DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
V.R.SIDDHARTHA ENGINEERING COLLEGE
KANURU

HOME ASSIGNMENT

Course Name	ADVANCED JAVA PROGRAMMING
Course Code	17CS2504A
Name of the Student	Shaik Hussain Saitaj
Roll No.	198W1A05I0
Section.	C
Group No.	8
Programme	B.TECH III YEAR V SEMESTER
Type of the Course	PROGRAMME CORE
Course Instructor	A.RAGHUVIRA PRATAP
Academic Year	2021-2022

Student Signature with Date (Submission date) :	Assessment Marks:	Assessed by:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
V.R.SIDDHARTHA ENGINEERING COLLEGE
KANURU

HOME ASSIGNMENT

Course Name	ADVANCED JAVA PROGRAMMING
Course Code	17CS2504A
Name of the Student	Kavya Sharmila Siram
Roll No.	198W1A05F8
Section.	C
Group No.	5
Programme	B.TECH III YEAR V SEMESTER
Type of the Course	PROGRAMME CORE
Course Instructor	B. RAGHUVIRA PRATAP
Academic Year	2021-2022

Student Signature with Date (Submission date) :	Assessment Marks:	Assessed by:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
V.R.SIDDHARTHA ENGINEERING COLLEGE
KANURU

HOME ASSIGNMENT

Course Name	ADVANCED JAVA PROGRAMMING
Course Code	17CS2504A
Name of the Student	Joseph Kishore Madda
Roll No.	198W1A05G2
Section.	C
Group No.	5
Programme	B.TECH III YEAR V SEMESTER
Type of the Course	PROGRAMME CORE
Course Instructor	C. RAGHUVIRA PRATAP
Academic Year	2021-2022

Student Signature with Date (Submission date) :	Assessment Marks:	Assessed by:

Case Study – 1

Write an RMI client server String operations application. RMI server provides two remotely accessible methods: long findStringLength(String s); //returns length of a String parameter boolean checkPalindrome(String s); //determines whether a String //parameter is palindrome or not.

Program:

```
// an RMI client server String operations application.
/*
    Author          :   Afrose
    Program Name:       stringOperations.java
    Program No      :   01
    Description     :   RMI Programming
*/
// Import required packages
import java.rmi.*;

public interface stringOperations extends Remote
{
    public long findStringLength(String s) throws RemoteException;  //returns length of a 
String parameter
    public boolean checkPalindrome(String s) throws RemoteException; //determines 
whether a String parameter is palindrome or not
}
```

stringOperationsRemote.java

```
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;

public class stringOperationsRemote extends UnicastRemoteObject implements 
stringOperations{
    stringOperationsRemote() throws RemoteException{
        super();
    }

    public long findStringLength(String s) {
        return s.length();
    }

    public boolean checkPalindrome(String str){
        int i = 0, j = str.length() - 1;
   while (i < j) {

    if (str.charAt(i) != str.charAt(j))              // If there is a mismatch
       return false;
    // Increment first pointer and decrement the other
    i++;
    j--;
   }
   return true;        // Given string is a palindrome
    }
}
```

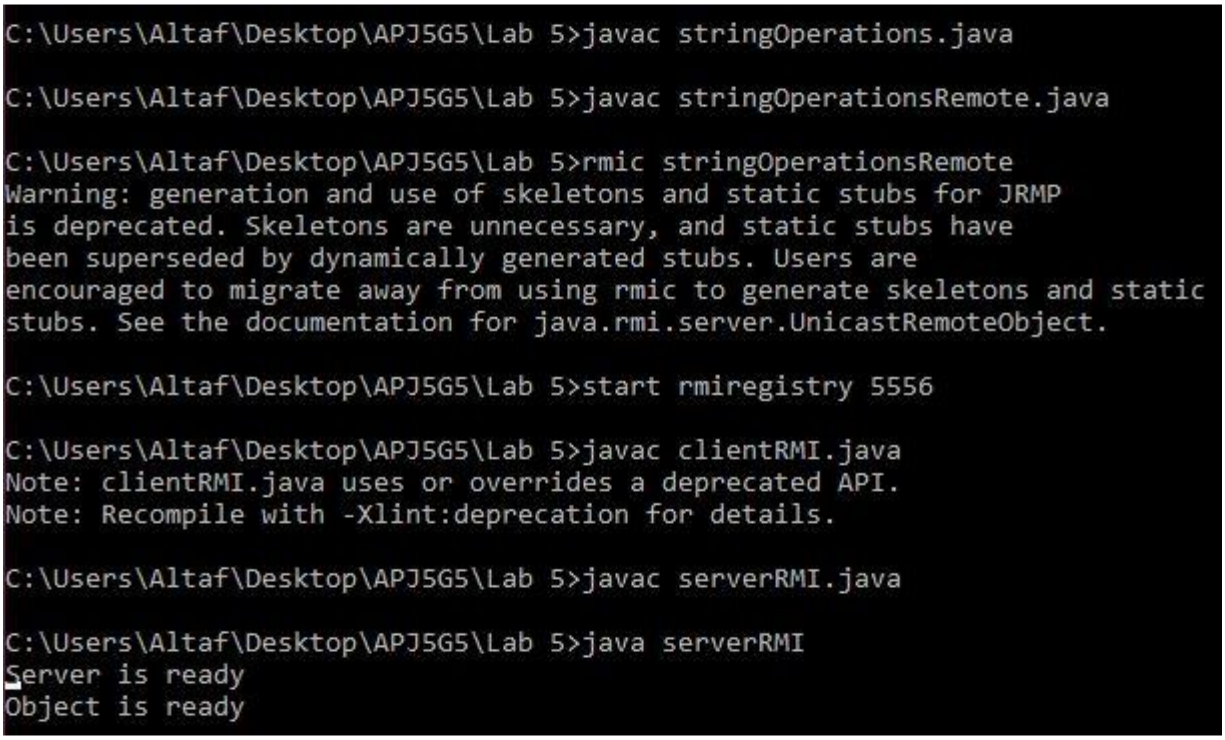
serverRMI.java

```
import java.rmi.*;
import java.rmi.registry.*;
public class serverRMI {
        public static void main(String a[]) {
                try {
                        stringOperationsRemote stub=new stringOperationsRemote();
                        Naming.rebind("rmi://localhost:5556/afrose",stub);
                        System.out.println("Server is ready");
                        System.out.println("Object is ready");
                } catch(Exception e){
                        System.out.println(e);
                }
        }
}
```

