AKASH KUMAR BANIK 24MCA0242

PMCA502P - JAVA PROGRAMMING LAB

DATE: 13-SEP-2024

Q1: Develop three Java files to implement the serialization and deserialization. a. Create a Bank Account class which represents account number, name and balance amount. Design methods to get input and print the details. b. Create a Java file to serialize ‘n’ objects of the Bank Account class. c. Create another Java file to deserialize those ‘n’ objects and print the details along with the interest amount for the balance money in the account. 5% of interest to be offered for everyone.

CODE:

BankAccount.java FILE:

```
import java.util.Scanner;

import java.io.Serializable;

public class BankAccount implements Serializable {

    int accNo, balAmt = 0;

    String name;

    public void getData() {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Account Number: ");

        accNo = sc.nextInt();

        sc.nextLine();

        System.out.println("Enter the name: ");

        name = sc.nextLine();

        System.out.println("Enter Account Balance: ");

        balAmt = sc.nextInt();
    }
    public void printData() {
        System.out.println("----------Account Details----------");
        System.out.println("Account Number: " + accNo);
        System.out.println("Account Name: " + name);
        System.out.println("Account Balance: " + balAmt);
    }
}
```

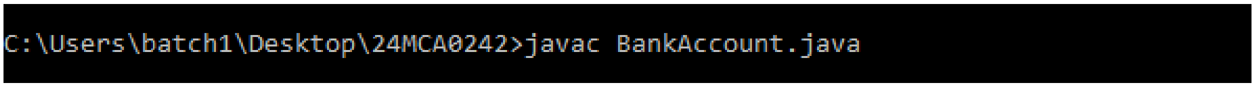
BankSerialize.java FILE:

```
import java.util.Scanner;
import java.io.*;
public class BankSerialize {
    public static void main(String[] args) throws IOException, FileNotFoundException, ClassNotFoundException {
        FileOutputStream fos = new FileOutputStream("bank_details.txt");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        int i, n;
        Scanner inp = new Scanner(System.in);
        System.out.println("Enter the number of Accounts: ");
        n = inp.nextInt();
        BankAccount[] objs = new BankAccount[n];

        for (i = 0; i < n; i++) {
            objs[i] = new BankAccount();
            objs[i].getData();
            oos.writeObject(objs[i]);
        }
        System.out.println("Data inserted into file bank_details.txt");
        oos.close();
        fos.close();
    }
}
```

BankDeserialize.java FILE:

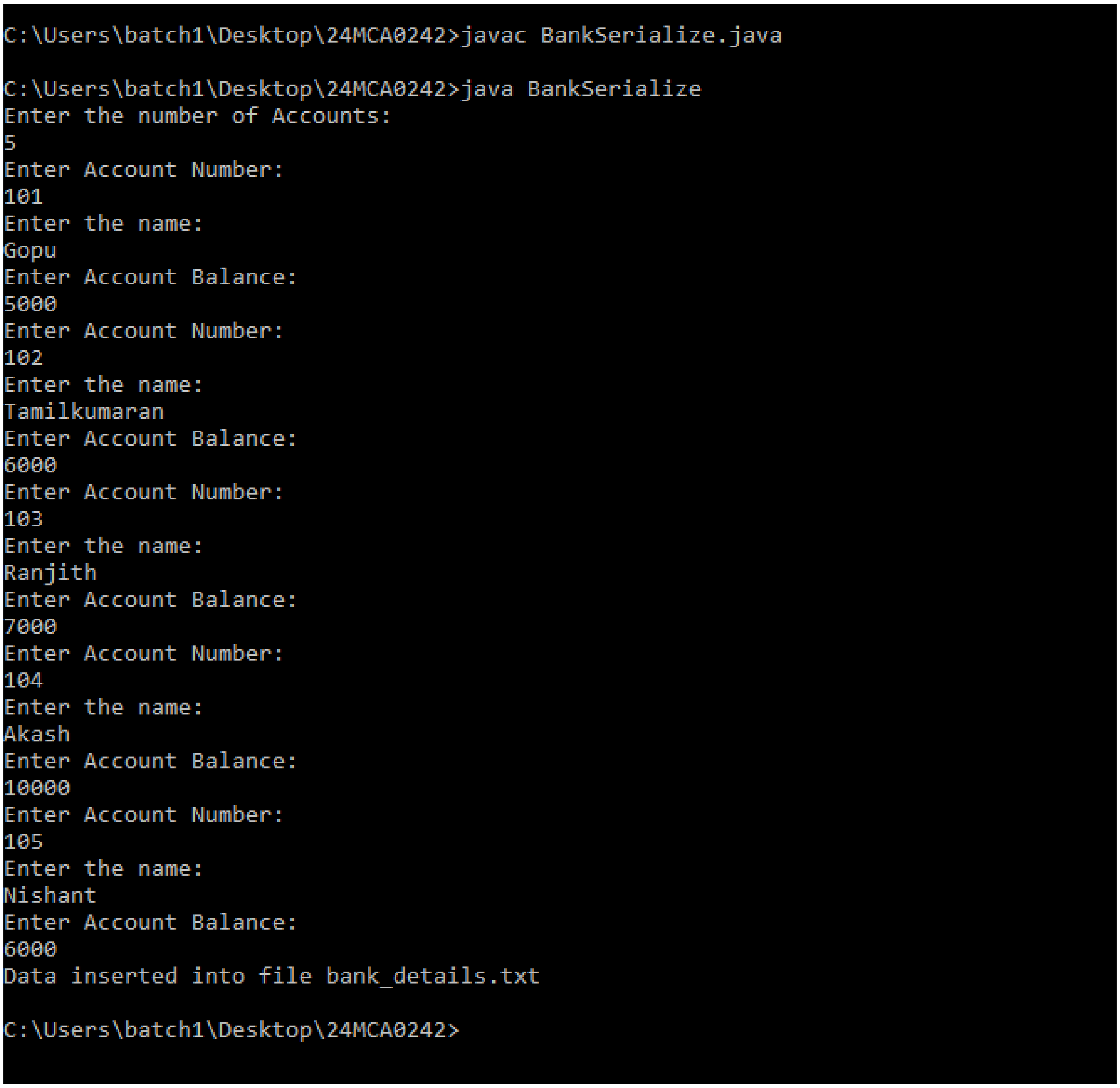
```
import java.util.Scanner;
import java.io.*;
public class BankDeSerialize {
    public static void main(String[] args) throws IOException, FileNotFoundException, ClassNotFoundException {
        FileInputStream fis = new FileInputStream("bank_details.txt");
        ObjectInputStream ois = new ObjectInputStream(fis);
        BankAccount obj1;
        System.out.println("Data read from file bank_details.txt");
        while (fis.available() > 0) {
            obj1 = (BankAccount) ois.readObject();
            obj1.printData();
        }
        ois.close();

        fis.close();

    }

}
```

