAME | PHONE | EMAIL | DATE_OF_JOINING |
+---+-----+-----+-----+-----+
| 6 | Sam | 6 | sam@mail.com | 2020-03-03 |
| 7 | Ron | 7 | ron@mail.com | 2020-05-16 |
| 8 | Sara | 8 | sara@mail.com | 2020-07-01 |
| 9 | Zara | 9 | zara@mail.com | 2020-08-20 |
| 10 | Yoji | 10 | yoji@mail.com | 2021-03-10 |
+---+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Program 9

Objective: Write a program to illustrate Batch Transaction with prepared statements.

Code:

BatchUpdateDemo_Connection.java

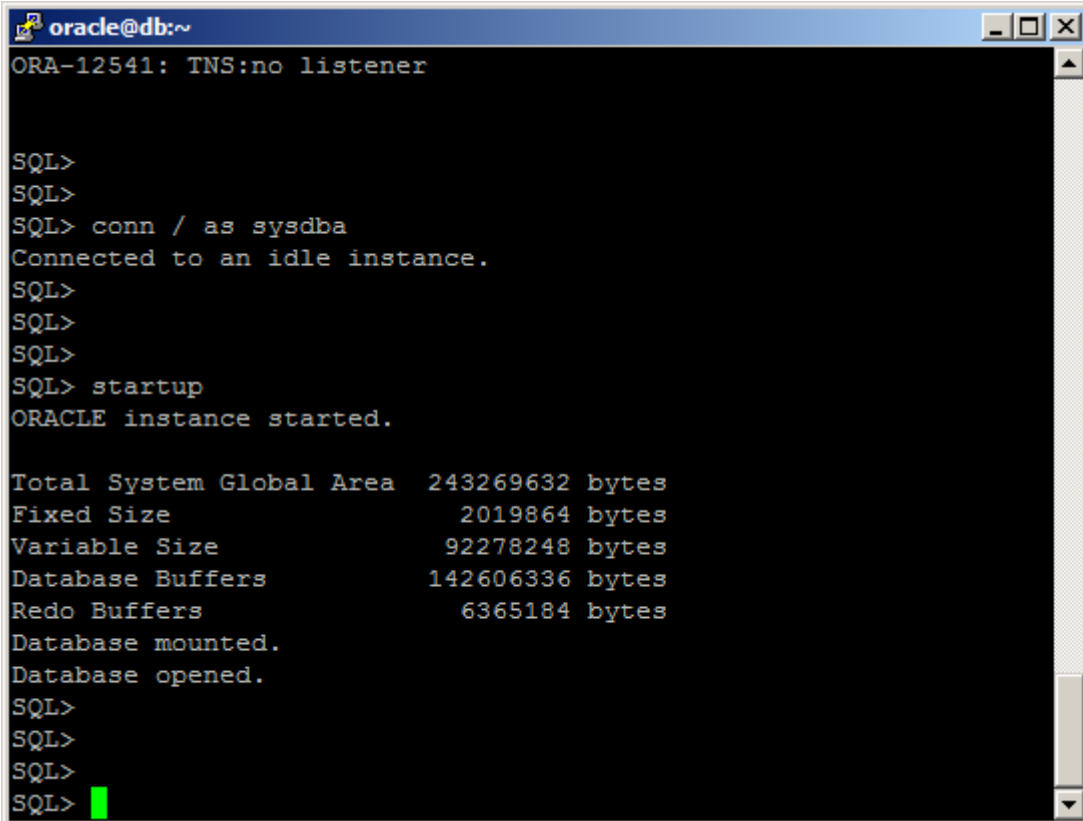
```
package webtech;
import java.sql.*;
public class BatchUpdateDemo_Connection {
    public static Connection getOracleConnection()throws Exception{
        String driver="oracle.jdbc.driver.OracleDriver";
        String URL="jdbc:oracle:thin:@localhost:1521:xe";
        String Username="DIVYA";
        String password="divya";
        Class.forName(driver);
        return DriverManager.getConnection(URL,Username,password); } }
```