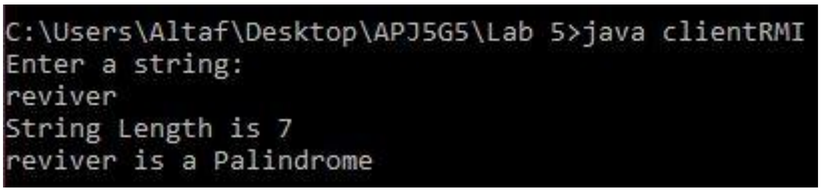
clientRMI.java

```
import java.rmi.*;
import java.io.*;

public class clientRMI {
        public static void main(String a[]) {
                try {
                        stringOperations
stub=(stringOperations)Naming.lookup("rmi://localhost:5556/afrose");
                        DataInputStream in =new DataInputStream(System.in);
                        System.out.println("Enter a string: ");
                        String s=in.readLine();
                        System.out.println("String Length is "+stub.findStringLength(s));
                        if(stub.checkPalindrome(s))
                                System.out.println(s+" is a Palindrome");
                        else
                                System.out.println(s+" is not a Palindrome");
                } catch(Exception e) {
                        System.out.println(e);
                }
        }
}
```

Output:



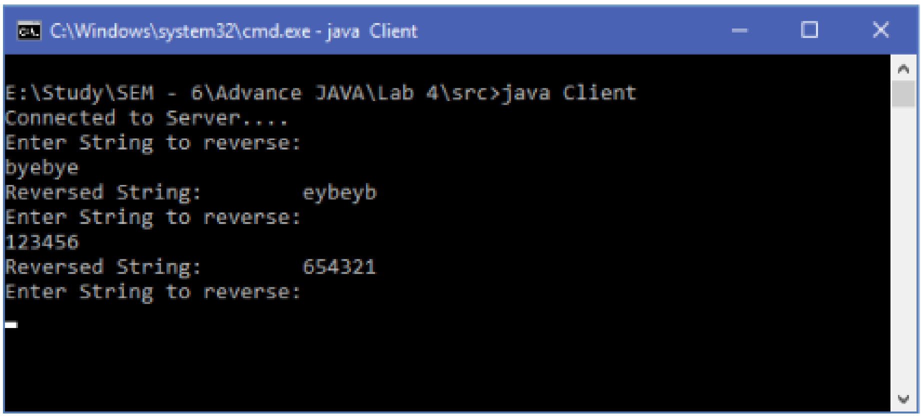
```
C:\Users\Altaf\Desktop\APJ5G5\Lab 5>javac stringOperations.java

C:\Users\Altaf\Desktop\APJ5G5\Lab 5>javac stringOperationsRemote.java

C:\Users\Altaf\Desktop\APJ5G5\Lab 5>rmic stringOperationsRemote
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

C:\Users\Altaf\Desktop\APJ5G5\Lab 5>start rmiregistry 5556

C:\Users\Altaf\Desktop\APJ5G5\Lab 5>javac clientRMI.java
Note: clientRMI.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

C:\Users\Altaf\Desktop\APJ5G5\Lab 5>javac serverRMI.java

C:\Users\Altaf\Desktop\APJ5G5\Lab 5>java serverRMI
Server is ready
Object is ready
```



```
C:\Users\Altaf\Desktop\APJ5G5\Lab 5>java clientRMI
Enter a string:
reviver
String Length is 7
reviver is a Palindrome
```

Case Study – 2

Implement Concurrent TCP Server programming in which more than one client can connect and communicate with Server for sending the string and server returns the reverse of string to each of client



```
E:\Study\SEM - 6\Advance JAVA\Lab 4\src>java Client
Connected to Server....
Enter String to reverse:
byebye
Reversed String:        eybeyb
Enter String to reverse:
123456
Reversed String:        654321
Enter String to reverse:
```

