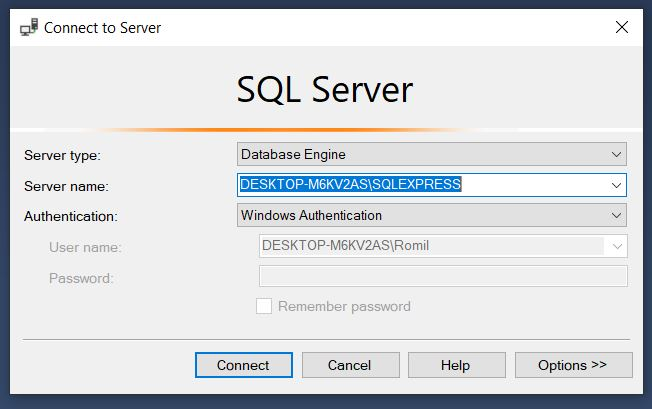
**Date: 21/10/2020**

**Practical no 7**

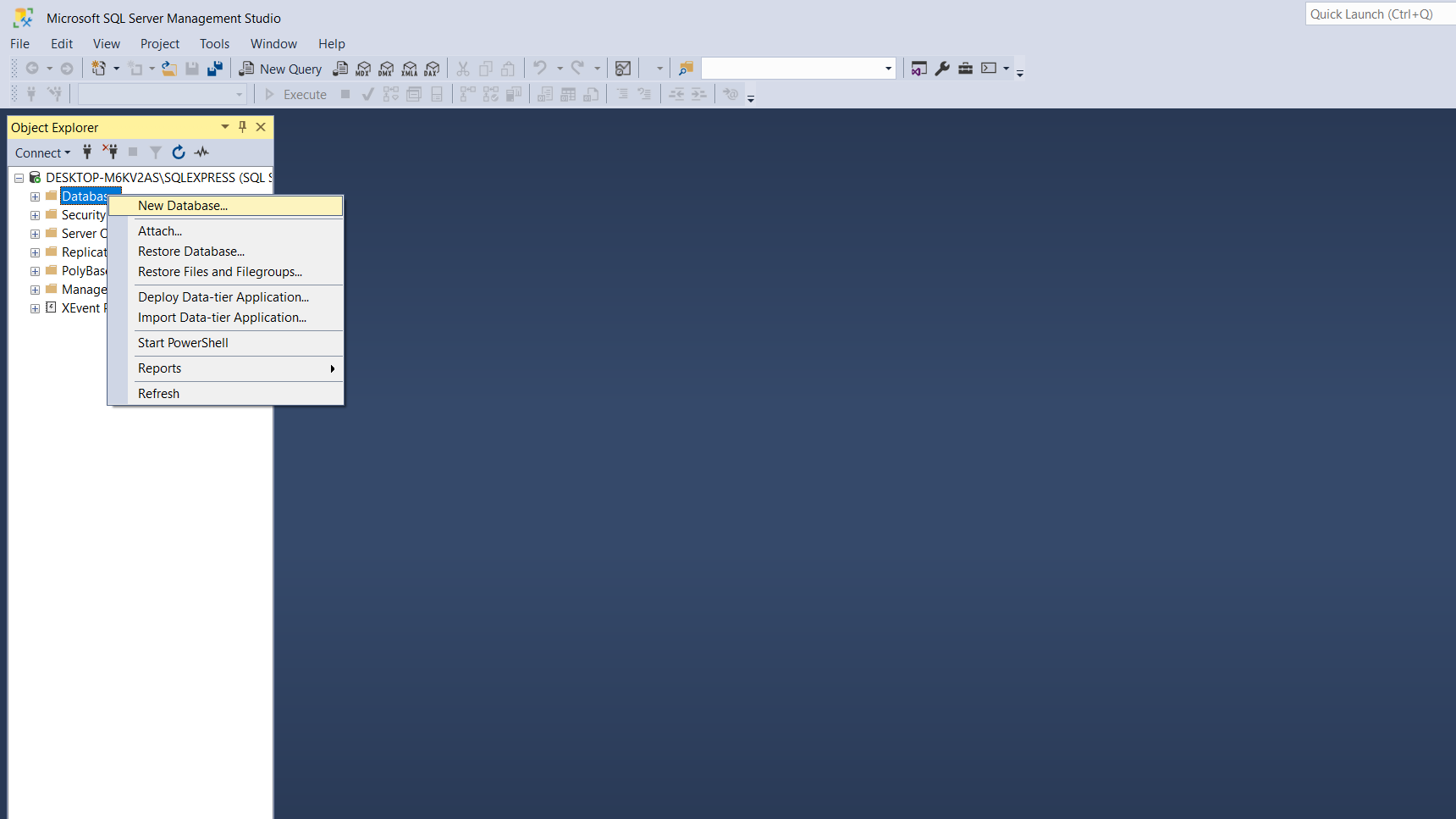
**AIM**: **:** Define a web service method that returns the contents of a database in a JSON string. The contents should be displayed in a tabular format.

**Steps for creating a Database:**

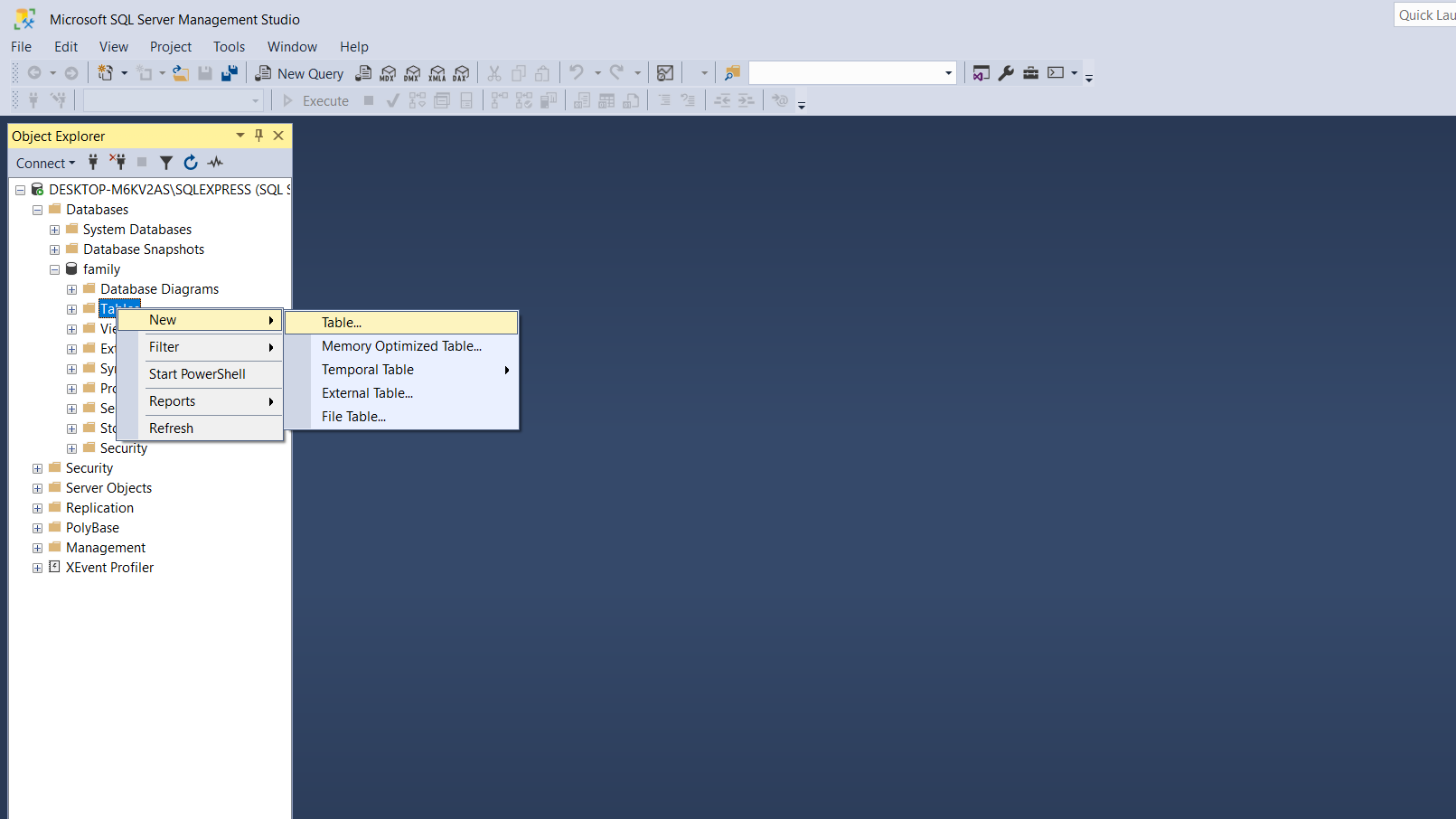
**1]** Open Microsoft SQL Server Management Studio. After opening the Management Studio it will ask to connect to a server. Select a proper server and click connect.



**2]** From object explorer right click on **Databases** and click **New Database.** Give name to Database.



3] Right Click on **Tables** and select **New ->Table**.

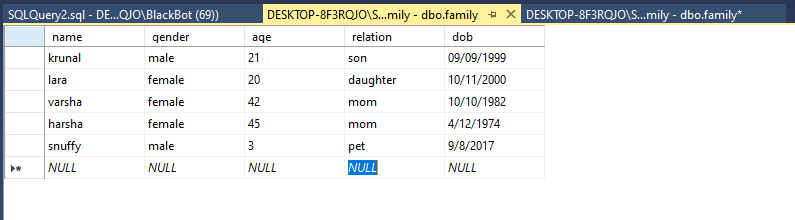


**4]** Edit column name and datatypes.

**5]** Save the table and give a name.

**6]** From **Tables** select the table you created. Right click on it and select **Edit top 200 rows.**

**7]** Enter the data into the table and save.



**CODE:-**

**Family.cs:**

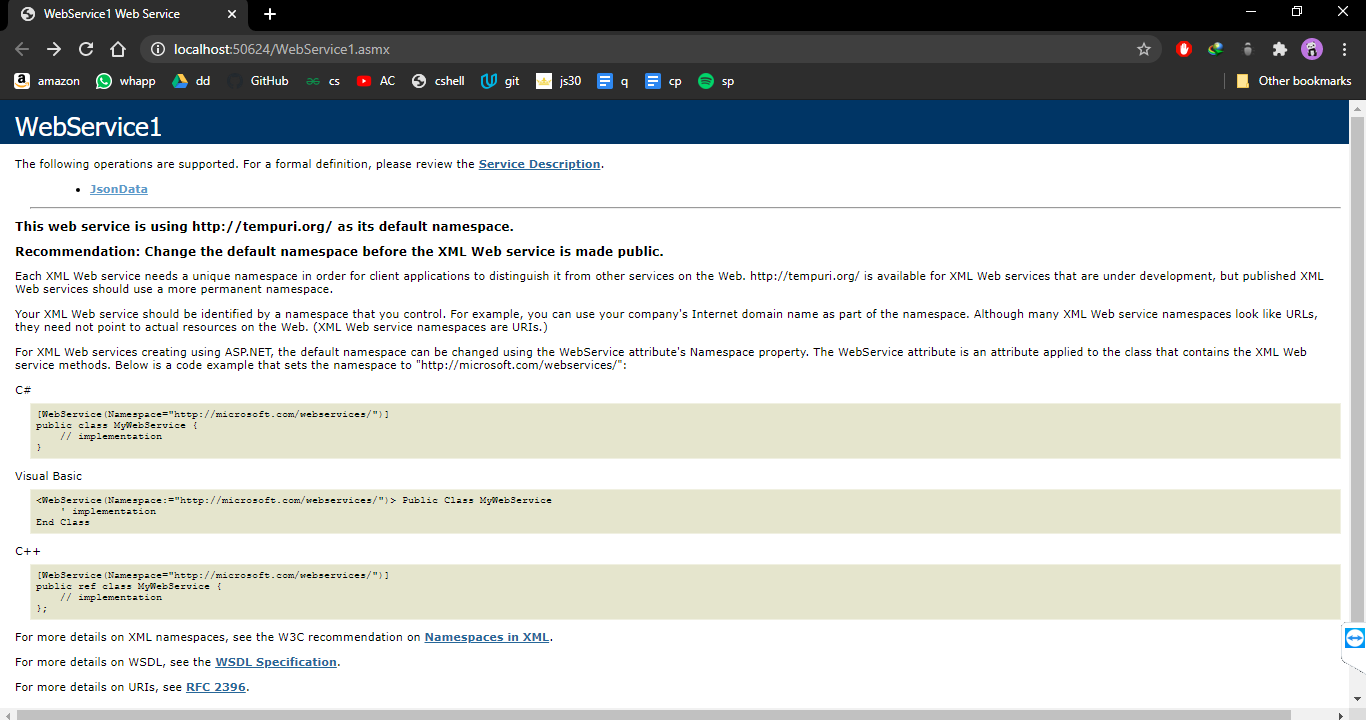
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  namespace prac7  {  public class Family  {  public string name { get; set; }  public string gender { get; set; }  public string age { get; set; }  public string relation { get; set; }  public string dob { get; set; }  }  } |