BatchUpdate.java

```
package webtech;
import java.sql.*;
public class BatchUpdate {
    static Savepoint sp1;
    public static void main(String args[]) throws Exception
    {
        Connection con=BatchUpdateDemo_Connection.getOracleConnection();
        Statement stmt=null;
        try{
            con.setAutoCommit(false);
            stmt=con.createStatement();
            stmt.addBatch("delete from emplo");
            stmt.addBatch("insert into emplo values(104,'ABC',5694)");
            stmt.addBatch("insert into emplo values(105,'DEF',6694)");
            int[] uc=stmt.executeBatch();
            sp1 = con.setSavepoint("spv1");
            stmt.addBatch("insert into emplo values(106,'GHI',7694)");
            stmt.addBatch("insert into emplo2 values(107,'JKL',8694)");
            int[] uc1=stmt.executeBatch();
            System.out.println("...."+uc.length);
            for(int i=0;i<uc.length;i++)
            {
                System.out.println(uc[i]);
            }
            con.commit();
        }
        catch(BatchUpdateException e)
        {
            System.out.println(e);
            int[] abc=e.getUpdateCounts();
            System.out.println("...."+abc.length);
            for(int i=0;i<abc.length;i++)
            {
                System.out.println(abc[i]);
            }
            con.rollback(sp1);
            con.commit();
        }
    }
}
```

Output:

```
D:\Java_Dev\WEB\java2s>java BatchUpdate
ClassNotFoundException: myDriver.ClassName
SQLException: No suitable driver
SQLState: 08001
Message: No suitable driver
Vendor: 0
```

A screenshot of an Oracle SQL*Plus command window. The title bar shows 'oracle@db:~'. The window contains the following text:

```
ORA-12541: TNS:no listener

SQL>
SQL>
SQL> conn / as sysdba
Connected to an idle instance.
SQL>
SQL>
SQL>
SQL> startup
ORACLE instance started.

Total System Global Area  243269632 bytes
Fixed Size                 2019864 bytes
Variable Size             92278248 bytes
Database Buffers          142606336 bytes
Redo Buffers               6365184 bytes
Database mounted.
Database opened.
SQL>
SQL>
SQL>
SQL>
```


Program 10

Objective: Install a database (Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page.

Code:

- **index.html**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
  <h1><a href="RegisterForm.html">Register new user</a><br>
  <a href="Display.jsp">Display existing users</a></h1>

</body>
</html>
```

- **RegisterForm.html**

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head> <body>
  <form action="Register.jsp" method="post">
    <table>
      <tr> <td>Name:</td>
        <td><input type="text" name="name"></td> </tr>
      <tr> <td>Password:</td>
        <td><input type="text" name="password"></td> </tr>
      <tr> <td>Email:</td>
        <td><input type="email" name="email"></td> </tr>
      <tr> <td>Phone Number:</td>
        <td><input type="number" name="phone"></td> </tr>
      <tr> <td></td>
        <td><input type="submit" value="Register"></td> </tr>
    </table>
  </form>
</body> </html>
```

- **Register.jsp**

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ page import="java.sql.Statement" %>
<% @ page import="connection.Connect" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
    String name = request.getParameter("name");
    String password = request.getParameter("password");
    String email = request.getParameter("email");
    String phone = request.getParameter("phone");
    try {
        Statement stmt = Connect.conn().createStatement();
        String query = "insert into users1
values('"+name+"','"+password+"','"+email+"','"+phone+"')";
        stmt.executeQuery(query);
        Connect.conn().close();
        stmt.close();
        out.println("Registration Successfull");    }
catch(Exception e) {
    out.println("Registration Unsuccessfull!");
    System.out.println(e);    }
finally {
    out.println("<br><a href='index.html'>Go to home page</a>");    } %>
</body>
</html>
```

- **Display.jsp**

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ page import="java.sql.*" %>
<% @ page import="connection.Connect" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
```

```

<%      try{
Statement stmt = Connect.conn().createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
        ResultSet.CONCUR_UPDATABLE);
ResultSet rs = stmt.executeQuery("select * from users1");
if(!rs.next()){    out.println("No users registered!");        }
else{
%>
        <h1>Users are:</h1><br>
        <table border="1">
        <thead>
        <tr>
                <td>Name</td>
                <td>Password</td>
                <td>Email</td>
                <td>Phone</td>
        </tr>
        </thead>
        <tbody>
        <%
                rs.previous();
while (rs.next()) {
String name = rs.getString(1);
String password = rs.getString(2);
String email = rs.getString(3);
int phone = rs.getInt(4);%>
                <tr>
                        <td><%= name %></td>
                        <td><%= password %></td>
                        <td><%= email %></td>
                        <td><%= phone %></td>
                </tr>
        <% } }%>
        </tbody>
        </table>
        <% }
        catch(Exception e) {
                System.out.println(e);
        }
        finally {
                out.println("<br><a href='index.html'>Go to home page</a>");
        }
        %>
</body></html>

