ADVANCE JAVA

PRACTICAL-1

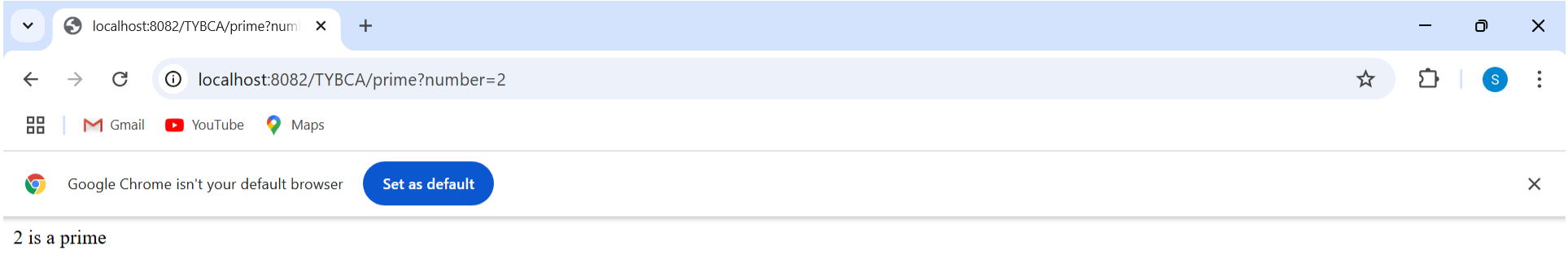
Servlet-1(1):-Write a servlet to determine whether the number is Prime or not.

Code:-Prime.java

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/prime")
public class prime extends HttpServlet {
	private static final long serialVersionUID = 1L;
	 public prime() {
	super();
	// TODO Auto-generated constructor stub
   }
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		PrintWriter pw;
		response.setContentType("text/html");
		pw=response.getWriter();
		String n = request.getParameter("number");
		int num = Integer.parseInt(n);
		int flag=0;
for(int i=2; i<=(num-1);i++)
{
	if(num%i==0)
	{
		flag=1;
		break;
	}
}
if(flag==0)
	pw.println(num+" is a prime");
else
	pw.println(num+" is not a prime");
```

```
            pw.close();
        }


}
```

OUTPUT



Practical-1

Servlet-1(2)Write a servlet to determine whether the entered name from the html form is Palindrome or not.

Code:-Palindrome.html

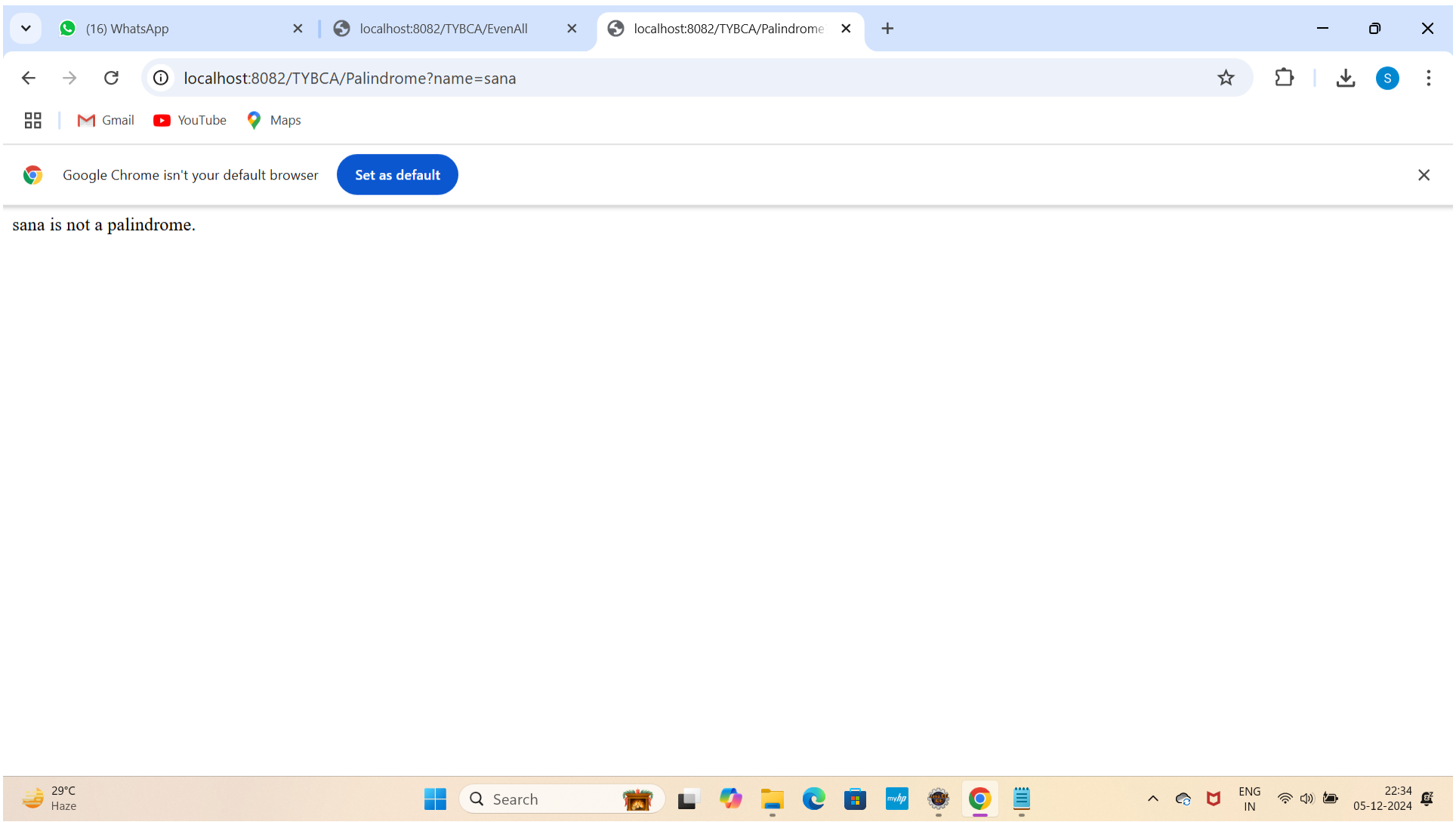
```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form name="frm" method="post" action="Palindrome.java">
Enter first number:<input type=”text" name=”txtpali”/>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Code:-Palindrome.java

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/Palindrome")
public class Palindrome extends HttpServlet {
	private static final long serialVersionUID = 1L;

    public Palindrome() {
        super();
        // TODO Auto-generated constructor stub
    }
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub
		PrintWriter pw;
		response.setContentType("text/html");
		pw=response.getWriter();
		String name=request.getParameter("name");
```

```
        StringBuffer sb=new StringBuffer(name);
        String rev=sb.reverse().toString();
        if(name.equalsIgnoreCase(rev))
                pw.println(name+ " is a palindrome.");
        else
                pw.println(name+ " is not a palindrome.");
        pw.close();
        }

}
```

OUTPUT



PRACTICAL-1

Servlet-1(3)Write a servlet program to print all the Even numbers between the two entered numbers by the user, let say user enters 5 and 500, so print the even numbers between 5 and 500.

Code:-EvenAll.html

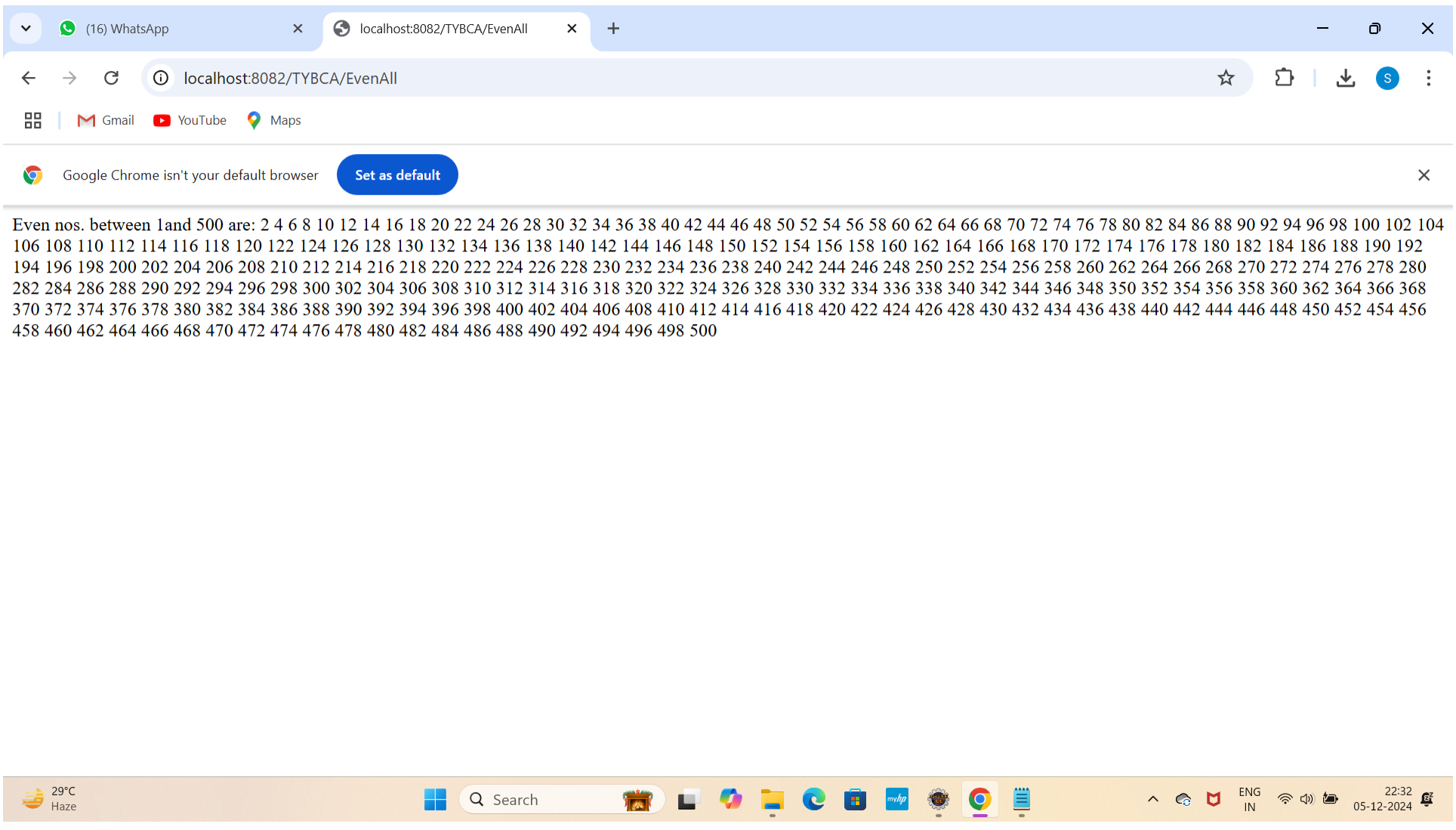
```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form name="frm" method="post" action="EvenAll">
Enter first number:<input type="text" name="t1" value=""/>
Enter second number:<input type="text" name="t2" value=""/>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

Code:-EvenAll.java

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/EvenAll")
public class EvenAll extends HttpServlet {
	private static final long serialVersionUID = 1L;
	public EvenAll() {
	super();
	// TODO Auto-generated constructor stub
}
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		PrintWriter pw;
	response.setContentType("text/html");
	pw=response.getWriter();
	int n1= Integer.parseInt(request.getParameter("t1"));
```

```
        int n2= Integer.parseInt(request.getParameter("t2"));
        pw.println("Even nos. between "+n1 +"and "+n2+" are: ");
            for(int i=n1;i<=n2;i++)
            {
                    if(i%2==0)
                            pw.print(" "+i);
            }

    }

}
```

OUTPUT



PRACTICAL-2

Servlet-2(1):-Write a servlet where user enters a name from a form and you send the length back of the name to him.

Code:-length.html

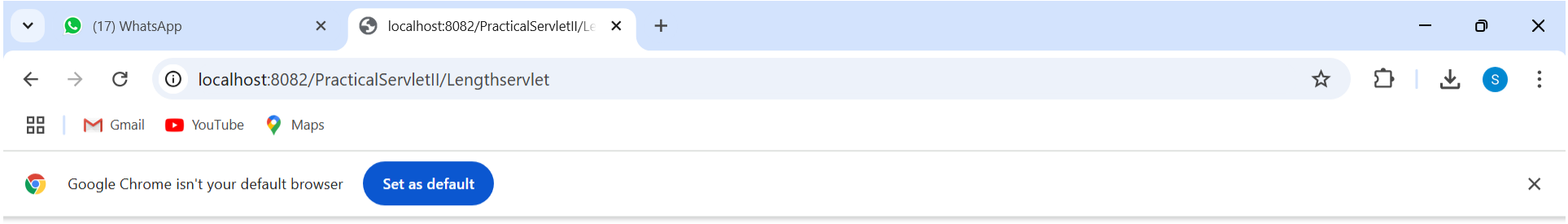
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<form action="Lengthservlet" method="post">
Name: <input type="text" name="name"/>
<input type="submit" value="calculate lenght">
</form>
</body>
</html>
```

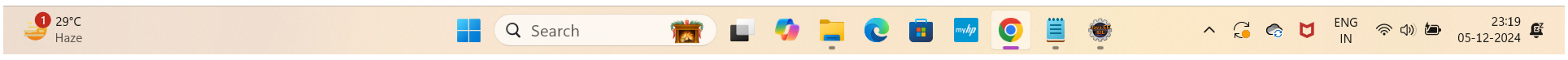
Code:-Lengthservlet.java

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
@WebServlet("/Lengthservlet")
public class Lengthservlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
 public Lengthservlet() {
    super();
    // TODO Auto-generated constructor stub
 }
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
                                                                    IOException {
            // TODO Auto-generated method stub
            PrintWriter pw=response.getWriter();
            response.setContentType("text/html");
            String name=request.getParameter("name");
            int length=name.length();
        pw.print("Length of " +name+ " is "+length);
```

```
        }

}
```

OUTPUT


(17) WhatsApp
localhost:8082/PracticalServletII/Le
localhost:8082/PracticalServletII/Lengthservlet
Gmail
YouTube
Maps
Google Chrome isn't your default browser
Set as default
Length of sanakausar is 10


29°C
Haze
Search
ENG
IN
23:19
05-12-2024

PRACTICAL-2

Servlet-2(2):-Write a HttpServlet to accept the values for following table and insert into it.

id	name	actor	actress	director	releaseDate	ratepoint

Code:-Movie.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<form method="post" action="Insertmovie">
    <div>
        <label>Enter Movie Id: </label>
        <input type="text" name="Id">
        </div>
        <br><br>
        <div>
        <label>Enter Movie Name: </label>
        <input type="text" name="name">
        </div>
        <br><br>
        <div>
        <label>Enter Actor Name: </label>
        <input type="text" name="actor">
        </div>
        <br><br>
        <div>
        <label>Enter Actress Name: </label>
        <input type="text" name="actress">
        </div>
        <br><br>
        <div>
        <label>Enter Director Name: </label>
        <input type="text" name="director">
        </div>
        <br><br>
        <div>
```

```
        <label>Enter Release Date: </label>
        <input type="text" name="rDate">
        </div>
        <br><br>
        <div>
        <label>Enter Rate Point: </label>
        <input type="text" name="rPoint">
        </div>
        <br><br>
        <div>
        <input type="submit" value="Add movie">
        </div>
</form>

</body>
</html>
```

