

Coding Task

1. It's built using; MS SQL Server, ASP .NET Restful API Controller, Swagger, XUnit, Unit Test (arrange, act, assert), Angular
2. Database file is located in: \database\db.bak
 - a. please restore it using MS SQL Server Express 2012 (or later)
3. Some approaches used here:
 - a. TDD (Test Driven Development): on Infrastructure (such as mockup data), and API (test real API in some environment)
 - b. DDD: <https://docs.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice>
 - c. SOLID principles; dependency injection using Autofac, Unity
4. Technical inclusions:
 - a. A simple page of CRUD (Create, Refresh, Update, Delete) with no pagination.
 - b. A listing page with pagination functions, can list down all cars with certain page size and page index.
 - c. Single Page Application. Using bootstrap and Angular
 - d. 2 Restful API, but choose 1 to invoke (.Net Core or .Net Framework)
 - e. Testing projects: integration testing (invoke API or connect database)
5. Steps to deploy in local PC:
 - a. Prepare a folder for the solution project, ex: C:\projects\angulardotnetpoc
 - b. Deploy Restful API: you can choose .Net Core or .Net framework
 - i. For .Net Core
 1. Open SLN file (with Visual Studio 2019); main .net core.sln
 2. Restore all nuget package, build all solutions
 3. Please adjust these files on DotNetCoreApi.csproj; appsettings.json
 4. Publish API of DotNetCoreApi.csproj using default publish.pubxml
 - a. The target folder: /bin/publish
 - b. Please change the target publish folder if required
 5. Create website in IIS, which folder is /bin/publish (the one specified above)
 6. For testing purpose, please bind it to <http://localhost:5000> (this is written in angular file)
 7. On the API Tester csproj. Ex: XUnitApiTester.csproj
 - a. Please adjust: appsettings.json (ex: this is to simulate API testing in Staging or DEV)
 - b. In this example, <http://localhost:5000> represents DEV
 - ii. For .Net Framework
 1. Open SLN file (with Visual Studio 2017); main .net fx.sln
 2. Restore all nuget package, build all solutions
 3. Please adjust these files on RestfulApi.csproj; ConnectionStrings.config
 4. Publish API of DotNetCoreApi.csproj using default publish.pubxml
 - a. The target folder: /bin/publish
 - b. Please change the target publish folder if required
 5. Create website in IIS, which folder is /bin/publish (the one specified above). Ex: <http://localhost:8065>
 - c. Testing Projects
 - i. Unit Testing Sample (.Net Framework)
 1. Open this SLN: main .net fx
 2. See InfrastructureTester.csproj, ApiTester.csproj
 3. For unit testing (testing logic purpose): just consume mockup class instead of real database connection
 4. For integration testing: we can connect database (see InfrastructureTester.csproj), or invoke API directly (see ApiTester.csproj)

- ii. XUnit Testing (.Net Core): for integration testing (directly invoke APIs) and Unit Testing (use mock data) for service layer.
 - 1. Using XUnit, FluentAssertions
 - 2. Open this SLN: main .net fx.sln
 - 3. Find this CSProj: XUnitApiTester.csproj
 - 4. See theory, fact, inlinedata. We can put some permutations within 1 method
 - 5. For integration testing, in this case we invoke API by using HttpClient (instead of database connection).
- d. Deploy UI (Angular)
 - i. Install node.js, as it's angular requirements, download from <https://nodejs.org/en/download/>
 - ii. Install node.js based on your OS, in my case, I use Windows Installer (.msi) 64bit
 - iii. Make sure node js and NPM is installed, open command prompt
 - 1. Run: node -v
 - 2. Afterwards you will see the node js version. Ex: v14.15.5
 - 3. Run: npm -version
 - 4. Afterwards you will see npm version. Ex: 6.14.11
 - iv. Prepare a folder for angular website. Ex: c:/websites
 - v. Install angular in that folder
 - 1. Run in command prompt: cd "c:/websites"
 - 2. Then run: npm install -g @angular/cli
 - 3. Make sure angular is installed, run: ng -version
 - 4. You'll see Angular version, ex: Angular CLI: 13.0.4
 - 5. Create angular website
 - a. Run this in the same folder: ng new main-ui
 - b. Then answer No
 - c. Then choose CSS
 - d. Wait until finish
 - e. Change directory to the newly created folder. Ex: cd "main-ui"
 - f. Install jquery, run:
 - i. npm install jquery --save
 - ii. npm install @types/jquery --save
 - g. Copy the front end source codes to this folder. Ex: from C:/projects/angulardotnetpoc/src/front-end to c:/websites/main-ui
 - h. Then run the angular, either prod or dev mode. Run in command prompt:
 - i. cd "c:/websites/main-ui"
 - ii. ng serve