```

- **Connect.java**

```


package connection;
import java.sql.Connection;



```





```
import java.sql.DriverManager;
import java.sql.SQLException;

public class Connect {
    public static Connection conn() throws ClassNotFoundException, SQLException {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection conn
        DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akku1","akku1");
        return conn;
    }
}
```

Output:

 Insert title here

    File | C:/Users/91735/Desktop/demo.html

Name:

Password:

Email:

Phone Number:

Program 11

Objective: To Develop a student Mark sheet by using Servlet and HTML with database Oracle.

Code:

Creating Database RESULT:

```
CREATE TABLE "RESULT"
( "ROLLNO" NUMBER,
  "NAME" VARCHAR2(40),
  "RESULT" VARCHAR2(40),
  "GRADE" VARCHAR2(40),
  CONSTRAINT "RESULT_PK" PRIMARY KEY ("ROLLNO") ENABLE
)
/
```

index.html

```
<html>
<body>
<form action="servlet/Search">
Enter your Rollno:<input type="text" name="roll"/><br/>

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

Search.java

```
import java.io.*;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;

public class Search extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String rollno=request.getParameter("roll");
        int roll=Integer.valueOf(rollno);
```

```

try{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

PreparedStatement ps=con.prepareStatement("select * from result where rollno=?");
ps.setInt(1,roll);

out.print("<table width=50% border=1>");
out.print("<caption>Result:</caption>");

ResultSet rs=ps.executeQuery();

/* Printing column names */
ResultSetMetaData rsmd=rs.getMetaData();
int total=rsmd.getColumnCount();
out.print("<tr>");
for(int i=1;i<=total;i++)
{
out.print("<th>"+rsmd.getColumnName(i)+"</th>");
}

out.print("</tr>");

/* Printing result */

while(rs.next())
{
out.print("<tr><td>"+rs.getInt(1)+"</td><td>"+rs.getString(2)+"
</td><td>"+rs.getString(3)+"</td><td>"+rs.getString(4)+"</td></tr>");
}
out.print("</table>");
}catch (Exception e2) {e2.printStackTrace();}
finally{out.close();}