OUTPUTS:



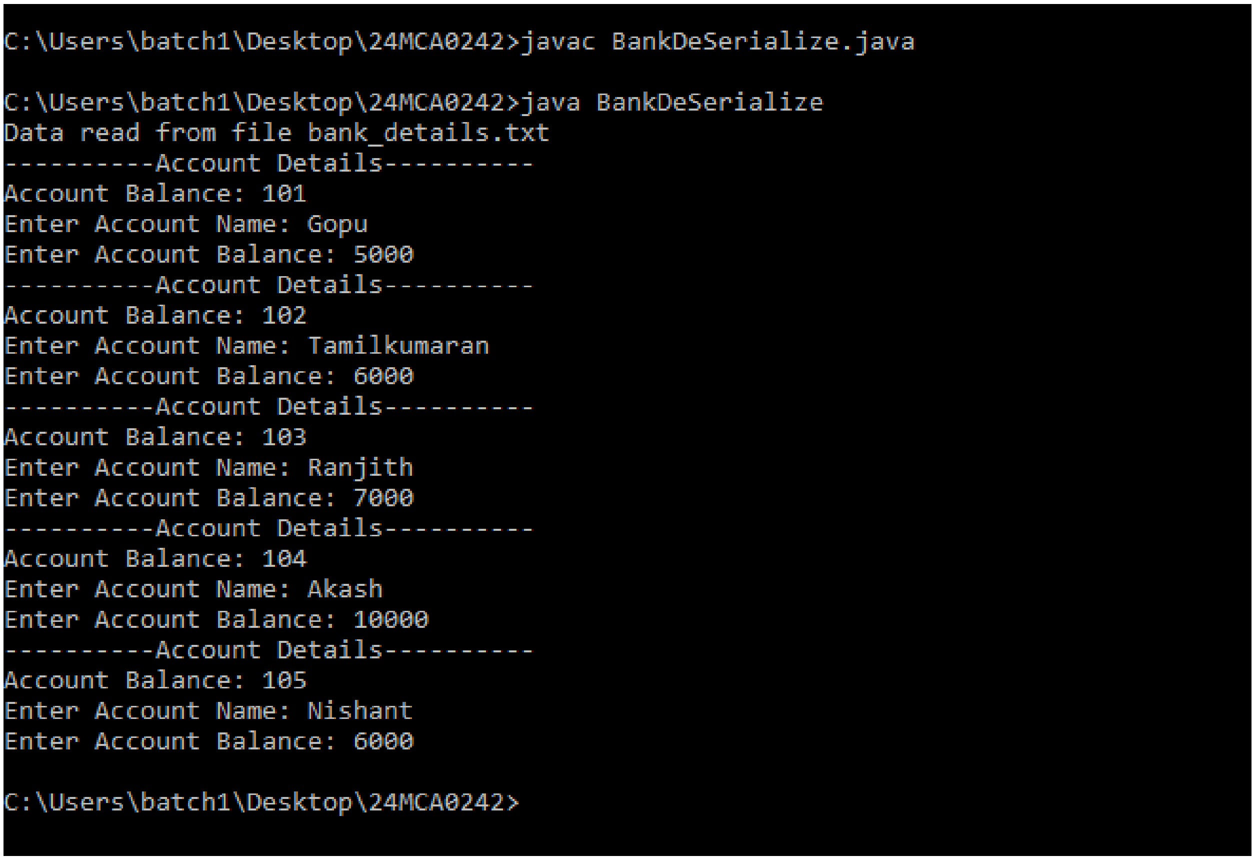
```
C:\Users\batch1\Desktop\24MCA0242>javac BankAccount.java
```

