**M.Sc. Computer Science Semester-I**

**Name:** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **Roll No** : \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Elective Paper I (**Trends in Cloud Computing**)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **INDEX** | | | | |
| **NO** | **DATE** | **TITLE** | **PAGE NO** | **SIGN** |
|  | | | | |
| 1 |  | SOAP Webservices |  |  |
| 2 |  | Create RestFul Services to perform CRUD operation |  |  |
| 3 |  | Create RestFul Services from Pattern |  |  |
| 4 |  | MVC application. |  |  |
| 5 |  | Service Endpoint |  |  |
| 6 |  | Restful services using WEB-API. |  |  |
| 7 |  | Web application using Azure |  |  |
| 8. |  | Install Google App Engine. Create hello world app and other simple web applications using python/java. |  |  |

**Practical 1**

**Aim:** Create SOAP webservices to find factorial of a number.

**Source Code:**

**FactorialWs.java**

package abc;

import javax.jws.WebService;

import javax.jws.WebMethod;

import javax.jws.WebParam;

@WebService(serviceName = "FactorialWs")

public class FactorialWs {

@WebMethod(operationName = "factorial")

public int factorial(@WebParam(name = "x") int x) {

int fact =1;

for(int i = 1; i <= x; i++){

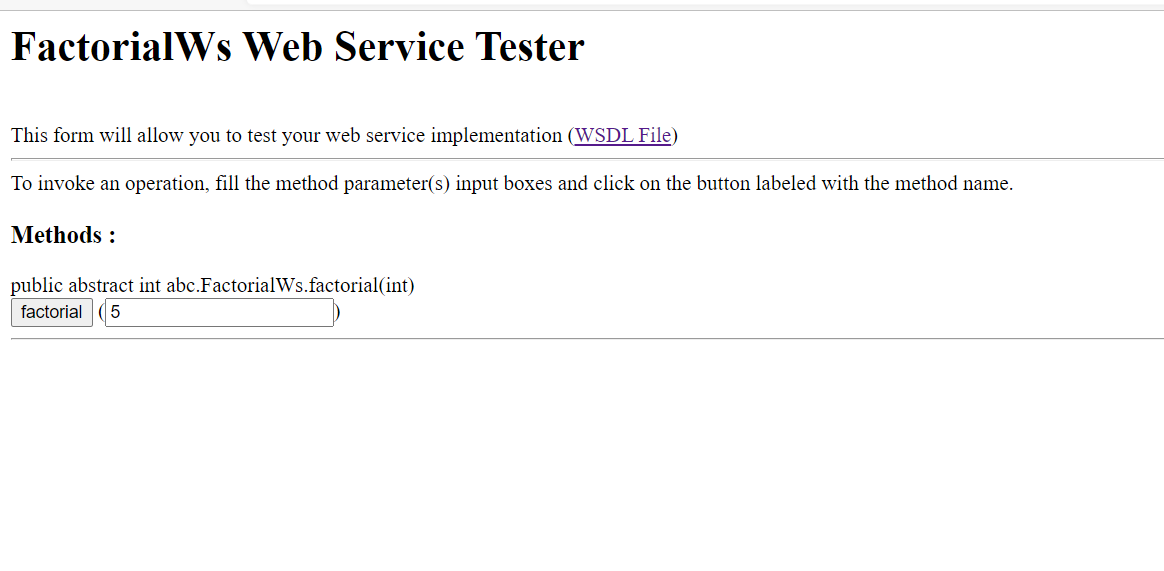
fact = fact \* i;

}

return fact;

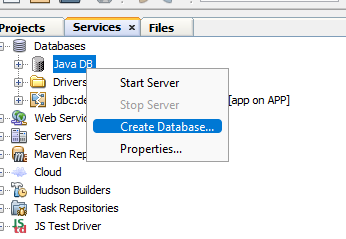
}

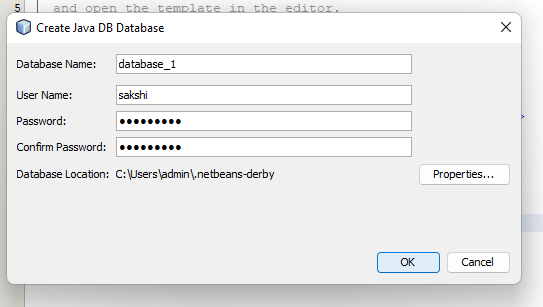
}

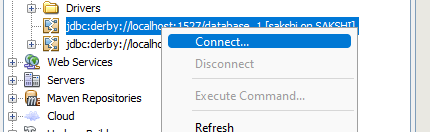
**Output:**

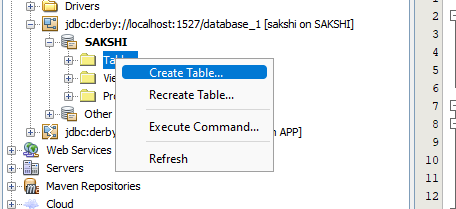
Practical-2

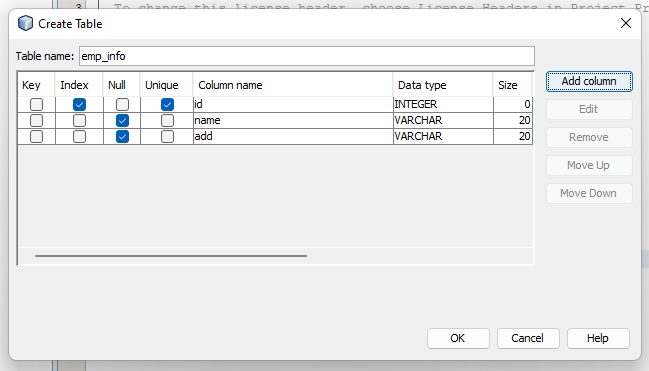
**Aim: RestFul Sevices using Java to perform CRUD operation.**

