ChatTestTask

============

Task was:

--------------

Create a simple chat page in .net core, using c#, angular, signalR, Sql Server.

Make use of ajax (or similar) to refresh when new chat messages arrive. Pick your choice of framework.

My solution description:

--------------

1. Solution implements very simple chat web application. To emulate multi-user environment you have to copy application URL to several web browser instances, so you will be able to send chat messages from one browser window to another.

2. Solution doesn't support user authentication.

3. Solution is based on next technologies:

- .NET Core - as main platform

- Angular 8 - as webb app hosting

-.NET Core SignalR - to simplify implementation of "real-time" application behavior

- Entity Framework Core (Code First) - as a basement of data access layer

4. ASP.NET SignalR is based on jQuery 2.0.3, so solution must be supported both by desktop and mobile browsers which are supports jQuery.

5. Template of Data Access Layer architecture based on EF has been taken from real project. DAL implements Unit Of Work pattern in its simplest form.

6. (local) used as application data source.

Installation:

-----------

1. Install Visual Studio 2019
2. To Run in IdentityAndAuthorizationServer folder .sln file and in appsettings.json folder change the connection string.
3. Then play this solution.
4. Ef core setup database migration nugget console write : add-migration [migration\_name]

and then write Update-Database

1. Here another folder one is AngularClient.In this where is src that folder address

Bar write cmd open then write npm install (If version update then write- npm update)

1. Then write npm stat in cmd prompt build the angular solution when completed build here

See the application run http:\localhost:4200 .Write this copy path and paste press enter then solution is loaded.

User's guide:

--------------

1. After application start you will have to define your chat first name, last name, Email. After your enter you email you will see chat window consists of:

2.1. Test field for new messages.

2.2. Send message button.

2.3. Chat log.

3. Each time user enter the chat he/she will:

3.1. Receive greeting message and date of his last appearance in the chat.

3.2. List of latest messages from other users published in the chat.

4. To emulate multi-user interactions you have to copy application URL and paste in address string of another instance of web-browser. So users will be able to send messages to each other.