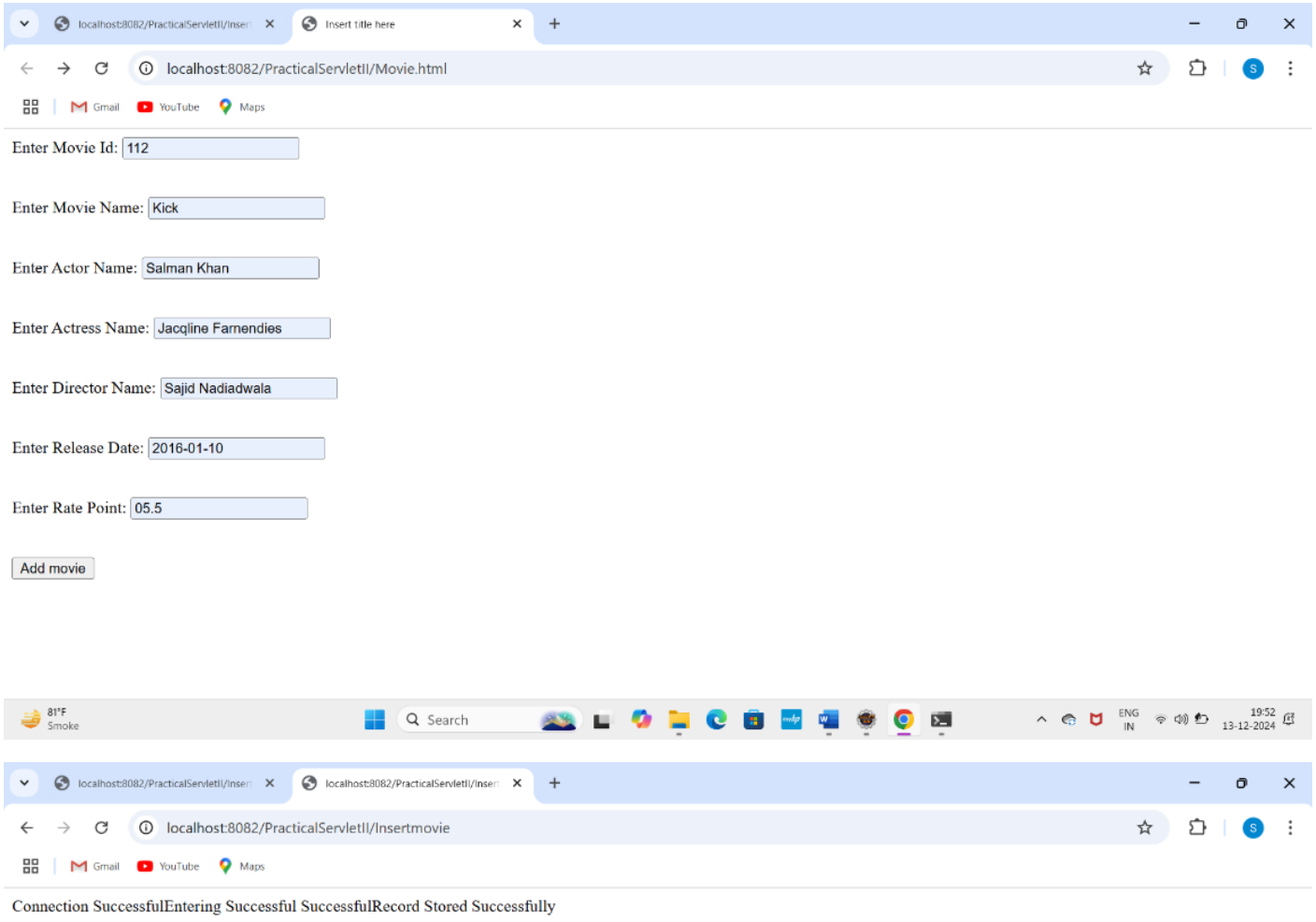
Code:-Insertmovie.java

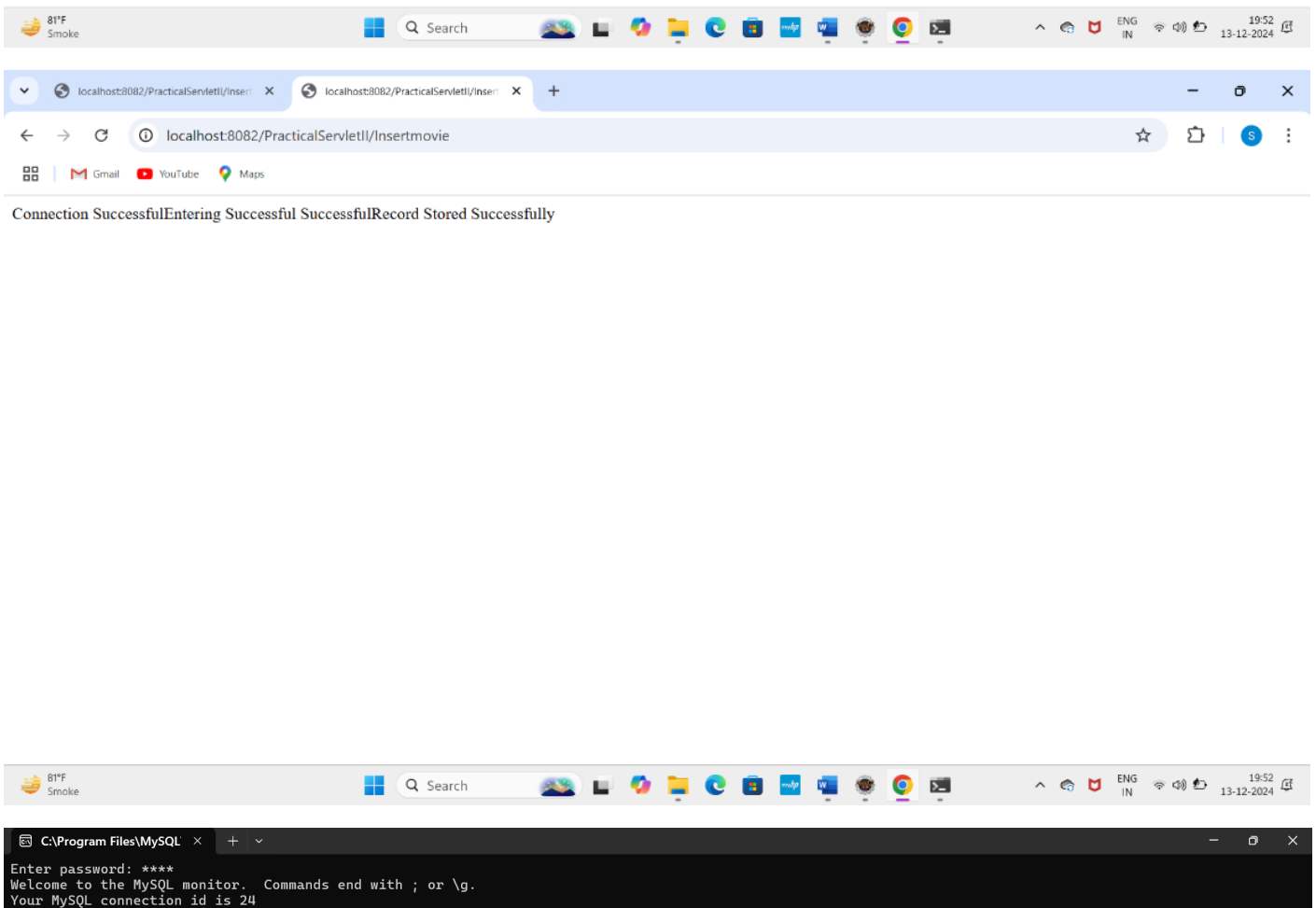
```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
@WebServlet("/Insertmovie")
public class Insertmovie extends HttpServlet {
        private static final long serialVersionUID = 1L;
    public Insertmovie() {
        super();
        // TODO Auto-generated constructor stub
                                                                    }
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                // TODO Auto-generated method stub
                PrintWriter pw=response.getWriter();
                response.setContentType("text/html");
                String name=request.getParameter("name");
```

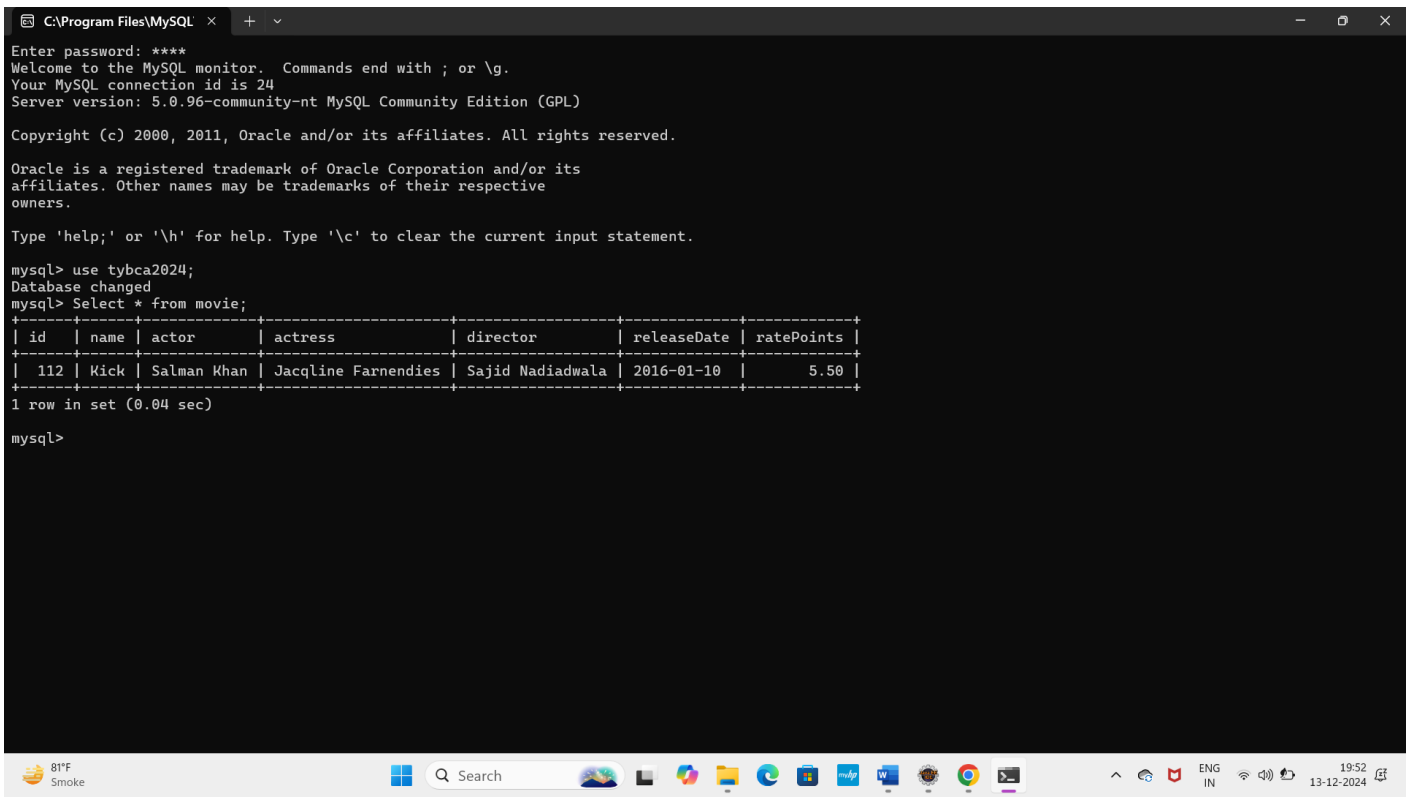
```
int id=Integer.parseInt(request.getParameter("Id"));

String actor=request.getParameter("actor");
String actress=request.getParameter("actress");
String director=request.getParameter("director");
String releaseDate=request.getParameter("rDate");
String rPoints=request.getParameter("rPoint");
float ratePoints=Float.parseFloat(rPoints);

try {
        Class.forName("com.mysql.jdbc.Driver");
        String url="jdbc:mysql://localhost:3306/tybca2024";
        String username="root";
        String password="root";

        Connection con=DriverManager.getConnection(url,username,password);
        pw.print("Connection Successful");
        String query="insert into movie(id,name,actor,actress,director,releaseDate,ratePoint)
values(?,?,?,?,?,?,?);";
        PreparedStatement pstmt=con.prepareStatement(query);
        pstmt.setInt(1, id);
        pstmt.setString(2, name);
        pstmt.setString(3, actor);
        pstmt.setString(4, actress);
        pstmt.setString(5, director);
        pstmt.setString(6, releaseDate);
        pstmt.setFloat(7, ratePoints);
        pw.print("Entering Successful Successful");
        int rowsInserted=pstmt.executeUpdate();

        if(rowsInserted==1); {
        pw.print("Record Stored Successfully");

        }

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

```
        }

}
```

OUTPUT


localhost:8082/PracticalServletII/Insert
Insert title here
localhost:8082/PracticalServletII/Movie.html
Gmail
YouTube
Maps
Enter Movie Id: 112
Enter Movie Name: Kick
Enter Actor Name: Salman Khan
Enter Actress Name: Jacqline Farnendies
Enter Director Name: Sajid Nadiadwala
Enter Release Date: 2016-01-10
Enter Rate Point: 05.5
Add movie


localhost:8082/PracticalServletII/Insert
localhost:8082/PracticalServletII/Insert
localhost:8082/PracticalServletII/Insertmovie
Gmail
YouTube
Maps
Connection SuccessfulEntering Successful SuccessfulRecord Stored Successfully


C:\Program Files\MySQL
Enter password: ****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 5.0.96-community-nt MySQL Community Edition (GPL)
Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use tybca2024;
Database changed
mysql> Select * from movie;
| id | name | actor | actress | director | releaseDate | ratePoints |
| 112 | Kick | Salman Khan | Jacqline Farnendies | Sajid Nadiadwala | 2016-01-10 | 5.50 |
1 row in set (0.04 sec)
mysql>

PRACTICAL-2

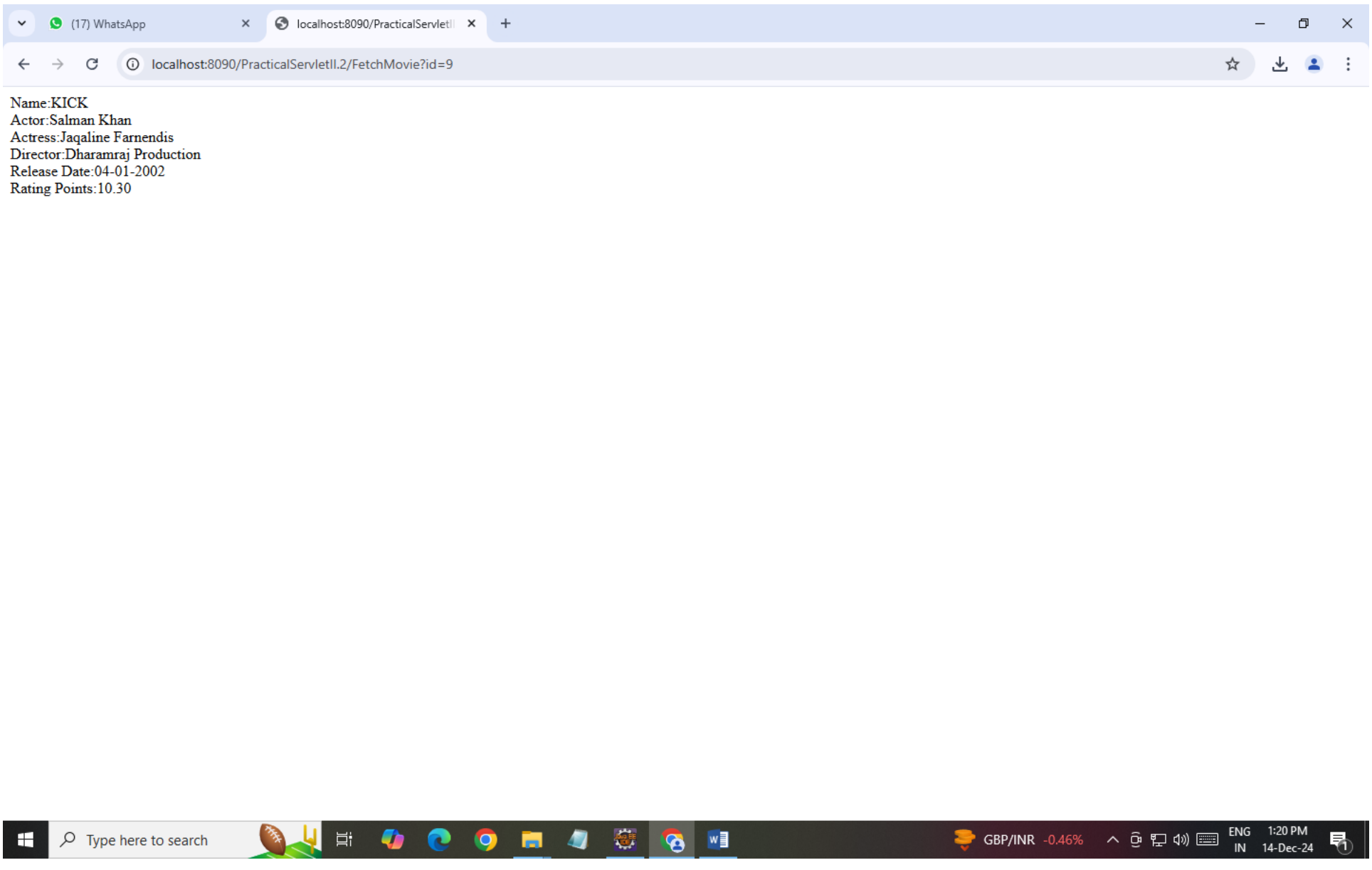
Servlet-2(3):-Write a Servlet to fetch the movie at id 9.

Code:- FetchMovie.java

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
public class FetchMovie extends HttpServlet {
	private static final long serialVersionUID = 1L;
    public FetchMovie() {
        super();
        // TODO Auto-generated constructor stub
    }
	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
		// TODO Auto-generated method stub

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

int id=Integer.parseInt(request.getParameter("Id"));

try {
		Class.forName("com.mysql.jdbc.Driver");
		String url="jdbc:mysql://localhost:3306/tybca2024";
		String username="root";
		String password="root";

		Connection con=DriverManager.getConnection(url,username,password);

		String query="Select * from moviee where Id=?";
```

```
PreparedStatement pstmt=con.prepareStatement(query);
pstmt.setInt(1, id);

ResultSet rs=pstmt.executeQuery();

if(rs.next()) {
        pw.println("Name:"+rs.getString("name")+"<br>");
        pw.println("Actor:"+rs.getString("actor")+"<br>");
        pw.println("Actress:"+rs.getString("actress")+"<br>");
        pw.println("Director:"+rs.getString("director")+"<br>");
        pw.println("Release Date:"+rs.getString("releaseDate")+"<br>");
        pw.println("Rating Point:"+rs.getString("ratePoint")+"<br>");
}
        else {
                pw.print("No record found..");
        }

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

OUTPUT



PRACTICAL-2

Servlet-2(4):-Write a servlet to fetch all the movie released between 3rd Jan 2015 and 3rd Jan 2016.

Code:-FetchMovie.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="FetchMoviesRelease" method="post">
<h3>Press OK to fetch all the movies released between</h3><h3>3rd jan 2015 and 3rd jan 2016</h3>
    <input type="submit" value="OK" />
</form>
</body>
</html>
```

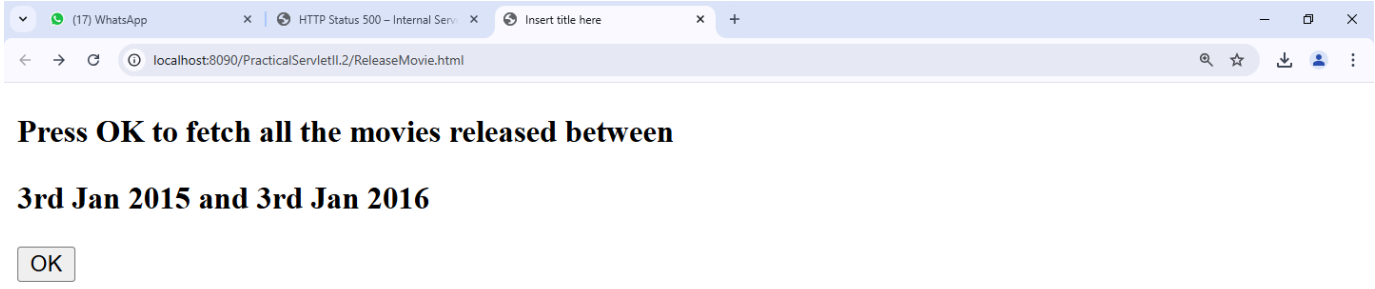
Code:-FetchMoviesReleased.java

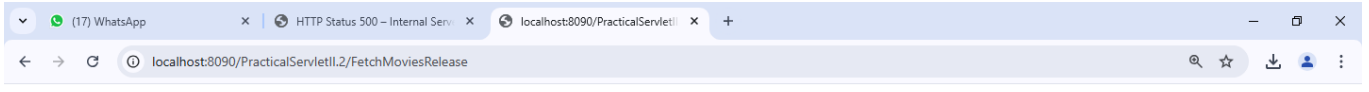
```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.ResultSetMetaData;
public class FetchMoviesRelease extends HttpServlet {
        private static final long serialVersionUID = 1L;
      public FetchMoviesRelease() {
      super();
      // TODO Auto-generated constructor stub
   }


      protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
```

```
		// TODO Auto-generated method stub
		PrintWriter out=response.getWriter();
		try {
			Class.forName("com.mysql.jdbc.Driver");
			String url="jdbc:mysql://localhost:3306/tybca2024";
			String username="root";
			String password="root";

			Connection con=DriverManager.getConnection(url,username,password);
			PreparedStatement ps = con.prepareStatement("select * from moviee where releaseDate
BETWEEN '2001-01-03' AND '2016-01-03'");
			ResultSet rs = ps.executeQuery();
			out.print("<table width=50% border=1>");
		out.print("<caption>Movie detail:</caption>");
		out.print("<tr>");
	ResultSetMetaData rsmd= rs.getMetaData();
		int total = rsmd.getColumnCount();
		for (int i = 1; i<= total; i++) {
			out.print("<th>" + rsmd.getColumnName(i)+ "</th>");

		}
		out.print("</tr>");
		while(rs.next()) {
			out.print("</tr>");
			for (int i =1; i <= total; i++) {
				out.print("<td>" +rs.getString(i) + "</td>");
			}
			out.print("</tr>");

			}
		out.print("</table>");

		}catch(ClassNotFoundException | SQLException ex) {
			out.println(ex);

			}finally {
			out.close();
		}

}
```

```
private void getMetaData() {
        // TODO Auto-generated method stub


}
        }
```

OUTPUT





Movie Details:

id	name	actor	actress	director	releaseDate	ratePoints
90	Hum	Amitbachan	Kimi Katkar	XYZ	2015-01-07	9.70
7	Dangal	Amir Kahn	Geeta-Babeeta	Nitesh Tiwari	2011-07-24	11.20

PRACTICAL-3

Servlet-3(1):-Write a servlet program to accept a number, if number is even redirect to purplesq.com if number is odd redirect to google.com.

