WEEK-6

AIM : Creating necessary tables for Application choosen using JDBC and establishing the database connectivity.

DESCRIPTION :

1. JDBC stands for Java Database Connectivity, it's an API that allows Java programs to interact with databases.

2. It provides a standard interface for connecting Java applications to database systems.

3. JDBC enables Java applications to execute SQL statements, retrieve results, and manipulate data stored in a database.

4. The JDBC API consists of a set of classes and interfaces in the java.sql package.

5. JDBC drivers act as intermediaries between Java applications and different types of databases.

6. There are four types of JDBC drivers: Type 1, Type 2, Type 3, and Type 4, each with varying degrees of platform dependency and performance.

7. To use JDBC, you need to load the appropriate driver class using Class.forName() method.

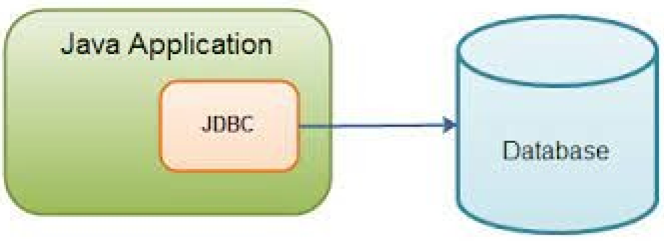
8. Connection interface represents a connection to the database and provides methods for creating statements and managing transactions.

9. Statement interface is used to execute SQL queries and updates against the database.

10. PreparedStatement interface extends Statement and provides precompiled SQL statements, offering better performance and security.

11. ResultSet interface represents the result set of a database query and provides methods for traversing and retrieving data.

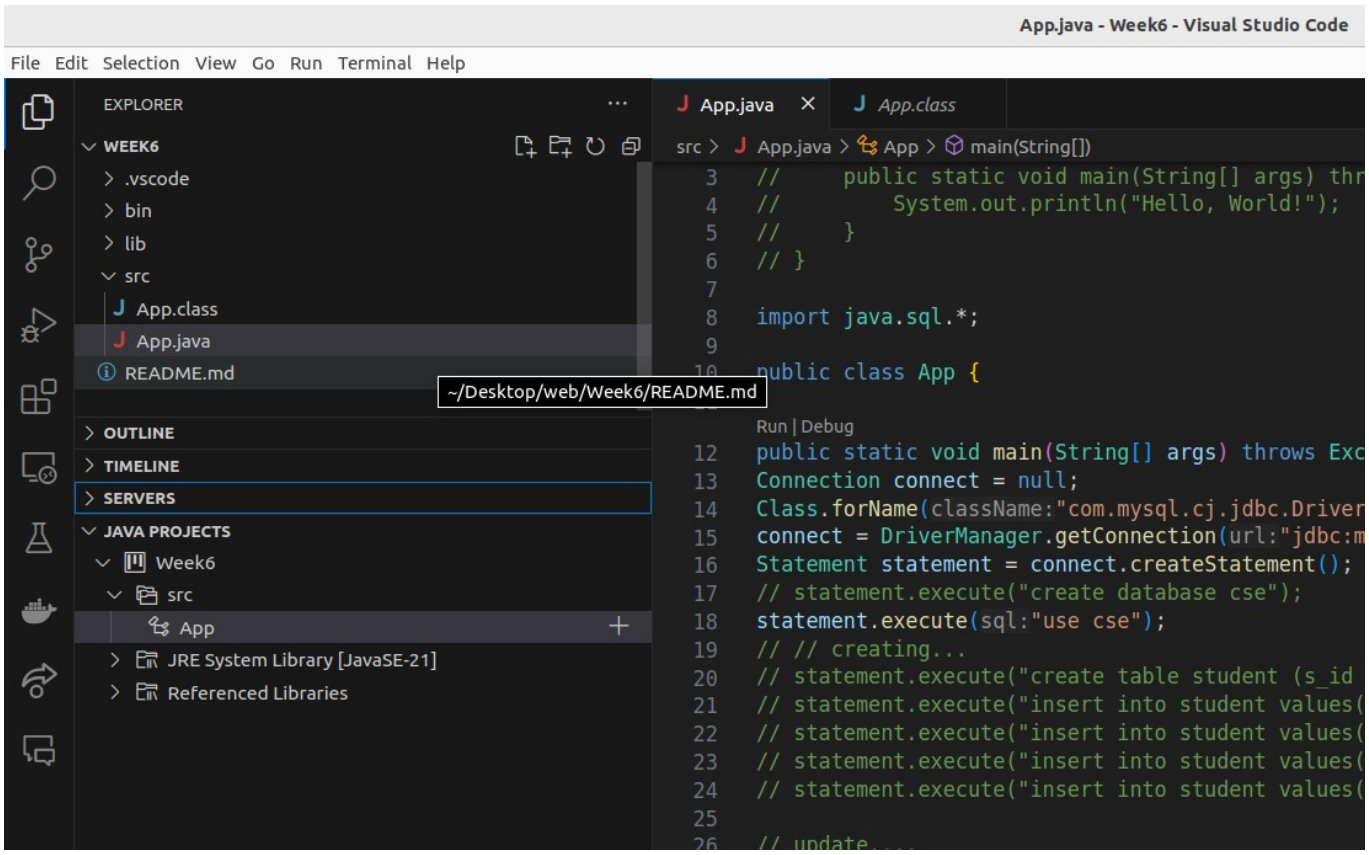
12. DriverManager class manages a list of database drivers and provides methods for establishing database connections.



