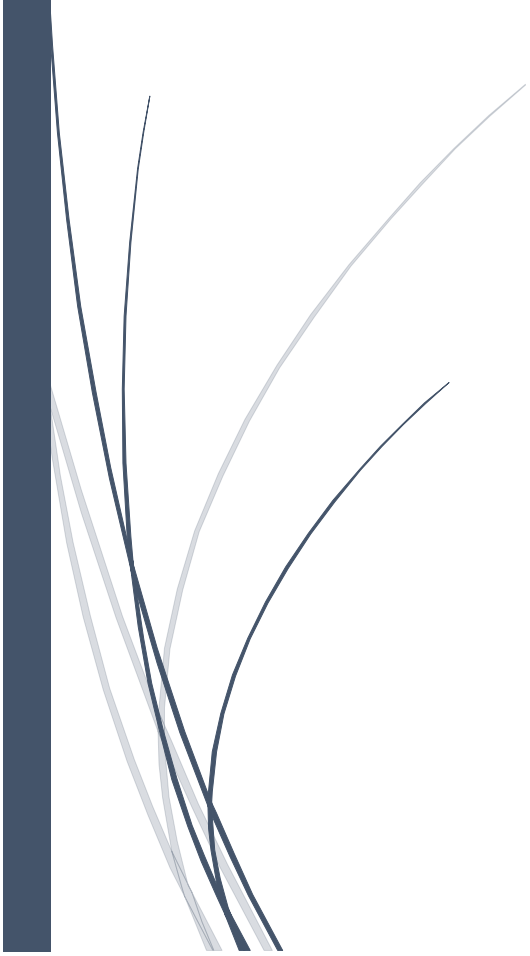
12/9/2020

APP Installation Instructions

AZLearn Capstone Project



Cohort 4.2

TECHCAREERS 2020

Contents

Prerequisites

- This application requires Git Bash terminal installed or accessible from one of the common IDE like VSCode or Visual Studio 2019
- This application requires Visual Studio 2019 and ASP.NET Core 3.1
- This application uses ReactJS.NET on ASP.NET Core and requires a few packages installation through NuGet
- Entity Framework (EF) Core is used in this application to perform data access against the MySQL database (MariaDB) and server (Apache)
- This application requires the use of Code First Migrations using EF in .NET Core to setup databases with dummy data
- This application requires a fundamental knowledge of SQL Syntax
- Developers tools such as XAMPP and Postman are recommended to engage, test, and use this application.

Assumption

This guide was written for Windows OS users. We would like to apologies for MacOs users since we could not cover the steps for MacOs.

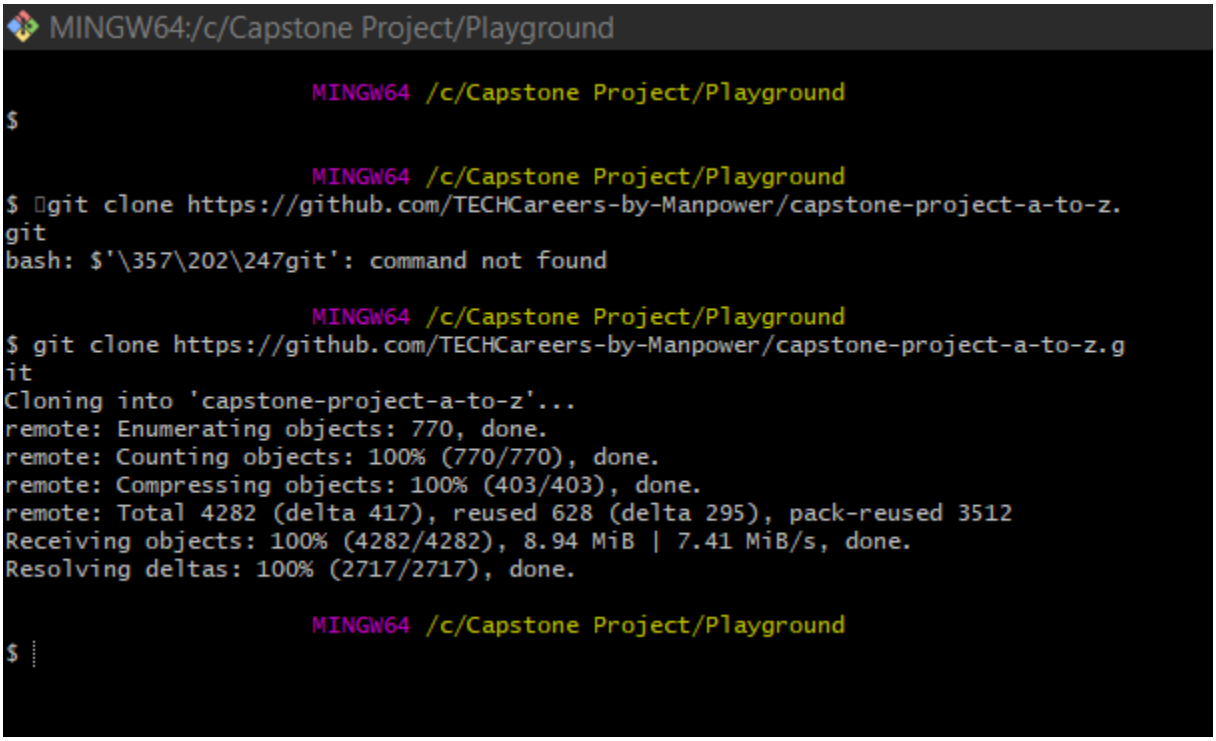
Installation

Project Cloning

To get started, navigate to a folder where you would like to download the project. Be mindful of the path characters limit, so the shorter the path is the better.

Right click anywhere in the folder and open Git Bash terminal and within Git run the following commands:

git clone https://github.com/TECHCareers-by-Manpower/capstone-project-a-to-z.git



Navigate to the project folder i.e., capstone-project-a-to-z and double click AZLearn.sln to open the solution/project using Visual Studio. Note we have kept the solution and project in same directory.