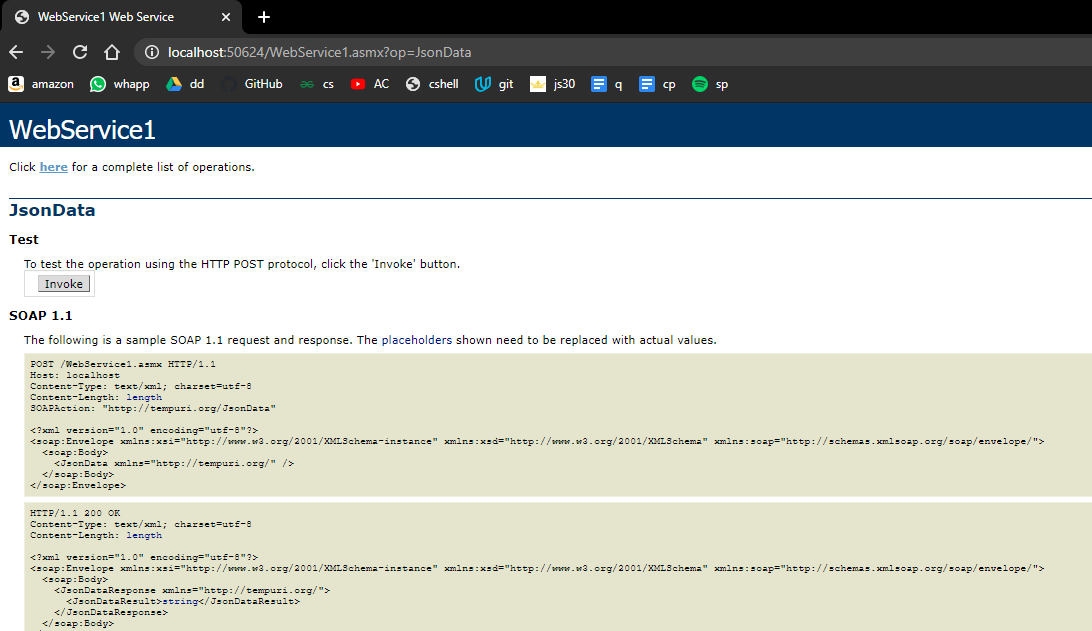
**Webservice.asmx.cs:**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Configuration;  using System.Data.SqlClient;  using System.Linq;  using System.Web;  using System.Web.Script.Serialization;  using System.Web.Services;  namespace prac7  {  /// <summary>  /// Summary description for WebService1  /// </summary>  [WebService(Namespace = "http://tempuri.org/")]  [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]  [System.ComponentModel.ToolboxItem(false)]  public class WebService1 : System.Web.Services.WebService  {  [WebMethod]  public string JsonData()  {  List<Family> objFamilylist = new List<Family>();  string str = ConfigurationManager.ConnectionStrings["DBContext"].ConnectionString;  using (SqlConnection con = new SqlConnection(str))  {  SqlCommand cmd = new SqlCommand("select \* from family", con);  cmd.Connection = con;  con.Open();  SqlDataReader rdr = cmd.ExecuteReader();  while (rdr.Read())  {  Family family = new Family();  family.name = rdr["Name"].ToString();  family.gender = rdr["Gender"].ToString();  family.age = rdr["Age"].ToString();  family.relation = rdr["Relation"].ToString();  family.dob = rdr["Dob"].ToString();  objFamilylist.Add(family);  }  JavaScriptSerializer js = new JavaScriptSerializer(); return js.Serialize(objFamilylist);  }  }  }  } |

**Web.config**:-

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <!--  For more information on how to configure your ASP.NET application, please visit  https://go.microsoft.com/fwlink/?LinkId=169433  -->  <configuration>  <system.web>  <compilation debug="true" targetFramework="4.7.1"/>  <httpRuntime targetFramework="4.7.1"/>  </system.web>  <connectionStrings>  <add name="DBContext" providerName="System.Data.SqlClient" connectionString="Data Source=DESKTOP-8F3RQJO\SQLEXPRESS01;Initial Catalog=family;Integrated Security=True "/>  </connectionStrings>  <system.serviceModel>  <bindings>  <basicHttpBinding>  <binding name="WebService1Soap" />  </basicHttpBinding>  </bindings>  <client>  <endpoint address="http://localhost:58555/WebService1.asmx"  binding="basicHttpBinding"  bindingConfiguration="WebService1Soap"  contract="ServiceReference1.WebService1Soap"  name="WebService1Soap" />  </client>  </system.serviceModel>  </configuration> |

****



**Now adding Webform in the same solution explorer .**

**(adding newtonsoft.json)**

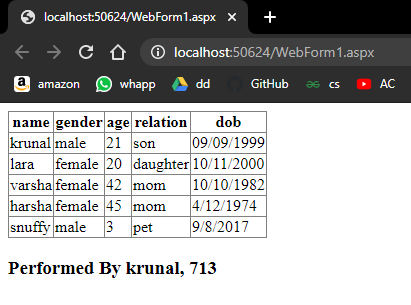
**Webform1.aspx.cs:**

|  |
| --- |
| <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="prac7.WebForm1" %>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml">  <head runat="server">  <title></title>  </head>  <body>  <form id="form1" runat="server">  <div>  <asp:GridView runat="server" ID="GridView"  OnSelectedIndexChanged="GridView\_SelectedIndexChanged"></asp:GridView>  </div>  <h3>Performed By krunal, 713</h3>  </form>  </body>  </html> |

Webform.aspx:

|  |
| --- |
| <%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="prac7.WebForm1" %>  <!DOCTYPE html>  <html xmlns="http://www.w3.org/1999/xhtml">  <head runat="server">  <title></title>  </head>  <body>  <form id="form1" runat="server">  <div>  <asp:GridView runat="server" ID="GridView"  OnSelectedIndexChanged="GridView\_SelectedIndexChanged"></asp:GridView>  </div>  <h3>Performed By krunal, 713</h3>  </form>  </body>  </html> |