Program : **App.java File :**

```
import java.sql.*;
public class App {

public static void main(String[] args) throws Exception {
Connection connect = null;
Class.forName("com.mysql.cj.jdbc.Driver");
connect = DriverManager.getConnection("jdbc:mysql://localhost:3306", "root",
"root");
Statement statement = connect.createStatement();
// statement.execute("create database cse");
statement.execute("use cse");

// // creating...
// statement.execute("create table student (s_id integer, s_name varchar(20), s_sec
varchar(5))");
// statement.execute("insert into student values(01,'Satyavanth','3')");
// statement.execute("insert into student values(02,'XYZ','3')");
// statement.execute("insert into student values(03,'ABC', '3')");
// statement.execute("insert into student values(04,'Randon guy', '3')");

// update...
// String sql1 = "update student set s_name='Satyavanth' where s_id=2";
// statement.executeUpdate(sql1);

// delete...
// String sql="delete from student where s_id=2";
// statement.executeUpdate(sql);

ResultSet resultSet = statement.executeQuery("select * from student");
while(resultSet.next()){
System.out.println("Student ID: "+resultSet.getString(1));
System.out.println("Student Name: "+resultSet.getString(2));
System.out.println("Student Section: "+resultSet.getString(3));
System.out.println();
}
resultSet.close();
statement.close();
connect.close();
}
}
```

<u>File Structure</u> :


App.java - Week6 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
WEEK6
.vscode
bin
lib
src
App.class
App.java
README.md
~/Desktop/web/Week6/README.md
OUTLINE
TIMELINE
SERVERS
JAVA PROJECTS
Week6
src
App
JRE System Library [JavaSE-21]
Referenced Libraries
App.java
App.class
src > App.java > App > main(String[])
Run | Debug

```java
//      public static void main(String[] args) thr
//          System.out.println("Hello, World!");
//      }
// }

import java.sql.*;

public class App {
public static void main(String[] args) throws Exc
Connection connect = null;
Class.forName(className:"com.mysql.cj.jdbc.Driver
connect = DriverManager.getConnection(url:"jdbc:m
Statement statement = connect.createStatement();
// statement.execute("create database cse");
statement.execute(sql:"use cse");
// // creating...
// statement.execute("create table student (s_id
// statement.execute("insert into student values(
// statement.execute("insert into student values(
// statement.execute("insert into student values(
// statement.execute("insert into student values(

// update....
```

<u>Output</u> :

```
Student ID: 1
Student Name: Satya
Student Section: 3

Student ID: 3
Student Name: ABC
Student Section: 3

Student ID: 21
Student Name: hello456@#
Student Section: abc
```

WEEK-7 and WEEK-8

AIM : Create the necessary servlets for the application chosen.

Scenario 1: Check the authenticity of the login details with the information available in database. If he is a valid user it must redirect to site resources otherwise it should stay in same page with invalid username/password message.

Scenario 2: Insert the details of the registration page into the database. If registration is successful it must display “Registration is successful”.

Scenario 3: Update the password field in the database.

DESCRIPTION :

1. Servlets are Java classes that extend the capabilities of servers to handle client requests.

2. They are a key component of server-side Java web applications, allowing dynamic content generation and interaction with clients.

3. Servlets run on the server-side and respond to requests from web browsers or other clients.

4. They adhere to the Java Servlet API specification, which defines a standard interface for writing servlets.

5. Servlets are deployed in a servlet container or servlet engine, such as Apache Tomcat or Jetty.

6. Servlets handle various types of requests, including HTTP GET, POST, PUT, DELETE, etc.

7. They can generate dynamic content, such as HTML, XML, JSON, etc., based on the client request and application logic.

8. Servlets often work in conjunction with JavaServer Pages (JSP) for creating dynamic web pages.

9. The HttpServlet class provides a convenient base class for servlets that handle HTTP requests and responses.

10. Servlet lifecycle includes initialization, handling requests, and destruction, managed by the servlet container.

11. Initialization of a servlet occurs when the container loads the servlet class or when the first request is received.

12. Servlets handle requests by overriding methods such as doGet() for handling HTTP GET requests and doPost() for handling POST requests.