Code:-AcceptNoRedirection.html

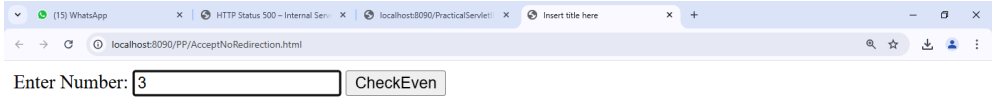
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<form name ="frm" action="CheckEven" method="post">
Enter Number: <input type="text" name="num" >
<input type="submit" value="CheckEven">
</form>
</body>
</html>
```

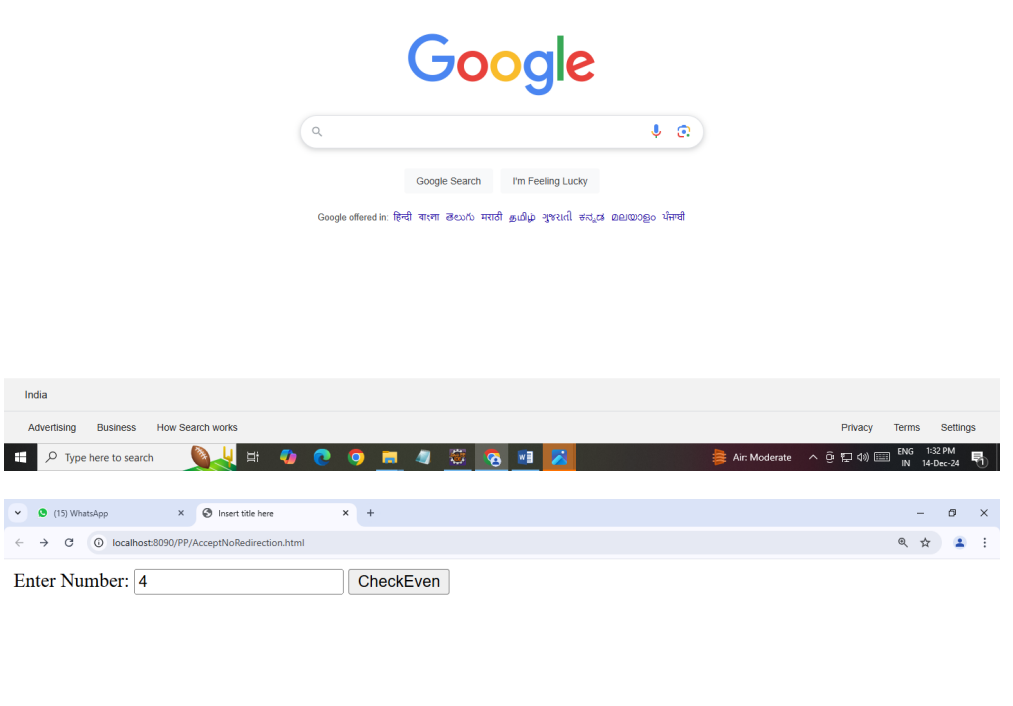
Code:-CheckEven.java

```
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
public class checkeven extends HttpServlet {
	private static final long serialVersionUID = 1L;
    public checkeven() {
      super();
      // TODO Auto-generated constructor stub
   }
	protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
		// TODO Auto-generated method stub
		PrintWriter pw;
		response.setContentType("text/html");
		pw=response.getWriter();
		 int i=Integer.parseInt(request.getParameter("num"));
		  if(i%2==0)
			response.sendRedirect("http://www.purplesq.com");
```

```
            else
                response.sendRedirect("http://www.google.com");
        }
}
```

OUTPUT





PRACTICAL-3

Servlet-3(2):-Write a servlet program to redirect the request to another servlet which requires a String as the Parameter and the other Servlet and the other servlet converts the string to lowercase.

Code:-lowercase.java

```
import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.io.PrintWriter;

public class lowercase extends HttpServlet {

	private static final long serialVersionUID = 1L;

    public lowercase() {

        super();

        // TODO Auto-generated constructor stub

    }

	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

		// TODO Auto-generated method stub

		PrintWriter pw=response.getWriter();

		response.setContentType("text/html");


		String text=request.getParameter("text");

		text=text.toLowerCase();


		pw.println("<html><body>");

		pw.println("<h1>Lower Case Servlet</h1>");

		pw.println("<b>"+text+"</b>");

	  pw.println("</body></html>");


		pw.close();



	}


}
```

PRACTICAL-3

Servlet-3(3):-Write a Java program to get the name from a html form, put the name in session and redirect the floe to another servlet and the other servlet displays the name put into the session.

Code:-Acceptname.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="Acceptname" method="post">
Name: <input type="text" name="name"/>
<input type="submit" value="Enter name">
</form>
</body>
</html>
```

Code:-AnotherServlet.java

```
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
public class AnotherServlet extends HttpServlet {
        private static final long serialVersionUID = 1L;
      public AnotherServlet() {
       super();
       // TODO Auto-generated constructor stub
   }
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                // TODO Auto-generated method stub
                HttpSession session=request.getSession(false);
                 PrintWriter pw=response.getWriter();
                 response.setContentType("text/html");
```

```
        if(session==null) {
                RequestDispatcher dispatacher=request.getRequestDispatcher("/Acceptname.html");
                dispatacher.forward(request, response);
        }
        else {
                String name=(String)session.getAttribute("name");
                pw.println(name+"is receieved from session");
        }
        pw.close();

        }

}
```

Code:Acceptname.java

```
import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import jakarta.servlet.http.HttpSession;
import java.io.IOException;
public class Acceptname extends HttpServlet {
         private static final long serialVersionUID = 1L;
      public Acceptname() {
        super();
        // TODO Auto-generated constructor stub
    }
        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
                // TODO Auto-generated method stub
                HttpSession session=request.getSession(true);
                String name=request.getParameter("name");

                session.setAttribute("name", name);
                RequestDispatcher dispatcher=request.getRequestDispatcher("AnotherServlet");
                dispatcher.forward(request, response);
        }
}
```


Insert title here
localhost:8090/PP/Acceptname.html
Name: Sana
submit

Type here to search
Watchlist ideas
ENG
IN
1:52 PM
14-Dec-24

localhost:8090/PP/AcceptName
localhost:8090/PP/AcceptName
Sana is received from session


Type here to search
Watchlist ideas
ENG
IN
1:52 PM
14-Dec-24

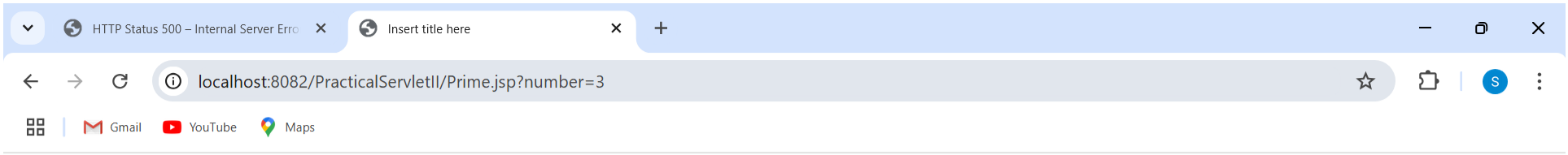
PRACTICAL-4

JSP-1(1):-Write a JSP code to accept a number and revert whether the number is prime or not.

Code:Prime.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<% int number=Integer.parseInt(request.getParameter("number"));

boolean flag = false;

for (int i= 2; i <= number / 2; ++i){
        if(number % i == 0){
                flag = true;
                break;
        }
}
if(!flag){
%>
<h1><%= number %>is a Prime Number</h1>
<%
}
else {
    %>
    <h1> <%= number%> is NOT a Prime Number</h1>

    <%
    }

    %>

    </body>
```

</html>



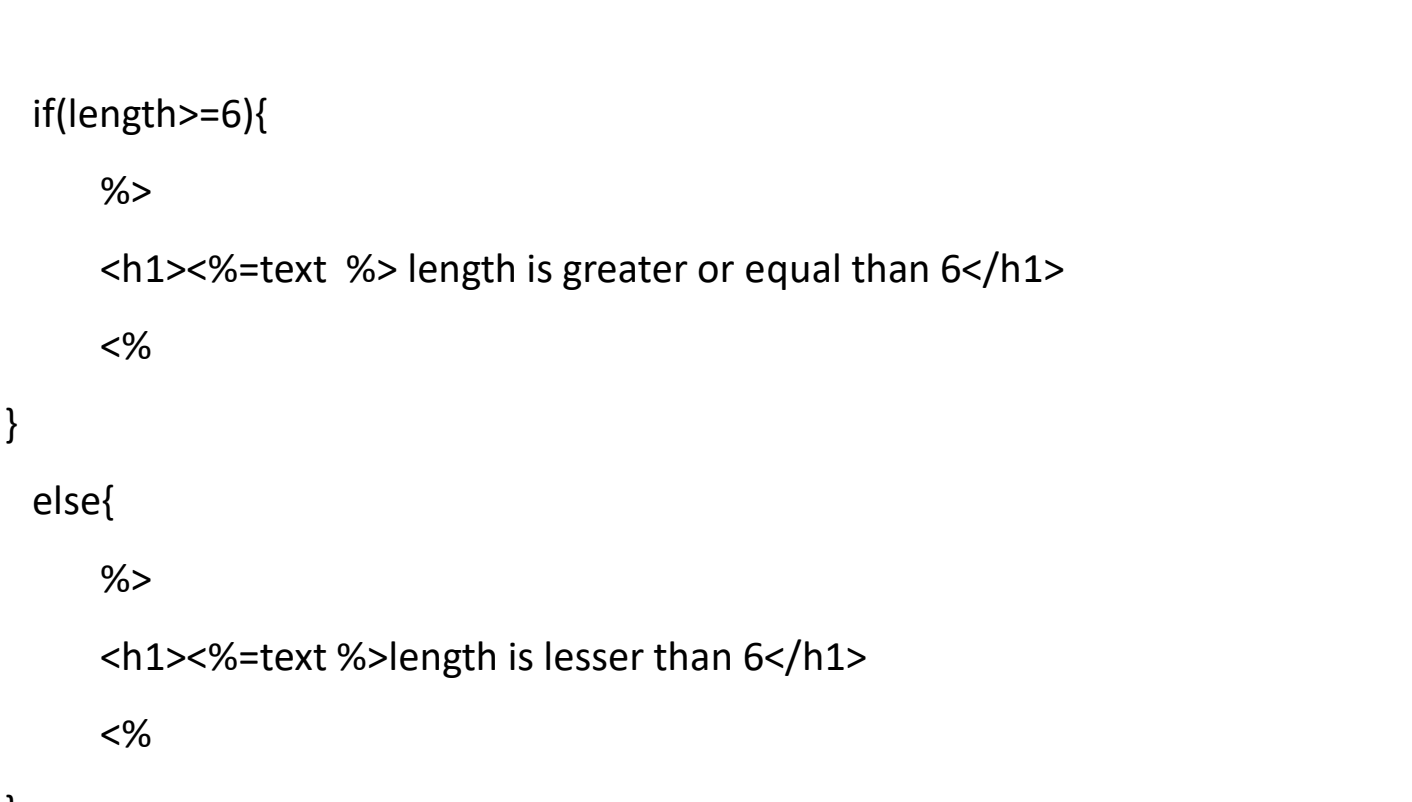
3is a Prime Number

PRACTICAL-4

JSP-1(2)Write a servlet Program to accept a String and determine whether the Strings length is greater than 6.

Code:-lengthCheck.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
   String text=request.getParameter("text");
   int length=text.length();

   if(length>=6){
         %>
         <h1><%=text  %> length is greater or equal than 6</h1>
         <%
}
   else{
         %>
         <h1><%=text %>length is lesser than 6</h1>
         <%
}
%>
</body>
</html>
```

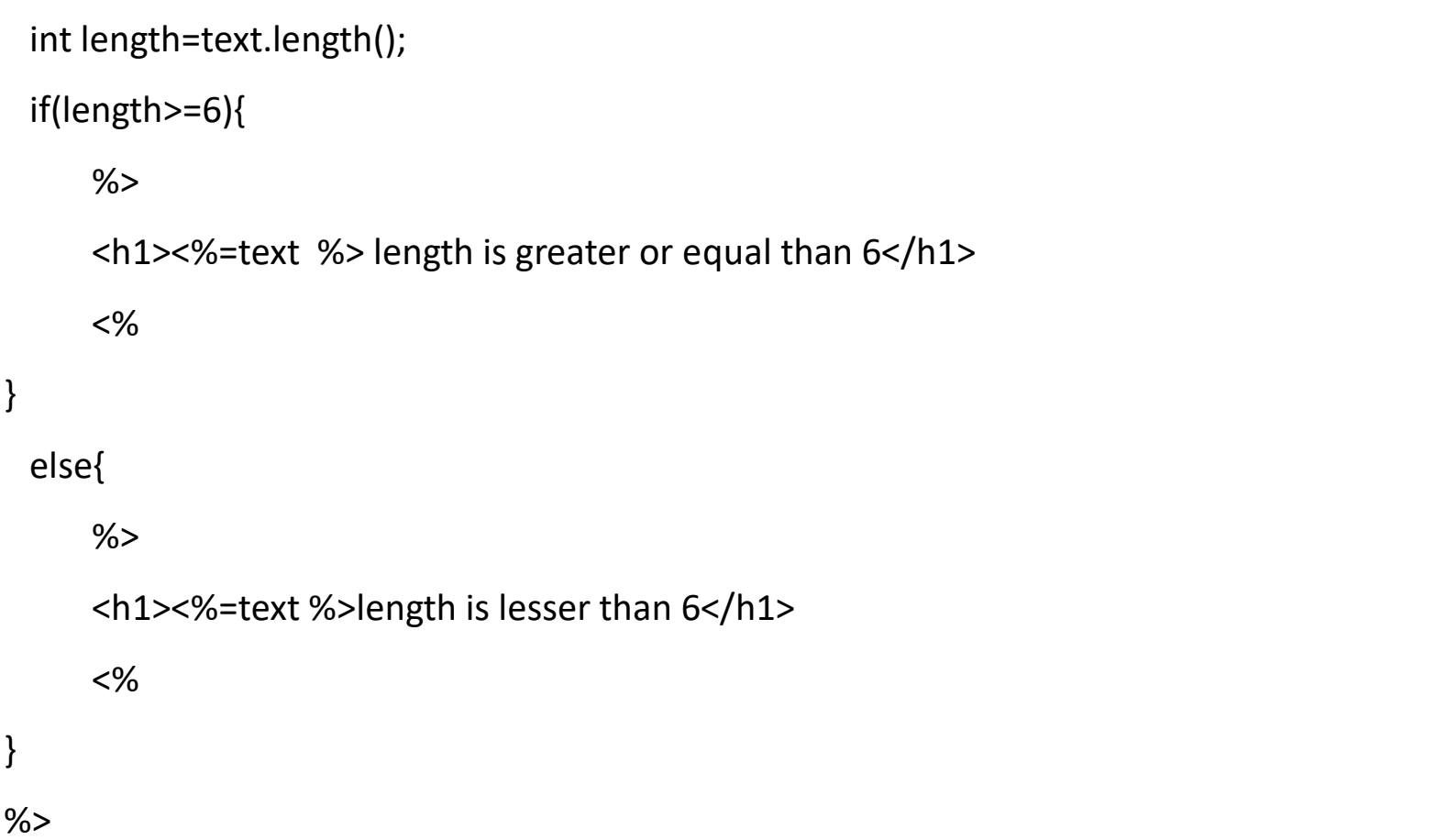


PRACTICAL-4

JSP-1(3):-Write a JSP program to redirect to Google.com.

Code:-Redirect.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
  String text=request.getParameter("text");
  int length=text.length();
  if(length>=6){
        %>
        <h1><%=text  %> length is greater or equal than 6</h1>
        <%
}
  else{
        %>
        <h1><%=text %>length is lesser than 6</h1>
        <%
}
%>
</body>
</html>
```



PRACTICAL-5

JSP-2(1):-Write a Java Program to print the following.

11111

2222

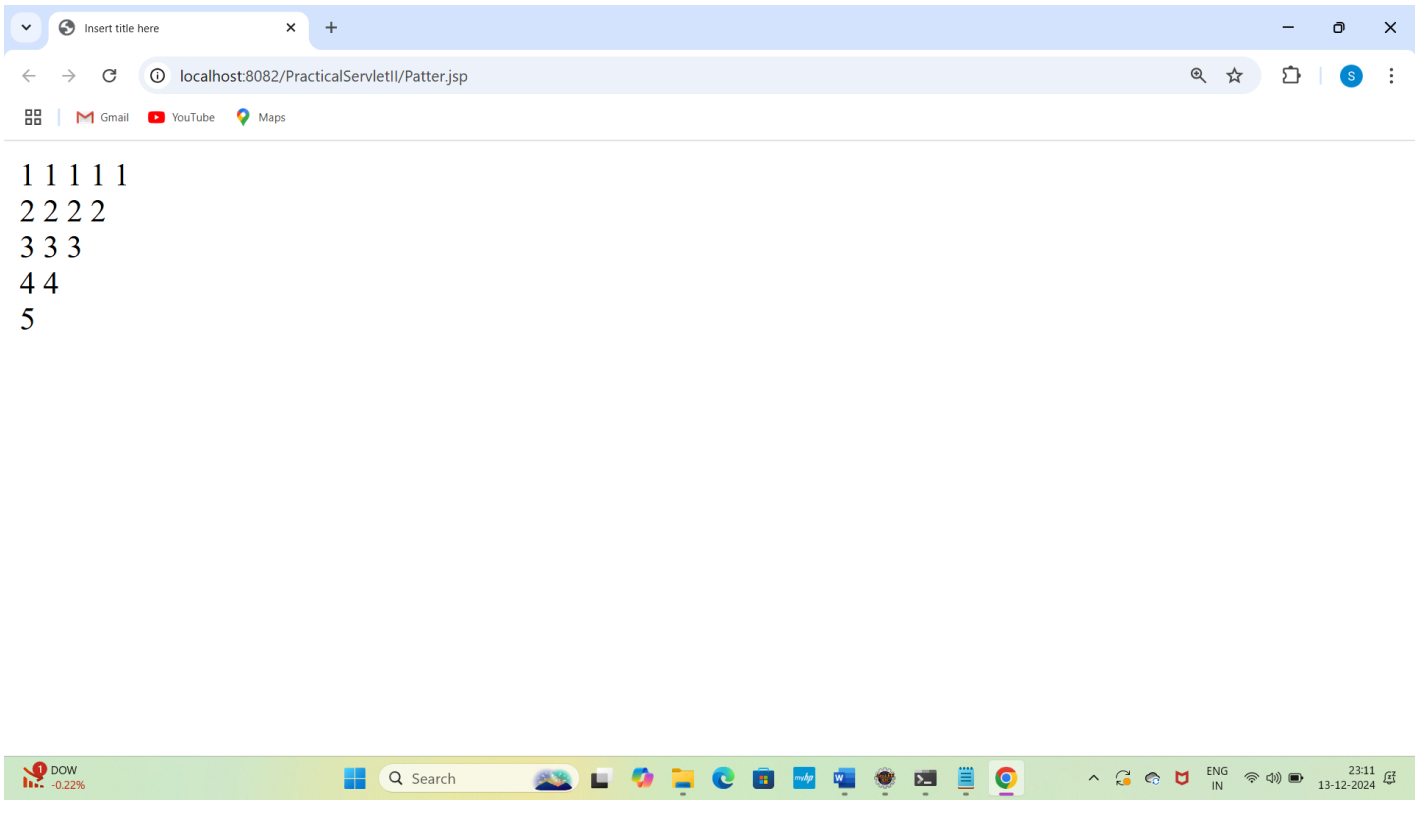
333

44

5

Code:- Pattern.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
int n=5;
for(int i=1; i<=n;i++)
{
        for(int j=1;j<=n-i+1;j++){
                out.println(i);
        }
        out.println("<br>");
        }
%>
</body>
</html>
```



PRACTICAL-5

JSP-2(2):-Write a JSP Code to store the data from a form into a database table using JDBC.

Code:-

PRACTICAL-5

JSP-2(3):-Make 2 JSP Pages, accept the username and password from first page, if the username is tom and password is tommy, login will be successful and redirect it to loggedin.jsp page and store the username in session from login and display it on loggedin.jsp.