**Output:-**

****

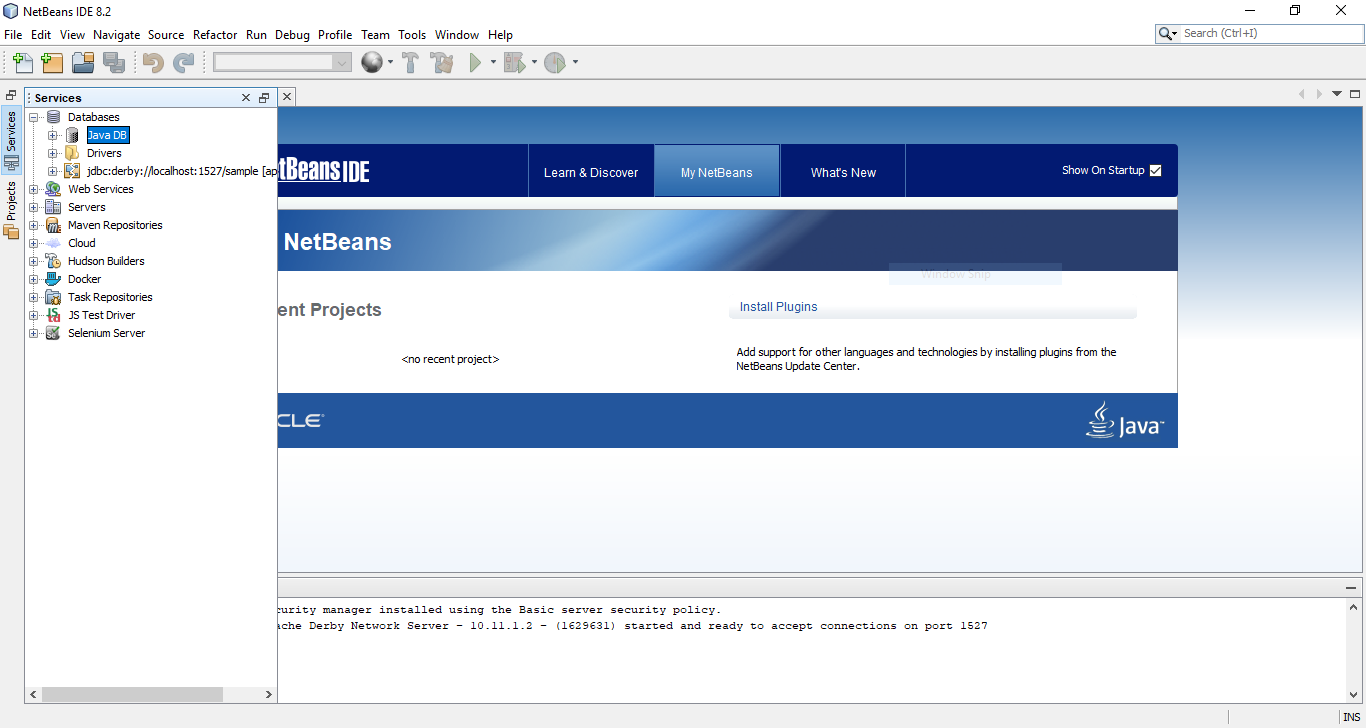
Date: 28/10/2020

**Practical no 8**

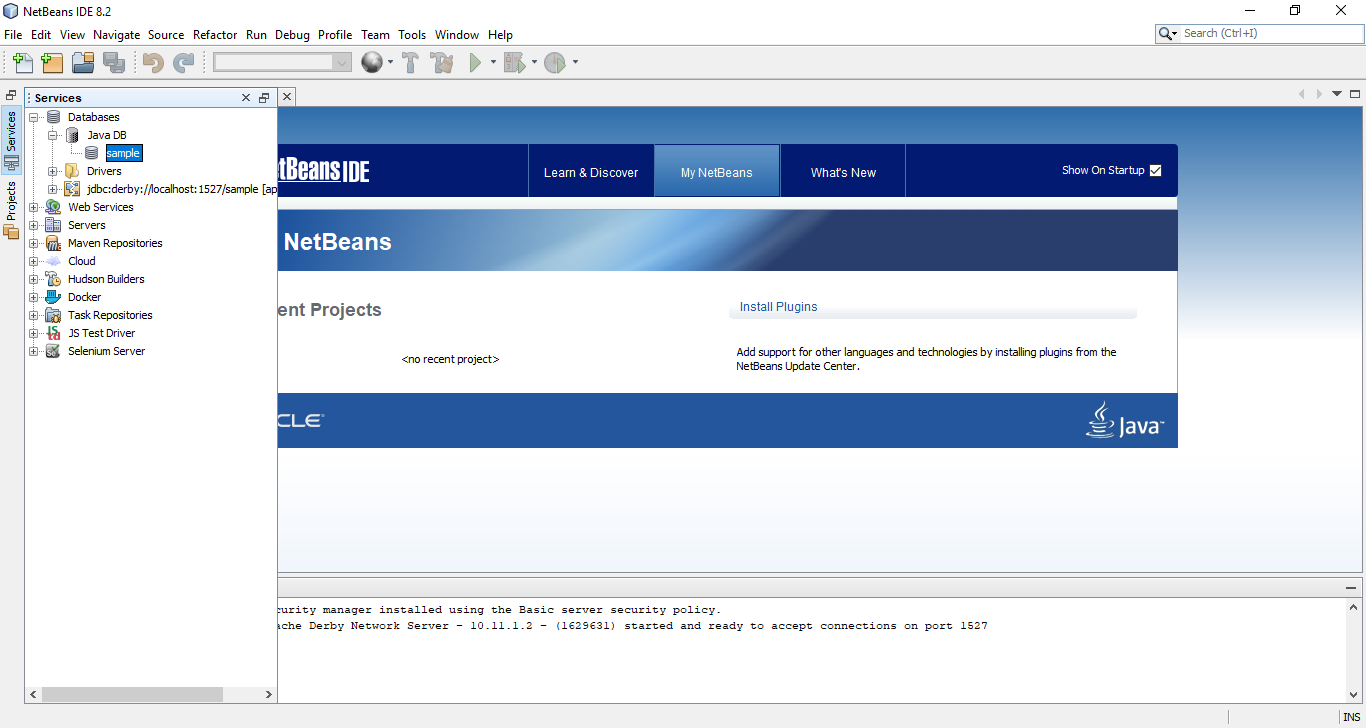
**AIM**: Define a RESTful web service that accepts the details to be stored in a database and performs CRUD operation**.**

**Steps :-**

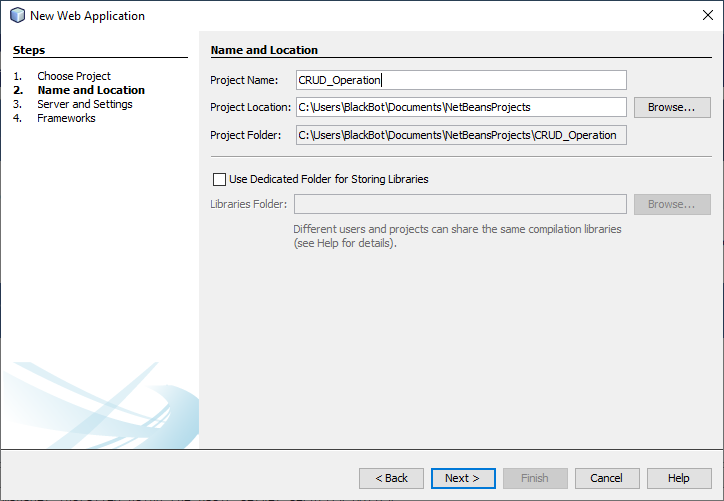
1. **Right click on Java DB** and then **click on Start Server** to start the server.

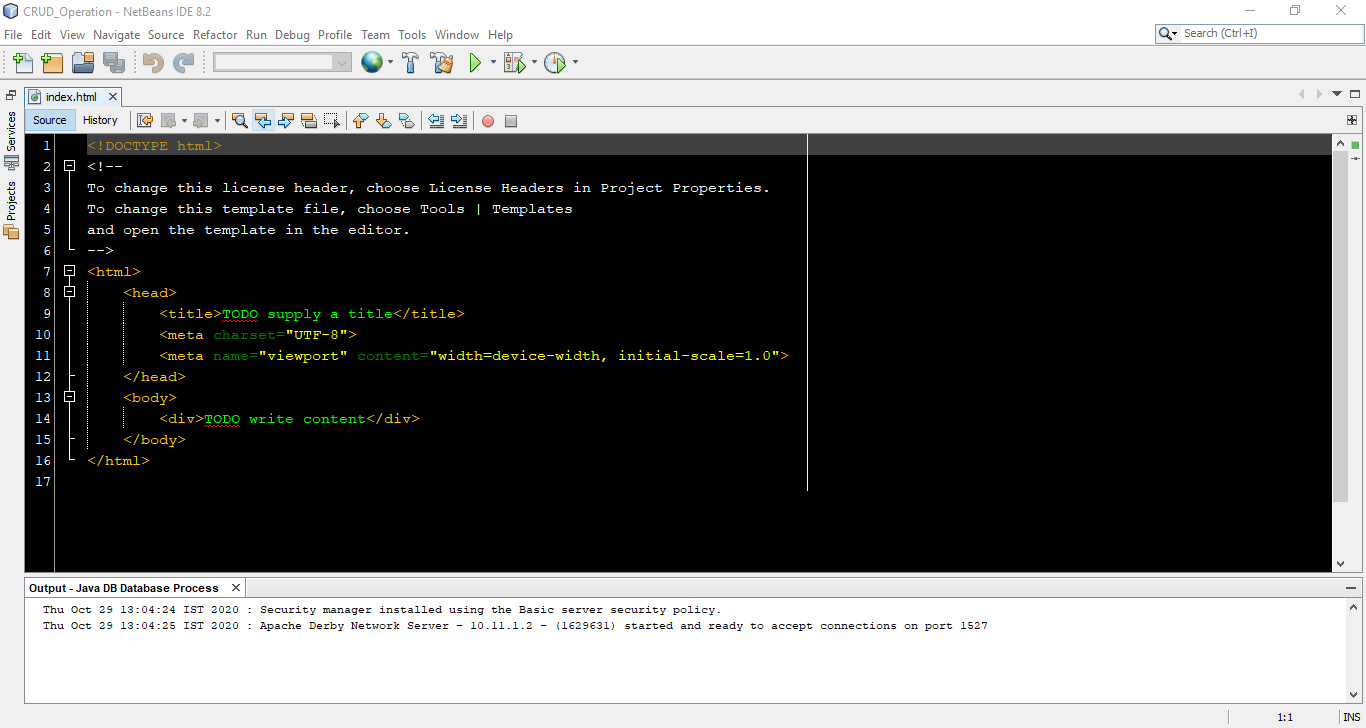


**2)** Now expand Java DB and **right click on sample** and then **click on connect** to connect the sample database with server.

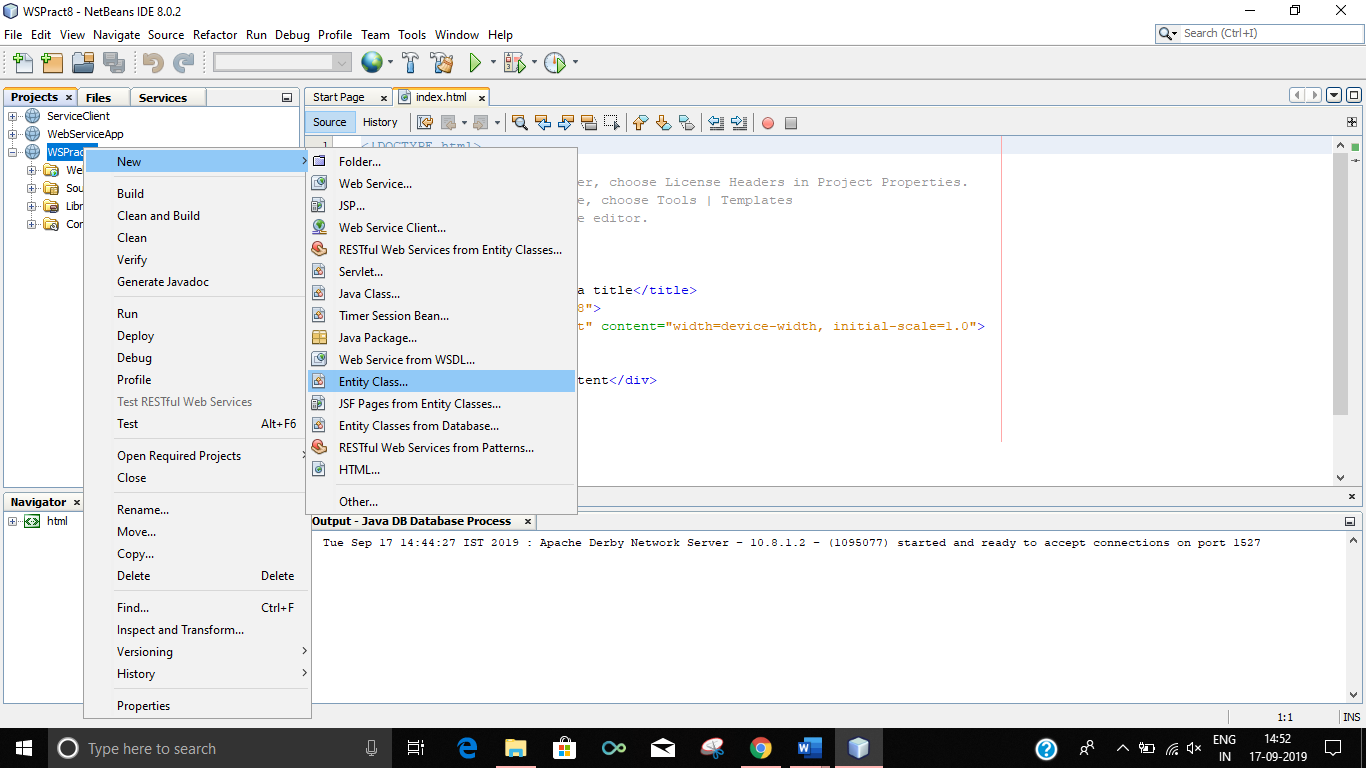


**3)** Now create a web application with the name **CRUD\_Operation**. A window will open like following pic.





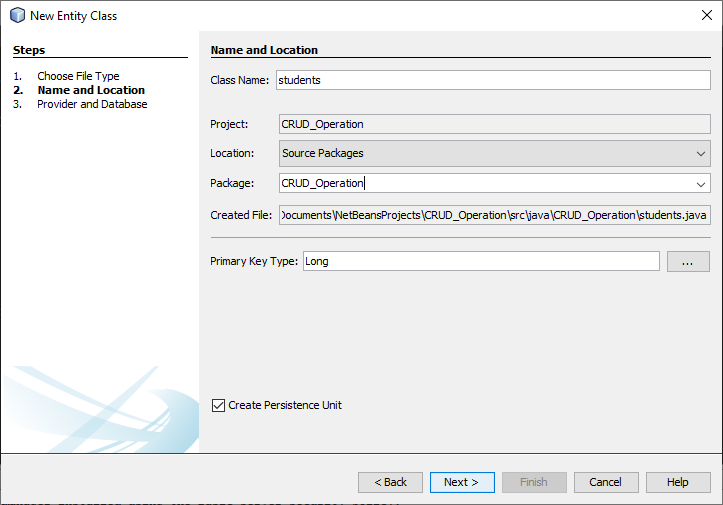
**4)** Create an entity class. **Right click on project name -> New -> Entity Class**.