Dot Net Packages Installation

Following commands will be used to restore the dependencies/ packages through Package Manager Console (IMPORTANT, ensure your current working directory is same as AZLearn.csproj). You can confirm using `PM> ls` command

```
PM> ls


    Directory: C:\Capstone Project\Playground\capstone-project-a-to-z


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2020-12-09   5:04 AM                .vscode
d-----        2020-12-09   5:06 AM                bin
d-----        2020-12-09   5:04 AM                ClientApp
d-----        2020-12-09   5:04 AM                Controllers
d-----        2020-12-09   5:04 AM                Data
d-----        2020-12-09   5:04 AM                Migrations
d-----        2020-12-09   5:04 AM                Models
d-----        2020-12-09   5:06 AM                obj
d-----        2020-12-09   5:04 AM                Pages
d-----        2020-12-09   5:04 AM                Project_Documentation
d-----        2020-12-09   5:04 AM                Properties
-a----        2020-12-09   5:04 AM           2581 .gitattributes
-a----        2020-12-09   5:04 AM          11231 .gitignore
-a----        2020-12-09   5:04 AM            168 appsettings.Development.json
-a----        2020-12-09   5:04 AM            192 appsettings.json
-a----        2020-12-09   5:04 AM           4045 AZLearn.csproj
-a----        2020-12-09   5:04 AM           1118 AZLearn.sln
-a----        2020-12-09   5:04 AM          10752 package-lock.json
-a----        2020-12-09   5:04 AM            503 Program.cs
-a----        2020-12-09   5:04 AM           2349 README.md
-a----        2020-12-09   5:04 AM           2700 Startup.cs
-a----        2020-12-09   5:04 AM            303 WeatherForecast.cs


PM>
```

Installations of Entity Framework Core can be done either through NuGet Package Manager for Solution... or through Visual Studio 2019 NuGet Package Manager Console (Package Manager). For that make sure you have ef globally installed already or run the following command while inside the project folder.

`dotnet tool install --global dotnet-ef`

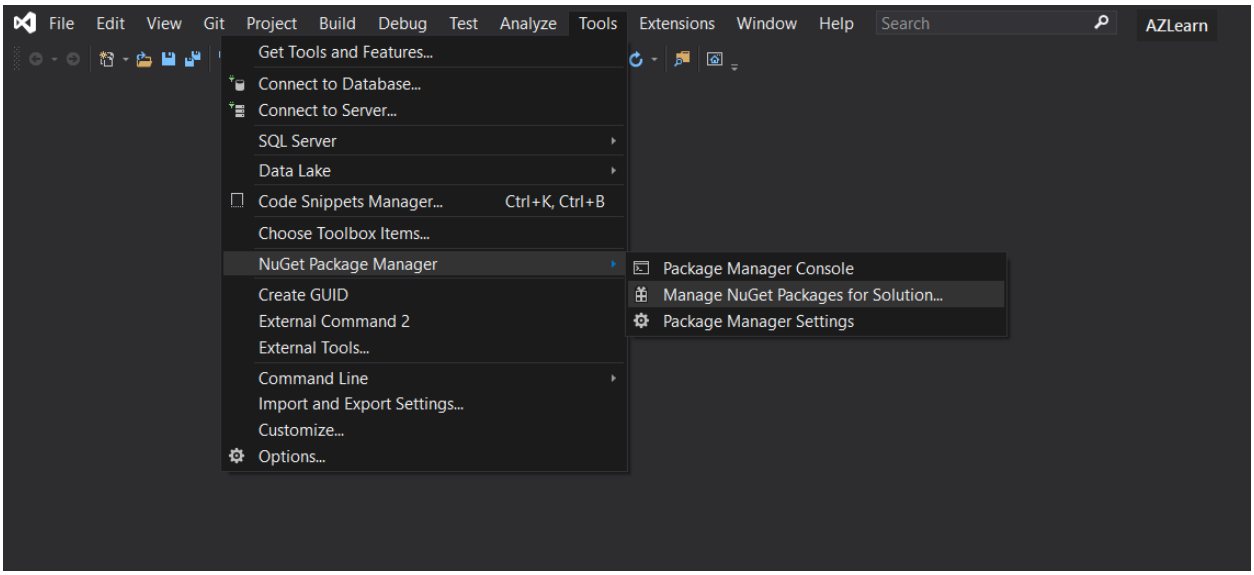
Then to restore the project specific packages, please run the following command through Package Manager Console

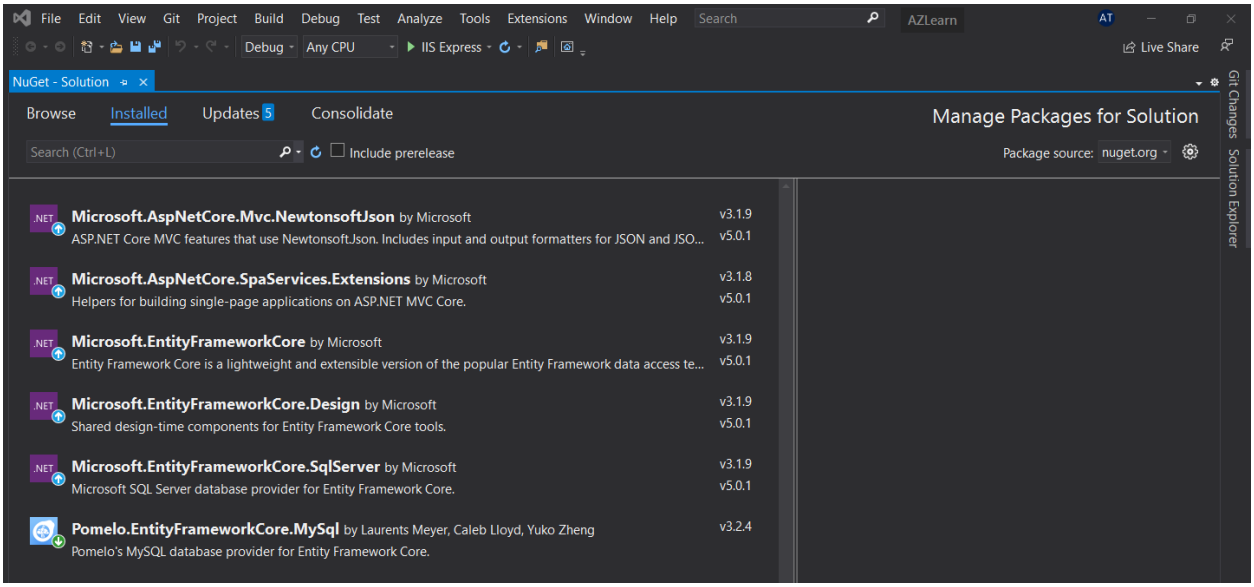
`PM> dotnet restore`

You might not need to restore packages, and for that the return from above command will be like below.

```
PM> dotnet restore
  Determining projects to restore...
  All projects are up-to-date for restore.
```

Please if you check your NuGet installed packages they should look like below. Note the versions in below screen shot as you must stick to those for the app to run successfully.