13. They can access request parameters, headers, and other information using HttpServletRequest object.

14. Servlets produce responses by writing content to the HttpServletResponse object, including headers and body.

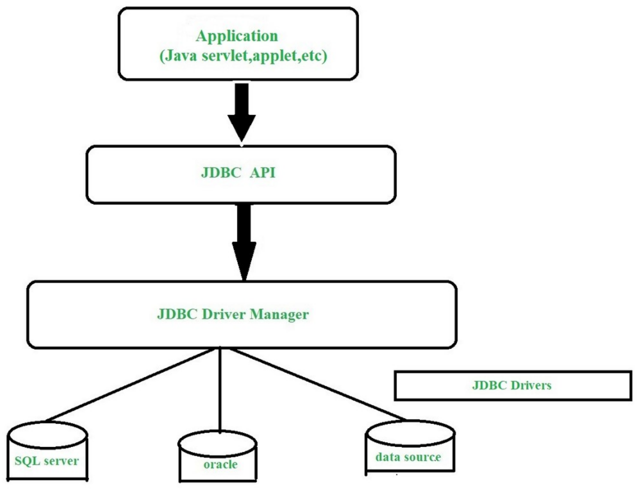
15. Servlets can maintain client sessions using HttpSession object to store session attributes across multiple requests.

16. They support various features like redirection, forwarding, and cookie management to manage client interactions.

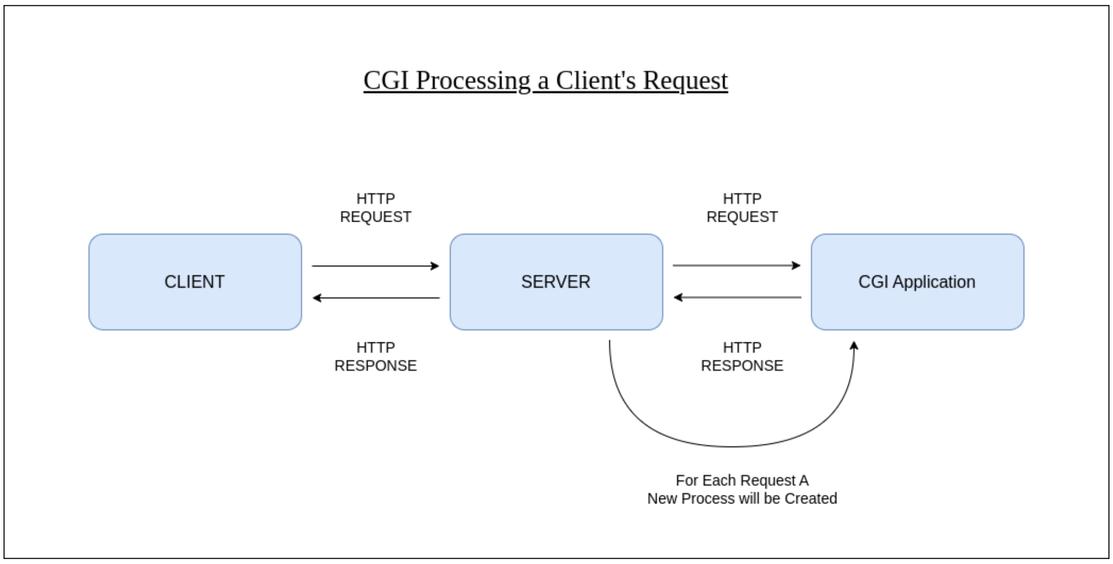
17. Servlets can interact with databases, other web services, or backend systems to process requests and generate responses.

18. They support multithreading, allowing multiple requests to be processed concurrently by a single servlet instance.

Java Application to Database Connection :



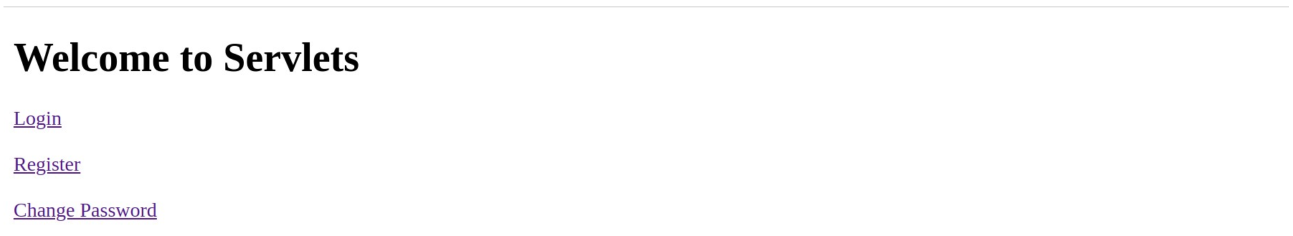
Architecture:


CGI Processing a Client's Request
HTTP REQUEST
HTTP REQUEST
CLIENT
SERVER
CGI Application
HTTP RESPONSE
HTTP RESPONSE
For Each Request A New Process will be Created

Program :

index.html File :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Start Page</title>
</head>
<body>
    <h1>Welcome to Servlets</h1>
    <!-- <form action="servlet" method="POST">
        <label for="s_id">Username:</label>
        <input type="text" id="s_id" name="s_id" required><br><br>
        <label for="s_name">Password:</label>
        <input type="password" id="s_name" name="s_name" required><br><br>
        <input type="submit" value="Login"><br><br>
    </form> -->