SERIALIZE:



```
C:\Users\batch1\Desktop\24MCA0242>javac BankSerialize.java

C:\Users\batch1\Desktop\24MCA0242>java BankSerialize
Enter the number of Accounts:
5
Enter Account Number:
101
Enter the name:
Gopu
Enter Account Balance:
5000
Enter Account Number:
102
Enter the name:
Tamilkumaran
Enter Account Balance:
6000
Enter Account Number:
103
Enter the name:
Ranjith
Enter Account Balance:
7000
Enter Account Number:
104
Enter the name:
Akash
Enter Account Balance:
10000
Enter Account Number:
105
Enter the name:
Nishant
Enter Account Balance:
6000
Data inserted into file bank_details.txt

C:\Users\batch1\Desktop\24MCA0242>
```

DESERIALIZE:



```
C:\Users\batch1\Desktop\24MCA0242>javac BankDeSerialize.java

C:\Users\batch1\Desktop\24MCA0242>java BankDeSerialize
Data read from file bank_details.txt
----------Account Details----------
Account Balance: 101
Enter Account Name: Gopu
Enter Account Balance: 5000
----------Account Details----------
Account Balance: 102
Enter Account Name: Tamilkumaran
Enter Account Balance: 6000
----------Account Details----------
Account Balance: 103
Enter Account Name: Ranjith
Enter Account Balance: 7000
----------Account Details----------
Account Balance: 104
Enter Account Name: Akash
Enter Account Balance: 10000
----------Account Details----------
Account Balance: 105
Enter Account Name: Nishant
Enter Account Balance: 6000

C:\Users\batch1\Desktop\24MCA0242>
```

Q2: Develop a package called mathprocess which has a class to do the following. a. Define a static method to find factorial for the given parameter and it returns it. b. Define another non-static method which takes array of integer as parameter and finds the average of the array. It returns the average value to the invoked program. Design a test program in Java to import this package and demonstrate the methods.

Code:

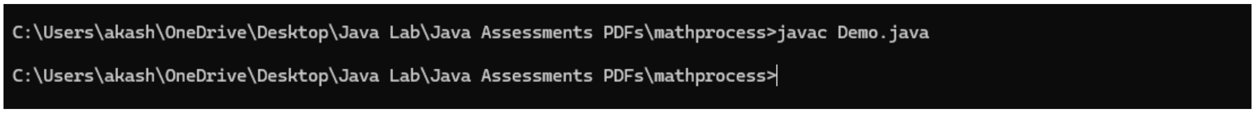
Demo.java File [PACKAGE]:

```
package mathprocess;

public class Demo {
    public static int getFactorial(int n) {
        int result = 1;
        for (int i = 1; i <= n; i++) {
            result *= i;
        }
        return result;
    }
    public double getAverage(int[] arr) {
        int sum = 0;
        for (int i = 0; i < arr.length; i++) {
            sum += arr[i];
        }
        double average = (double) sum / arr.length;
        return average;
    }
}
```