If you have encountered any issues restoring the packages from previous steps, you may proceed with the following packages manual instructions as follows.

In Package Manager Console run the following commands, one at a time.

```
PM> dotnet add package Microsoft.EntityFrameworkCore.Design –version 3.1.9
```

```
PM> dotnet add package Microsoft.EntityFrameworkCore.SqlServer –version 3.1.9
```

```
PM> dotnet add package Pomelo.EntityFrameworkCore.MySql –version 3.2.4
```

```
PM> dotnet add package Microsoft.AspNetCore.SpaServices –version 3.1.9
```

```
PM>  dotnet add package Microsoft.AspNetCore.Mvc.NewtonsoftJson --version 3.1.9
```

```
PM>  dotnet add package Microsoft.EntityFrameworkCore --version 3.1.9
```

```
PM>  dotnet add package Microsoft.AspNetCore.SpaServices.Extensions --version 3.1.8
```

NPM Packages Installation

Next, installations of packages relating to ReactJS.NET libraries can be done through the Package Manager Console by running these commands. You might choose to run the same set of commands from the Developer PowerShell instead of Package Manager Console where progress bars are more visual and user friendly showing the installation progress.

```
PM>  cd ClientApp
```

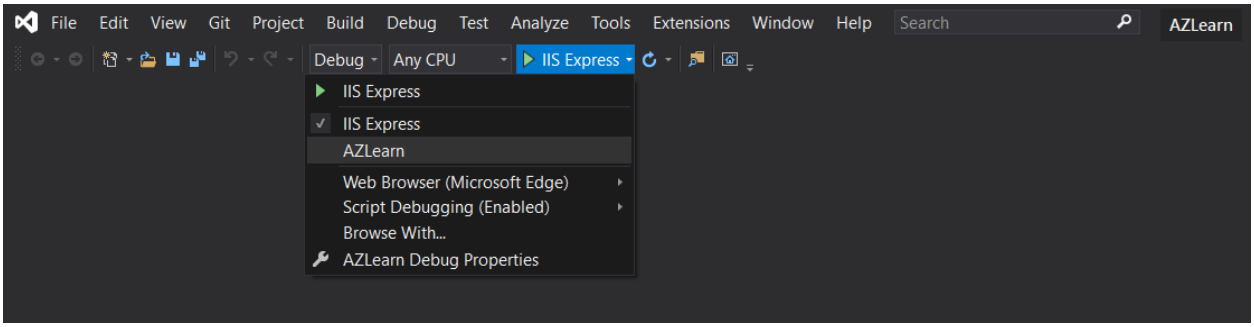
```
PM>  npm install
```

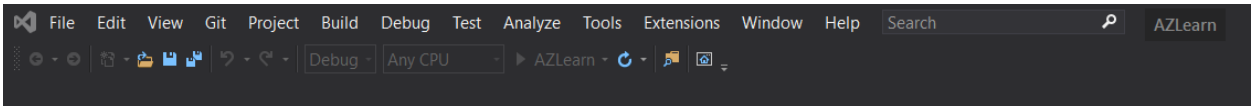
Once the installation is complete, you should see similar screen to below,

```
added 1994 packages from 828 contributors and audited 1998 packages in 81.844s

120 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

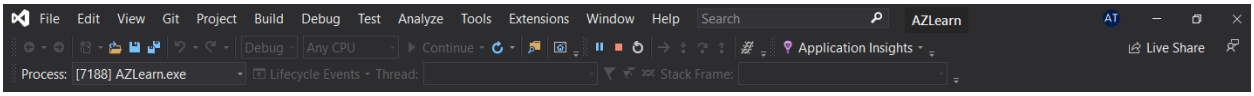
Application Initial Run

Change the complier run option from IIS Express to AZLearn, and run the application (click on the green angle triangle button)

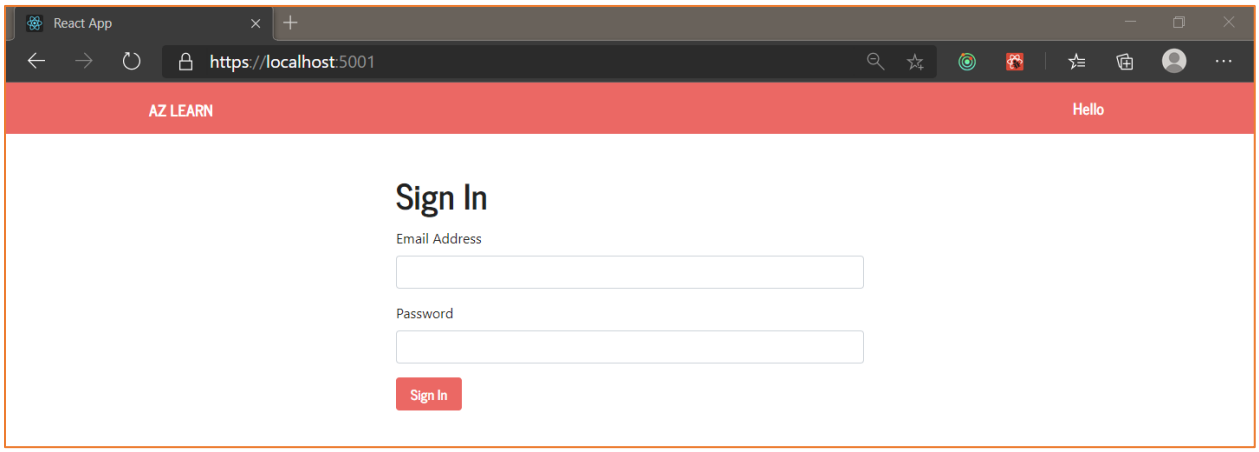




If all previous steps were done properly, you should see the run bar as in following screenshot. Note the additional buttons that showed up like the stop and pause buttons.



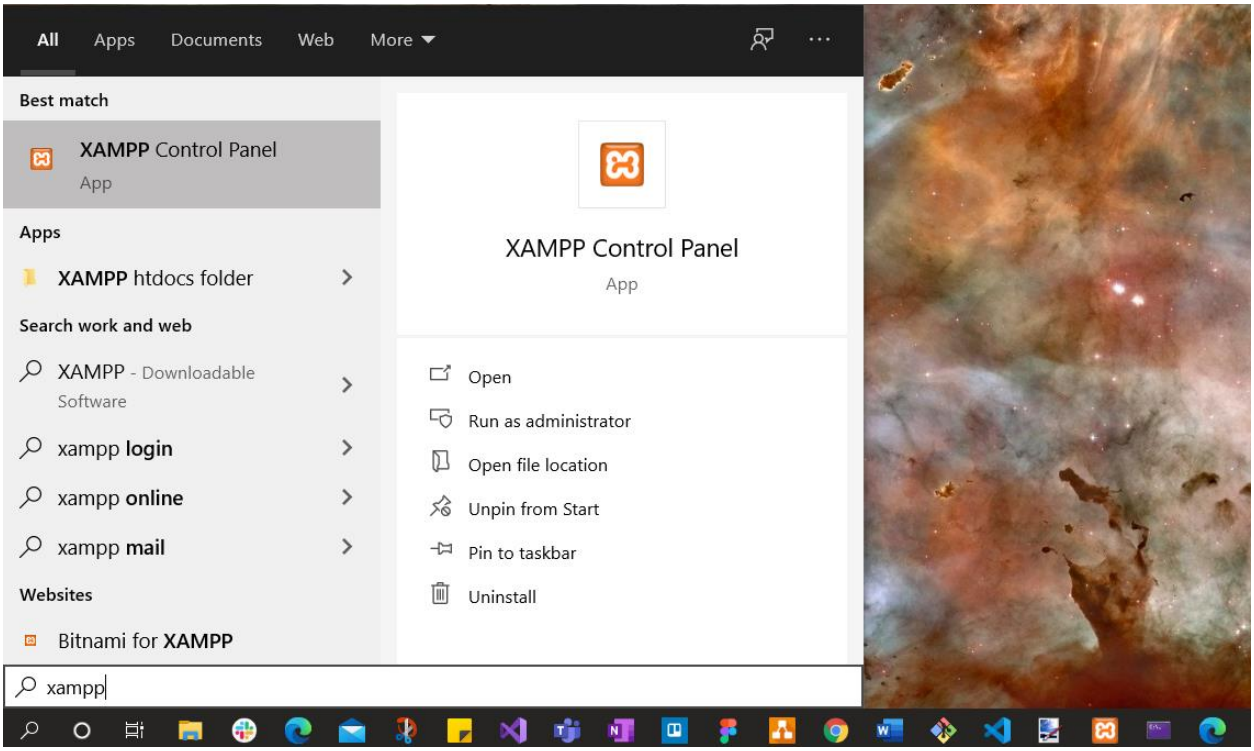
Also, your default browser should run and open the application sign in page as follows. Note that the localhost port number is 5001. This is important as this port is the one used to link the application both ends.



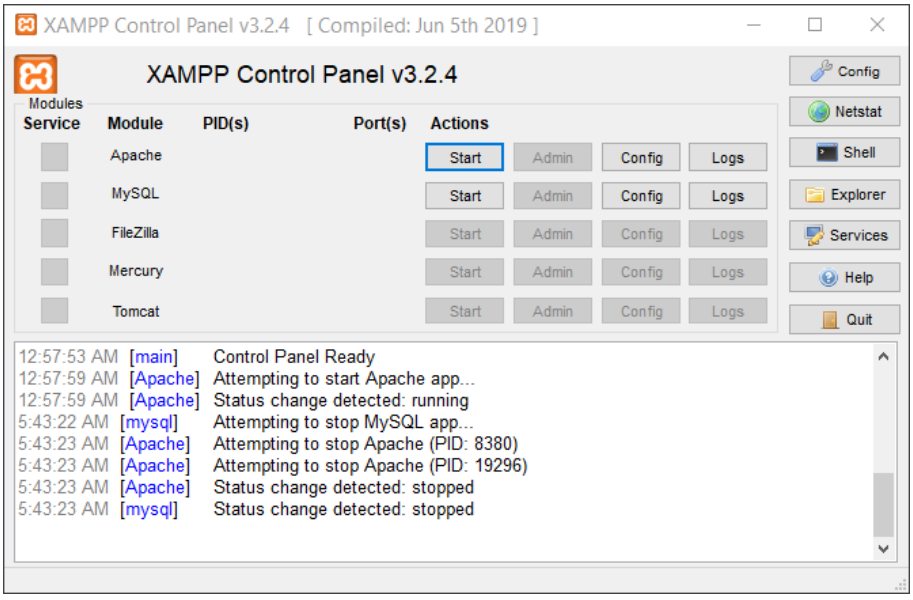
At this point, the application is not yet ready for use, as we are missing the backend database setup, which will be coved in the following steps.

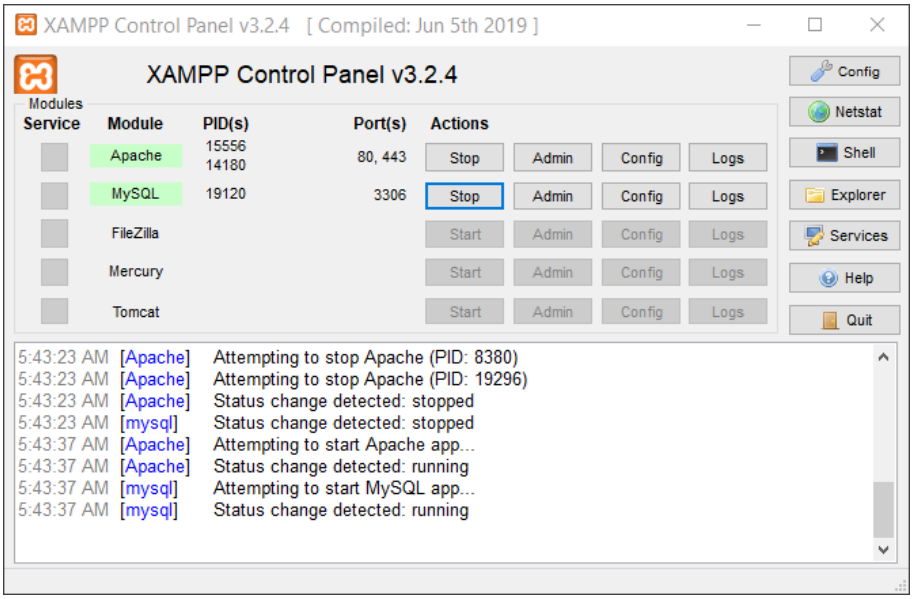