Program:

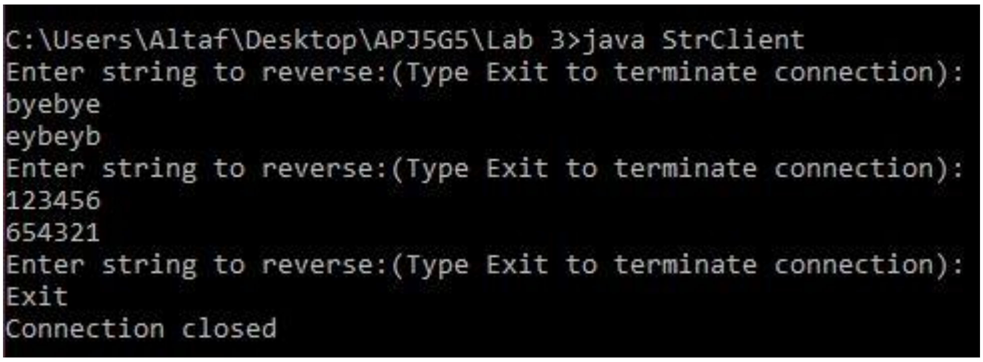
// A Server side network program that runs more than 1 client to reverse a string.
/*

```
    Author          :       Afrose
    Program Name:           StrServer.java
    Lab Cycle       :       02
    Description     :       Network Programming

*/
// Import required packages
import java.io.*;
import java.text.*;
import java.util.*;
import java.net.*;

// Server class
public class StrServer
{
    public static void main(String[] args) throws IOException
    {
        ServerSocket ss = new ServerSocket(5056);

        while (true) {
            Socket s = null;
            try {
                s = ss.accept();
                DataInputStream dis = new
DataInputStream(s.getInputStream());
                DataOutputStream dos = new
DataOutputStream(s.getOutputStream());
                Thread t = new ClientHandler(s, dis, dos);
                t.start();
            }
            catch (Exception e){
                s.close();
                e.printStackTrace();
            }
        }
    }
}

// ClientHandler class
class ClientHandler extends Thread {

    final DataInputStream dis;
    final DataOutputStream dos;
    final Socket s;

    public ClientHandler(Socket s, DataInputStream dis, DataOutputStream dos) {
        this.s = s;
        this.dis = dis;
        this.dos = dos;
    }

    public void run() {
        String received;
        String toreturn="";
        char ch;
        while (true) {
            try {
                dos.writeUTF("Enter string to reverse:(Type Exit to terminate connection): ");

                received = dis.readUTF();

                if(received.equals("Exit"))
                {
                    System.out.println("Closing this connection.");
                    this.s.close();
                    System.out.println("Connection closed");
                    break;
                }
                toreturn="";
                for (int i=0; i<received.length(); i++) {
           ch= received.charAt(i);
           toreturn= ch+toreturn;
          }
                dos.writeUTF(toreturn);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        try {
            // closing resources
            this.dis.close();
            this.dos.close();

        }catch(IOException e){
            e.printStackTrace();
        }
    }
}
```

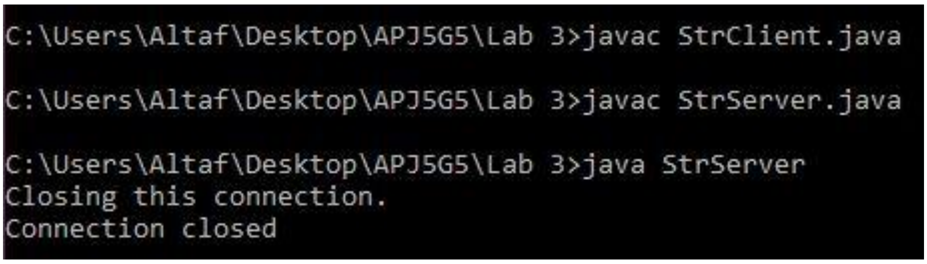
strClient.java

```
// Import required packages
import java.io.*;
import java.net.*;
import java.util.*;

// Client class
public class StrClient {
    public static void main(String[] args) throws IOException {
        try {
            Scanner scn = new Scanner(System.in);
            Socket s=new Socket("localhost",5056);
            DataInputStream dis = new DataInputStream(s.getInputStream());
            DataOutputStream dos = new
DataOutputStream(s.getOutputStream());

            while (true) {
                System.out.println(dis.readUTF());
                String tosend = scn.nextLine();
                dos.writeUTF(tosend);

                if(tosend.equals("Exit")){
                    s.close();
                    System.out.println("Connection closed");
                    break;
                }

                String received = dis.readUTF();
                System.out.println(received);
            }

            scn.close();
            dis.close();
            dos.close();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
}
```

Output:



```
C:\Users\Altaf\Desktop\APJ5G5\Lab 3>java StrClient
Enter string to reverse:(Type Exit to terminate connection):
byebye
eybeyb
Enter string to reverse:(Type Exit to terminate connection):
123456
654321
Enter string to reverse:(Type Exit to terminate connection):
Exit
Connection closed
```



```
C:\Users\Altaf\Desktop\APJ5G5\Lab 3>javac StrClient.java

C:\Users\Altaf\Desktop\APJ5G5\Lab 3>javac StrServer.java

C:\Users\Altaf\Desktop\APJ5G5\Lab 3>java StrServer
Closing this connection.
Connection closed
```

Case Study – 3

Consider Bank table with attributes AccountNo, CustomerName, Balance, Phone and Address. Write a JDBC database application which allows insertion, updation and deletion of records in Bank table. Print values of all customers whose balance is greater than 20,000.