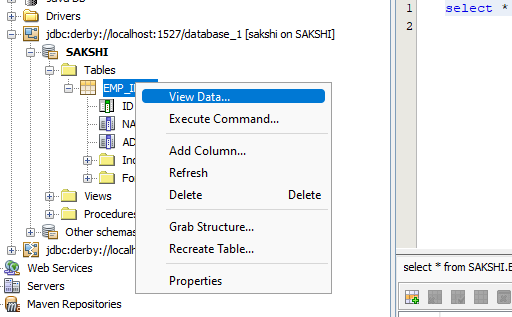


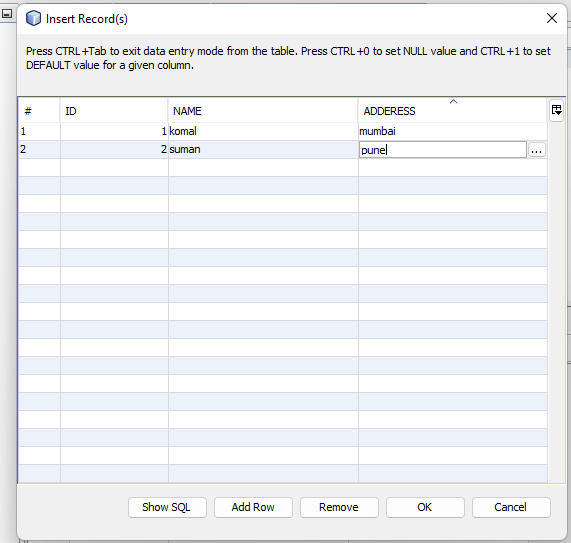


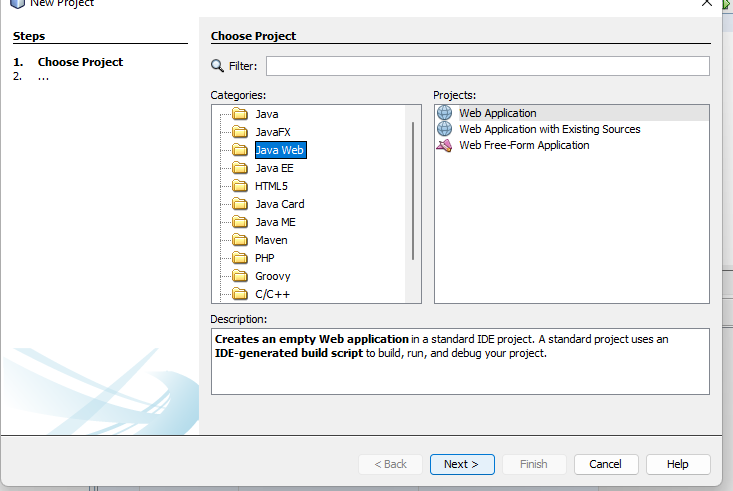


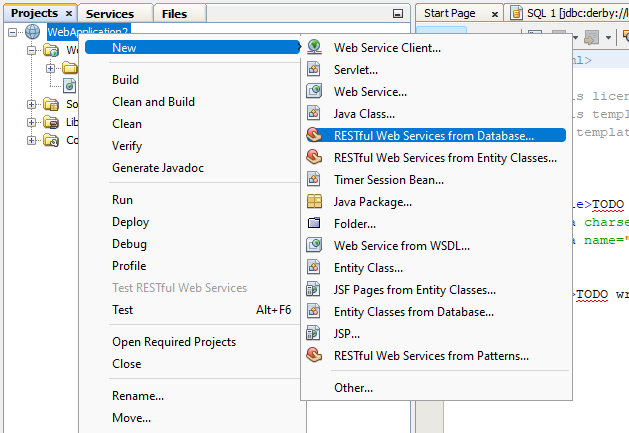


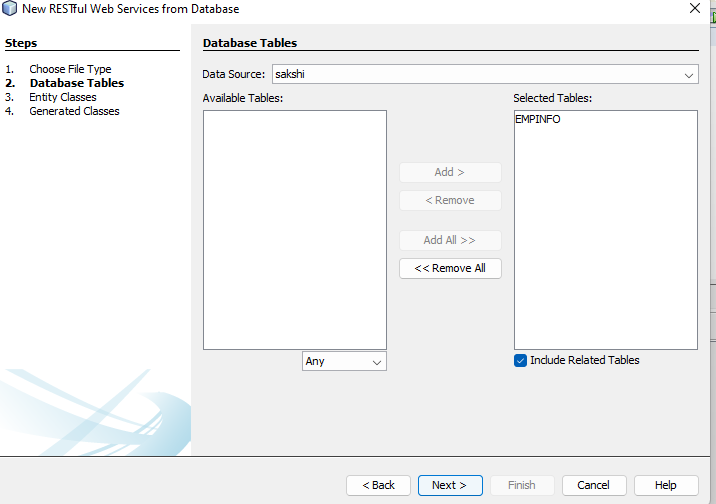


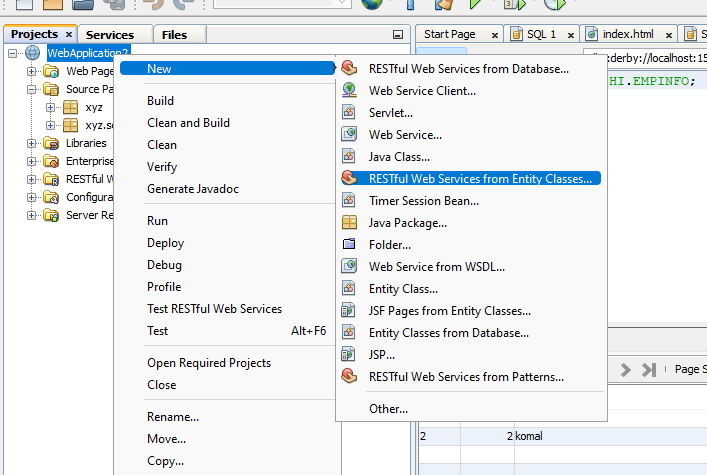


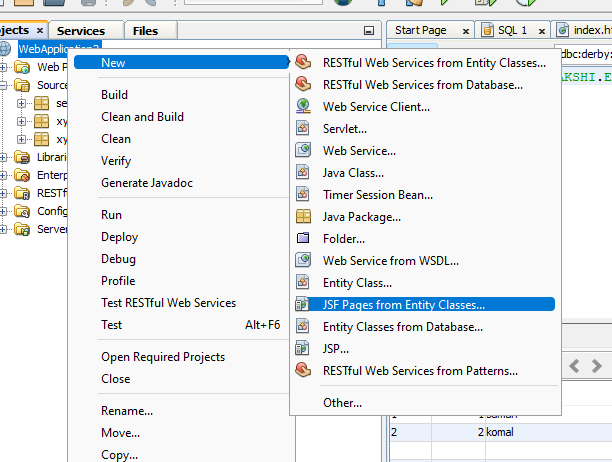


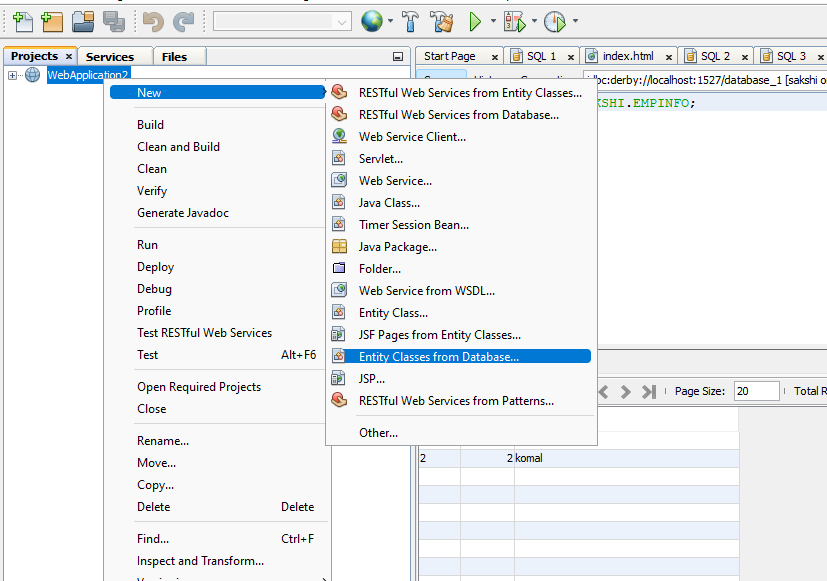




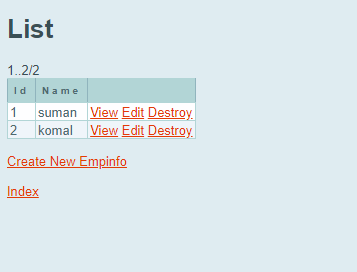




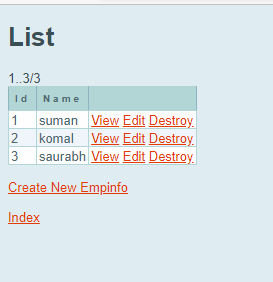




Then clean and build->deploy->run

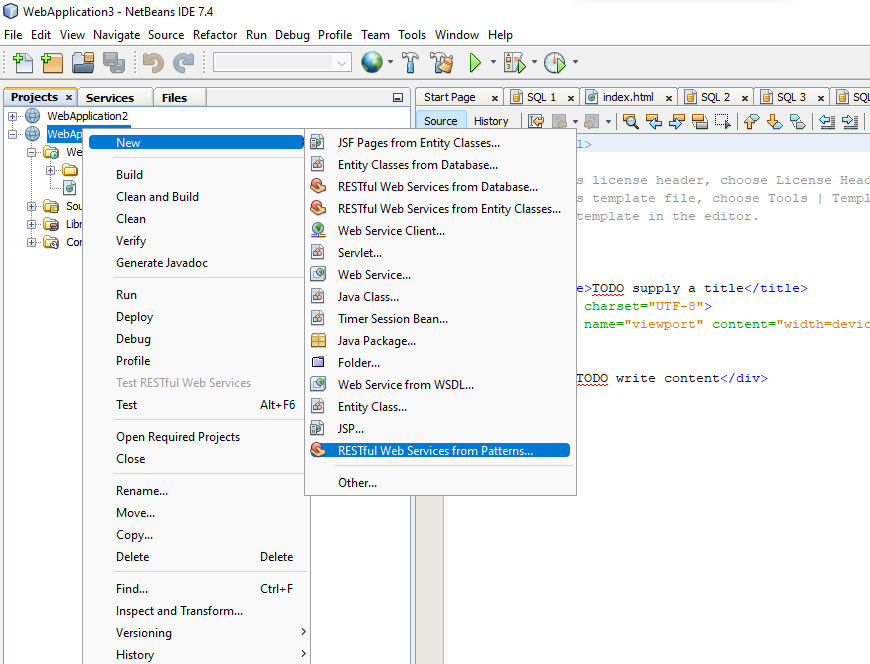


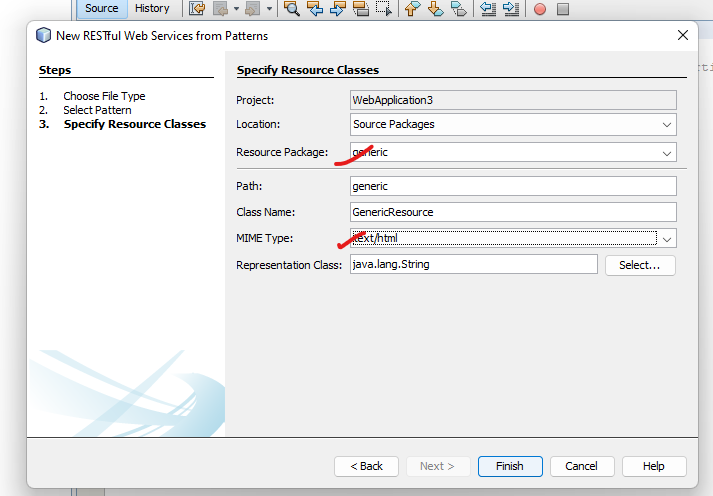
Add 1 more entry



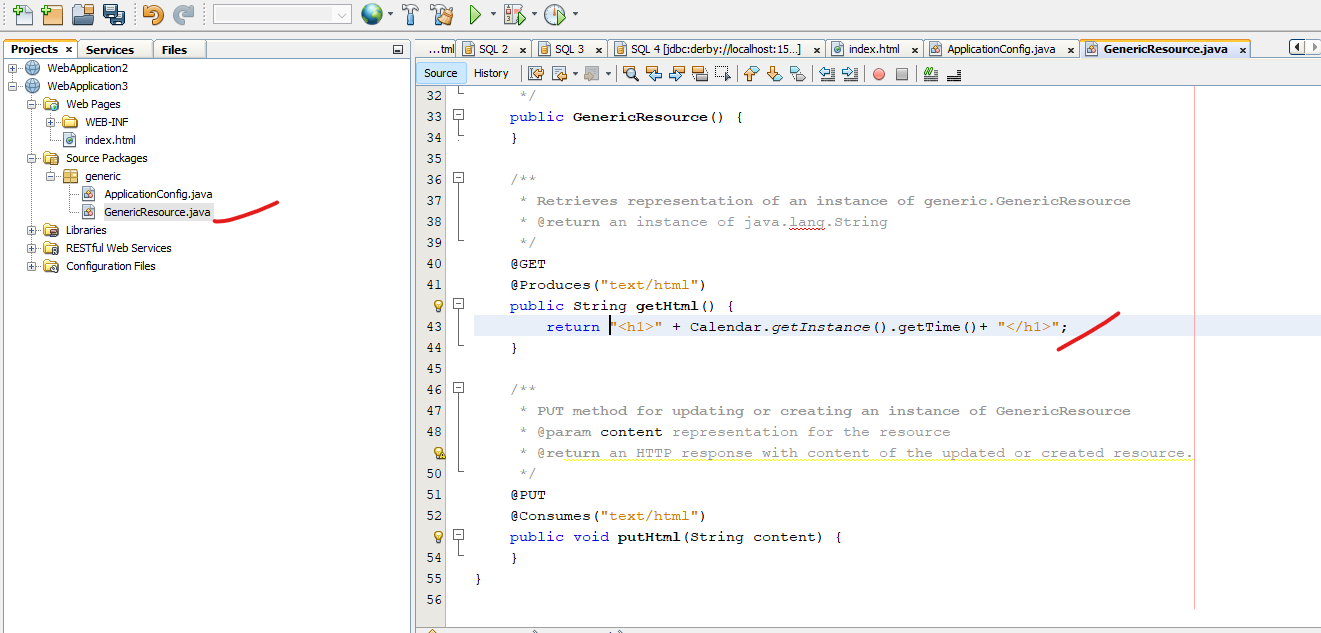
Practical-3

Create web application and add restful web services from pattern..