Database Server Start

To establish a database connection using MariaDB via XAMPP. Run XAMPP from your windows start menu.

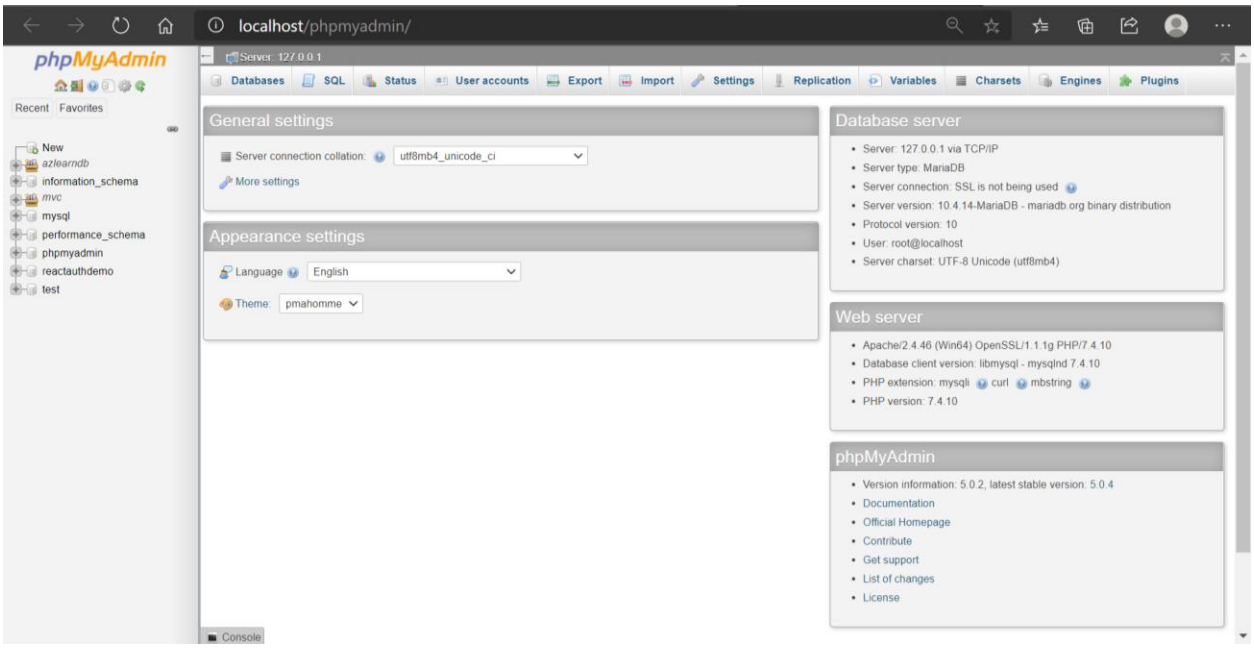


Once XAMPP is open, click on MySQL and Apache Start buttons to start the servers





Next, click on Admin next to MySQL to redirect you to phpMyAdmin dashboard.

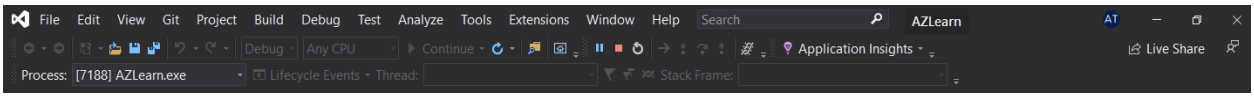


Database Migration

The application is built on Code Frist approach; hence first migration is required to create database. The following steps are meant to guide you through the app first migration.

So, back to Visual Studio 2019 – assuming the project still open, if not, refer to the first few steps of this document.

First, stop the application is running from the red stop button.



Then, using Package Manager Console, make sure you are still in the project folder, where AZLearn.csproj is using `PM> ls` command. It is likely that you are still in ClientApp folder from the previous ReactJs installation. If that is the case use `PM> cd ..` to go back to the project folder.

If you are in the right folder, you should see something like the following screenshot when using `PM>ls`

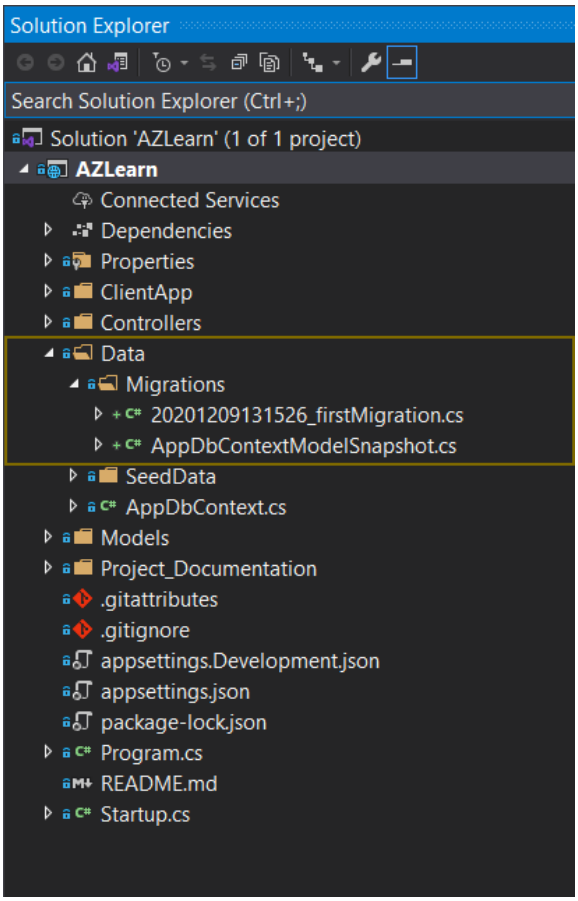
```
PM> ls