    <a href="login.html">Login</a><br><br>
    <a href="register.html">Register</a> <br><br>
    <a href="changePassword.html">Change Password</a><br><br>
</body>
</html>
```



Welcome to Servlets

Login

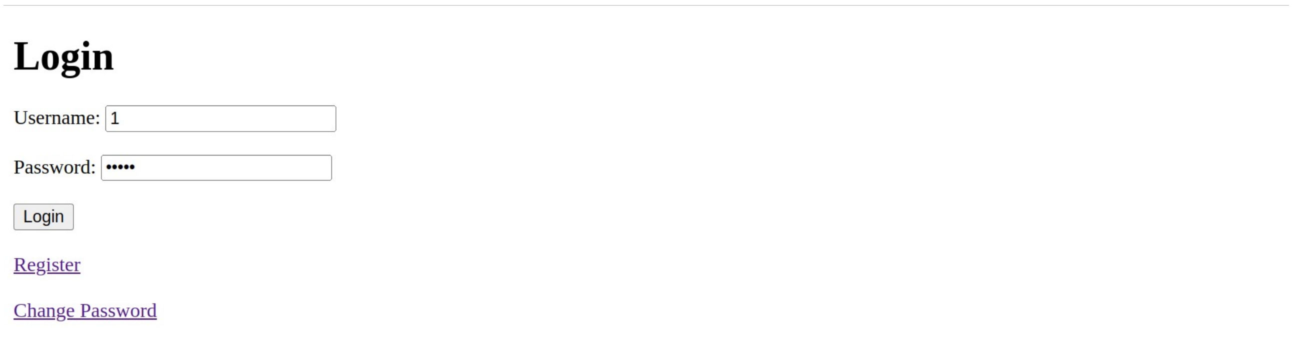
Register

Change Password

login.html File :

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Page</title>
</head>
<body>
    <h1>Login</h1>
    <form action="servlet" method="POST">
        <label for="s_id">Username:</label>
        <input type="text" id="s_id" name="s_id" required><br><br>
        <label for="s_name">Password:</label>
        <input type="password" id="s_name" name="s_name" required><br><br>
        <input type="submit" value="Login"><br><br>
    </form>
    <a href="register.html">Register</a><br><br>
    <a href="changePassword.html">Change Password</a>
</body>
</html>
```

Output :



