**Name – Ganesh Halpatrao**

**Class - TYBSc CS Roll no 2096**

**Subject - Webservices Practical (sub 4 )**

Practical

# Practical 1

## Q Simple SOAP Webservices To perform Athematic Operation and Java Client

ANS

Step 1 – Create WebApplication

1. Then take new Webservice
2. Code /\*

|  |
| --- |
| \* 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.  \*/  package abc;  import javax.jws.WebService;  import javax.jws.WebMethod;  import javax.jws.WebParam;  /\*\*  \*  \* @author admin  \*/  @WebService(serviceName = "AddWebServices")  public class AddWebServices {  /\*\*  \* This is a sample web service operation  \*/  @WebMethod(operationName = "addition")  public int addition (@WebParam(name = "a")int a,@WebParam(name = "b")int b ) {  return (a+b);  }  } |

1. Then Deploy webapplication
2. Then copy WSDL file
3. Then Create java Application
4. Then Add Webservice Client>then Copy paste url of wsdl >
5. Then code

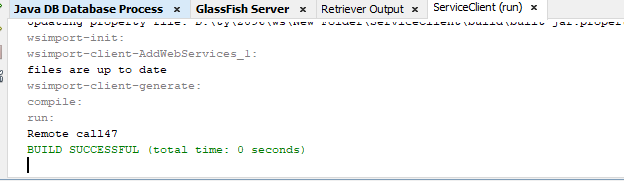
|  |
| --- |
| package serviceclient;  /\*\*  \*  \* @author admin  \*/  public class ServiceClient {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  System.out.println("Remote call"+ addition\_1(24,23));  } |

1. Then take for webservice refernces take
2. Addition code

|  |
| --- |
| private static int addition\_1(int a, int b) {  abc.AddWebServices\_Service service = new abc.AddWebServices\_Service();  abc.AddWebServices port = service.getAddWebServicesPort();  return port.addition(a, b); |

1. then run serviceClient

output



We have to do it for all

Like subtraction division multiplication

# Practical 2

## Q Create SOAP Webservices to find Factorial of a Number and Consume using Servlet client

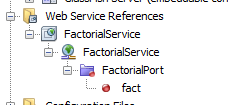
Solution :-

1. Create Web application
   1. Then add Web service
   2. Then write for the factorial of a number in web service
      1. Code

|  |
| --- |
| package abc;  import javax.jws.WebService;  import javax.jws.WebMethod;  import javax.jws.WebParam;  /\*\*  \*  \* @author admin  \*/  @WebService()  public class Factorial {  int x, fact=1;  @WebMethod(operationName = "fact")  public String fact1(@WebParam(name = "x") int x)  {  for (int y=1;y<=x;y++){  fact=fact\*y;  }  return "factorial of "+x +" is "+fact;  }  } |

1. Deploy and run the web service
2. Copy the WSDL file URL
3. Then create another web application
   1. Create a new webservice client
   2. Paste WSDL
   3. In index write the following code

|  |
| --- |
| <html>  <head>  <title>TODO supply a title</title>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width">  </head>  <body>  <form name="f1" method ="GET" action ="NewServlet">  <br>Enter 1st number<input type="text" name="t1">  <input type="Submit" value="submit">  </form>    </body>  </html> |

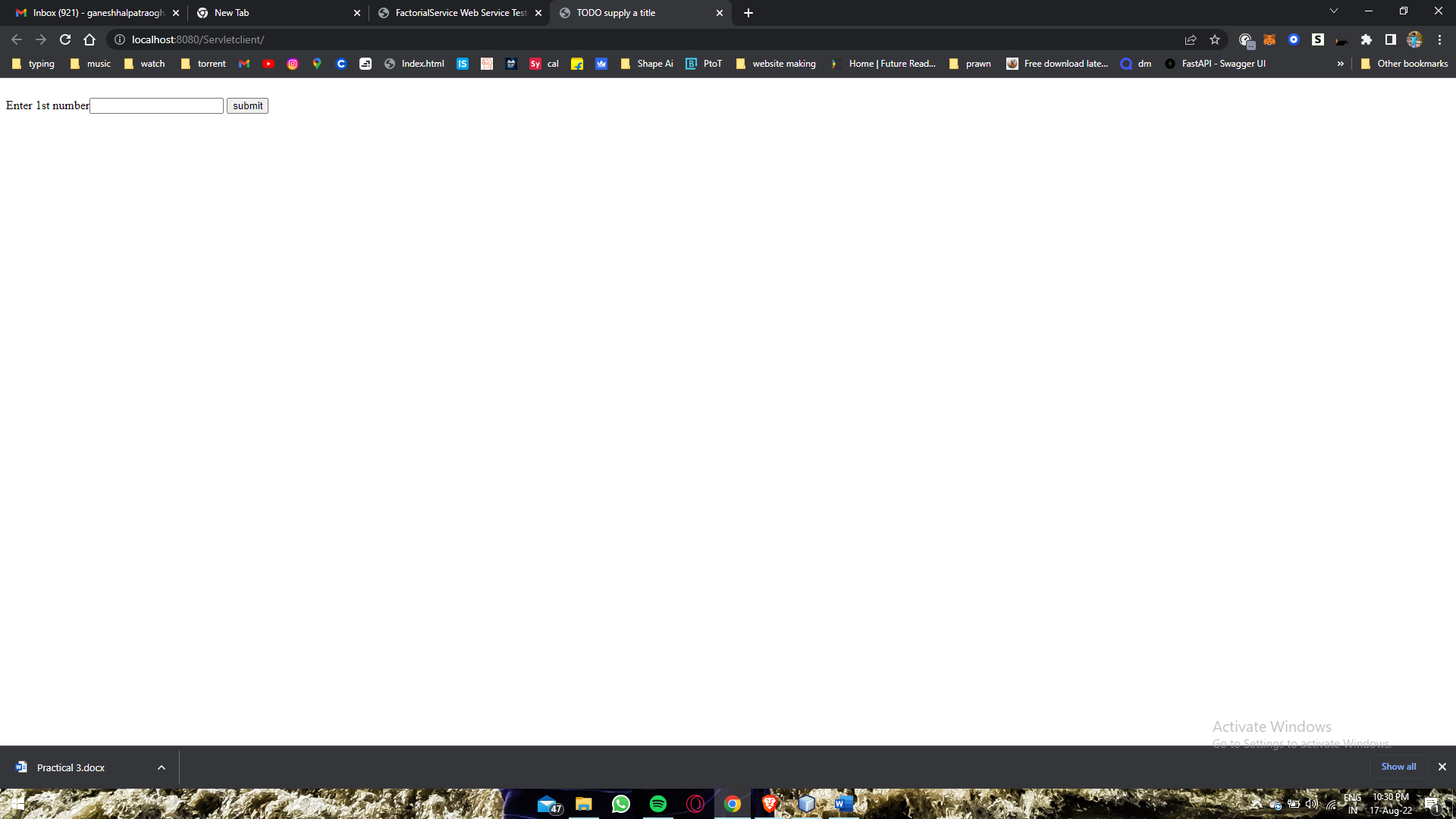
* 1. Then Create new Servlet file
  2. Code in servlet
  3. But drag and drop the from
  4.  after main method (so you get the



|  |
| --- |
| @WebServlet(name = "NewServlet", urlPatterns = {"/NewServlet"})  public class NewServlet extends HttpServlet {  @WebServiceRef(wsdlLocation = "WEB-INF/wsdl/localhost\_12667/WebApplication1/FactorialService.wsdl")  private abc.FactorialService service;  /\*\*  \* Processes requests for both HTTP <code>GET</code> and <code>POST</code>  \* methods.  \*  \* @param request servlet request  \* @param response servlet response  \* @throws ServletException if a servlet-specific error occurs  \* @throws IOException if an I/O error occurs  \*/  protected void processRequest(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  response.setContentType("text/html;charset=UTF-8");  int x = Integer.parseInt(request.getParameter("t1"));  try (PrintWriter out = response.getWriter()) {  /\* TODO output your page here. You may use following sample code. \*/  out.println("<!DOCTYPE html>");  out.println("<html>");  out.println("<head>");  out.println("<title>Servlet NewServlet</title>");  out.println("</head>");  out.println("<body>");  out.println("<h1>Servlet NewServlet at " + request.getContextPath() + "</h1>");  out.println("<h1>factorial= " + fact(x) + "</h1>");  out.println("</body>");  out.println("</html>");  }  }  // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">  /\*\*  \* Handles the HTTP <code>GET</code> method.  \*  \* @param request servlet request  \* @param response servlet response  \* @throws ServletException if a servlet-specific error occurs  \* @throws IOException if an I/O error occurs  \*/  @Override  protected void doGet(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  processRequest(request, response);  }  /\*\*  \* Handles the HTTP <code>POST</code> method.  \*  \* @param request servlet request  \* @param response servlet response  \* @throws ServletException if a servlet-specific error occurs  \* @throws IOException if an I/O error occurs  \*/  @Override  protected void doPost(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException {  processRequest(request, response);  }  /\*\*  \* Returns a short description of the servlet.  \*  \* @return a String containing servlet description  \*/  @Override  public String getServletInfo() {  return "Short description";  }// </editor-fold>  private String fact(int x) {  // Note that the injected javax.xml.ws.Service reference as well as port objects are not thread safe.  // If the calling of port operations may lead to race condition some synchronization is required.  abc.Factorial port = service.getFactorialPort();  return port.fact(x);  }  } |