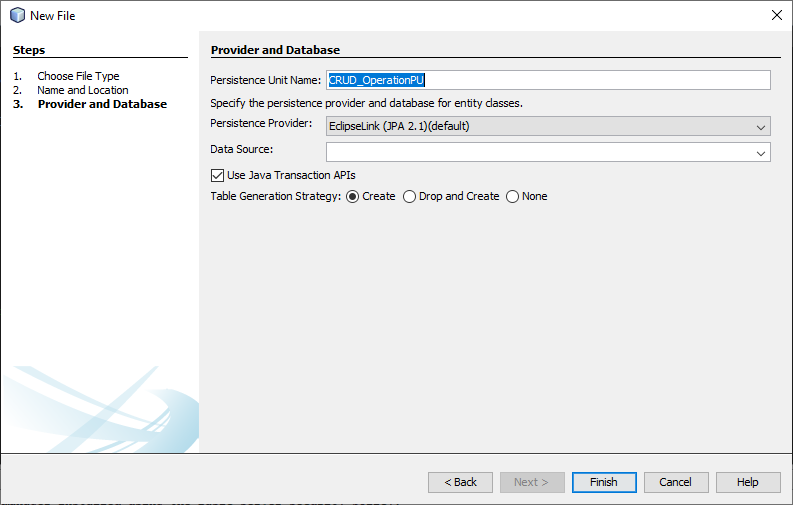
**5)** A window will appear like bellow pic. Enter following data and click on Next

**Class Name : students**

**Package name : CRUD\_Operation**

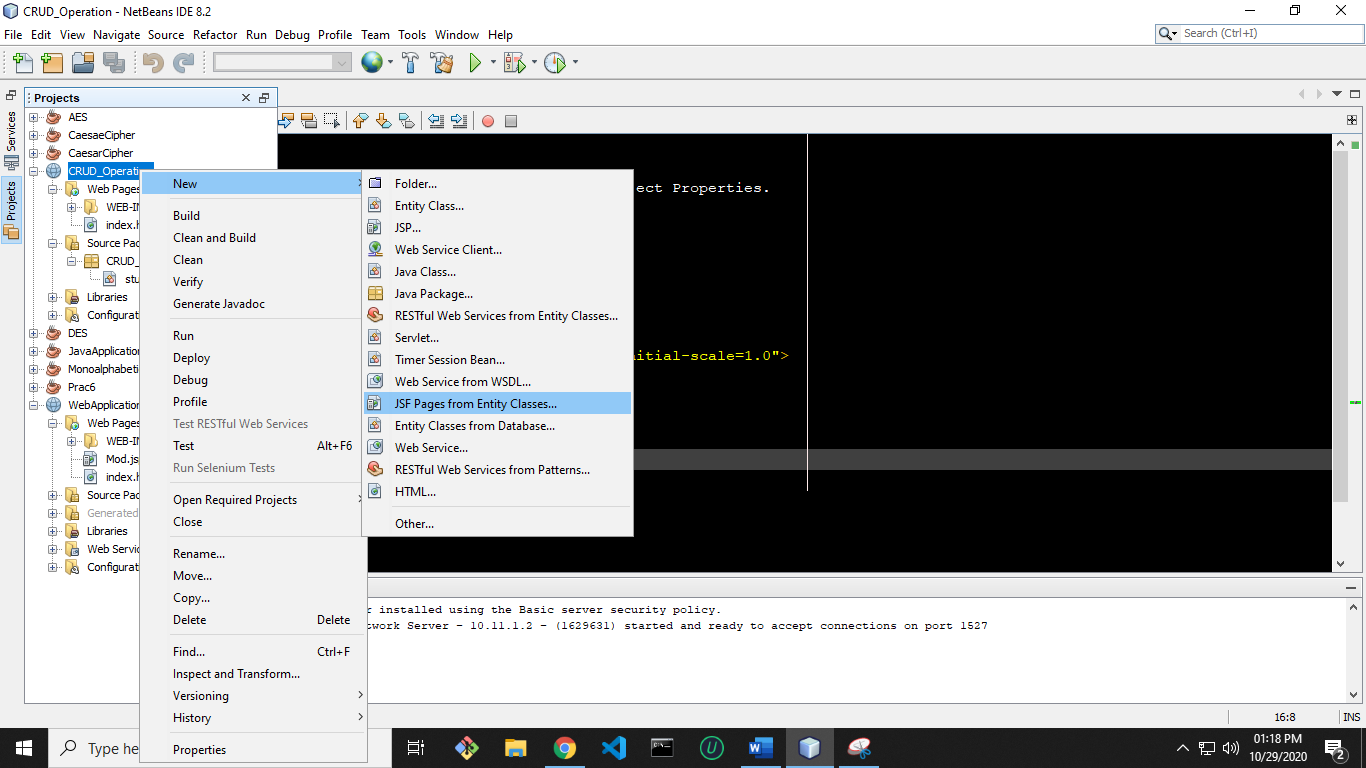


**6)** Click on Finish.

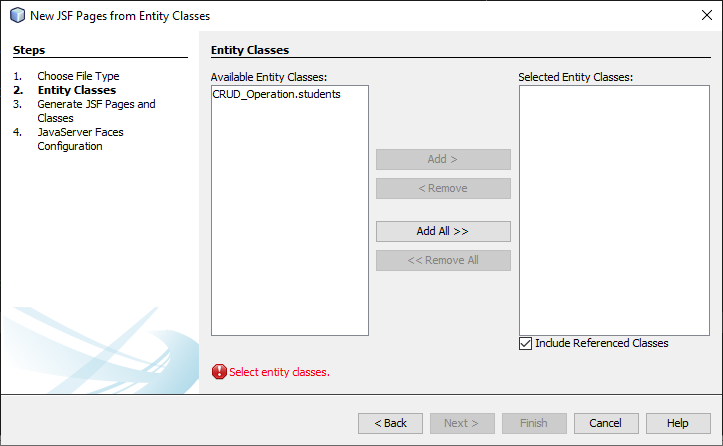


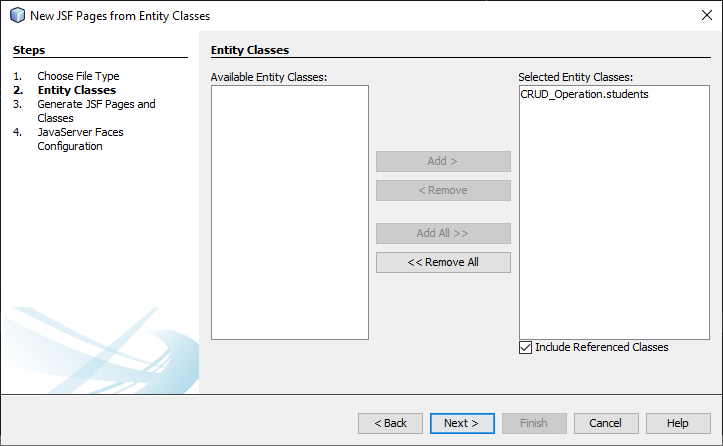
**7)** Right click on project name and create JSF Pages from Entity Classes.

**Right click on project name -> New -> JSF Pages from Entity Classes**

****

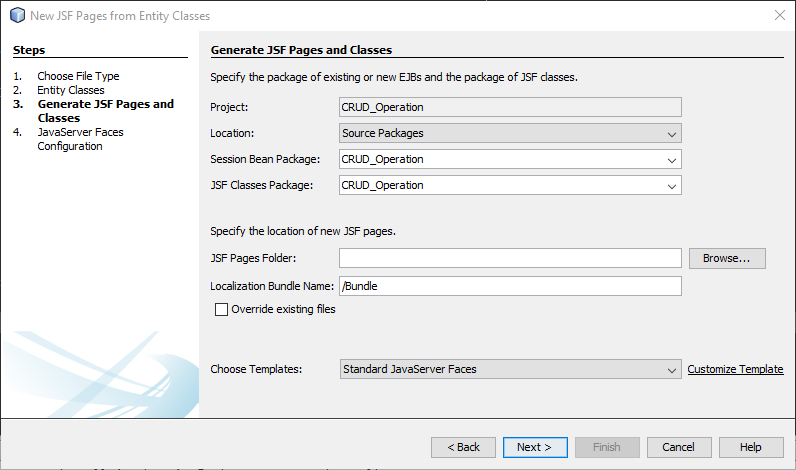
**8)** **Select wspract8.students** and **click on Add button and then Next button** on below.