    Directory: C:\Capstone Project\Playground\capstone-project-a-to-z


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2020-12-09   5:04 AM                .vscode
d-----        2020-12-09   5:06 AM                bin
d-----        2020-12-09   5:04 AM                ClientApp
d-----        2020-12-09   5:04 AM                Controllers
d-----        2020-12-09   5:04 AM                Data
d-----        2020-12-09   5:04 AM                Migrations
d-----        2020-12-09   5:04 AM                Models
d-----        2020-12-09   5:06 AM                obj
d-----        2020-12-09   5:04 AM                Pages
d-----        2020-12-09   5:04 AM                Project_Documentation
d-----        2020-12-09   5:04 AM                Properties
-a----        2020-12-09   5:04 AM           2581 .gitattributes
-a----        2020-12-09   5:04 AM          11231 .gitignore
-a----        2020-12-09   5:04 AM            168 appsettings.Development.json
-a----        2020-12-09   5:04 AM            192 appsettings.json
-a----        2020-12-09   5:04 AM           4045 AZLearn.csproj
-a----        2020-12-09   5:04 AM           1118 AZLearn.sln
-a----        2020-12-09   5:04 AM          10752 package-lock.json
-a----        2020-12-09   5:04 AM            503 Program.cs
-a----        2020-12-09   5:04 AM           2349 README.md
-a----        2020-12-09   5:04 AM           2700 Startup.cs
```

Using Package Manager Console, run the following commands for database first migration.

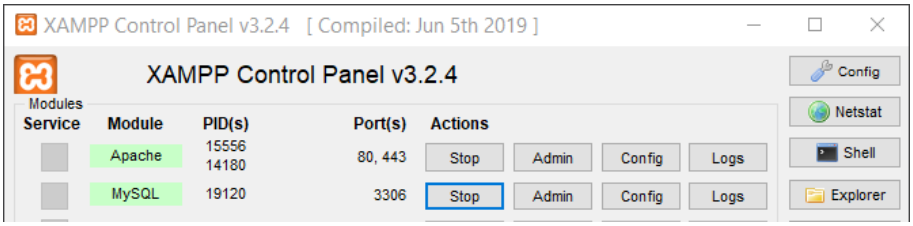
```
PM> dotnet ef migrations add firstMigration --context AZLearn.Data.AppDbContext -o Data/Migrations
```

If the first migration was successfully processed, you should see messages like in below screenshots. And the following files would have show up in the solution explorer.

```
PM> dotnet ef migrations add firstMigration --context AZLearn.Data.AppDbContext -o Data/Migrations
No project was found. Change the current working directory or use the --project option.
PM> cd ..
PM> dotnet ef migrations add firstMigration --context AZLearn.Data.AppDbContext -o Data/Migrations
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
PM>
```



Make sure your XAMPP is still running and both of Apache and MySQL buttons are still green.



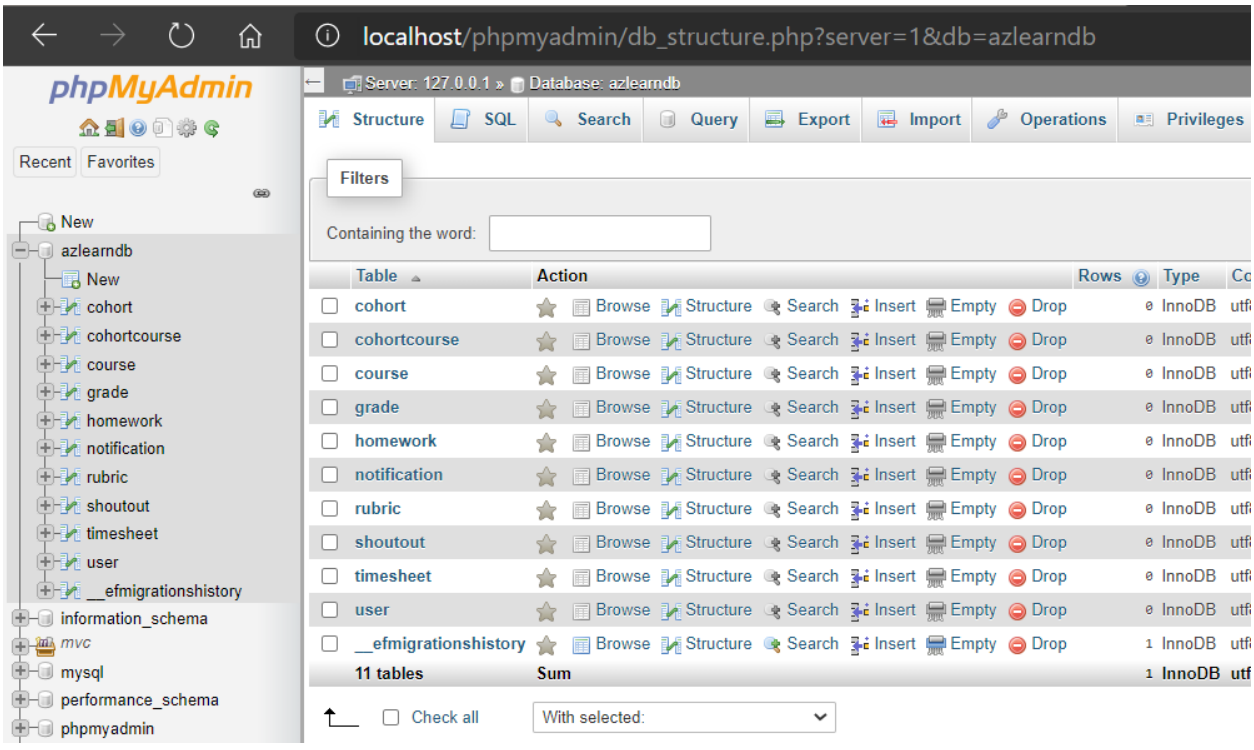
To update the database, Using Package Manager Console, run the following command

```
PM> dotnet ef database update
```

If all is done right so far, you should see successful migration update like below in the console