Then run the whole application

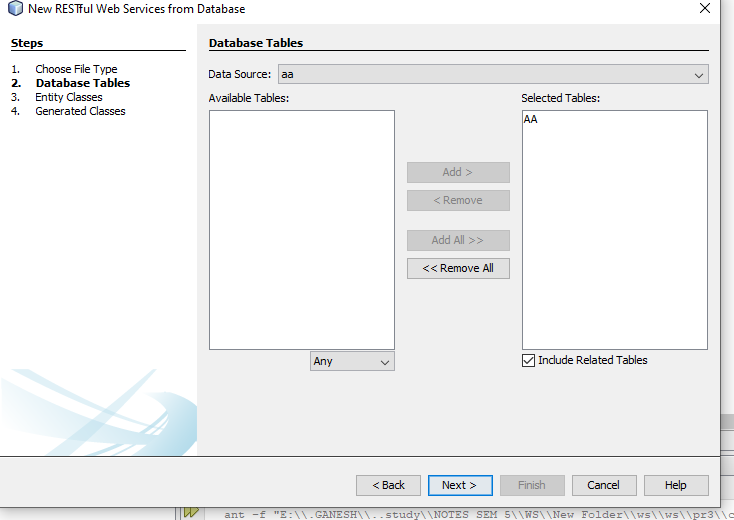
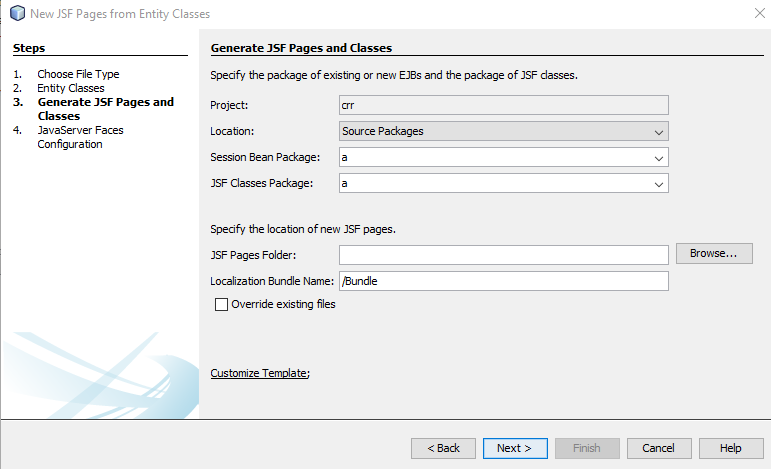
Output ;



# Practical 3

## Q Create restful services to perform CRUD Operation

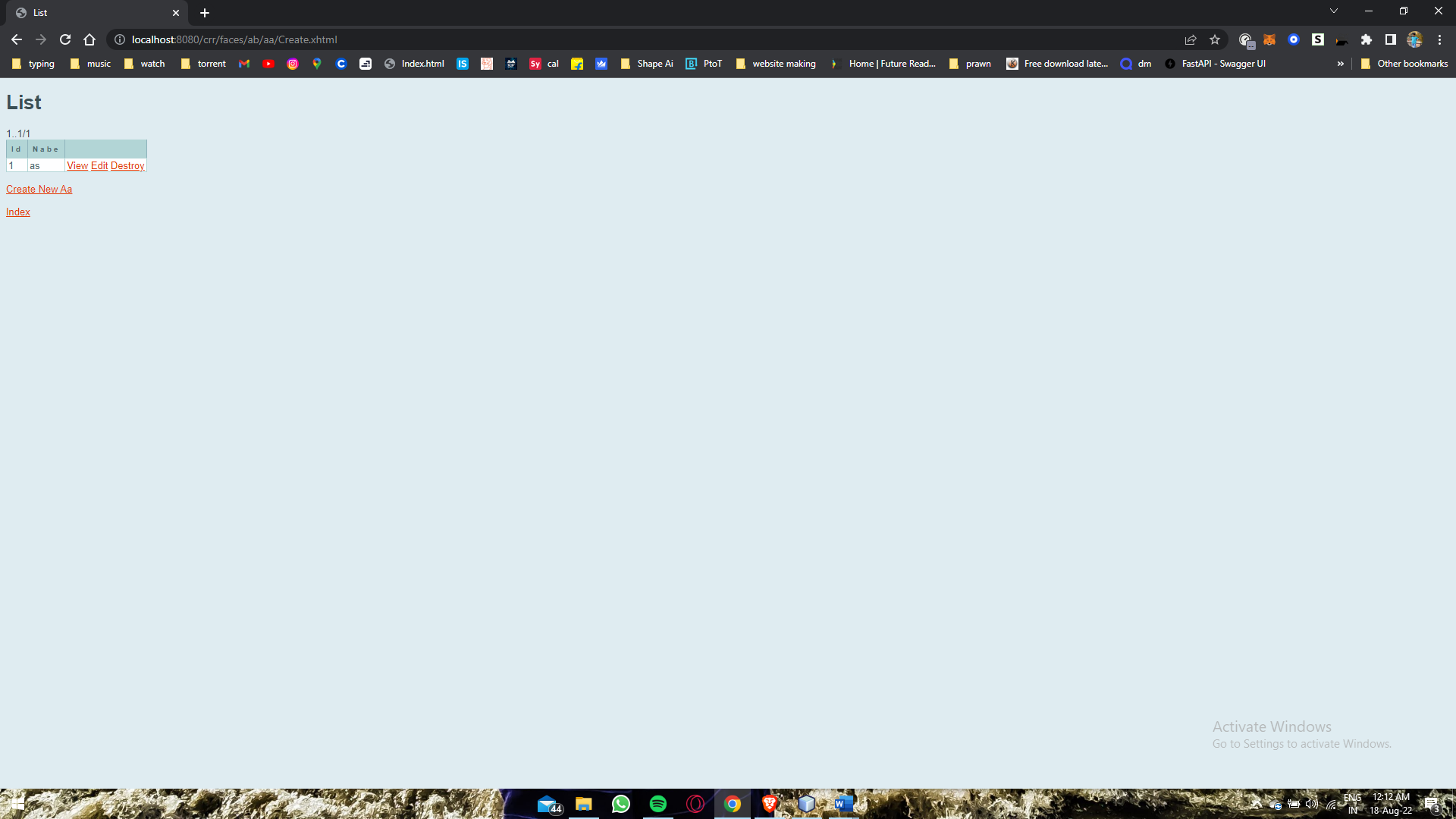
Solution

1. Create a Database
   1. Go TO services window
   2. Then create a new database
      1. Then add table
         1. Add columns (among one should having primary key )
         2. Then connect the database
2. Then Create Restful Web services From database
   1. Then give data source (give the same name as database )
      1. Then add tables from available 
      2. Then next and give package name and finish
3. Then create Entity class from database(same as above)
4. Then create JSF Page from entity classes
   1. Add entity classes
   2. Then give name to every packages
   3. 



* 1. Then clean and build

1. Then run
2. Output

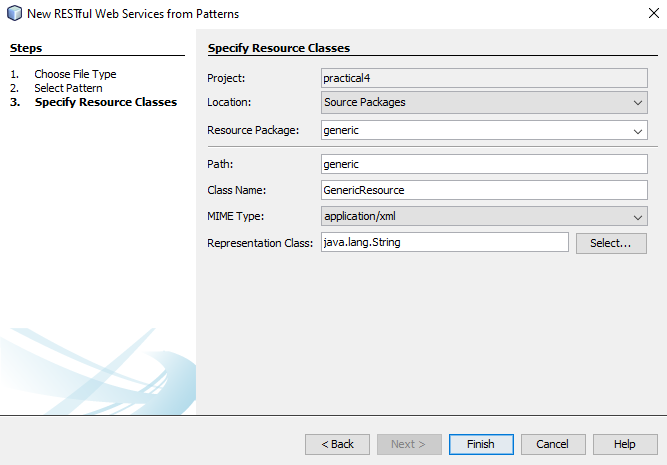


# Practical 4

## Q. Create Restful Services from pattern to Return Current date and time

Solution

1. Create web application
2. Then create a restful web services from patterns

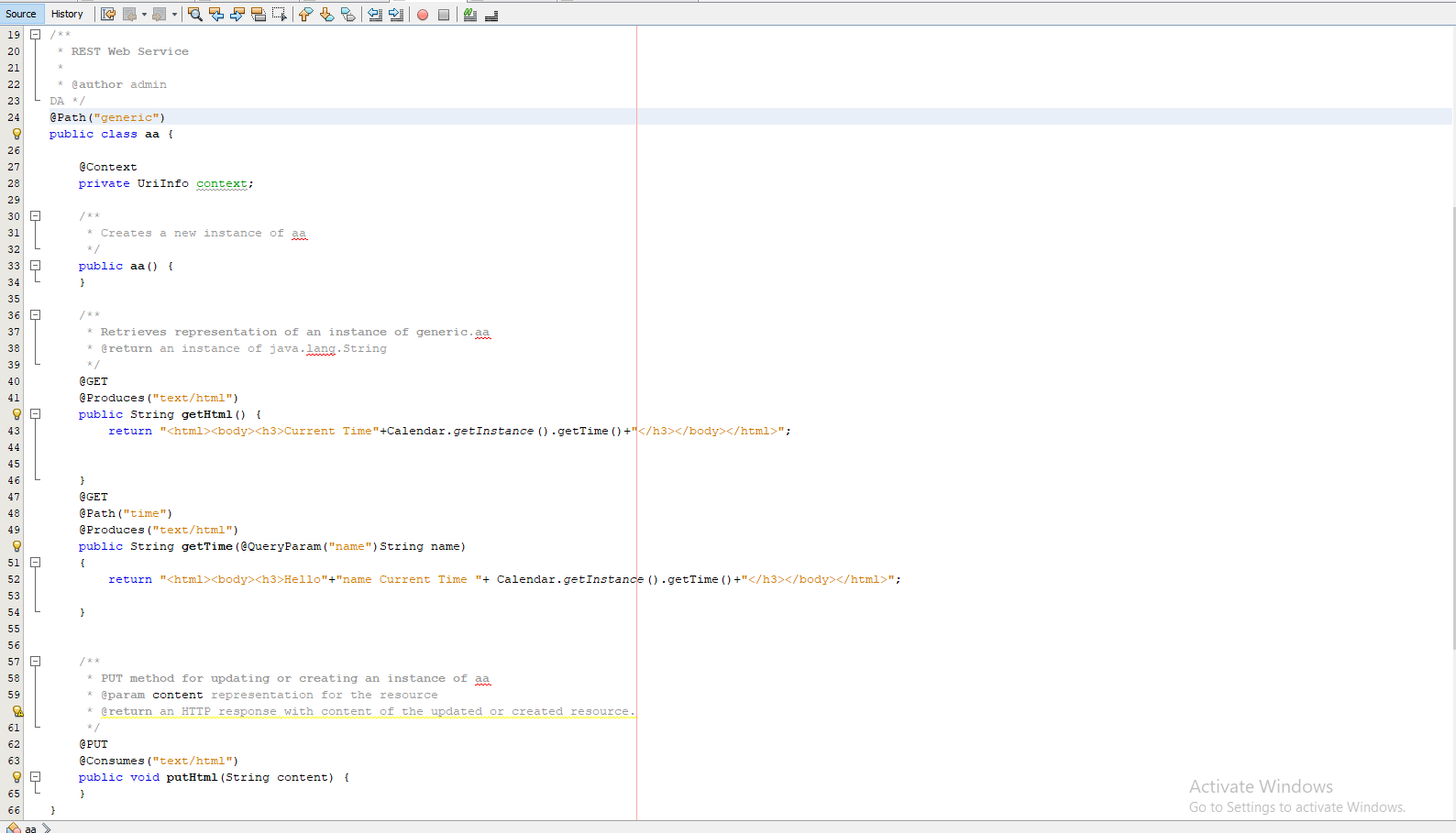




* + 1. Resources package and path name should be same
    2. Then Change MIME Type to “text/html”.