Code:-login.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="login.jsp">
User name:<input type="text" name="username">
<br><br>
Password:<input type="password" name="password">
<br><br>
<input type="submit">
</form>
</body>
</html>
```

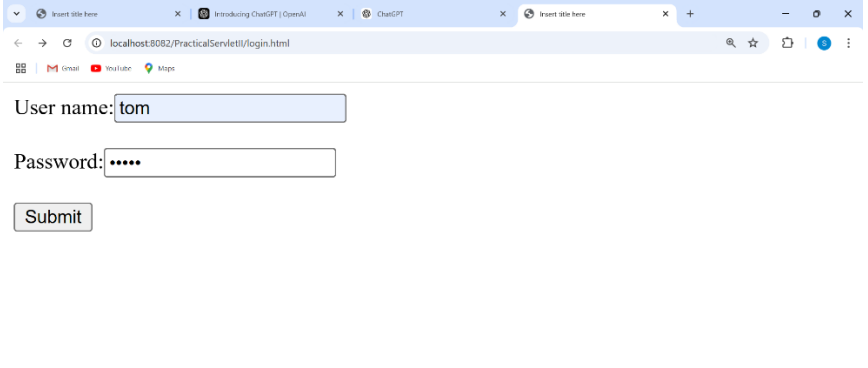
Code:-login.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
String username=request.getParameter("username");
String password=request.getParameter("password");

 if(username.equals("tom")&& password.equals("tommy"))
 {
        session.setAttribute("username", username);
        RequestDispatcher dispatcher=request.getRequestDispatcher("/Loggedin.jsp");
        dispatcher.forward(request,response);
```

```
}
 else{
        session.invalidate();
        RequestDispatcher dispatcher=request.getRequestDispatcher("/login.html");
        dispatcher.forward(request,response);

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

Code:-loggedin.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
   pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<%
  String username=(String)session.getAttribute("username");
%>
<h1>Welcome <%=username %></h1>
</body>
</html>
```



Welcome tom

PRACTICAL-6

JDBC-1(1):-Create the database as given below. Enter the sample data.

id	temp	city	precipitation	wind	humidity	date
int	float	varchar	float	int	float	date
	25.5C	mumbai	3.0%	25km/hr	20%	2016-04-18

C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe

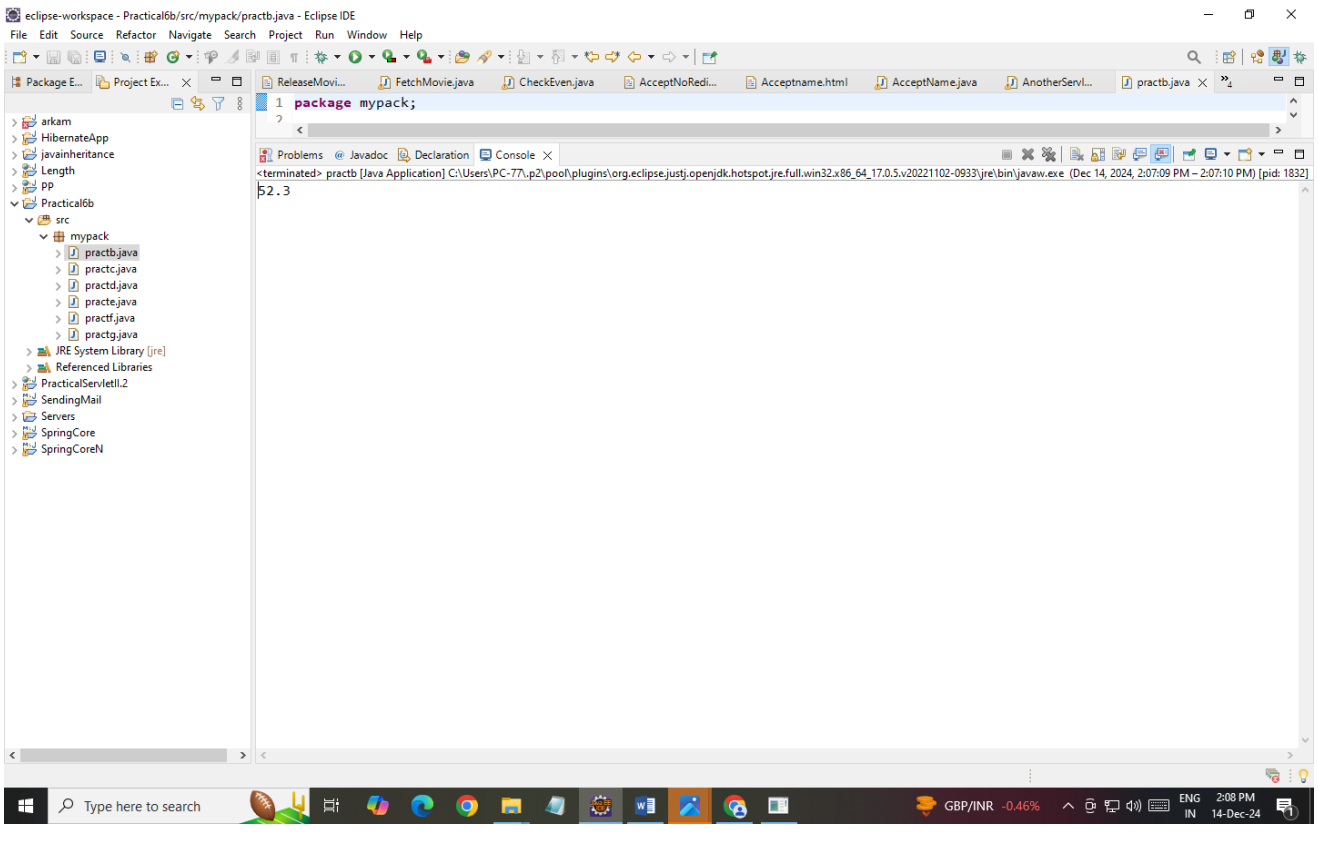
```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.0.37-community-nt MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use tubca2024;
ERROR 1049 (42000): Unknown database 'tubca2024'
mysql> use tybca2024;
Database changed
mysql> Select * from sample1;
+------+-------+----------+---------------+------+----------+------------+
| id   | temp  | city     | precipitation | wind | humidity | date       |
+------+-------+----------+---------------+------+----------+------------+
|    1 | 34.20 | Kurla    |         12.40 |   23 |     4.30 | 2017-10-10 |
|    2 | 39.70 | Mumbai   |         13.60 |   24 |     3.60 | 2016-01-10 |
|    3 | 25.60 | Banglore |         15.70 |   22 |     4.30 | 2016-01-15 |
|    4 | 12.60 | Delhi    |         17.50 |   16 |     4.30 | 2016-01-29 |
|    5 | 50.40 | Mumbai   |         23.60 |   19 |     7.70 | 2016-01-19 |
|    6 | 13.60 | Mumbai   |         17.40 |   20 |     5.70 | 2016-01-27 |
|    7 | 52.30 | Mumbai   |         34.40 |   24 |     6.60 | 2016-01-13 |
|    8 | 42.30 | Banglore |         22.20 |   27 |     6.70 | 2016-02-24 |
|    9 | 45.30 | Banglore |         19.30 |   26 |     7.40 | 2016-02-26 |
|   10 | 27.70 | Banglore |         17.70 |   29 |     8.20 | 2016-02-03 |
+------+-------+----------+---------------+------+----------+------------+
10 rows in set (0.00 sec)

mysql>
```

PRACTICAL-6

JDBC-1(2):-Show the highest temperature for the city of Mumbai for the month of January 2016.

Code:-practicalb.java

```
package Mypack;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class practicalb {
public static void main(String[] args) {
                // TODO Auto-generated method stub
   String url ="jdbc:mysql://localhost:3306/tybca2024";
   String user = "root";
   String password ="root";
   String query = "SELECT max(temp) as htemp FROM sample2 WHERE city='Mumbai' and MONTH(date)=01
and YEAR(date)=2016";
   try
   {
      Connection conn = DriverManager.getConnection(url, user, password);
      PreparedStatement preparedStatement = conn.prepareStatement(query);
      ResultSet resultSet = preparedStatement.executeQuery();{
            resultSet.next();
            System.out.println(resultSet.getDouble("htemp"));
            }
      }catch (SQLException e) {
            e.printStackTrace();
      }
}
```



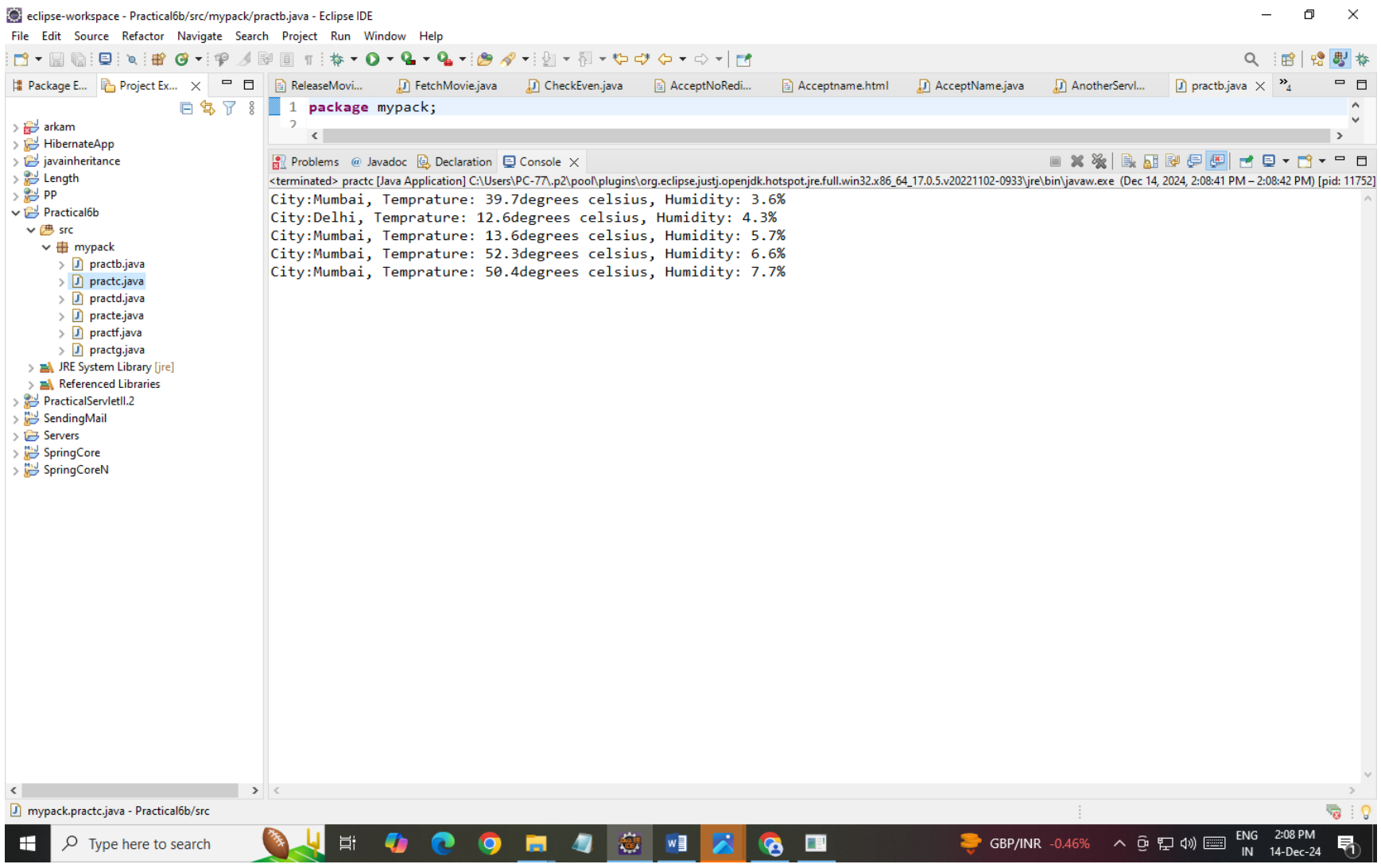
PRACTICAL-6

JDBC-1(3):-List the temperatures for the city of Mumbai and Delhi in order of their humidity(from highest to lowest).

Code:-practicalc.java

```
package Mypack;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class practicalc {

	public static void main(String[] args) {

		// TODO Auto-generated method stub

		String url ="jdbc:mysql://localhost:3306/tybca2024";

	String user = "root";

	String password ="root";

	String query = "SELECT city, temp, humidity FROM sample2 " + "WHERE city IN ('Mumbai', 'Delhi')ORDER BY humidity";

	try (

		Connection conn = DriverManager.getConnection(url, user, password);

		PreparedStatement preparedStatement = conn.prepareStatement(query);

		ResultSet resultSet = preparedStatement.executeQuery()){

			while(resultSet.next()){

				String city = resultSet.getString("city");

				double temperature = resultSet.getDouble("temp");

				double humidity = resultSet.getDouble("humidity");

				System.out.println("city: " + city + ", temperatur: "+ temperature +" degree Celsius, Humidity:" + humidity + "%");

			}

		}catch (SQLException e) {

				e.printStackTrace();

				}

	}

}
```

OUTPUT


eclipse-workspace - Practical6b/src/mypack/practb.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E... Project Ex...
arkam
HibernateApp
javainheritance
Length
PP
Practical6b
src
mypack
practb.java
practc.java
practd.java
practe.java
practf.java
practg.java
JRE System Library [jre]
Referenced Libraries
PracticalServletII.2
SendingMail
Servers
SpringCore
SpringCoreN
ReleaseMovi... FetchMovie.java CheckEven.java AcceptNoRedi... Acceptname.html AcceptName.java AnotherServl... practb.java
1 package mypack;
Problems Javadoc Declaration Console
<terminated> practc [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 2:08:41 PM – 2:08:42 PM) [pid: 11752]
City:Mumbai, Temprature: 39.7degrees celsius, Humidity: 3.6%
City:Delhi, Temprature: 12.6degrees celsius, Humidity: 4.3%
City:Mumbai, Temprature: 13.6degrees celsius, Humidity: 5.7%
City:Mumbai, Temprature: 52.3degrees celsius, Humidity: 6.6%
City:Mumbai, Temprature: 50.4degrees celsius, Humidity: 7.7%
mypack.practc.java - Practical6b/src
Type here to search
GBP/INR -0.46%
ENG IN
2:08 PM 14-Dec-24

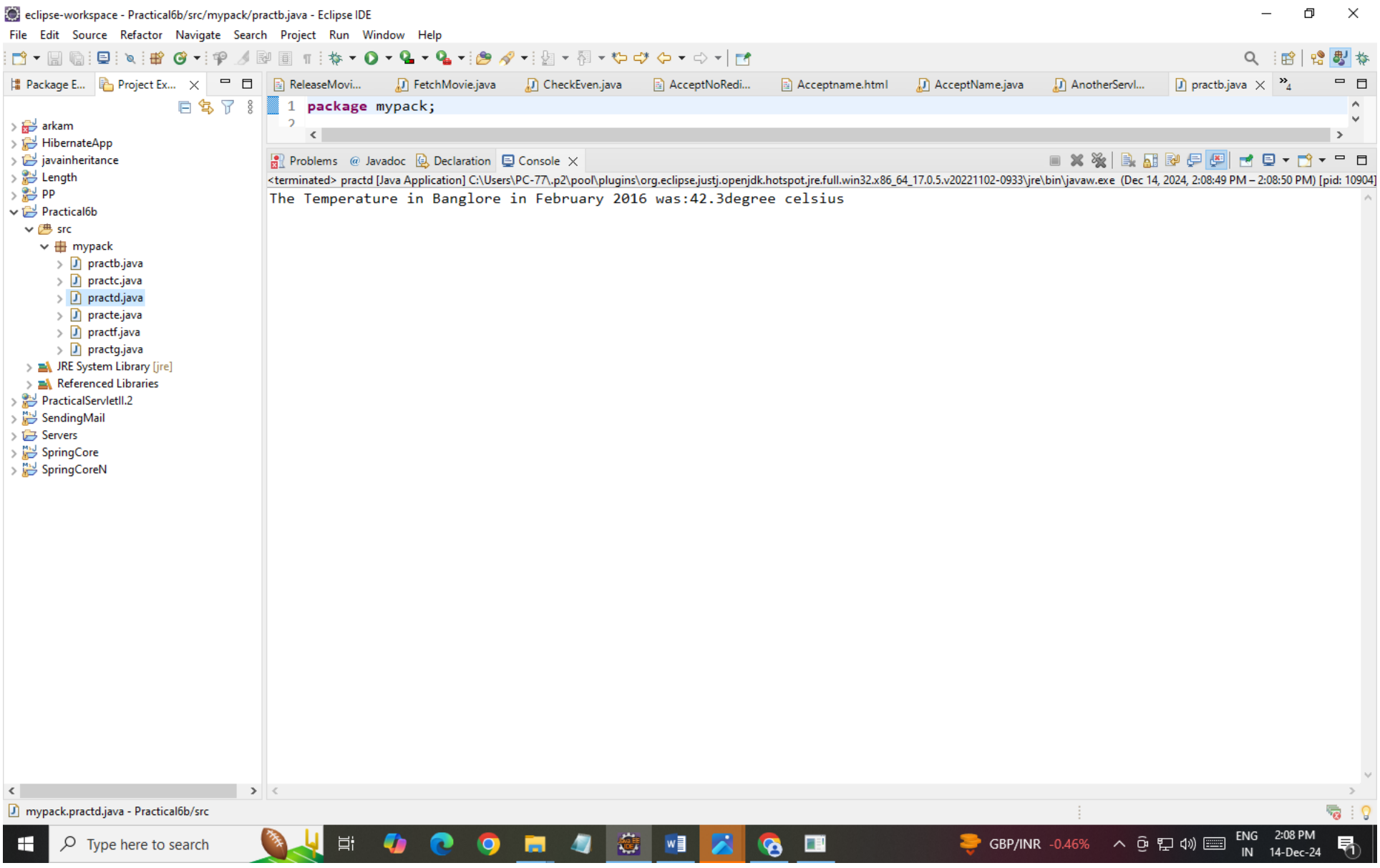
PRACTICAL-6

JDBC-1(4):-List the temperature for the city of Banglore for the month of February 2016.

Code:-practicalD.java

```
package Mypack;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
public class practicalD {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String url ="jdbc:mysql://localhost:3306/tybca2024";
	String user = "root";
	String password ="root";
	String query = "SELECT temp FROM sample2 " + "WHERE city ='banglore' AND MONTH(date) = 2 AND YEAR(date)=2016";
	try (
		Connection conn = DriverManager.getConnection(url, user, password);
		PreparedStatement preparedStatement = conn.prepareStatement(query);
		ResultSet resultSet = preparedStatement.executeQuery()){
			if (resultSet.next()){
				double temperature = resultSet.getDouble("temp");
				System.out.println("The temperature in banglor is Februry 2016 was:"+ temperature + "degrees Celsius");
			}
			else {
				System.out.println("No temperature data found for Banglore in February 2016");
			}
	}catch (SQLException e) {
				e.printStackTrace();
				}
}

}
```

OUTPUT


eclipse-workspace - Practical6b/src/mypack/practb.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E... Project Ex...
arkam
HibernateApp
javainheritance
Length
pp
Practical6b
src
mypack
practb.java
practc.java
practd.java
practe.java
practf.java
practg.java
JRE System Library [jre]
Referenced Libraries
PracticalServletII.2
SendingMail
Servers
SpringCore
SpringCoreN
ReleaseMovi... FetchMovie.java CheckEven.java AcceptNoRedi... Acceptname.html AcceptName.java AnotherServl... practb.java
1 package mypack;
Problems Javadoc Declaration Console
<terminated> practd [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 2:08:49 PM – 2:08:50 PM) [pid: 10904]
The Temperature in Banglore in February 2016 was:42.3degree celsius
mypack.practd.java - Practical6b/src
Type here to search
GBP/INR -0.46%
ENG
IN
2:08 PM
14-Dec-24

PRACTICAL-6

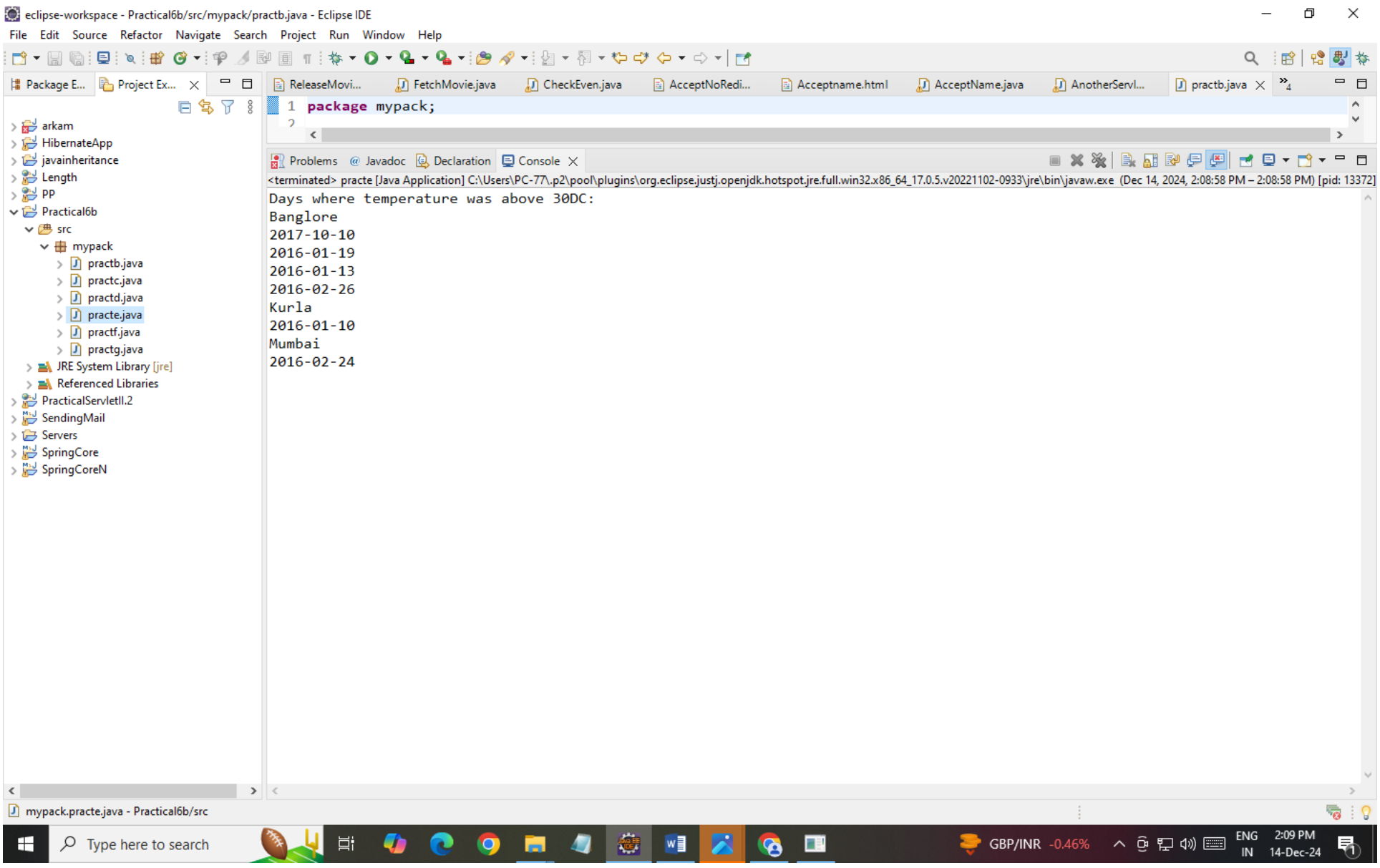
JDBC-1(5):-Show all the days where temperature was above 30dc,using Set.