Program:

```
/***************
             Author         :       Afrose
            Program Name :        Bank.java
            Lab Cycle       :       03
            Description     :       JDBC Connectivity

***************/
import java.sql.*;
import java.util.*;
public class Bank
{

            public static void main(String args[])
            {
                        Connection con=null;
                        Statement st=null;
                        ResultSet rs=null;
                        try
                        {
                                    Scanner sc=new Scanner(System.in);



                                    Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
                    String url="jdbc:oracle:thin:@localhost:1521:XE";
                    String username="system";
                    String password="admin";
                    con=DriverManager.getConnection(url,username,password);

                    System.out.println("1.Insert new account \n 2.Update records(withdraw or
deposit) \n 3.Delete a record\n 4.Retrieve accounts whose balance is 12000\n");
                    int num=sc.nextInt();
                    switch(num)
                    {
                            case 1:  System.out.println("Enter Account No of new record:\n");
                                    int accno=sc.nextInt();
                                    System.out.println("Enter balance of new
record:\n");
                                    int balance=sc.nextInt();
                                    System.out.println("Enter Customer name of new
record:\n");
                                    String name=sc.nextLine();

                                    String query1="INSERT INTO banker
VALUES(?,?,?)";
                                    PreparedStatement
pt1=con.prepareStatement(query1);
                                    pt1.setInt(1,accno);
                                    pt1.setString(3,name);
                                    pt1.setInt(2,balance);
                                    int rows1=pt1.executeUpdate();
                                    if(rows1>0)
                                            System.out.println("New record inserted
succesfull!!!\n");
                                    else
                                            System.out.println("New record not
inserted!!!!\n");
```

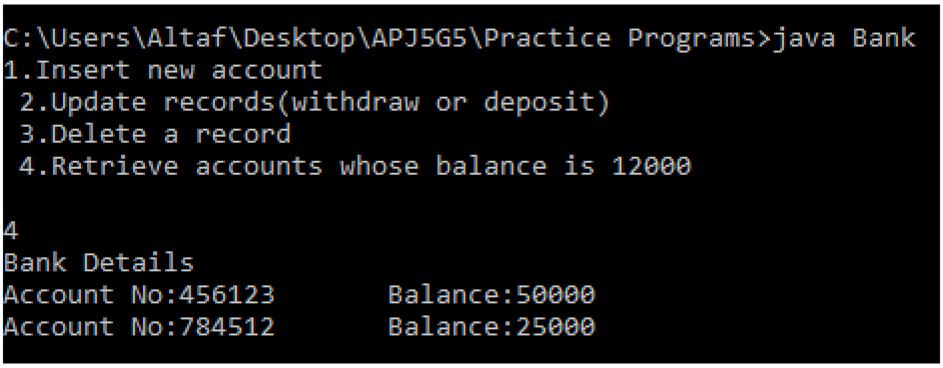
```
                                        break;

                              case 2: System.out.println("Enter Account No of the record:");
                                        int accn=sc.nextInt();
                                        System.out.println("1.withdraw \n 2.deposit\n");
                                        String ch=sc.nextLine();
                                        if(ch.equals("1"))
                                        {
                                             System.out.println("Enter how much you
want to withdraw:\n");
                                             int w_amt=sc.nextInt();
                                             String q1="select amount from banker";
                                             st=con.createStatement();
                                             rs=st.executeQuery(q1);
                                             int bal=rs.getInt(1);
                                             bal=bal-w_amt;

                                             String query2="UPDATE banker SET
amount=? WHERE acc_no=?";
                                             PreparedStatement
pt2=con.prepareStatement(query2);
                                             pt2.setInt(1,bal);
                                             pt2.setInt(2,accn);
                                             int rows2=pt2.executeUpdate();
                                             if(rows2>0)
                                                  System.out.println("Record updated
succesfully!!!! \n");
                                             else
                                                  System.out.println("Record not
updated!!!!\n");



                                        }
```

```
        else{
            System.out.println("Enter how much you want to deposit:\n");
            int w_amt=sc.nextInt();
            String q1="select balance from Bank";
            st=con.createStatement();
            rs=st.executeQuery(q1);
            int bal=rs.getInt(1);

            bal=bal+w_amt;

            String query2="UPDATE Bank SET balance=? WHERE acc_no=?";
            PreparedStatement pt3=con.prepareStatement(query2);
            pt3.setInt(1,bal);
            pt3.setInt(2,accn);
            int rows3=pt3.executeUpdate();
            if(rows3>0)
                System.out.println("Record updated succesfully~!!! \n");
            else
                System.out.println("Record not updated!!!\n");

        }
        break;
case 3: System.out.println("Enter Account No of the record:");
        int acc=sc.nextInt();
        String query3="DELETE FROM banker WHERE acc_no=?";
        PreparedStatement pt4=con.prepareStatement(query3);
        pt4.setInt(1,acc);
        int rows=pt4.executeUpdate();
```

```
                                        if(rows>0)
                                                System.out.println("Record deleted
succesfully!!!\n");
                                        else
                                                System.out.println("Record not
deleted!!!\n");
                                        break;
                                case 4: String query="select * from banker WHERE
amount>20000";
                                        st=con.createStatement();
                                        rs=st.executeQuery(query);
                                        System.out.println("Bank Details");
                                        while(rs.next())
                                        {
                                                System.out.println("Account
No:"+rs.getString(1)+"\tBalance:"+rs.getString(2));
                                        }
                                        break;

                        }
                }
                catch(Exception ex)
                {
                        System.out.println("Connection is unsuccessful");
                }
                finally
                {
                        try{
                                st.close();
                                rs.close();
                                con.close();
                        }
                        catch(Exception ee)
                        {
```

```
                                        System.out.println(ee);
                                }
                        }
                }
        }
```