1. Then in GenericResource.java write the code

|  |
| --- |
| @GET  @Produces("text/html")  public String getHtml() {  return "<html><body><h3>Current Time"+Calendar.getInstance().getTime()+"</h3></body></html>";      }  @GET  @Path("time")  @Produces("text/html")  public String getTime(@QueryParam("name")String name)  {  return "<html><body><h3>Hello"+"name Current Time "+ Calendar.getInstance().getTime()+"</h3></body></html>";    } |

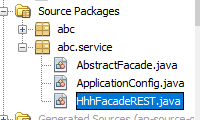
* 1. 

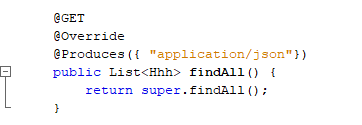
1. Then Deploy the Web application
2. Then test the Restful web services

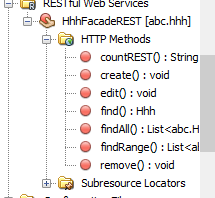
# Practical 5

## Q. Create Restful Services from Database and return data in JSON format

Solution

1. Create a web application
   1. Connect the database
2. Then take new Restful Webservices from database
   1. Then add table to it
   2. Then make changes in
   3. 
   4. Then in 66 line



* 1. Then test
  2.  then test findall

1. Then the copy the url of the
2. Then write code in index.html

|  |
| --- |
| <!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></title>  </head>  <body>  <script language="javascript">    var req=new XMLHttpRequest();  req.open("GET","http://localhost:12667/f/webresources/abc.hhh/");    req.onload=function()  {    var data =JSON.parse(this.response);    var tb = document.getElementById("t");  for(var i=0;i<data.length; i++)  {  var r=tb.insertRow();  var c1=r.insertCell();  var c2=r.insertCell();    c1.innerHTML = data[i].agb;  c2.innerHTML = data[i].gg;    }  };  req.send();  </script>  <table id="t" border="1">  <tr>  <th>AGB</th>  <th>GG</th>  </tr>    </table>  </body>    </html> |

1. Then the run whole application

Output

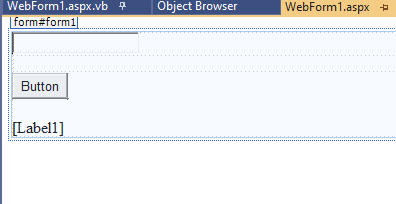
|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | | **AGB** | **GG** | |

# Practical 6

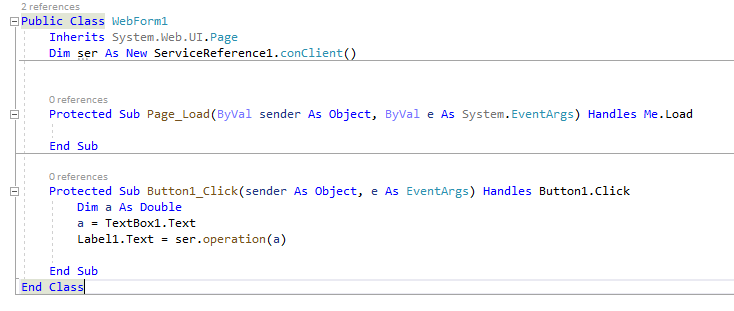
## Q. Create SOAP Web service to convert temperature to Celsius to Fahrenheit and Consume using .NET client .

1. Create a soap webservice then write the code

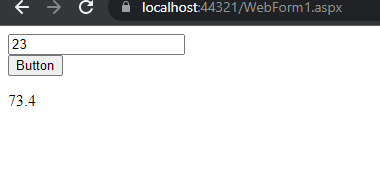
|  |
| --- |
| @WebMethod(operationName = "operation")  public double operation(@WebParam(name = "a") double a) {  double call=(a\*1.8)+32;    return call;  } |

1. Then run and deploy
   1. Then copy then wsdl file
2. Then create a .net application
   1. Create a ASP .net web application
      1. Use empty form
      2. Create new web form
   2. Then add References
      1. RC on Reference > Then Add service reference
      2. Paste wsdl link
   3. Layout for web form 
   4. Then double click on button
   5. Code

|  |
| --- |
| Protected Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click  Dim a As Double  a = TextBox1.Text  Label1.Text = ser.operation(a)  End Sub |



1. Then run
2. Output



# Practical 7

## Q. Create SOAP webservice to perform addition of two numbers and consume using php client

1. Create web service
   1. Code for addition

|  |
| --- |
| @WebMethod(operationName = "addition")  public int addition(@WebParam(name = "a") int a, @WebParam(name = "b") int b) {  //TODO write your implementation code here:  return a+b;  } |

* 1. Run and deploy
  2. Copy wsdl

1. Create new notepad file and write for the php

|  |
| --- |
| <?php  $Client = New SoapClient("http://localhost:12667/practical7/addition?WSDL");  print\_r($Client->addition(12,34));  ?> |

* 1. Save the file in (C:\wamp\www)
     1. With extension (.php )
     2. And all text file format
  2. Start wamp server
  3. Then from