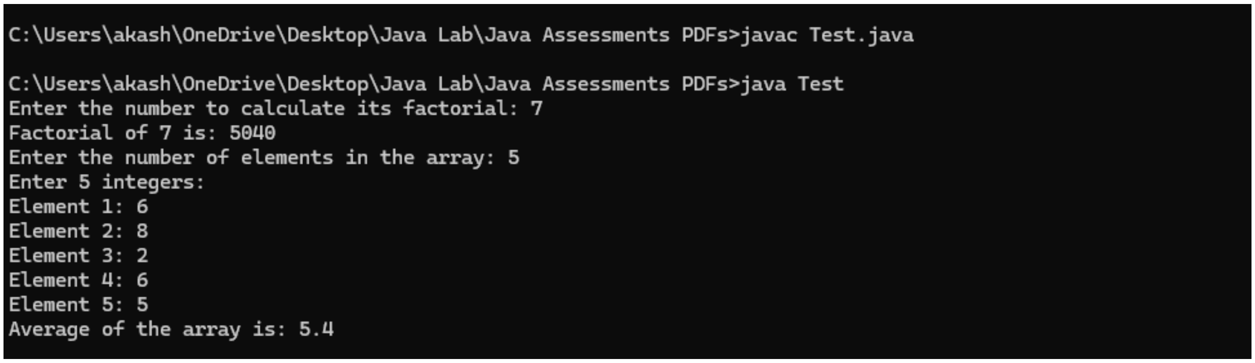
Test.java File:

```
import mathprocess.Demo;
import java.util.Scanner;
public class Test {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number to calculate its factorial: ");

        int num = sc.nextInt();

        int factorial = Demo.getFactorial(num);

        System.out.println("Factorial of " + num + " is: " + factorial);

        System.out.print("Enter the number of elements in the array: ");

        int size = sc.nextInt();

        int[] numbers = new int[size];

        System.out.println("Enter " + size + " integers:");

        for (int i = 0; i < size; i++) {

          System.out.print("Element " + (i + 1) + ": ");

          numbers[i] = sc.nextInt();

        }

        Demo demo = new Demo();

        double average = demo.getAverage(numbers);

        System.out.println("Average of the array is: " + average);

    }

}
```

OUTPUTS:



```
C:\Users\akash\OneDrive\Desktop\Java Lab\Java Assessments PDFs\mathprocess>javac Demo.java

C:\Users\akash\OneDrive\Desktop\Java Lab\Java Assessments PDFs\mathprocess>
```



```
C:\Users\akash\OneDrive\Desktop\Java Lab\Java Assessments PDFs>javac Test.java

C:\Users\akash\OneDrive\Desktop\Java Lab\Java Assessments PDFs>java Test
Enter the number to calculate its factorial: 7
Factorial of 7 is: 5040
Enter the number of elements in the array: 5
Enter 5 integers:
Element 1: 6
Element 2: 8
Element 3: 2
Element 4: 6
Element 5: 5
Average of the array is: 5.4
```

AKASH KUMAR BANIK

24MCA0242

DATE: 20-SEP-2024

1. Create a table in Derby back end to represent employee details like id, name, gender, BP (Baisc Pay), DA(Dearness Allowance), HRA(House Rent Allowance) in percentage (the value must be between 1 and 100), hobbies of employee.
2. Develop a console based Java application to insert ‘n’ records into the table. Collect the input for the table from users. (Note: Should include ‘derbyclient.jar’ in libraries under the project. Should start the JavaDB server under ‘Services’)

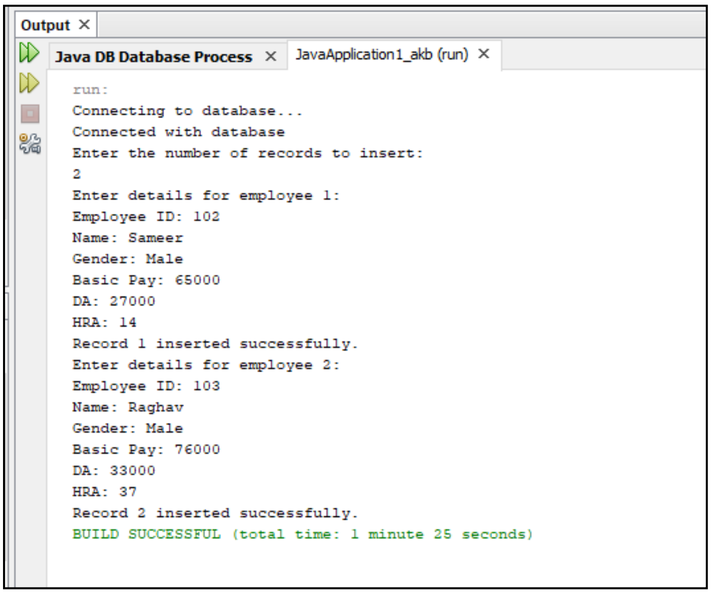
CODE 1 and 2:

```
package javaapplication1_akb;

import java.sql.*;

import java.util.Scanner;

public class JavaApplication1_akb {

    static String url = "jdbc:derby://localhost:1527/akb_db";

    static String uid = "akb";

    static String pwd = "akb";

    static Connection con;

    static PreparedStatement ps;

    static int Emp_Id;

    static String E_name;

    static String Gender;

    static int BasicPay;

    static int DA;

    static int HRA;

    static String query;

    static Scanner inp = new Scanner(System.in);

    public static void main(String[] args) {

        try {
```