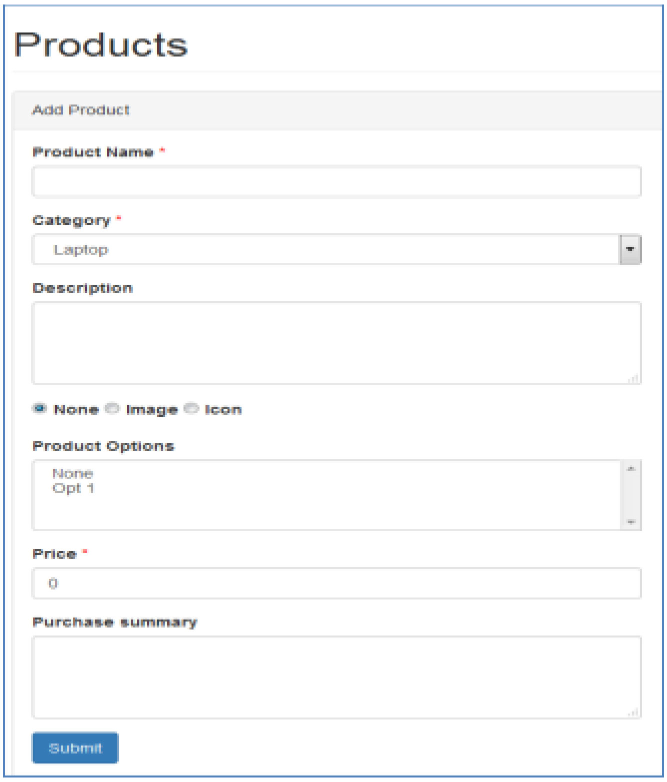
Output:



```
C:\Users\Altaf\Desktop\APJ5G5\Practice Programs>java Bank
1.Insert new account
 2.Update records(withdraw or deposit)
 3.Delete a record
 4.Retrieve accounts whose balance is 12000

4
Bank Details
Account No:456123       Balance:50000
Account No:784512       Balance:25000
```

Case Study – 4

Write a servlet which accepts product details from html form and stores the product details into database.



Products

Add Product

Product Name *

Category *

Laptop

Description

● None ○ Image ○ Icon

Product Options

None
Opt 1

Price *

0

Purchase summary

Submit

Program:

```
/***************
    Author       :   Afrose
    Program Name :   ProductServlet.java
    Program No   :   04
    Description  :   Java Servlets
***************/
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;
import java.sql.*;

//define the servlet class by extending httpservlet abstract
public class ProductsServlet extends HttpServlet{
    public void doPost(HttpServletRequest req,HttpServletResponse res)throws
IOException,ServletException{
        res.setContentType("text/html;charset=UTF-8");
        PrintWriter out = res.getWriter();
    String proid = req.getParameter("product");
    String catid= req.getParameter("category");
    String price = req.getParameter("price");


    try{
      Class.forName("oracle.jdbc.driver.OracleDriver");
      Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","admin");
      PreparedStatement ps = con.prepareStatement("insert into product values(?,?,?)");
      ps.setString(1,proid);
      ps.setString(2,catid);
      ps.setString(3,price);


      int i=ps.executeUpdate();
      if(i>0){
        out.print("<font color=\"green\" size=\"20\">Product Added Successfully.</font>");
      }
    }
    catch(Exception ee){
      out.println(ee.getMessage());
      ee.printStackTrace();
    }
  }}
```