Code:-practicale.java

```
package Mypack;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.HashSet;
import java.util.Set;
public class practicalE {
	public static void main(String[] args) {
	// TODO Auto-generated method stub
	String url ="jdbc:mysql://localhost:3306/tybca2024";
String user = "root";
String password ="root";
Set<String> highTemperatureDays = new HashSet<>();
String query = "SELECT date, city FROM sample2 WHERE temp > 30";
try (
	Connection conn = DriverManager.getConnection(url, user, password);
	PreparedStatement preparedStatement = conn.prepareStatement(query);
	ResultSet resultSet = preparedStatement.executeQuery()){
		while(resultSet.next()){
			String date = resultSet.getString("date");
			String city = resultSet.getString("city");
			highTemperatureDays.add(date);
			highTemperatureDays.add(city);
		}
		if(!highTemperatureDays.isEmpty()) {
			System.out.println("Day where temperaturte was above 30 C:");
			for (String day : highTemperatureDays) {
				System.out.println(day);
			}
			}else {
				System.out.println("No day found where temperature was above 30 c");
			}
}catch (SQLException e) {
			e.printStackTrace();
```

```
        }


    }
    }
```

OUTPUT


eclipse-workspace - Practical6b/src/mypack/practb.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E... Project Ex...
arkam
HibernateApp
javainheritance
Length
PP
Practical6b
src
mypack
practb.java
practc.java
practd.java
practe.java
practf.java
practg.java
JRE System Library [jre]
Referenced Libraries
PracticalServletII.2
SendingMail
Servers
SpringCore
SpringCoreN
ReleaseMovi... FetchMovie.java CheckEven.java AcceptNoRedi... Acceptname.html AcceptName.java AnotherServl... practb.java
1 package mypack;
Problems Javadoc Declaration Console
<terminated> practe [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 2:08:58 PM – 2:08:58 PM) [pid: 13372]
Days where temperature was above 30DC:
Banglore
2017-10-10
2016-01-19
2016-01-13
2016-02-26
Kurla
2016-01-10
Mumbai
2016-02-24
mypack.practe.java - Practical6b/src
Type here to search
GBP/INR -0.46%
ENG
IN
2:09 PM
14-Dec-24

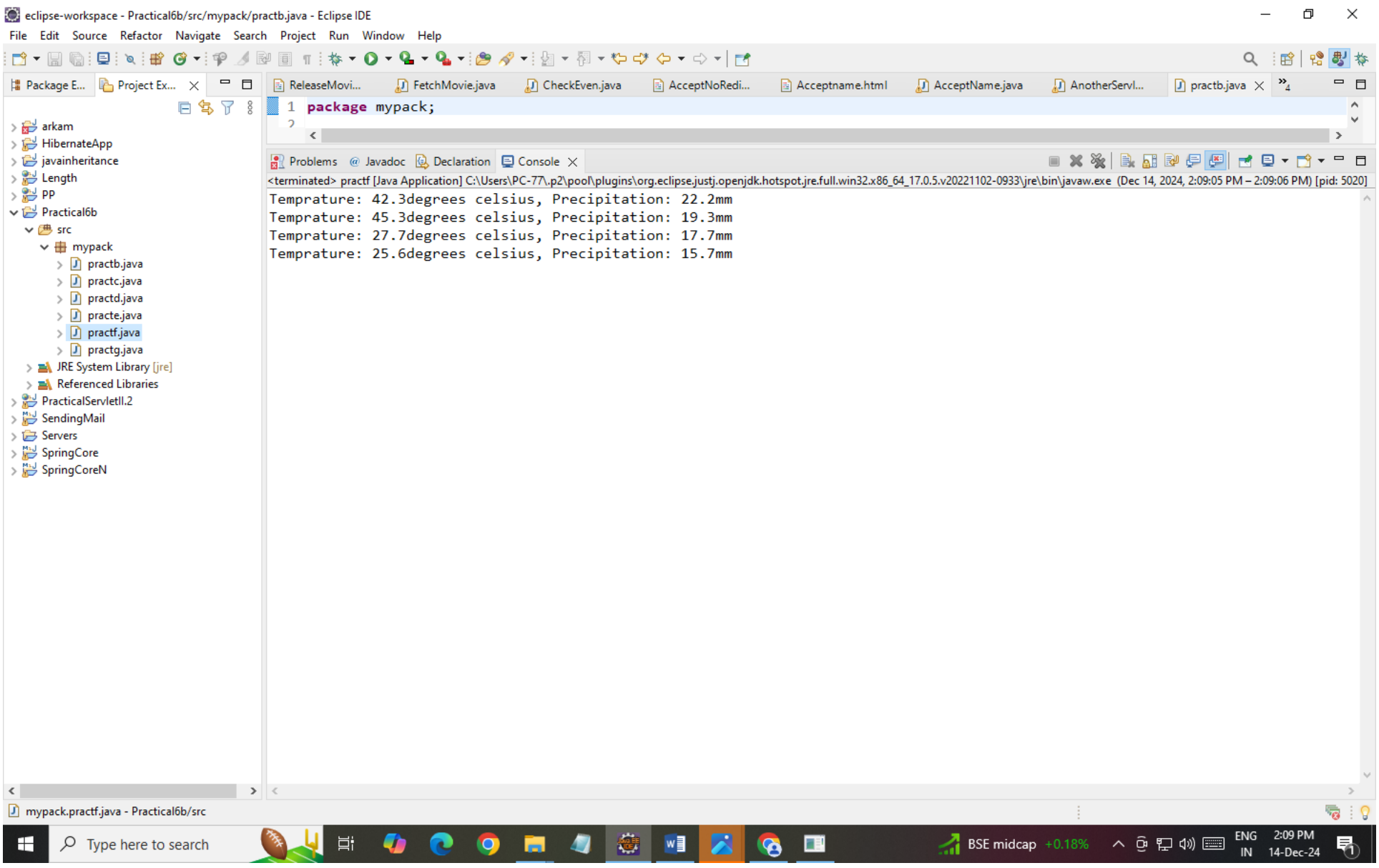
PRACTICAL-6

JDBC-1(6):-List the temperatures for the city of Banglore in order of their precipitation(from highest to lowest).

Code:-practicalf.java

```
package Mypack;


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


public class practicalF {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String url ="jdbc:mysql://localhost:3306/tybca2024";
	String user = "root";
		String password ="root";
		String query = "SELECT temp, precipitation FROM sample2  WHERE city = 'banglore' ORDER BY precipitation DESC";
		try (
			Connection conn = DriverManager.getConnection(url, user, password);
			PreparedStatement preparedStatement = conn.prepareStatement(query);
			ResultSet resultSet = preparedStatement.executeQuery()){
				while (resultSet.next()){
					double temperature = resultSet.getDouble("temp");
					double precipitation = resultSet.getDouble("precipitation");
					System.out.println("Temperature: "+ temperature + "degrees Celsius, Precipitation" + precipitation +"mm");
				}


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


	}
```

OUTPUT


eclipse-workspace - Practical6b/src/mypack/practb.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E... Project Ex...
arkam
HibernateApp
javainheritance
Length
PP
Practical6b
src
mypack
practb.java
practc.java
practd.java
practe.java
practf.java
practg.java
JRE System Library [jre]
Referenced Libraries
PracticalServletII.2
SendingMail
Servers
SpringCore
SpringCoreN
ReleaseMovi... FetchMovie.java CheckEven.java AcceptNoRedi... Acceptname.html AcceptName.java AnotherServl... practb.java
1 package mypack;
Problems Javadoc Declaration Console
<terminated> practf [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 2:09:05 PM – 2:09:06 PM) [pid: 5020]
Temprature: 42.3degrees celsius, Precipitation: 22.2mm
Temprature: 45.3degrees celsius, Precipitation: 19.3mm
Temprature: 27.7degrees celsius, Precipitation: 17.7mm
Temprature: 25.6degrees celsius, Precipitation: 15.7mm
mypack.practf.java - Practical6b/src
Type here to search
BSE midcap +0.18%
ENG
IN
2:09 PM
14-Dec-24

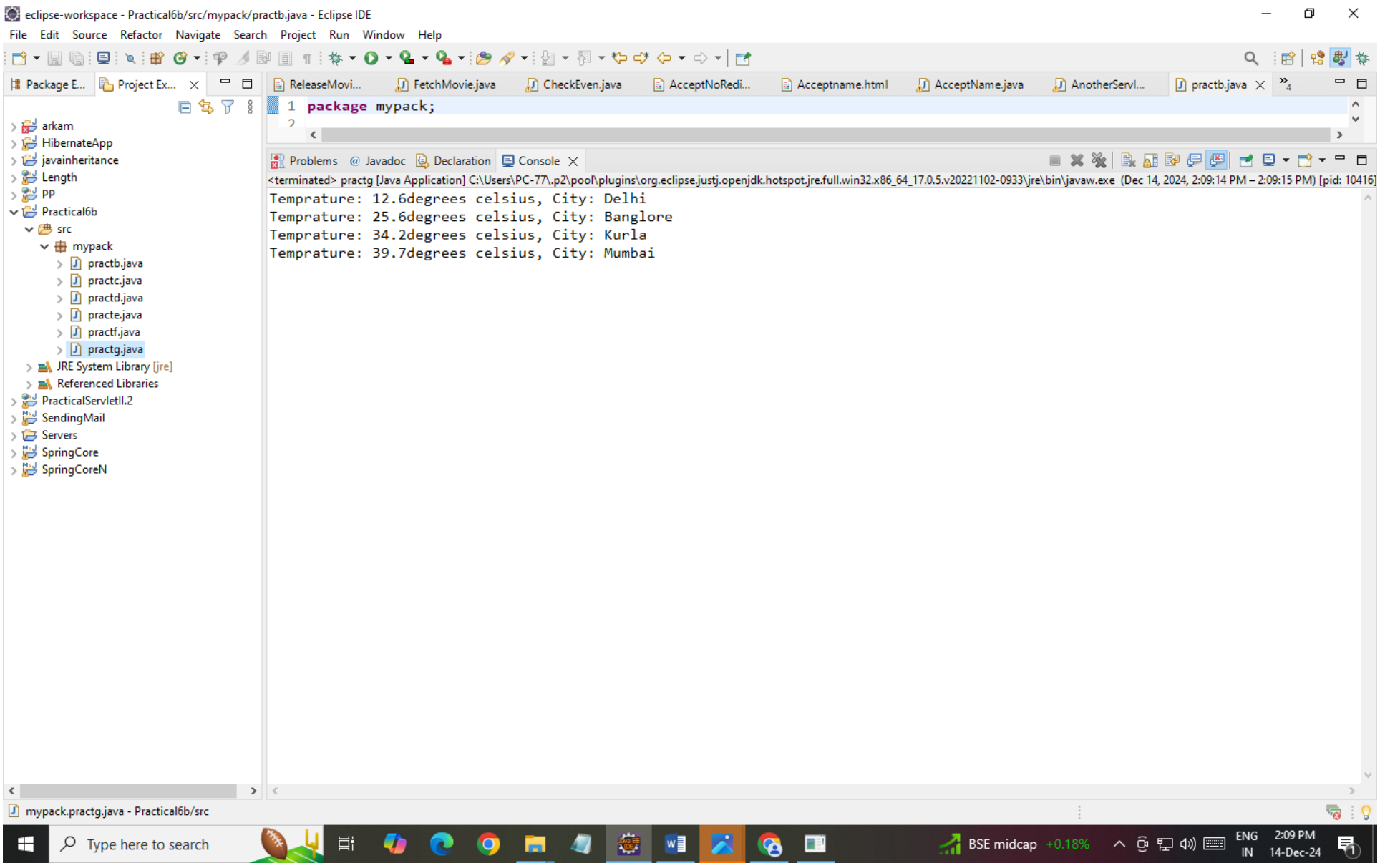
PRACTICAL-6

JDBC-1(7):-Sort the list of temperature according to the name of the cities.

Code:-practicalg.java

```
package Mypack;


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


public class practicalF {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		String url ="jdbc:mysql://localhost:3306/tybca2024";
	String user = "root";
		String password ="root";
		String query = "SELECT temp, precipitation FROM sample2  WHERE city = 'banglore' ORDER BY precipitation DESC";
		try (
			Connection conn = DriverManager.getConnection(url, user, password);
			PreparedStatement preparedStatement = conn.prepareStatement(query);
			ResultSet resultSet = preparedStatement.executeQuery()){
				while (resultSet.next()){
					double temperature = resultSet.getDouble("temp");
					double precipitation = resultSet.getDouble("precipitation");
					System.out.println("Temperature: "+ temperature + "degrees Celsius, Precipitation" + precipitation +"mm");
				}


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


	}
```

OUTPUT


eclipse-workspace - Practical6b/src/mypack/practb.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E... Project Ex...
arkam
HibernateApp
javainheritance
Length
pp
Practical6b
src
mypack
practb.java
practc.java
practd.java
practe.java
practf.java
practg.java
JRE System Library [jre]
Referenced Libraries
PracticalServletII.2
SendingMail
Servers
SpringCore
SpringCoreN
ReleaseMovi... FetchMovie.java CheckEven.java AcceptNoRedi... Acceptname.html AcceptName.java AnotherServl... practb.java
1 package mypack;
Problems Javadoc Declaration Console
<terminated> practg [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 2:09:14 PM – 2:09:15 PM) [pid: 10416]
Temprature: 12.6degrees celsius, City: Delhi
Temprature: 25.6degrees celsius, City: Banglore
Temprature: 34.2degrees celsius, City: Kurla
Temprature: 39.7degrees celsius, City: Mumbai
mypack.practg.java - Practical6b/src
Type here to search
BSE midcap +0.18%
ENG IN
2:09 PM 14-Dec-24

PRACTICAL-7

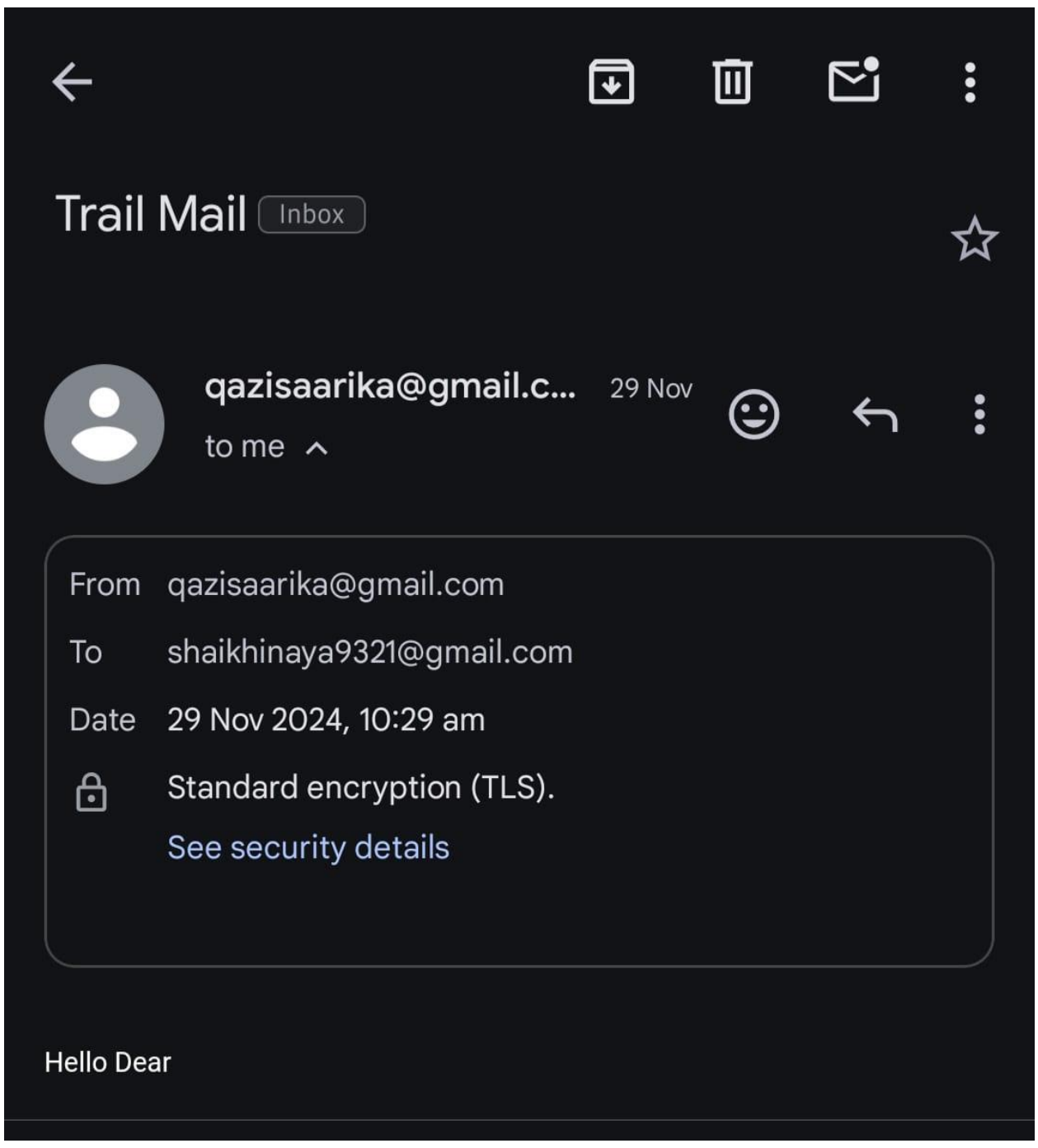
Java Email:-1.Send a text email using Java Code.

