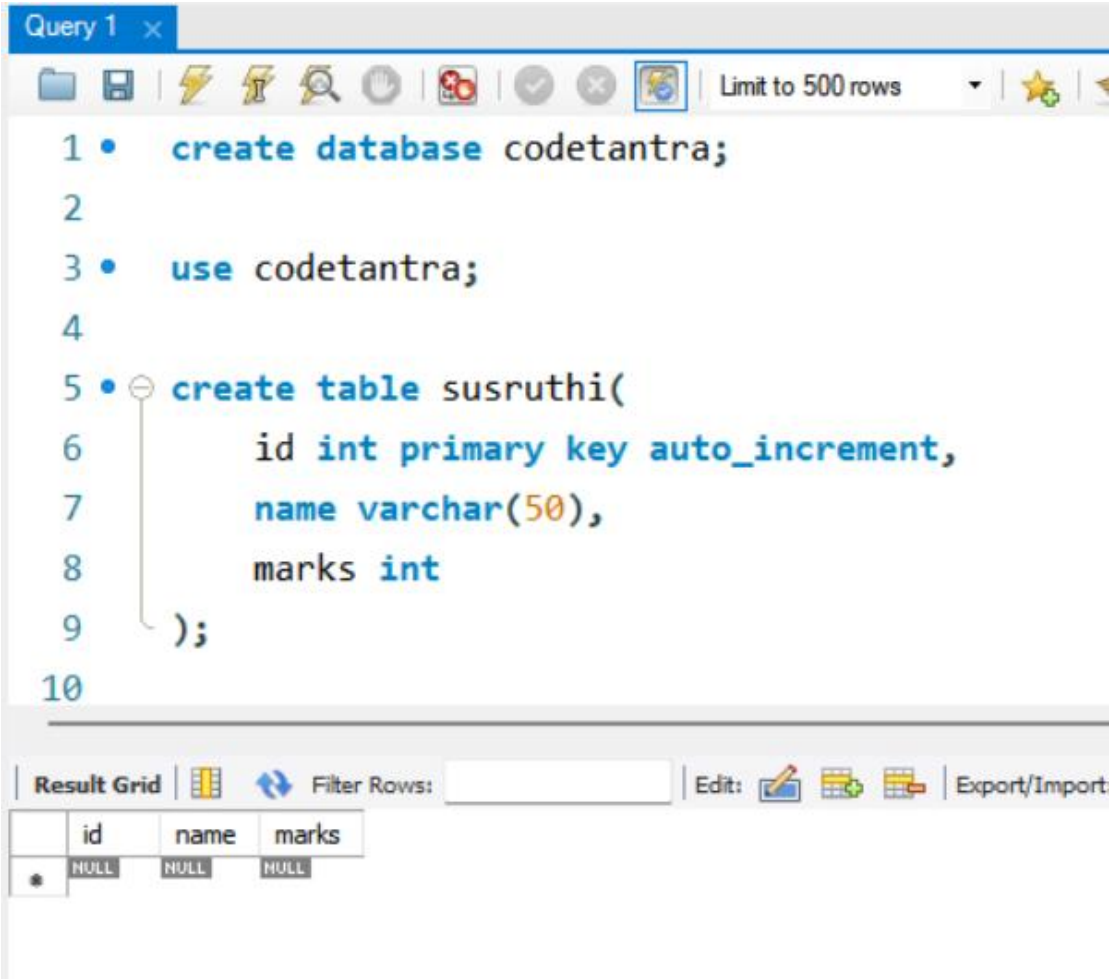
1. Write a program to insert and retrieve the data from database using JDBC.

Before the steps also write what is jdbc components

Steps:

1. Create a table in your database.



2. Open Eclispe (Enterprise Web Developers) and create Java Project(File>New>JavaProject) and create Class.

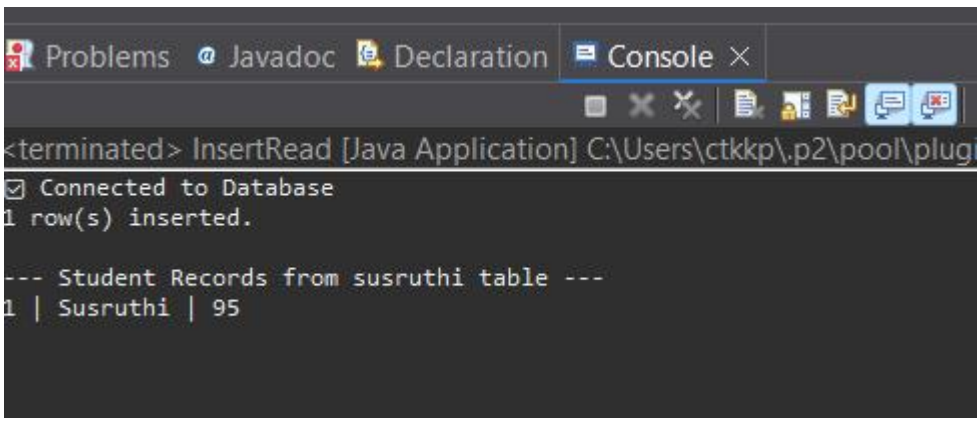
3. After creating the Class, Configure the MySql Connector jar file(buildpath>configure build path>libraries>classpath>AddExternal Jars and add the file)

code :

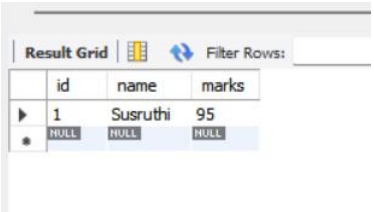
InsertRead.java

```java
package labmanual;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;

public class InsertRead {

    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/codetantra"; // Database URL
        String user = "root"; // MySQL username
        String password = "root"; // MySQL password

        try {
            // 1. Load JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // 2. Connect to database
            Connection con = DriverManager.getConnection(url, user, password);
            System.out.println("☑ Connected to Database");

            // 3. Create Statement
            Statement stmt = con.createStatement();

            // 4. Insert Data into susruthi table
            String name = "Susruthi";
            int marks = 95;
            String insertSQL = "INSERT INTO susruthi (name, marks) VALUES ('" + name + "', " + marks + ")";
            int rowsInserted = stmt.executeUpdate(insertSQL);
            System.out.println(rowsInserted + " row(s) inserted.");

            // 5. Retrieve Data from susruthi table
            String selectSQL = "SELECT * FROM susruthi";
            ResultSet rs = stmt.executeQuery(selectSQL);

            System.out.println("\n--- Student Records from susruthi table ---");
            while (rs.next()) {
                int id = rs.getInt("id");
                String studentName = rs.getString("name");
                int studentMarks = rs.getInt("marks");
                System.out.println(id + " | " + studentName + " | " + studentMarks);
            }

            // 6. Close resources
            rs.close();
            stmt.close();
            con.close();

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

Console Ouput:



In Databse:



Result Grid | Filter Rows:

id	name	marks
1	Susruthi	95
NULL	NULL	NULL

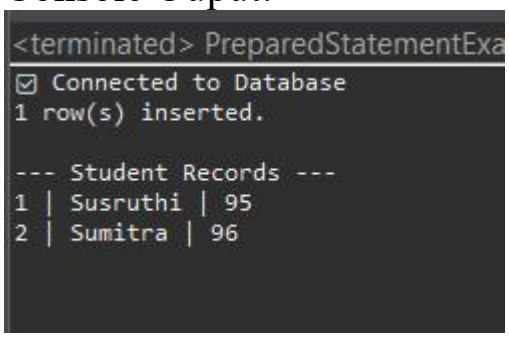
2. **Write a program to demonstrate the use of Prepared Statement and Result Set interface.**

1. Create a Class name in the Same Package(e.g. PreparedStatementExample)

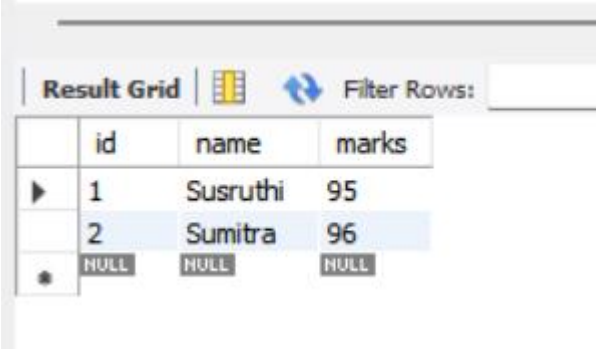
InsertRead.java | PreparedStatementExample.java

```java
package labmanual;

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

public class PreparedStatementExample {

    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/codetantra"; // Database URL
        String user = "root"; // MySQL username
        String password = "root"; // MySQL password


        try {
            // 1. Load JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // 2. Connect to database
            Connection con = DriverManager.getConnection(url, user, password);
            System.out.println("☑ Connected to Database");

            // 3. Insert data using PreparedStatement
            String insertSQL = "INSERT INTO susruthi (name, marks) VALUES (?, ?)";
            PreparedStatement pstmt = con.prepareStatement(insertSQL);
            pstmt.setString(1, "Sumitra");
            pstmt.setInt(2, 96);
            int rowsInserted = pstmt.executeUpdate();
            System.out.println(rowsInserted + " row(s) inserted.");

            // 4. Retrieve data using ResultSet
            String selectSQL = "SELECT * FROM susruthi";
            pstmt = con.prepareStatement(selectSQL);
           ResultSet rs = pstmt.executeQuery();

            System.out.println("\n--- Student Records ---");
            while (rs.next()) {
                int id = rs.getInt("id"); // Column name
                String name = rs.getString("name");
                int marks = rs.getInt("marks");
                System.out.println(id + " | " + name + " | " + marks);
            }

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

Console Ouput:



Database Output:



3. **Servlet Programming Servlet Execution on tomcat A servlet program to print hello world A servlet program to display request details A servlet program to handle user form A servlet program to create a cookie A servlet program to display**
cookie A servlet program to do session tracking Write a program to implement chat Server using Server Socket and Socket
class. Write a Servlet program to send username and password using HTML forms and authenticate the user

1. Create Dynamic web Project(file>new>other>dwp) and give class name
2. Install Apache Tomcat
3. configure servlet api jar file in the project(buildpath>configure buildpath>downloads>apachetomacat>lib>servlet api)
4. Configure apache tomcat server(run on server > choose version of tomcat in Apache> browse the path and select jre version and run) and start the server

1. **Servlet Execution on Tomcat**

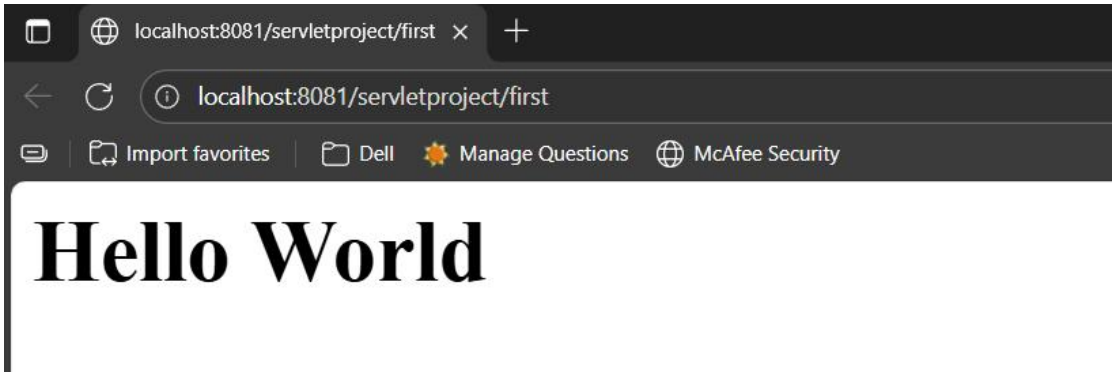
- Install Tomcat (e.g., Apache Tomcat 9.0).
- Add servlet-api.jar to your project's classpath (comes with Tomcat in lib folder).
- Create your Java servlet program inside src folder of a Dynamic Web Project in Eclipse (or manually in Tomcat's WEB-INF/classes).
- Configure web.xml OR use @WebServlet annotation.
- Deploy WAR to Tomcat's webapps folder and start Tomcat.

2. Display Hello World By Running on server.

InsertRead.java | PreparedStatementExample.java | ServletClass.java

```java
package servletproject;

import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@WebServlet("/first")
public class ServletClass extends HttpServlet{
	public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
		response.setContentType("text/html");
		PrintWriter out = response.getWriter();
		out.println("<h1>Hello World</h1>");
	}

}
```

On Default Browser:



3. Servlet to Display Request Details

Code:

```java
@WebServlet("/first")
public class ServletClass extends HttpServlet{
	public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
		response.setContentType("text/html");
		PrintWriter out = response.getWriter();
		out.println("<h1>Hello World</h1>");
		out.println("<h2>Request Details</h2>");
		out.println("Method: " + request.getMethod() + "<br>");
		out.println("URI: " + request.getRequestURI() + "<br>");
		out.println("Protocol: " + request.getProtocol() + "<br>");
		out.println("Remote Address: " + request.getRemoteAddr() + "<br>");
		out.println("User Agent: " + request.getHeader("User-Agent") + "<br>");
	}

}
```

Browser Output:

Request Details

Method: GET
URI: /servletproject/first
Protocol: HTTP/1.1
Remote Address: 0:0:0:0:0:0:0:1
User Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.

4. Servlet to Handle User Form

form.html

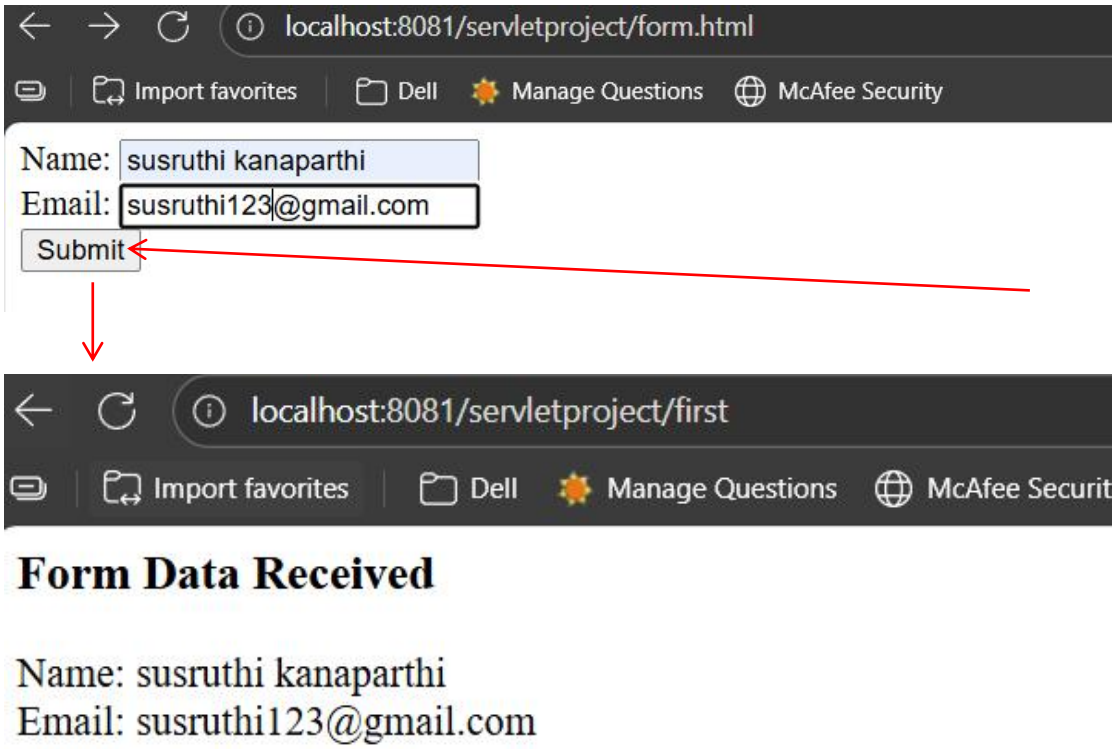
InsertRead.java | PreparedStatementExample.java | ServletClass.java | form.html

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<form action="first" method="post">
	Name: <input type="text" name="name"><br>
	Email: <input type="text" name="email"><br>
	<input type="submit" value="Submit">
</form>

</body>
</html>
```

ServletClass.java

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      String name = request.getParameter("name");
      String email = request.getParameter("email");

      out.println("<h3>Form Data Received</h3>");
      out.println("Name: " + name + "<br>Email: " + email);
  }
```

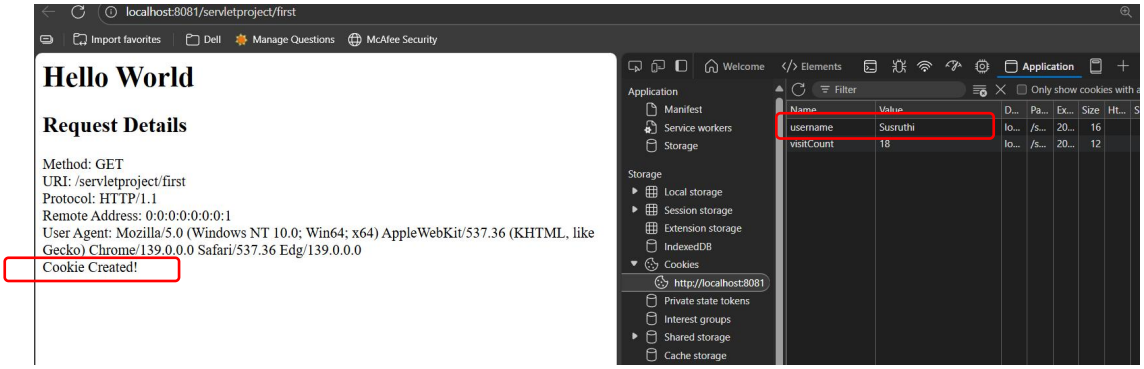
Browser Ouput:



5. Creation of Cookie

```
Cookie cookie = new Cookie("username", "Susruthi");
cookie.setMaxAge(60*60); // 1 hour
response.addCookie(cookie);

response.setContentType("text/html");
response.getWriter().println("Cookie Created!");
```

Output:



6. Display Cookie

```
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    for (Cookie c : cookies) {
        out.println(c.getName() + " = " + c.getValue() + "<br>");
    }
} else {
    out.println("No cookies found.");
}
```

Output:

Cookie Created! visitCount = 18
username = Susruthi

7. Session tracking Servlet

Code:

```
HttpSession session = request.getSession();
Integer count = (Integer) session.getAttribute("visitCount");

if (count == null) count = 0;
count++;
session.setAttribute("visitCount", count);

response.setContentType("text/html");
response.getWriter().println("Visit Count: " + count);
```

Browser Ouput:

JSESSIONID = 4665CFDD8F544D56419C015390105D1E
Visit Count: 2

8. Chat Server using ServerSocket and Socket

ChatServer.java

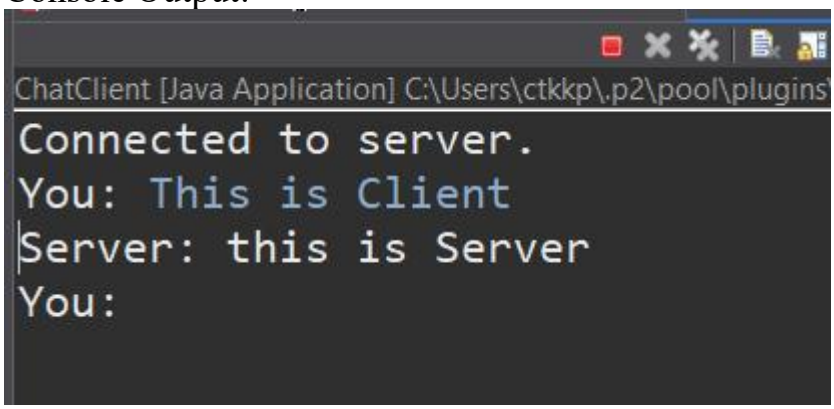
```java
package servletproject;
import java.io.*;
import java.net.*;

public class ChatServer {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(5000);
        System.out.println("Chat Server started on port 5000...");
        Socket socket = serverSocket.accept();
        System.out.println("Client connected.");

        BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

        BufferedReader console = new BufferedReader(new InputStreamReader(System.in));

        String msg;
        while (true) {
            msg = in.readLine();
            if (msg == null || msg.equalsIgnoreCase("bye")) break;
            System.out.println("Client: " + msg);

            System.out.print("You: ");
            out.println(console.readLine());
        }
        socket.close();
        serverSocket.close();
    }
}
```

ChatCleint.java

```java
package servletproject;

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

public class ChatClient {
    public static void main(String[] args) throws IOException {
        Socket socket = new Socket("localhost", 5000);
        System.out.println("Connected to server.");

        BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

        BufferedReader console = new BufferedReader(new InputStreamReader(System.in));

        String msg;
        while (true) {
            System.out.print("You: ");
            out.println(console.readLine());
            msg = in.readLine();
            if (msg == null || msg.equalsIgnoreCase("bye")) break;
            System.out.println("Server: " + msg);
        }
        socket.close();
    }
}
```

Explanation: Run as java Application both(ChatServer and ChatCleint)

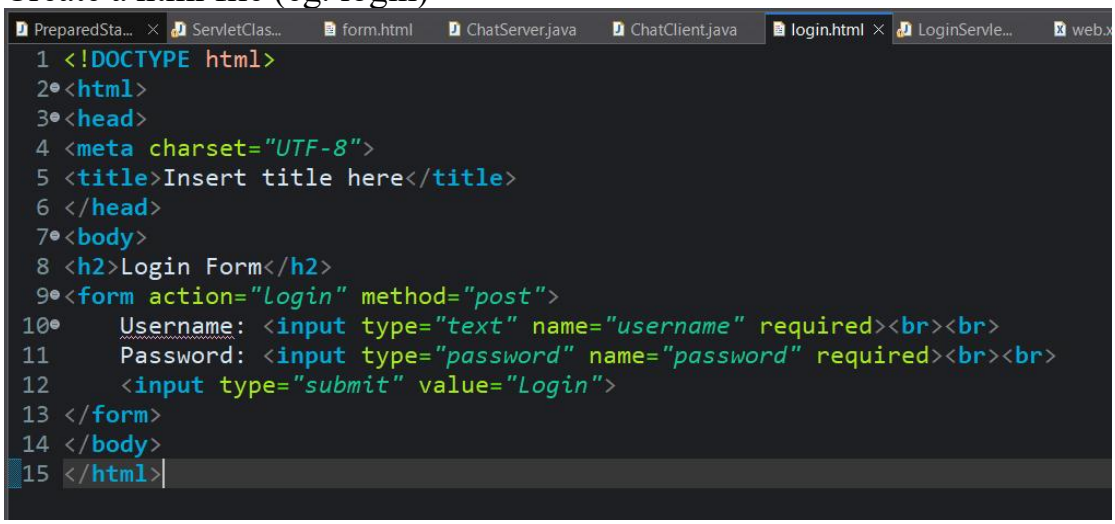
Console Output:



9. Login Validation

login.html

Create a html file (eg. login)



