LAB 6: Database Programming in Servlets

Lab Activities

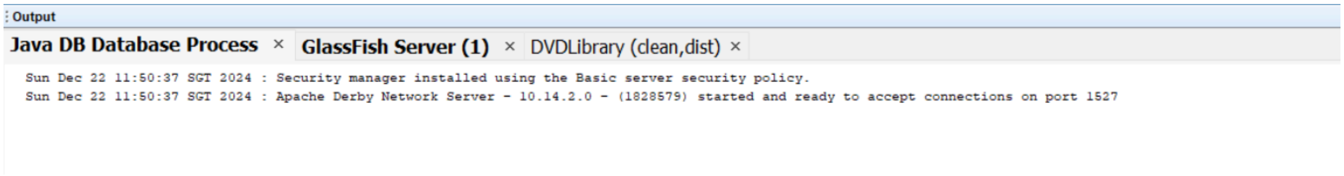
1. Create a database and a table in Java Derby

Task 1: Starting the Server and Creating a Database

The Java DB Database menu options are displayed when you right-click the Java DB node in the Services window. This contextual menu items allow you to start and stop the database server, create a new database instance, as well as register database servers in the IDE (as demonstrated in the previous step).

To start the database server:

a. In the Services window, right-click the Java DB node and choose Start Server. Note the following output in the Output window, indicating that the server has started:

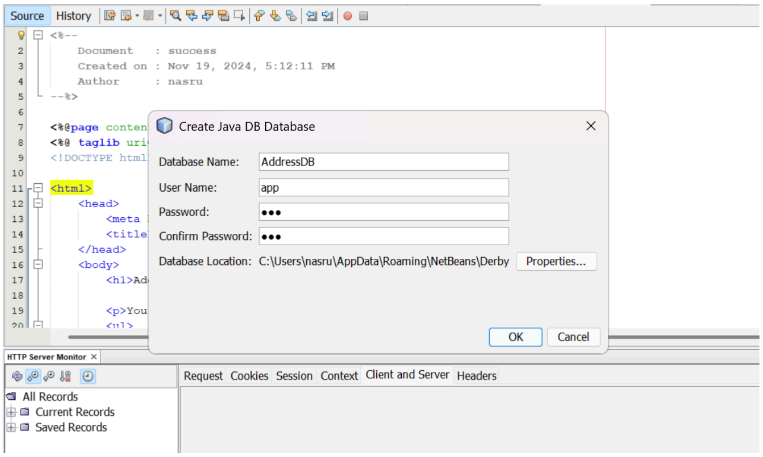


```
Sun Dec 22 11:50:37 SGT 2024 : Security manager installed using the Basic server security policy.
Sun Dec 22 11:50:37 SGT 2024 : Apache Derby Network Server - 10.14.2.0 - (1828579) started and ready to accept connections on port 1527
```

b. Right-click the Java DB node and choose Create Database to open the Create Java DB Database dialog.

c. Type AddressDB for the Database Name.

d. Type app for the User Name and Password. Click OK.



```
<%--
    Document   : success
    Created on : Nov 19, 2024, 5:12:11 PM
    Author     : nasru
--%>

<%@page conten
<%@ taglib uri
<!DOCTYPE html

<html>
    <head>
        <meta
        <title
    </head>
    <body>
        <h1>Ad

        <p>You
        <ul>
```