Login

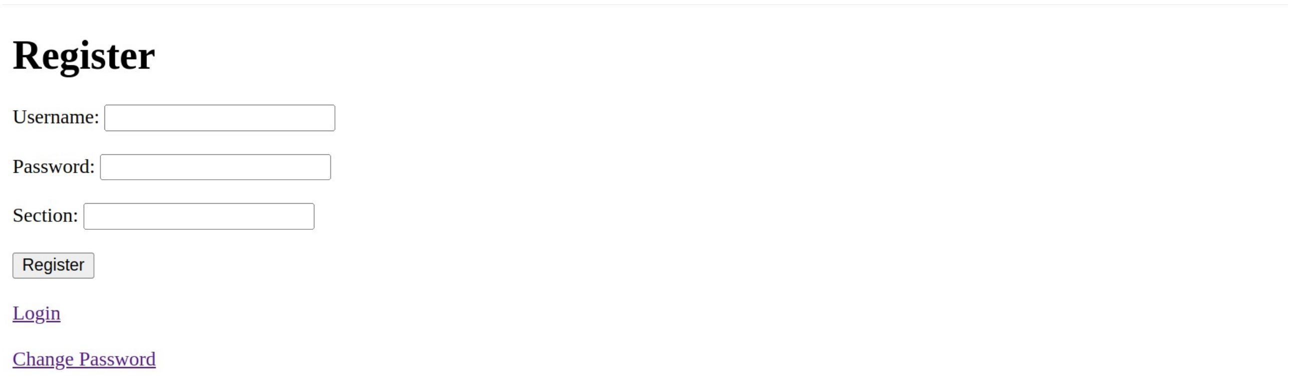
Username: 1

Password: •••••

register.html File:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Page</title>
</head>
<body>
    <h1>Register</h1>
    <form action="servlet1" method="POST">
        <label for="s_id">Username:</label>
        <input type="text" id="s_id" name="s_id" required><br><br>
        <label for="s_name">Password:</label>
        <input type="password" id="s_name" name="s_name" required><br><br>
        <label for="s_sec">Section:</label>
        <input type="text" id="s_sec" name="s_sec" required><br><br>
        <input type="submit" value="Register"><br><br>
    </form>
    <a href="login.html">Login</a> <br><br>
    <a href="changePassword.html">Change Password</a>
</body>
</html>
```

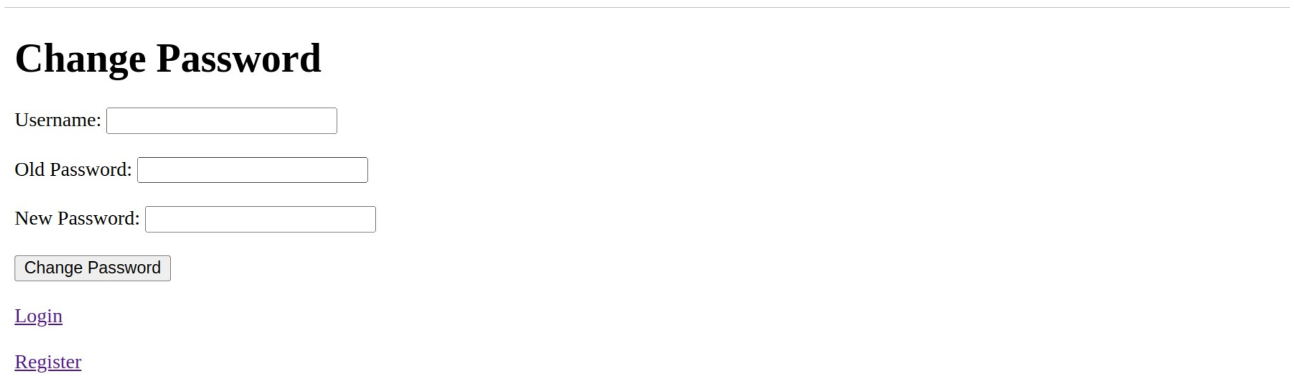
Output :



changePassword.html File :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Password Change</title>
</head>
<body>
    <h1>Change Password</h1>
    <form action="servlet2" method="POST">
        <label for="s_id">Username:</label>
        <input type="text" id="s_id" name="s_id" required><br><br>
        <label for="s_name">Old Password:</label>
        <input type="password" id="s_name" name="s_name" required><br><br>
        <label for="s_sec">New Password:</label>
        <input type="password" id="new_pass" name="new_pass"
required><br><br>
        <input type="submit" value="Change Password"><br><br>
    </form>
    <a href="login.html">Login</a> <br><br>
    <a href="register.html">Register</a>
</body>
</html>
```

Output :


Change Password

Username:

Old Password:

New Password:

servlet.java File :

```
package com.satyavanth;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;

public class servlet extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        int username = Integer.parseInt(request.getParameter("s_id"));
        String password = request.getParameter("s_name");
        try {
            Connection con = null;
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/cse",
"root", "root");
            Statement stmt = con.createStatement();
            String query = "select * from student where s_id="+username+" and
s_name='"+password+"'";
            ResultSet rs = stmt.executeQuery(query);
            if (rs.next()) {
                out.println("Welcome " + username);

            } else {
                out.println("Invalid username or password");
            }
        } catch (Exception e) {
            out.println("Error"+e);
        }
        out.close();
    }

}
```