* 1. Click and select local host

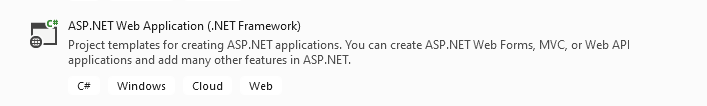
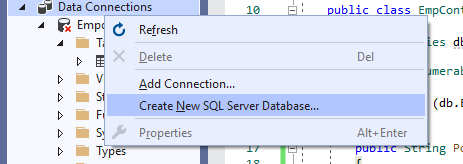
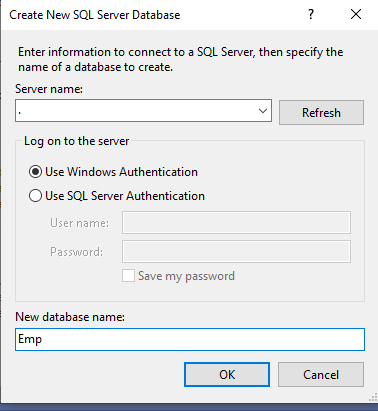
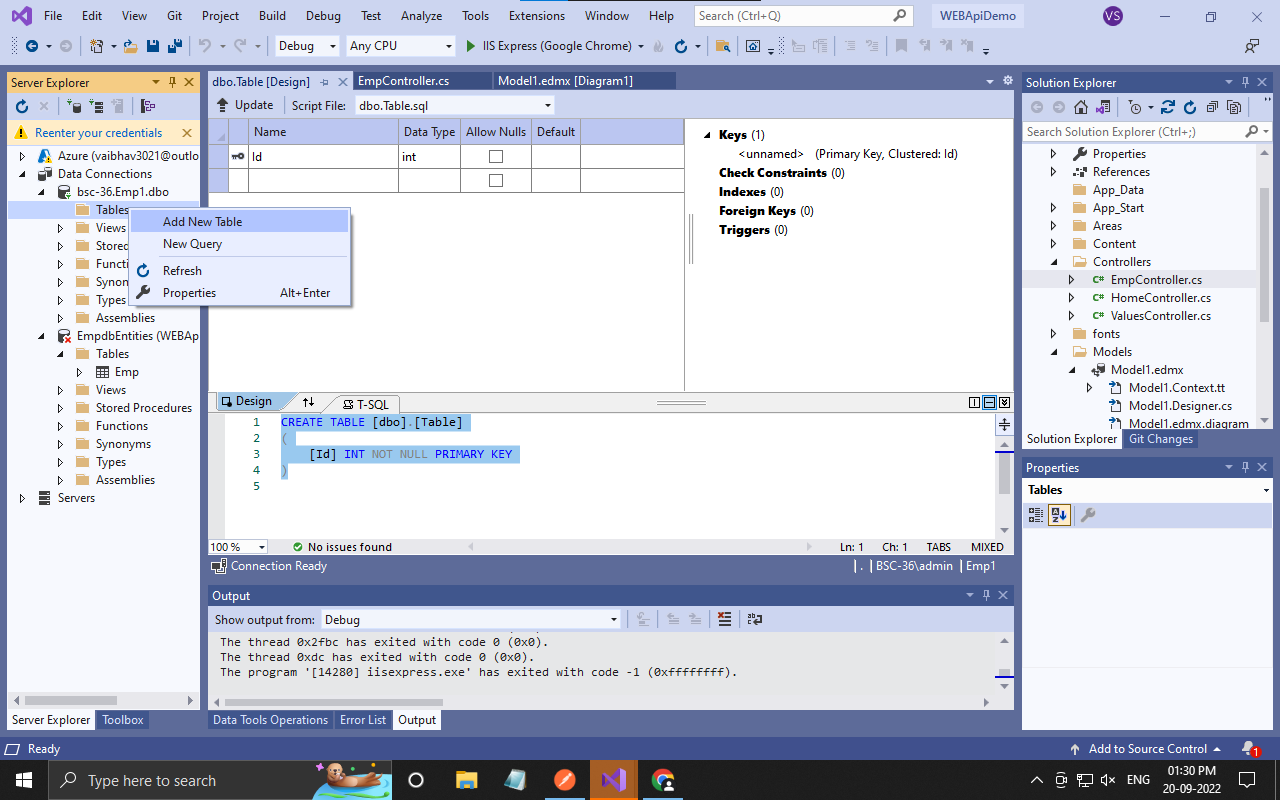
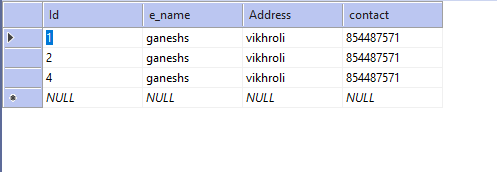
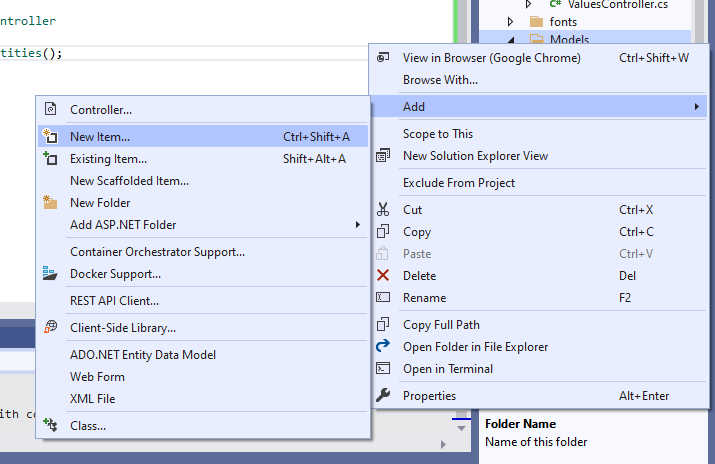
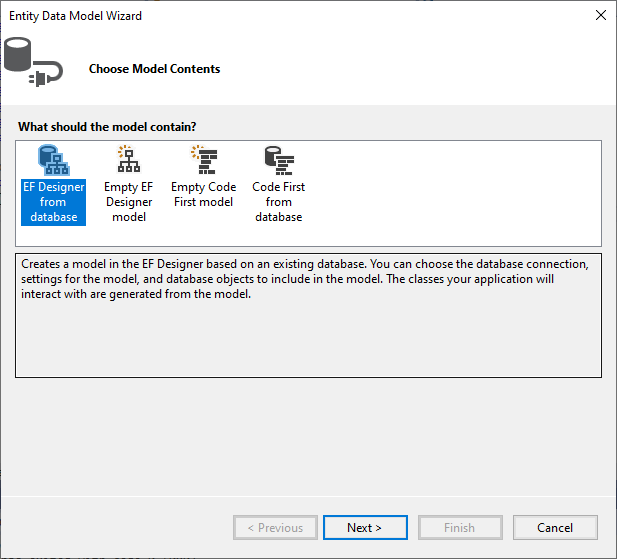
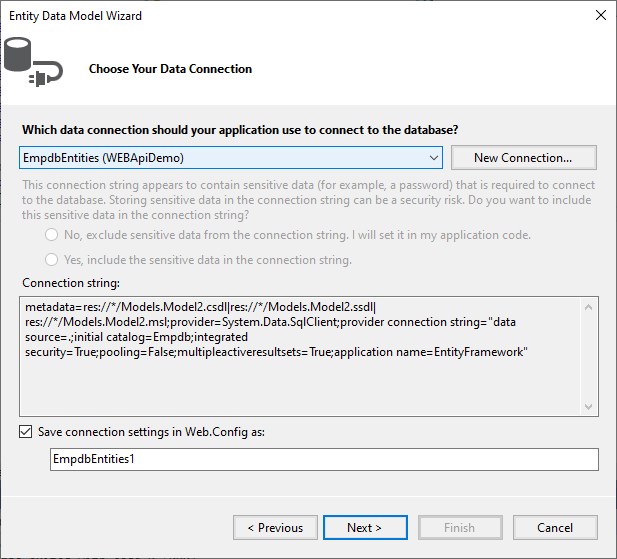
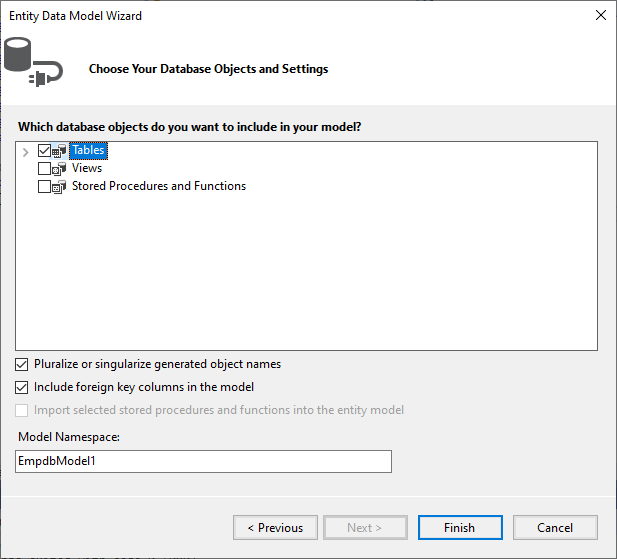
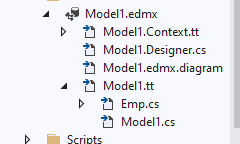
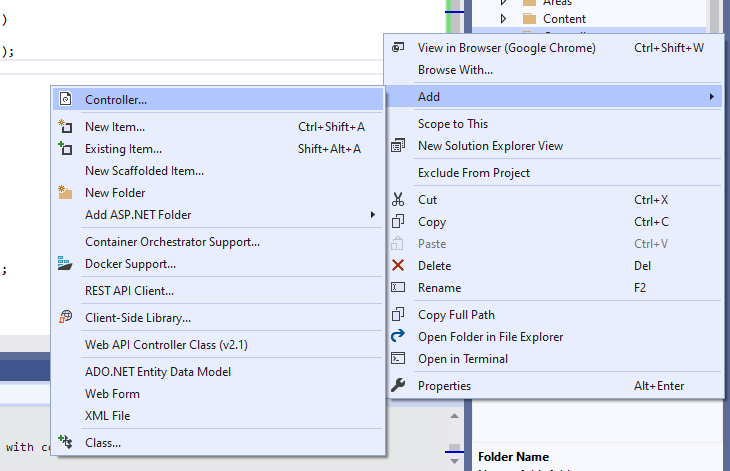
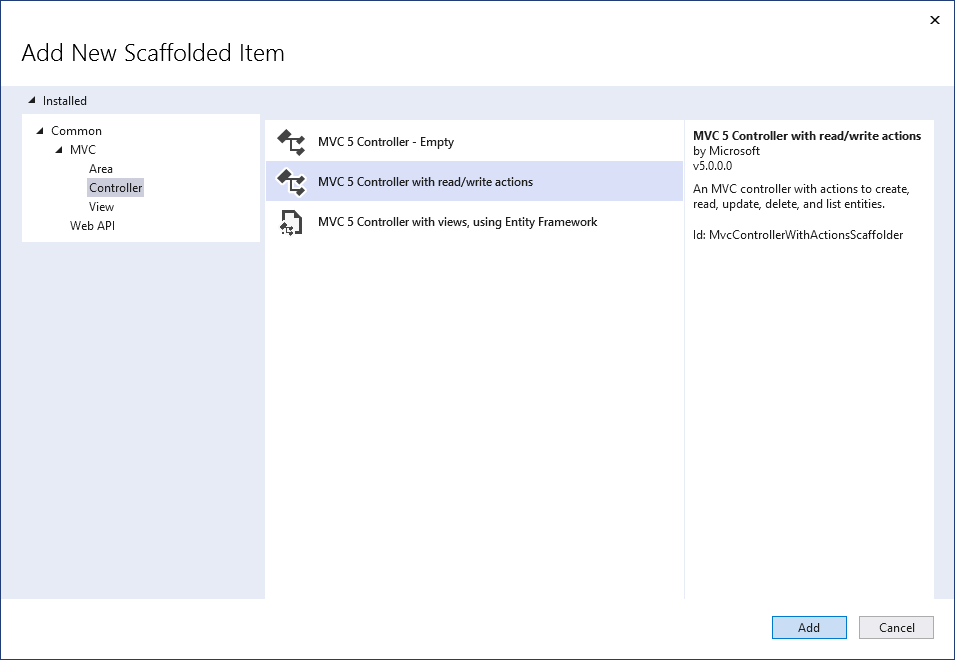
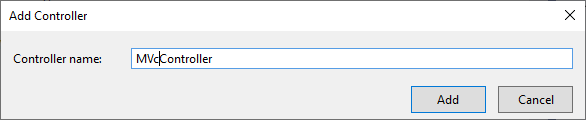
1. NOTE (go in phpinfo and check whether you have soap enabled if not then follow this steps)
   1. Go to this location (C:\wamp\bin\apache\Apache2.2.11\bin)
      1. Then open php file
      2. Then search (php\_Soap.dll)
      3. Then remove (;) for it and save the file
   2. Restart the server
2. Then again to wamp server then localhost
   1. Your project