Create Java DB Database

Database Name: AddressDB

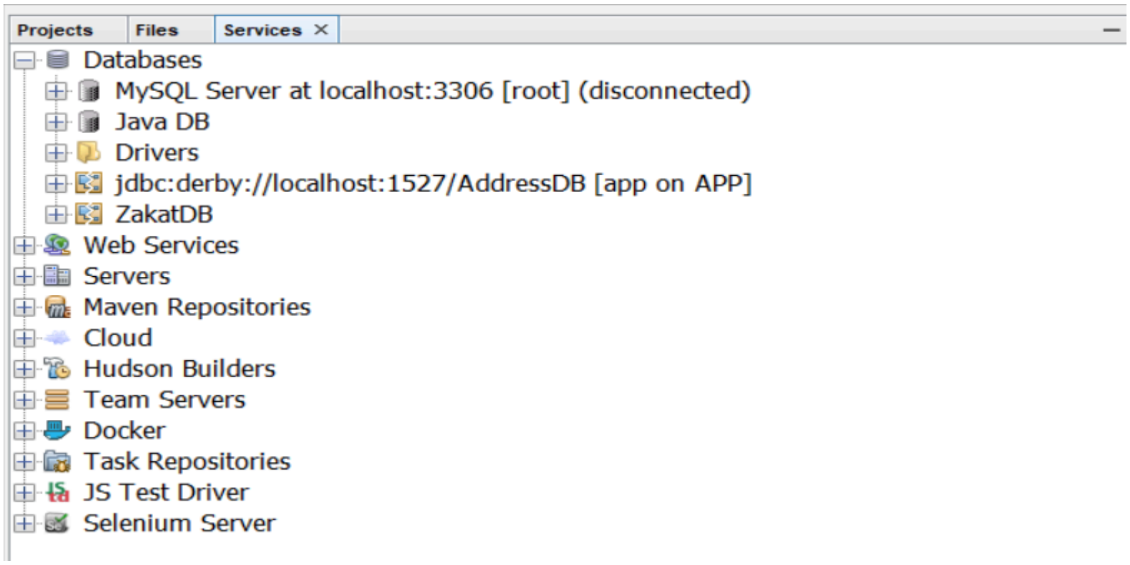
User Name: app

Password: •••

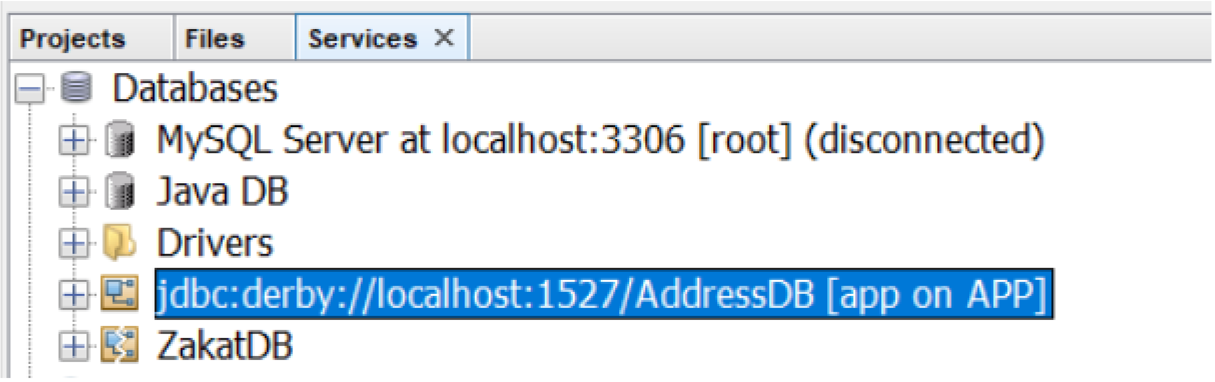
Confirm Password: •••

Database Location: C:\Users\nasru\AppData\Roaming\NetBeans\Derby

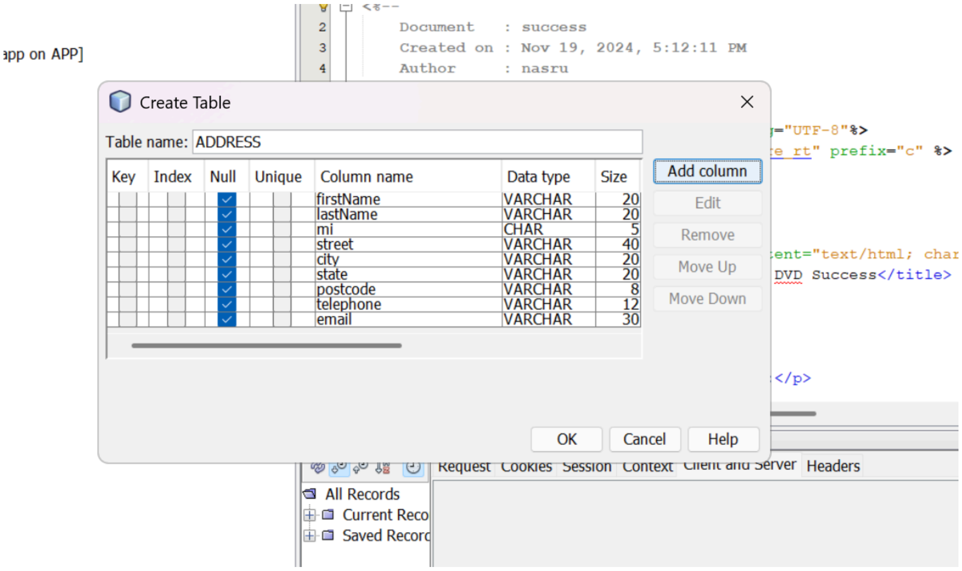
After you create the database, if you expand the Databases node in the Services window you can see that the IDE created a database connection and that the database was added to the list under the Java DB node.



Task 2: Connecting to the Database


Projects
Files
Services
Databases
MySQL Server at localhost:3306 [root] (disconnected)
Java DB
Drivers
jdbc:derby://localhost:1527/AddressDB [app on APP]
ZakatDB

Task 3) Creating Tables



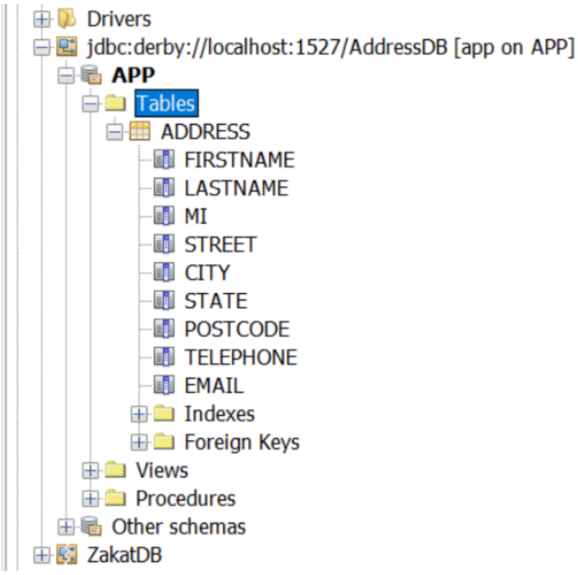
```
Document   : success
Created on : Nov 19, 2024, 5:12:11 PM
Author     : nasru
```

Create Table

Table name: ADDRESS

Key	Index	Null	Unique	Column name	Data type	Size
		✓		firstName	VARCHAR	20
		✓		lastName	VARCHAR	20
		✓		mi	CHAR	5
		✓		street	VARCHAR	40
		✓		city	VARCHAR	20
		✓		state	VARCHAR	20
		✓		postcode	VARCHAR	8
		✓		telephone	VARCHAR	12
		✓		email	VARCHAR	30

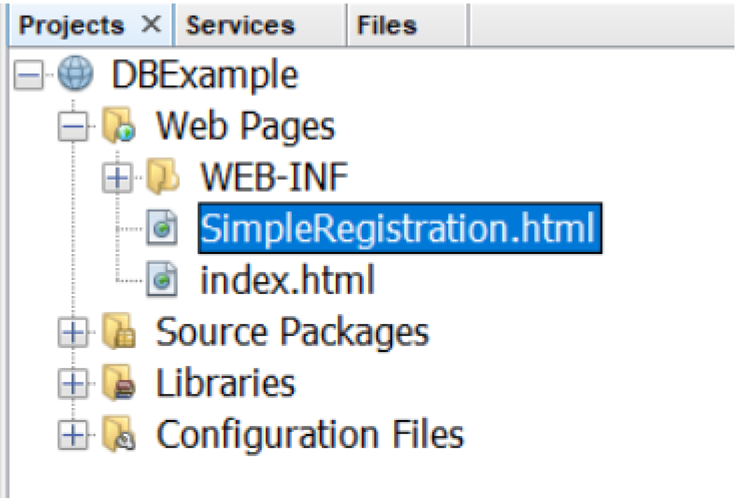
f. When you are sure that your Create Table dialog contains the same specifications as those shown above, click OK. The IDE generates the ADDRESS table in the database, and you can see a new ADDRESS table node (▦) display under the Tables node.



- Drivers
- jdbc:derby://localhost:1527/AddressDB [app on APP]
  - APP
    - Tables
      - ADDRESS
        - FIRSTNAME
        - LASTNAME
        - MI
        - STREET
        - CITY
        - STATE
        - POSTCODE
        - TELEPHONE
        - EMAIL
        - Indexes
        - Foreign Keys
    - Views
    - Procedures
  - Other schemas
- ZakatDB