```
PM> dotnet ef database update
Build started...
Build succeeded.
Applying migration '20201209131526_firstMigration'.
Done.
PM>
```

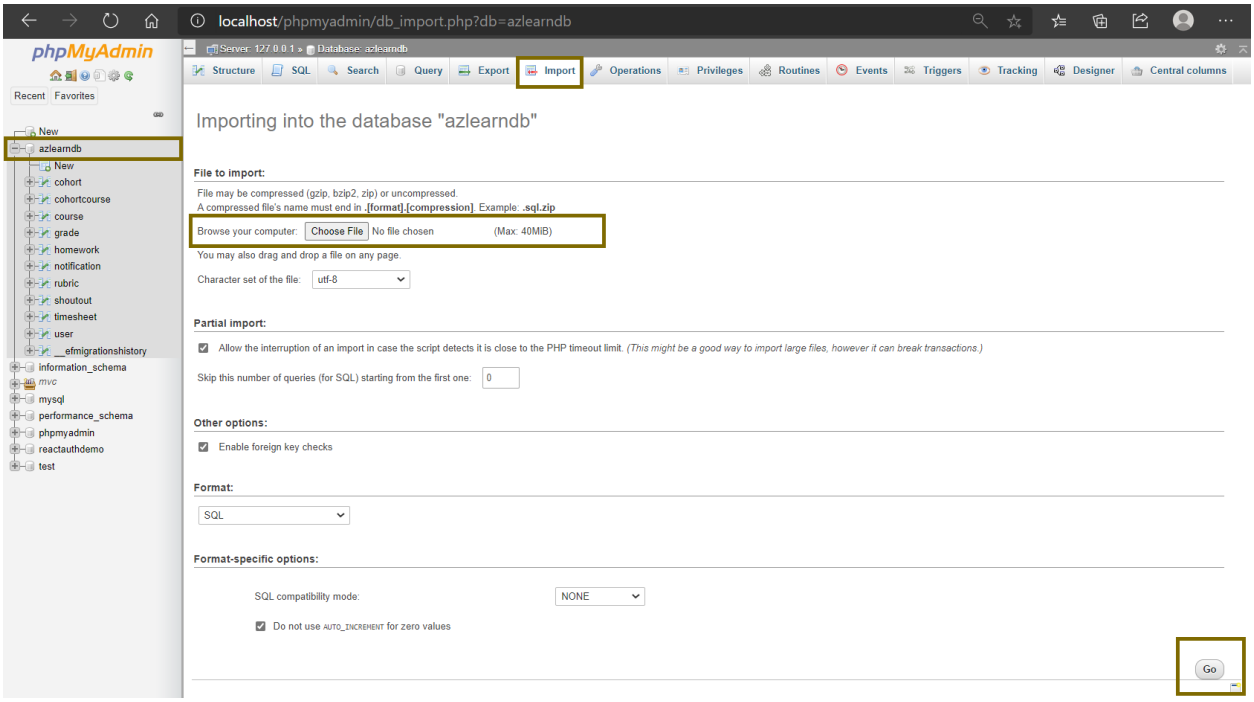
And in the phpMyAdmin, the database would be created, and should look like the following screenshot (N.B., do not forget to refresh your phpMyAdmin page on the browser to see the changes post migration). Note that azlearndb is now in your database list. Also, take note that the Rows column is showing 0 all the way.



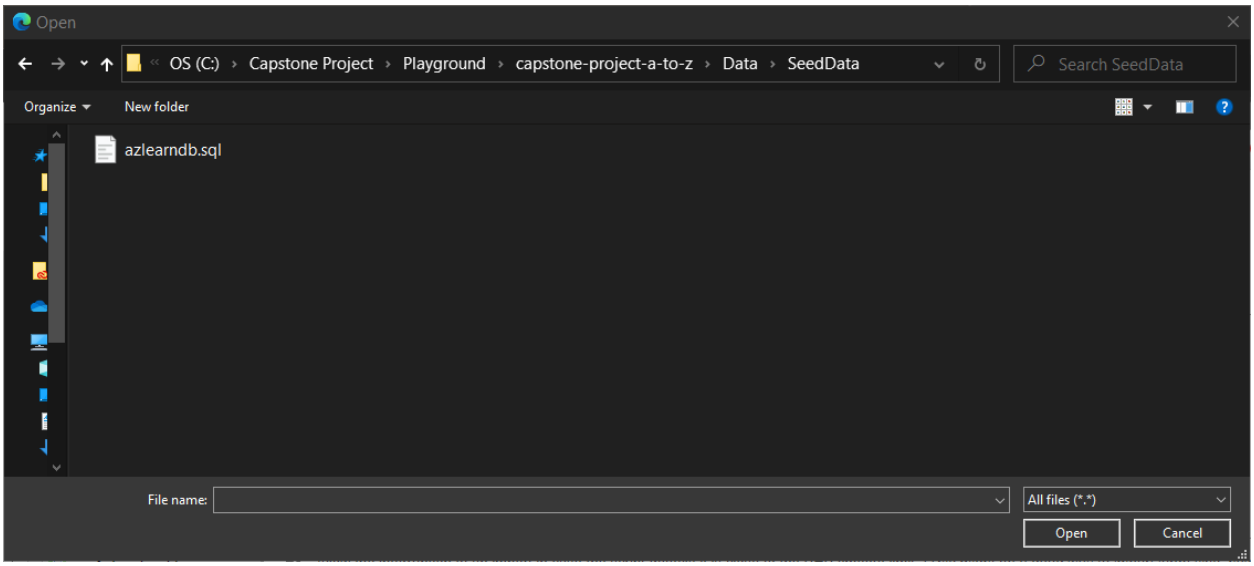
Database Data Seeding

To test the application with actual database data, you will need to seed data, and for that follow the following few remaining steps done in phpMyAdmin page.

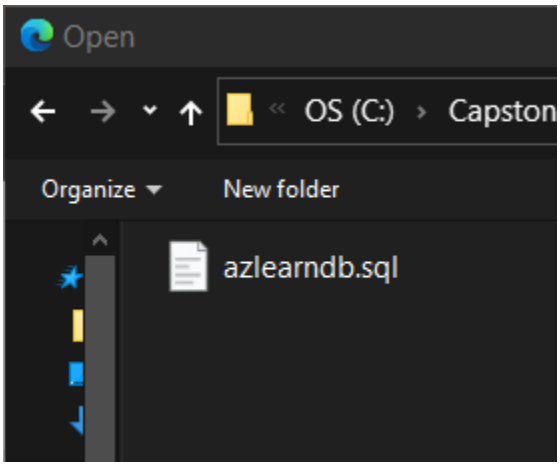
Click on the database azlearndb and then click Import tab as shown below.



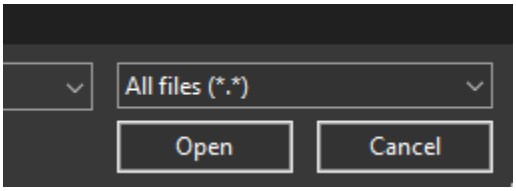
Click Choose File to upload the sql script that has the seed data.



Inside the project folder, navigate to Data/SeedData where you will find azlearndb.sql file.

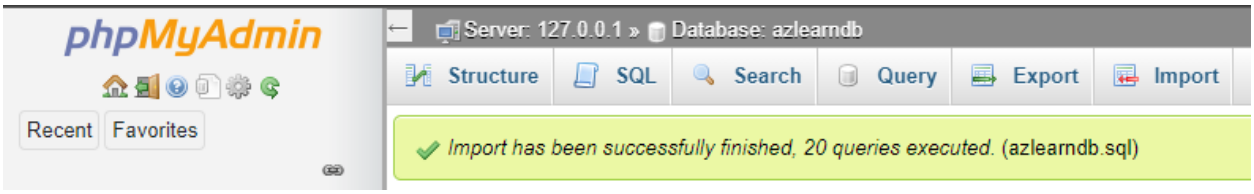


Choose azlearndb.sql and click Open.



Now, click Go in phpMyAdmin page to run the script file

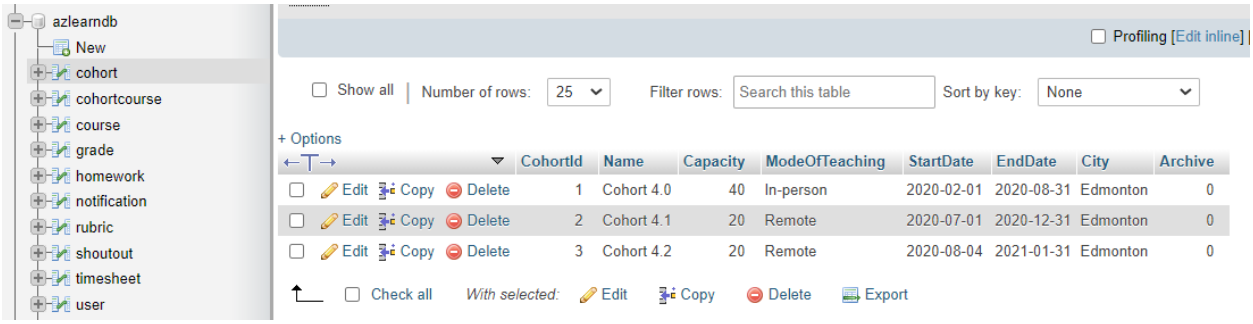
Upon successful script run, a success message will be displayed



Now, go back and click on azlearndb database to double check that all tables are populated with seed data. Note that the Rows column is now showing count of the available records.

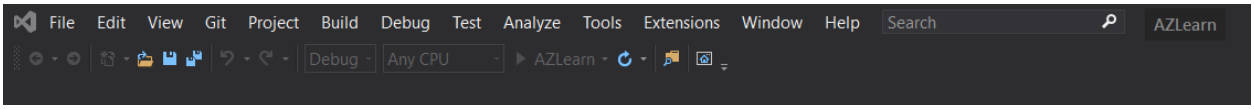
	Table	Action	Rows	Type	Collation	Size	Overhead
☐	cohort	Browse Structure Search Insert Empty Drop	3	nnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	cohortcourse	Browse Structure Search Insert Empty Drop	15	nnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	course	Browse Structure Search Insert Empty Drop	15	nnoDB	utf8mb4_general_ci	16.0 KiB	-
☐	grade	Browse Structure Search Insert Empty Drop	2,640	nnoDB	utf8mb4_general_ci	224.0 KiB	-