****

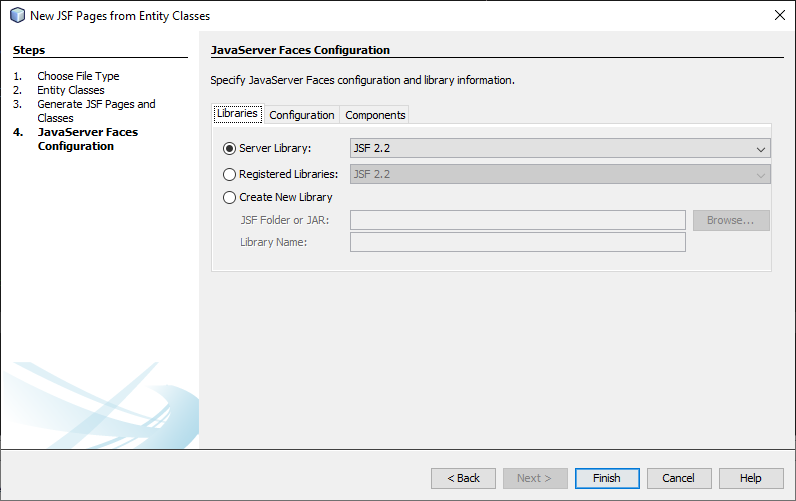


**9)**A window like below will appear on the screen. **Enter the data into that** **window as**

**entered in below pic** and **click on Next button**

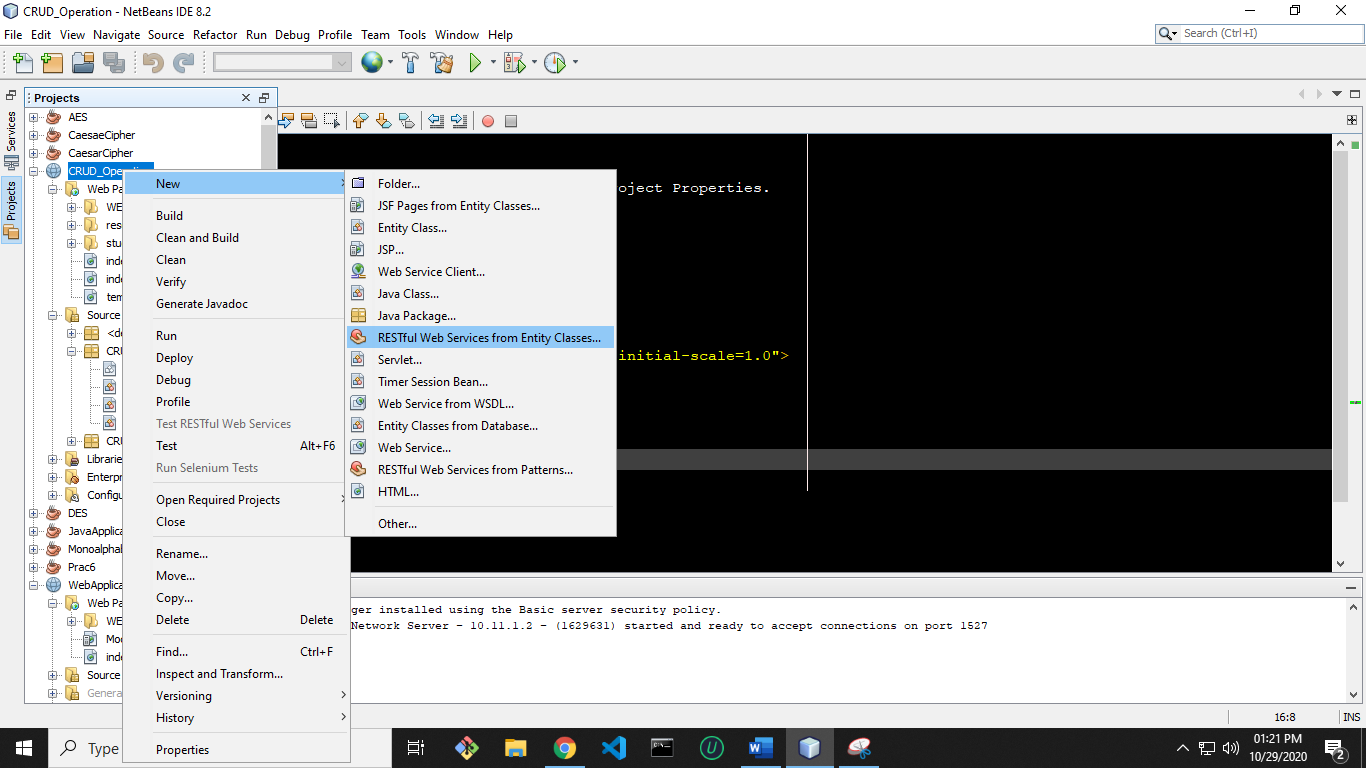
****

**10**)Now **click on Finish**

****

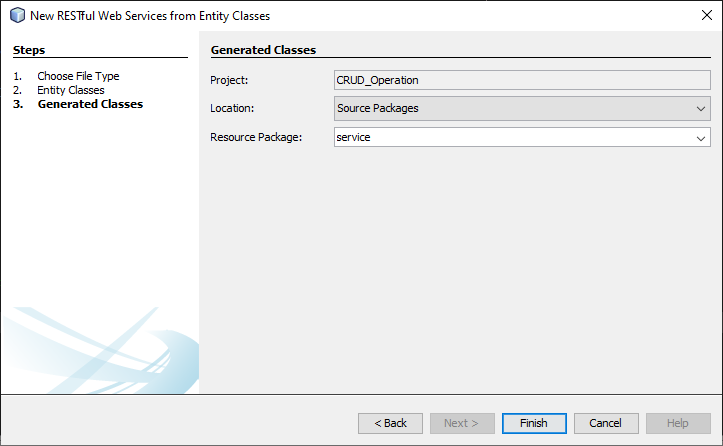
**11)** Right click on project name and create RESTful Web Services from Entity Classes.

**Right click on project name -> New -> RESTful Web Services from Entity Classes**

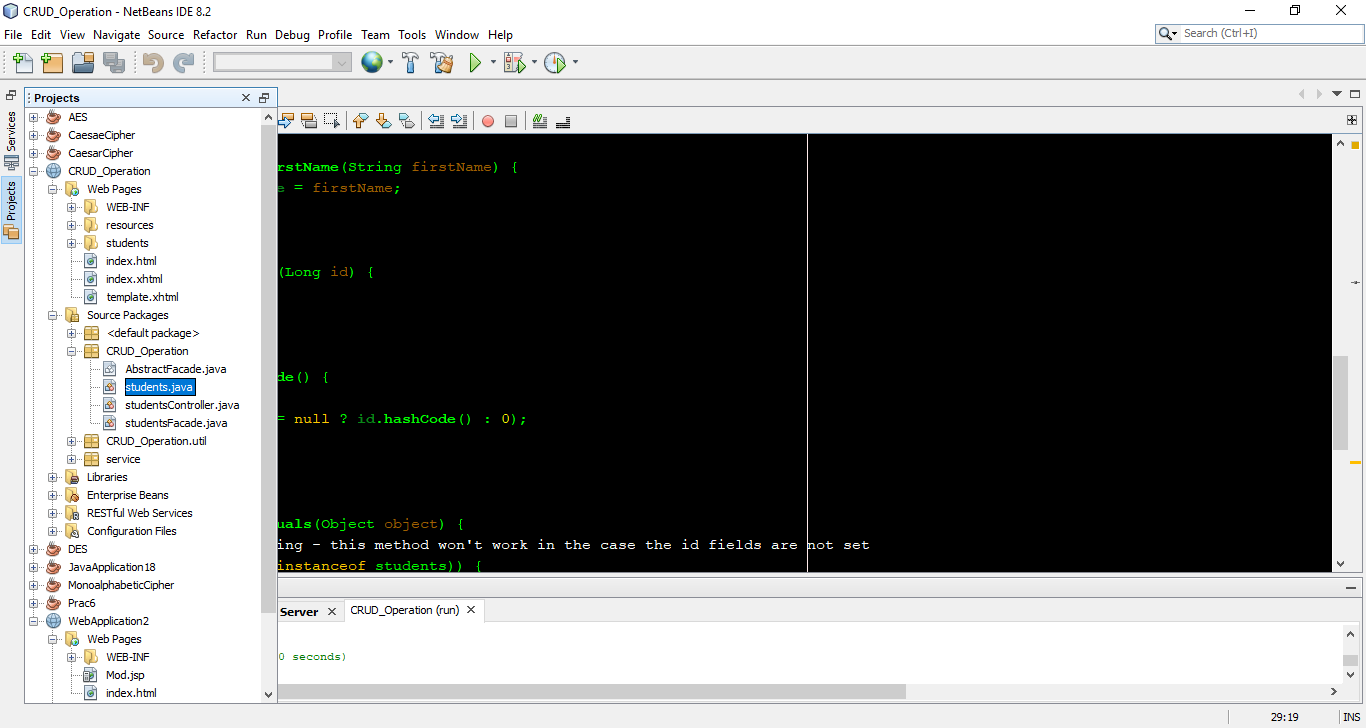


**12) Repeat step 9** and then it will go on next page. Then **enter the service** in

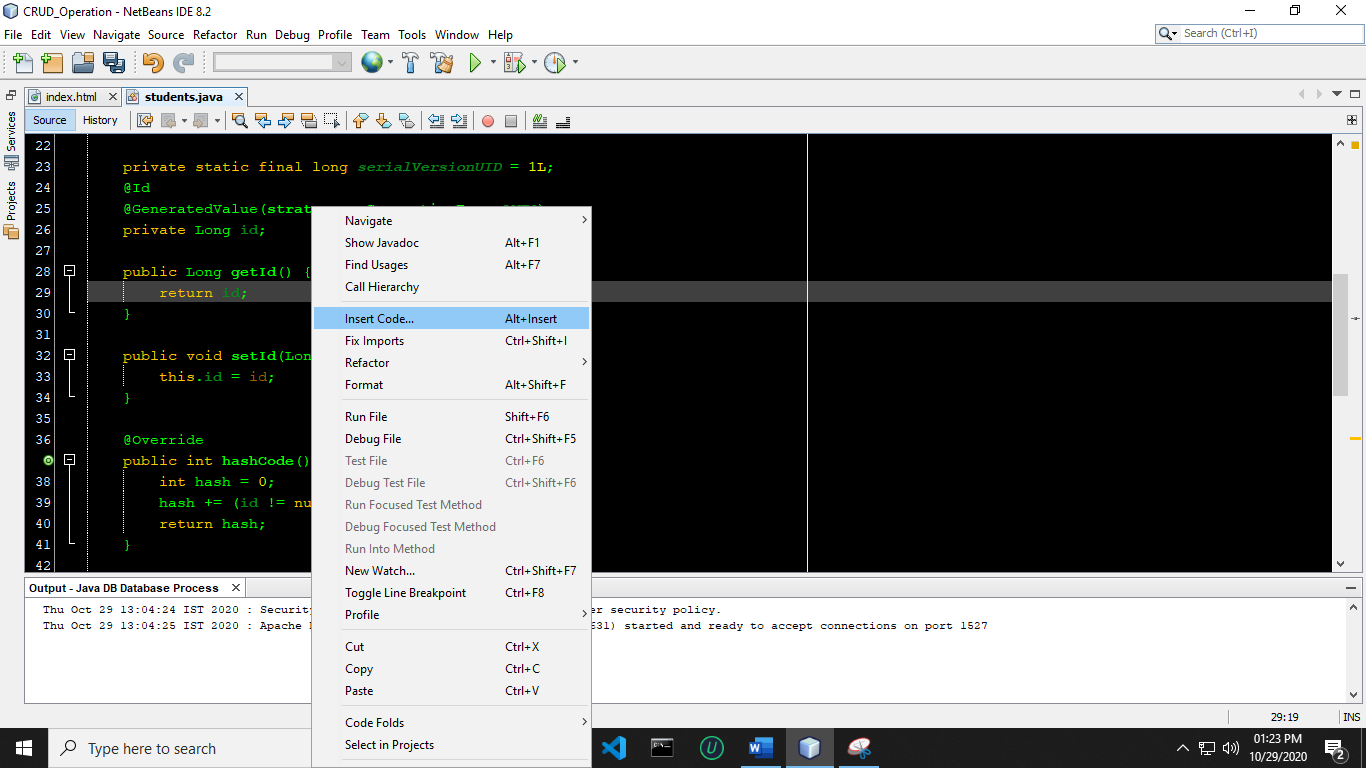
Resource Package and then **click on Finish button.**

****

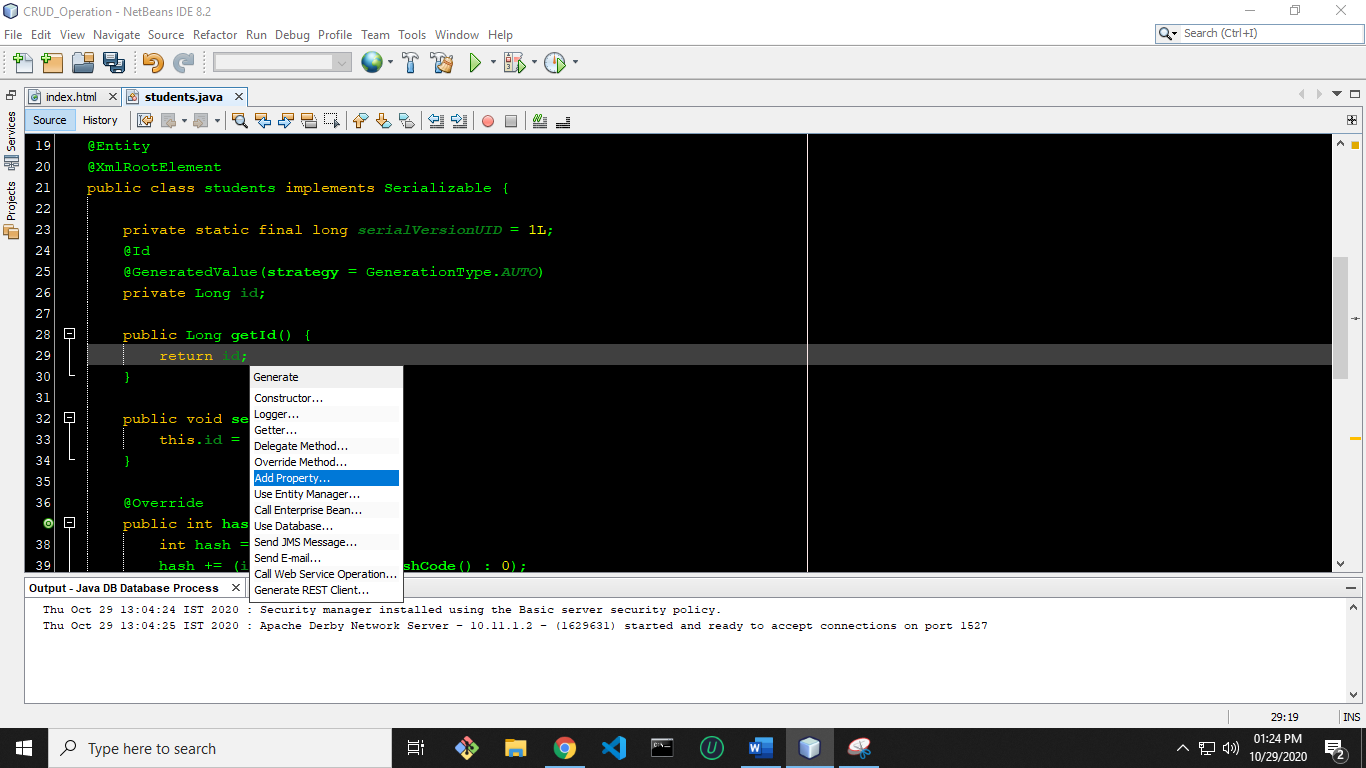
**13)** Now **open students.java file under wspract8 package.**



