**Tasks & Instructions**

1. **JavaScript test (Angular Task(Took me 4 Hours to complete the task)):**

* Created the project using Angular-10 (latest version).
* Search the movies using OMDB-API.
* Created the account on API (to get the API-Key).
* Design the whole application.
* Implement the search feature.
* Implement the routing on components.
* Implement substrings for long movie names.
* Implement the full pager view of the poster.
* Please use CMD or Git-Bash for running the project ( cd to the directory angulartask make sure you have the latest angular cli version(10) installed on global scale, then type in the ng serve command project will be build and open your browser(google chrome, Mozilla Firefox etc.) and type in the URL section localhost:4200)).

1. **C# test and algorithm skills (Sudoku Challenge(Took me 5 Hours to complete the task))**

* Took the challenge from the page <http://www.exeliatech.net/tests/SudokuPuzzleValidator.cs>
* Implement the functionality as required.
* Made a little change on validation method for the sub square size because we have the multiple size sub squares.
* It is a console application start the project and see the result.

1. **C# test for web based back-end((Took me 3 Hours to complete the task)**

* Created the project in .Net Core 3.1 (API Structure)
* Project was coded using Code First approach
* Implement the required functionalities
* Open the project solution and Just change the connection string in appsettings.json
* Make sure you have installed .Net Core 3.1
* Perform the migrations
* Run the project and note the URL in the browser
* Use Postman or fiddler or any other API testing tool
* For getting all the Breweries use request type (Get) and

localhost:44385/ Breweries

* For getting Breweries by ID use request type (Get) and

localhost:44385/ Breweries/1 or any other ID

* For Updating the Brew use request type (PUT) and

localhost:44385/ Breweries/1 or any other ID

* For deleting the Breweries use request type (Delete) and

localhost:44385/ Breweries/1 or any other ID

* For adding the Brew use request type (POST) and

localhost:44385/ Breweries with the Jason object

* For searching the Breweries use request type (POST) and

localhost:44385/ Breweries/Search? searchVal=anyname

* For Adding the Rating of the Breweries use request type (POST) and
* localhost:44385/ Breweries/AddRating?Brewid=2& RateVal=4
* Rate value must be within 1 to 5. It will auto calculate the average of the all ratings and update the table with the average value.
* It’s an open API does not need any API-Key or any other security.
* For the security of the API we can use Json Web Tokens(JWT).