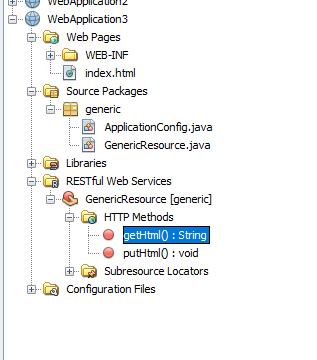




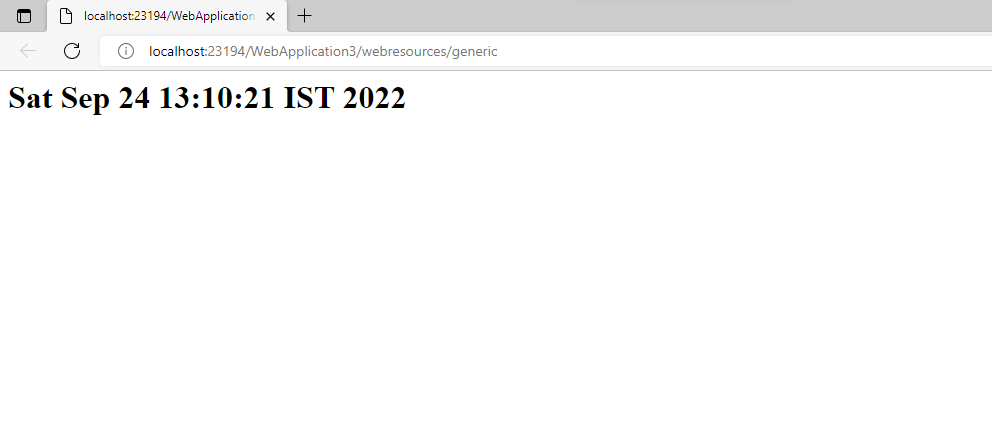
Write code in generic source…



After writing code clean and build web application->deploy->Test “getHtml():String”



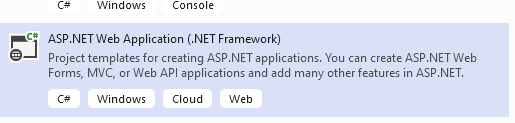
Output:

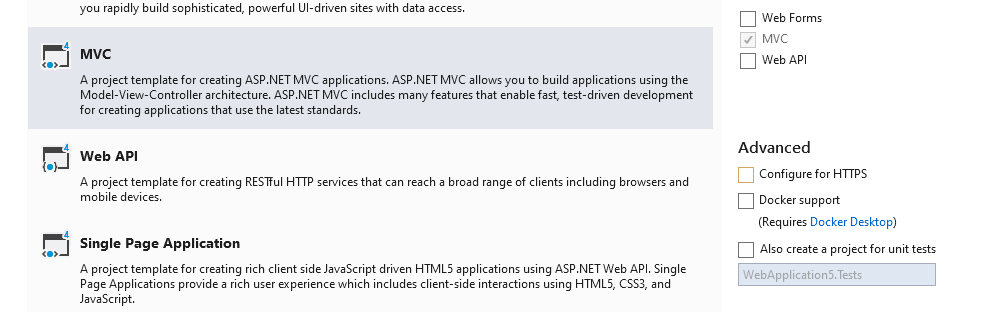


Practical 4

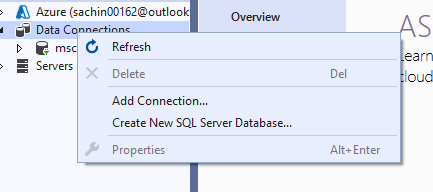
**Aim: Create MVC Application.**

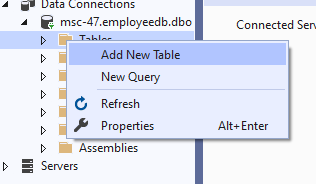
Create new project->select c sharp->select mvc and remove checkbox of http->create…..

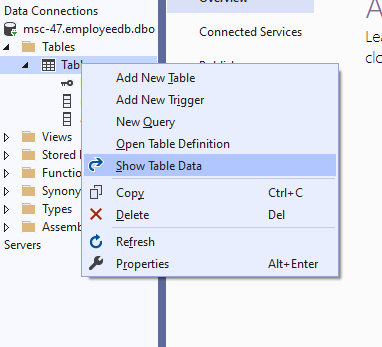




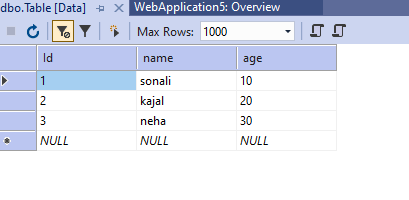
Create database and create table..accordingly