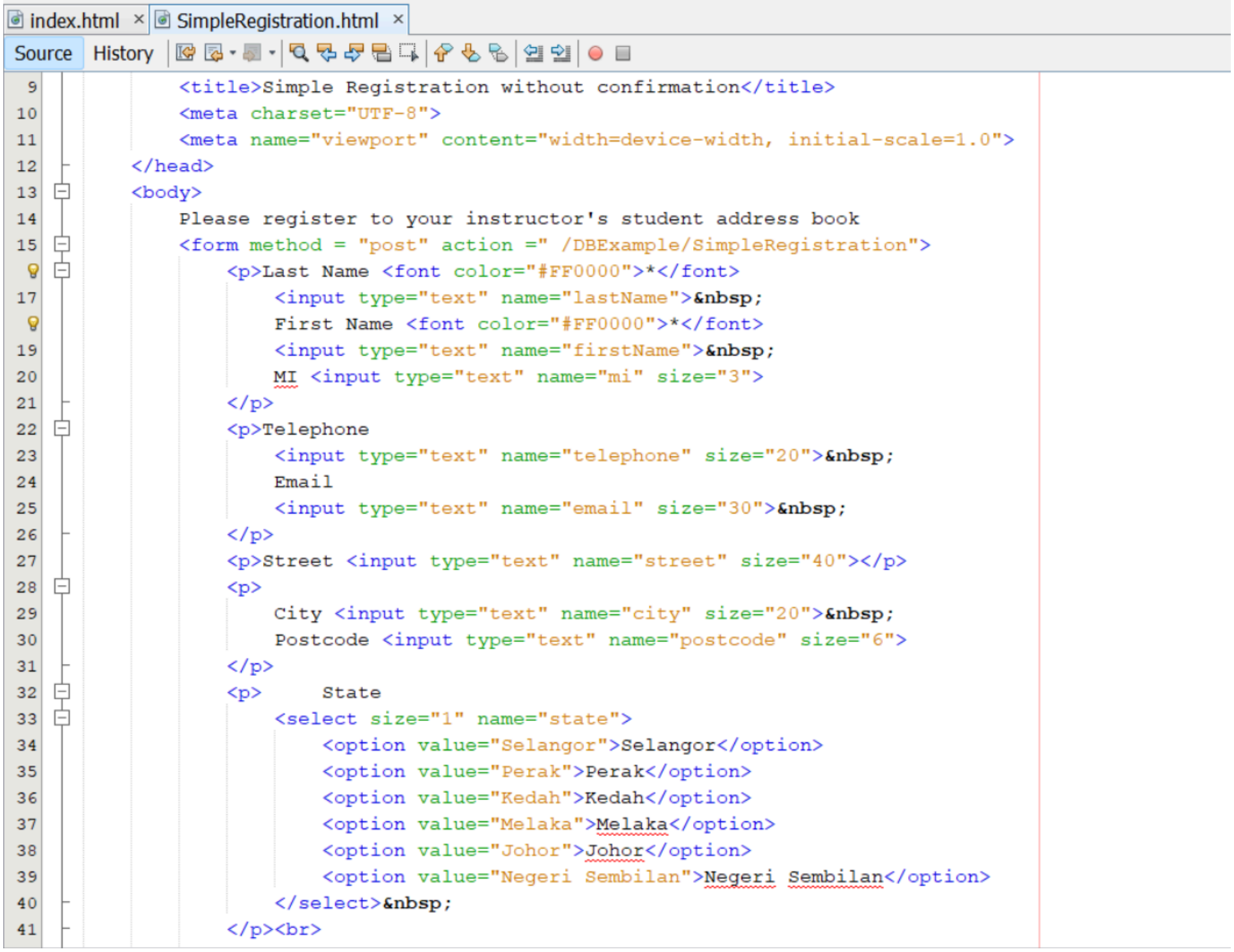
2. Creating web components.

Task 1: Create a web project



Task 2: Create an HTML file

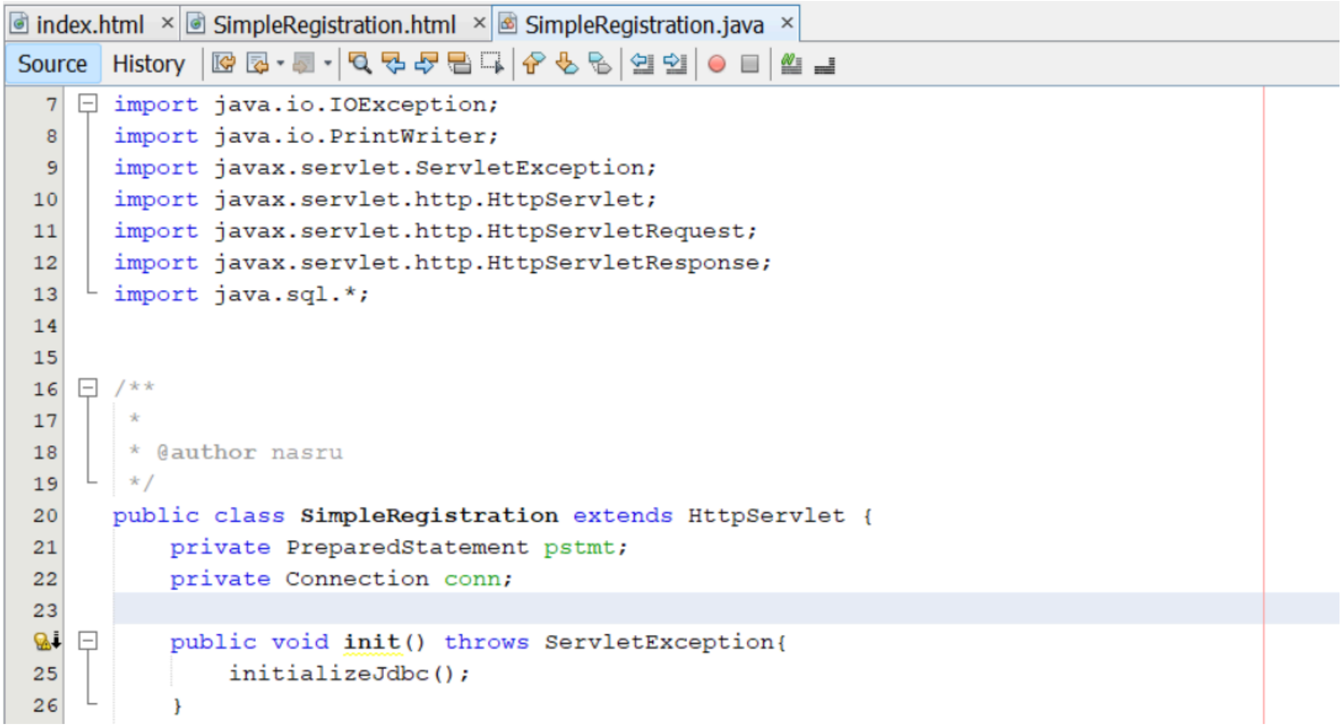
a. Create an HTML file name **SimpleRegistration.html** for collecting the data and sending it to the database using the post method.