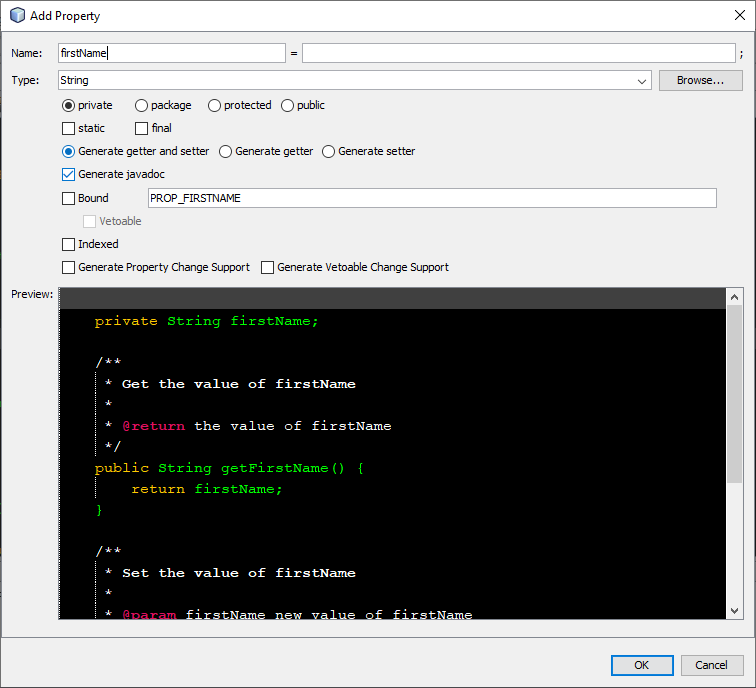
**14)** In this file **below line private Long id;** do the **right click and select Insert Code**.



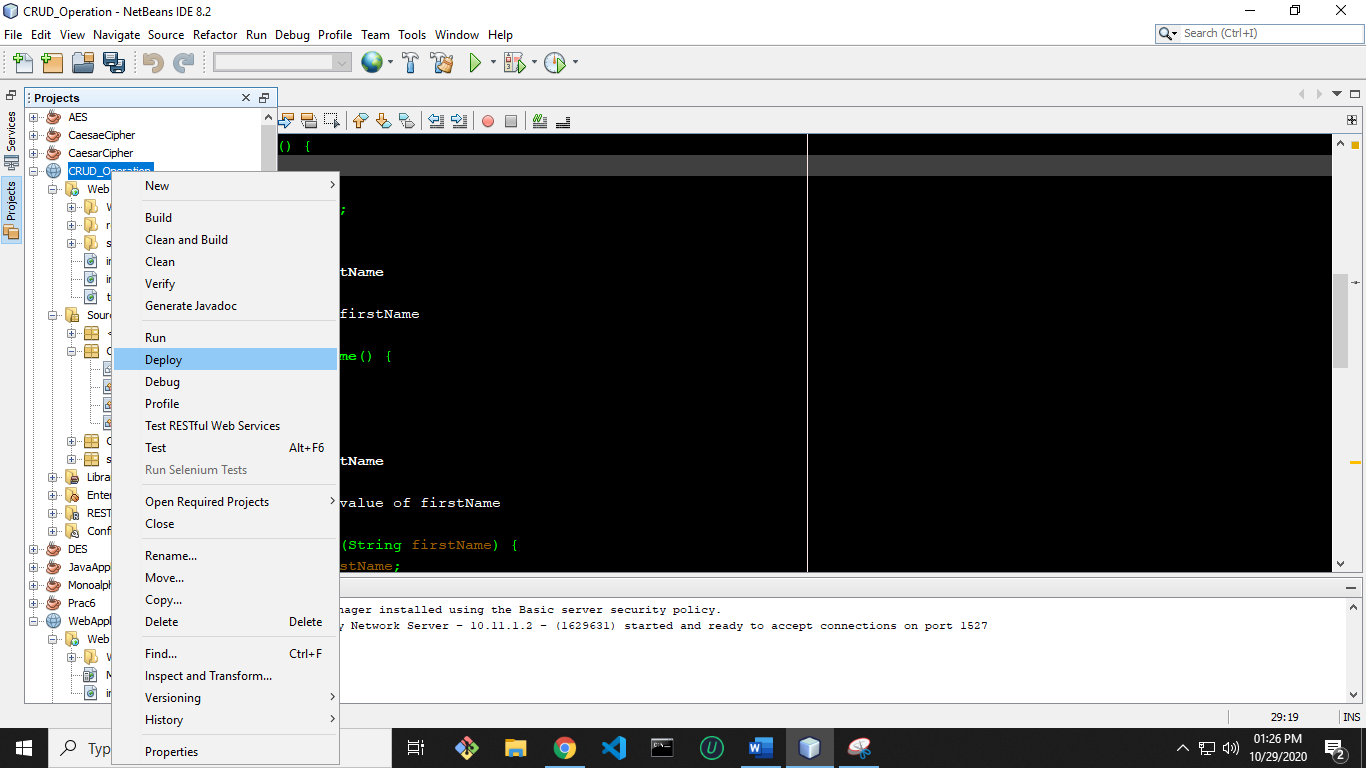
**15)** A new list will appear. **Click on Add Property**.



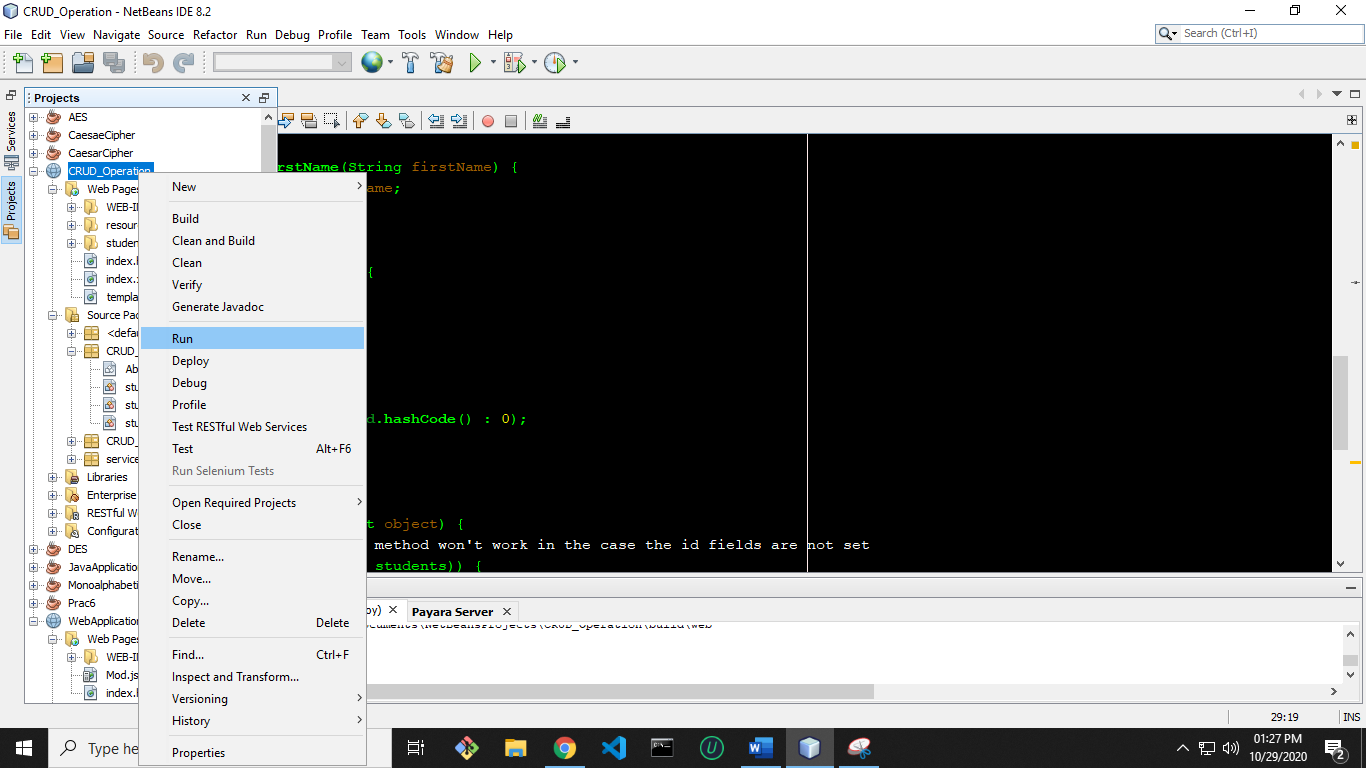
**16)** A new window will open. Enter name as **firstName**. Make sure name should be exact same as of mine and then **click on OK button**. Actually we are setting getter and setter method for firstName.

****

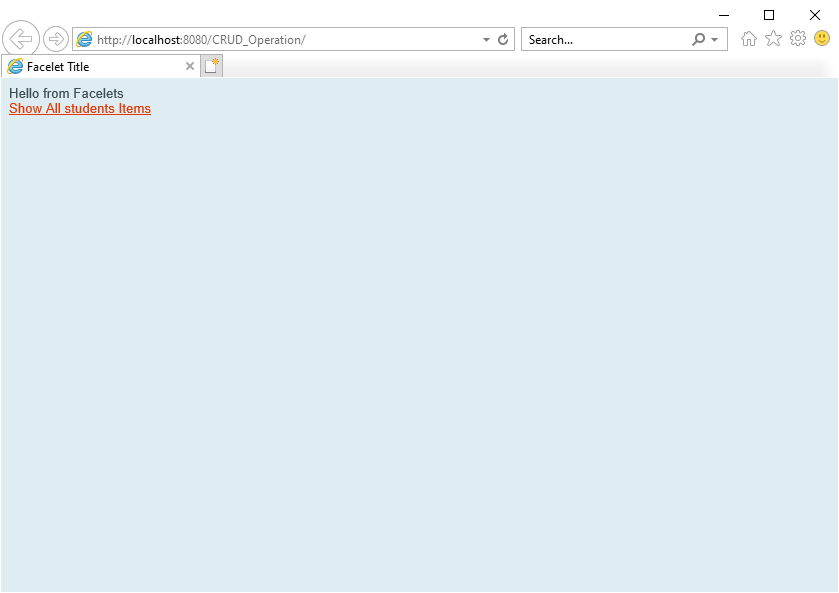
**17)** Now **right click on web application name and Deploy** it.