Code:-App.java

```
package com.sendmail;

import java.util.Properties;

import java.io.File;

import javax.mail.Message;

import javax.mail.PasswordAuthentication;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeBodyPart;

import javax.mail.internet.MimeMessage;

import javax.mail.internet.MimeMultipart;

public class App

{

        private static final String host = "smtp.gmail.com";

        private static final String username = "qazisaarika@gmail.com";

        private static final String password = "yaxs tnbu bbsy eniq";

        private static final int port = 587;

        private static final String to = "shaikhinaya9321@gmail.com";


    public static void main( String[] args )

    {

        String msg = "Hello Dear";

        String submsg = "Trail Mail";

        //sendMail(msg, submsg, to, username);

    }

private static void sendMail(String message, String subject, String to2, String from) {

        // TODO Auto-generated method stub

        Properties properties = new Properties();

        properties.put("mail.smtp.host", host);

        properties.put("mail.smtp.port", port);

        properties.put("mail.smtp.auth", "true");

        properties.put("mail.smtp.starttls.enable","true");


        Session session = Session.getInstance(properties, new javax.mail.Authenticator() {

                protected PasswordAuthentication getPasswordAuthentication() {

                        return new PasswordAuthentication(username, password);
```

```
        }
});
session.setDebug(true);;
MimeMessage m = new MimeMessage(session);
try {
        m.setFrom(new InternetAddress(from));
        m.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
        m.setSubject(subject);
        m.setText(message);
        Transport.send(m);
        System.out.println("Email sent Successfully!");
}catch(Exception e) {
        e.printStackTrace();
    }
}
}
```

OUTPUT



PRACTICAL-7

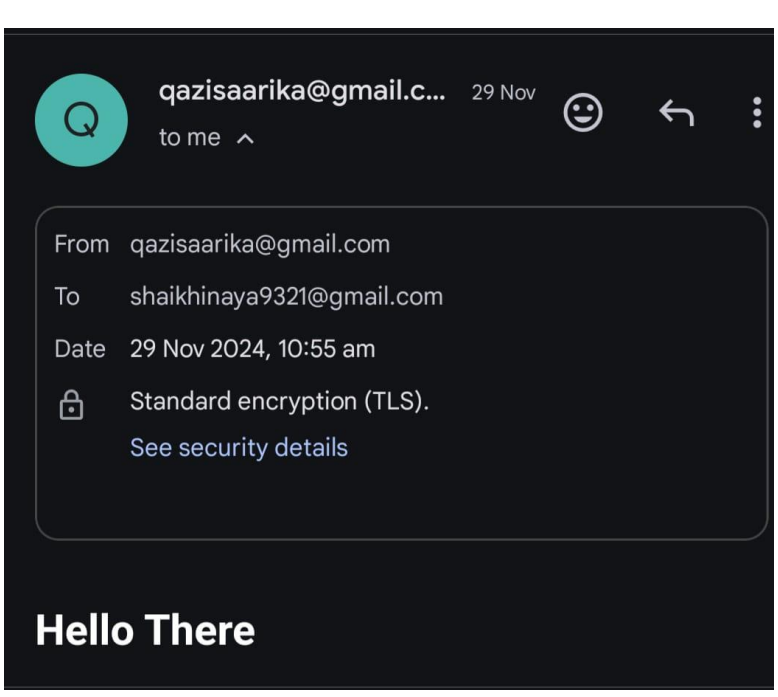
Java Email:-2.Send a HTML Email using Java Code.

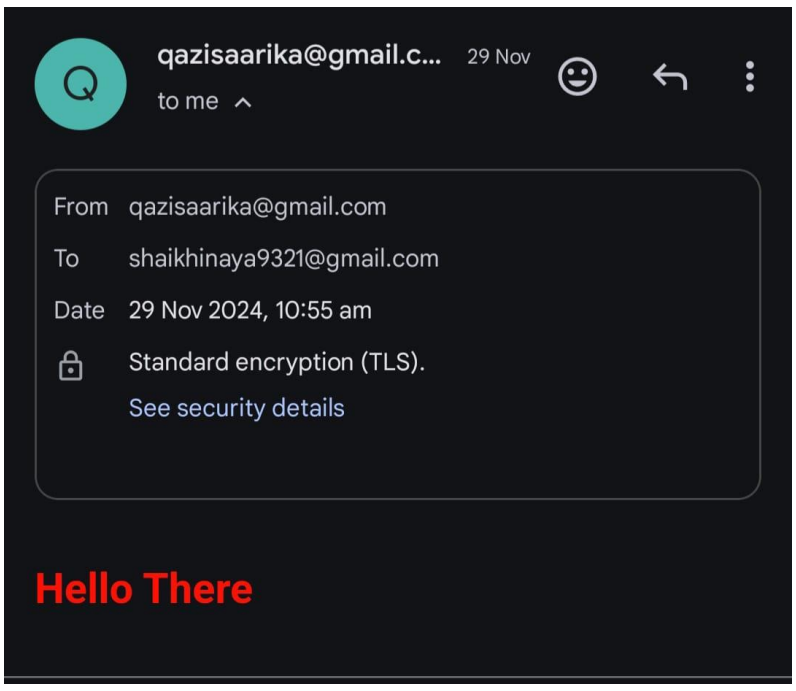
Code:-App.java

```
package com.sendmail;


import java.util.Properties;
import java.io.File;


import javax.mail.Message;
import javax.mail.PasswordAuthentication;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;


/**
 * Hello world!
 *
 */
public class App
{
        private static final String host = "smtp.gmail.com";
        private static final String username = "qazisaarika@gmail.com";
        private static final String password = "yaxs tnbu bbsy eniq";
        private static final int port = 587;
        private static final String to = "shaikhinaya9321@gmail.com";


    public static void main( String[] args )
    {
        String msg = "Hello Dear";
        String submsg = "Trail Mail";
        sendHTMLMail(msg, submsg, to, username);
    }


private static void sendHTMLMail(String msg, String subject, String to2, String from) {
                // TODO Auto-generated method stub
        Properties properties = new Properties();
```

```
properties.put("mail.smtp.host", host);
properties.put("mail.smtp.port", port);
properties.put("mail.smtp.auth", "true");
properties.put("mail.smtp.starttls.enable","true");

Session session = Session.getInstance(properties, new javax.mail.Authenticator() {
        protected PasswordAuthentication getPasswordAuthentication() {
                return new PasswordAuthentication(username, password);
        }
});
session.setDebug(true);;
MimeMessage m = new MimeMessage(session);
try {
        m.setFrom(new InternetAddress(from));
        m.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
        m.setSubject(subject);
        m.setContent("<h1><font color='red'>Hello There</h1>","text/html");
        Transport.send(m);
        System.out.println("Email sent Successfully!");
    }catch(Exception e) {
        e.printStackTrace();
    }
    }
}
```

OUTPUT





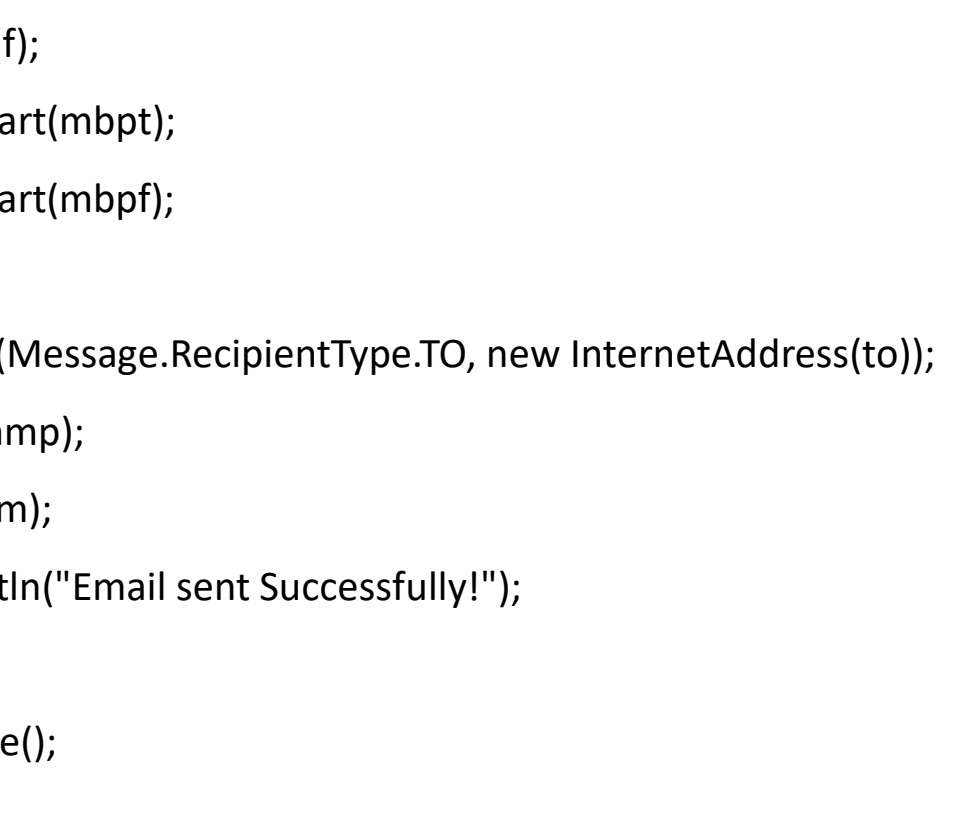
PRACTICAL-7

Java Email:-3. Send an HTML with an attachment using Java Code.

Code:-App.java

```
package com.sendmail;

import java.util.Properties;

import java.io.File;

import javax.mail.Message;

import javax.mail.PasswordAuthentication;

import javax.mail.Session;

import javax.mail.Transport;

import javax.mail.internet.InternetAddress;

import javax.mail.internet.MimeBodyPart;

import javax.mail.internet.MimeMessage;

import javax.mail.internet.MimeMultipart;

public class App

{

        private static final String host = "smtp.gmail.com";

        private static final String username = "qazisaarika@gmail.com";

        private static final String password = "yaxs tnbu bbsy eniq";

        private static final int port = 587;

        private static final String to = "shaikhinaya9321@gmail.com";


    public static void main( String[] args )

    {

        String msg = "Hello Dear";

        String submsg = "Trail Mail";

        sendAttach(msg, submsg, to, username);

    }


private static void sendAttach(String msg, String submsg, String to2, String username2) {

                // TODO Auto-generated method stub

        Properties properties = new Properties();

        properties.put("mail.smtp.host", host);

        properties.put("mail.smtp.port", port);

        properties.put("mail.smtp.auth", "true");

        properties.put("mail.smtp.starttls.enable","true");


        Session session = Session.getInstance(properties, new javax.mail.Authenticator() {

                protected PasswordAuthentication getPasswordAuthentication() {
```

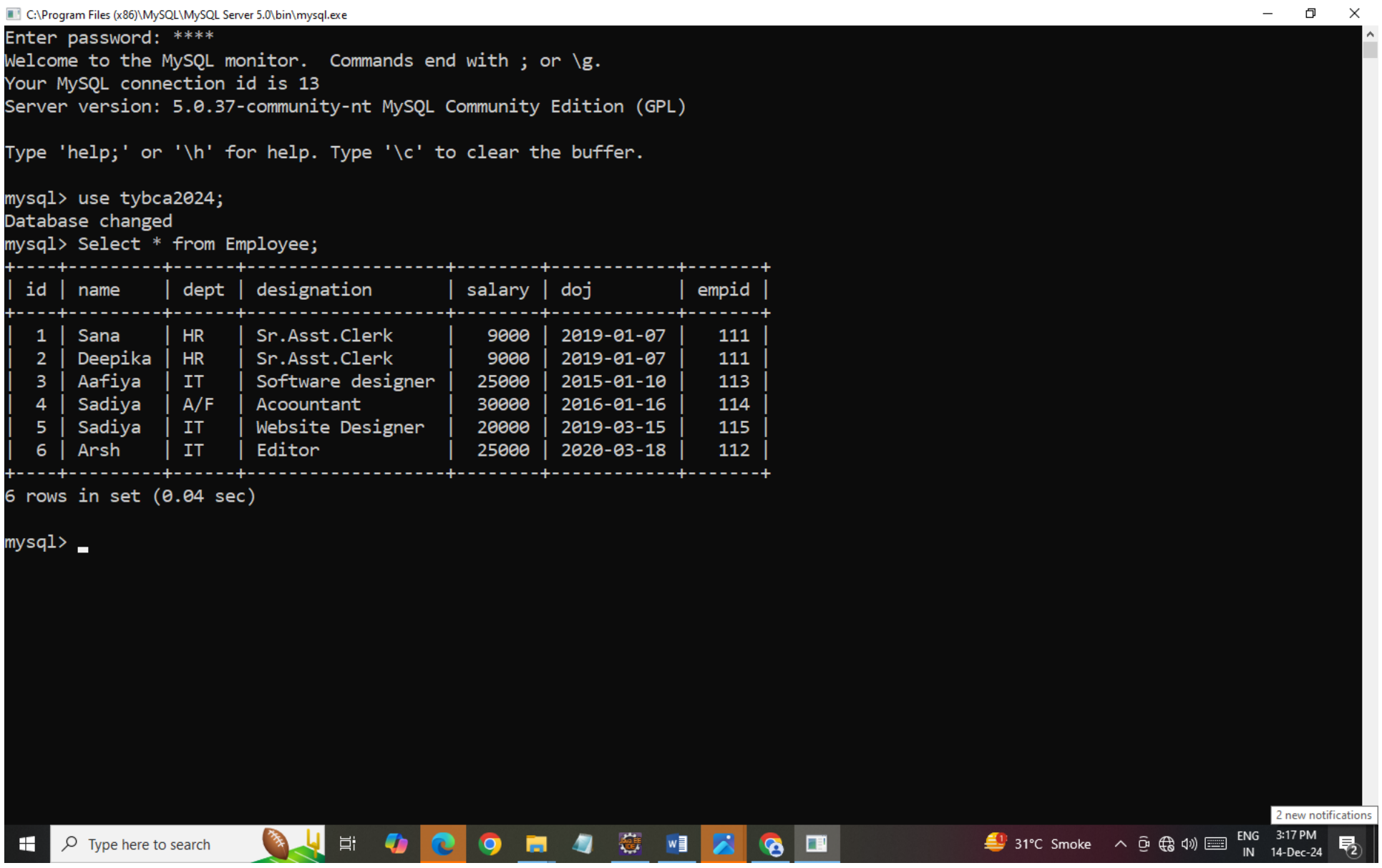
```
                return new PasswordAuthentication(username, password);
        }
});
session.setDebug(true);;
MimeMessage m = new MimeMessage(session);
try {
        String path ="E:\\Sana_wt.html\\";
        MimeMultipart mmp=new MimeMultipart();
        MimeBodyPart mbpt=new MimeBodyPart();
        MimeBodyPart mbpf=new MimeBodyPart();

        mbpt.setText(msg);;
        File f=new File(path);
                mbpf.attachFile(f);
                mmp.addBodyPart(mbpt);
                mmp.addBodyPart(mbpf);

                m.addRecipient(Message.RecipientType.TO, new InternetAddress(to));
                m.setContent(mmp);
                Transport.send(m);
                System.out.println("Email sent Successfully!");
        }catch(Exception e) {
                e.printStackTrace();
        }
}
```



PRACTICAL-8

Hibernate1(1):-Table(Employee:id int Primary key, name varchar(200),dept varchar(50), designation varchar(100), salary float, dateofjoin date)

Table:-



Code:-Employee.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.org/dtd/hibernate-mapping-3.0.dtd">


        <hibernate-mapping>

        <class name="com.mypack.Employee" table="employee">

        <id name="empid">
        <generator class="assigned"></generator>
        </id>

        <property name="name" column="name"></property>
           <property name="dept" column="dept"></property>
           <property name="designation" column="designation"></property>
           <property name="salary" column="salary"></property>
           <property name="doj" column="doj"></property>
        </class>



        </hibernate-mapping>
```

Code:-hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://hibernate.org/dtd/hibernate-configuration-3.0.dtd">

   <hibernate-configuration>

      <session-factory>
           <property name="hbm2ddl.auto">update</property>
           <property name="dialect">org.hibernate.dialect.MySQLDialect</property>
           <property name="connection.url">jdbc:mysql://localhost:3306/tybca2024</property>
           <property name="connection.username">root</property>
```

```
            <property name="connection.password">root</property>
            <property name="show_sql">true</property>
            <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
      <mapping resource="Employee.hbm.xml"/>
      <mapping class="com.mypack.Employee"/>
      </session-factory>
   </hibernate-configuration>
```

Code:-Employee.java

```
package com.mypack;

public class Employee {
      private int empid;
      private String name;
      private String dept;
      private String designation;
      private float salary;
      private String doj;

      @Override
      public String toString() {
            return "Employee [empid=" + empid + ", name=" + name + ", dept=" + dept + ", designation=" +
designation
                        + ", salary=" + salary + ", doj=" + doj + "]";
      }
      public int getEmpid() {
            return empid;
      }
      public void setEmpid(int empid) {
            this.empid = empid;
      }
      public String getName() {
            return name;
      }
      public void setName(String name) {
            this.name = name;
      }
      public String getDept() {
            return dept;
      }
      public void setDept(String dept) {
            this.dept = dept;
      }
      public String getDesignation() {
            return designation;
      }
      public void setDesignation(String designation) {
            this.designation = designation;
      }
      public float getSalary() {
            return salary;
      }
      public void setSalary(float salary) {
            this.salary = salary;
      }
      public String getDoj() {
            return doj;
      }
      public void setDoj(String doj) {
            this.doj = doj;
      }
      public Employee() {
            super();
            // TODO Auto-generated constructor stub
      }
}
```

PRACTICAL-8

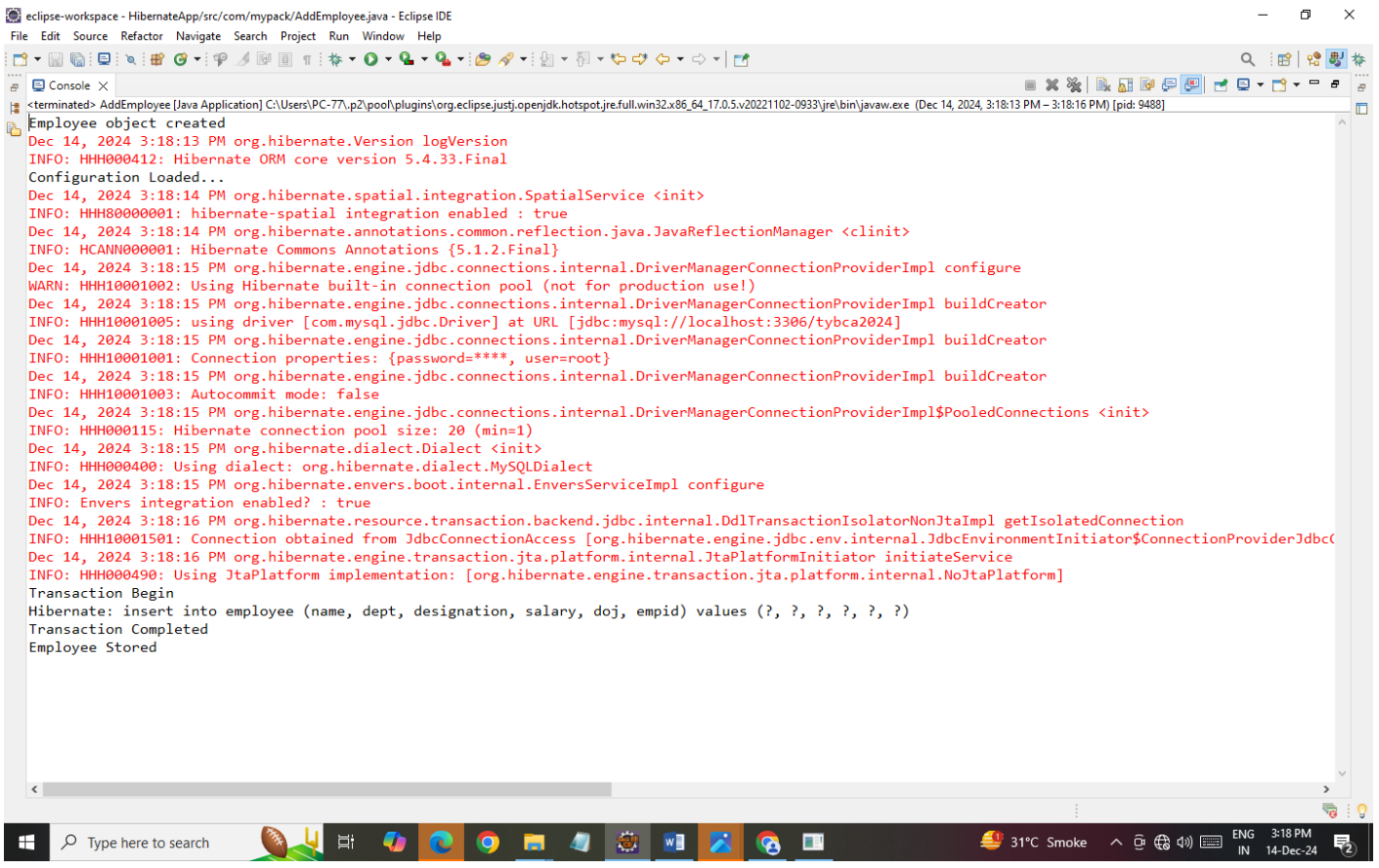
Hibernate1(2):-Write an Hibernate API to add an Employee in given table.