Products.html:

```
<!DOCTYPE html>
<html>
<head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Products details</title>
        <style>
                body{
```

```
                font-family: Times;
            }
            form{
                width: 400px;
                margin: auto;
                border: 1px solid;
                padding: 10px;
            }
            input,textarea,select{
                margin-bottom: 15px;
            }
        </style>
    </head>
    <body>
            <form method="POST" action="http://localhost:8090/5G5/products">
                <h2>Products</h2>
                <label style="background-color: lightgray;">Add Product</label><br>

                <label>Product Name*</label><br>
                <input type="text" name="product" required><br>

                <label>Category*</label><br>
                <select id="category" name="category" required>
                    <option value="laptop">Laptop</option>
                    <option value="desktop">Desktop</option>
                    <option value="phone">Phone</option>
                    <option value="tablet">Tablet</option>
                    <option value="watch">Watch</option>
                </select><br>

                <label>Description</label><br>
                <textarea rows = "5" cols = "50" name = "description">
```

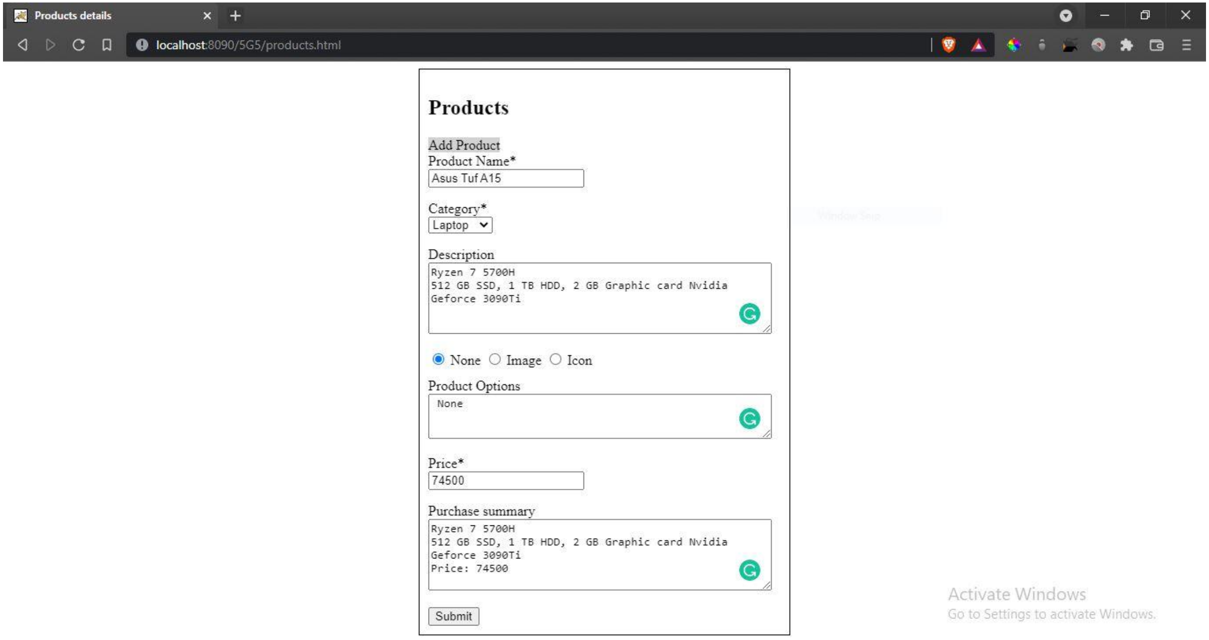
```
          </textarea><br>

          <input type="radio" name="option" value="None">
                    <label>None</label>
                    <input type="radio" name="option" value="image">
                    <label>Image</label>
                    <input type="radio" name="option" value="icon">
                    <label>Icon</label><br>

                    <label>Product Options</label><br>
                    <textarea rows = "3" cols = "50" name = "options">
          </textarea><br>

          <label>Price*</label><br>
          <input type="text" name="price" required><br>

          <label>Purchase summary</label><br>
                    <textarea rows = "5" cols = "50" name = "summary">
          </textarea><br>

                    <button type="submit">Submit</button>
               </form>
</body>
</html>
```

Add these to **web.xml** file

```
<servlet>
          <servlet-name>FourthServlet</servlet-name>
          <servlet-class>ProductsServlet</servlet-class>
</servlet>
<servlet-mapping>
          <servlet-name>FourthServlet</servlet-name>
          <url-pattern>/products</url-pattern>
```

```
</servlet-mapping>
```

Output:


Products details
localhost:8090/5G5/products.html
Activate Windows
Go to Settings to activate Windows.

Products

Add Product
Product Name*
Asus Tuf A15

Category*
Laptop

Description
```
Ryzen 7 5700H
512 GB SSD, 1 TB HDD, 2 GB Graphic card Nvidia
Geforce 3090Ti
```