* 1. Open then file

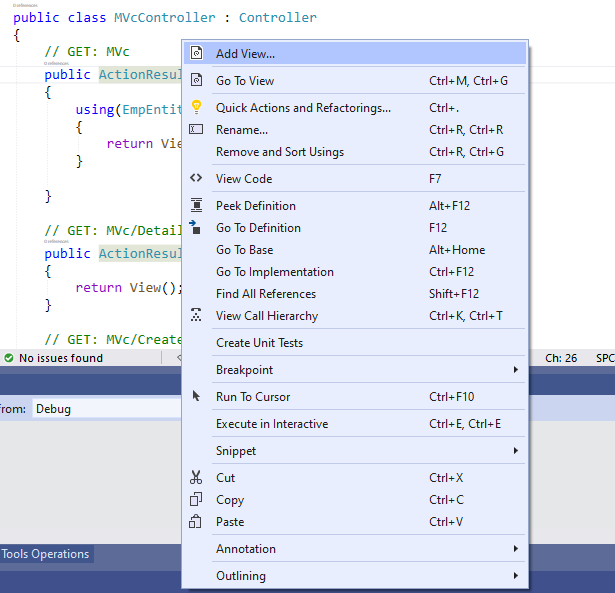
# Practical 8

## Q MVC

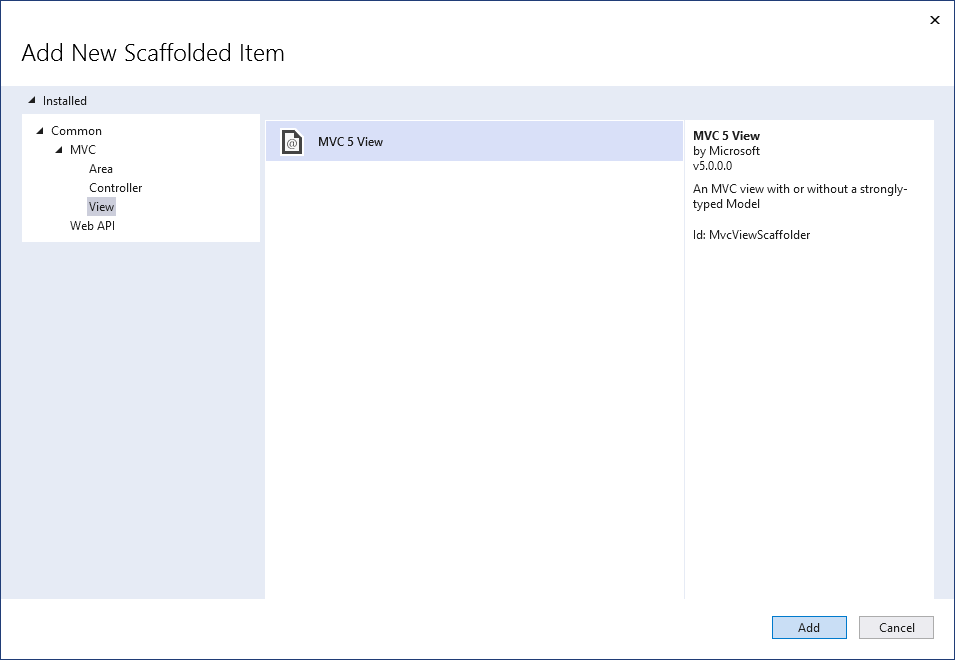
1. Open Visual studio 2019
2. Create new project  language C#
3. 
4. Name it as” Emp”
5. Create and the select **MVC**
6. Untick “configure for HTTPS” from advance option
7. Go to server explorer > data connections > create new sql server database
8. 
9. Server name as “.” > new database name as “Emp”
10. Server name as “.” > new database name as “Emp”
11. 
12. Then rc table > create new table > give table names
13. As per picture
14. 
15. Fill at least one data
16. Go to models > add > new item>
17. 
18. Then select > data >ADO.NET> name as default
19. 
20. Select database
21. 
22. Select table
23. 
24. 
25. This will appear in models
26. Right click controller > add > controller….
27. 
28. Select MVC > controller > second option> add
29. 
30. Name it as MVc
31. 
32. Make changes in highlighted part

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Web;  using System.Web.Mvc;  using MVcDemo.Models;  namespace MVcDemo.Controllers  {  public class MVcController : Controller  {  // GET: MVc  public ActionResult Index()  {  using(EmpEntities db=new EmpEntities())  {  return View(db.Tables.ToList());  }    }  // GET: MVc/Details/5  public ActionResult Details(int id)  {  return View();  }  // GET: MVc/Create  public ActionResult Create()  {  return View();  }  // POST: MVc/Create  [HttpPost]  public ActionResult Create(Table et)  {  try  {  using (EmpEntities db = new EmpEntities())  {  db.Tables.Add(et);  db.SaveChanges();  return RedirectToAction("INDEX");  }    }  catch  {  return View();  }  }  // GET: MVc/Edit/5  public ActionResult Edit(int id)  {  return View();  }  // POST: MVc/Edit/5  [HttpPost]  public ActionResult Edit(int id, FormCollection collection)  {  try  {  // TODO: Add update logic here  return RedirectToAction("Index");  }  catch  {  return View();  }  }  // GET: MVc/Delete/5  public ActionResult Delete(int id)  {  return View();  }  // POST: MVc/Delete/5  [HttpPost]  public ActionResult Delete(int id, FormCollection collection)  {  try  {  // TODO: Add delete logic here  return RedirectToAction("Index");  }  catch  {  return View();  }  }  }  } |

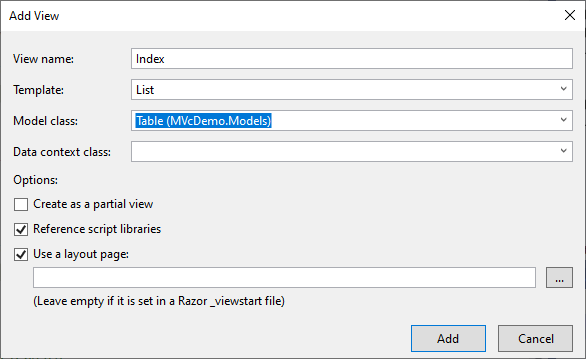
1. Right click on actionresult > add view>