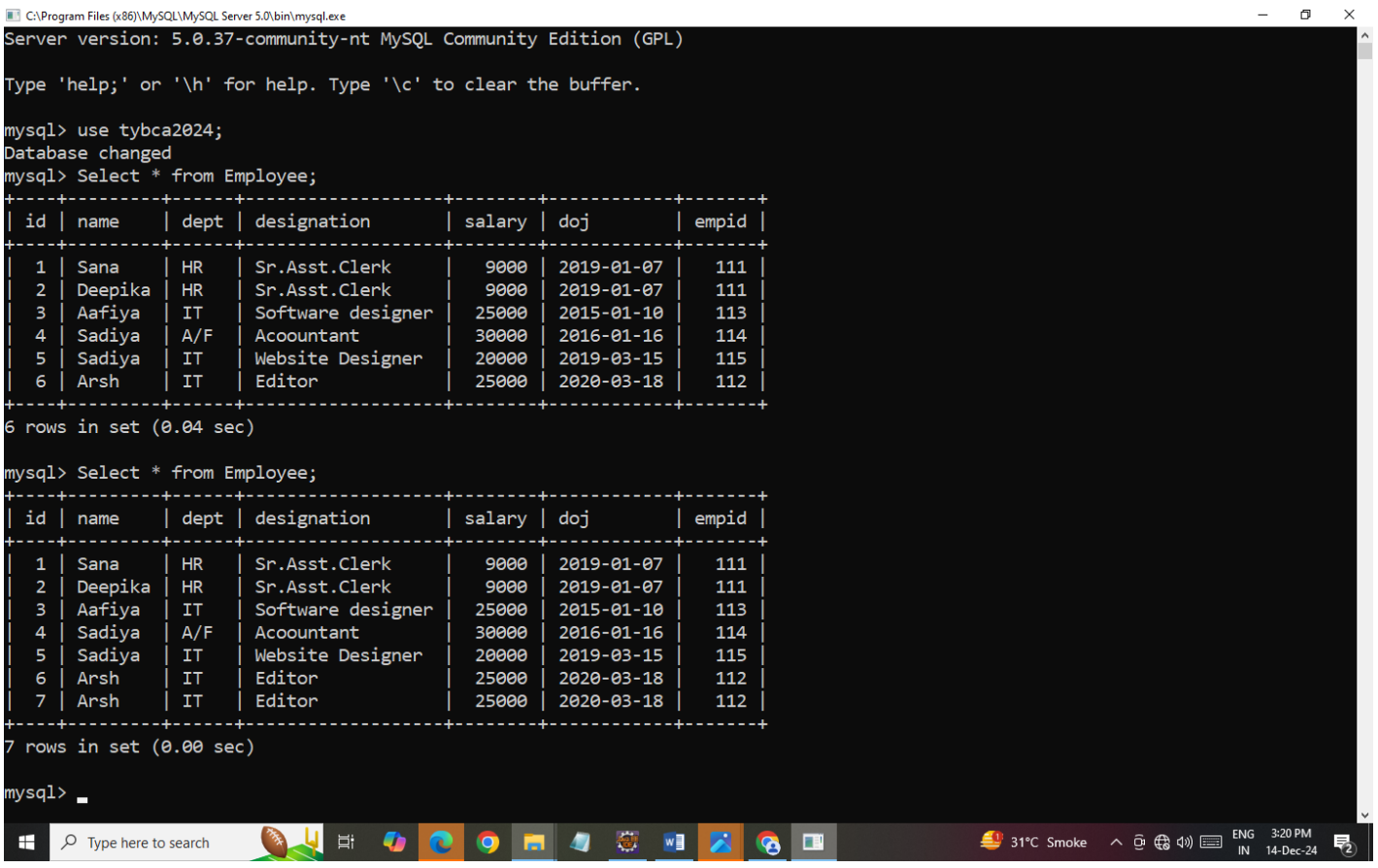
Code:-Addemployee.java

```
package com.mypack;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
public class Addemployee {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		Employee employee=new Employee();
		employee.setEmpid(114);
		employee.setName("Kuman");
		employee.setDept("IT");
		employee.setDesignation("Manager");
		employee.setSalary(200000.00f);
		employee.setDoj("2019-08-20");
		System.out.println("Employee object created");

		try {
			Configuration cfg=new Configuration();
			cfg.configure();

			System.out.println("Configuration Loaded...");

			SessionFactory sf=cfg.buildSessionFactory();
			Session s=sf.openSession();
			Transaction tx=s.beginTransaction();
			System.out.println("Transaction Begin");
			s.save(employee);
			tx.commit();
			System.out.println("transaction completed");
			s.close();
			System.out.println("Employee Stored");
		}
		catch(Exception e)
		{
			e.printStackTrace();
```

```
        }
    }
}
```

OUTPUT


eclipse-workspace - HibernateApp/src/com/mypack/AddEmployee.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
<terminated> AddEmployee [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 3:18:13 PM – 3:18:16 PM) [pid: 9488]
Employee object created
Dec 14, 2024 3:18:13 PM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate ORM core version 5.4.33.Final
Configuration Loaded...
Dec 14, 2024 3:18:14 PM org.hibernate.spatial.integration.SpatialService <init>
INFO: HHH80000001: hibernate-spatial integration enabled : true
Dec 14, 2024 3:18:14 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
Dec 14, 2024 3:18:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Dec 14, 2024 3:18:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/tybca2024]
Dec 14, 2024 3:18:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=root}
Dec 14, 2024 3:18:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
Dec 14, 2024 3:18:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Dec 14, 2024 3:18:15 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Dec 14, 2024 3:18:15 PM org.hibernate.envers.boot.internal.EnversServiceImpl configure
INFO: Envers integration enabled? : true
Dec 14, 2024 3:18:16 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcC
Dec 14, 2024 3:18:16 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Transaction Begin
Hibernate: insert into employee (name, dept, designation, salary, doj, empid) values (?, ?, ?, ?, ?, ?)
Transaction Completed
Employee Stored
Type here to search
31°C Smoke
ENG IN
3:18 PM 14-Dec-24


C:\Program Files (x86)\MySQL\MySQL Server 5.0\bin\mysql.exe
Server version: 5.0.37-community-nt MySQL Community Edition (GPL)
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> use tybca2024;
Database changed
mysql> Select * from Employee;
id	name	dept	designation	salary	doj	empid
1	Sana	HR	Sr.Asst.Clerk	9000	2019-01-07	111
2	Deepika	HR	Sr.Asst.Clerk	9000	2019-01-07	111
3	Aafiya	IT	Software designer	25000	2015-01-10	113
4	Sadiya	A/F	Acoountant	30000	2016-01-16	114
5	Sadiya	IT	Website Designer	20000	2019-03-15	115
6	Arsh	IT	Editor	25000	2020-03-18	112
6 rows in set (0.04 sec)
mysql> Select * from Employee;
id	name	dept	designation	salary	doj	empid
1	Sana	HR	Sr.Asst.Clerk	9000	2019-01-07	111
2	Deepika	HR	Sr.Asst.Clerk	9000	2019-01-07	111
3	Aafiya	IT	Software designer	25000	2015-01-10	113
4	Sadiya	A/F	Acoountant	30000	2016-01-16	114
5	Sadiya	IT	Website Designer	20000	2019-03-15	115
6	Arsh	IT	Editor	25000	2020-03-18	112
7	Arsh	IT	Editor	25000	2020-03-18	112
7 rows in set (0.00 sec)
mysql>
Type here to search
31°C Smoke
ENG IN
3:20 PM 14-Dec-24

PRACTICAL-8

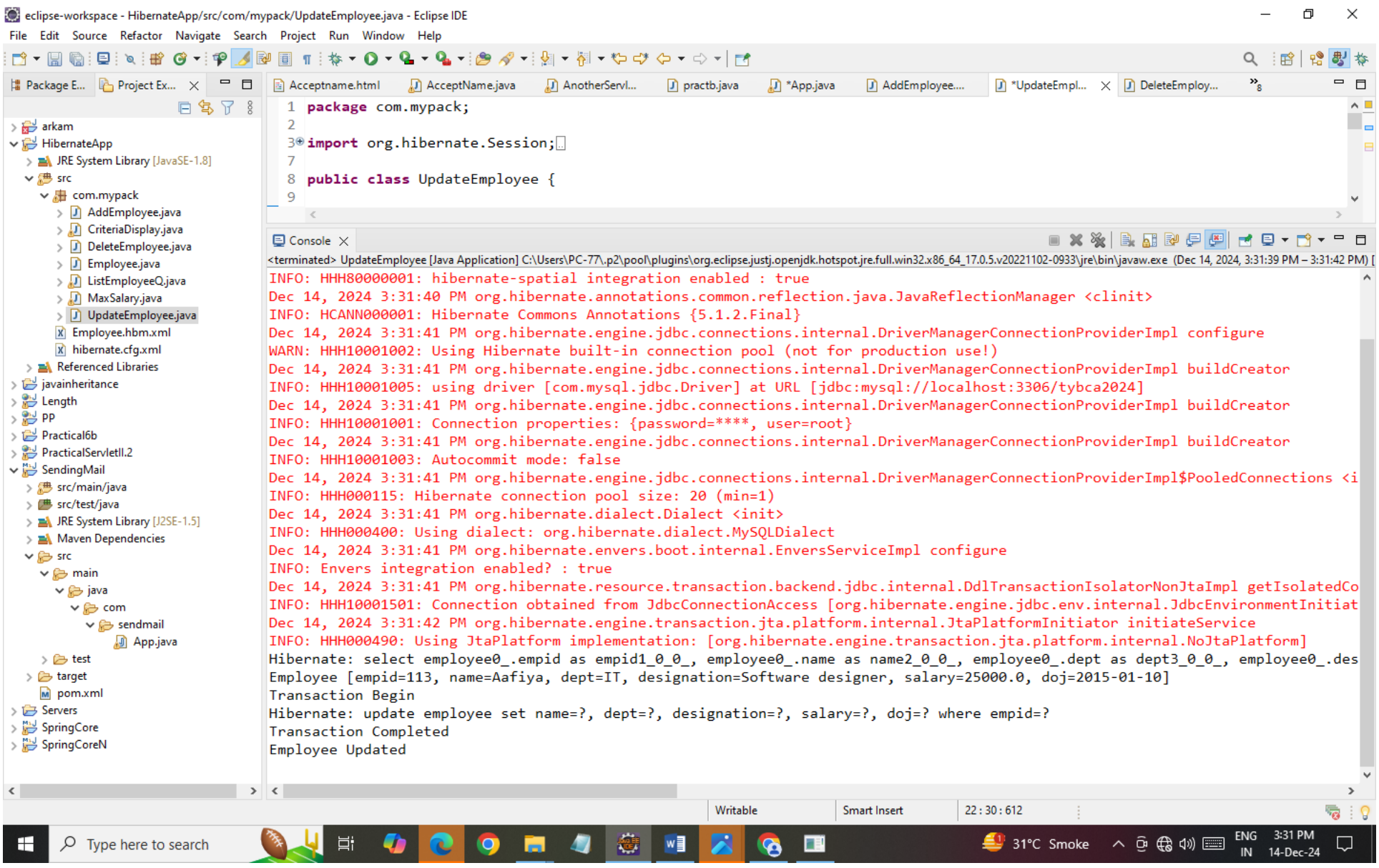
Hibernate-1(3):-Write an Hibernate API to update the name of employee on id 1.

Code:-UpdateEmployee.java

```
package com.mypack;


import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;


public class UpdateEmploye {

	public static void main(String[] args) {
// TODO Auto-generated method stub
try {
		Configuration cfg=new Configuration();
		cfg.configure();

		System.out.println("Configuration Loaded...");

		SessionFactory sf=cfg.buildSessionFactory();
		Session s=sf.openSession();
		Employee e=(Employee)s.get(Employee.class,new Integer(113));
		System.out.println(e);
		e.setName("sumaan");
		s.update(e);

		Transaction tx=s.beginTransaction();
		System.out.println("Transaction Begin");
		//s.save(employee);

		tx.commit();
		System.out.println("transaction completed");
		s.close();
		System.out.println("Employee Updated");
}
catch(Exception e)
{
		e.printStackTrace();
```

```
        }
    }
    }
```

OUTPUT


eclipse-workspace - HibernateApp/src/com/mypack/UpdateEmployee.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E... Project Ex...
arkam
HibernateApp
JRE System Library [JavaSE-1.8]
src
com.mypack
AddEmployee.java
CriteriaDisplay.java
DeleteEmployee.java
Employee.java
ListEmployeeQ.java
MaxSalary.java
UpdateEmployee.java
Employee.hbm.xml
hibernate.cfg.xml
Referenced Libraries
javainheritance
Length
pp
Practical6b
PracticalServletII.2
SendingMail
src/main/java
src/test/java
JRE System Library [J2SE-1.5]
Maven Dependencies
src
main
java
com
sendmail
App.java
test
target
pom.xml
Servers
SpringCore
SpringCoreN
Acceptname.html
AcceptName.java
AnotherServl...
practb.java
*App.java
AddEmployee....
*UpdateEmpl...
DeleteEmploy...
1 package com.mypack;
2
3 import org.hibernate.Session;
7
8 public class UpdateEmployee {
9
Console
<terminated> UpdateEmployee [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 3:31:39 PM – 3:31:42 PM) [
INFO: HHH80000001: hibernate-spatial integration enabled : true
Dec 14, 2024 3:31:40 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
Dec 14, 2024 3:31:41 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Dec 14, 2024 3:31:41 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/tybca2024]
Dec 14, 2024 3:31:41 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=root}
Dec 14, 2024 3:31:41 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
Dec 14, 2024 3:31:41 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <i
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Dec 14, 2024 3:31:41 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Dec 14, 2024 3:31:41 PM org.hibernate.envers.boot.internal.EnversServiceImpl configure
INFO: Envers integration enabled? : true
Dec 14, 2024 3:31:41 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedCo
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiat
Dec 14, 2024 3:31:42 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select employee0_.empid as empid1_0_0_, employee0_.name as name2_0_0_, employee0_.dept as dept3_0_0_, employee0_.des
Employee [empid=113, name=Aafiya, dept=IT, designation=Software designer, salary=25000.0, doj=2015-01-10]
Transaction Begin
Hibernate: update employee set name=?, dept=?, designation=?, salary=?, doj=? where empid=?
Transaction Completed
Employee Updated
Writable
Smart Insert
22 : 30 : 612
Type here to search
31°C Smoke
ENG
IN
3:31 PM
14-Dec-24

PRACTICAL-8

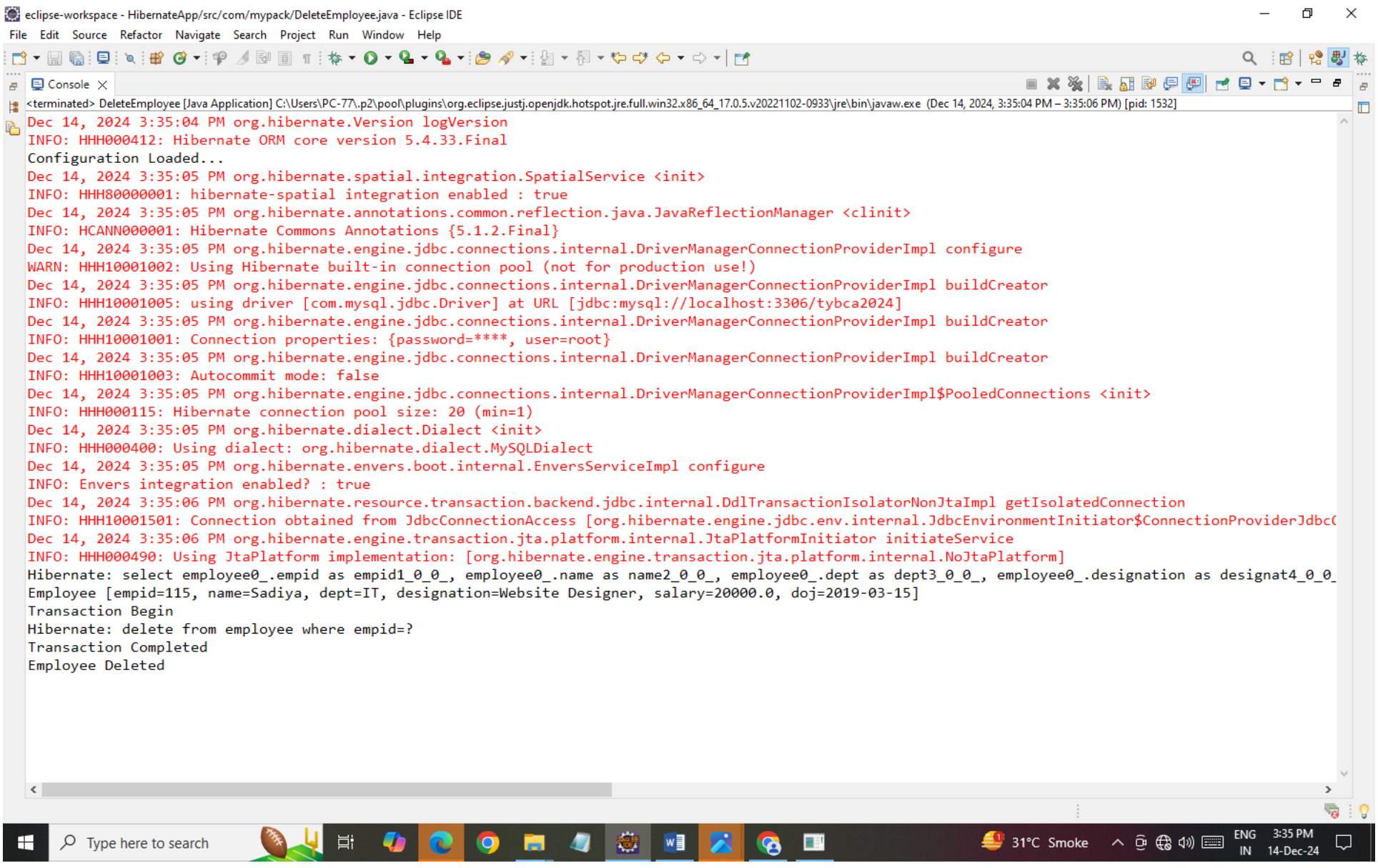
Hibernate-1(4):-Write an Hibernate API to Delete an employee on id 3.