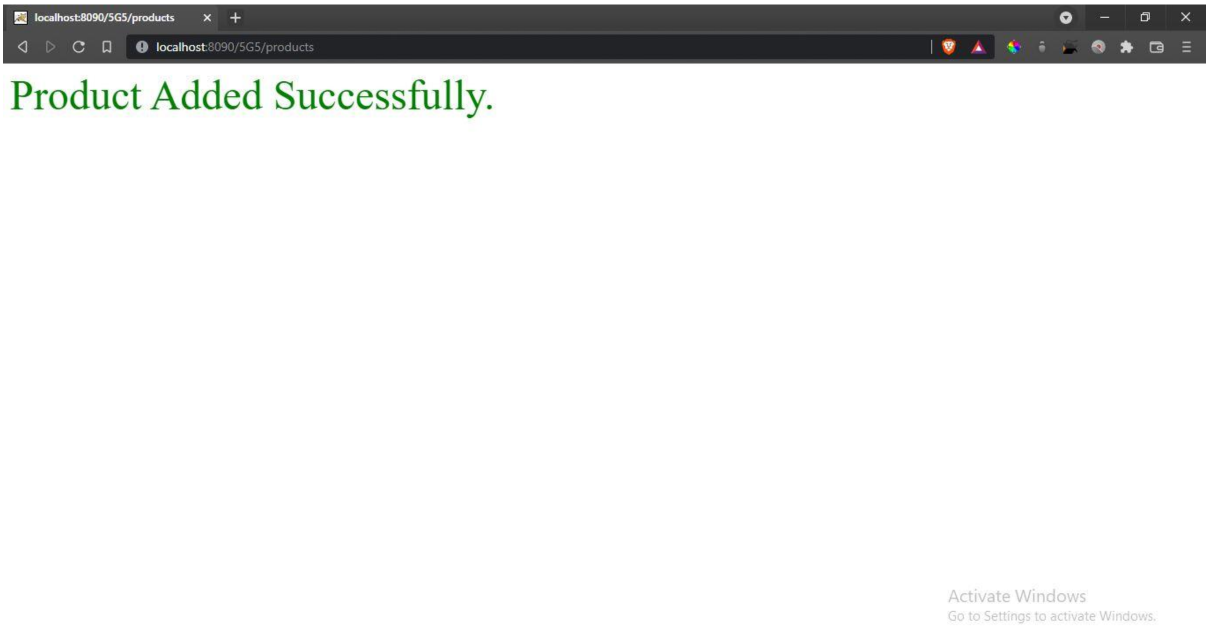
● None ○ Image ○ Icon

Product Options
```
 None
```

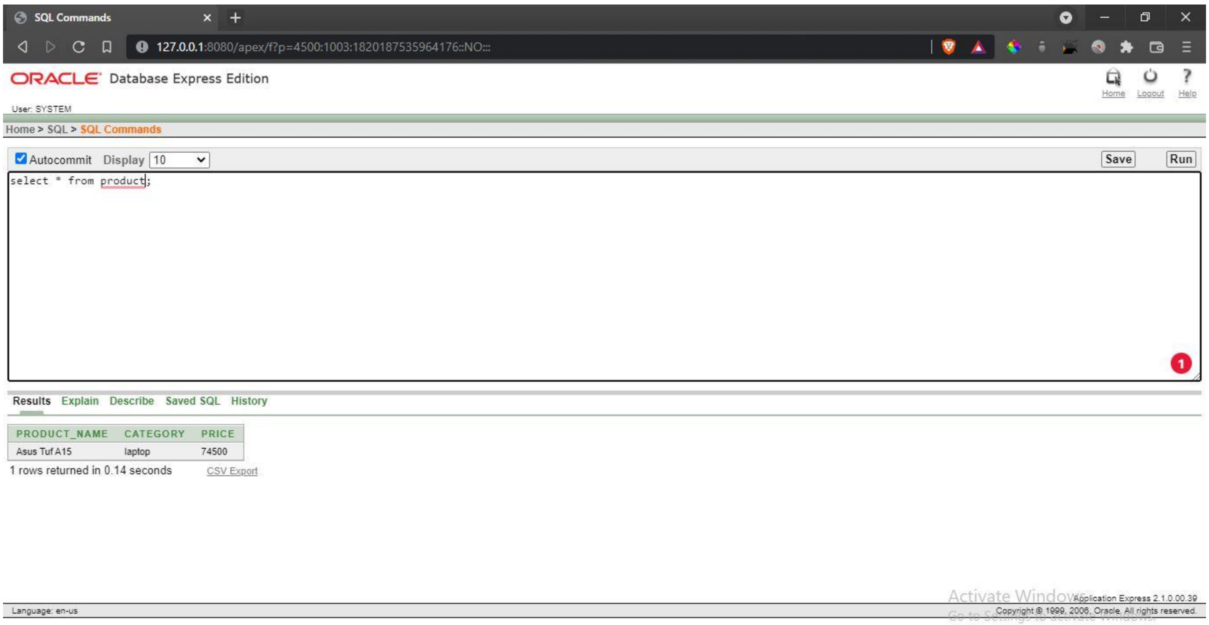
Price*
74500

Purchase summary
```
Ryzen 7 5700H
512 GB SSD, 1 TB HDD, 2 GB Graphic card Nvidia
Geforce 3090Ti
Price: 74500
```

Submit


localhost:8090/5G5/products
localhost:8090/5G5/products
Activate Windows
Go to Settings to activate Windows.

Product Added Successfully.



SQL Commands
127.0.0.1:8080/apex/f?p=4500:1003:1820187535964176::NO:::
ORACLE Database Express Edition
User: SYSTEM
Home > SQL > SQL Commands
☑ Autocommit Display 10
Save
Run
Results Explain Describe Saved SQL History
Language: en-us
Application Express 2.1.0.00.39
Copyright © 1999, 2006, Oracle. All rights reserved.