}
}

```

web.xml

```

<web-app>

<servlet>
<servlet-name>Search</servlet-name>

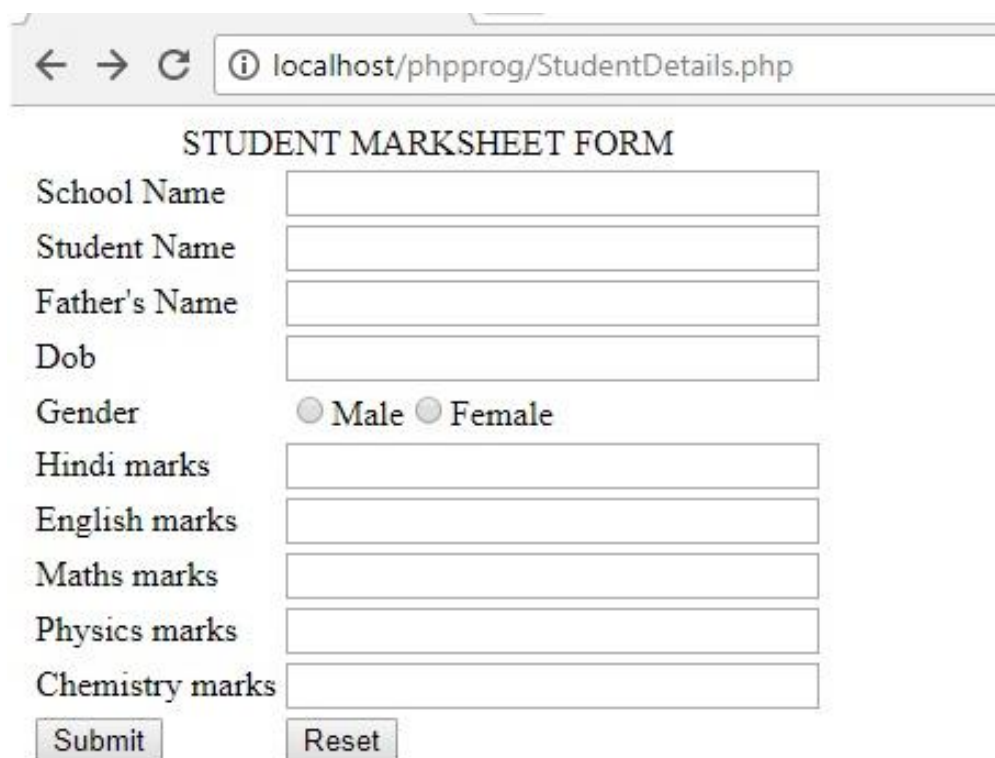
```

```
<servlet-class>Search</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>Search</servlet-name>
<url-pattern>/servlet/Search</url-pattern>
</servlet-mapping>

</web-app>
```

Output:



A screenshot of a web browser window displaying a form titled "STUDENT MARKSHEET FORM". The browser's address bar shows the URL "localhost/phpprog/StudentDetails.php". The form contains the following fields and controls:

- School Name:
- Student Name:
- Father's Name:
- Dob:
- Gender: ☐ Male ☐ Female
- Hindi marks:
- English marks:
- Maths marks:
- Physics marks:
- Chemistry marks:
- Submit:
- Reset:

Program 12

Objective: Design and implement a simple servlet for an entry form of student details and send it to store at database server like SQL, Oracle or MS Access.

Code:

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="/register" method="post">
<table>
<tr>
<td>Roll Number:</td>
<td><input type="number" name="roll"></td>
</tr>
<tr>
<td>student name:</td>
<td><input type="text" name="name"></td>
</tr>
<tr>
<td>Course:</td>
<td><input type="text" name="course"></td>
</tr>
<tr>
<td>branch:</td>
<td><input type="text" name="branch"></td>
</tr>
<tr>
<td></td>
<td><input type="submit" value="add"></td>
</tr>
</table>
</form>
</body>
</html>
```

Connect.java

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

public class Connect {
public static Connection conn() throws ClassNotFoundException, SQLException {
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection conn =
    DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akku1","akku1");
return conn;
}
```



```
}
```

AddInDatabase.java

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Statement;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

// Servlet implementation class AddInDatabase

public class AddInDatabase extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        // TODO Auto-generated method stub
        PrintWriter out = response.getWriter();
        String roll = request.getParameter("roll");
        String name = request.getParameter("name");
        String course = request.getParameter("course");
        String branch = request.getParameter("branch");
        try {
            Statement stmt = Connect.conn().createStatement();
            String query = "insert into student values("+roll+", '"+name+"', '"+course+"', '"+branch+"')";
            stmt.executeQuery(query);
            Connect.conn().close();
            stmt.close();
            out.println("Registration Successfull");
        } catch(java.sql.SQLIntegrityConstraintViolationException e) {
            out.println("Error resgistering this student. A student with same roll number exists!");
        }
        catch(Exception e) { out.println("Registration Unsuccessfull!");
            System.out.println(e);
        }
        finally {
            out.println("<br><a href='index.html'>Go to home
page</a>");
        }
    }
}
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-
app_2_4.xsd">
  <display-name>Servlet Store student details in database</display-name>
  <servlet>
    <description>
    </description>
    <display-name>AddInDatabase</display-name>
    <servlet-name>AddInDatabase</servlet-name>
    <servlet-class>AddInDatabase</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>AddInDatabase</servlet-name>
    <url-pattern>/register</url-pattern>
  </servlet-mapping>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

Output:

Student Form

Student Name:

Student Age:

Email ID:

Pin code:

Join Date:

are you agree terms and conditions? ☐

Program 13

Objective: Write a JSP which insert the details of the 3 or 4 users who register with the web site by using registration form.
Authenticate the user when he submits the login form using the user name and password from the database

Code:

Connect.java