☐	homework	Browse Structure Search Insert Empty Drop	15	nnoDB	utf8mb4_general_ci	64.0 KiB	-
☐	notification	Browse Structure Search Insert Empty Drop	0	nnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	rubric	Browse Structure Search Insert Empty Drop	132	nnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	shoutout	Browse Structure Search Insert Empty Drop	0	nnoDB	utf8mb4_general_ci	112.0 KiB	-
☐	timesheet	Browse Structure Search Insert Empty Drop	300	nnoDB	utf8mb4_general_ci	48.0 KiB	-
☐	user	Browse Structure Search Insert Empty Drop	26	nnoDB	utf8mb4_general_ci	32.0 KiB	-
☐	__efmigrationshistory	Browse Structure Search Insert Empty Drop	1	nnoDB	utf8mb4_general_ci	16.0 KiB	-
	11 tables	Sum	3,147	nnoDB	utf8mb4_general_ci	656.0 KiB	0 B

Moreover, if you click on any of the tables, you should see the actual rows, for example, click on cohort table. Now, the seed data is available to use the application.



Assurance Test

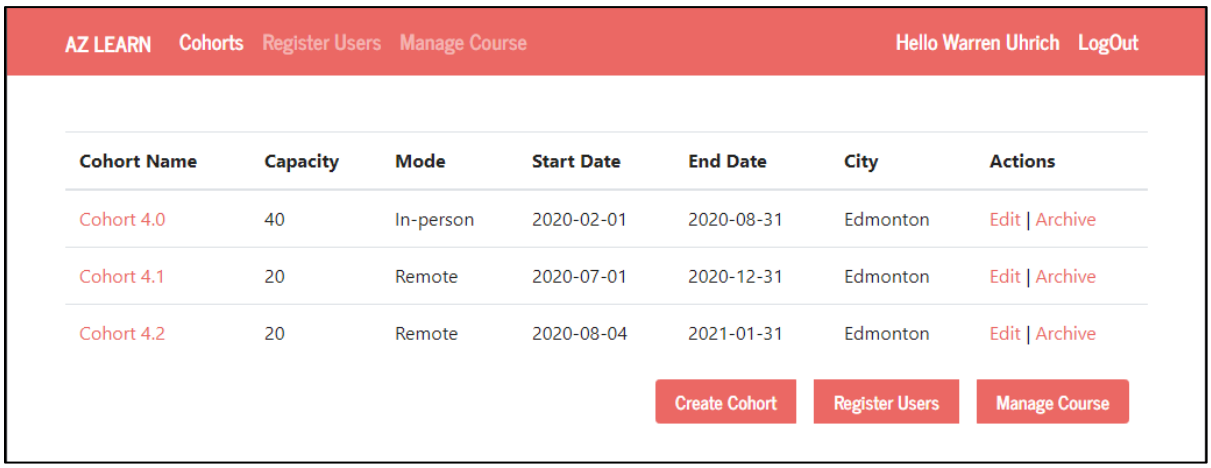
To confirm that everything is tied together, go back to Visual Studio 2019, and run the application using AZLearn.



Once the application opens, try to login using the following user credentials

(email: warren.uhrich@azlearn.ca, password abc.123)

Upon successful login, you will be in the instructor dashboard (i.e., Cohorts Summary).



For more details on how to use the app, please refer to the Application User Guide that can be found in the project GitHub Repo, under Project Documentation Folder.

For more details on how to test the app, please refer to the Application Testing Instructions that can be found in the project GitHub Repo, under Project Documentation Folder.

We hope you have enjoyed using this document and that it was trouble free. In case you have any support needs, please do not hesitate to reach out to the project contributors for support or clarifications.

End of Document