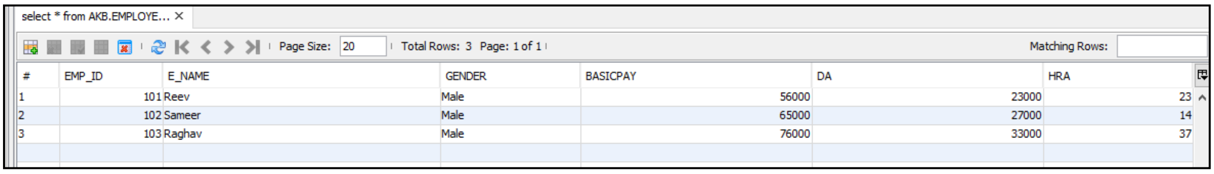
```
        System.out.println("Connecting to database...");

        con = DriverManager.getConnection(url, uid, pwd);

        System.out.println("Connected with database");

        System.out.println("Enter the number of records to insert:");

        int n = inp.nextInt();

        inp.nextLine();

        query = "INSERT INTO Employee (Emp_Id, E_Name, Gender, BasicPay, DA, HRA) VALUES (?, ?, ?, ?, ?, ?)";

        ps = con.prepareStatement(query);

        for (int i = 0; i < n; i++) {

            System.out.println("Enter details for employee " + (i + 1) + ":");

            System.out.print("Employee ID: ");

            Emp_Id = inp.nextInt();

            inp.nextLine();

            System.out.print("Name: ");

            E_name = inp.nextLine();

            System.out.print("Gender: ");

            Gender = inp.nextLine();

            System.out.print("Basic Pay: ");

            BasicPay = inp.nextInt();

            System.out.print("DA: ");

            DA = inp.nextInt();

            System.out.print("HRA: ");

            HRA = inp.nextInt();

            ps.setInt(1, Emp_Id);

            ps.setString(2, E_name);

            ps.setString(3, Gender);

            ps.setInt(4, BasicPay);
```

```
                ps.setInt(5, DA);

                ps.setInt(6, HRA);

                int result = ps.executeUpdate();

                if (result == 1) {

                    System.out.println("Record " + (i + 1) + " inserted successfully.");

                } else {

                    System.out.println("Unable to insert record " + (i + 1) + ".");

                }

            }

            con.close();

        } catch (SQLException e) {

            System.out.println("Error in SQLException: " + e.toString());

        }

    }

}
```

OUTPUT FOR 1 and 2:



```
run:
Connecting to database...
Connected with database
Enter the number of records to insert:
2
Enter details for employee 1:
Employee ID: 102
Name: Sameer
Gender: Male
Basic Pay: 65000
DA: 27000
HRA: 14
Record 1 inserted successfully.
Enter details for employee 2:
Employee ID: 103
Name: Raghav
Gender: Male
Basic Pay: 76000
DA: 33000
HRA: 37
Record 2 inserted successfully.
BUILD SUCCESSFUL (total time: 1 minute 25 seconds)
```



#	EMP_ID	E_NAME	GENDER	BASICPAY	DA	HRA
1	101	Reev	Male	56000	23000	23
2	102	Sameer	Male	65000	27000	14
3	103	Raghav	Male	76000	33000	37

3. Create a Java Web Application with servlet to do the following:

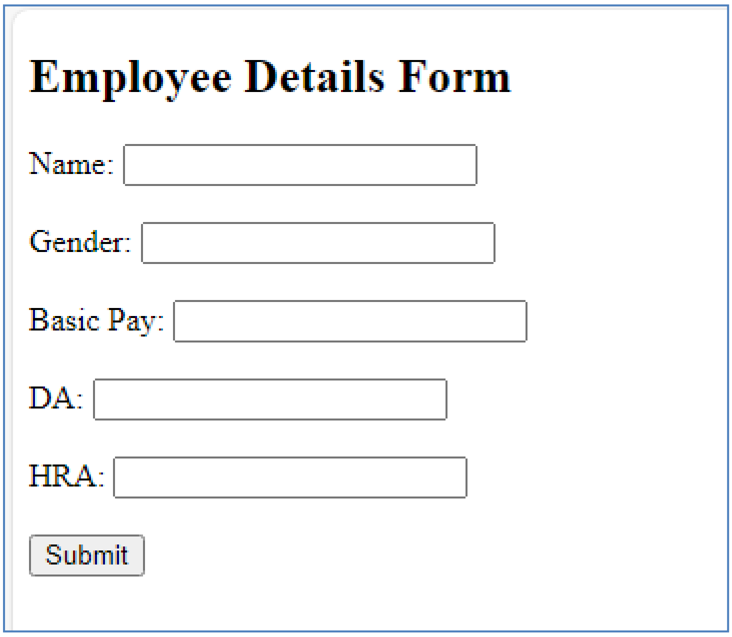
a. Create a HTML form to collect the employee details and transfer the details to the servlet upon clicking the submit button of the form.

b. Display the employee details through the servlet code.

