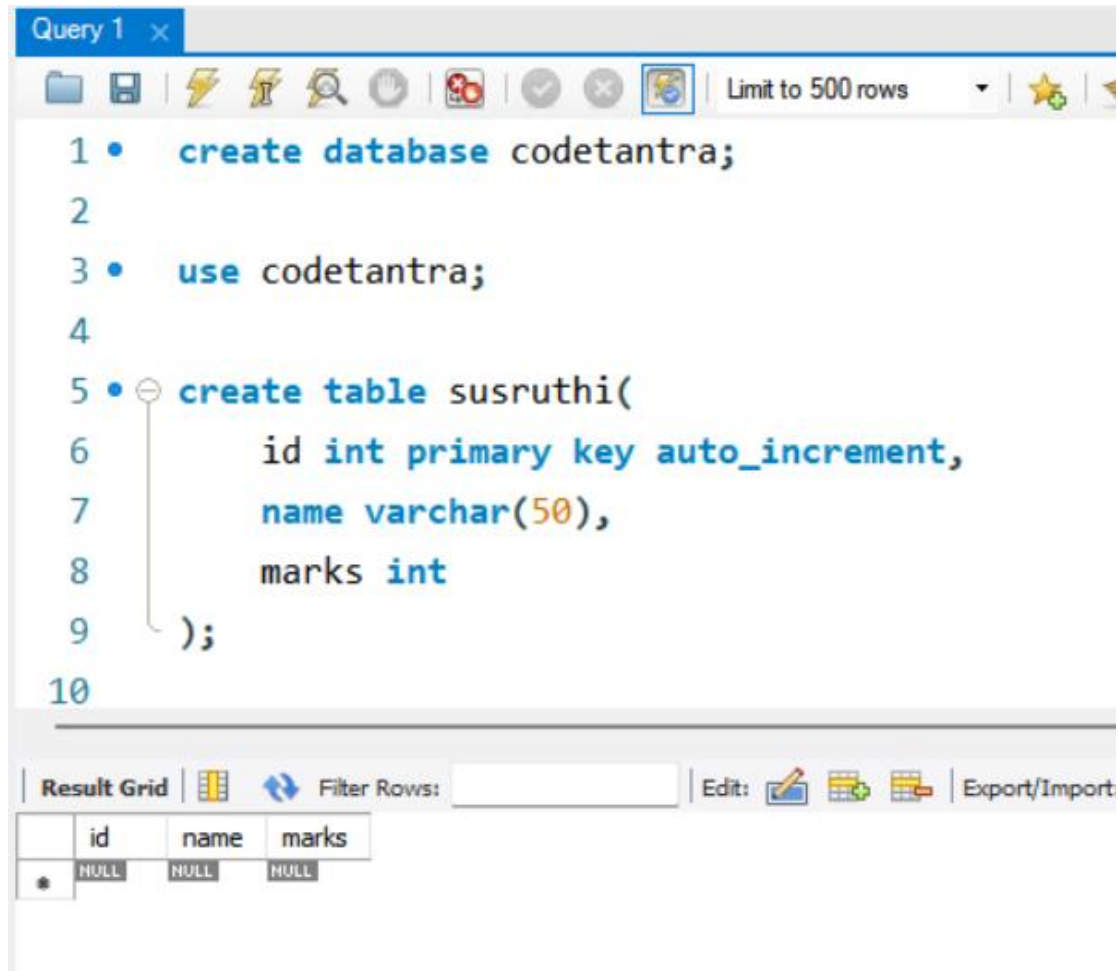


1. Write a program to insert and retrieve the data from database using JDBC.

Before the steps also write what is jdbc components

Steps:

1. Create a table in your database.



The screenshot shows a database query editor window titled "Query 1". The SQL code entered is as follows:

```
1 • create database codetantra;
2
3 • use codetantra;
4
5 • create table susruthi(
6     id int primary key auto_increment,
7     name varchar(50),
8     marks int
9 );
10
```

Below the query editor, there is a "Result Grid" section. It shows a table with three columns: "id", "name", and "marks". The first row of data shows "NULL", "NULL", and "NULL".