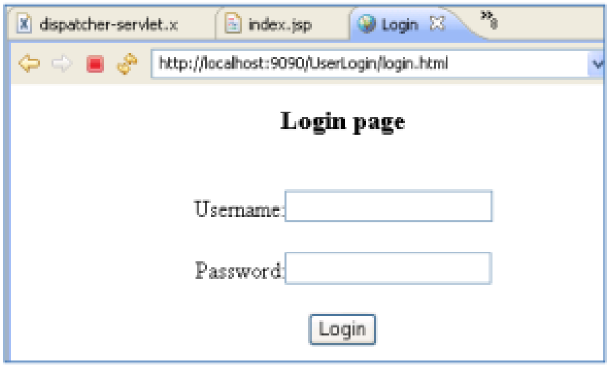
```
select * from product;
```

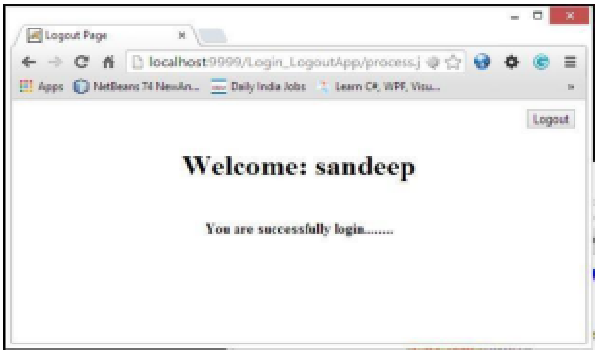
PRODUCT_NAME	CATEGORY	PRICE
Asus Tuf A15	laptop	74500

1 rows returned in 0.14 seconds   CSV Export

Case Study – 5

Write a program to create login form using HTML. When form is submitted to JSP, fetch submitted details and check whether user is valid or not?





Program:

```
   <!--  Author       :   Afrose

    Program Name :   loginJSP.html

    Program No   :   05

    Description  :   Java Server Pages -->

<head>

    <title>Login</title>

    <meta charset="UTF-8">

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

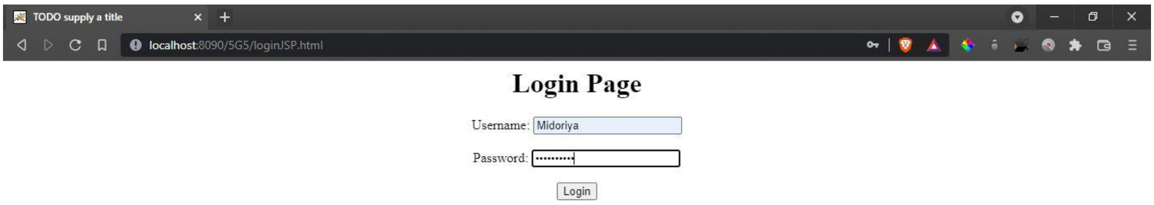
```
</head>
<body>
    <center>
        <h1>Login Page</h1>
        <form method="post" action="login.jsp">
            Username: <input type="text" name="uname"><br><br>
            Password: <input type="password" name="pass"><br><br>
            <input type="submit" name="submit" value="Login">
        </form>
    </center>
</body>
</html>
```

Login.jsp:

```
<%@ page import="java.io.*" %>
<%@ page import="javax.servlet.*" %>
<%@ page import="javax.servlet.http.*" %>
<%@ page import="java.sql.*" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Login</title>
</head>

<%
String username=request.getParameter("uname");
String pass=request.getParameter("pass");
try{
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","admin");
```

```
    Statement ps = con.createStatement();
    String query = "SELECT * from Registrations where uname='"+username+"' and password='"+pass+"'";
    ResultSet rs = ps.executeQuery(query);


    if(rs.next()){
        out.println("<body><center>");
        out.println("<h1>Welcome: "+rs.getString(2)+"</h1>");
        out.println("<h3>You are successfully logged in</h3>");
        out.println("</center></body>");
    } else{
        out.println("<br> <center><font color=\"red\">Invalid Credentials</font></center>");
    }
} catch(Exception e){
    out.println(e.getMessage());
    e.printStackTrace();
}
%>
</html>
```

Output:

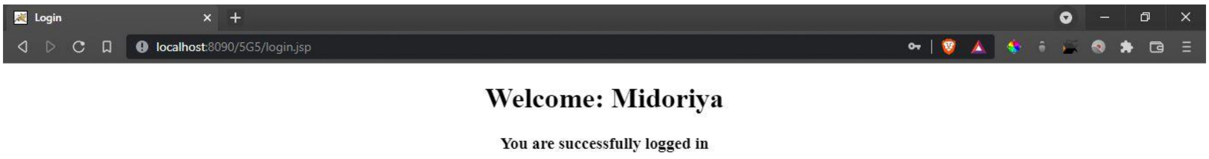


Login Page

Username: Midoriya

Password: ••••••••••

Login



Welcome: Midoriya

You are successfully logged in