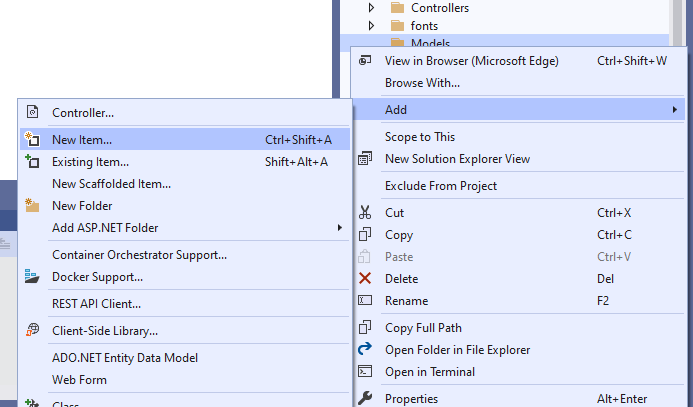


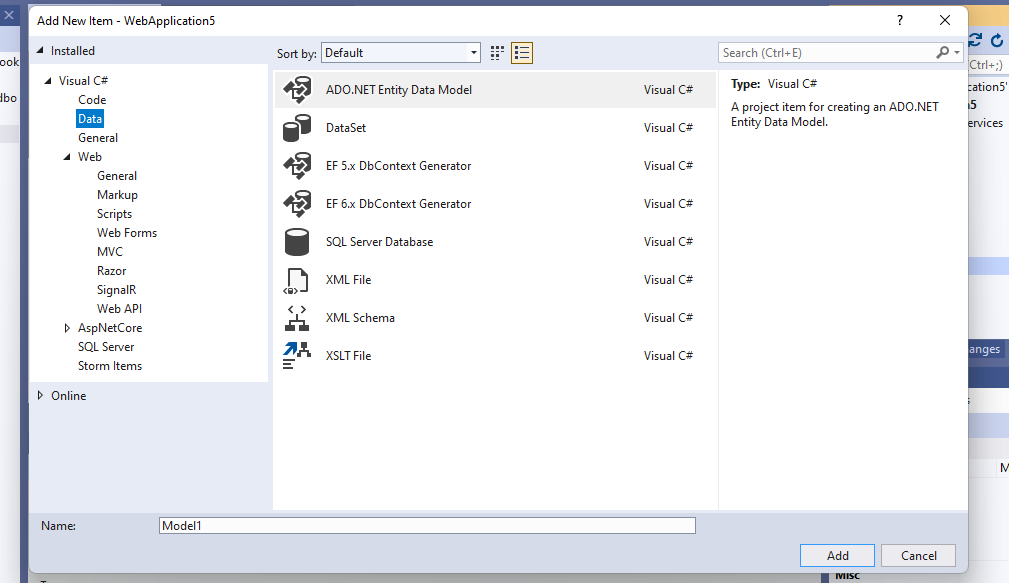


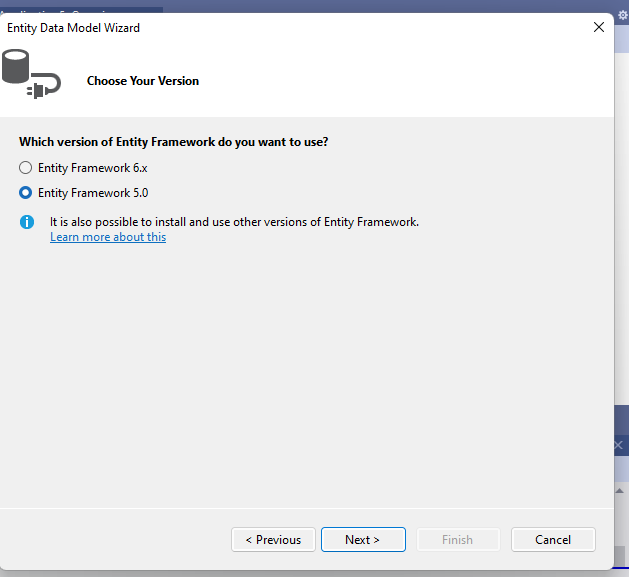
Add data to table…

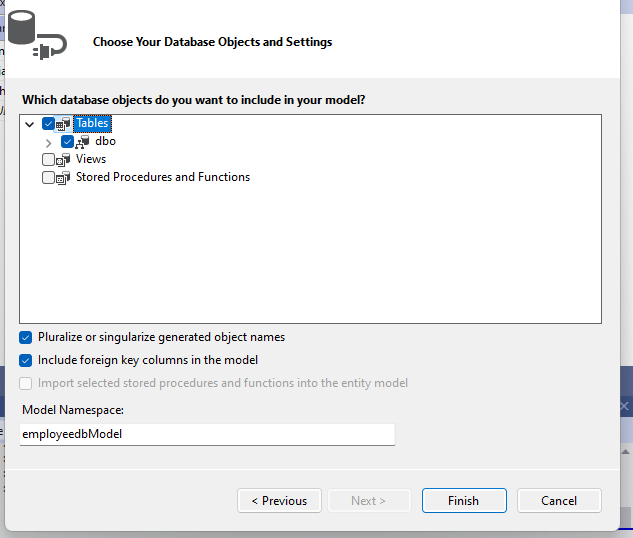


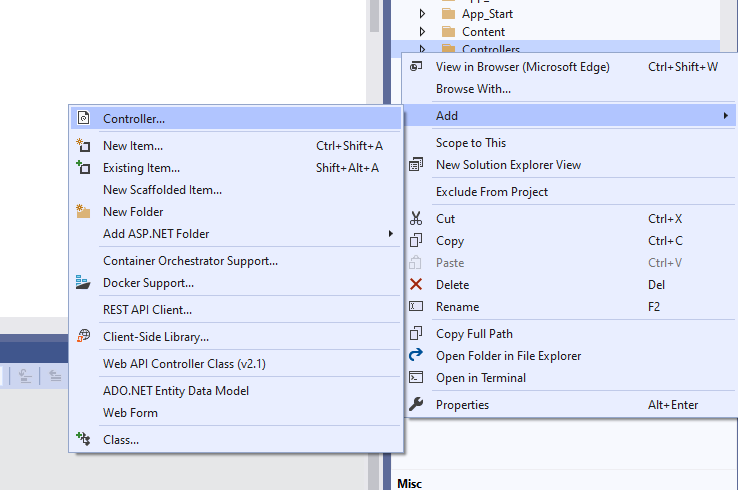
Update the dataafter that add models and controllers….