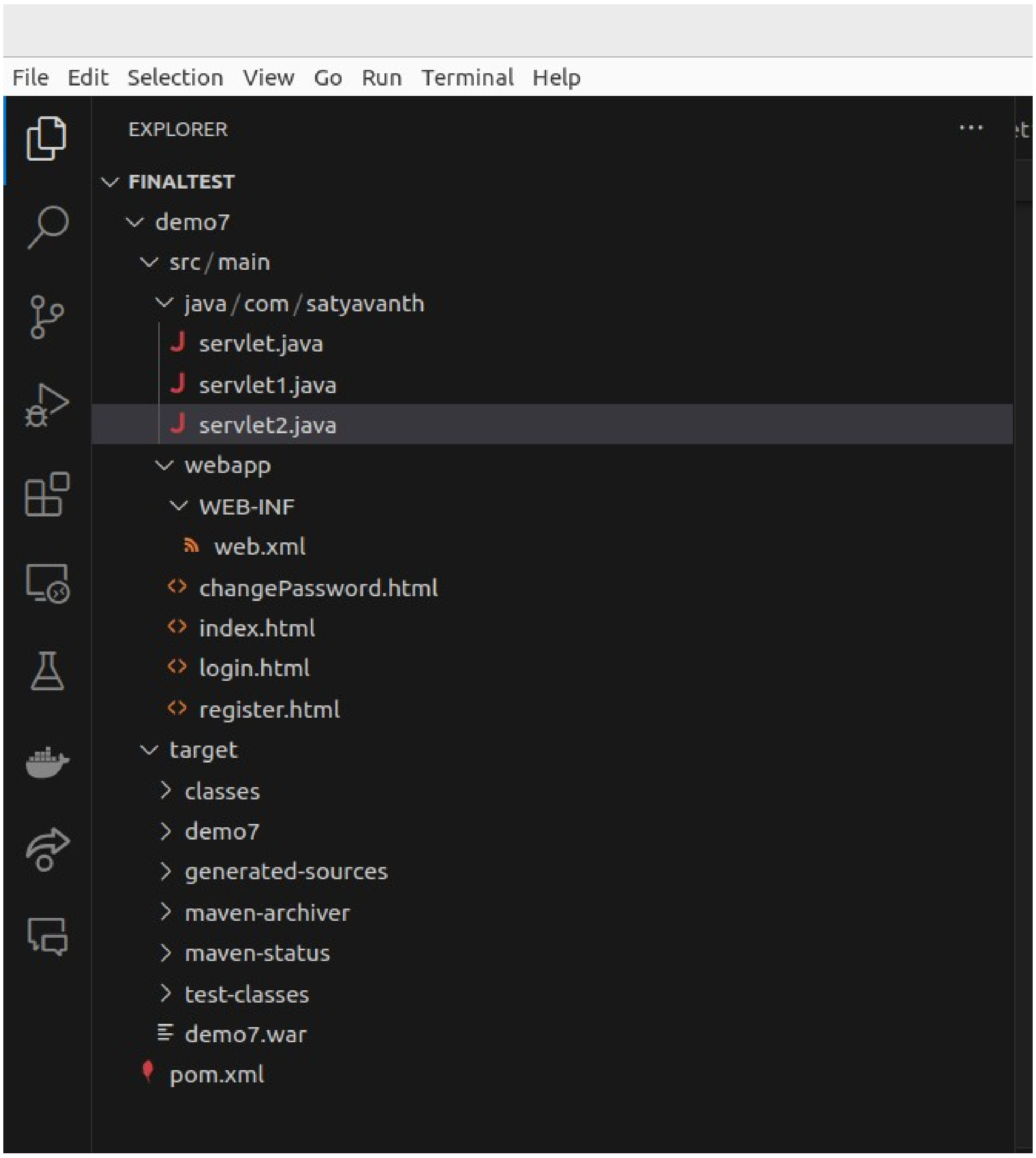
servlet1.java File :

```
package com.satyavanth;

import java.io.IOException;
import java.io.PrintWriter;
// import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;

public class servlet1 extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        int username = Integer.parseInt(request.getParameter("s_id"));
        String password = request.getParameter("s_name");
        String section = request.getParameter("s_sec");
        try {
            Connection con = null;
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/cse",
"root", "root");
            String query = "insert into student
values("+username+",'"+password+"','"+section+"')";
            PreparedStatement pstmt = con.prepareStatement(query);
            int rowInserted = pstmt.executeUpdate();
            if (rowInserted > 0) {
                out.println("Registration successfulll");
            } else {
                out.println("Registration not successfull");
            }
        } catch (Exception e) {
            out.println("Error"+e);
        }
        out.close();
    }

}
```

servlet2.java File :

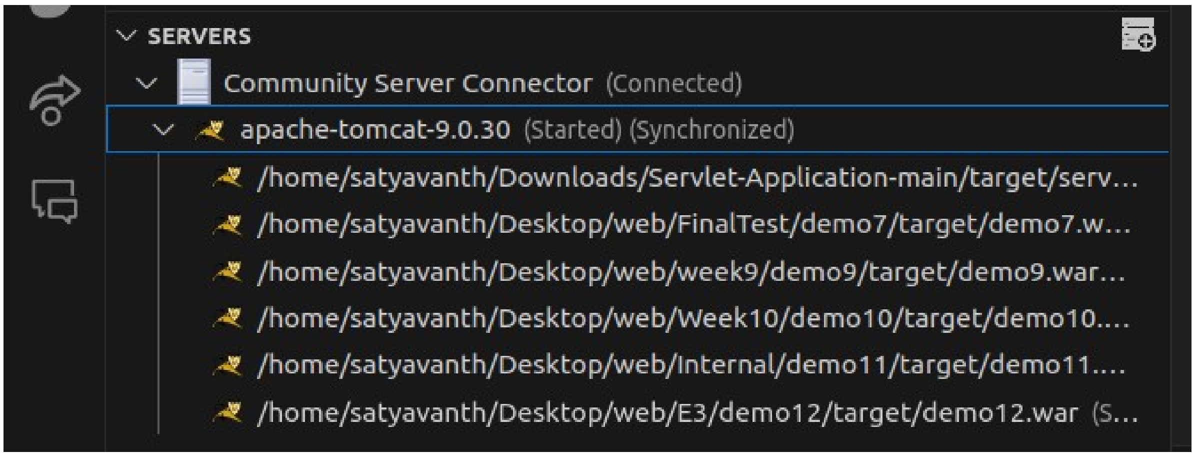
```
package com.satyavanth;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.sql.*;

public class servlet2 extends HttpServlet {
    public void doPost(HttpServletRequest request, HttpServletResponse response)
throws IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        int username = Integer.parseInt(request.getParameter("s_id"));
        String password = request.getParameter("s_name");
        String newpassword = request.getParameter("new_pass");
        try {
            Connection con = null;
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/cse",
"root", "root");
            Statement stmt = con.createStatement();
            String query = "update student set s_name='"+newpassword+"' where
s_id="+username+" and s_name='"+password+"'";
            int rowAffected = stmt.executeUpdate(query);
            // ResultSet rs = stmt.executeQuery(query);
            if (rowAffected > 0) {
                out.println("Password changed successfully");
            } else {
                out.println("Invalid username or old password");
            }

        } catch (Exception e) {
            out.println("Error"+e);
        }
        out.close();
    }

}
```