```
package connection;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class Connect {
    public static Connection conn() throws ClassNotFoundException, SQLException {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection conn =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","akku1","akku1");
        return conn; }
}
```

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Login Page</h1>
<form action="Login.jsp" method="get"><br>
    <table>
        <tr>
            <td>Username</td>
            <td><input type="text" name="uname"></td>
        </tr>
        <tr>
            <td>Password</td>
            <td><input type="text" name="upass"></td>
        </tr>
        <tr>
            <td></td>
            <td><input type="submit" value="Login"></td>
        </tr>
        <tr>
            <td></td>
            <td><center><a href="RegisterForm.html">
New user? Register here.</a></center></td>
        </tr>
    </table>
</form>
</body>
</html>
```

RegisterForm.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<h1>Registration Page</h1>
    <form action="Register.jsp" method="post"><br>
        <table>
            <tr>
                <td>Name</td>
                <td><input type="text" name="name"></td>
            </tr>
            <tr>
                <td>Username</td>
                <td><input type="text" name="uname"></td>
            </tr>
            <tr>
                <td>Password</td>
                <td><input type="text" name="upass"></td>
            </tr>
            <tr>
                <td>Email</td>
                <td><input type="text" name="email"></td>
            </tr>
            <tr>
                <td>Phone Number</td>
                <td><input type="number" name="phone"></td>
            </tr>
            <tr>
                <td></td>
                <td><input type="submit" value="Register here"></td>
            </tr>
        </table>
    </form>
    <a href="index.html">Go back to home page</a>
</body>
</html>
```

Login.jsp

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"% >
<% @ page import="java.sql.*" %>
<% @ page import="connection.Connect" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
    try{boolean exist=false;
Statement stmt = Connect.conn().createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
    ResultSet rs = stmt.executeQuery("select * from users");
    if(!rs.next()){
```

```

        %>
        <h1>No users registered! Register first.</h1>
        <%
    }
    else{
        rs.previous();
        String uname = request.getParameter("uname");
        String password = request.getParameter("upass");
        while (rs.next()) {
            String name = rs.getString(2);
            String pass = rs.getString(3);
            if(uname.equals(name)&&password.equals(pass)) {
                exist=true;
                break;      } }
        if(exist){      %>
            <h1>You have successfully logged in <%= rs.getString(1) %></h1>
            <br>
            <strong>Your username is <%= rs.getString(2)%><br>
            Your email is <%= rs.getString(4)%><br>
            Your phone number is <%= rs.getInt(5)%></strong>
            <%          }
        else{
            %>
            <h1>Wrong username or password!</h1>
            <%
        } } }
    catch(Exception e) {
        System.out.println(e);
    }
    finally {
        out.println("<br><a href='index.html'>Go to home page</a>");}%>
</body>
</html>

```

Register.jsp

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<% @ page import="java.sql.Statement" %>
<% @ page import="connection.Connect" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Insert title here</title>
</head>
<body>
<%
String name = request.getParameter("name");
String uname = request.getParameter("uname");