CODE:

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/EmployeeServlet")
public class EmployeeServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String name = request.getParameter("name");
        String gender = request.getParameter("gender");
        String basicPay = request.getParameter("basicPay");
        String da = request.getParameter("da");
        String hra = request.getParameter("hra");
        out.println("<html><body>");
        out.println("<h2>Employee Details</h2>");
        out.println("<p>Name: " + name + "</p>");
        out.println("<p>Gender: " + gender + "</p>");
        out.println("<p>Basic Pay: " + basicPay + "</p>");
        out.println("<p>DA: " + da + "</p>");
        out.println("<p>HRA: " + hra + "</p>");
        out.println("</body></html>");
    }
}
```

OUTPUT:



AKASH KUMAR BANIK

24MCA0242

DATE: 04-10-2024

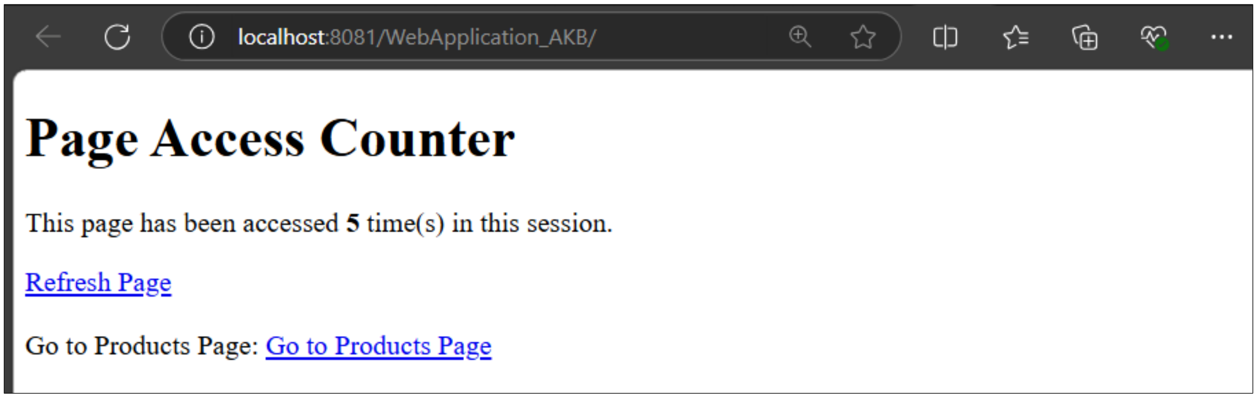
Q1: Develop a JSP application to display access count of the webpage using sessions.

CODE:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ page import="jakarta.servlet.http.HttpSession" %>
<!DOCTYPE html>
<html>
<head>
   <title>Access Count</title>
</head>
<body>
   <h1>Page Access Counter</h1>
   <%
      Integer accessCount = (Integer) session.getAttribute("accessCount");
      if (accessCount == null) {
         accessCount = 0;
      }
      accessCount++;
      session.setAttribute("accessCount", accessCount);
   %>
```

```
  <p>This page has been accessed <strong><%= accessCount %></strong> time(s) in this session.</p>
  <a href="accessCount.jsp">Refresh Page</a> <br><br>
   Go to Products Page: <a href="accessPage.html">Go to Products Page</a>
</body>
</html>
```

OUTPUT:



Page Access Counter

This page has been accessed **5** time(s) in this session.

Refresh Page

Go to Products Page: Go to Products Page

Q2. Create the following tables:

a) "Product" with necessary information like Product ID (varchar), Product name (varchar), supplier ID (varchar), stock available(number), opening stock (number), last supply date (Date), unit price (number).

b) "Supplier" containing Supplier ID (varchar), Suppler Name (varchar), Email (varchar), Mobile Phone number (varchar)

c) "User" containing UserId (varchar), Username (varchar), password (varchar), role (varchar- Allowed values are {DEO,POS,MGR})

Develop a JSP page to

d) Add new product. Prevent duplication of Product ID and entering non existing supplier ID.

e) Fetch product based on ID.