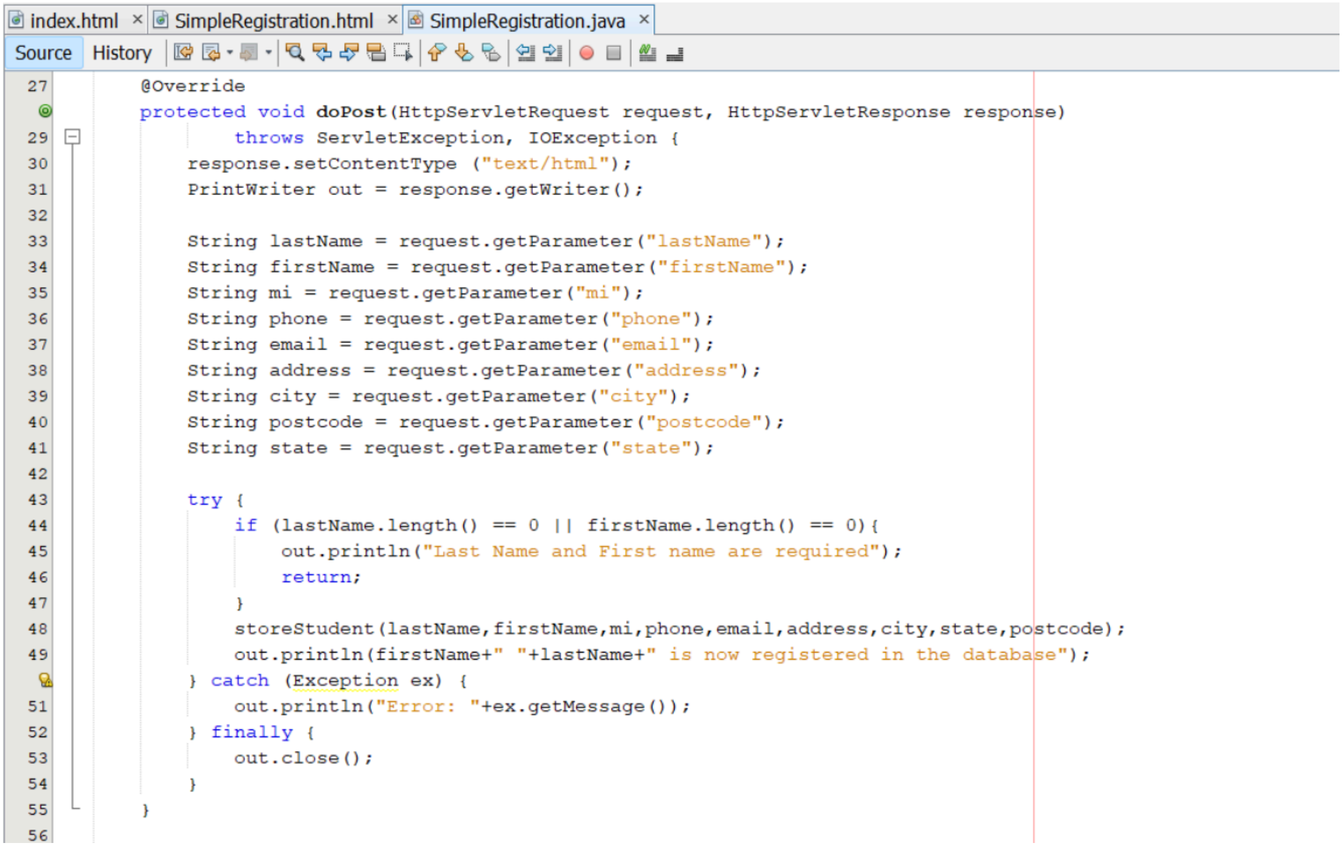
```
        <title>Simple Registration without confirmation</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        Please register to your instructor's student address book
        <form method = "post" action =" /DBExample/SimpleRegistration">
            <p>Last Name <font color="#FF0000">*</font>
                <input type="text" name="lastName">&nbsp;
                First Name <font color="#FF0000">*</font>
                <input type="text" name="firstName">&nbsp;
                MI <input type="text" name="mi" size="3">
            </p>
            <p>Telephone
                <input type="text" name="telephone" size="20">&nbsp;
                Email
                <input type="text" name="email" size="30">&nbsp;
            </p>
            <p>Street <input type="text" name="street" size="40"></p>
            <p>
                City <input type="text" name="city" size="20">&nbsp;
                Postcode <input type="text" name="postcode" size="6">
            </p>
            <p>     State
                <select size="1" name="state">
                    <option value="Selangor">Selangor</option>
                    <option value="Perak">Perak</option>
                    <option value="Kedah">Kedah</option>
                    <option value="Melaka">Melaka</option>
                    <option value="Johor">Johor</option>
                    <option value="Negeri Sembilan">Negeri Sembilan</option>
                </select>&nbsp;
            </p><br>
```

Task 3: Create a Servlet

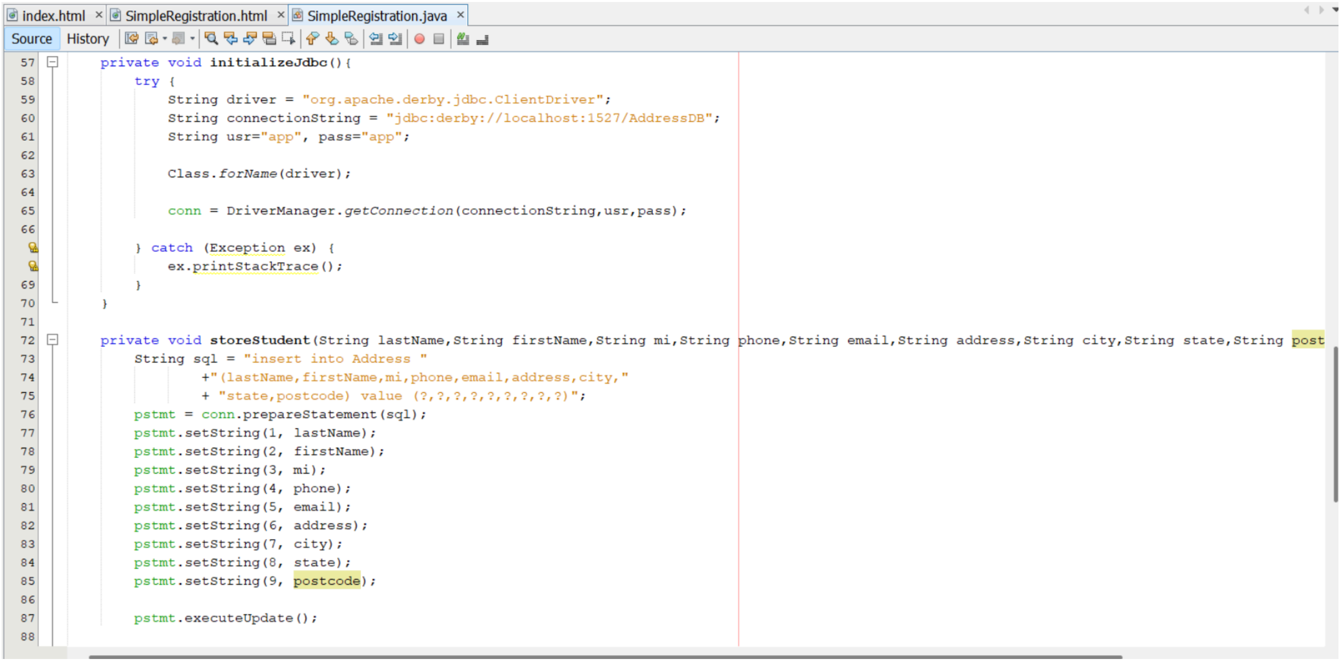
a. Create a Java Servlet with the following characteristic: Class name: **SimpleRegistration**