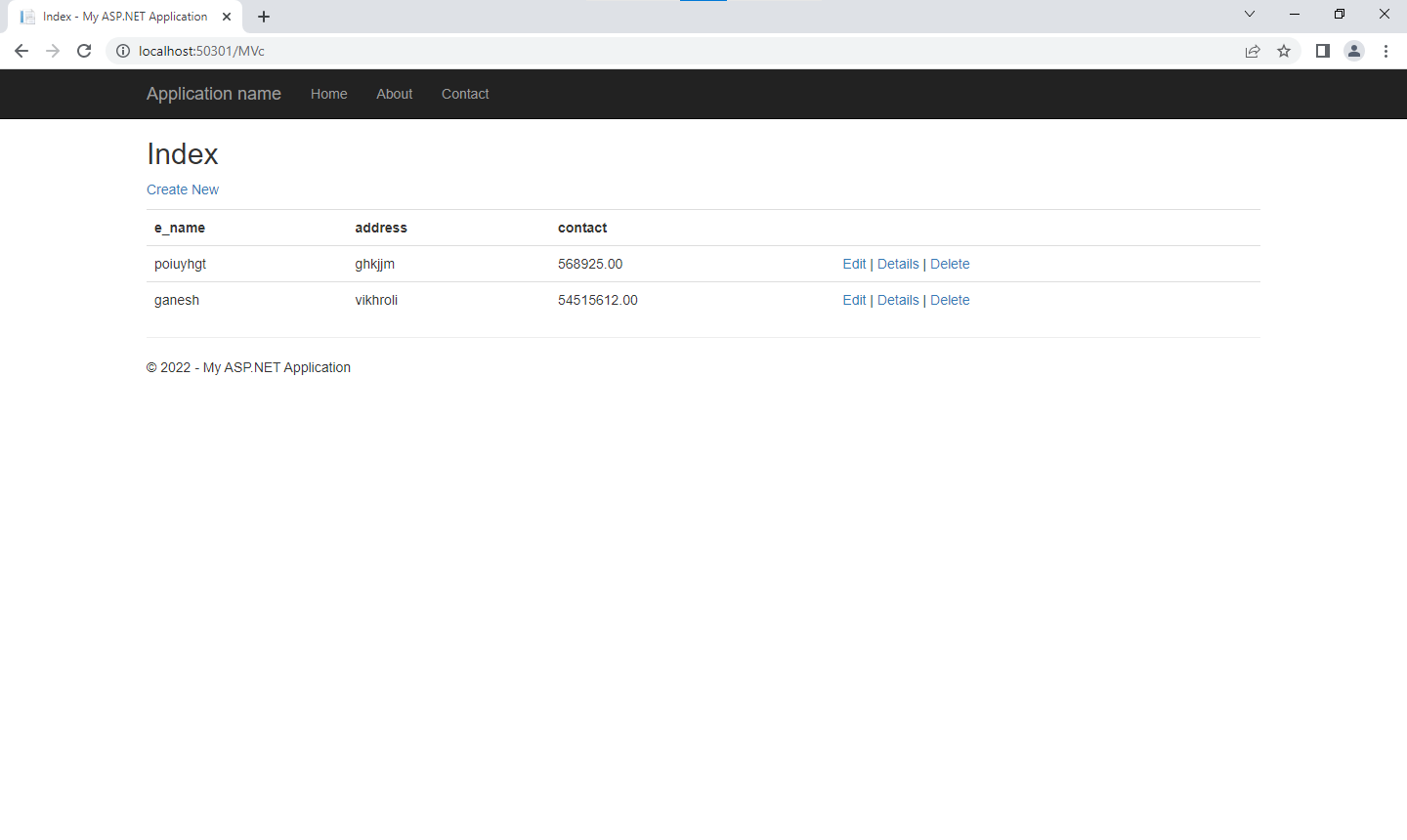
1. MVC>view > MVC 5 view > add

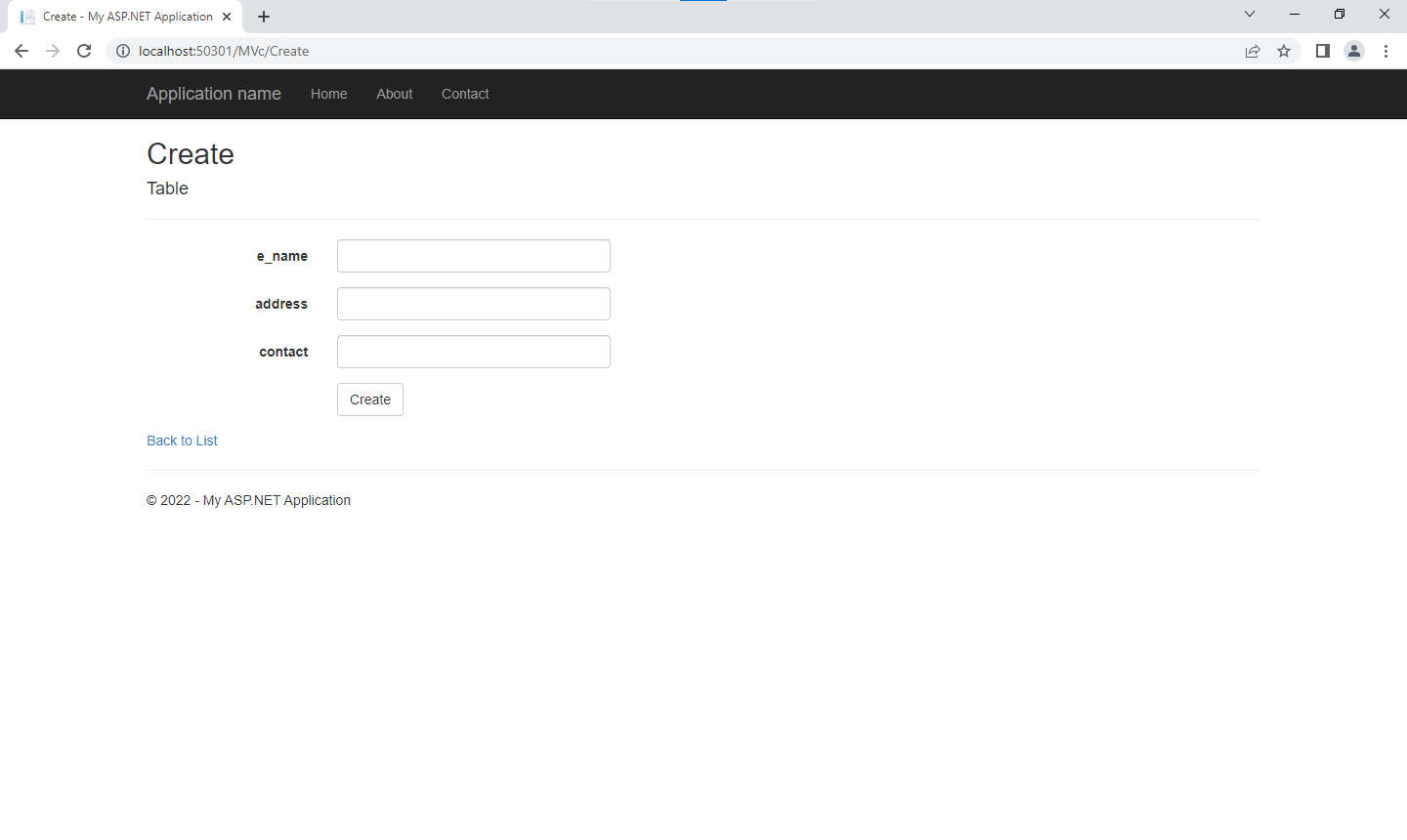


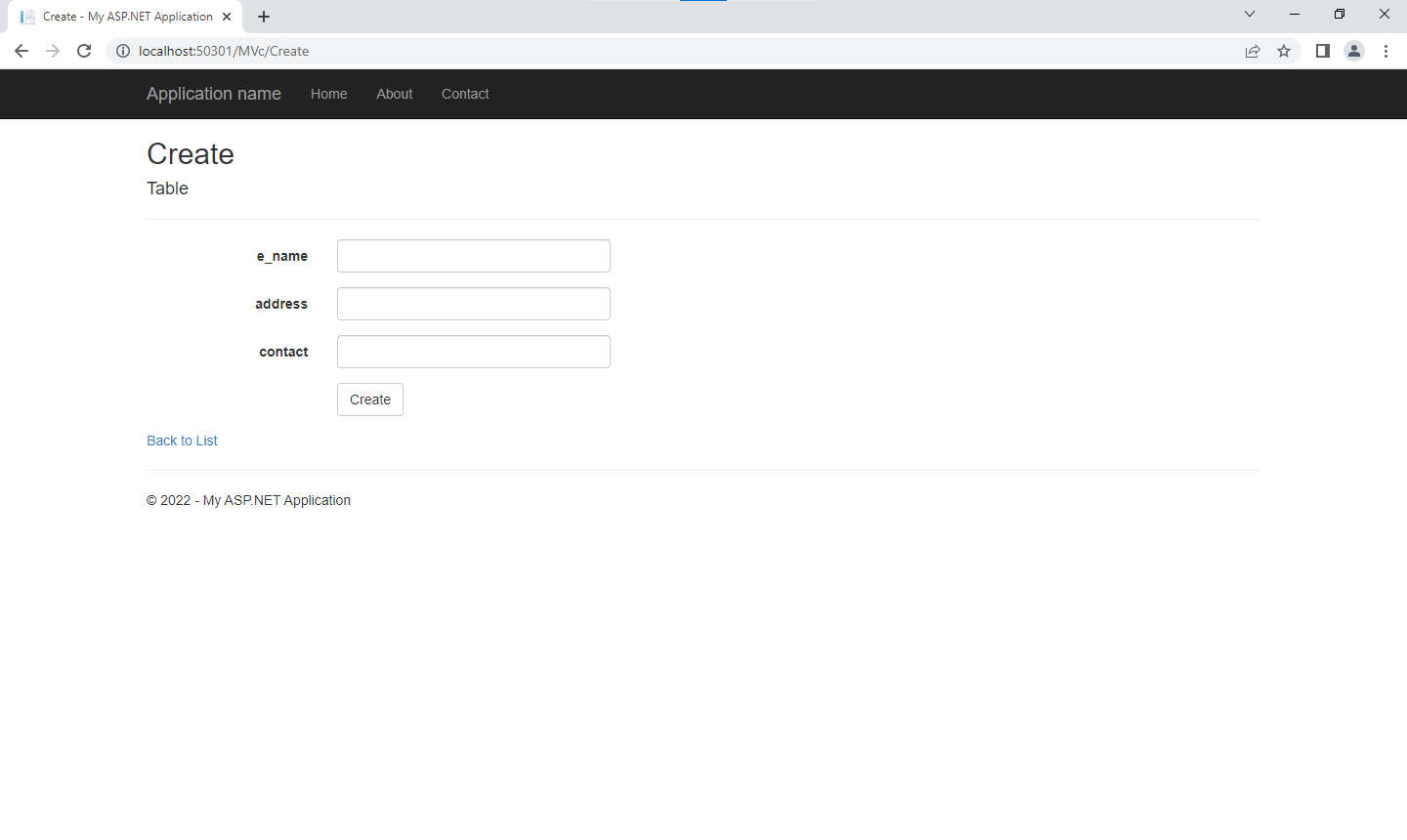
1. view name as default>template select list > model class as “table(MVcDemo Models “)>add



1. for create do same as list but instead of list select create option
2. then it will be redirected to the html file
3. and simply run the file and perform the need full



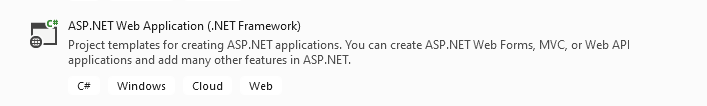




# Practical 9

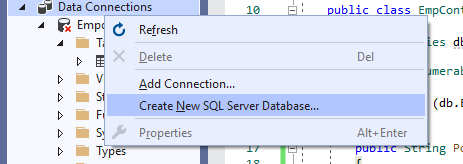
## Q RESTFUL Services Using WebAI

1. Open Visual studio 2019
   1. Create new project language C#

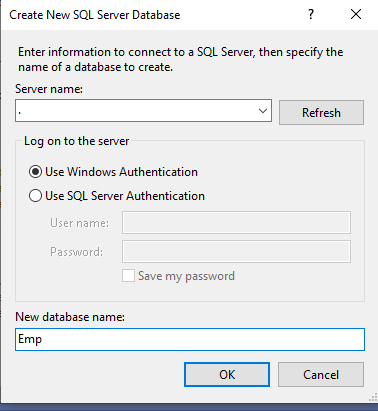


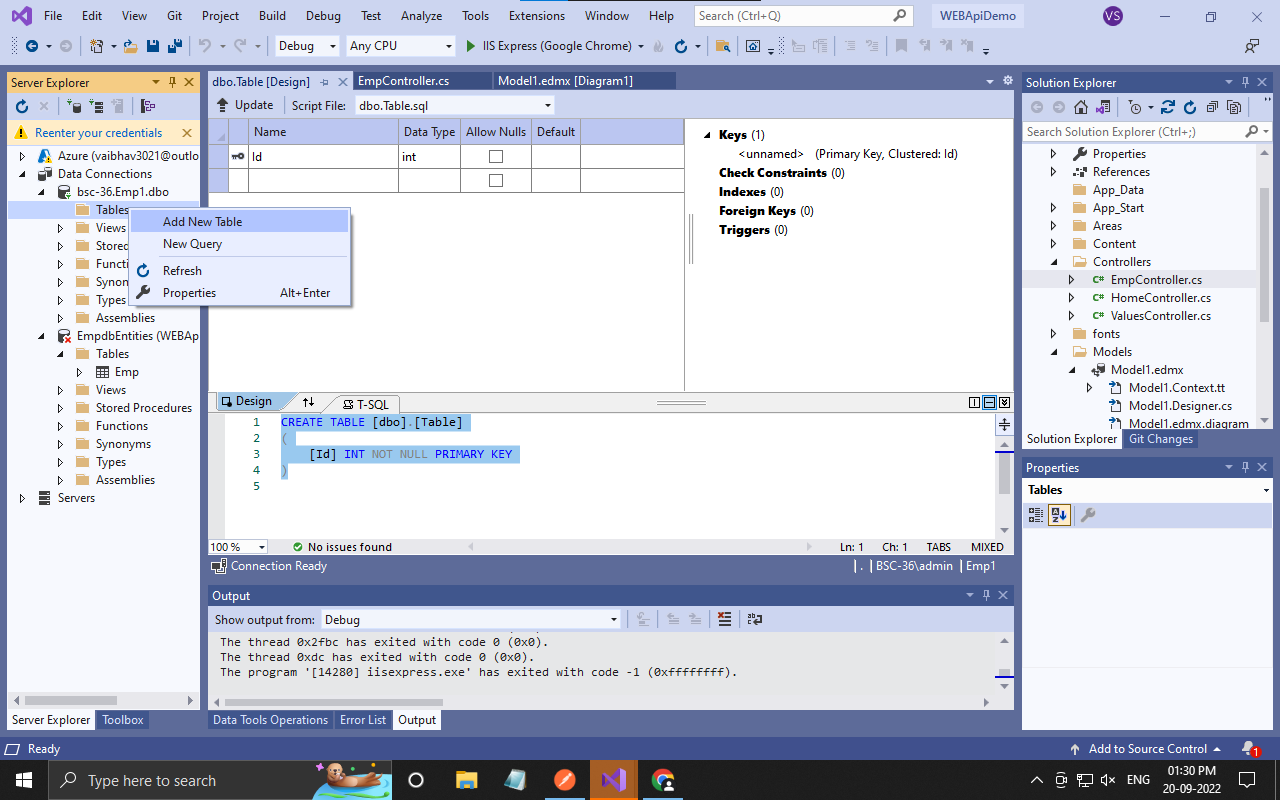
* 1. Name it as” Emp”
  2. Create and the select web api