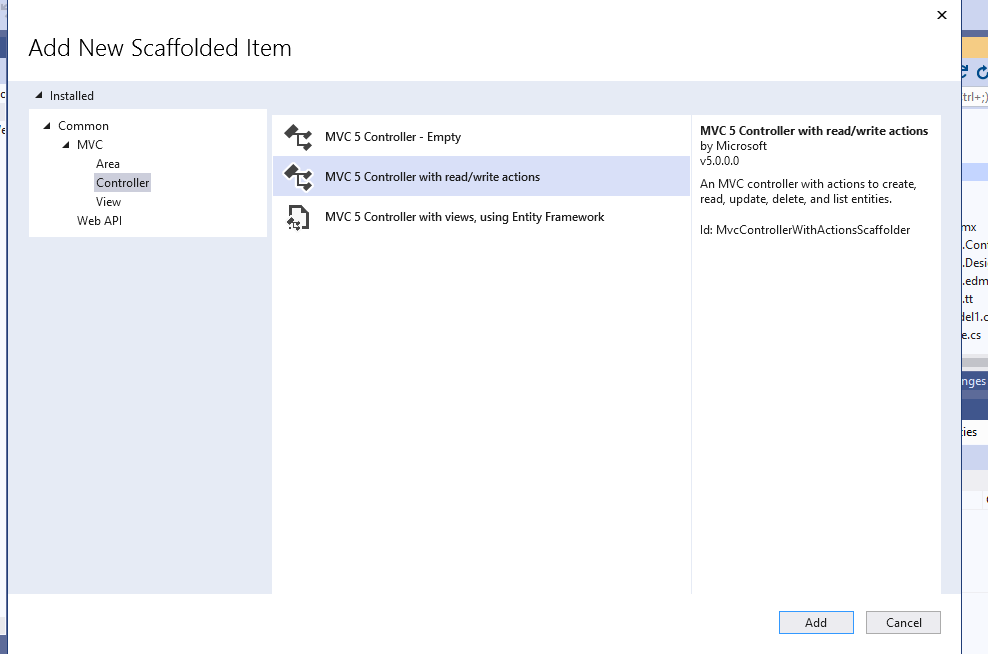


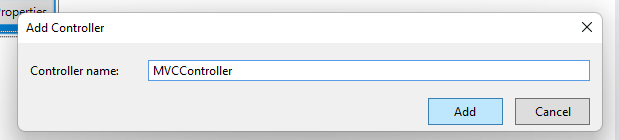




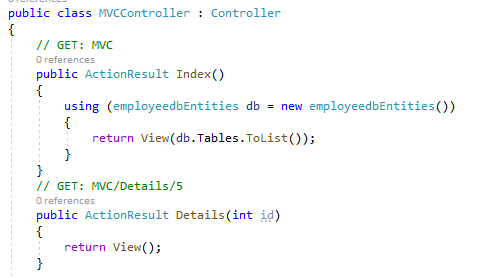


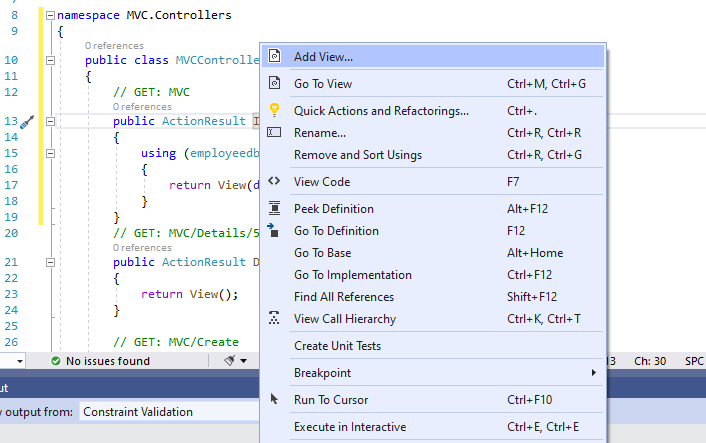


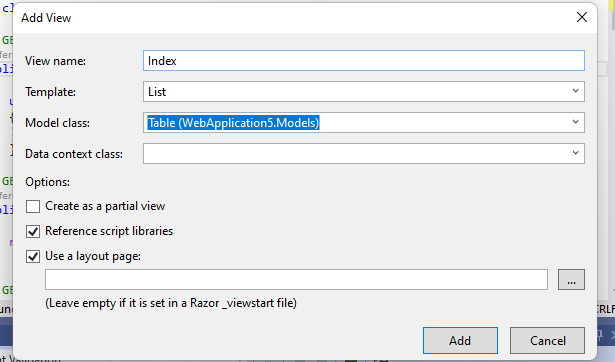


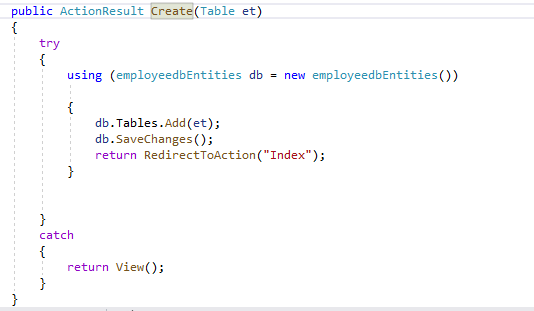


Write the code in index And rc-.add view

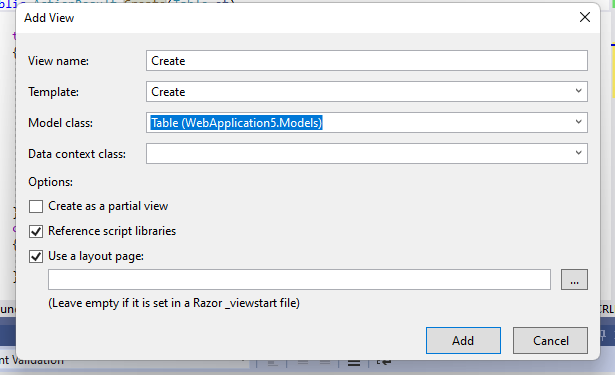




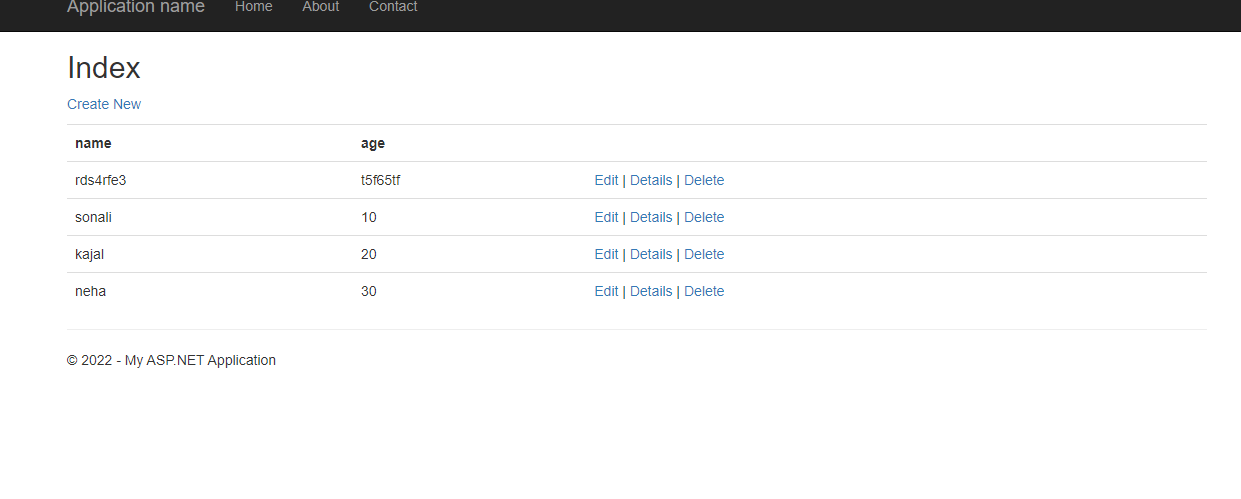




Add view in create->write code…



**Output:**



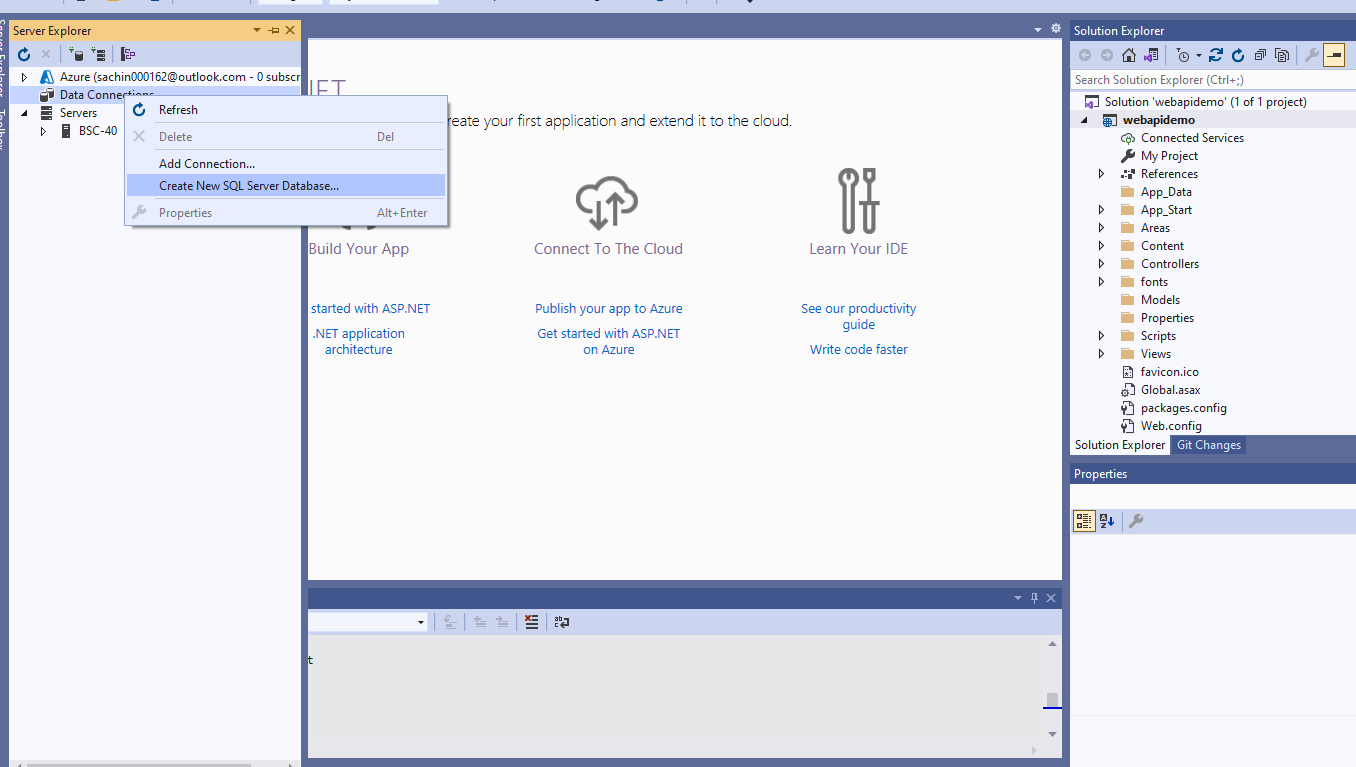
**Practical-5**

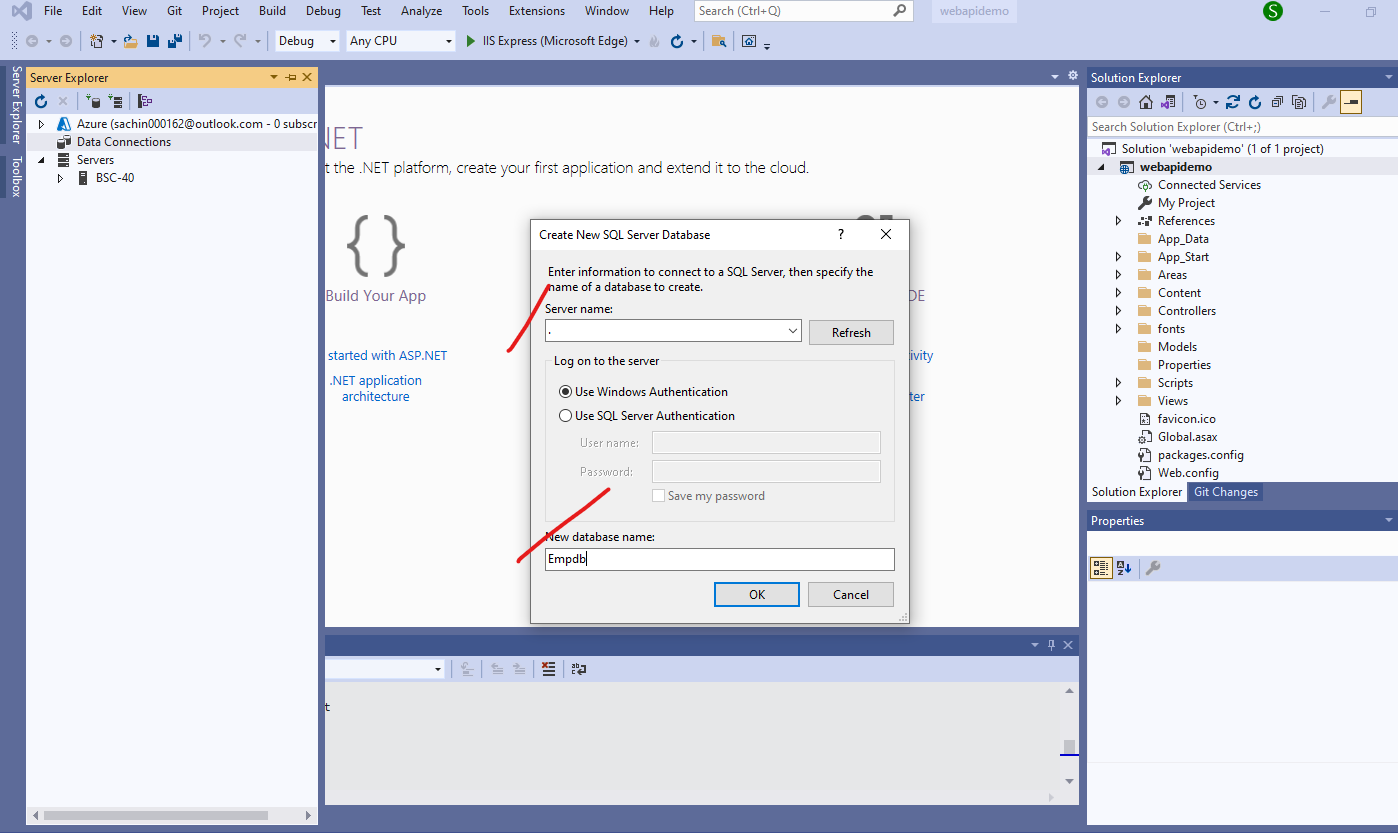
**Aim: Create Service Endpoint.**

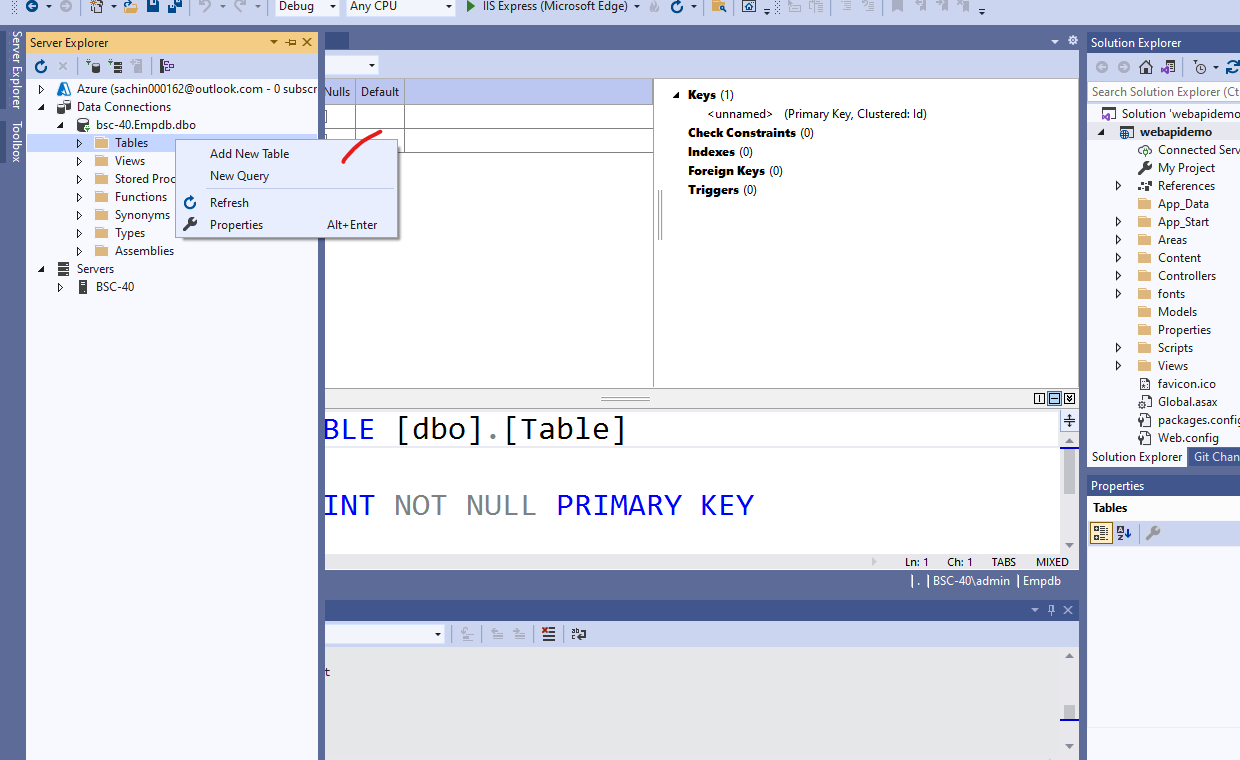
**Practical-6**

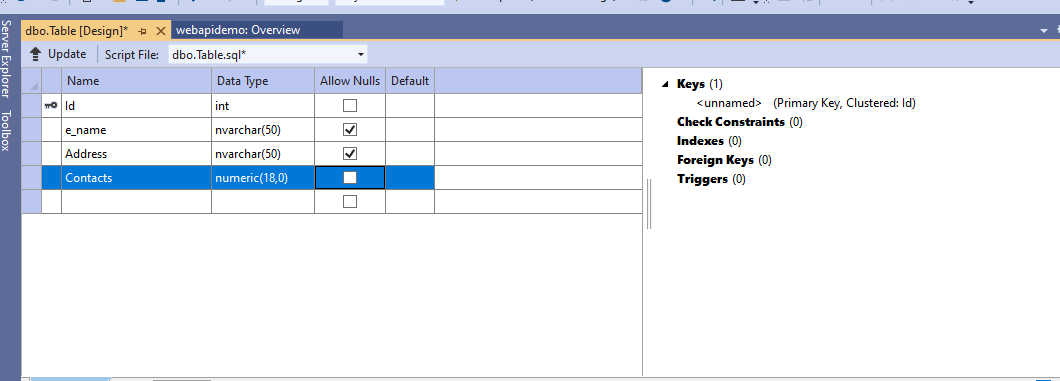
**Aim: RestFul Services using Web API.**

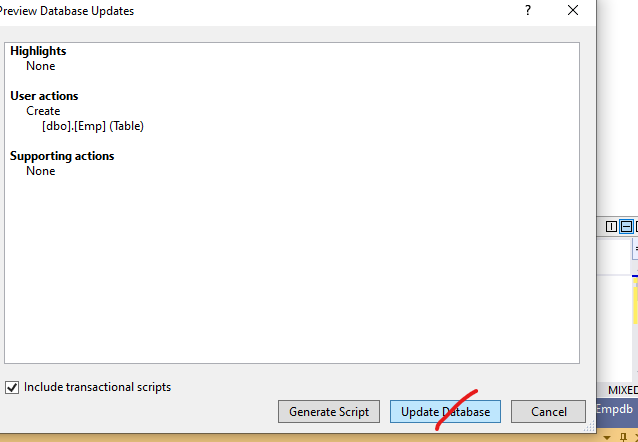
Create new project->select c sharp->select Web API

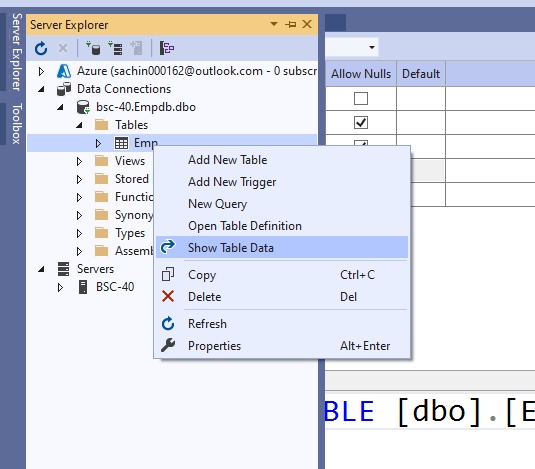


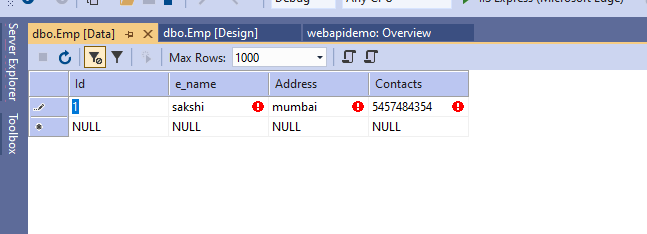




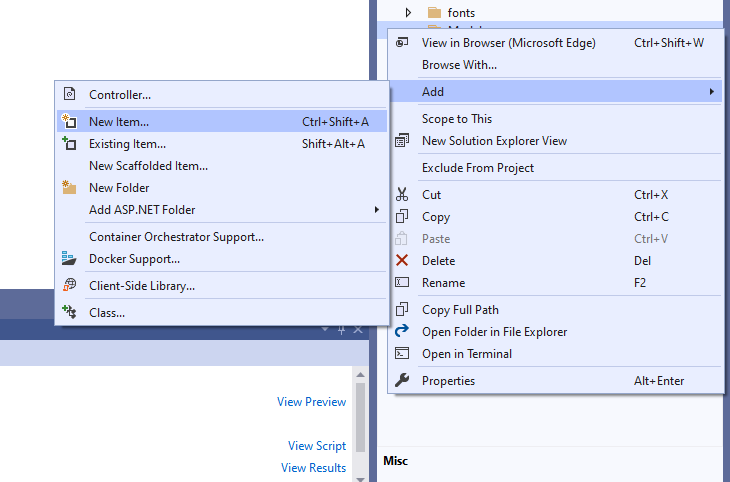


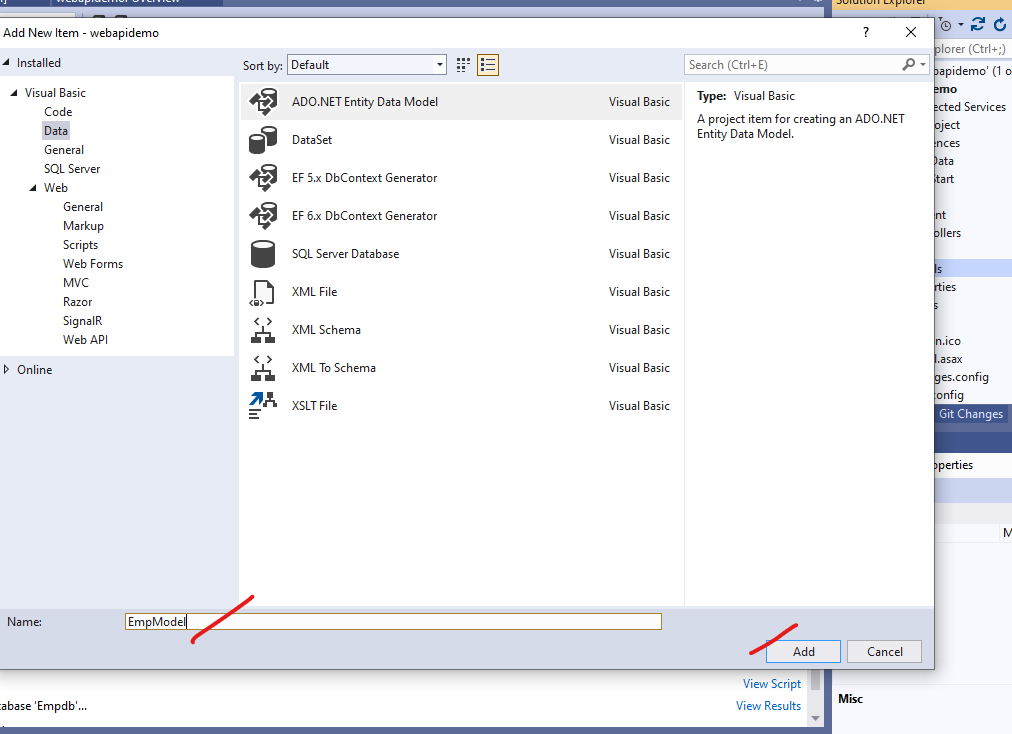


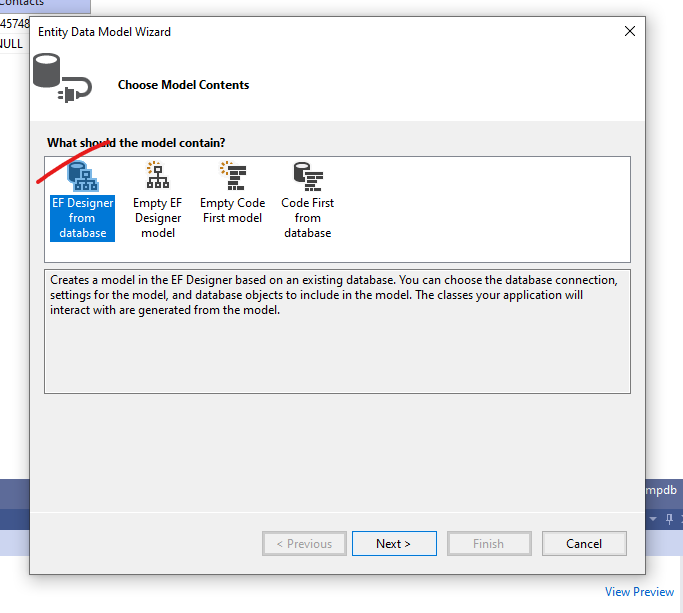


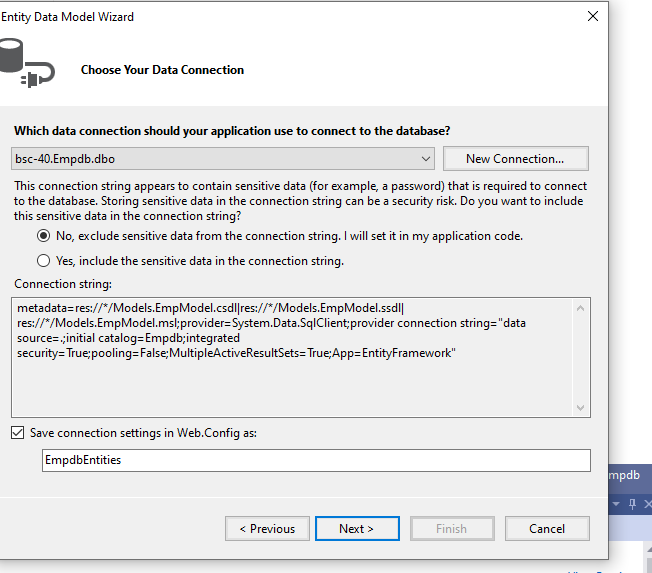


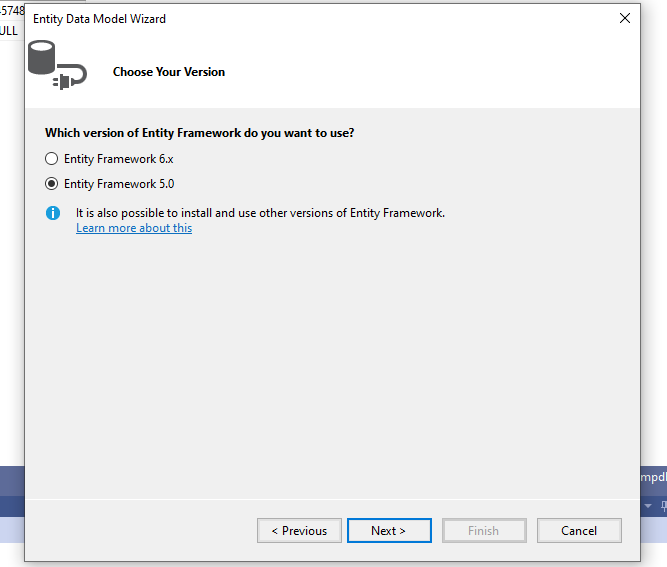
GO TO RIGHT SIDE AND RIGHT CLICK ON MODELS ADD NEW ITEM

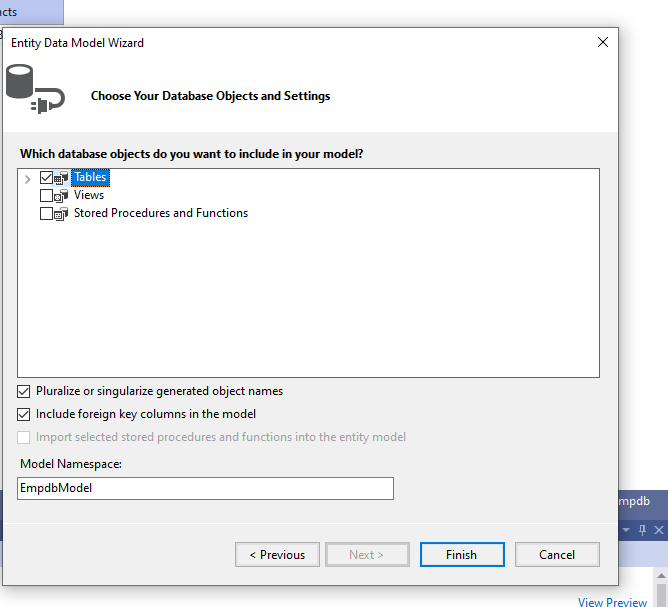


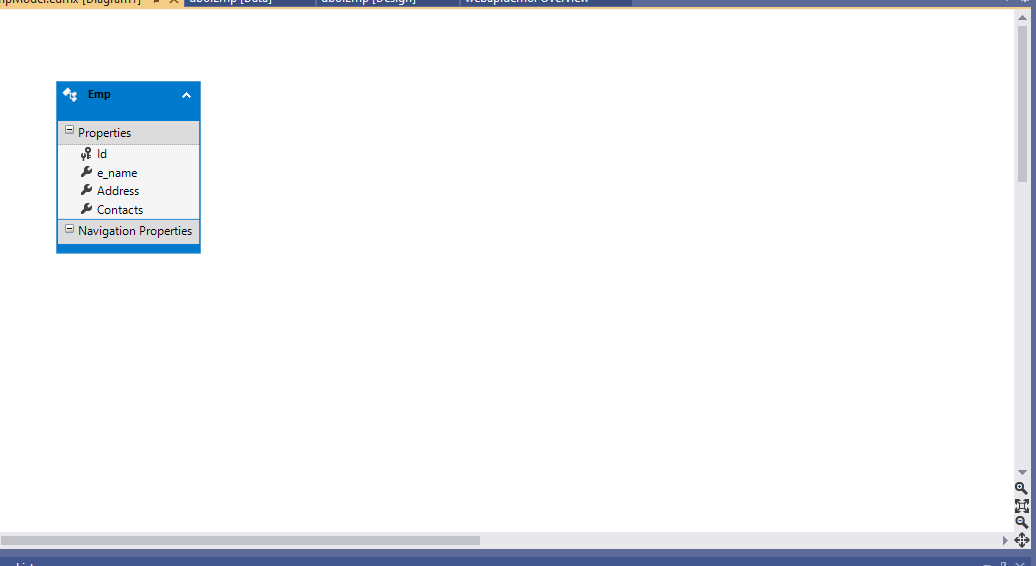




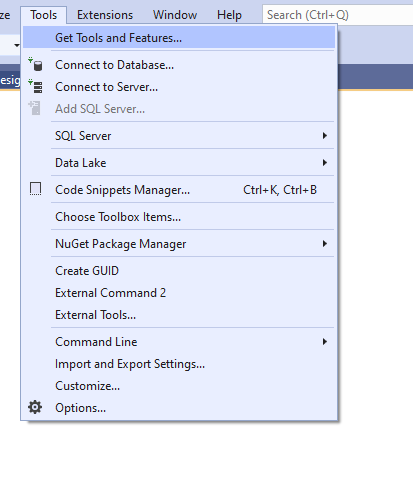




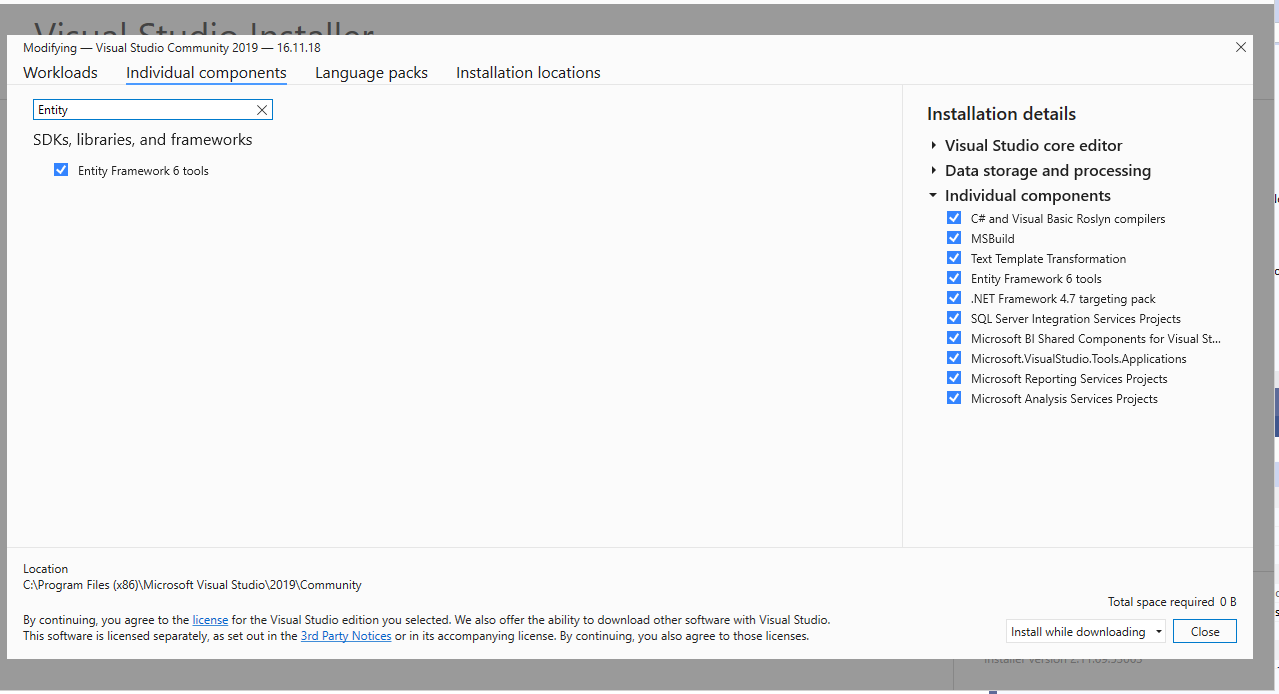


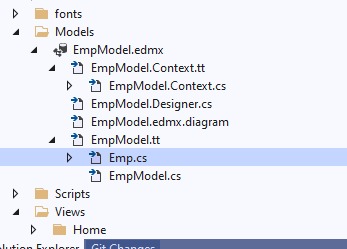


If not showing this then click on tools:

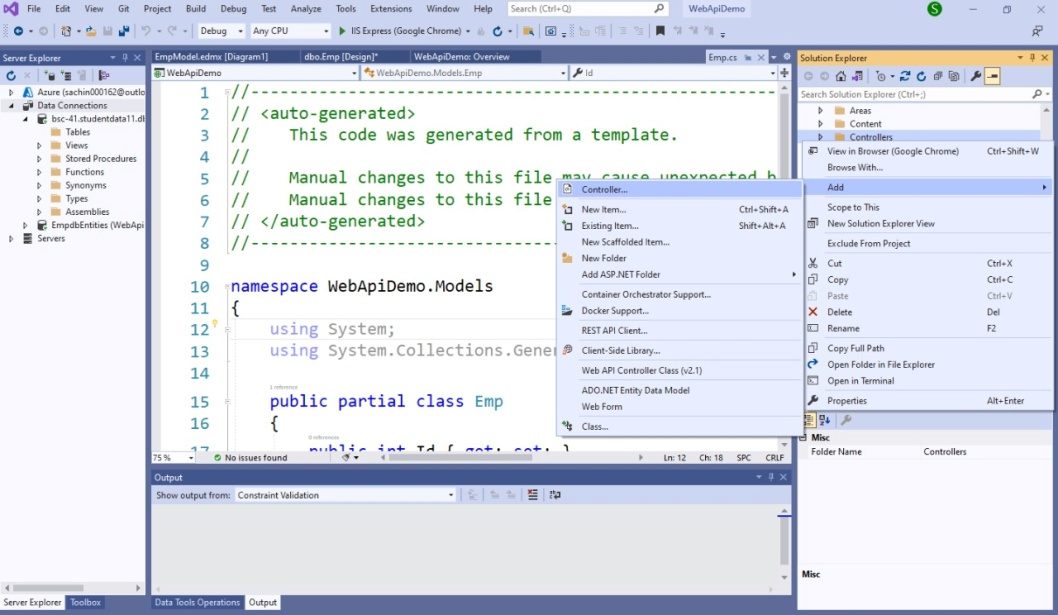


check the entity FRAMEWORK



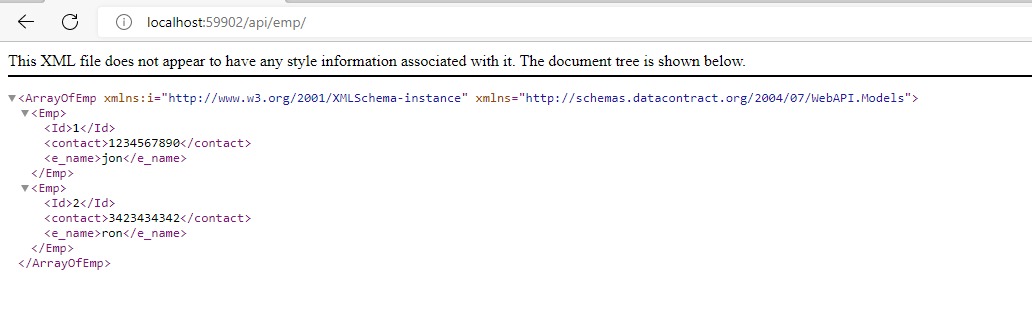


RIGHT CLICK ON CONTROLLER AND ADD CONTROLLER…..



GO TO CONTROLLERS AND ADD CONTROLLER NAME IT AS EMPCONTROLLER…..CLICK ON EMPCONTROLLER.CS DOUBLE CLICK ON IT….WRITE CODE….





Execute on postman and add screenshot here:

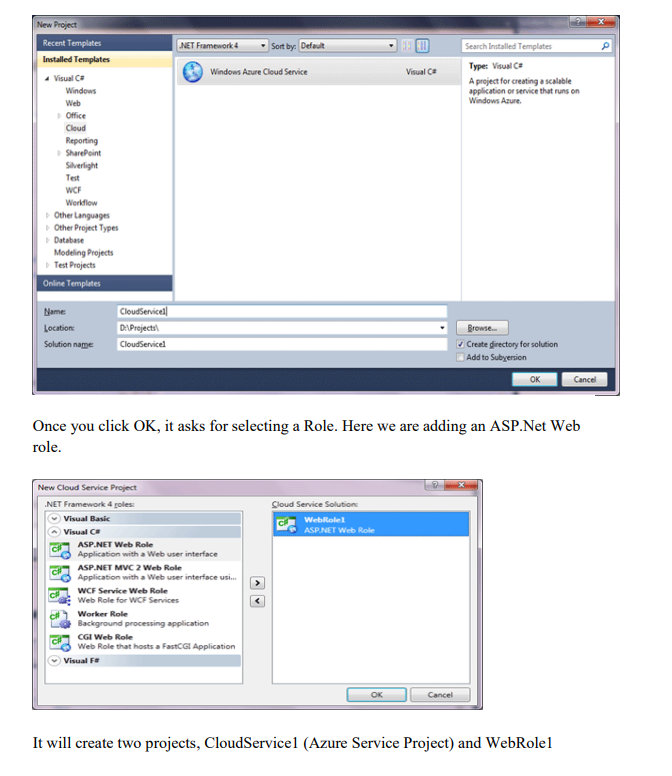
**Practical-07**

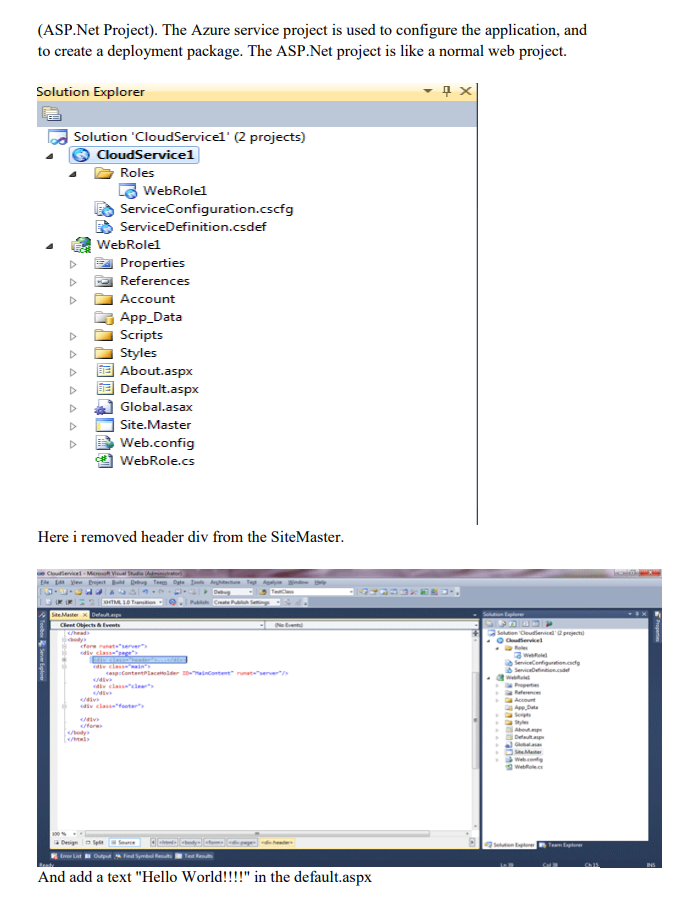
Aim: Develop a Windows Azure Hello World application

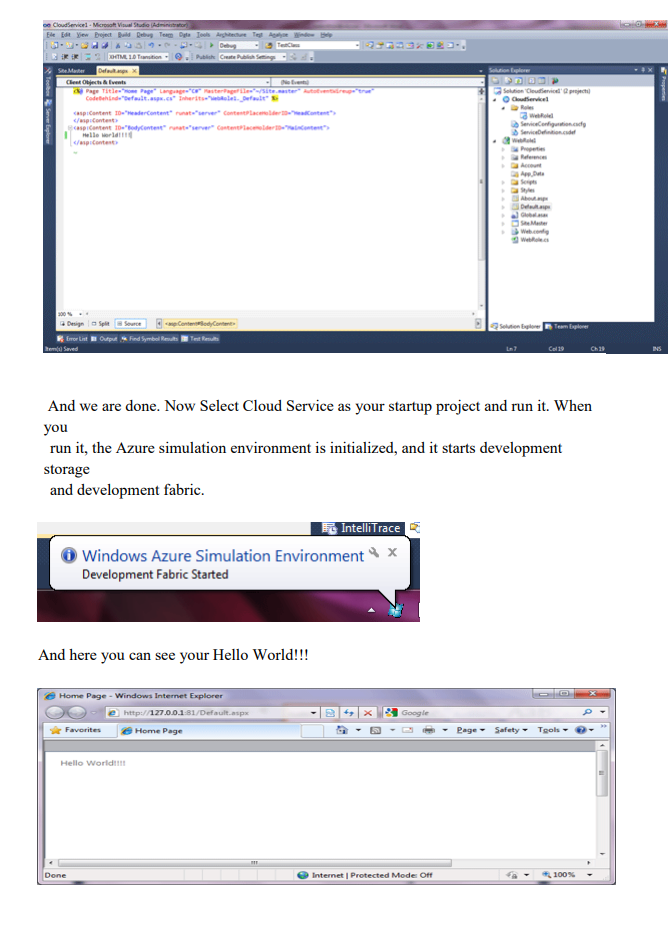
In this experiment, we will see how to create a "Hello World!!!"

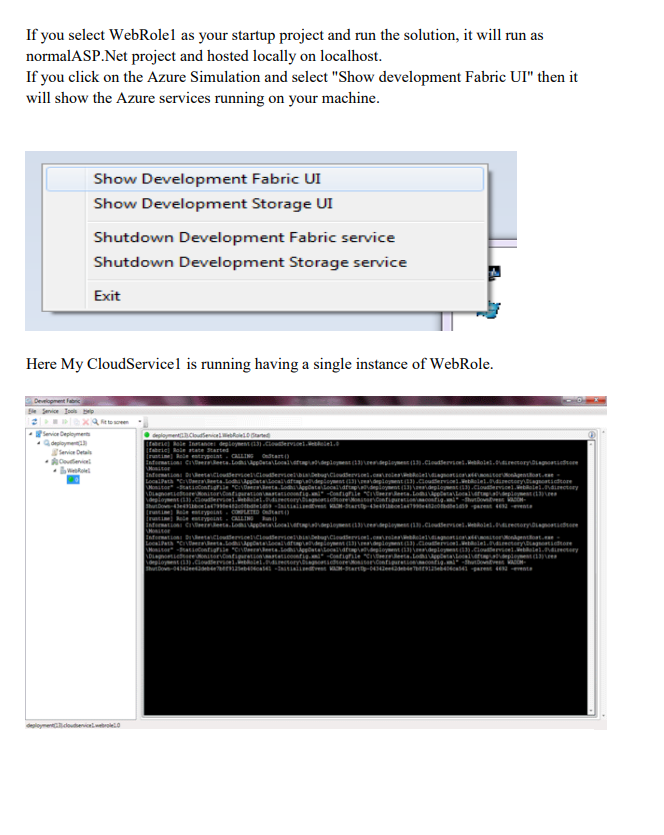
Start Visual Studio and select a new project.

In the cloud template select Windows Azure Cloud Service. Once you click OK, it asks for selecting a Role. Here we are adding an ASP.Net.









**Practical-08**

**Aim:** To Install Google App Engine. Create hello world app and other simple web applications using python/java.

**Procedure**: Use Eclipse to create a Google App Engine (GAE) Java project (hello world example), run it locally, and deploy it to Google App Engine account.

**Tools used:**

1. JDK 1.6

2. Eclipse 3.7 + Google Plugin for Eclipse

3. Google App Engine Java SDK 1.6.3.1 P.S Assume JDK1.6 and Eclipse 3.7 are installed.

1. Install Google Plugin for Eclipse

Read this guide – how to install Google Plugin for Eclipse. If you install the Google App Engine Java SDK together with “Google Plugin for Eclipse“, then go to step 2, Otherwise, get the Google App Engine Java SDK and extract it.

2. Create New Web Application Project In Eclipse toolbar, click on the Google icon, and select “New Web Application Project…”