Code:-deleteEmployee.java

```
package com.mypack;


import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;


public class DeletEmployee {

	public static void main(String[] args) {
// TODO Auto-generated method stub
try {
		Configuration cfg=new Configuration();
		cfg.configure();

		System.out.println("Configuration Loaded...");

		SessionFactory sf=cfg.buildSessionFactory();
		Session s=sf.openSession();
		Employee e=(Employee)s.get(Employee.class,new Integer(113));
		System.out.println(e);
		s.delete(e);

		Transaction tx=s.beginTransaction();
		System.out.println("Transaction Begin");
		//s.save(employee);

		tx.commit();
		System.out.println("transaction completed");
		s.close();
		System.out.println("Employee delete");
}
catch(Exception e)
{
		e.printStackTrace();
}
```

```
}

}
```

OUTPUT


eclipse-workspace - HibernateApp/src/com/mypack/DeleteEmployee.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Console
<terminated> DeleteEmployee [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 3:35:04 PM – 3:35:06 PM) [pid: 1532]
Dec 14, 2024 3:35:04 PM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate ORM core version 5.4.33.Final
Configuration Loaded...
Dec 14, 2024 3:35:05 PM org.hibernate.spatial.integration.SpatialService <init>
INFO: HHH80000001: hibernate-spatial integration enabled : true
Dec 14, 2024 3:35:05 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
Dec 14, 2024 3:35:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Dec 14, 2024 3:35:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/tybca2024]
Dec 14, 2024 3:35:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=root}
Dec 14, 2024 3:35:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
Dec 14, 2024 3:35:05 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Dec 14, 2024 3:35:05 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Dec 14, 2024 3:35:05 PM org.hibernate.envers.boot.internal.EnversServiceImpl configure
INFO: Envers integration enabled? : true
Dec 14, 2024 3:35:06 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcC
Dec 14, 2024 3:35:06 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select employee0_.empid as empid1_0_0_, employee0_.name as name2_0_0_, employee0_.dept as dept3_0_0_, employee0_.designation as designat4_0_0_
Employee [empid=115, name=Sadiya, dept=IT, designation=Website Designer, salary=20000.0, doj=2019-03-15]
Transaction Begin
Hibernate: delete from employee where empid=?
Transaction Completed
Employee Deleted
Type here to search
31°C Smoke
ENG
IN
3:35 PM
14-Dec-24

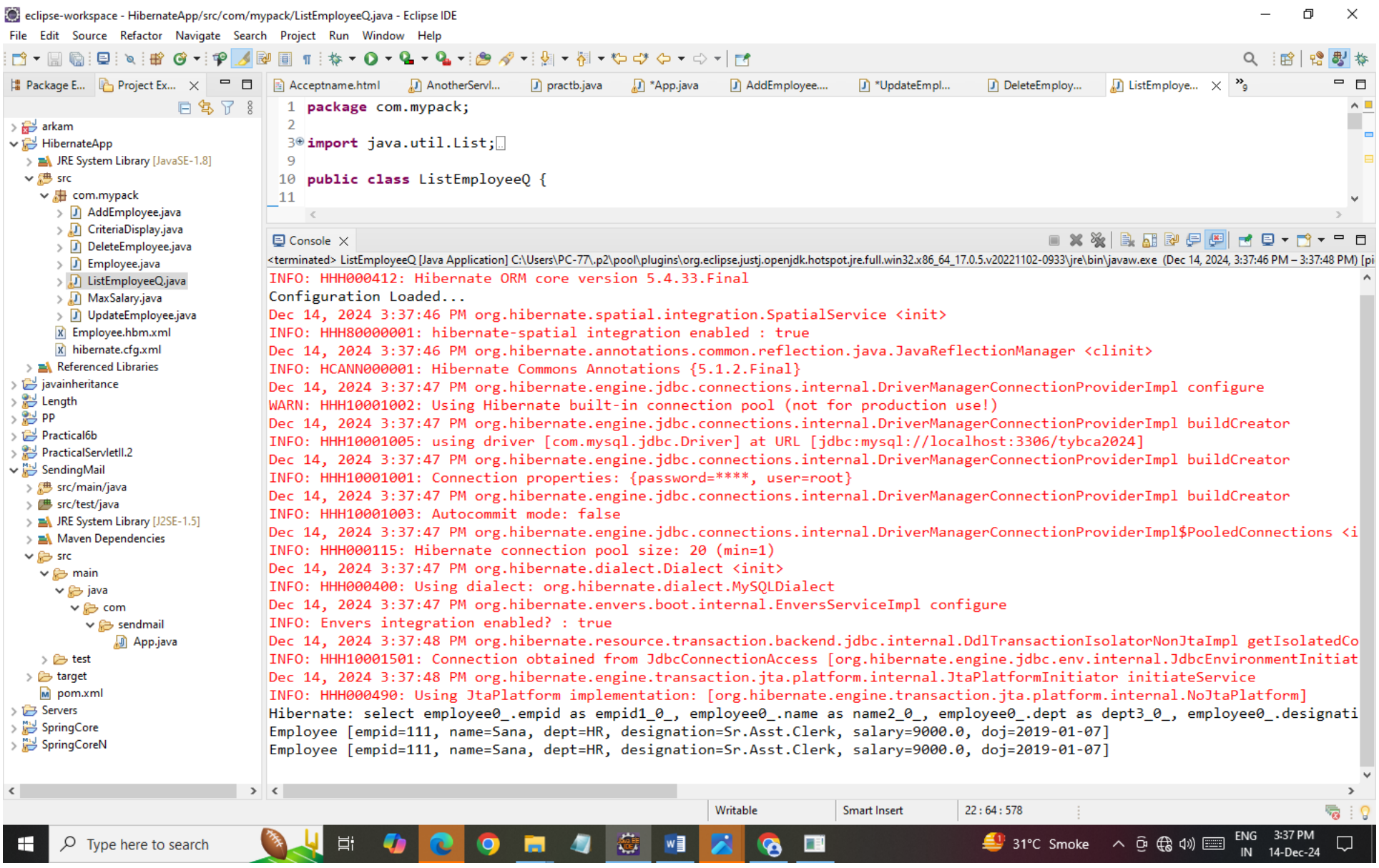
PRACTICAL-9

Hibernate-2(1):-Write an Hibernate API to get all the employee working in IT department using Hibernate Query Lannguage.

Code:-ListEmployeeQ.java

```
package com.mypack;


import java.util.List;


import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import org.hibernate.query.Query;


public class ListEmployeeQ {


	public static void main(String[] args) {
		// TODO Auto-generated method stub
		try {
			Configuration cfg=new Configuration();
			cfg.configure();
			System.out.println("Configuration Loaded...");


			SessionFactory sf=cfg.buildSessionFactory();
			Session s=sf.openSession();
			Query q=s.createQuery("from Employee where dept='HR'");
			List<Employee> l=q.list();
			for (Employee e:l)
				System.out.println(e);
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
	}


}
```

OUTPUT


eclipse-workspace - HibernateApp/src/com/mypack/ListEmployeeQ.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E... Project Ex...
arkam
HibernateApp
JRE System Library [JavaSE-1.8]
src
com.mypack
AddEmployee.java
CriteriaDisplay.java
DeleteEmployee.java
Employee.java
ListEmployeeQ.java
MaxSalary.java
UpdateEmployee.java
Employee.hbm.xml
hibernate.cfg.xml
Referenced Libraries
javainheritance
Length
pp
Practical6b
PracticalServletII.2
SendingMail
src/main/java
src/test/java
JRE System Library [J2SE-1.5]
Maven Dependencies
src
main
java
com
sendmail
App.java
test
target
pom.xml
Servers
SpringCore
SpringCoreN
Acceptname.html
AnotherServl...
practb.java
*App.java
AddEmployee....
*UpdateEmpl...
DeleteEmploy...
ListEmploye...
1 package com.mypack;
2
3 import java.util.List;
9
10 public class ListEmployeeQ {
11
Console
<terminated> ListEmployeeQ [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 3:37:46 PM – 3:37:48 PM) [pi
INFO: HHH000412: Hibernate ORM core version 5.4.33.Final
Configuration Loaded...
Dec 14, 2024 3:37:46 PM org.hibernate.spatial.integration.SpatialService <init>
INFO: HHH80000001: hibernate-spatial integration enabled : true
Dec 14, 2024 3:37:46 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
Dec 14, 2024 3:37:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Dec 14, 2024 3:37:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/tybca2024]
Dec 14, 2024 3:37:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=root}
Dec 14, 2024 3:37:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
Dec 14, 2024 3:37:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <i
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Dec 14, 2024 3:37:47 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Dec 14, 2024 3:37:47 PM org.hibernate.envers.boot.internal.EnversServiceImpl configure
INFO: Envers integration enabled? : true
Dec 14, 2024 3:37:48 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedCo
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiat
Dec 14, 2024 3:37:48 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Hibernate: select employee0_.empid as empid1_0_, employee0_.name as name2_0_, employee0_.dept as dept3_0_, employee0_.designati
Employee [empid=111, name=Sana, dept=HR, designation=Sr.Asst.Clerk, salary=9000.0, doj=2019-01-07]
Employee [empid=111, name=Sana, dept=HR, designation=Sr.Asst.Clerk, salary=9000.0, doj=2019-01-07]
Writable
Smart Insert
22 : 64 : 578
Type here to search
31°C Smoke
ENG
IN
3:37 PM
14-Dec-24

PRACTICAL-9

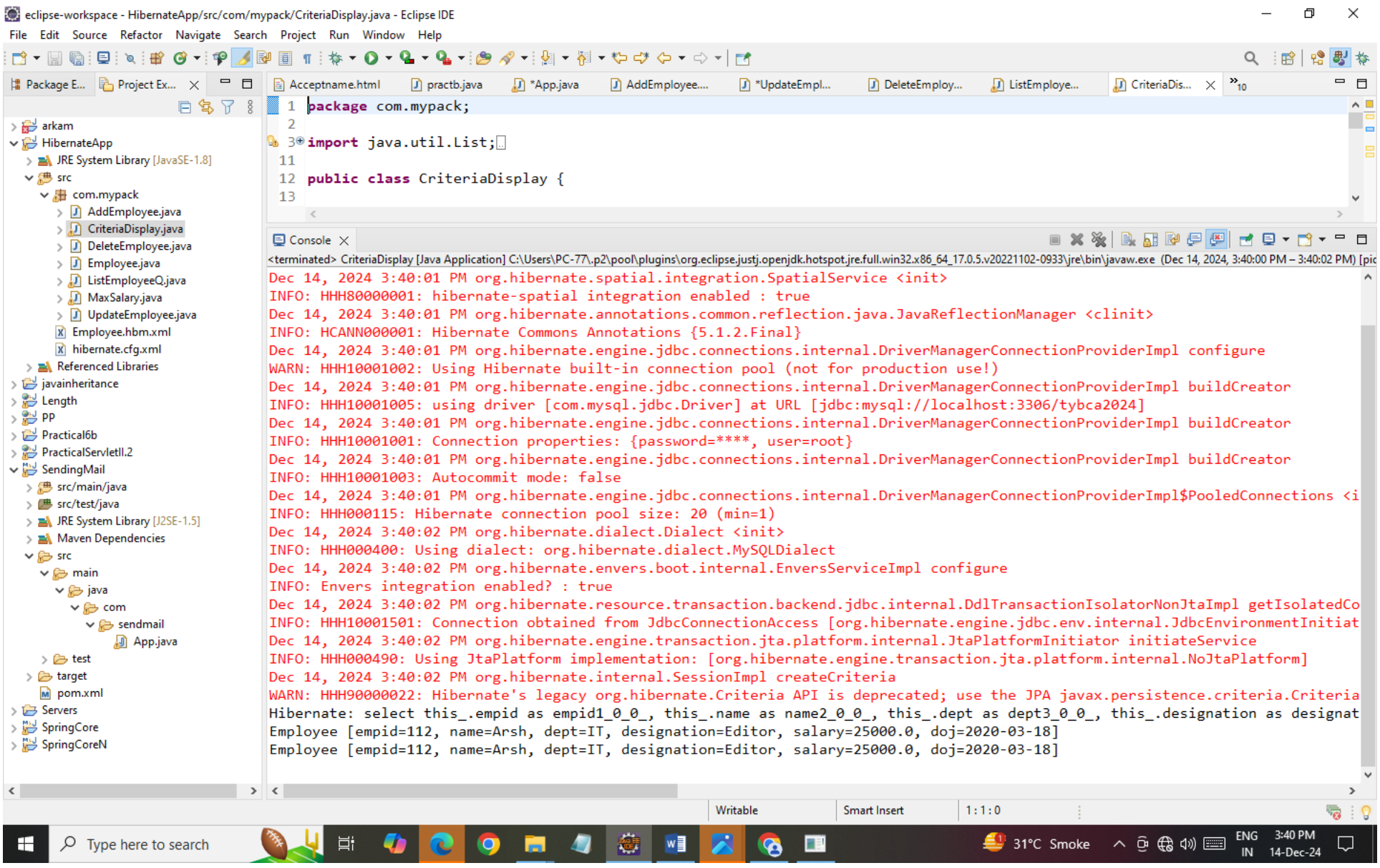
Hibernate-2(2):-Write a Hibernate API to get all the employee who joined after 2016-01-01 and work in HR department using Hibernate Criterion Query Language.

Code:-CriteriaDisplay.java

```
package com.mypack;


import java.util.List;


import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import org.hibernate.criterion.Restrictions;

public class CriteriaDisplaye {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		try {
			Configuration cfg=new Configuration();
			cfg.configure();

			System.out.println("Configuration Loaded...");

			SessionFactory sf=cfg.buildSessionFactory();
			Session s=sf.openSession();
		Criteria c=s.createCriteria(Employee.class);
		c.add(Restrictions.eqOrIsNull("dept", "HR"));
		c.add(Restrictions.gt("doj", "2019-01-07"));
		List<Employee> l1=c.list();
		if(l1.isEmpty())
			System.out.println("No such record");
		else
		{
			for(Employee e:l1)
				System.out.println(e);
		}
}
```

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

OUTPUT



PRACTICAL-9

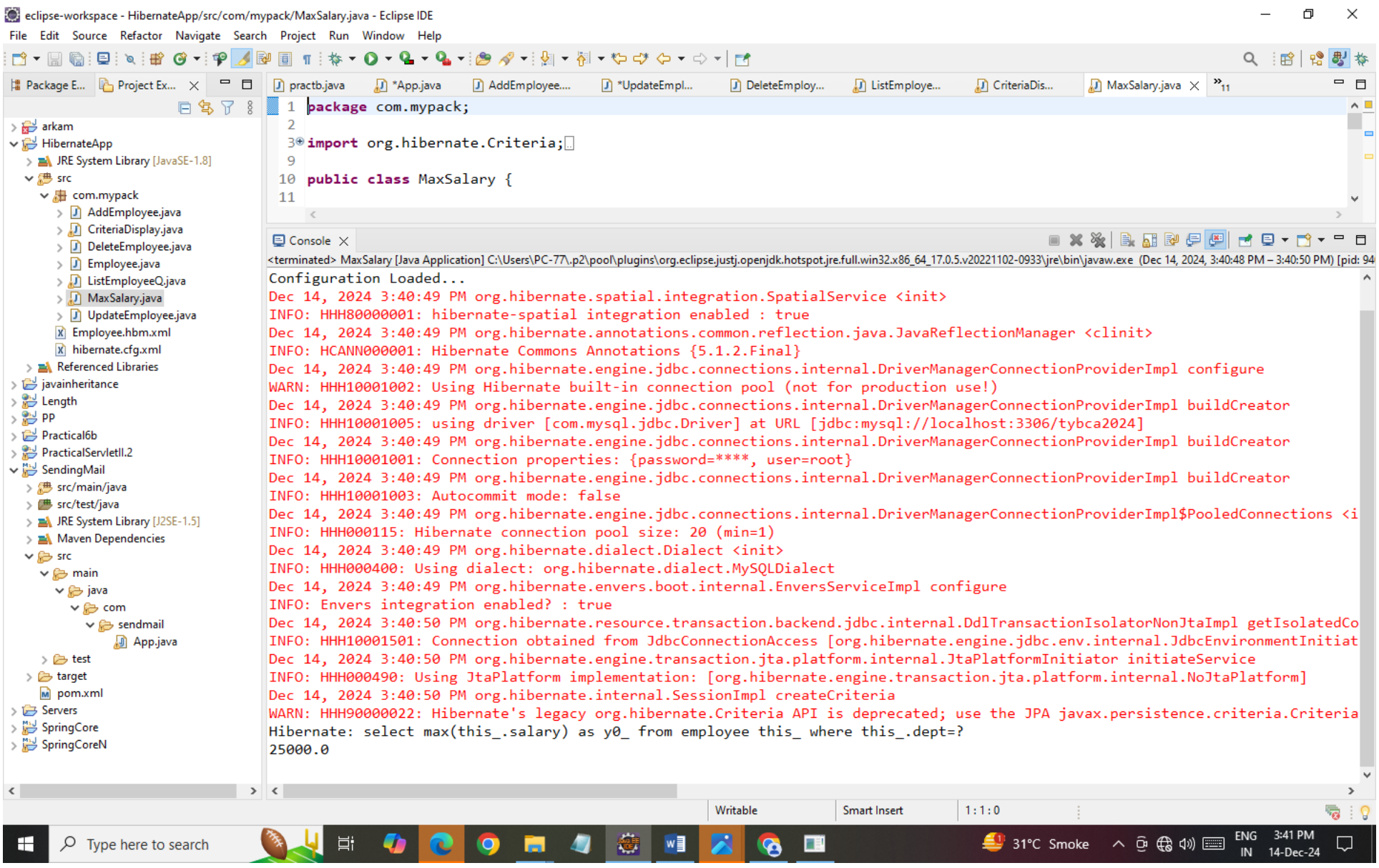
Hibernate-2(3):-Write a Hibernate API to find the maximum salary in IT department using Projection.

Code:-MaxSalary.java

```
package com.mypack;


import java.util.List;


import org.geolatte.geom.crs.Projection;
import org.hibernate.Criteria;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.criterion.Projections;
import org.hibernate.criterion.Restrictions;
import org.hibernate.cfg.Configuration;

public class MaxSalary {

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		try {
			Configuration cfg=new Configuration();
			cfg.configure();

			System.out.println("Configuration Loaded...");

			SessionFactory sf=cfg.buildSessionFactory();
			Session s=sf.openSession();
			Criteria c=s.createCriteria(Employee.class);
			c.add(Restrictions.eqOrIsNull("dept", "HR"));
			c.setProjection(Projections.max("salary"));
			float salary=(float)c.uniqueResult();
			System.out.println(salary);
		}
		catch(Exception e)
		{
			e.printStackTrace();
		}
}
```

}

OUTPUT


eclipse-workspace - HibernateApp/src/com/mypack/MaxSalary.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package E... Project Ex...
arkam
HibernateApp
JRE System Library [JavaSE-1.8]
src
com.mypack
AddEmployee.java
CriteriaDisplay.java
DeleteEmployee.java
Employee.java
ListEmployeeQ.java
MaxSalary.java
UpdateEmployee.java
Employee.hbm.xml
hibernate.cfg.xml
Referenced Libraries
javainheritance
Length
pp
Practical6b
PracticalServletII.2
SendingMail
src/main/java
src/test/java
JRE System Library [J2SE-1.5]
Maven Dependencies
src
main
java
com
sendmail
App.java
test
target
pom.xml
Servers
SpringCore
SpringCoreN
practb.java
*App.java
AddEmployee....
*UpdateEmpl...
DeleteEmploy...
ListEmploye...
CriteriaDis...
MaxSalary.java
1 package com.mypack;
2
3 import org.hibernate.Criteria;
9
10 public class MaxSalary {
11
Console
<terminated> MaxSalary [Java Application] C:\Users\PC-77\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.5.v20221102-0933\jre\bin\javaw.exe (Dec 14, 2024, 3:40:48 PM – 3:40:50 PM) [pid: 94
Configuration Loaded...
Dec 14, 2024 3:40:49 PM org.hibernate.spatial.integration.SpatialService <init>
INFO: HHH80000001: hibernate-spatial integration enabled : true
Dec 14, 2024 3:40:49 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
Dec 14, 2024 3:40:49 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
Dec 14, 2024 3:40:49 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/tybca2024]
Dec 14, 2024 3:40:49 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=root}
Dec 14, 2024 3:40:49 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
Dec 14, 2024 3:40:49 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <i
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Dec 14, 2024 3:40:49 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Dec 14, 2024 3:40:49 PM org.hibernate.envers.boot.internal.EnversServiceImpl configure
INFO: Envers integration enabled? : true
Dec 14, 2024 3:40:50 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedCo
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiat
Dec 14, 2024 3:40:50 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
Dec 14, 2024 3:40:50 PM org.hibernate.internal.SessionImpl createCriteria
WARN: HHH90000022: Hibernate's legacy org.hibernate.Criteria API is deprecated; use the JPA javax.persistence.criteria.Criteria
Hibernate: select max(this_.salary) as y0_ from employee this_ where this_.dept=?
25000.0
Writable
Smart Insert
1 : 1 : 0
Type here to search
31°C Smoke
ENG
IN
3:41 PM
14-Dec-24