File Structure :



```
FINALTEST
  demo7
    src / main
      java / com / satyavanth
        servlet.java
        servlet1.java
        servlet2.java
      webapp
        WEB-INF
          web.xml
        changePassword.html
        index.html
        login.html
        register.html
    target
      classes
      demo7
      generated-sources
      maven-archiver
      maven-status
      test-classes
      demo7.war
    pom.xml
```

Servers Started :



```
SERVERS
  Community Server Connector (Connected)
    apache-tomcat-9.0.30 (Started) (Synchronized)
      /home/satyavanth/Downloads/Servlet-Application-main/target/serv...
      /home/satyavanth/Desktop/web/FinalTest/demo7/target/demo7.w...
      /home/satyavanth/Desktop/web/week9/demo9/target/demo9.war...
      /home/satyavanth/Desktop/web/Week10/demo10/target/demo10....
      /home/satyavanth/Desktop/web/Internal/demo11/target/demo11....
      /home/satyavanth/Desktop/web/E3/demo12/target/demo12.war (S...
```

WEEK-9 and WEEK-10

AIM : Create the necessary JSP's for the application chosen.

Scenario : Perform the simple web page using JSP Elements

DESCRIPTION :

1. JSP (JavaServer Pages) is a technology used to create dynamic web pages using Java.

2. It allows embedding Java code within HTML pages to generate dynamic content.

3. JSP files have a .jsp extension and are translated into servlets by the server at runtime.

4. JSP simplifies the process of creating dynamic web content by combining HTML and Java code.

5. JSP pages are compiled into servlets by the servlet container, making them highly efficient.

6. JSP follows the MVC (Model-View-Controller) architecture, separating business logic from presentation.

7. They provide a way to separate the presentation layer from the business logic, enhancing maintainability.

8. JSP supports both static content (HTML) and dynamic content (Java code) within the same page.

9. JSP tags, such as <% %> for Java code and <%= %> for expressions, are used to embed Java code within HTML.

10. Standard JSP tags, known as JSTL (JavaServer Pages Standard Tag Library), provide reusable components for common tasks like iteration, conditional logic, formatting, etc.

11. Custom tags or tag libraries can be created to encapsulate complex functionality and promote reusability.

12. JSP pages can access request parameters, session attributes, and other data using implicit objects like request, response, session, etc.

13. They support expression language (EL) for accessing and manipulating data stored in JavaBeans, session attributes, and other scopes.

14. JSP pages can include other JSP files or static content using directives like <%@ include %> and <%@ taglib %> for tag library inclusion.

15. JSP pages can be internationalized by using resource bundles and incorporating localized text.

16. JSPs can handle HTTP requests like GET and POST, process form data, and generate dynamic responses.

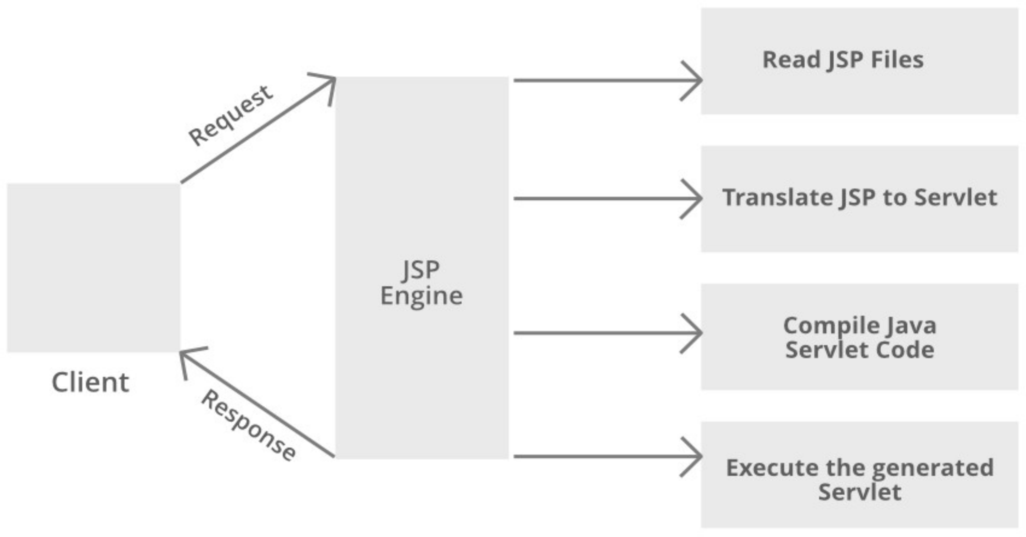
17. JSP pages can interact with databases, web services, and other backend systems using Java code.