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


/**
 *
 * @author nasru
 */
public class SimpleRegistration extends HttpServlet {
    private PreparedStatement pstmt;
    private Connection conn;

    public void init() throws ServletException{
        initializeJdbc();
    }
```



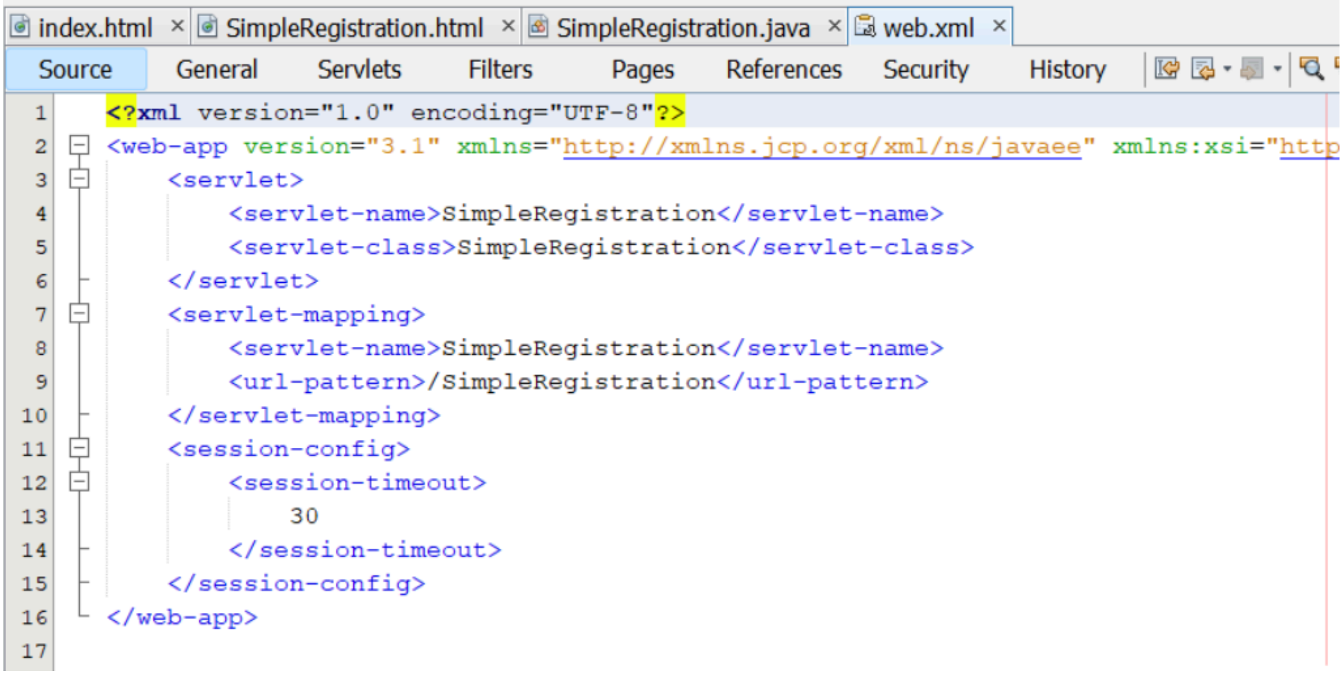
```java
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType ("text/html");
        PrintWriter out = response.getWriter();

        String lastName = request.getParameter("lastName");
        String firstName = request.getParameter("firstName");
        String mi = request.getParameter("mi");
        String phone = request.getParameter("phone");
        String email = request.getParameter("email");
        String address = request.getParameter("address");
        String city = request.getParameter("city");
        String postcode = request.getParameter("postcode");
        String state = request.getParameter("state");

        try {
            if (lastName.length() == 0 || firstName.length() == 0){
                out.println("Last Name and First name are required");
                return;
            }
            storeStudent(lastName,firstName,mi,phone,email,address,city,state,postcode);
            out.println(firstName+" "+lastName+" is now registered in the database");
        } catch (Exception ex) {
            out.println("Error: "+ex.getMessage());
        } finally {
            out.close();
        }
    }

```



```java
    private void initializeJdbc(){
        try {
            String driver = "org.apache.derby.jdbc.ClientDriver";
            String connectionString = "jdbc:derby://localhost:1527/AddressDB";
            String usr="app", pass="app";

            Class.forName(driver);

            conn = DriverManager.getConnection(connectionString,usr,pass);

        } catch (Exception ex) {
            ex.printStackTrace();
        }
    }

    private void storeStudent(String lastName,String firstName,String mi,String phone,String email,String address,String city,String state,String post
        String sql = "insert into Address "
                +"(lastName,firstName,mi,phone,email,address,city,"
                + "state,postcode) value (?,?,?,?,?,?,?,?,?)";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, lastName);
        pstmt.setString(2, firstName);
        pstmt.setString(3, mi);
        pstmt.setString(4, phone);
        pstmt.setString(5, email);
        pstmt.setString(6, address);
        pstmt.setString(7, city);
        pstmt.setString(8, state);
        pstmt.setString(9, postcode);

        pstmt.executeUpdate();
```

3. Configuring the Web application

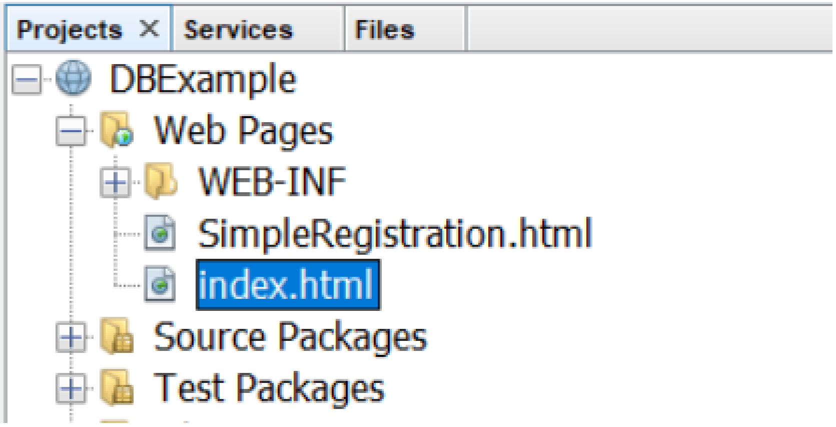
Task 1 : Verify the view servlet configuration