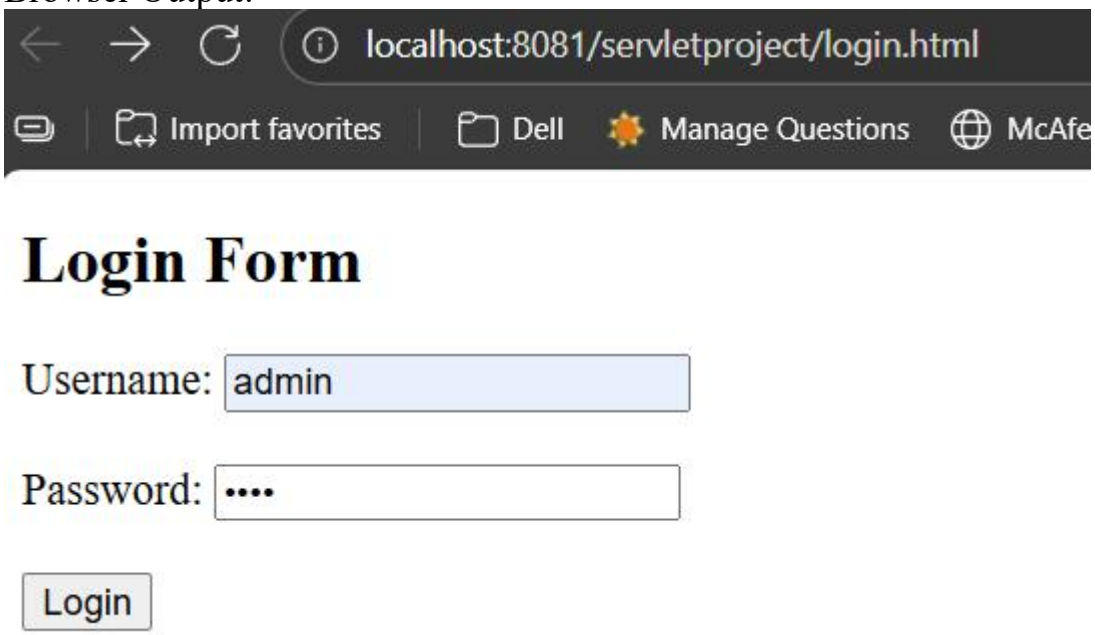
LoginServlet.java

```java
package servletproject;

import java.io.IOException;
import java.io.PrintWriter;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class LoginServlet extends HttpServlet {

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

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

        String user = request.getParameter("username");
        String pass = request.getParameter("password");

        if ("admin".equals(user) && "1234".equals(pass)) {
            out.println("<h3>Login Successful</h3>");
        } else {
            out.println("<h3>Invalid Username or Password</h3>");
        }
    }
}
```

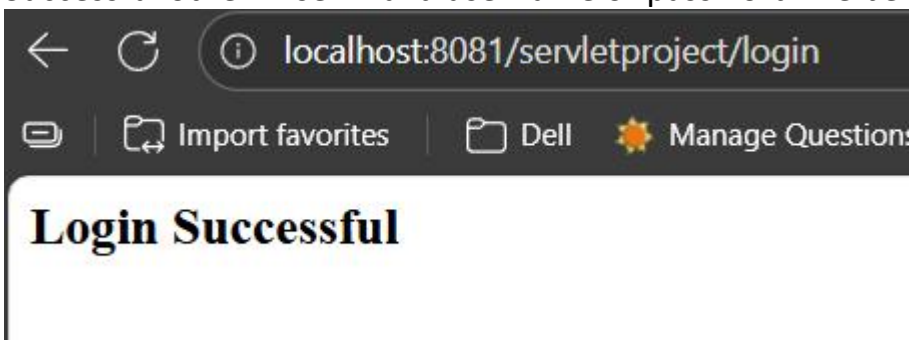
Web.xml(in WEB-INF Folder)

```xml
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
         http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
         version="3.1">
    <servlet>
        <servlet-name>LoginServlet</servlet-name>
        <servlet-class>servletproject.LoginServlet</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>LoginServlet</servlet-name>
        <url-pattern>/login</url-pattern>
    </servlet-mapping>
</web-app>
```

Browser Output:



If user Give username as admin and password as 1234 it gives Login Successful other wise invalid username or password like below.



5.Stop the server.