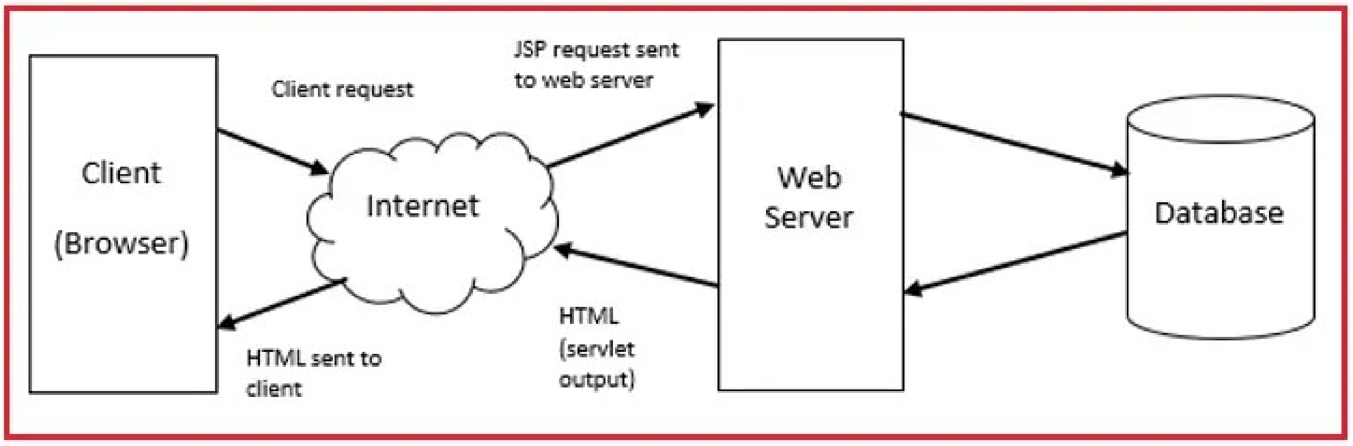
18. JSPs provide support for session management, allowing the maintenance of user sessions across multiple requests.

19. JSP pages are often combined with servlets to create robust web applications, where servlets handle the business logic and JSP handles the presentation.

20. JSP pages are widely used in web development due to their simplicity, flexibility, and integration with Java EE technologies.

<u>Architecture of JSP :</u>

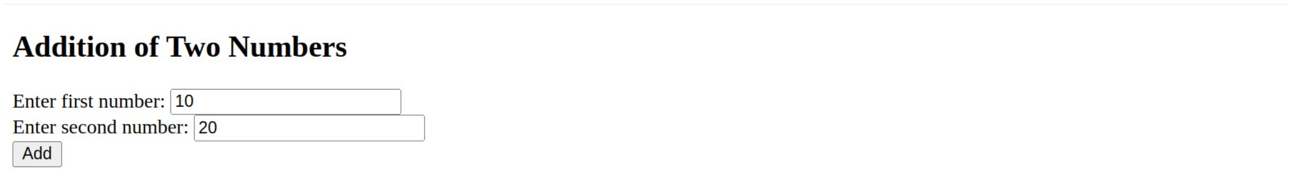




index.jsp File:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Addition</title>
</head>
<body>
    <h2>Addition of Two Numbers</h2>
    <form action="addition.jsp" method="post">
        Enter first number: <input type="text" name="num1"><br>
        Enter second number: <input type="text" name="num2"><br>
        <input type="submit" value="Add">
    </form>
</body>
</html>
```

Output :

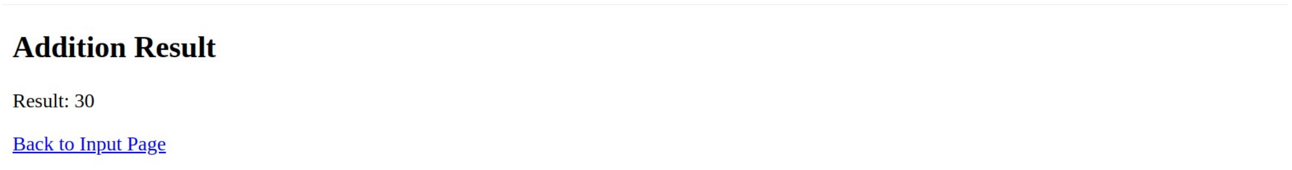


```
Addition of Two Numbers

Enter first number: 10
Enter second number: 20
Add
```

addition.jsp File :

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Addition Result</title>
</head>
<body>

    <h2>Addition Result</h2>
    <%
        if (request.getMethod().equals("POST")) {
            try {
                int num1 = Integer.parseInt(request.getParameter("num1"));
                int num2 = Integer.parseInt(request.getParameter("num2"));
                int sum = num1 + num2;
    %>
                <p>Result: <%= sum %></p>
    <%
            }
            catch (NumberFormatException e) {
    %>
                <p>Please enter valid numbers.</p>
    <%
            }
        } else {
    %>
        <p>Please submit the form to see the result.</p>
    <%
        }
    %>
    <a href="index.jsp">Back to Input Page</a>

</body>
</html>
```

Output :



Addition Result

Result: 30

Back to Input Page