**18)** Now **right click on project name** and **run** it.



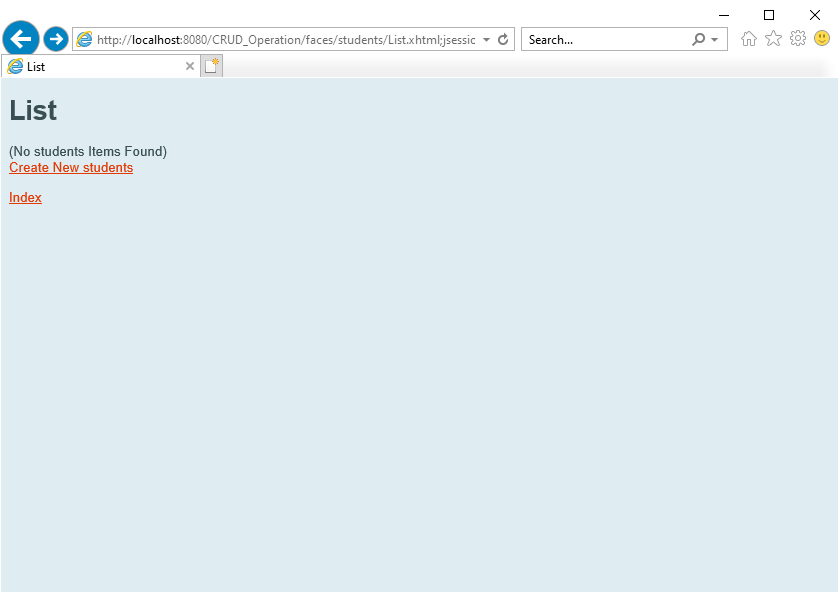
**19)** A window will open in browser like below.

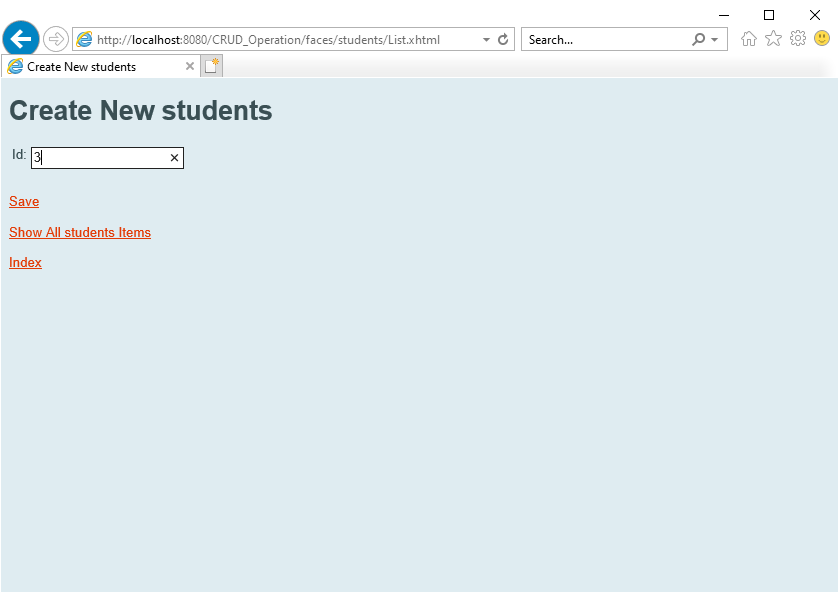


**20)** Now **click on Show All students** Items for CRUD operation.

Just click on Create New students. **Enter a name into FirstName and id into Id**. Now **click**

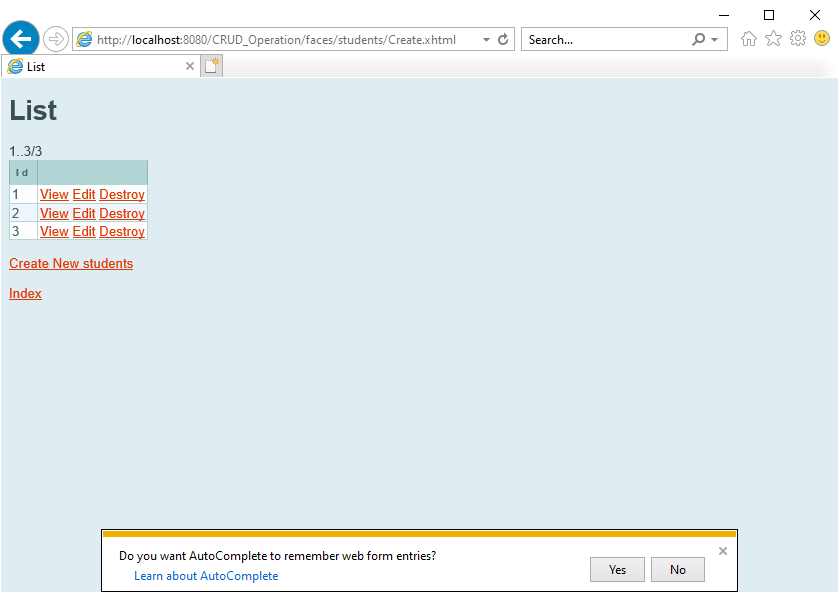
**on Save** option to savethe data.



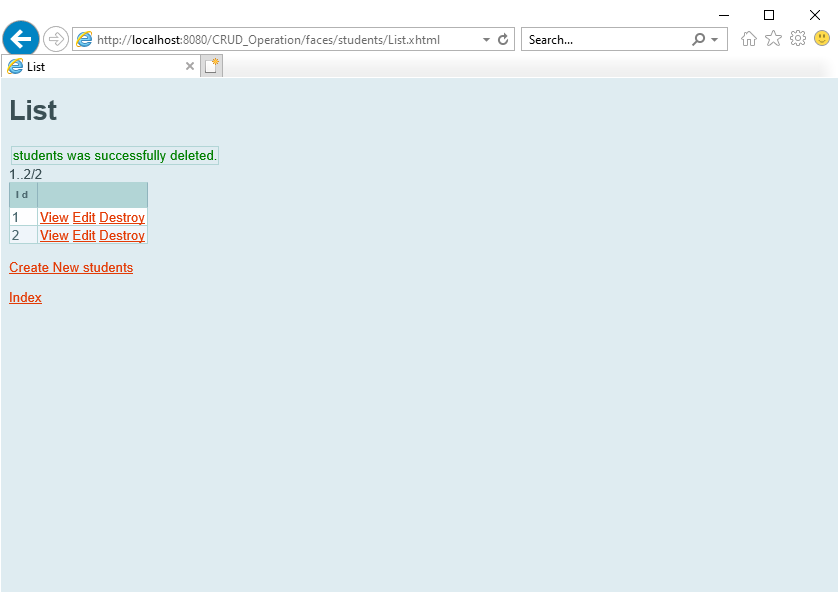


**21)** Now **click on Show All students Items to view all records** whether our data is

entered or not.



**22)** **Delete the Record**

****

Date: 18/11/2020

**Practical no 9**

**AIM**: Write a JAX-WS web service to perform the following operations. Define a

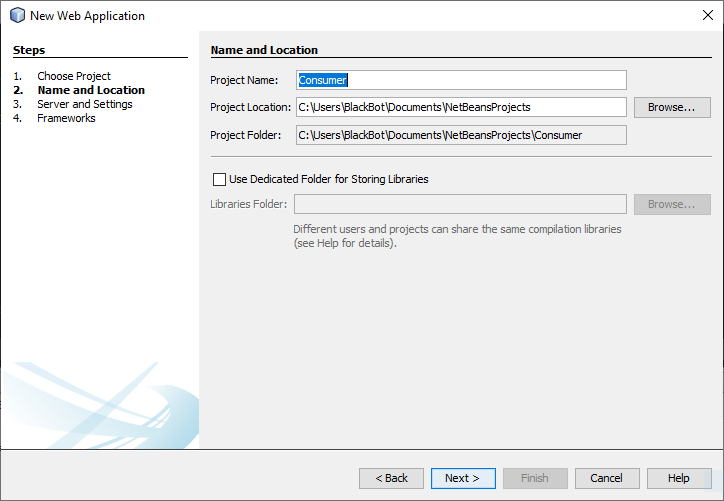
Servlet / JSP that consumes the web service.

**Note**: To do this practical, follow the steps from 1 to 18 present in practical – 7.

After that do not close or restart the NetBeans.