f) View the list of products for which stock available is less than 50

CODE:

addProduct.jsp File:

```
<%@ page import="java.sql.*" %>
<html>
<body>
  <form action="AddProductServlet" method="post">
    Product ID: <input type="text" name="productID"><br>
    Product Name: <input type="text" name="productName"><br>
    Supplier ID: <input type="text" name="supplierID"><br>
    Stock Available: <input type="number" name="stockAvailable"><br>
    Opening Stock: <input type="number" name="openingStock"><br>
    Last Supply Date: <input type="date" name="lastSupplyDate"><br>
    Unit Price: <input type="number" step="0.01" name="unitPrice"><br>
    <input type="submit" value="Add Product">
  </form>
</body>
</html>
```

fetchProduct.jsp File:

```
<%@ page import="java.sql.*" %>
<html>
<body>
```

```
    <form action="FetchProductServlet" method="get">
        Product ID: <input type="text" name="productID"><br>
        <input type="submit" value="Fetch Product">
    </form>
</body>
</html>
```

viewStock.jsp File:

```
<%@ page import="java.sql.*" %>
<html>
<body>
    <h2>Products with Stock Less Than 50</h2>
    <%
        try {
            Class.forName("org.apache.derby.jdbc.ClientDriver");
            Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/Demo", "akash", "akash");
            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM Product WHERE StockAvailable < 50");
            while (rs.next()) {
                out.println("Product ID: " + rs.getString("ProductID") + "<br>");
                out.println("Product Name: " + rs.getString("ProductName") + "<br>");
                out.println("Stock Available: " + rs.getInt("StockAvailable") + "<br><br>");
            }
            con.close();
        } catch (Exception e) {
            out.println("<h2>Error: " + e.getMessage() + "</h2>");
        }
    %>
</body>
</html>
```

index.html File:

```
<!DOCTYPE html>
<html>
<head>
    <title>Product Management</title>
</head>
<body>
    <h1>Welcome to Product System</h1>
    <ul>
        <li><a href="addProduct.jsp">Add New Product</a></li>
        <li><a href="fetchProduct.jsp">Fetch Product by ID</a></li>
        <li><a href="viewStock.jsp">View Products</a></li>
    </ul>
</body>
</html>
```

AddProductServlet File:

```
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
public class AddProductServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String productID = request.getParameter("productID");
            String productName = request.getParameter("productName");
            String supplierID = request.getParameter("supplierID");
            int stockAvailable = Integer.parseInt(request.getParameter("stockAvailable"));
            int openingStock = Integer.parseInt(request.getParameter("openingStock"));
            String lastSupplyDate = request.getParameter("lastSupplyDate");
            double unitPrice = Double.parseDouble(request.getParameter("unitPrice"));
            try {
                Class.forName("org.apache.derby.jdbc.ClientDriver");
                Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/Demo",
"akash", "akash");
                PreparedStatement ps = con.prepareStatement("INSERT INTO Product VALUES (?, ?,
?, ?, ?, ?, ?)");
                ps.setString(1, productID);
                ps.setString(2, productName);
                ps.setString(3, supplierID);
                ps.setInt(4, stockAvailable);
                ps.setInt(5, openingStock);
                ps.setString(6, lastSupplyDate);
                ps.setDouble(7, unitPrice);
                ps.executeUpdate();
                con.close();
                out.println("<html><body><h2>Product added successfully!</h2></body></html>");
            } catch (Exception e) {
                out.println("<html><body><h2>Error: " + e.getMessage() + "</h2></body></html>");
            }
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
```

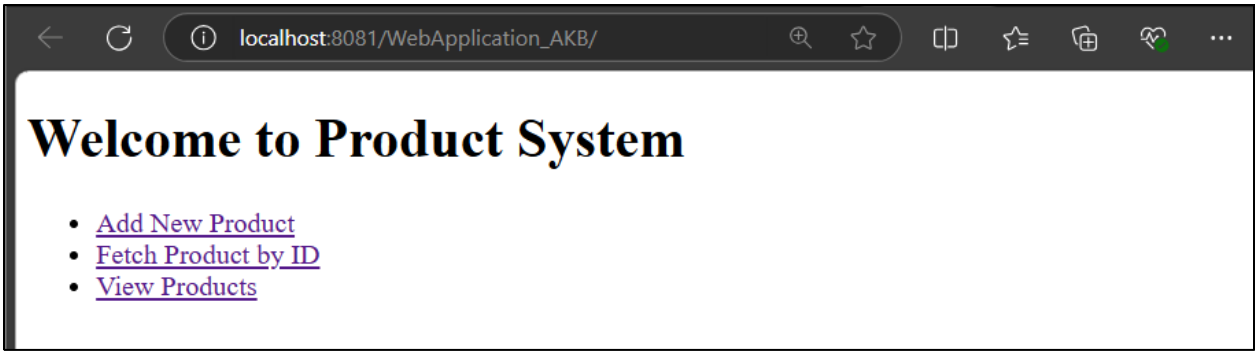
```
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {
        return "AddProductServlet";
    }
}
```