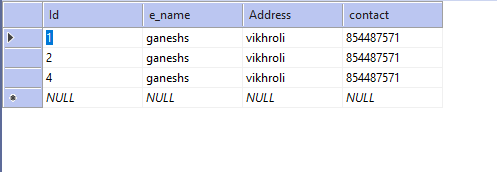
1. Go to server explorer > data connections > create new sql server database



* 1. Server name as “.” > new database name as “Emp”

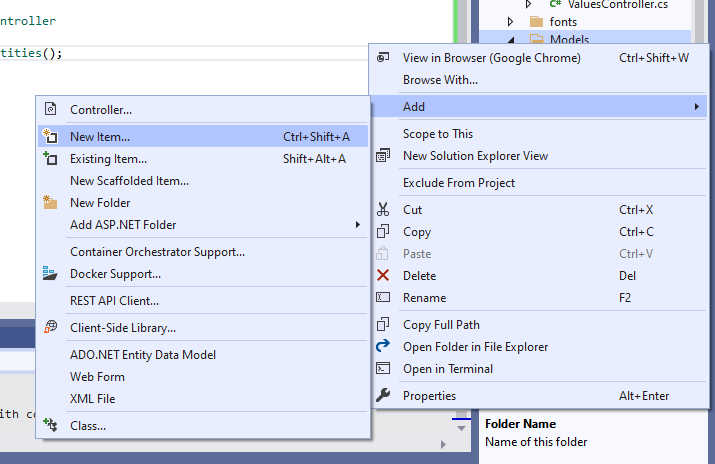


* + 1. Then rc table > create new table > give table names
    2. As per picture

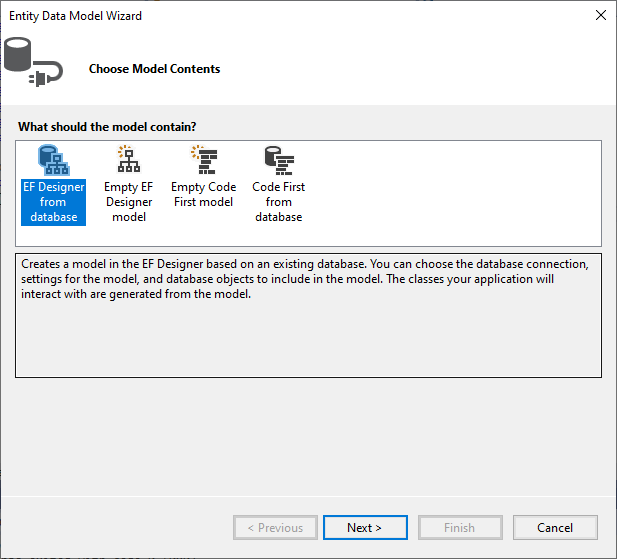


* + 1. Fill at least one data

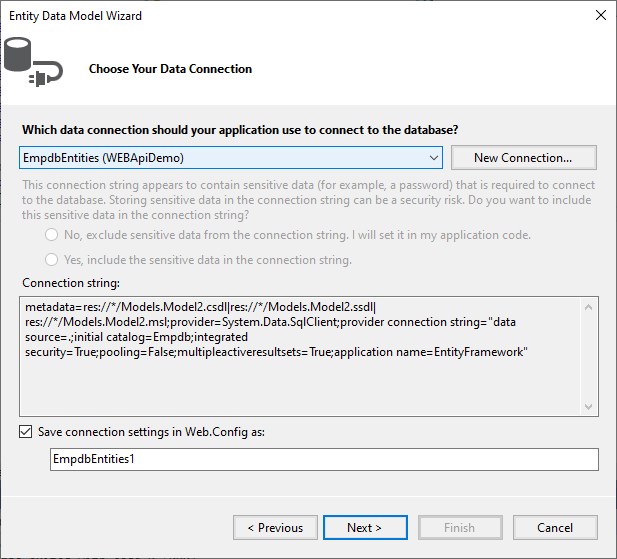
1. Go to models > add > new item>



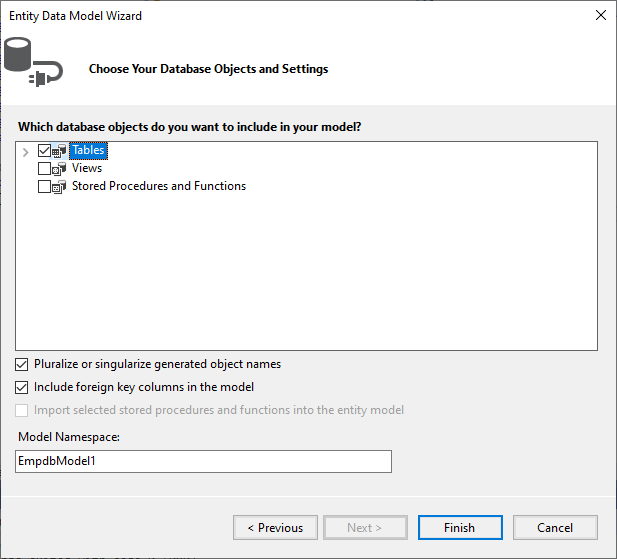
* 1. Then select > data >ADO.NET> name as default

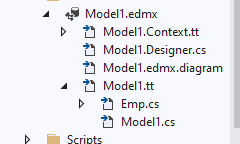


* + 1. Select database

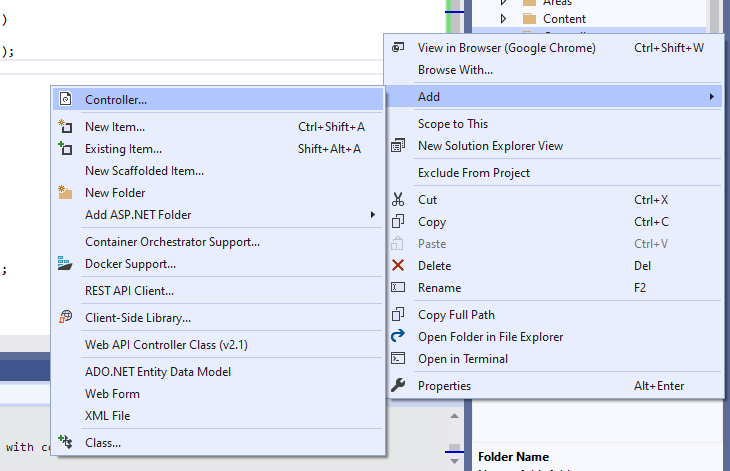


* + 1. Select table

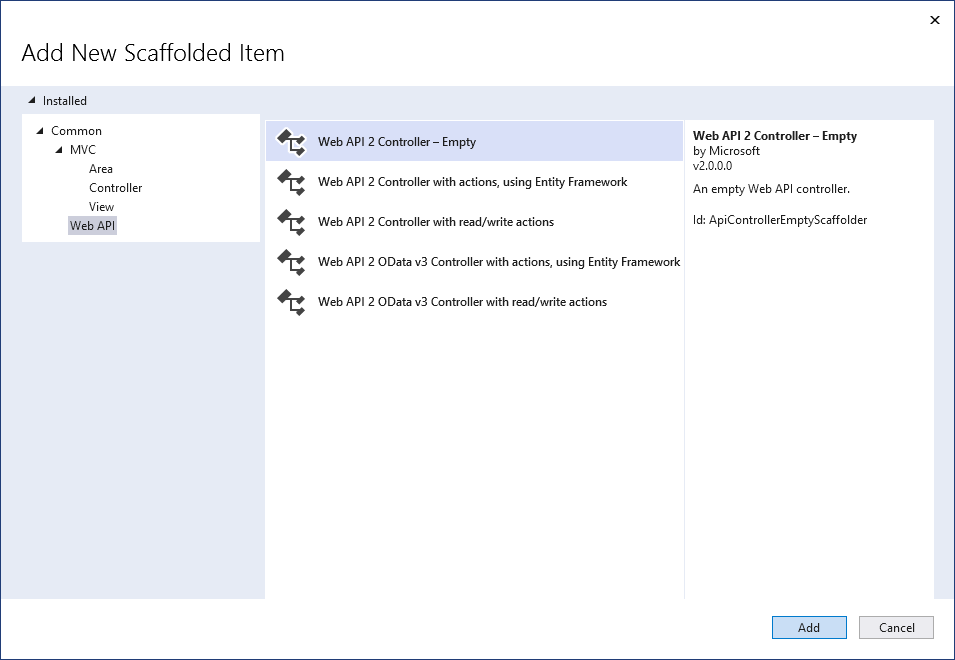




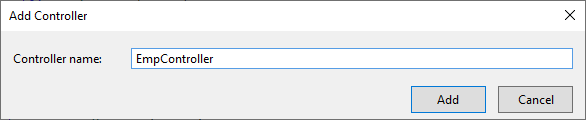
1. This will appear in models
2. Right click controller > add > controller….