	id	name	marks
*	NULL	NULL	NULL

2. Open Eclipse (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 :

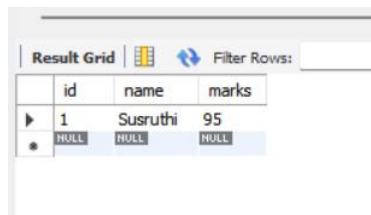
```
1 InsertRead.java X
2
3 package labmanual;
4
5 import java.sql.Connection;
6 import java.sql.DriverManager;
7 import java.sql.ResultSet;
8 import java.sql.Statement;
9
10 public class InsertRead {
11
12     public static void main(String[] args) {
13         String url = "jdbc:mysql://localhost:3306/codetantra"; // Database URL
14         String user = "root"; // MySQL username
15         String password = "root"; // MySQL password
16
17         try {
18             // 1. Load JDBC driver
19             Class.forName("com.mysql.cj.jdbc.Driver");
20
21             // 2. Connect to database
22             Connection con = DriverManager.getConnection(url, user, password);
23             System.out.println("Connected to Database");
24
25             // 3. Create Statement
26             Statement stmt = con.createStatement();
27
28             // 4. Insert Data into susruthi table
29             String name = "Susruthi";
30             int marks = 95;
31             String insertSQL = "INSERT INTO susruthi (name, marks) VALUES ('" + name + "', " + marks + ")";
32             int rowsInserted = stmt.executeUpdate(insertSQL);
33             System.out.println(rowsInserted + " row(s) inserted.");
34
35             // 5. Retrieve Data from susruthi table
36             String selectSQL = "SELECT * FROM susruthi";
37             ResultSet rs = stmt.executeQuery(selectSQL);
38
39             System.out.println("\n--- Student Records from susruthi table ---");
40             while (rs.next()) {
41                 int id = rs.getInt("id");
42                 String studentName = rs.getString("name");
43                 int studentMarks = rs.getInt("marks");
44                 System.out.println(id + " | " + studentName + " | " + studentMarks);
45             }
46
47             // 6. Close resources
48             rs.close();
49             stmt.close();
50             con.close();
51
52         } catch (Exception e) {
53             e.printStackTrace();
54         }
55     }
56 }
```

Console Ouput:

```
Problems Javadoc Declaration Console X
<terminated> InsertRead [Java Application] C:\Users\ctkkp\p2\pool\plug
Connected to Database
1 row(s) inserted.

--- Student Records from susruthi table ---
1 | Susruthi | 95
```

In Database:



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)

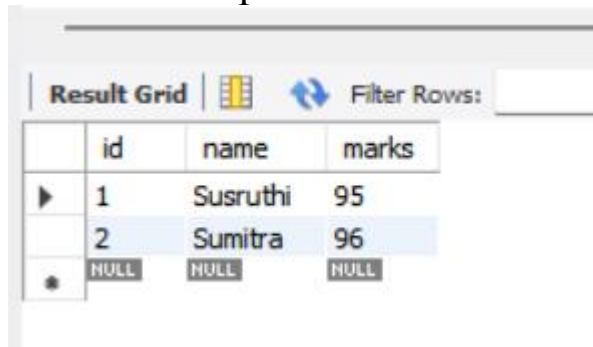
```
InsertRead.java  PreparedStatementExample.java ×
1 package labmanual;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7
8 public class PreparedStatementExample {
9
10     public static void main(String[] args) {
11         String url = "jdbc:mysql://localhost:3306/codetantra"; // Database URL
12         String user = "root"; // MySQL username
13         String password = "root"; // MySQL password
14
15         try {
16             // 1. Load JDBC driver
17             Class.forName("com.mysql.cj.jdbc.Driver");
18
19             // 2. Connect to database
20             Connection con = DriverManager.getConnection(url, user, password);
21             System.out.println("Connected to Database");
22
23             // 3. Insert data using PreparedStatement
24             String insertSQL = "INSERT INTO susruthi (name, marks) VALUES (?, ?)";
25             PreparedStatement pstmt = con.prepareStatement(insertSQL);
26             pstmt.setString(1, "Sumitra");
27             pstmt.setInt(2, 96);
28             int rowsInserted = pstmt.executeUpdate();
29             System.out.println(rowsInserted + " row(s) inserted.");
30
31             // 4. Retrieve data using ResultSet
32             String selectSQL = "SELECT * FROM susruthi";
33             pstmt = con.prepareStatement(selectSQL);
34             ResultSet rs = pstmt.executeQuery();
35
36             System.out.println("\n--- Student Records ---");
37             while (rs.next()) {
38                 int id = rs.getInt("id"); // Column name
39                 String name = rs.getString("name");
40                 int marks = rs.getInt("marks");
41                 System.out.println(id + " | " + name + " | " + marks);
42             }
43
44         } catch (Exception e) {
45             e.printStackTrace();
46         }
47     }
48 }
49
50
```