FetchProductServlet File:

```
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
```

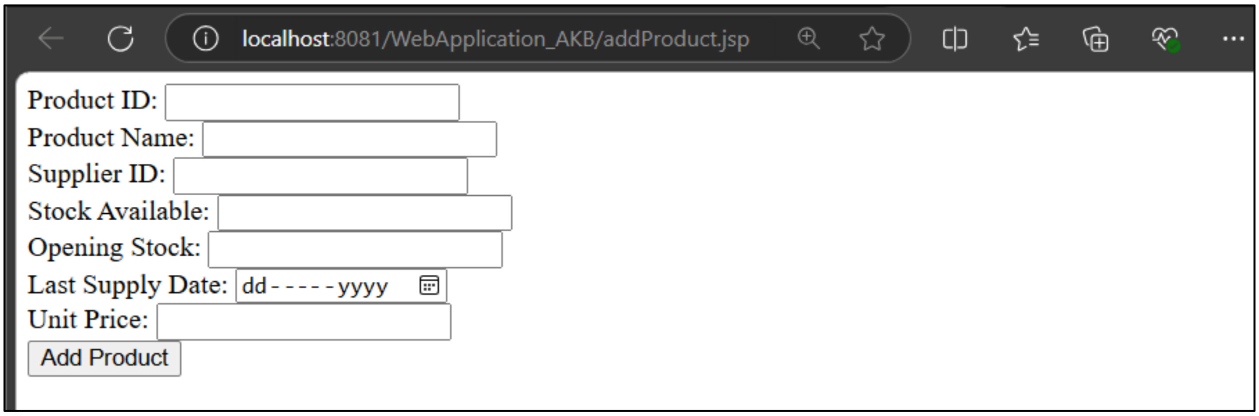
```
public class FetchProductServlet extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            String productID = request.getParameter("productID");
            try {
                Class.forName("org.apache.derby.jdbc.ClientDriver");
                Connection con = DriverManager.getConnection("jdbc:derby://localhost:1527/Demo", "akash", "akash");
                PreparedStatement ps = con.prepareStatement("SELECT * FROM Product WHERE ProductID = ?");
                ps.setString(1, productID);
                ResultSet rs = ps.executeQuery();
                if (rs.next()) {
                    out.println("<html><body>");
                    out.println("<h2>Product Details</h2>");
                    out.println("Product ID: " + rs.getString("ProductID") + "<br>");
                    out.println("Product Name: " + rs.getString("ProductName") + "<br>");
                    out.println("Supplier ID: " + rs.getString("SupplierID") + "<br>");
                    out.println("Stock Available: " + rs.getInt("StockAvailable") + "<br>");
                    out.println("Opening Stock: " + rs.getInt("OpeningStock") + "<br>");
                    out.println("Last Supply Date: " + rs.getString("LastSupplyDate") + "<br>");
                    out.println("Unit Price: " + rs.getDouble("UnitPrice") + "<br>");
                    out.println("</body></html>");
                } else {
                    out.println("<html><body><h2>No product found with ID: " + productID + "</h2></body></html>");
                }
                con.close();
            } catch (Exception e) {
                out.println("<html><body><h2>Error: " + e.getMessage() + "</h2></body></html>");
            }
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }
    @Override
    public String getServletInfo() {

        return "FetchProductServlet";

    }

}
```

OUTPUT:

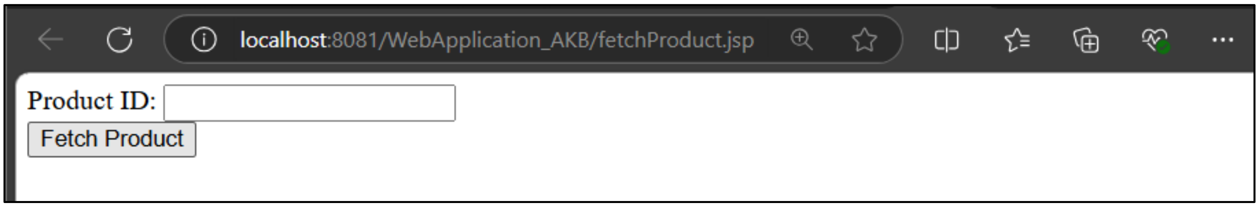

localhost:8081/WebApplication_AKB/

Welcome to Product System

- Add New Product
- Fetch Product by ID
- View Products


localhost:8081/WebApplication_AKB/addProduct.jsp

Product ID:
Product Name:
Supplier ID:
Stock Available:
Opening Stock:
Last Supply Date: dd - - - - - yyyy
Unit Price:


localhost:8081/WebApplication_AKB/fetchProduct.jsp

Product ID: