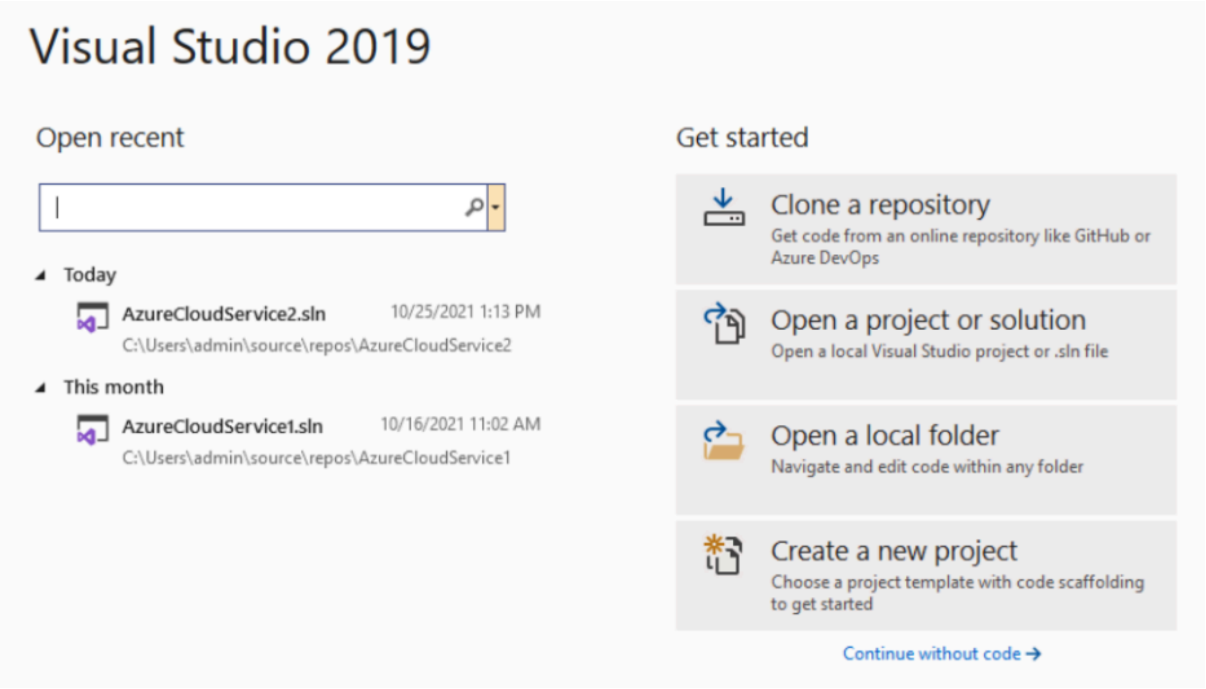
Practical No: 8.1

To develop Application for Windows Azure/Amazon AWS using Windows Azure Platform Training Kit and Visual Studio.

Step 1: - To develop an application for Windows Azure on Visual Studio 2019, install Visual Studio 2019.

Step 2: - Open Visual Studio 2019. Click on "Create New Project".



Visual Studio 2019

Open recent

Today

AzureCloudService2.sln 10/25/2021 1:13 PM
C:\Users\admin\source\repos\AzureCloudService2

This month

AzureCloudService1.sln 10/16/2021 11:02 AM
C:\Users\admin\source\repos\AzureCloudService1

Get started

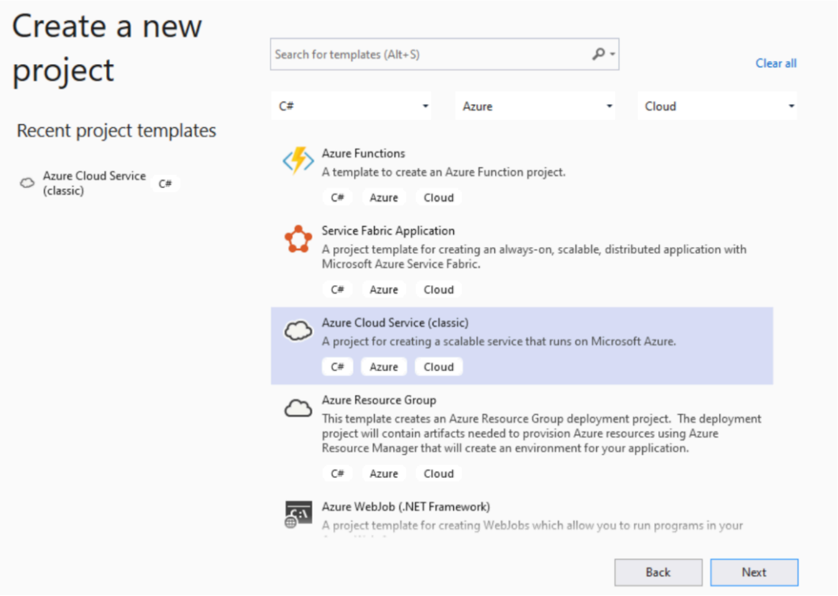
Clone a repository
Get code from an online repository like GitHub or Azure DevOps

Open a project or solution
Open a local Visual Studio project or .sln file

Open a local folder
Navigate and edit code within any folder

Create a new project
Choose a project template with code scaffolding to get started

Step 3: - After clicking on Create new project, a New Project Window will open



Create a new project

Recent project templates

Azure Cloud Service (classic) C#

Search for templates (Alt+S)

C# | Azure | Cloud

Azure Functions
A template to create an Azure Function project.
C# Azure Cloud

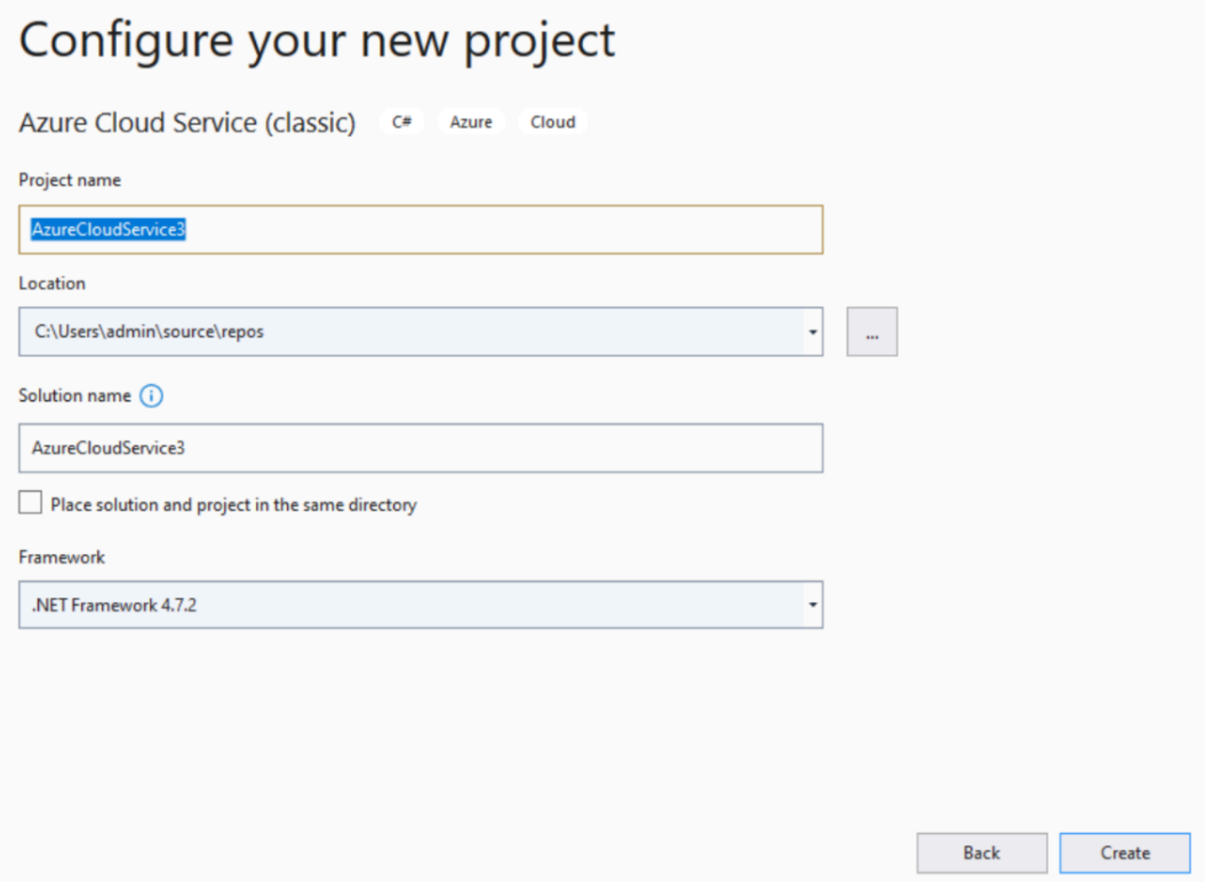
Service Fabric Application
A project template for creating an always-on, scalable, distributed application with Microsoft Azure Service Fabric.
C# Azure Cloud

Azure Cloud Service (classic)
A project for creating a scalable service that runs on Microsoft Azure.
C# Azure Cloud

Azure Resource Group
This template creates an Azure Resource Group deployment project. The deployment project will contain artifacts needed to provision Azure resources using Azure Resource Manager that will create an environment for your application.
C# Azure Cloud

Azure WebJob (.NET Framework)
A project template for creating WebJobs which allow you to run programs in your

Step 4: - In above window, choose Language as C#, Platform as Azure, Project as Cloud. Then select Azure Cloud Service(classic) option. Click Next. Configure project window will appear.



Configure your new project

Azure Cloud Service (classic) C# Azure Cloud

Project name: AzureCloudService3

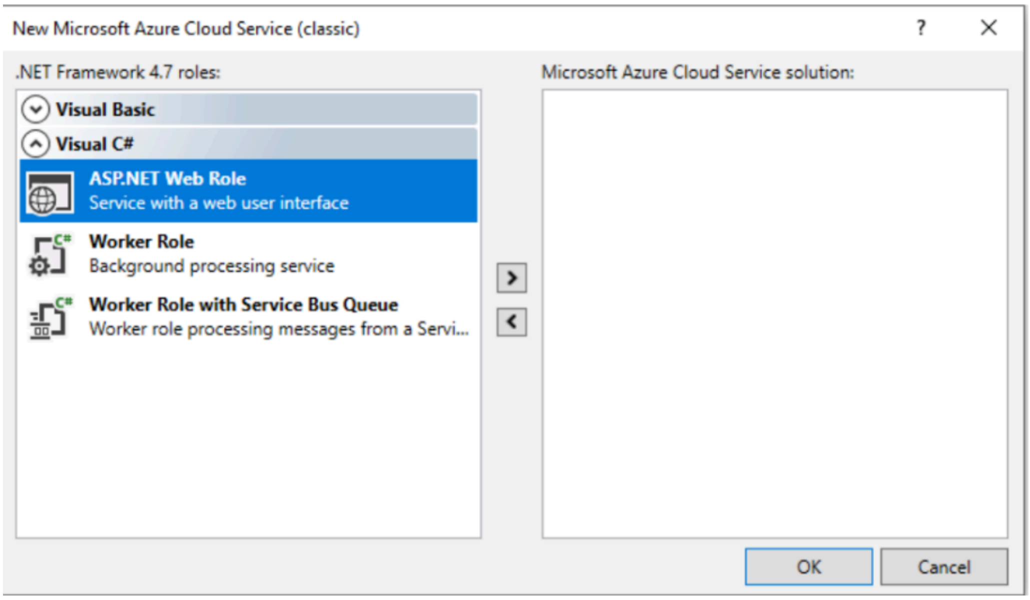
Location: C:\Users\admin\source\repos

Solution name: AzureCloudService3

☐ Place solution and project in the same directory

Framework: .NET Framework 4.7.2

Step 5: - Enter Project name(AzureCloudService3), then click on Create button. New Microsoft Azure Cloud Service(Classic) window will appear as below. Select Visual C# -> ASP.net Web Role.



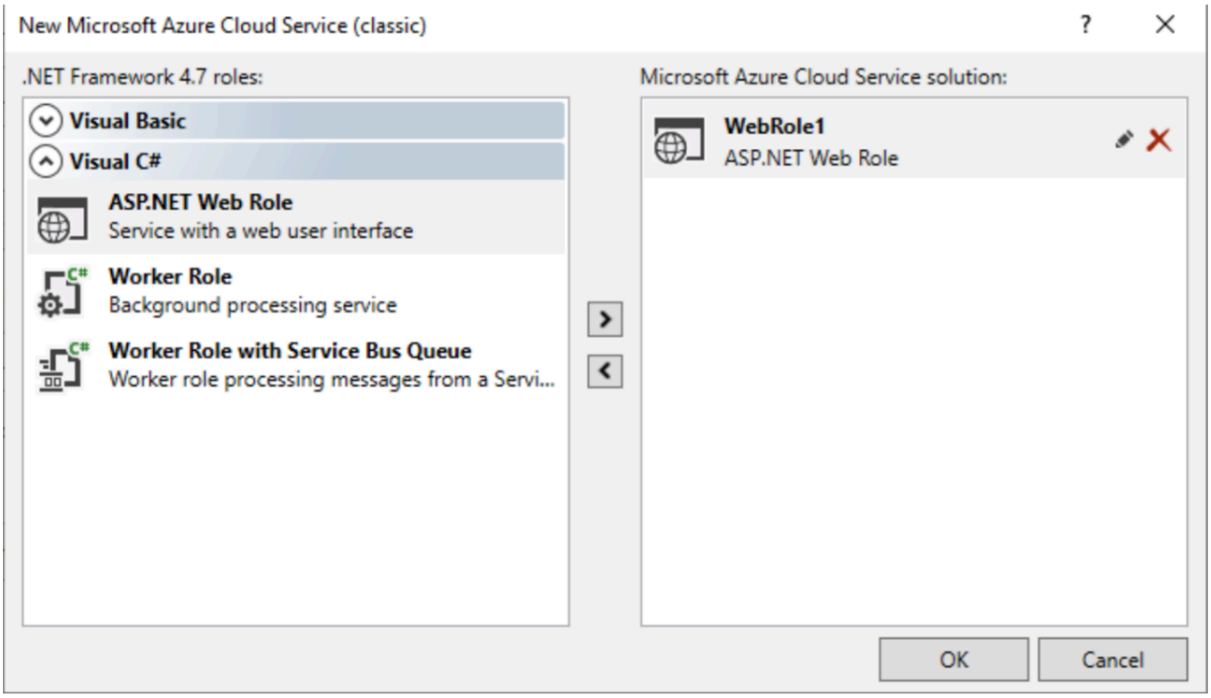
New Microsoft Azure Cloud Service (classic)

.NET Framework 4.7 roles:

- Visual Basic
- Visual C#
  - ASP.NET Web Role — Service with a web user interface
  - Worker Role — Background processing service
  - Worker Role with Service Bus Queue — Worker role processing messages from a Servi...

Microsoft Azure Cloud Service solution:

Step 6: - Click on the “>” button to add a role to the solution. The window will appear.



New Microsoft Azure Cloud Service (classic)

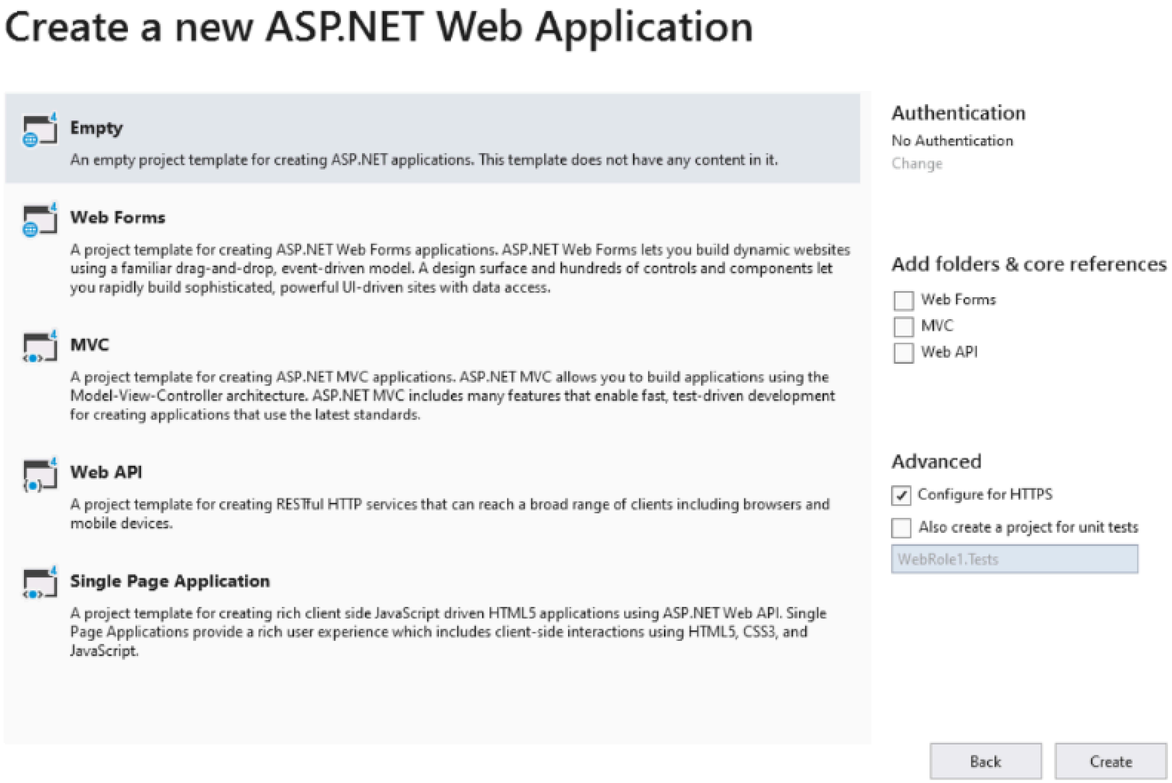
.NET Framework 4.7 roles:

- Visual Basic
- Visual C#
- **ASP.NET Web Role** Service with a web user interface
- **Worker Role** Background processing service
- **Worker Role with Service Bus Queue** Worker role processing messages from a Servi...

Microsoft Azure Cloud Service solution:

- **WebRole1** ASP.NET Web Role

Step 7: - Click on the OK button. The window will appear.



Create a new ASP.NET Web Application

Empty
An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms
A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC
A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API
A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

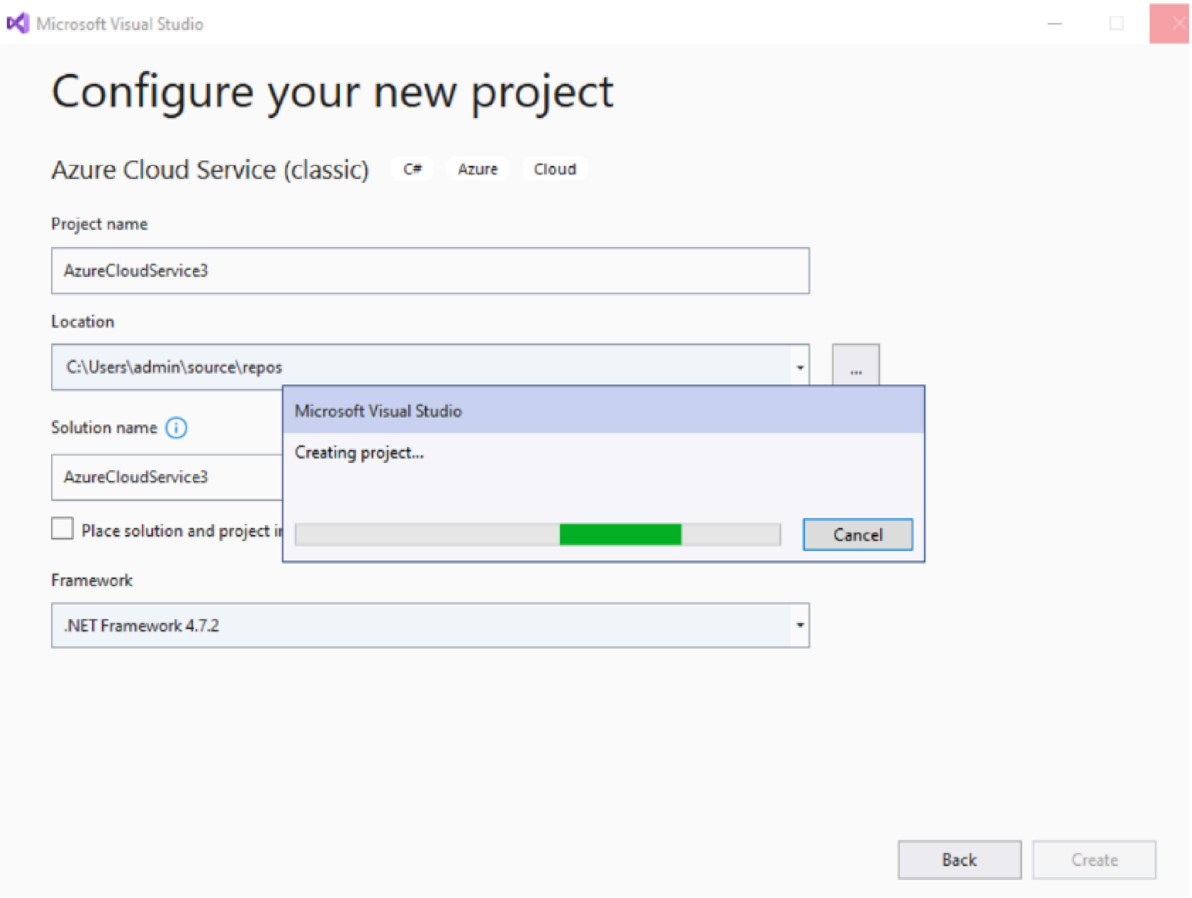
Single Page Application
A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication
No Authentication
Change

Add folders & core references
☐ Web Forms
☐ MVC
☐ Web API

Advanced
☑ Configure for HTTPS
☐ Also create a project for unit tests
WebRole1.Tests

Step 8: - Select Empty Option for creating empty project template and then click on Create Button. The window will appear.



Configure your new project

Azure Cloud Service (classic)

Project name: AzureCloudService3

Location: C:\Users\admin\source\repos

Solution name: AzureCloudService3

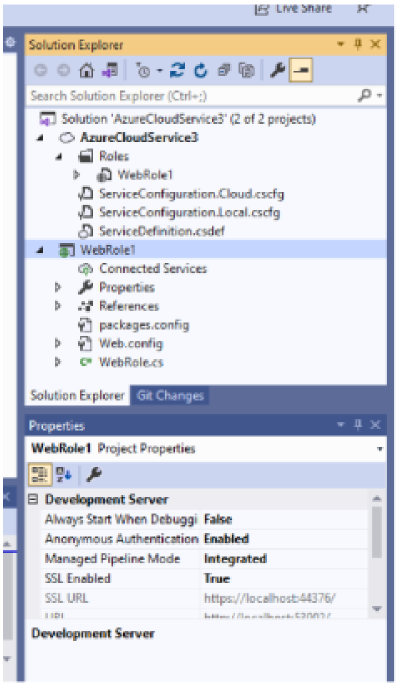
□ Place solution and project i

Framework: .NET Framework 4.7.2

Microsoft Visual Studio

Creating project...

Step 9: - Right Click on WebRole1 in Solution Explorer Window.



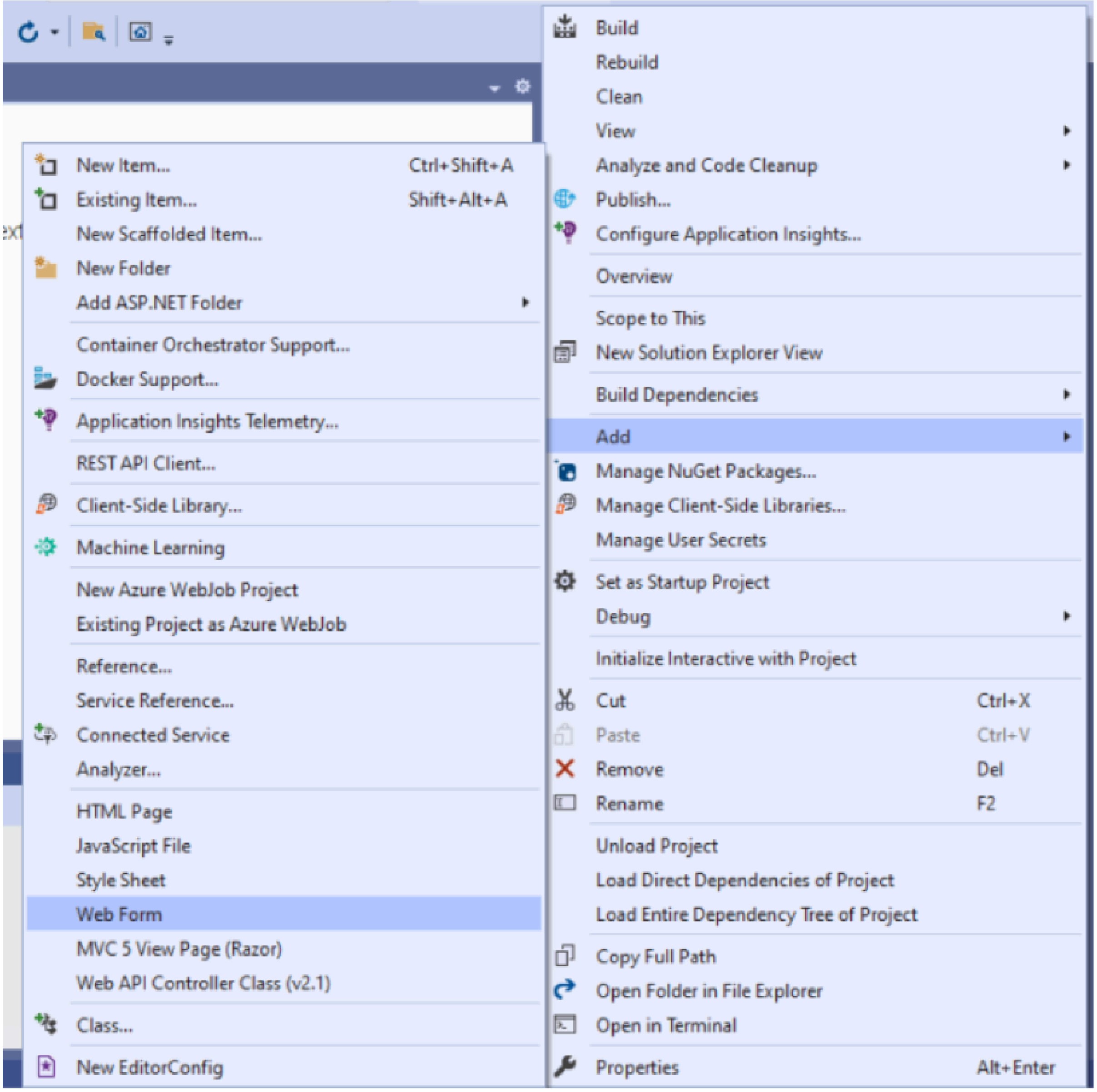
WebRole1 Project Properties

⊟ Development Server

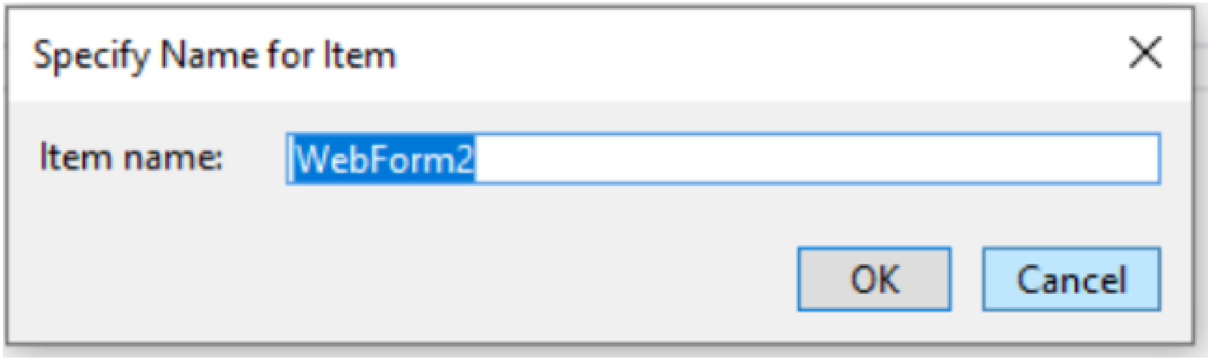
Always Start When Debuggi	False
Anonymous Authentication	Enabled
Managed Pipeline Mode	Integrated
SSL Enabled	True
SSL URL	https://localhost:44376/

Development Server

Step 10: - Then click on Add Button. Select Web Form



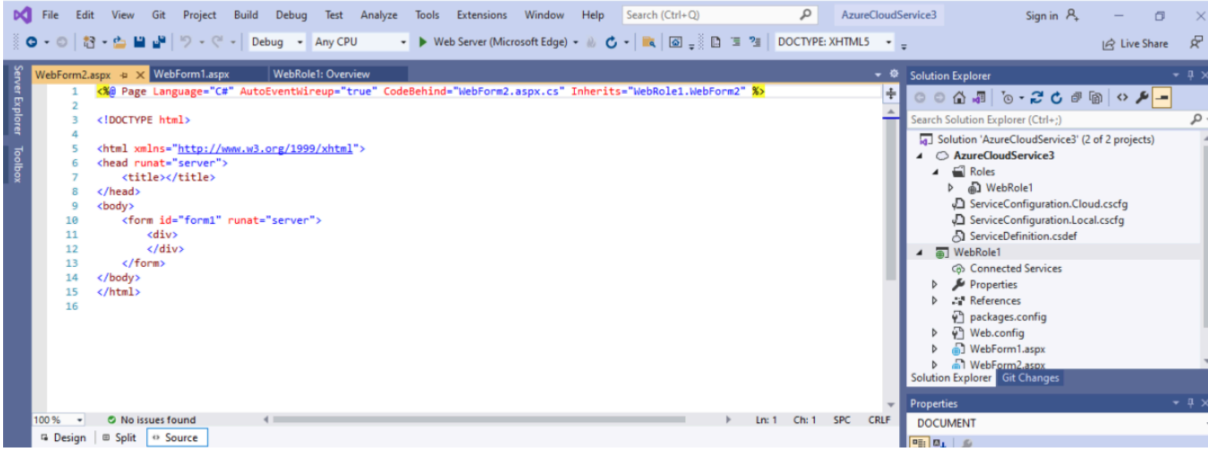
Step 11: - Give Name to Web Form



Specify Name for Item

Item name: WebForm2

Step 12: - Click on OK Button. The window will appear.



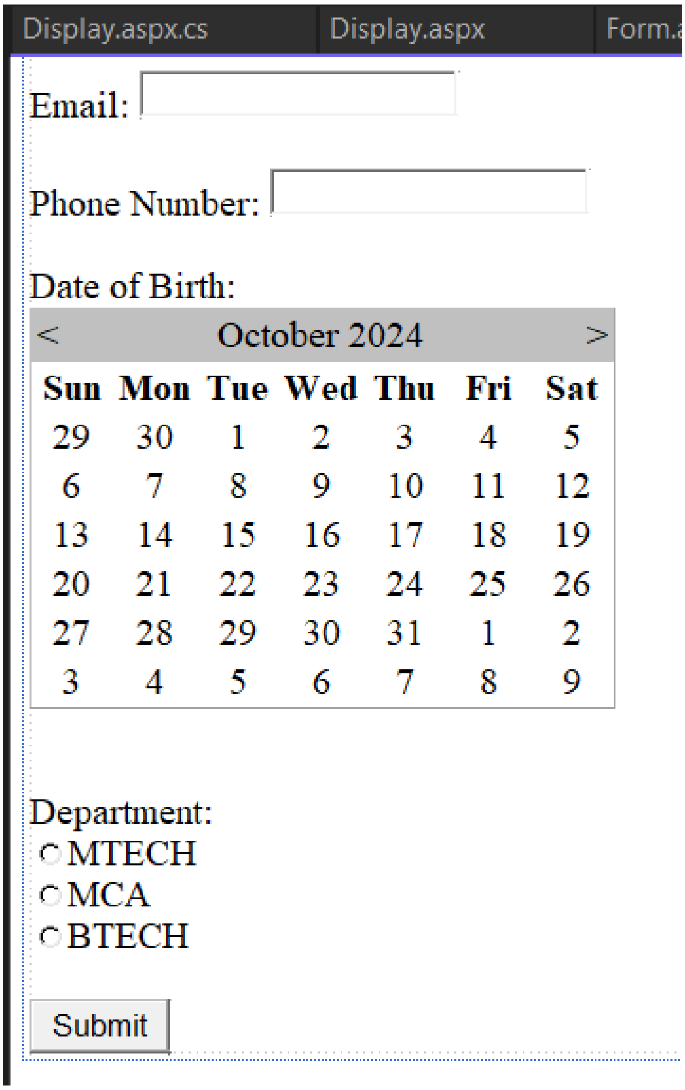
```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="WebRole1.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
        </div>
    </form>
</body>
</html>
```

Step 13: - Write the following code in .aspx and .aspx.cs and then click on Debug and Execute the project.

Form.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Form.aspx.cs" Inherits="WebRole1.Form" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Simple Form</title>
</head>
<body>
    <form id="form1" runat="server">
       <div>
           <h2>Enter Your Details</h2>
           <label for="txtFullName">Full Name:</label>
           <asp:TextBox ID="txtFullName" runat="server"></asp:TextBox><br /><br />
           <label for="txtEmail">Email:</label>
           <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox><br /><br />
           <label for="txtPhone">Phone Number:</label>
           <asp:TextBox ID="txtPhone" runat="server"></asp:TextBox><br /><br />
           <label for="calendarDOB">Date of Birth:</label>
```

```
            <asp:Calendar ID="calendarDOB" runat="server"></asp:Calendar><br
/><br />
            <label>Department:</label><br />
            <asp:RadioButton ID="rbMTECH" runat="server"
GroupName="Department" Text="MTECH" /><br />
            <asp:RadioButton ID="rbMCA" runat="server"
GroupName="Department" Text="MCA" /><br />
            <asp:RadioButton ID="rbBTECH" runat="server"
GroupName="Department" Text="BTECH" /><br /><br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click"/>
        </div>
    </form>
</body>
</html>
```



Email:

Phone Number:

Date of Birth:

<			October 2024			>
Sun	**Mon**	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Department:

○ MTECH
○ MCA
○ BTECH

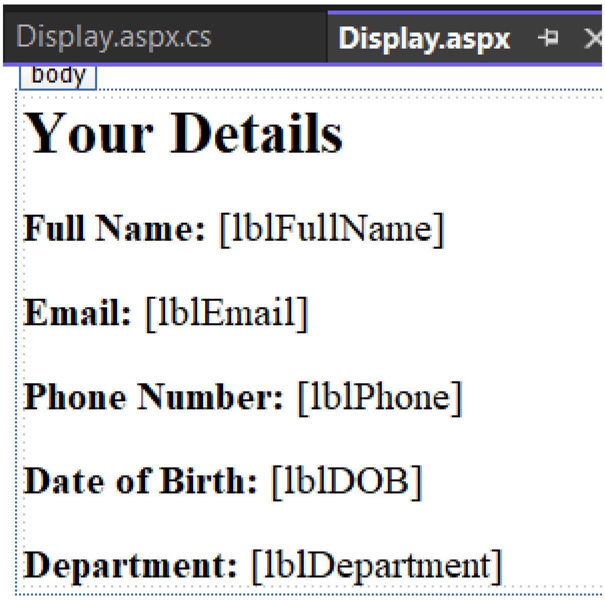
Form.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebRole1
{
   public partial class Form : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
      }
      protected void btnSubmit_Click(object sender, EventArgs e)
      {
         // Retrieve values from controls
         string fullName = txtFullName.Text;
         string email = txtEmail.Text;
         string phone = txtPhone.Text;
         string dob = calendarDOB.SelectedDate != DateTime.MinValue ?
calendarDOB.SelectedDate.ToShortDateString() : "";
         string department = "";
         if (rbMTECH.Checked)
            department = "MTECH";
         else if (rbMCA.Checked)
            department = "MCA";
         else if (rbBTECH.Checked)
            department = "BTECH";
         // Redirect to Display.aspx with query string
         Response.Redirect("Display.aspx?FullName=" +
HttpUtility.UrlEncode(fullName) +
                        "&Email=" + HttpUtility.UrlEncode(email) +
                        "&Phone=" + HttpUtility.UrlEncode(phone) +
                        "&DOB=" + HttpUtility.UrlEncode(dob) +
                        "&Department=" + HttpUtility.UrlEncode(department));
      }
```

```
    }
}
```

Display.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Display.aspx.cs" Inherits="WebRole1.Display" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Display Details</title>
</head>
<body>
     <form id="form1" runat="server">
        <div>
            <h2>Your Details</h2>
            <p><strong>Full Name:</strong> <asp:Label ID="lblFullName"
runat="server"></asp:Label></p>
            <p><strong>Email:</strong> <asp:Label ID="lblEmail"
runat="server"></asp:Label></p>
            <p><strong>Phone Number:</strong> <asp:Label ID="lblPhone"
runat="server"></asp:Label></p>
            <p><strong>Date of Birth:</strong> <asp:Label ID="lblDOB"
runat="server"></asp:Label></p>
            <p><strong>Department:</strong> <asp:Label ID="lblDepartment"
runat="server"></asp:Label></p>
        </div>
    </form>
</body>
</html>
```



Your Details

Full Name: [lblFullName]

Email: [lblEmail]

Phone Number: [lblPhone]

Date of Birth: [lblDOB]

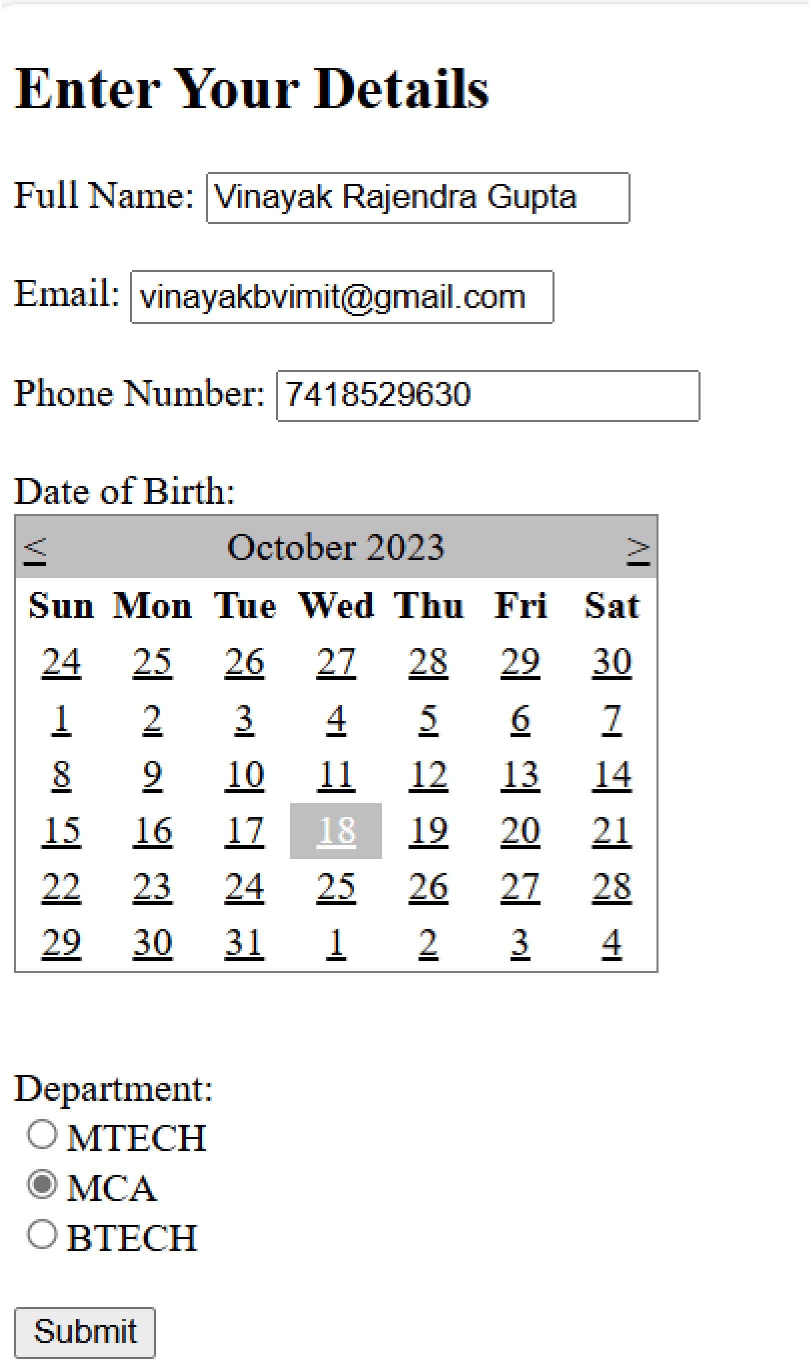
Department: [lblDepartment]

Display.aspx.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace WebRole1
{
   public partial class Display : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            // Retrieve data from query string
            lblFullName.Text = Request.QueryString["FullName"] ?? "No Data";
            lblEmail.Text = Request.QueryString["Email"] ?? "No Data";
            lblPhone.Text = Request.QueryString["Phone"] ?? "No Data";
            lblDOB.Text = Request.QueryString["DOB"] ?? "No Data";
            lblDepartment.Text = Request.QueryString["Department"] ?? "No Data";
         }
      }
```

```
    }
}
```

Output: -



Enter Your Details

Full Name: Vinayak Rajendra Gupta

Email: vinayakbvimit@gmail.com

Phone Number: 7418529630

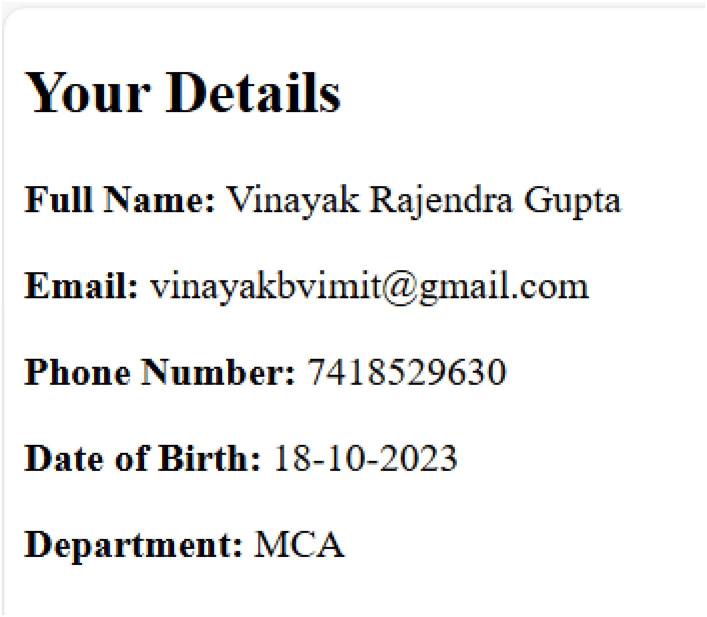
Date of Birth:

< October 2023 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
24	25	26	27	28	29	30
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Department:

○ MTECH
● MCA
○ BTECH



Your Details

Full Name: Vinayak Rajendra Gupta

Email: vinayakbvimit@gmail.com

Phone Number: 7418529630

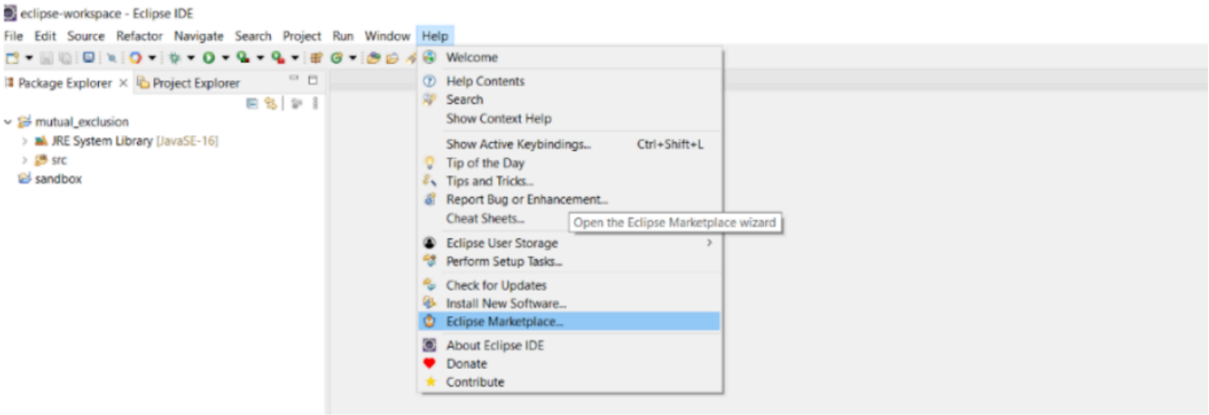
Date of Birth: 18-10-2023

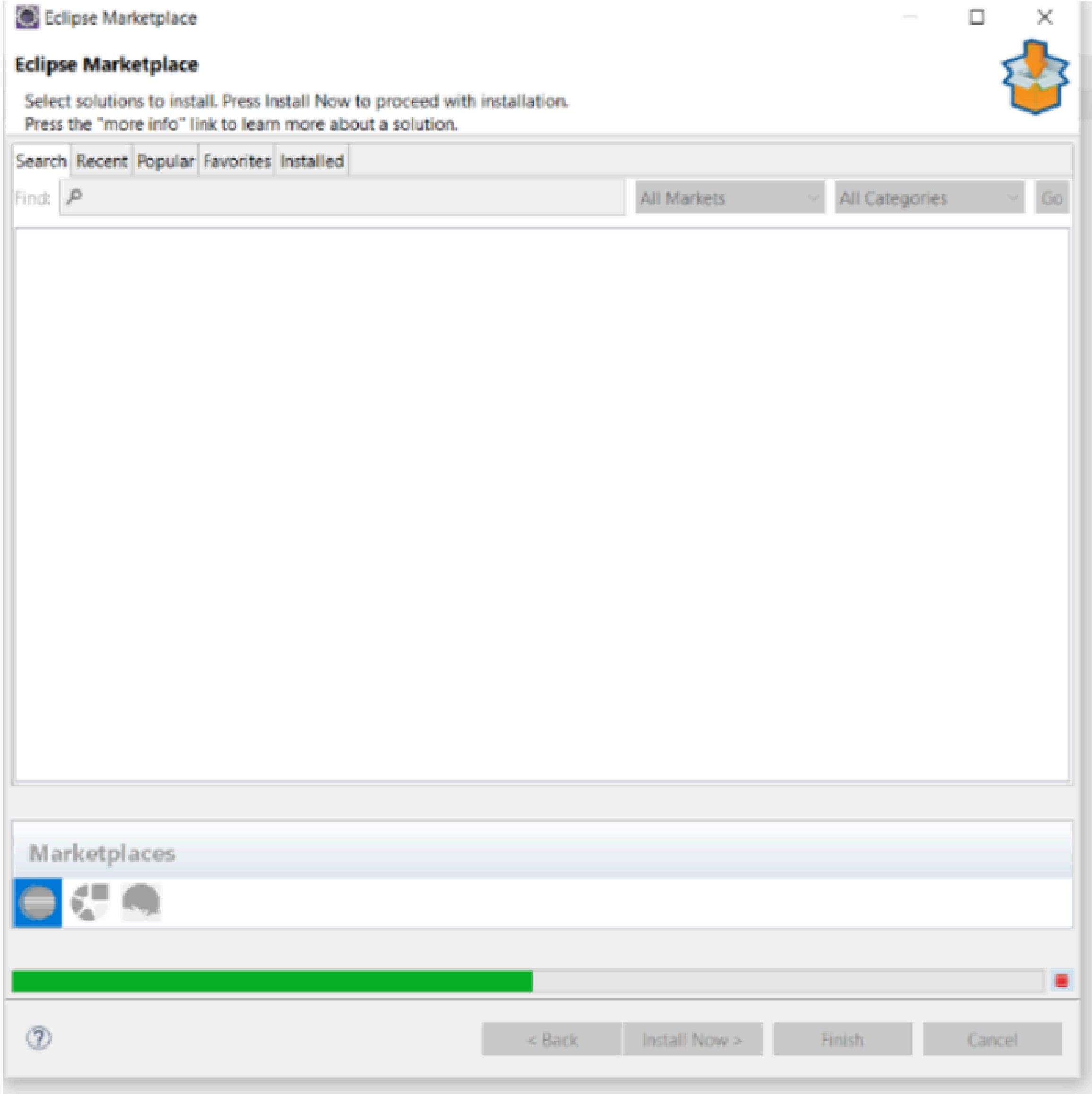
Department: MCA

Practical No: - 8.2

Write a Program to developed an Application using Google App Engine by using Eclipse IDE

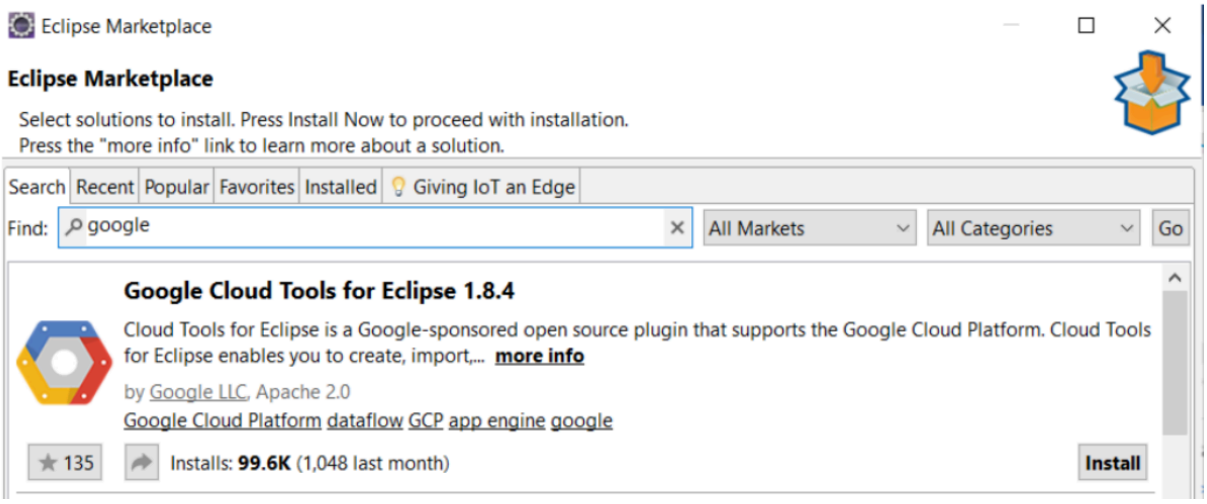
Step 1: - Click on Help. Eclipse Marketplace





Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Step 2: - Once the Eclipse Marketplace window appear in search textbox write "google" click on enter



Eclipse Marketplace

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Find: google

Google Cloud Tools for Eclipse 1.8.4

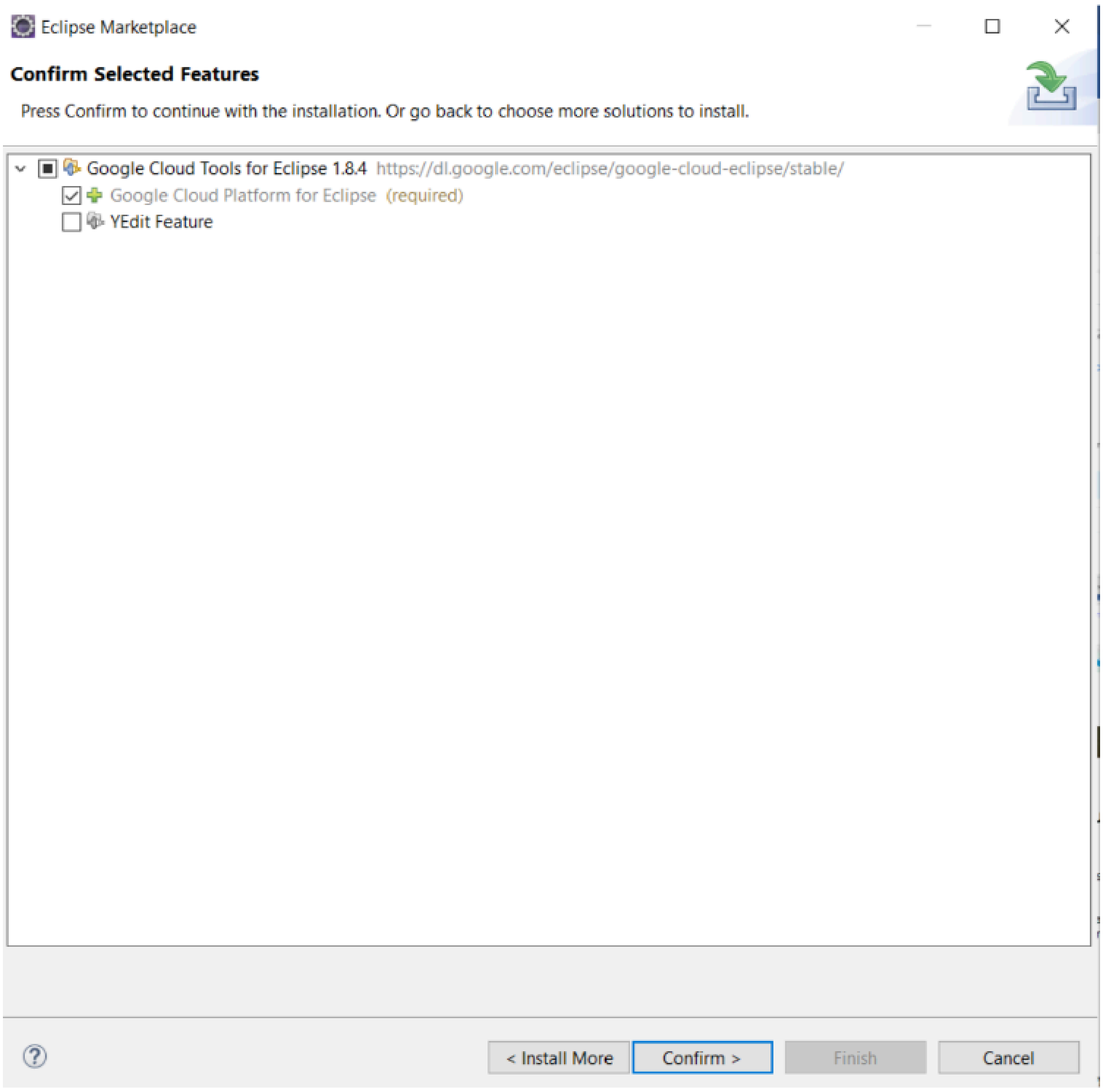
Cloud Tools for Eclipse is a Google-sponsored open source plugin that supports the Google Cloud Platform. Cloud Tools for Eclipse enables you to create, import,... **more info**

by Google LLC, Apache 2.0

Google Cloud Platform dataflow GCP app engine google

★ 135 Installs: **99.6K** (1,048 last month)

Step 3: - Click on confirm

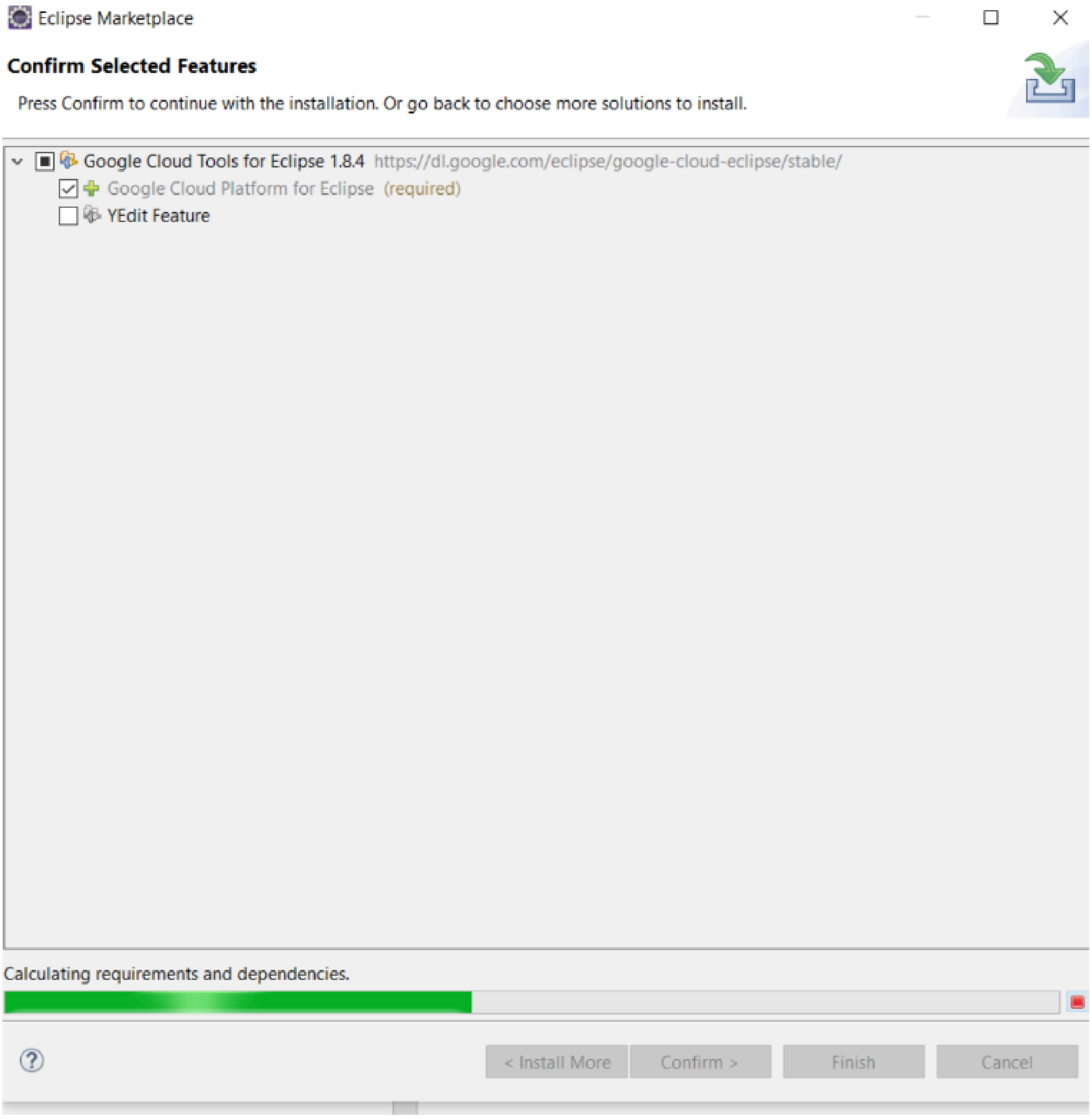


Eclipse Marketplace

Confirm Selected Features

Press Confirm to continue with the installation. Or go back to choose more solutions to install.

- ■ Google Cloud Tools for Eclipse 1.8.4 https://dl.google.com/eclipse/google-cloud-eclipse/stable/
  - ☑ Google Cloud Platform for Eclipse (required)
  - □ YEdit Feature



Eclipse Marketplace

Confirm Selected Features

Press Confirm to continue with the installation. Or go back to choose more solutions to install.

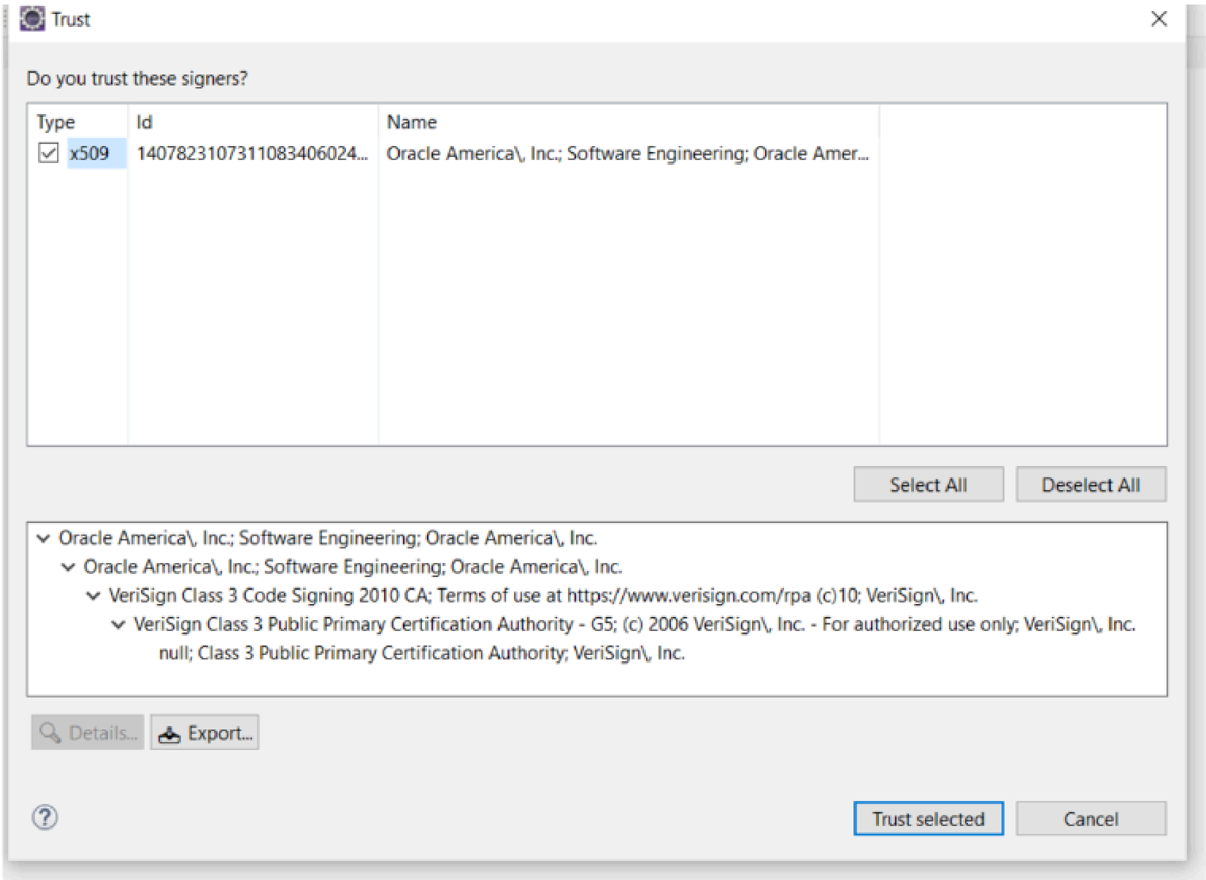
■ Google Cloud Tools for Eclipse 1.8.4 https://dl.google.com/eclipse/google-cloud-eclipse/stable/

☑ Google Cloud Platform for Eclipse (required)

□ YEdit Feature

Calculating requirements and dependencies.

Step 4: - Once the below window appear click on trust selected



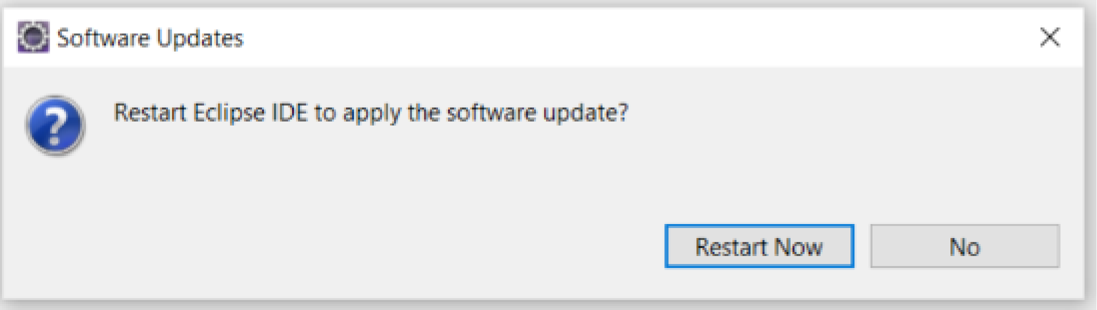
Trust

Do you trust these signers?

Type	Id	Name
☑ x509	1407823107311083406024...	Oracle America\, Inc.; Software Engineering; Oracle Amer...

- Oracle America\, Inc.; Software Engineering; Oracle America\, Inc.
  - Oracle America\, Inc.; Software Engineering; Oracle America\, Inc.
    - VeriSign Class 3 Code Signing 2010 CA; Terms of use at https://www.verisign.com/rpa (c)10; VeriSign\, Inc.
      - VeriSign Class 3 Public Primary Certification Authority - G5; (c) 2006 VeriSign\, Inc. - For authorized use only; VeriSign\, Inc.
        - null; Class 3 Public Primary Certification Authority; VeriSign\, Inc.

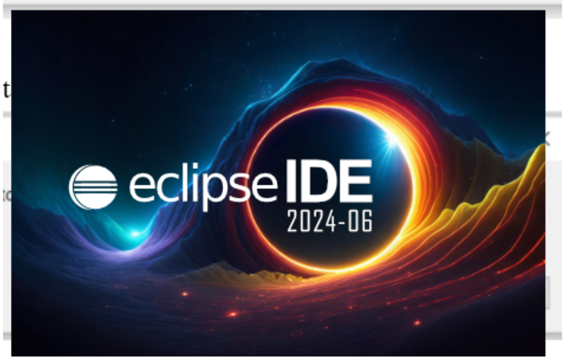
Step 5: - Click on restart now



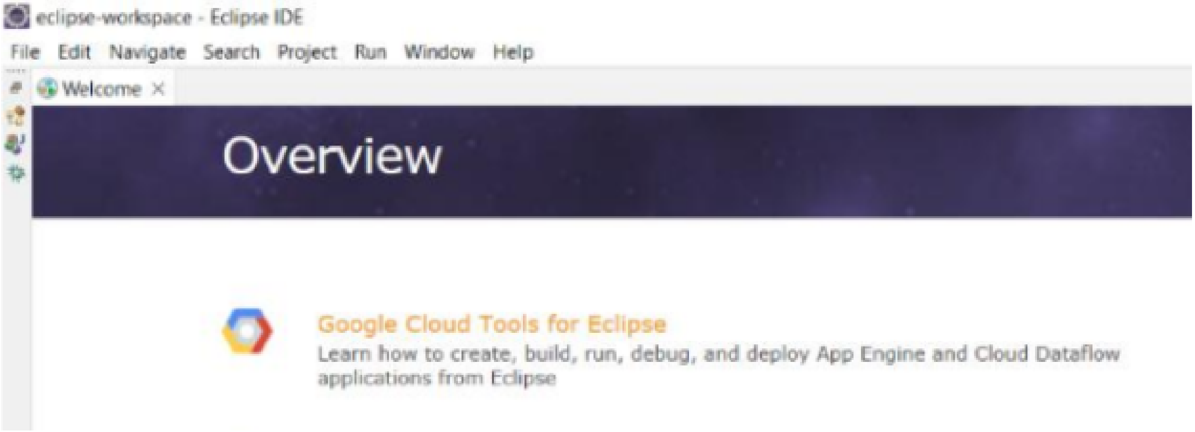
Software Updates

Restart Eclipse IDE to apply the software update?

Step 6: - Below window will appear



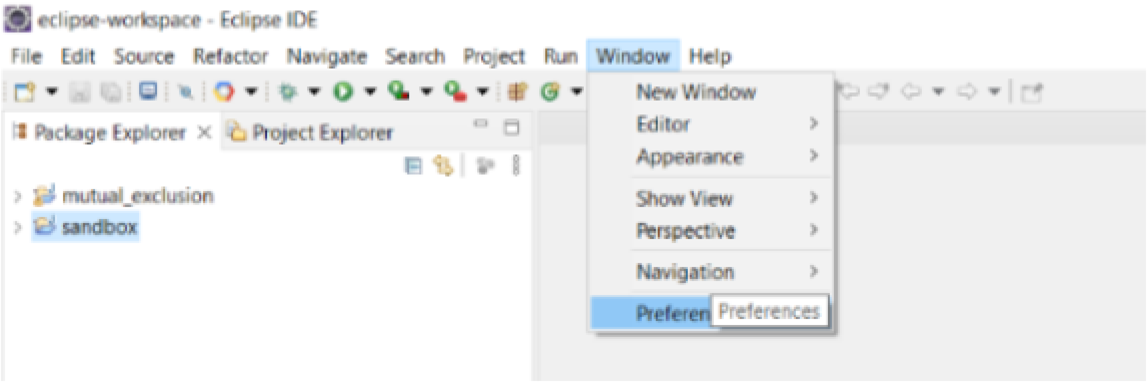
Step 7: - Below window will appear close the welcome page


eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Welcome ×
Overview

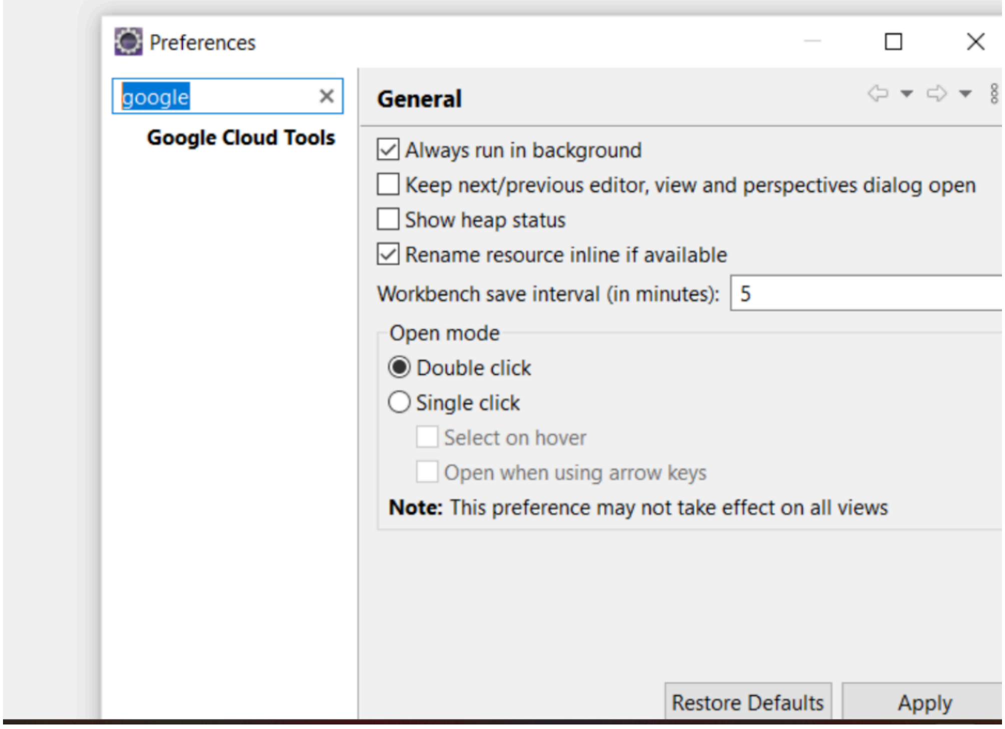
Google Cloud Tools for Eclipse

Learn how to create, build, run, debug, and deploy App Engine and Cloud Dataflow applications from Eclipse

Step 8: - Once the eclipse window appear, click on Window -> Preferences



Step 9: - Below Preferences window will appear, search for google



☑ Always run in background

□ Keep next/previous editor, view and perspectives dialog open

□ Show heap status

☑ Rename resource inline if available

Workbench save interval (in minutes): 5

Open mode

● Double click

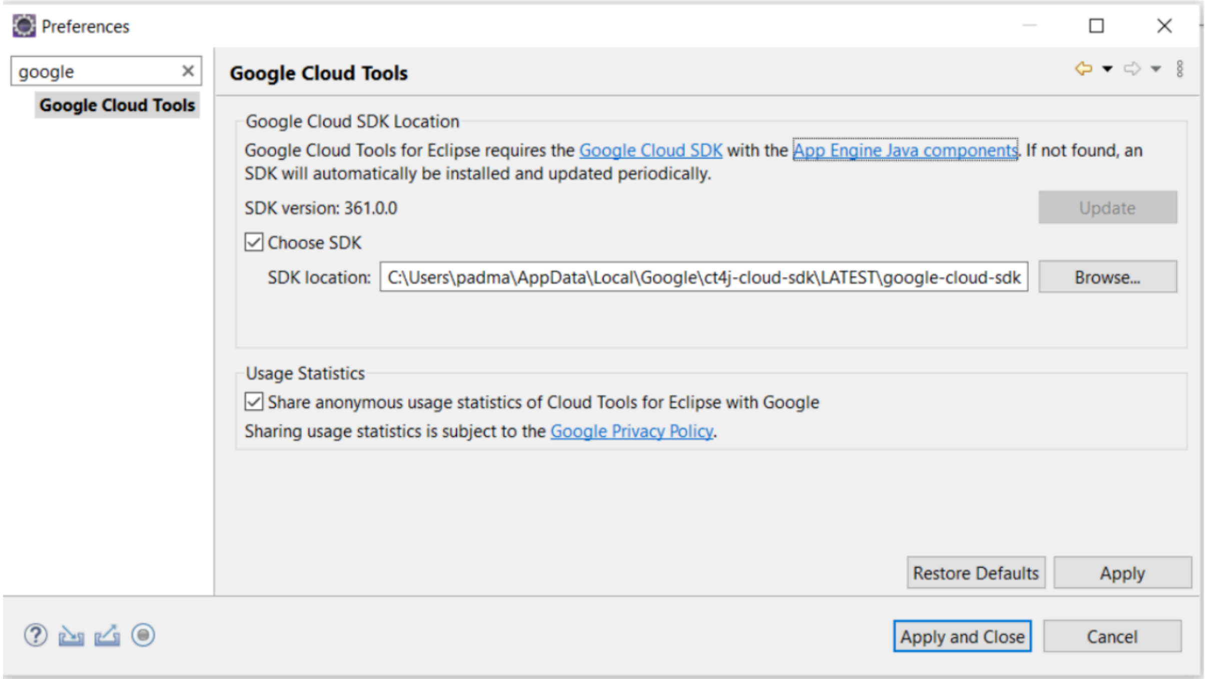
○ Single click

□ Select on hover

□ Open when using arrow keys

Note: This preference may not take effect on all views

Step 10: - Click for App Engine Java Components and click on Apply and Close



Google Cloud SDK Location

Google Cloud Tools for Eclipse requires the Google Cloud SDK with the App Engine Java components. If not found, an SDK will automatically be installed and updated periodically.

SDK version: 361.0.0

☑ Choose SDK

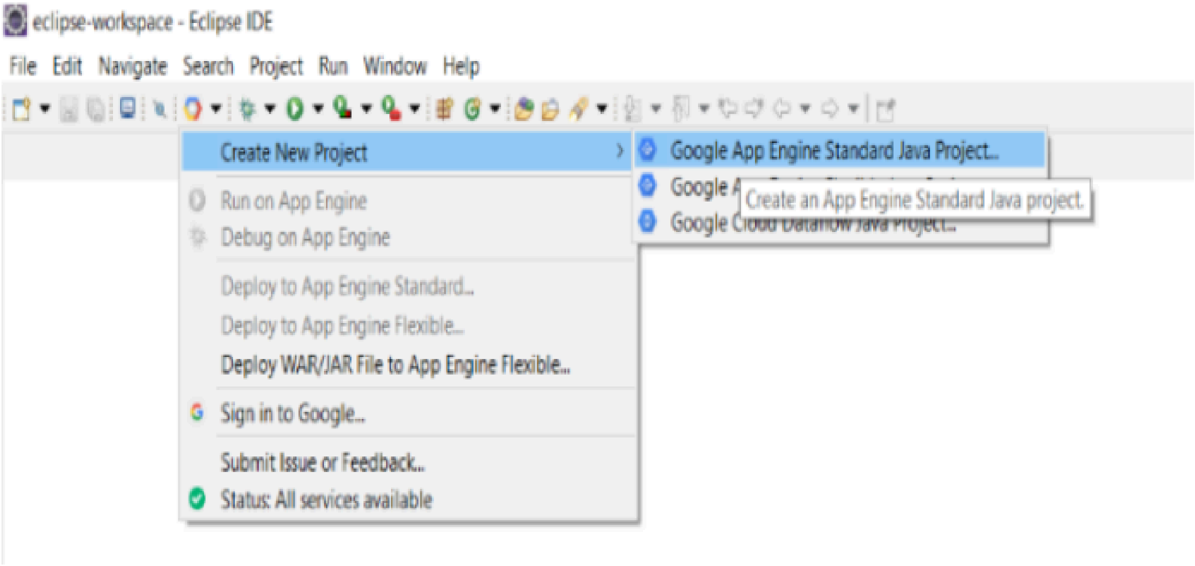
SDK location: C:\Users\padma\AppData\Local\Google\ct4j-cloud-sdk\LATEST\google-cloud-sdk

Usage Statistics

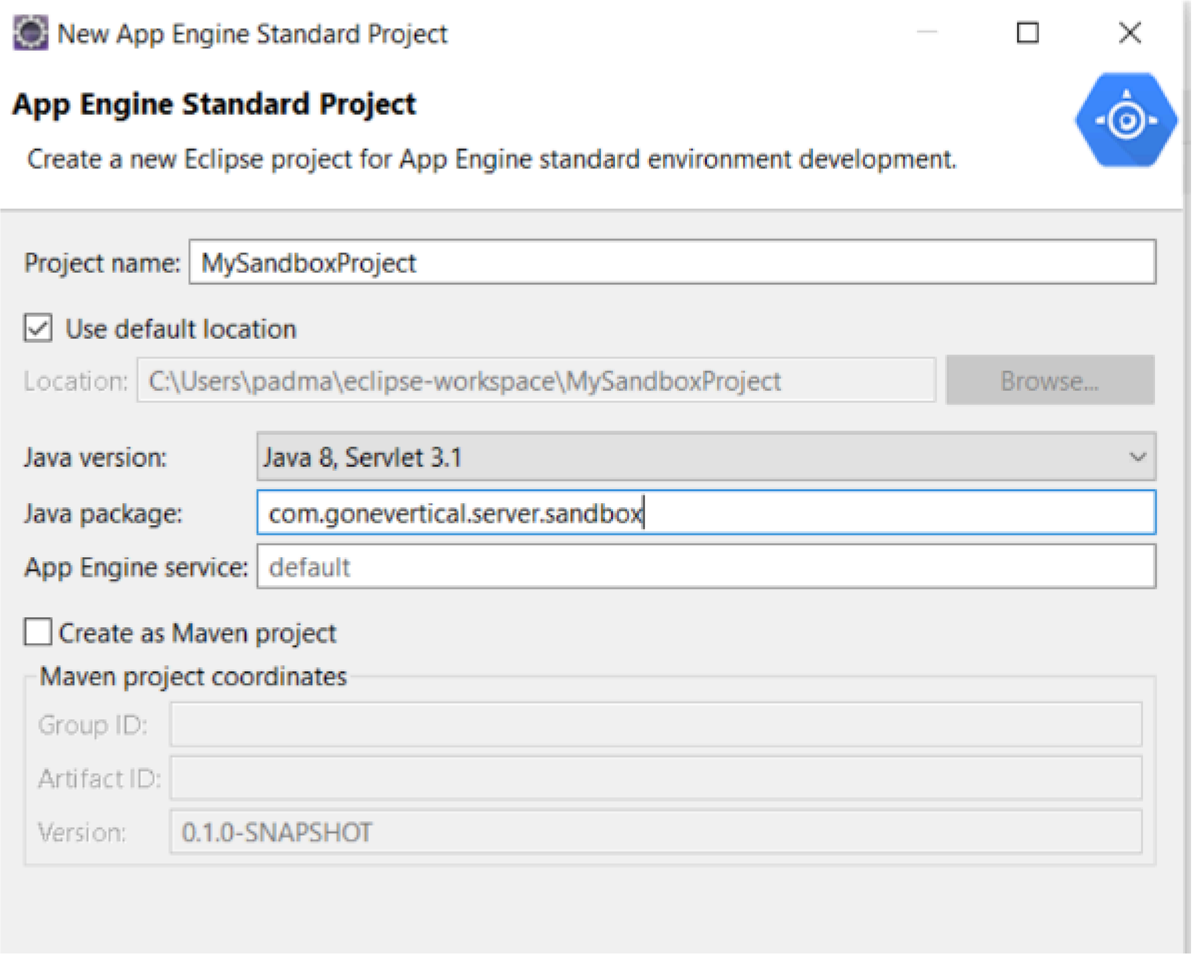
☑ Share anonymous usage statistics of Cloud Tools for Eclipse with Google

Sharing usage statistics is subject to the Google Privacy Policy.

Step 11: - Now create a “Google App Engine Standard Java Project”


eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Create New Project
Run on App Engine
Debug on App Engine
Deploy to App Engine Standard...
Deploy to App Engine Flexible...
Deploy WAR/JAR File to App Engine Flexible...
Sign in to Google...
Submit Issue or Feedback...
Status: All services available
Google App Engine Standard Java Project...
Google A
Google Cloud Dataflow Java Project...
Create an App Engine Standard Java project.

Step 12: - Below screen will appear New App Engine Standard Project. Provide the Project Name & Java Package as mentioned below:
Project Name: My SandboxProject
Java Package: com.gonevertical.server.sandbox
Click on Next



New App Engine Standard Project

App Engine Standard Project

Create a new Eclipse project for App Engine standard environment development.

Project name: MySandboxProject

☑ Use default location

Location: C:\Users\padma\eclipse-workspace\MySandboxProject

Java version: Java 8, Servlet 3.1

Java package: com.gonevertical.server.sandbox

App Engine service: default

☐ Create as Maven project

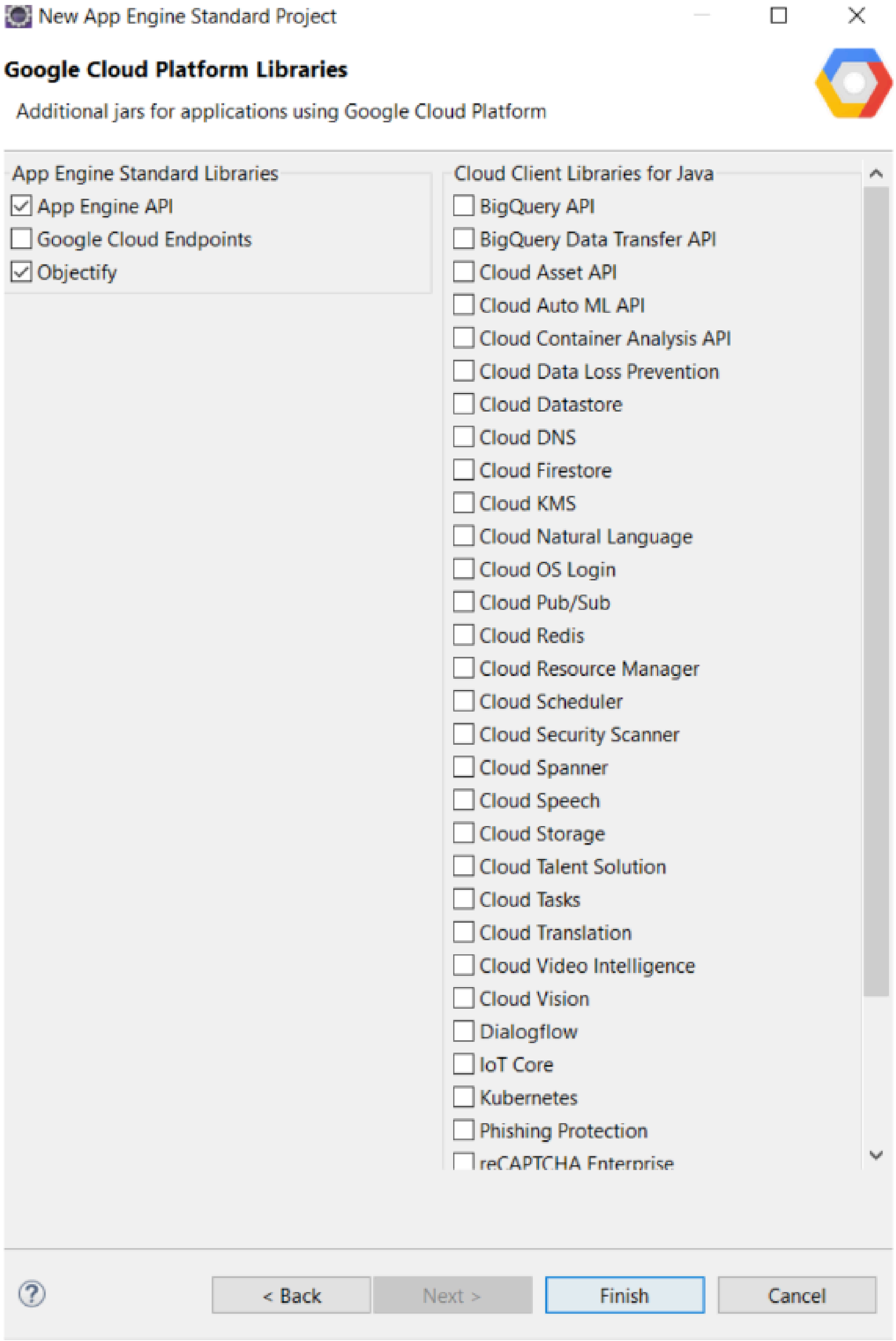
Maven project coordinates

Group ID:

Artifact ID:

Version: 0.1.0-SNAPSHOT

Step 13: - Below screen will appear. From App Engine Standard Libraries
Select 1.App Engine API 2.Objectify
Click on Finish



Google Cloud Platform Libraries

Additional jars for applications using Google Cloud Platform

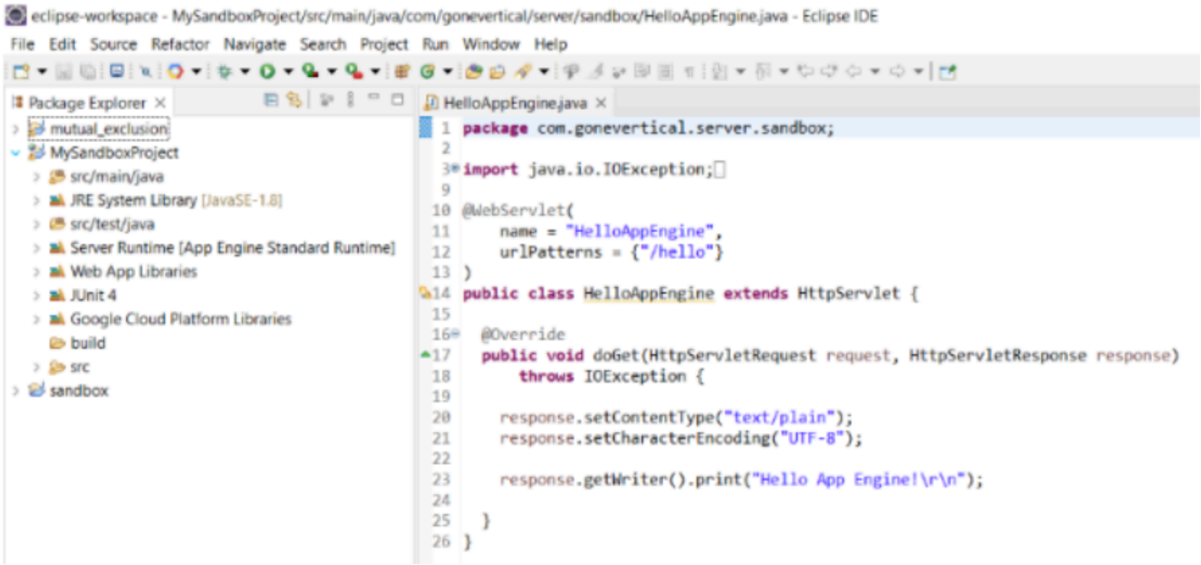
App Engine Standard Libraries

- ☑ App Engine API
- □ Google Cloud Endpoints
- ☑ Objectify

Cloud Client Libraries for Java

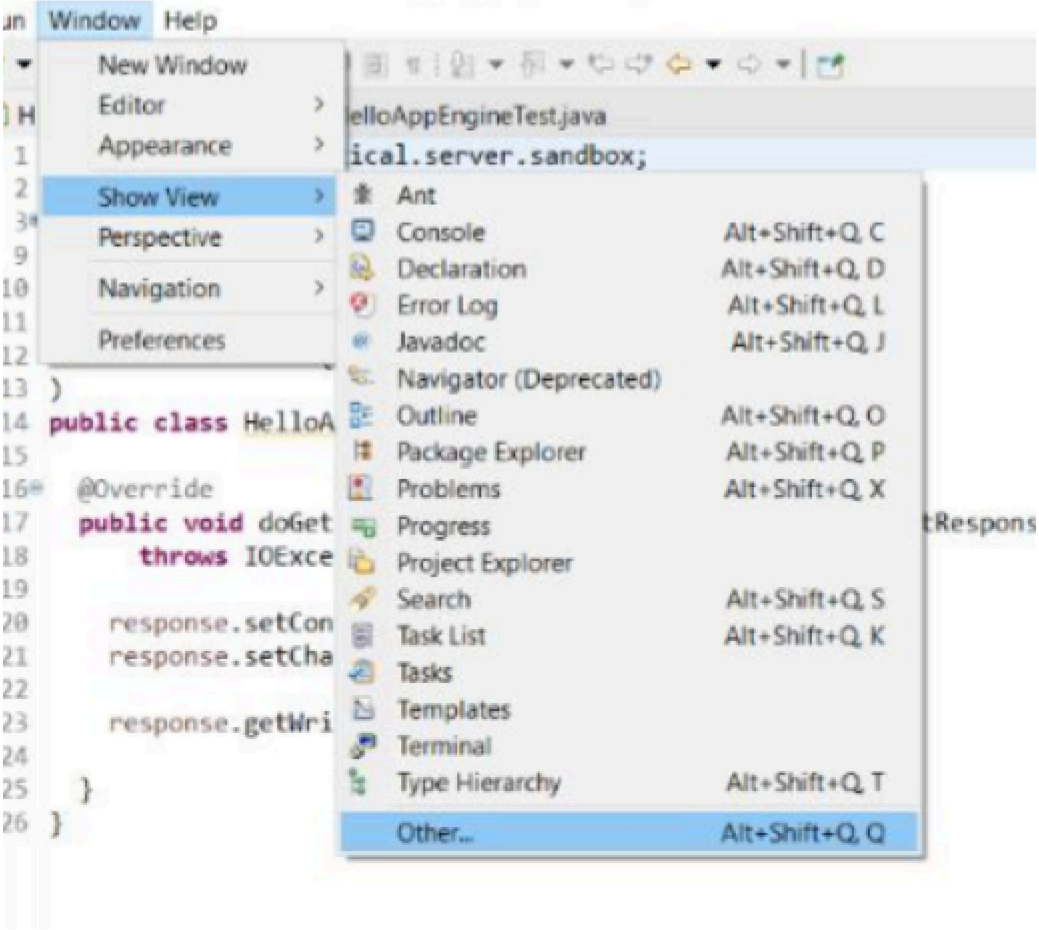
- □ BigQuery API
- □ BigQuery Data Transfer API
- □ Cloud Asset API
- □ Cloud Auto ML API
- □ Cloud Container Analysis API
- □ Cloud Data Loss Prevention
- □ Cloud Datastore
- □ Cloud DNS
- □ Cloud Firestore
- □ Cloud KMS
- □ Cloud Natural Language
- □ Cloud OS Login
- □ Cloud Pub/Sub
- □ Cloud Redis
- □ Cloud Resource Manager
- □ Cloud Scheduler
- □ Cloud Security Scanner
- □ Cloud Spanner
- □ Cloud Speech
- □ Cloud Storage
- □ Cloud Talent Solution
- □ Cloud Tasks
- □ Cloud Translation
- □ Cloud Video Intelligence
- □ Cloud Vision
- □ Dialogflow
- □ IoT Core
- □ Kubernetes
- □ Phishing Protection
- □ reCAPTCHA Enterprise

Step 14: - Below Screen will Appear "HelloAppEngine.java"


eclipse-workspace - MySandboxProject/src/main/java/com/gonevertical/server/sandbox/HelloAppEngine.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
mutual_exclusion
MySandboxProject
src/main/java
JRE System Library [JavaSE-1.8]
src/test/java
Server Runtime [App Engine Standard Runtime]
Web App Libraries
JUnit 4
Google Cloud Platform Libraries
build
src
sandbox
HelloAppEngine.java

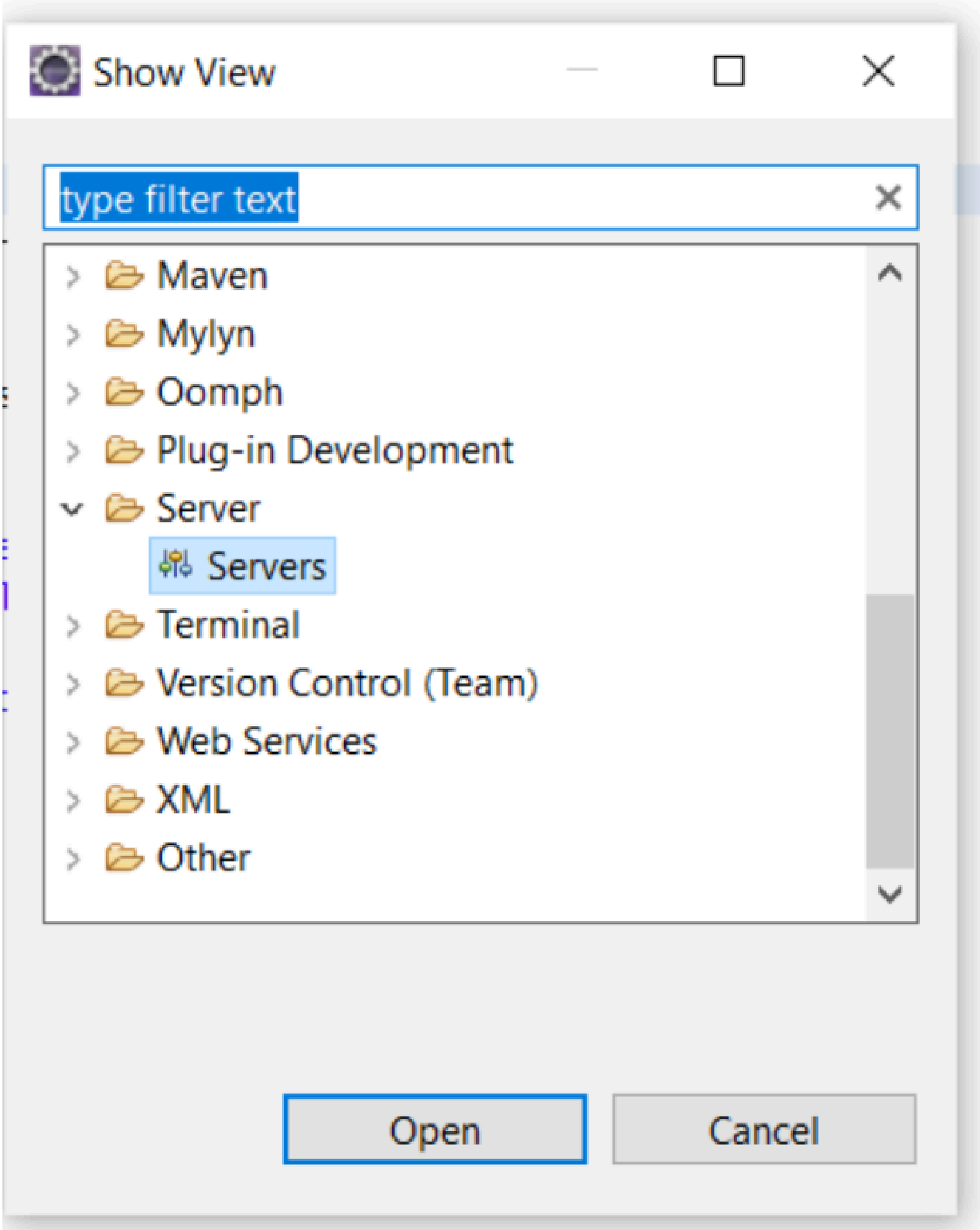
```java
package com.gonevertical.server.sandbox;

import java.io.IOException;

@WebServlet(
    name = "HelloAppEngine",
    urlPatterns = {"/hello"}
)
public class HelloAppEngine extends HttpServlet {

  @Override
  public void doGet(HttpServletRequest request, HttpServletResponse response)
      throws IOException {

    response.setContentType("text/plain");
    response.setCharacterEncoding("UTF-8");

    response.getWriter().print("Hello App Engine!\r\n");

  }
}
```

Step 15: - Go to Window->Show View->Other

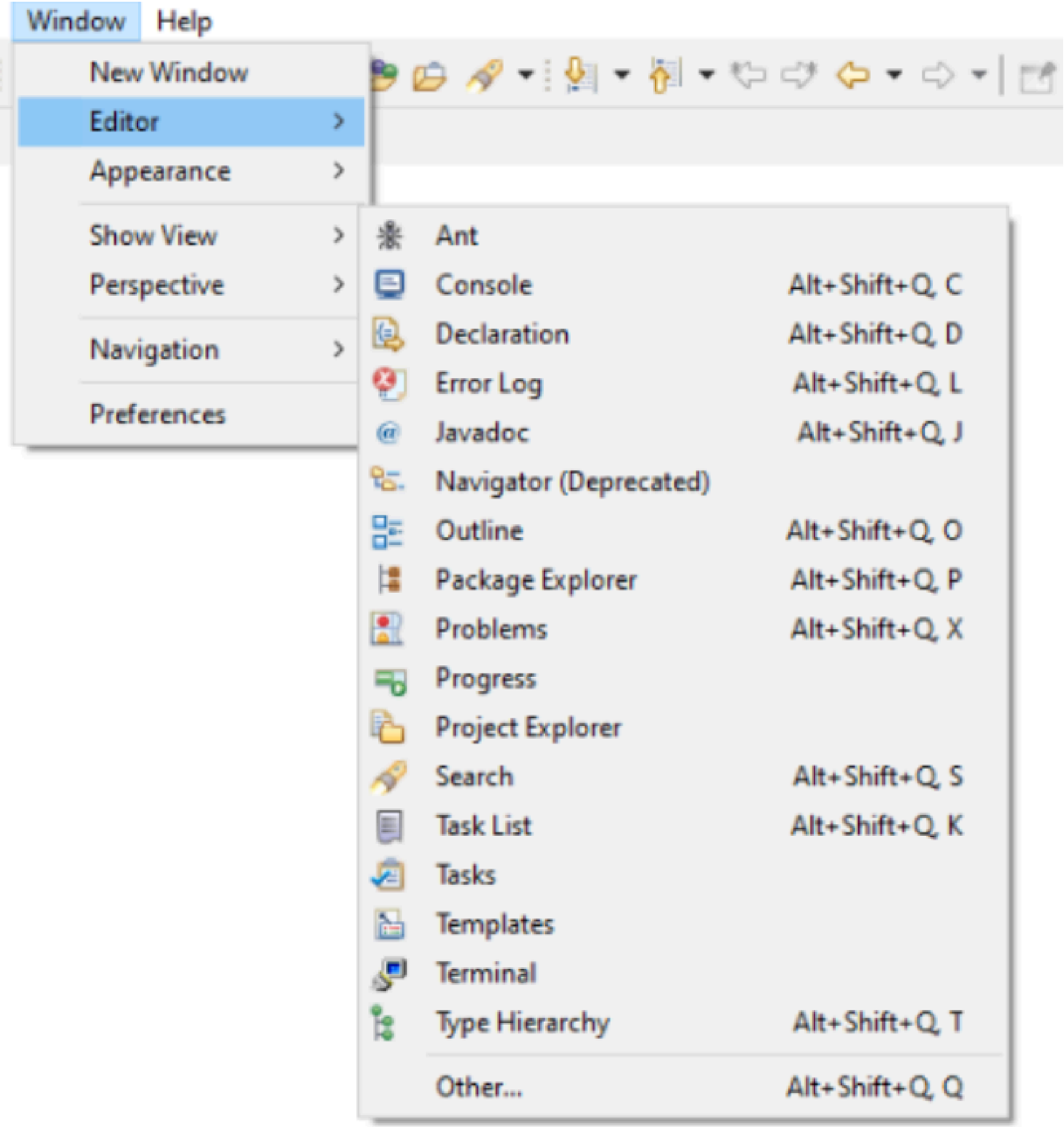


Step 16: - After clicking on other below Screen of Show View will appear

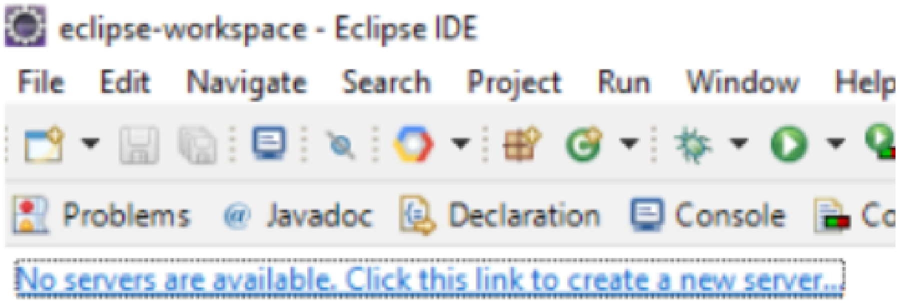
1. Click on Server->Servers
2. Click on Open



Step 17: - Click on Window->Editor->Console

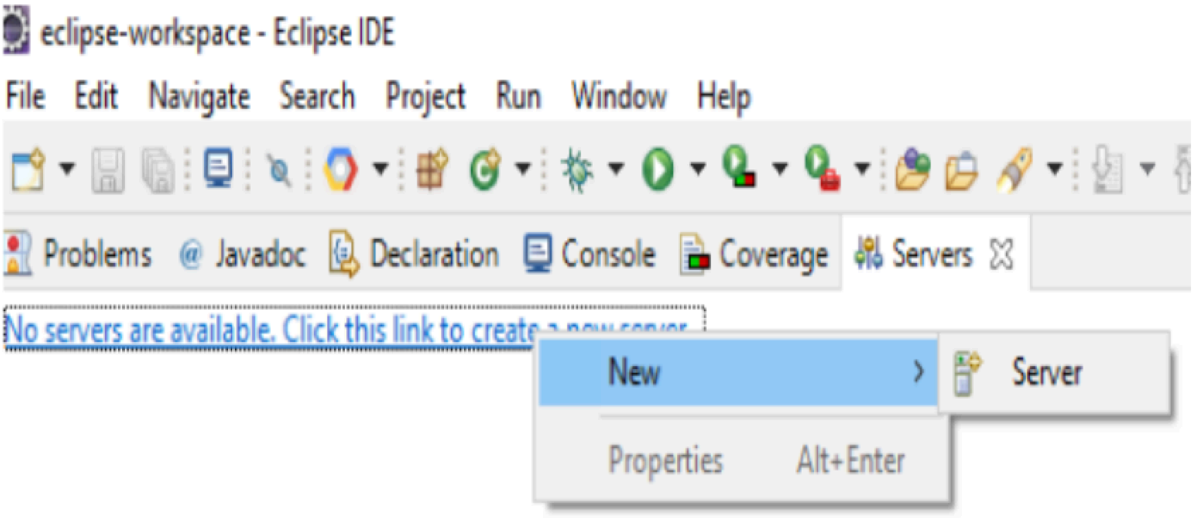


Step 18: - Once the Console screen appear ->Click on Servers

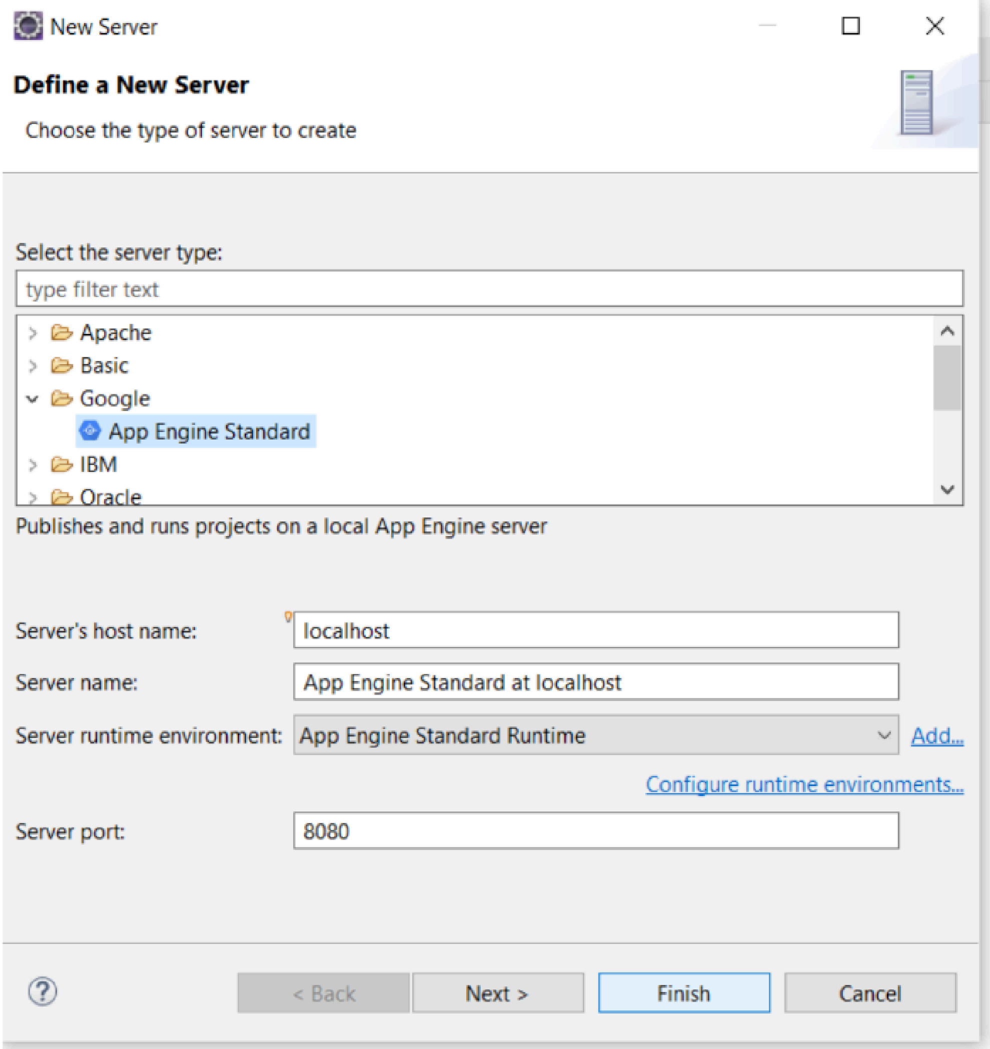


No servers are available. Click this link to create a new server...

Step 19: - Right click on server ->New->Server


eclipse-workspace - Eclipse IDE
File Edit Navigate Search Project Run Window Help
Problems
Javadoc
Declaration
Console
Coverage
Servers
No servers are available. Click this link to create a new server...
New
Properties Alt+Enter
Server

Step 20: - To define a new server will below screen appear 1. Select App Engine Standard 2.Click on Next 3.Click on Finish



Define a New Server

Choose the type of server to create

Select the server type:

Publishes and runs projects on a local App Engine server

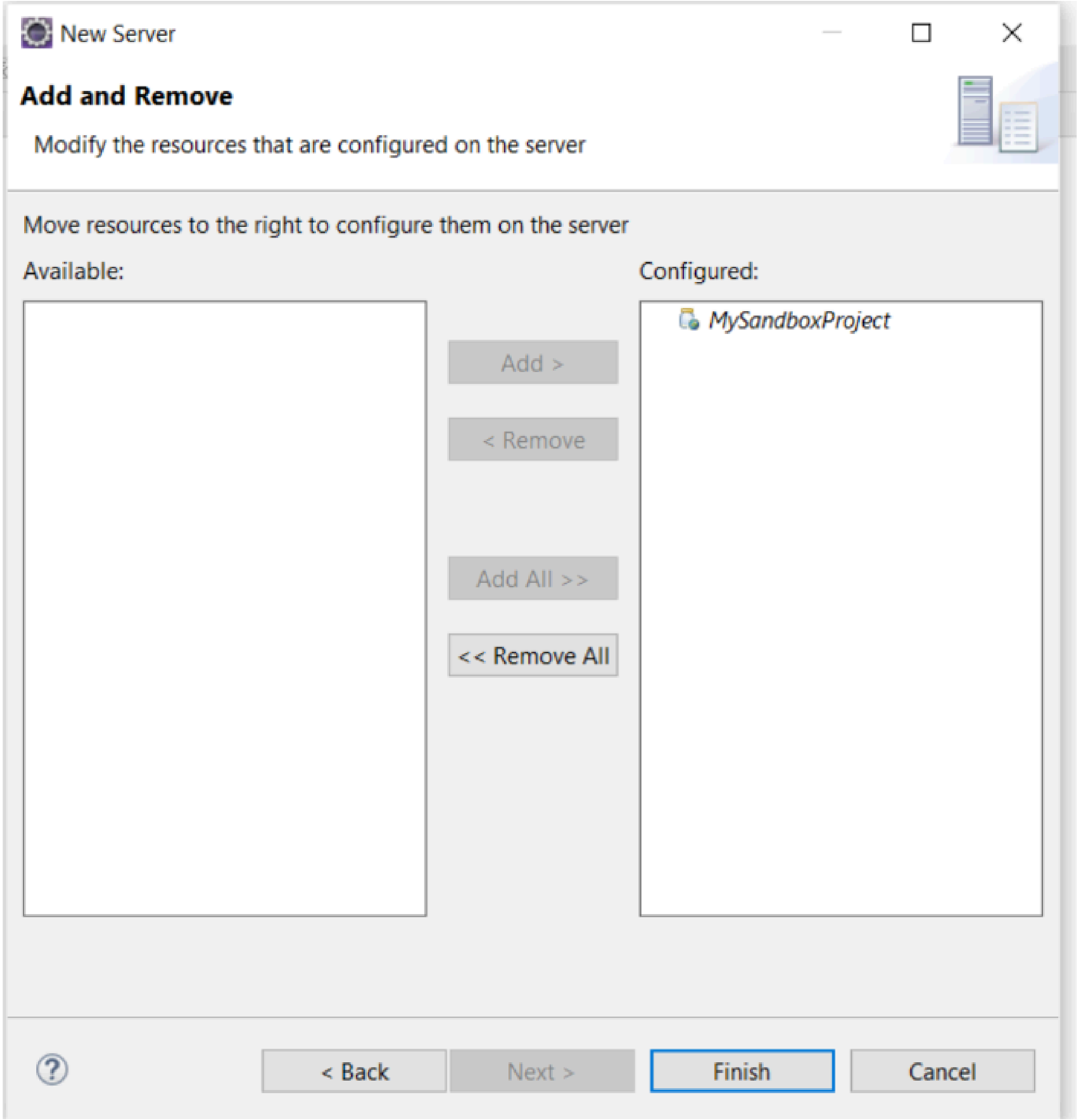
Server's host name: localhost

Server name: App Engine Standard at localhost

Server runtime environment: App Engine Standard Runtime

Server port: 8080

Step 21: - Select from Available Project: MySandboxProject and then click on Add Button. Click on Finish Button



New Server

Add and Remove

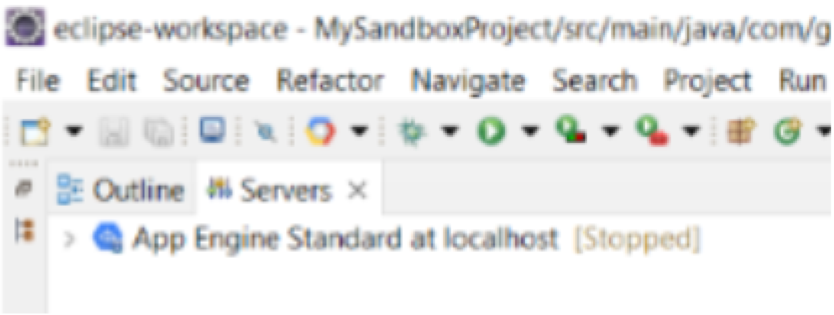
Modify the resources that are configured on the server

Move resources to the right to configure them on the server

Available:

Configured:

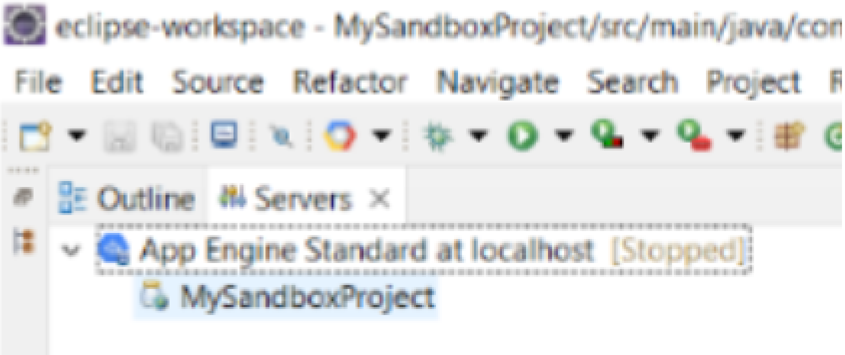
MySandboxProject



eclipse-workspace - MySandboxProject/src/main/java/com/g

App Engine Standard at localhost [Stopped]

Step 22: - Click On App Engine Standard At localhost ->MySandboxProject

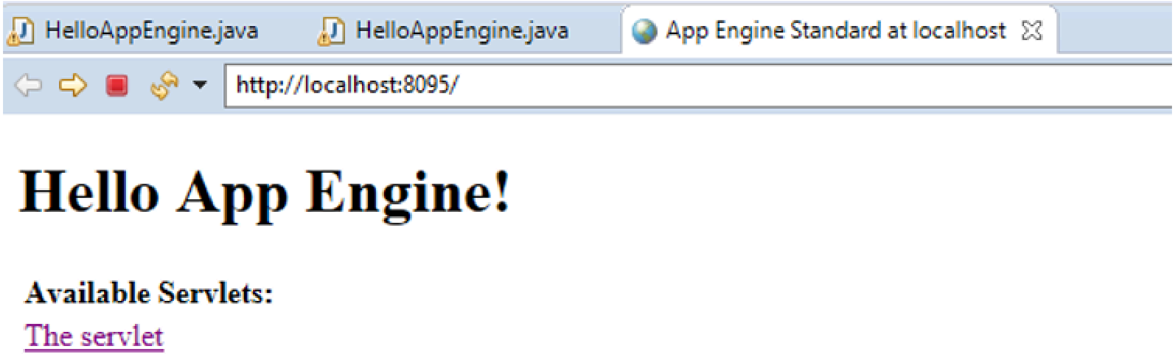

eclipse-workspace - MySandboxProject/src/main/java/con
File Edit Source Refactor Navigate Search Project R
Outline
Servers
App Engine Standard at localhost [Stopped]
MySandboxProject

Step 23: - Execute the following code, Right click on App Engine Standard At localhost->Click on Debug

Code: -

```
import java.io.IOException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet(
  name = "HelloAppEngine!",
  urlPatterns = {"/hello"}
)
public class HelloAppEngine extends HttpServlet {
 @Override
 public void doGet(HttpServletRequest request, HttpServletResponse response)
   throws IOException {
  response.setContentType("text/plain");
  response.setCharacterEncoding("UTF-8");
  response.getWriter().print("Hello from JITESH To App Engine!\r\n");
 }
}
```

Output: -

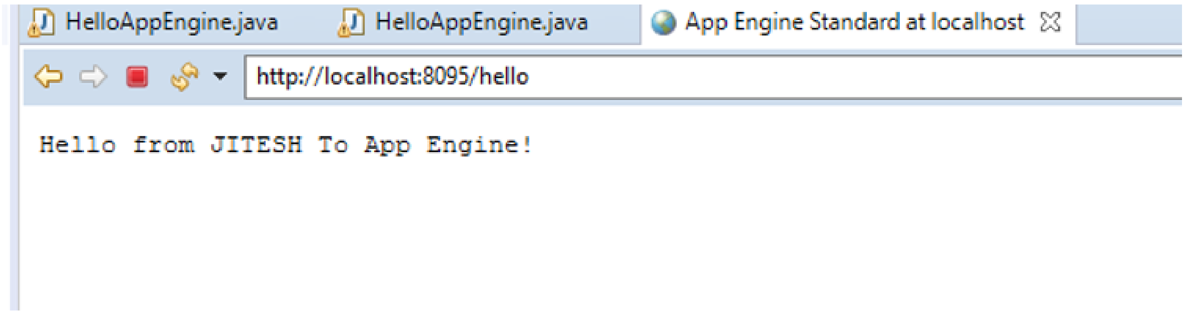


http://localhost:8095/

Hello App Engine!

Available Servlets:

The servlet



http://localhost:8095/hello

```
Hello from JITESH To App Engine!
```