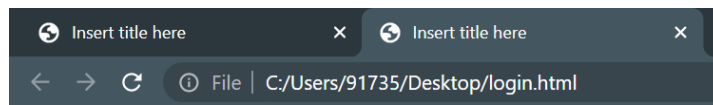
```

```

String password = request.getParameter("upass");
String email = request.getParameter("email");
String phone = request.getParameter("phone");
try {
    Statement stmt = Connect.conn().createStatement();
    String query = "insert into users
values('"+name+"','"+uname+"','"+password+"','"+email+"','"+phone+"')";
    stmt.executeQuery(query);
    Connect.conn().close();
    stmt.close();
    out.println("Registration Successfull");
}catch(java.sql.SQLIntegrityConstraintViolationException e) {
    out.println("Error resgistering this user. A student with same username exists!");
}
catch(Exception e) {
    out.println("Registration Unsuccessfull!");
    System.out.println(e);
}
finally {
    out.println("<br><a href='index.html'>Go to home page</a>");
} %>
</body>
</html>

```

Code:

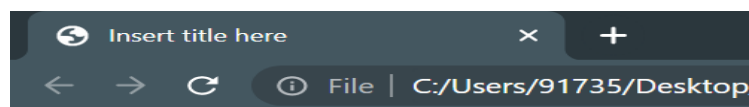


Login Page

Username

Password

[New user? Register here.](#)



Registration Page

Name

Username

Password

Email

Phone Number

[Go back to home page](#)

Program 14

Objective: Store 5 Students (name, branch, rollno, age) objects in list. Perform any sorting technique so as to display list in ascending order of rollno and display in descending order of their age.

Code:

```
import java.util.*;
public class StudentListTest
{
    public static void main(String[] args)
    {
        TreeSet<Student> s1 = new TreeSet<Student>(new RollComparator());
        TreeSet<Student> s2 = new TreeSet<Student>(new AgeComparator());

        s1.add(new Student("Ajay Singh","CSE",10001,21));
        s1.add(new Student("Ajay Sharma","ME",30001,22));
        s1.add(new Student("Ajay Yadav","EC",20001,23));
        s1.add(new Student("Ajay Jha ","EE",40001,22));

        s2.add(new Student("Ajay Singh","CSE",10001,21));
        s2.add(new Student("Ajay Sharma","ME",30001,24));
        s2.add(new Student("Ajay Yadav","EC",20001,23));
        s2.add(new Student("Ajay Jha ","EE",40001,22));

        Iterator i1 = s1.iterator();
        Iterator i2 = s2.iterator();

        System.out.println();      System.out.println();      System.out.println();
        System.out.println("According to Roll Number(Ascending Order)");

        while(i1.hasNext())
        {
            Student e = (Student) i1.next();
            System.out.println(e);
        }
        System.out.println();      System.out.println();      System.out.println();
        System.out.println("According to Age(Descending Order)");

        while(i2.hasNext())
        {
            Student e = (Student) i2.next();
            System.out.println(e);
        }
    }
}
```

```

class Student
{
    String name,branch;
    Integer roll,age;

    public Student(String name,String branch,Integer roll,Integer age)
    {
        this.name=name;
        this.branch=branch;
        this.roll=roll;
        this.age=age;
    }
    public String toString()
    {
        return "Name:- "+name+"\t Branch:- "+branch+"\t RollNo:- "+roll+"\t Age:-"+age;
    }
}
class RollComparator implements Comparator<Student>
{
    public int compare(Student s1,Student s2)
    {
        return s1.roll.compareTo(s2.roll);
    }
}
class AgeComparator implements Comparator<Student>
{
    public int compare(Student s1,Student s2)
    {
        return s2.age.compareTo(s1.age);
    }
}

```

```

G:\PSIT\WebTech>java StudentListTest

According to Roll Number(Ascending Order)
Name:- Ajay Singh      Branch:- CSE    RollNo:- 10001  Age:-21
Name:- Ajay Yadav      Branch:- EC     RollNo:- 20001  Age:-23
Name:- Ajay Sharma     Branch:- ME     RollNo:- 30001  Age:-22
Name:- Ajay Jha        Branch:- EE     RollNo:- 40001  Age:-22