```xml
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xmlns:xsi="http
    <servlet>
        <servlet-name>SimpleRegistration</servlet-name>
        <servlet-class>SimpleRegistration</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>SimpleRegistration</servlet-name>
        <url-pattern>/SimpleRegistration</url-pattern>
    </servlet-mapping>
    <session-config>
        <session-timeout>
            30
        </session-timeout>
    </session-config>
</web-app>
```

Task 2: Create the homepage

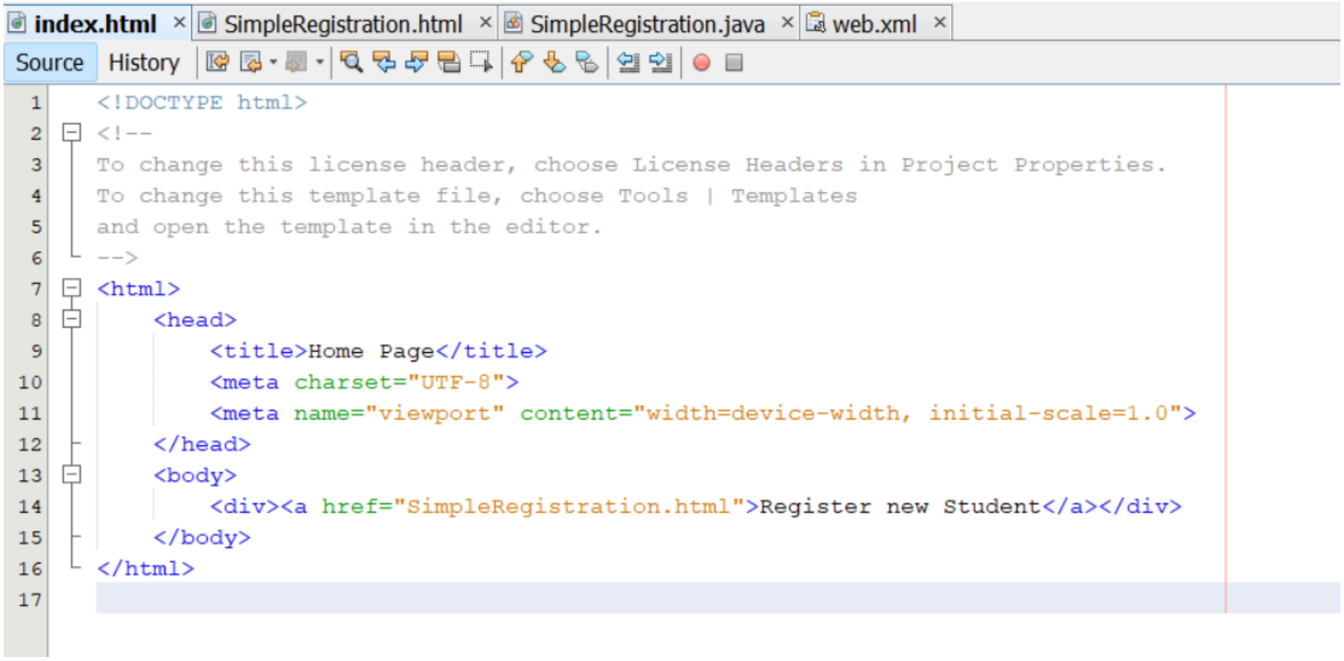
a. Create an HTML file with the following characteristic:



b. Edit the index.html file so that it displays the following text.

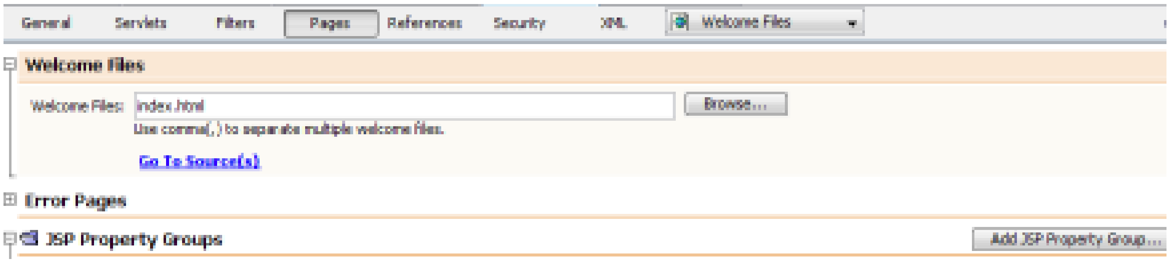
Register new student

Edit the index.html file to add a link to the SimpleRegistration.html page.



