**PRACTICAL NO:3**

**Create an Asp.Net** **Web Core Api and Configure Monitoring.**

Setting up the Development Environment

Let’s set up.

1. Install Visual Studio 2017.

2. Install the Microsoft Azure Service Fabric SDK.

3. Create the Translator Text API in your Azure subscription and make a note of the access key.

4. Create an empty Azure SQL Database and keep the connection string with SQL Authentication handy.

5. Make sure that the Service Fabric local cluster on Windows is in a running state.

6. Make sure that the Service Fabric Azure cluster on Windows is in a running state.

Create an ASP.NET Core Web API

Now let’s start the API.

1. Launch Visual Studio 2017 as an administrator.

2. Create a project by selecting File 🡪 New 🡪 Project.

3. In the New Project dialog, choose Cloud 🡪Service Fabric Application.

4. Name the Service Fabric application **EmployeeApp** and click OK.

A screenshot of a computer

Description automatically generated

5. Name the stateless ASP.NET Core service **Employee.Stateless.Api** (as seen in Figure) and click OK.

A screenshot of a computer

Description automatically generated

6. Choose the API and click OK. Make sure that ASP.NET Core 2.2 is selected, as shown in Figure.

A screenshot of a computer

Description automatically generated

7. Right-click the employee.stateless.api project and select Add 🡪 Connected Service.

A screenshot of a computer

Description automatically generated

8. Choose Monitoring with Application Insights, as seen in Figure.

A computer screen with text boxes

Description automatically generated

9. Click Get Started, as seen in Figure.

A screenshot of a computer

Description automatically generated

Choose the right Azure subscription and Application Insights resource. Once done, click Register, as seen in Figure.

A computer screen with a white box

Description automatically generated

It takes a few minutes to create the Application Insights resource in

your Azure subscription. During the registration process, you see the

screen shown in Figure.

A screenshot of a computer

Description automatically generated

11. Once the Application Insights configuration is complete, you see the status as 100%. If you see the Add SDK button (as shown in Figure), click it to achieve 100% status, as seen in Figure.



A computer screen with a white box

Description automatically generated

12. To confirm the Application Insights configuration, check the instrumentation key in appsettings.json.

13. Right-click the employee.stateless.api project to add dependencies for the following NuGet packages.

a. Microsoft.EntityFrameworkCore.SqlServer

b. Microsoft.ApplicationInsights.ServiceFabric.Native

c. Microsoft.ApplicationInsights.AspNetCore

We are done with the configuration. Now let’s add

EmployeeController.

1. Right-click the employee.stateless.api project and a folder called Models. Add the following classes from the sources folder.

a. AppSettings.cs

b. Employee.cs

c. SampleContext.cs

d. TranslationResponse.cs

2. Right-click the employee.stateless.api project and add a file named DbInitializer.cs. Replace that content with the following content.

A computer screen shot of a computer

Description automatically generated

3. Open Api.cs and replace the contents of the CreateServiceInstanceListeners method with the following content.

A computer screen with text on it

Description automatically generated

A computer screen with a white screen

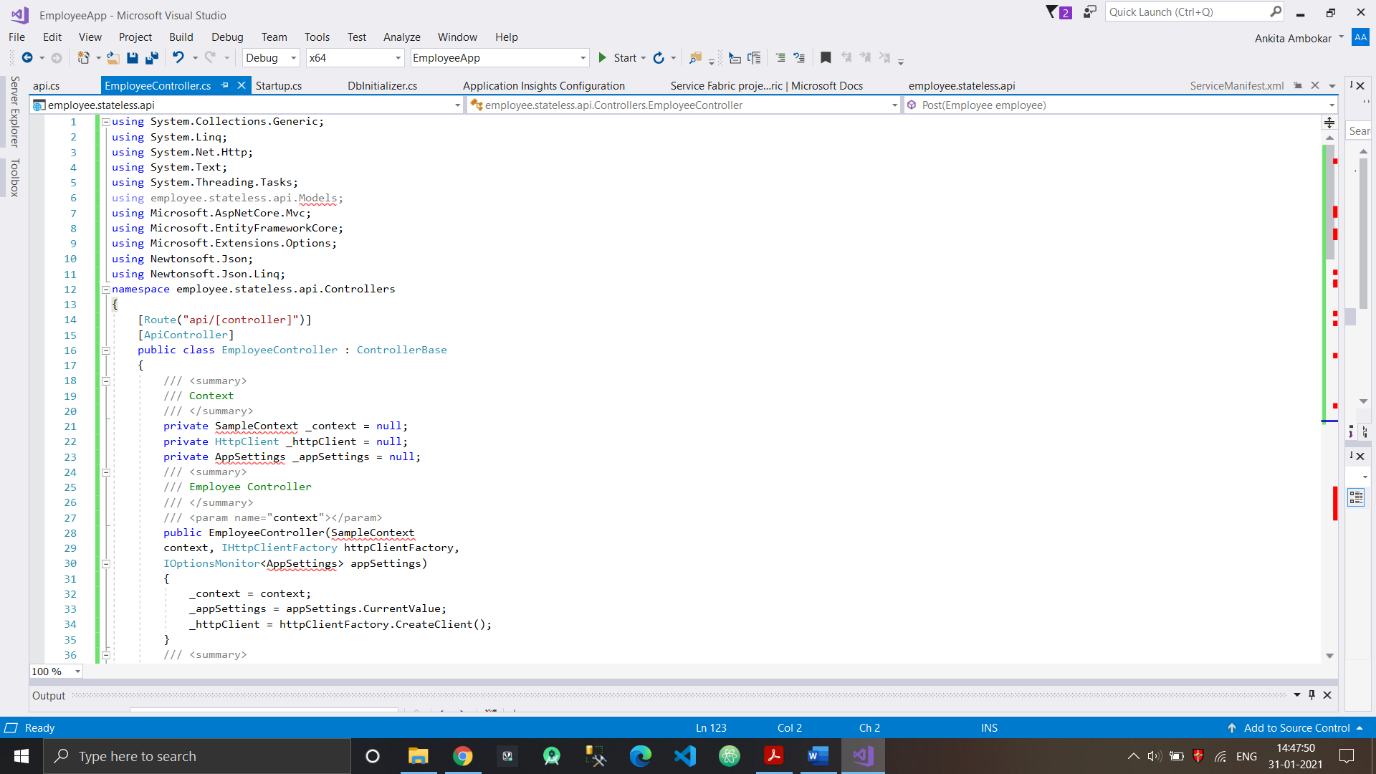
Description automatically generated

4. Open Startup.cs and replace the contents of the ConfigureServices method with the following content.

A computer screen with a white screen

Description automatically generated

5. Right-click the controller folder in the employee.stateless.api project and add a controller called EmployeeController.cs. Replace that content with the following content.



A computer screen with text on it

Description automatically generated

A computer screen with text on it

Description automatically generated

6. Open AppSettings.json and make sure that the

content looks similar to your connection strings.

{

"ConnectionStrings": {

"DefaultConnection": "Server=<<YOUR\_SERVER>>;Database=

<<YOUR\_DATABASE>>;User ID=<<USER\_ID>>;Password=<<PASSWORD>>

;Trusted\_Connection=False;Encrypt=True;MultipleActiveResult

Sets=True;"

},

"AppSettings": {

"TranslationApiUrl": "https://api.cognitive.

microsofttranslator.com",

"AccessKey": "<<YOUR ACCESS KEY TO TRANSLATION API>>"

},

"Logging": {

"LogLevel": {

"Default": "Warning"

}

},

"AllowedHosts": "\*",

"ApplicationInsights": {

"InstrumentationKey": "<<YOUR INSTRUMENTATION KEY OF YOUR

APP INSIGHTS RESOURCE>>"

}