**ASP.NET Core Workshop**

**ADO.NET**

# Requirements

In this workshop, we will create a simple web application using ASP.NET ADO.NET with the following features.

1. When the users access /Course, the app displays a summary of all courses.

Graphical user interface, text, application, email

Description automatically generated

1. When the users click on the **Details** action above, the details of the respective course will be displayed.

Graphical user interface, text, application

Description automatically generated

1. The app can also display a summary of all lecturers.

Graphical user interface

Description automatically generated

1. It also displays all courses the respective lecturer teaches.

Graphical user interface, text, application, email

Description automatically generated

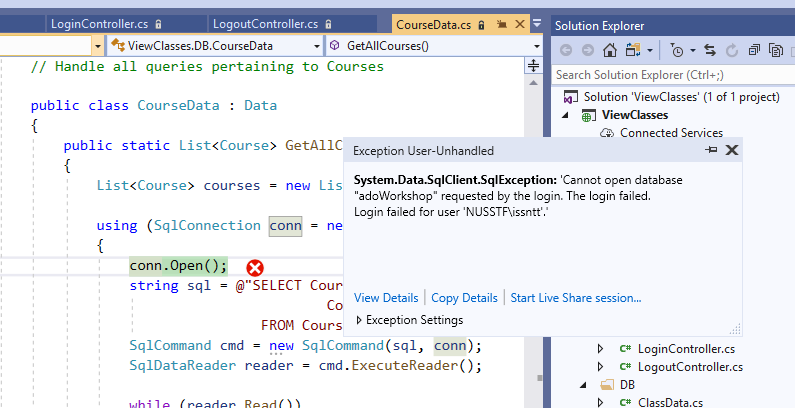
# Prepare Code

## Option 1

Follow the steps in the given videos.

## Option 2

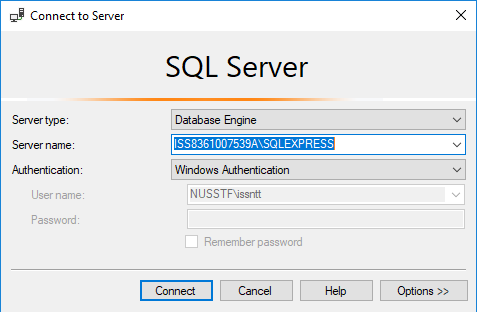
1. Download, unzip and open solution **ADO.NET Example** from **ADO.NET Workshop Start.zip**.
2. Run the **ADO.NET Example** project
3. In the browser, launch the /Course, for example: <http://localhost:7209/Course> (change port number accordingly)
4. An exeption like what is shown below is thrown in class **CourseData** (could you explain why?)



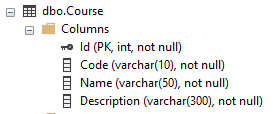
# Prepare Database

1. Open Microsoft SQL Server Management Studio.

Remember location of the **Server name.** We will need to copy the string of server name later.



1. Click Connect.
2. Create a new database named **adoWorkshop**.
3. Select the new database, create a table named **Course**, with following columns



1. Input some data into the **Course** table. For example

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Code** | **Name** | **Description** |
| 1 | FOPCS | Foundations of Programming with C# | Write programs using C# with Visual Studio |
| 2 | OOPCS | Object Oriented with C# | Understand and program on main Object Oriented concepts |
| 3 | MVC.NET | ASP.NET MVC Programming | Understand and develop web applications using ASP.NET MVC |

1. In project ADO.NETWorkshop, open class **/Data/CourseData.cs**
2. Change the **connectString** value to to what is shown in the **Server name** on your SQL Server Management Studio.

**Note**: different versions of SQL Server use different strings. For example, Express version may need to include something like “<computer\_name>\SQLEXPRESS”, while Developer vesrion only needs “<computer\_name>”.

1. Re-run the ADO.NETWorkshop project, and test with the project URL <http://localhost:7209/Course> (change your port number accordingly) again.

The webpage should display the courses in the database.

Graphical user interface, text, application, email

Description automatically generated

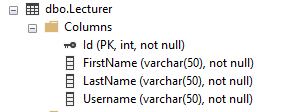
1. Investigate the class **/DB/CourseData.cs** to see how ADO.NET is used to implement the features.

# Implement Features

1. Implement a feature so that clicking on the **Details** action above will display the details of the respective course.

Graphical user interface, text, application

Description automatically generated

1. Extend the web application above to store and list all Lectures.   
   Table **Lecturer** looks like this  
   

Create some mock data into your tables to help test your web application. For example:

Graphical user interface

Description automatically generated

1. Further extend the web application to store and list all courses a lecturer teaches. One lecturer can teach more than one courses, and a course can be taught by more than one lecturers.

Create some mock data into your tables to help test your web application. For example:

Graphical user interface, text, application, email

Description automatically generated

1. (Optional) Further extend the web application to add creating new courses feature.
2. (Optional) Further extend the web application to add searching courses by **Name** and by **Course Code**.