```
<!DOCTYPE html>
<!--
To change this license header, choose License Headers in Project Properties.
To change this template file, choose Tools | Templates
and open the template in the editor.
-->
<html>
    <head>
        <title>Home Page</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    </head>
    <body>
        <div><a href="SimpleRegistration.html">Register new Student</a></div>
    </body>
</html>
```

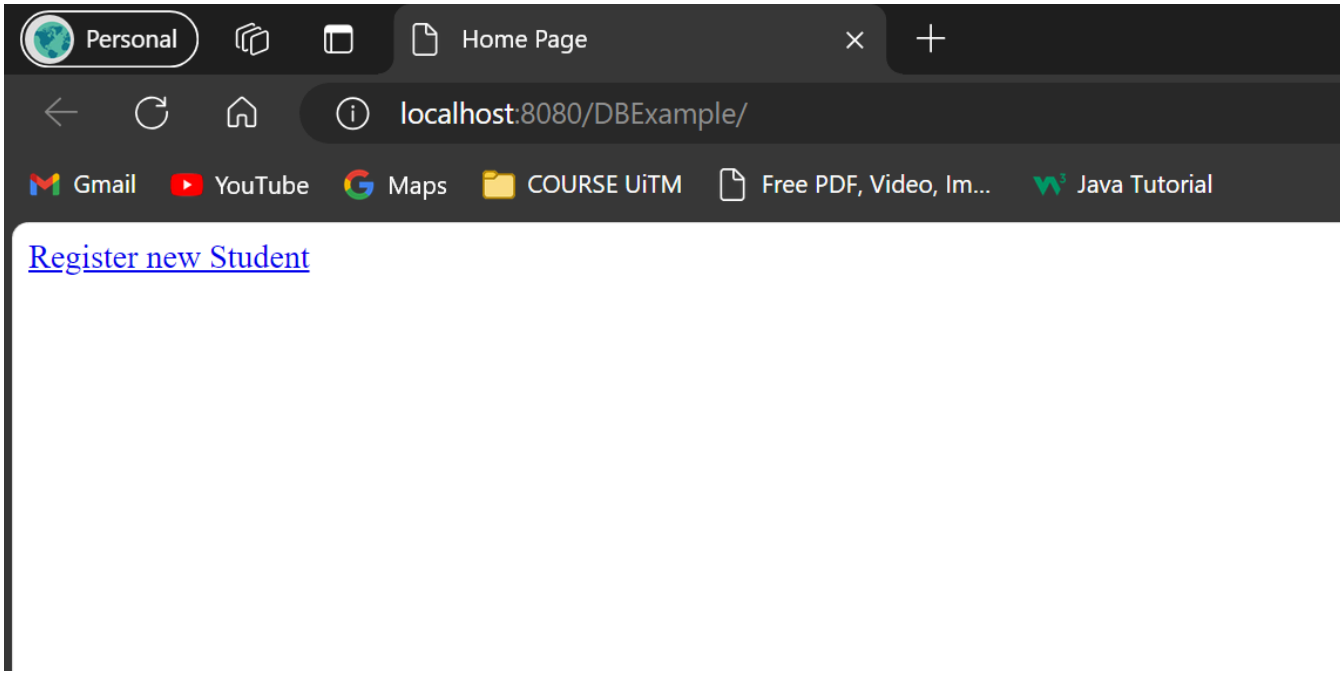
c. Open the web.xml deployment descriptor and set index.html as the welcome file.



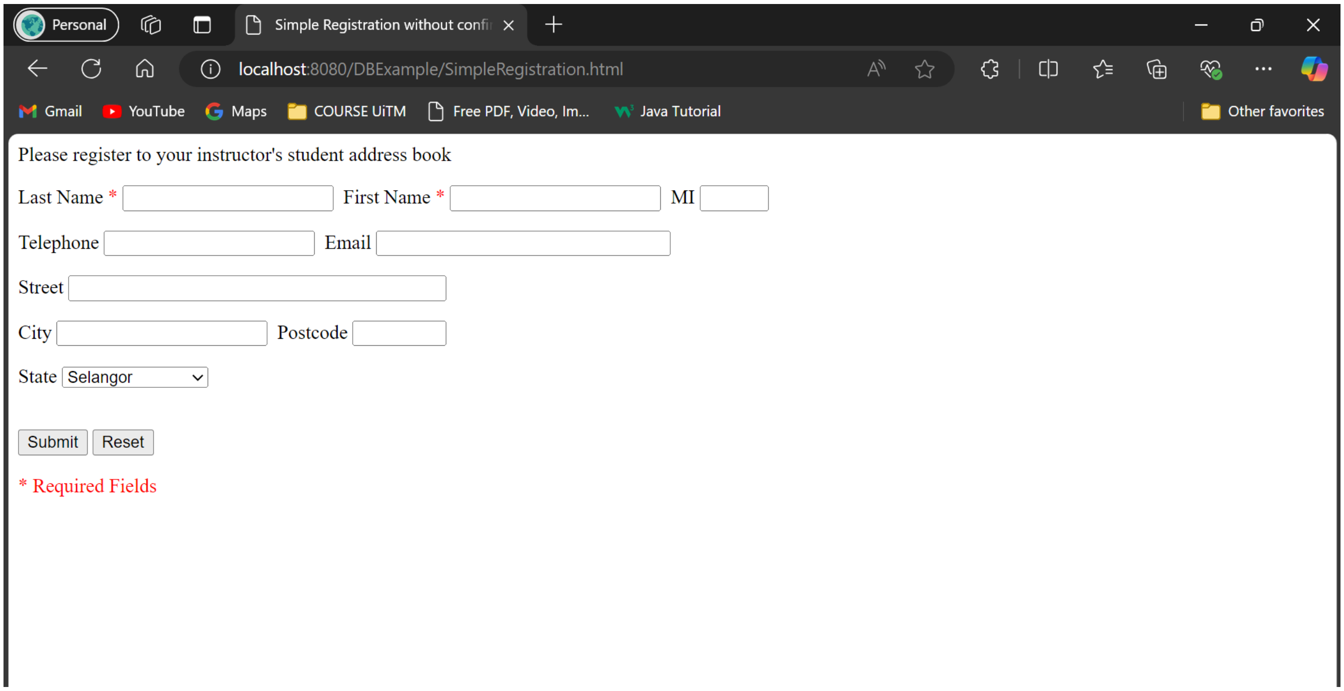
Welcome Files: index.html

Use comma(,) to separate multiple welcome files.

Task 3: Deploy the web application



Register new Student



Please register to your instructor's student address book

Last Name * ______ First Name * ______ MI ______

Telephone ______ Email ______

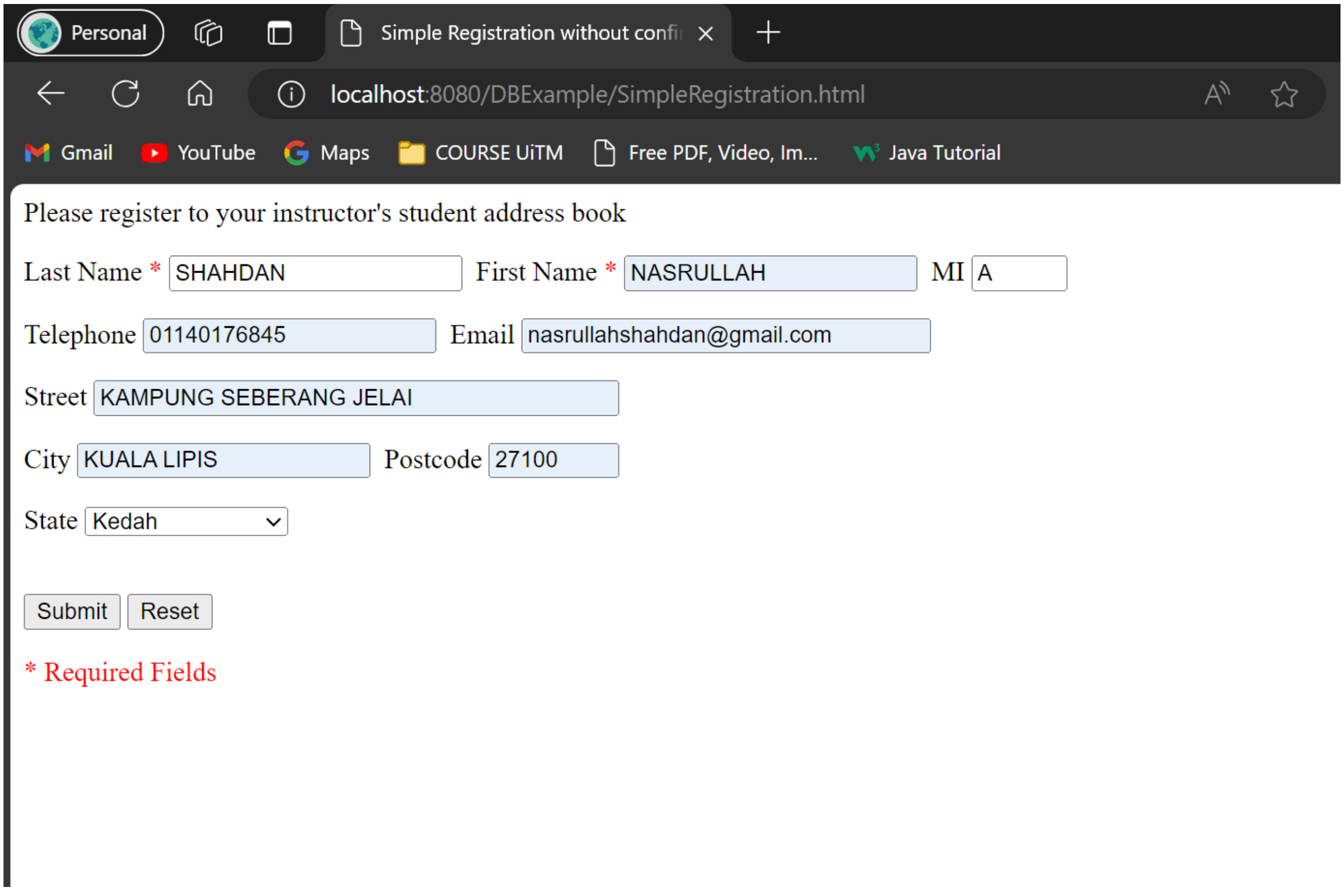
Street ______

City ______ Postcode ______

State Selangor

Submit Reset