Console Output:

```
<terminated> PreparedStatementExa
[☒] Connected to Database
1 row(s) inserted.

--- Student Records ---
1 | Susruthi | 95
2 | Sumitra | 96
```

Database Output:



The screenshot shows a database application window with a 'Result Grid' tab. The grid contains three rows of data: a header row with columns 'id', 'name', and 'marks'; a data row with values 1, Susruthi, and 95; and another data row with values 2, Sumitra, and 96. A third row is visible with NULL values. The application also features a 'Filter Rows' button and a search input field.

	id	name	marks
▶	1	Susruthi	95
	2	Sumitra	96
•	NULL	NULL	NULL

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>apachetomcat>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

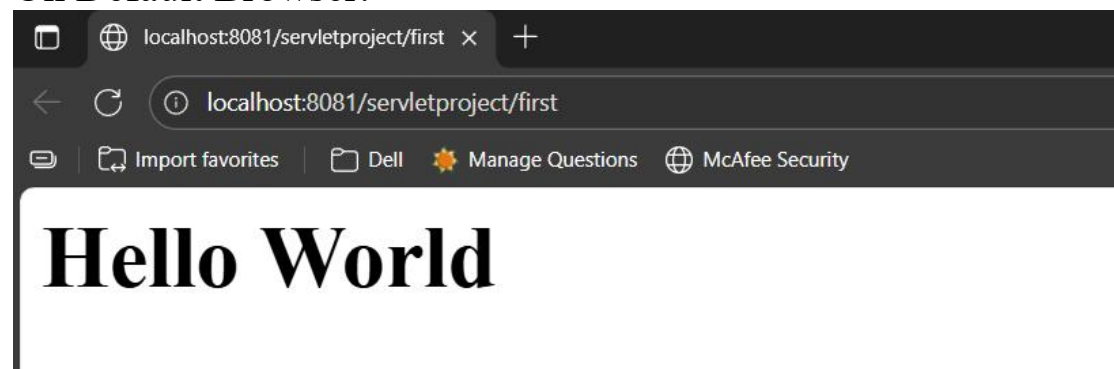
1. Servlet Execution on Tomcat

- 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.

```
1 package servletproject;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import jakarta.servlet.annotation.WebServlet;
7 import jakarta.servlet.http.HttpServlet;
8 import jakarta.servlet.http.HttpServletRequest;
9 import jakarta.servlet.http.HttpServletResponse;
10
11 @WebServlet("/first")
12 public class ServletClass extends HttpServlet{
13     public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
14         response.setContentType("text/html");
15         PrintWriter out = response.getWriter();
16         out.println("<h1>Hello World</h1>");
17     }
18 }
19 }
20 }
```

On Default Browser:



3. Servlet to Display Request Details

Code:


```

@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 x
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 <form action="first" method="post">
9     Name: <input type="text" name="name"><br>
10    Email: <input type="text" name="email"><br>
11    <input type="submit" value="Submit">
12 </form>
13
14 </body>
15 </html>

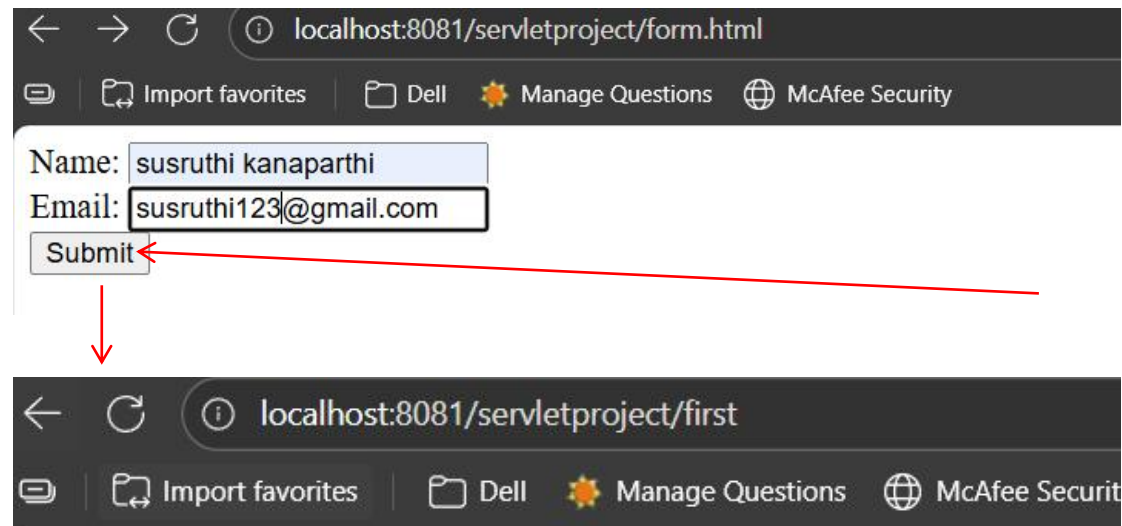
```

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:



Form Data Received

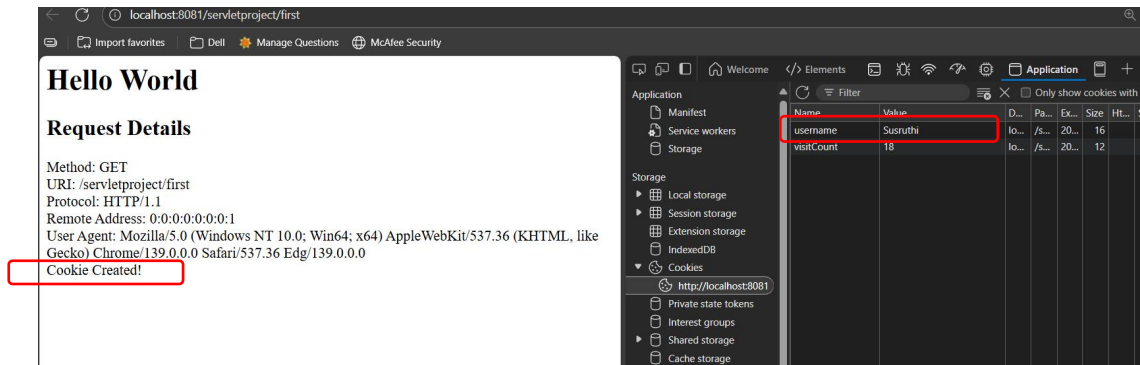
Name: susruthi kanaparthi
Email: susruthi123@gmail.com

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 Output:

JSESSIONID = 4665CFDD8F544D56419C015390105D1E
 Visit Count: 2

8. Chat Server using ServerSocket and Socket

ChatServer.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

```
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:

```
ChatClient [Java Application] C:\Users\cttkp\p2\pool\plugins
Connected to server.
You: This is Client
Server: this is Server
You:
```

9. Login Validation

login.html

Create a html file (eg. login)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 <h2>Login Form</h2>
9 <form action="login" method="post">
10 Username: <input type="text" name="username" required><br><br>
11 Password: <input type="password" name="password" required><br><br>
12 <input type="submit" value="Login">
13 </form>
14 </body>
15 </html>
```