* + 1. Select web api > web api 2 controller - empty



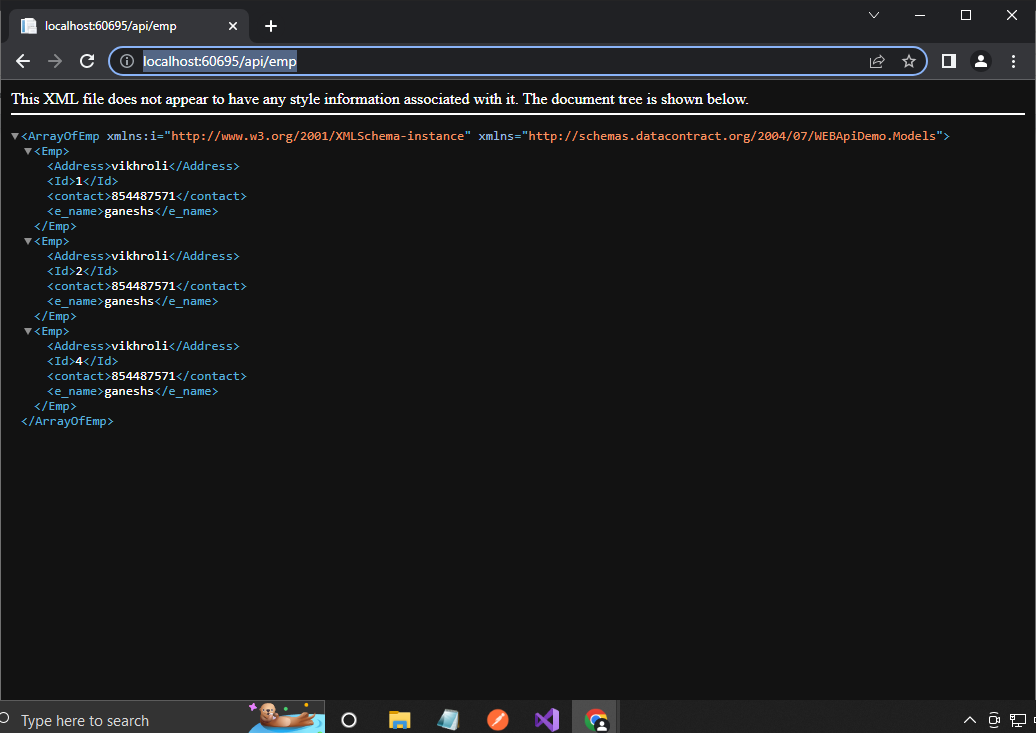
* 1. Give name as EmpController> then add



* 1. Then go in EmpController.cs then write code in it

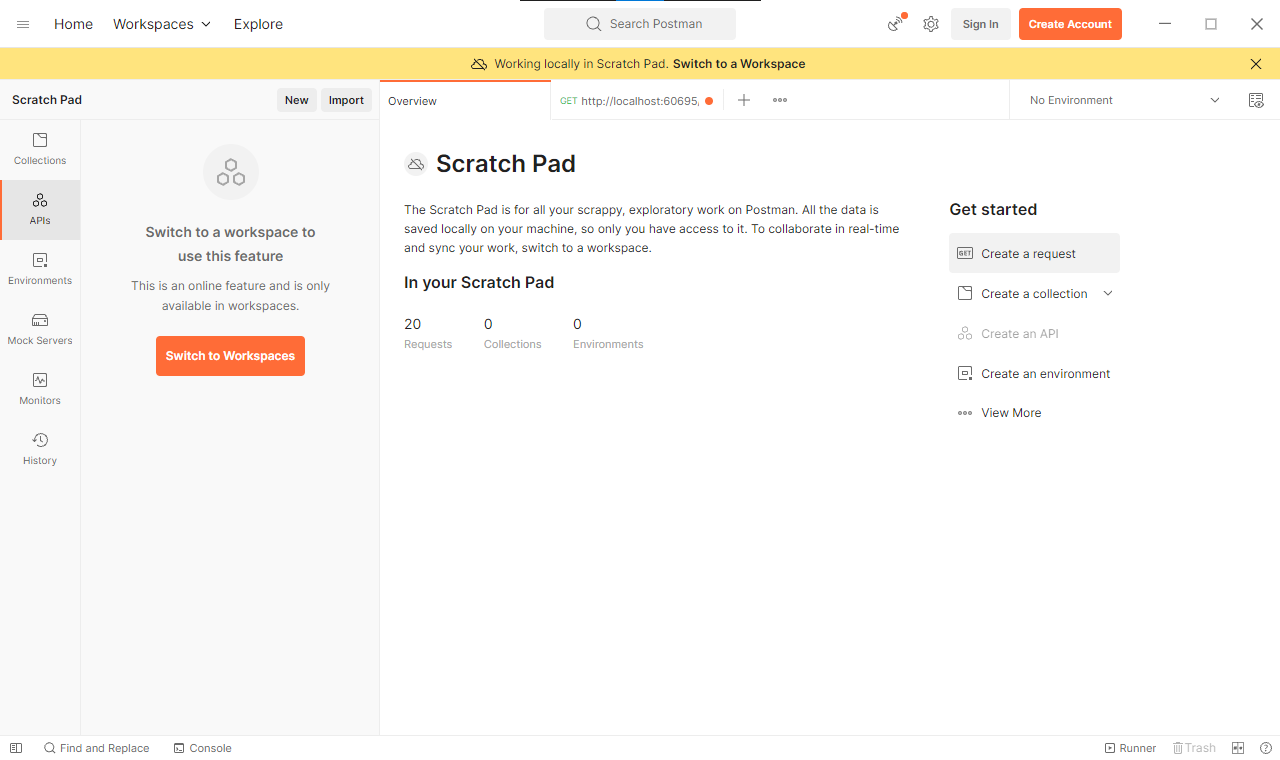
|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Net;  using System.Net.Http;  using System.Web.Http;  using WEBApiDemo.Models;  namespace WEBApiDemo.Controllers  {  public class EmpController : ApiController  {  EmpdbEntities db = new EmpdbEntities();  public IEnumerable<Emp> Get()  {  return (db.Emps.ToList());  }  public String Post(Emp et)  {  db.Emps.Add(et);  db.SaveChanges();  return "data added";  }  public String Delete(int id)  {  Emp et=db.Emps.Find(id);  db.Emps.Remove(et);  return "Record deleted ";  }  }  } |

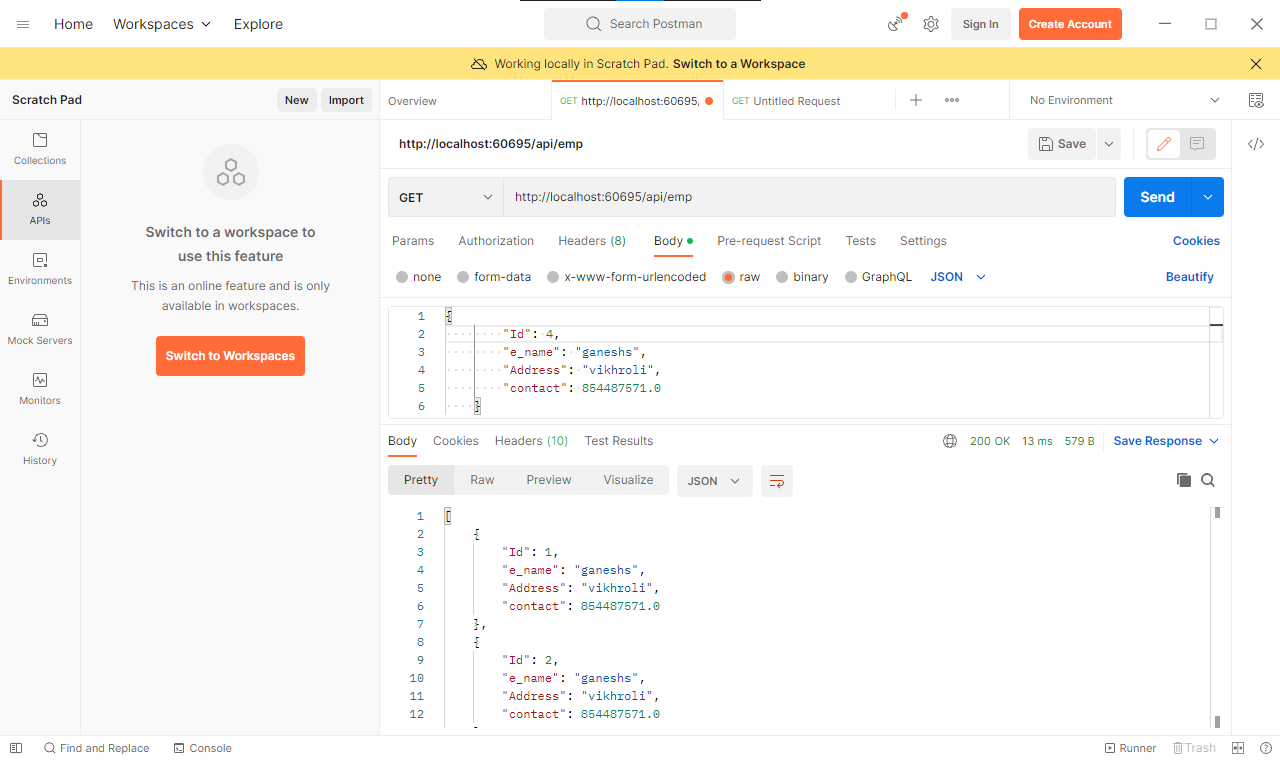
* 1. Then run the code
  2. Go to this url “http://localhost:60695/api/emp”

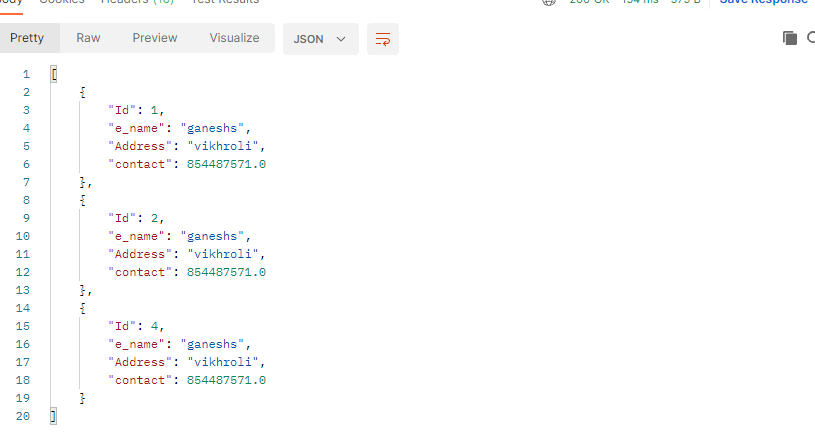


* 1. Copy url

1. Download and open postman



* 1. Create a new request
     1. Past the URL in search bar and send (web browser should be running in background )
     2. Output for get



* 1. For post select post method > go to body > select raw and json
  2. Then write the data u want to insert

1. output