According to Age(Descending Order)
Name:- Ajay Sharma     Branch:- ME     RollNo:- 30001  Age:-24
Name:- Ajay Yadav      Branch:- EC     RollNo:- 20001  Age:-23
Name:- Ajay Jha        Branch:- EE     RollNo:- 40001  Age:-22
Name:- Ajay Singh      Branch:- CSE    RollNo:- 10001  Age:-21

G:\PSIT\WebTech>_

```


PROGRAM 15

OBJECTIVE: Perform union (AUB), intersection and difference (A-B) operations using set collection.

CODE:

```
package com.corejava;

import java.util.*;

class SetOperations

{

    public static void main(String str[])

    {

        Collection<String> col1 = new TreeSet<String>();

        col1.add("a");
        col1.add("b");          col1.add("c");

        Collection<String> col2 = new TreeSet<String>();

        col2.add("b");
        col2.add("c");          col2.add("d");
        col2.add("e");

        /*For Union*/

        Collection<String> temp1 = new TreeSet<String>(col1);
        temp1.addAll(col2);

        System.out.println("Union of Set-A and Set-B "+temp1);
```

```
        /*For Intersection*/

        Collection<String> temp2 = new TreeSet<String>(col1);
temp2.retainAll(col2);

        System.out.println("Intersection of Set-A and Set-B "+temp2);


        /*For Difference*/

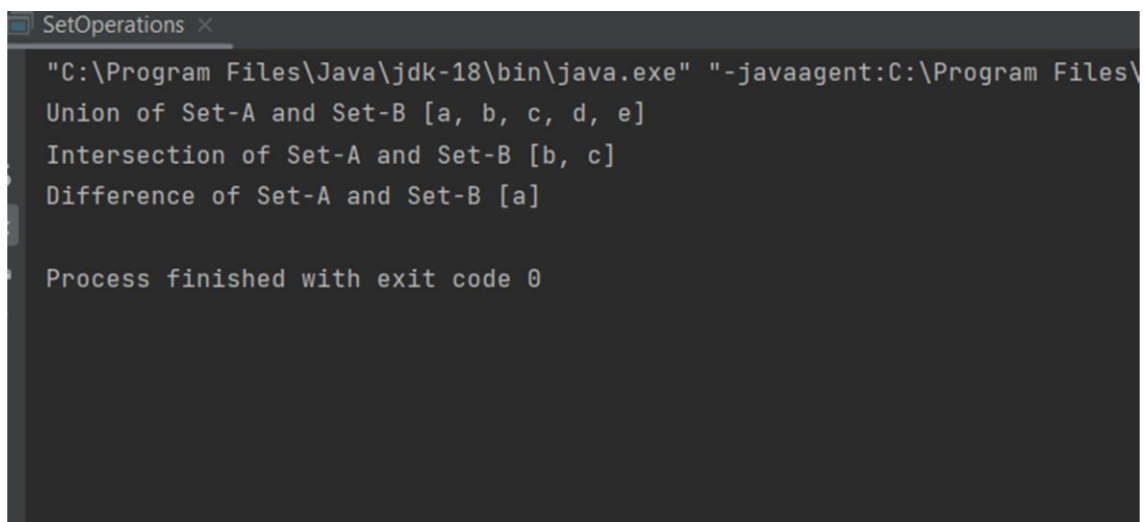
        Collection<String> temp3 = new TreeSet<String>(col1);
temp3.removeAll(col2);

        System.out.println("Difference of Set-A and Set-B "+temp3);

    }

}
```

Output-



```
SetOperations x
"C:\Program Files\Java\jdk-18\bin\java.exe" "-javaagent:C:\Program Files\
Union of Set-A and Set-B [a, b, c, d, e]
Intersection of Set-A and Set-B [b, c]
Difference of Set-A and Set-B [a]

Process finished with exit code 0
```

Program 16

OBJECTIVE: Create a Map that consists of Country-Capital pair to store information about various countries along with their capital name. Display the entries in sorted order of Country and Capital.

CODE:

```
package com.democollection;

import java.util.HashMap; import
java.util.LinkedHashMap; import
java.util.Map;

import java.util.stream.Stream;

public class MapSortByKeyExample {

    public static void main(String[] args) {

        Map<String, String> countryCapitalMap = new HashMap<String,
String>();
        countryCapitalMap.put("guyana", "georgetown");
        countryCapitalMap.put("nepal", "kathmandu");
        countryCapitalMap.put("australia", "canberra");
        countryCapitalMap.put("india", "new delhi");

        countryCapitalMap.put("japan", "tokyo");

        System.out.println("Original Map : \n" + countryCapitalMap);

        Map<String, String> sortedMap1 = new LinkedHashMap<String, String>();

        Stream<Map.Entry<String, String>> stream1 = countryCapitalMap.entrySet()

            .stream();
```

```
stream1.sorted(Map.Entry.comparingByKey()).forEachOrdered(  
c -> sortedMap1.put(c.getKey(), c.getValue()));
```

```
System.out.println("Map sorted by key : \n" + sortedMap1);
```

```
Map<String, String> sortedMap2 = new LinkedHashMap<String, String>();
```

```
Stream<Map.Entry<String, String>> stream2 = countryCapitalMap.entrySet()
```

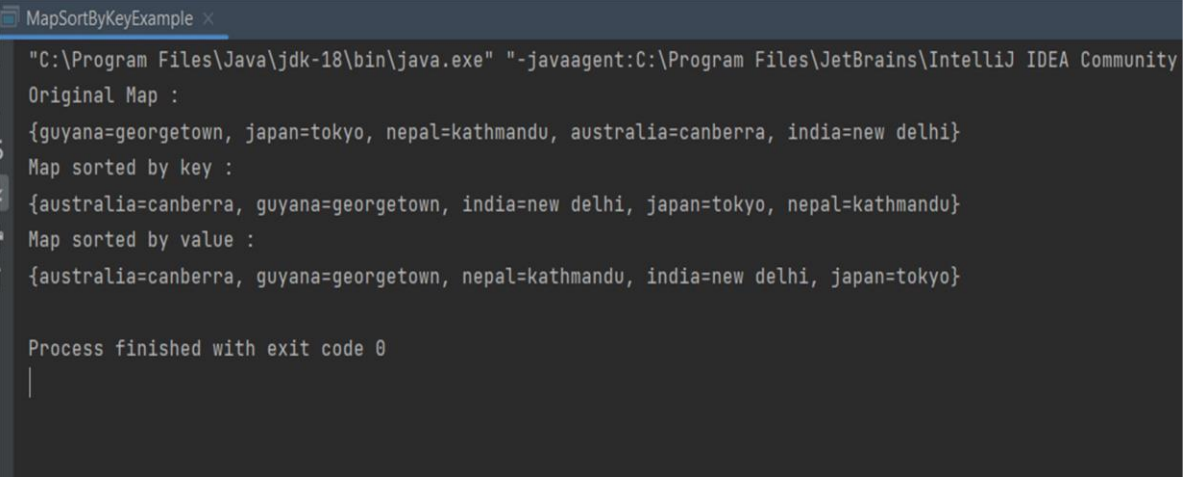
```
.stream();
```

```
stream2.sorted(Map.Entry.comparingByValue()).forEachOrdered(  
c -> sortedMap2.put(c.getKey(), c.getValue()));
```

```
System.out.println("Map sorted by value : \n" + sortedMap2);
```

```
}
```

```
}
```



```
MapSortByKeyExample x  
"C:\Program Files\Java\jdk-18\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community  
Original Map :  
{guyana=georgetown, japan=tokyo, nepal=kathmandu, australia=canberra, india=new delhi}  
Map sorted by key :  
{australia=canberra, guyana=georgetown, india=new delhi, japan=tokyo, nepal=kathmandu}  
Map sorted by value :  
{australia=canberra, guyana=georgetown, nepal=kathmandu, india=new delhi, japan=tokyo}  
  
Process finished with exit code 0  
|
```

OUTPUT —