* Required Fields


Personal
Simple Registration without confi
localhost:8080/DBExample/SimpleRegistration.html
Gmail
YouTube
Maps
COURSE UiTM
Free PDF, Video, Im...
Java Tutorial

Please register to your instructor's student address book

Last Name * SHAHDAN First Name * NASRULLAH MI A

Telephone 01140176845 Email nasrullahshahdan@gmail.com

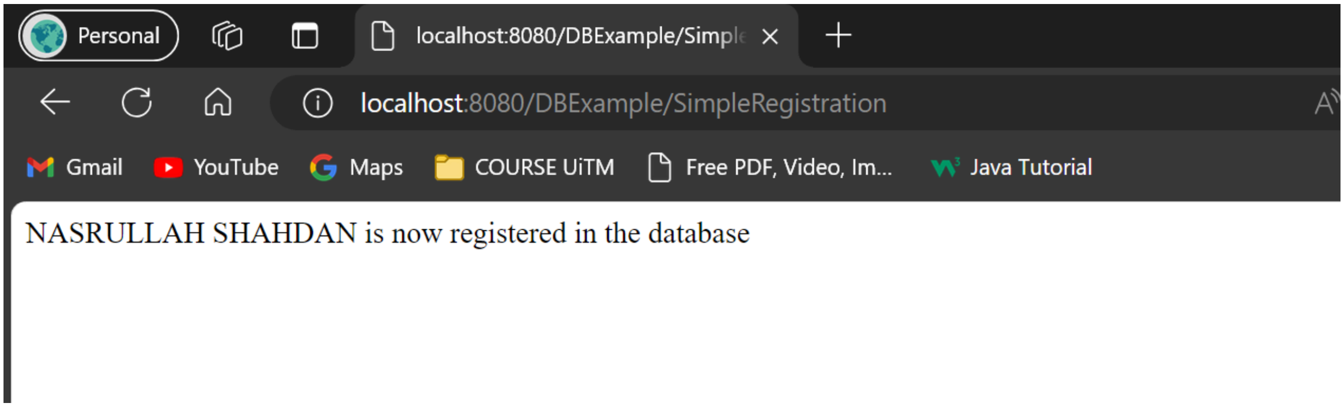
Street KAMPUNG SEBERANG JELAI

City KUALA LIPIS Postcode 27100

State Kedah

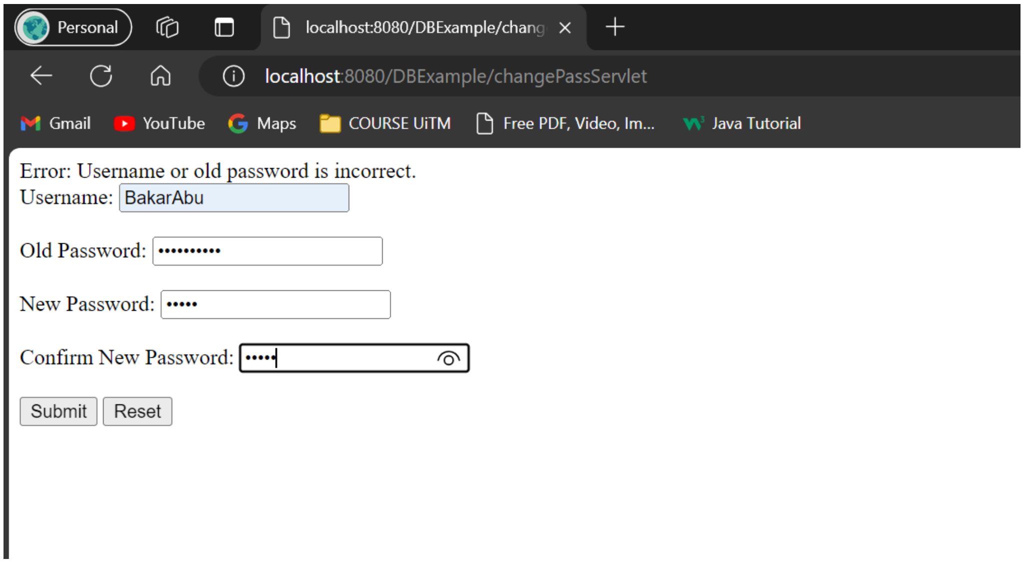
Submit Reset

* Required Fields


Personal
localhost:8080/DBExample/Simple
localhost:8080/DBExample/SimpleRegistration
Gmail
YouTube
Maps
COURSE UiTM
Free PDF, Video, Im...
Java Tutorial

NASRULLAH SHAHDAN is now registered in the database

Postlab Exercise

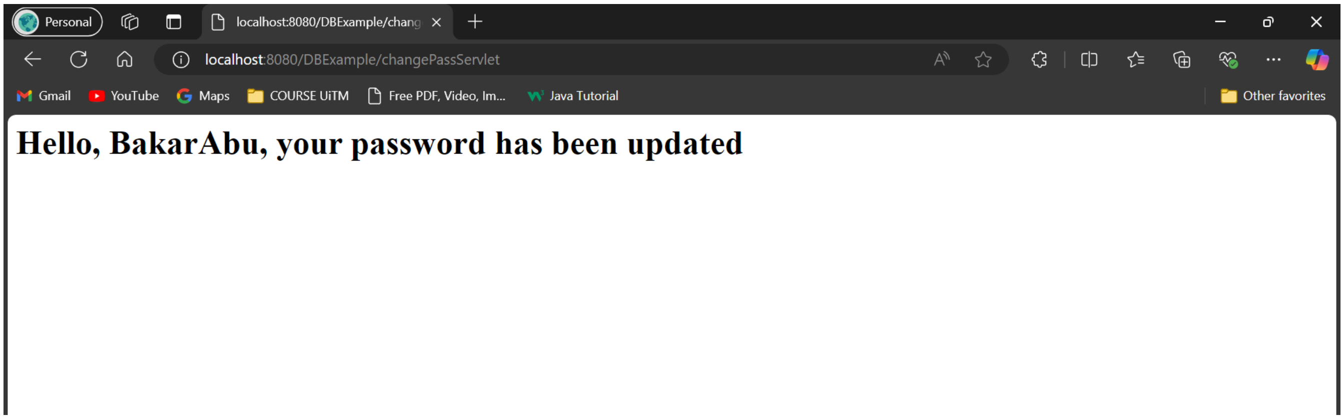


Error: Username or old password is incorrect.
Username: BakarAbu

Old Password: ••••••••••

New Password: •••••

Confirm New Password: •••••



Hello, BakarAbu, your password has been updated