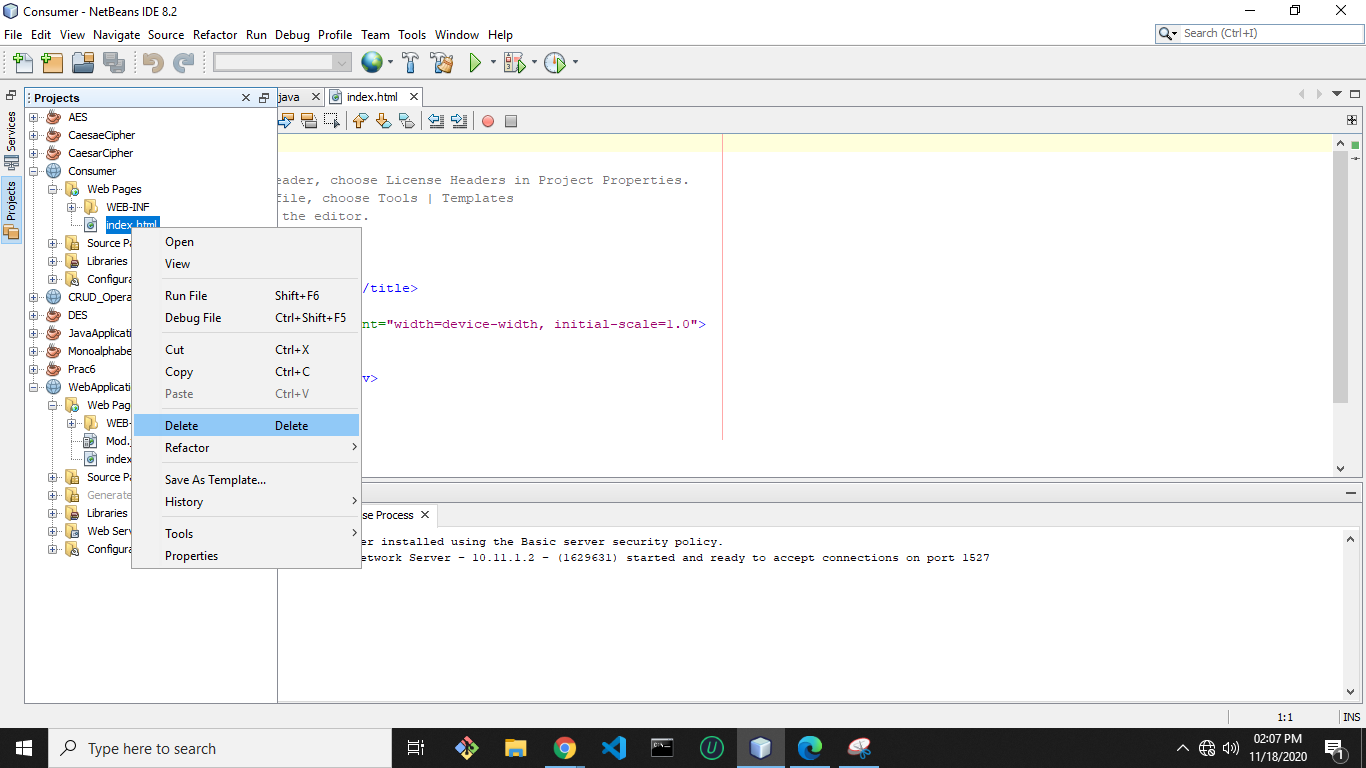
**Steps :-**

1. Create an another Web Application project and Give name as Consumer**.**

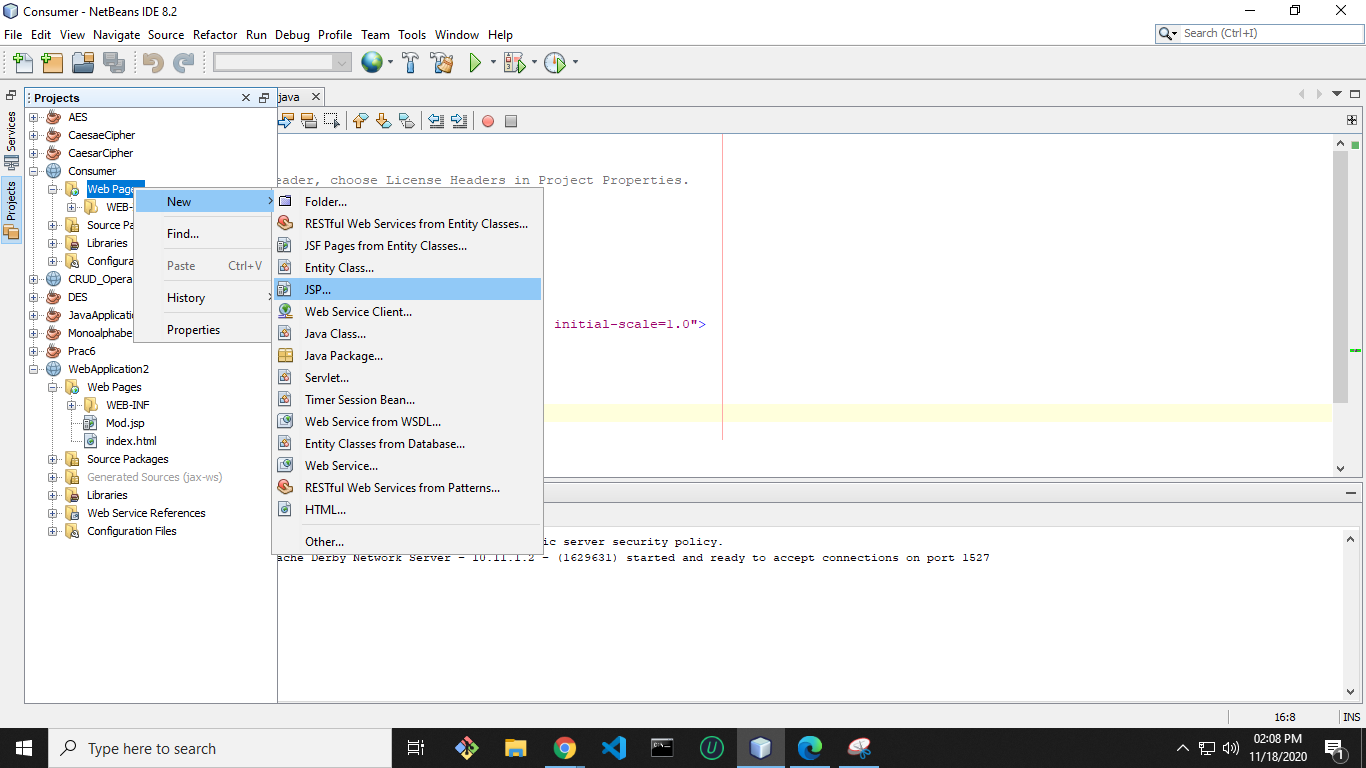
****

2) And create it. Next Finish.

3) Now right click on index.html and delete it.

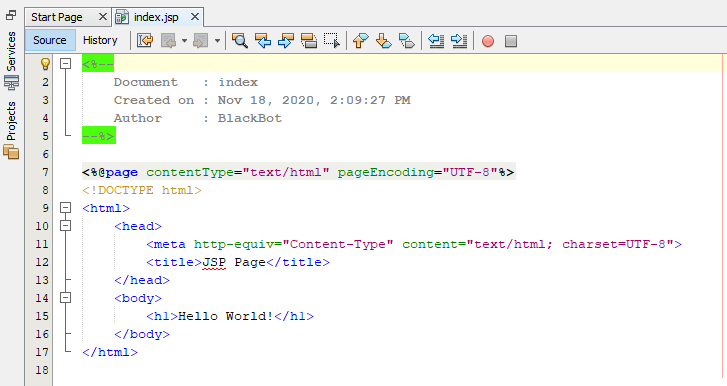


4) Now right click on Web Pages and select JSP to add a JSP page.



5) Give index name to it and then click on Finish.

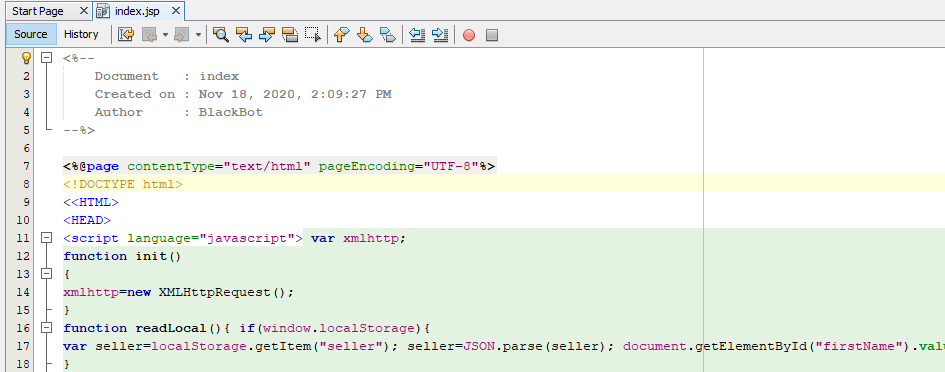
6) Now index.jsp page will open like below.



7) Now select HTML content of index.jsp file and replace it with following bold

letter codes.

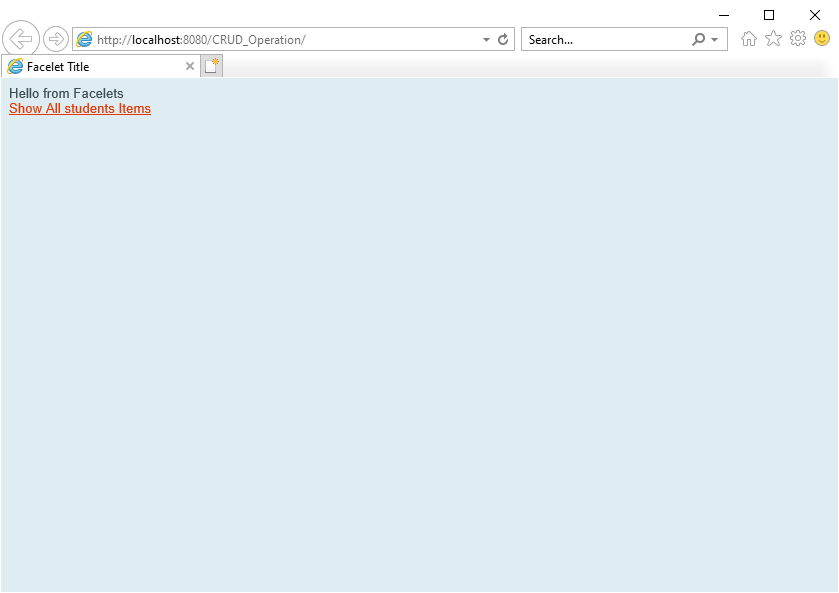
|  |
| --- |
| <%@page contentType="text/html" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <<HTML>  <HEAD>  <script language="javascript"> var xmlhttp;  function init()  {  xmlhttp=new XMLHttpRequest();  }  function readLocal(){ if(window.localStorage){  var seller=localStorage.getItem("seller");  seller=JSON.parse(seller);  document.getElementById("firstName").value=seller.firstName;  document.getElementById("sellerid").value=seller.id;  }  }  function saveLocal()  {  var sellerid=document.getElementById("sellerid");  var url="http://localhost:8080/CRUD\_Operation/webresources/com.kk.seller/"+ sellerid.value;  xmlhttp.open('GET',url,true); xmlhttp.send(null);  xmlhttp.onreadystatechange =function(){  if(xmlhttp.readyState===4){alert("6"+sellerid);  if(xmlhttp.status===200){alert("7"+sellerid);  var seller =eval("("+xmlhttp.responseText+")");  if(window.localStorage){  localStorage.setItem("seller",JSON.stringify(seller));  alert("information stored successfully"+seller.firstName);  }  else{ alert("notstored");  }}  else  alert("error");  }  }  }  </script>  </head>  <body onLoad ="init()">  <table>  <tr>  <td>Enter id:</td>  <td><input type="text" id="sellerid"/>  <input type="button" value="load employee in local browser" onClick="saveLocal()"/>  </td>  </tr>  <tr>  <td>read from local</td>  <td><input type="button" value="Send values" onClick="readLocal()"/></td>  </tr>  <tr>  <td>first Name:</td>  <td> <input type="text" id="firstName"/></td>  </tr>  <tr>  performed by krunal 713  </tr>  </table>  </body>  </html> |



9) <http://localhost:8080/CRUD_Operation/webresources/com.kk.seller/>

Red part is the URL of which is obtained in browser by running of

CRUD\_Operation web application.



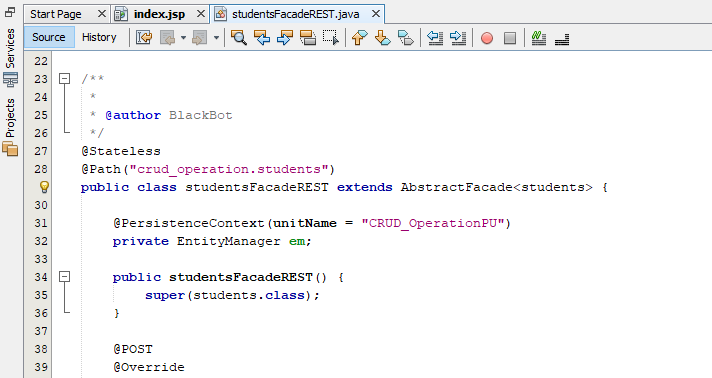
**Blue part** in above link is static. But red part is dynamic which is based on

practical 7. So if you have changed name anywhere then the above URL will

change accordingly.

10) Now open studentsFacadeREST.java file by follow below pic available in

CRUD\_Operation web application.



11) Now delete the selected part. Because it will return data in XML format to the

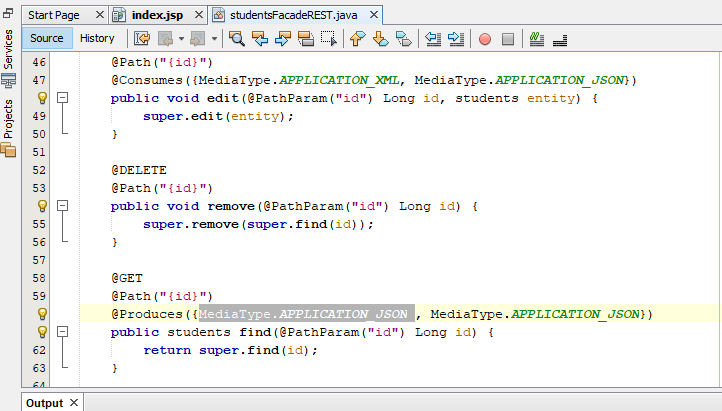
consumer.

But we have written javascript in index.jsp for JSON data format

only.

After that deploy the **CRUD\_Operation** web application; so that it will

update the changes.



12) Now run the Consumer application. A window will open in browser like

below.

13) Now if you will enter an id into Enter id textbox which you have created in

database (id from data in last step of practical 7) and then click on load

employee in local browser button. It will get the detail of particular entered id

and will store it into local storage of browser. Now click on Send Values to get

the first name of entered id.