LoginServlet.java

```

1 package servletproject;
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import jakarta.servlet.ServletException;
7 import jakarta.servlet.http.HttpServlet;
8 import jakarta.servlet.http.HttpServletRequest;
9 import jakarta.servlet.http.HttpServletResponse;
10
11 public class LoginServlet extends HttpServlet {
12
13     protected void doPost(HttpServletRequest request, HttpServletResponse response)
14         throws ServletException, IOException {
15
16         response.setContentType("text/html");
17         PrintWriter out = response.getWriter();
18
19         String user = request.getParameter("username");
20         String pass = request.getParameter("password");
21
22         if ("admin".equals(user) && "1234".equals(pass)) {
23             out.println("<h3>Login Successful</h3>");
24         } else {
25             out.println("<h3>Invalid Username or Password</h3>");
26         }
27     }
28 }

```

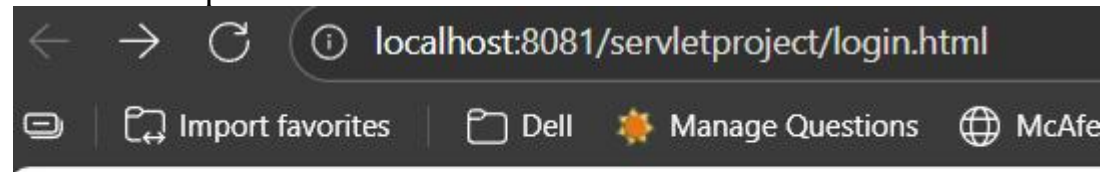
Web.xml(in WEB-INF Folder)

```

1 http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd (xsi:schemaLocation with co
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
5     http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
6     version="3.1">
7     <servlet>
8         <servlet-name>LoginServlet</servlet-name>
9         <servlet-class>servletproject.LoginServlet</servlet-class>
10    </servlet>
11
12    <servlet-mapping>
13        <servlet-name>LoginServlet</servlet-name>
14        <url-pattern>/login</url-pattern>
15    </servlet-mapping>
16 </web-app>
17

```

Browser Output:

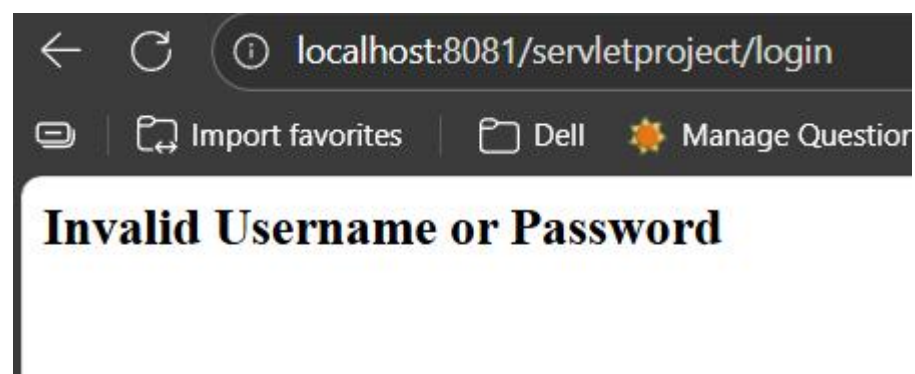
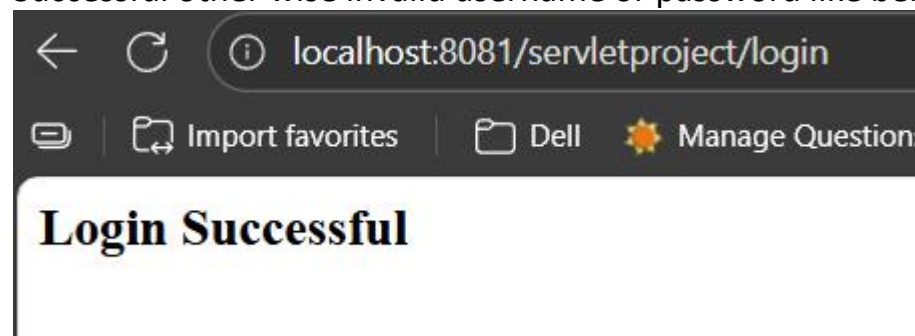


Login Form

Username:

Password:

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.