.Net Developer Test

Please read the entire document before starting your test, you will have a period of 1 hour to send concerns of the requirement you may have, these will only be attended if they are related to doubts in the way the final product should look or function (please attach to your question the section of the paragraph that generated that doubt) will not be considered technical questions to be clarified.

The application must be developed in the version of **.Net Framework 4.6.1** using the pattern MVC and using the database described in the mail. In case of not being able to use this .NET version, please explain main reasons with technical sustentation.

It is required to build a web application with a Login system managed from the **backend**, must have 2 nature of roles: **Admin and Regular User**.

The Admin role must have the capacity to manage, create, read, update, delete (**CRUD**) both **Admin and Regular User** in another window of the application that is not the home page.

When the user entering to the application, this one should greet and if it's an user **Admin** should greet and show a button in the greeting frame that leads to a history of all records of all login of the application with date and user name.

The home page should show the following information of the logged user: Role, Age, Gender and Nationality, if you are **Admin** you must have the ability to modify these values.

Users must be able to navigate to a next page where they can reserve a list of products described below.

There are 4 products which can be requested by users with a described availability:

Description	**Items amount**
Product 1	2 units
Product 2	1 unit
Product 3	50 units
Product 4	1 unit

Regular users will have to be subject to product availability and Admin users will have the ability to reset product availability values one by one as long as they enter a verification key that must be changeable from a separate API, which must be created in the same solution and where it must only execute the Post method and the request body will be the full name of the developer plus the new key in case it is not the developer's name should not overwrite the previous key and return an error message.

Then create a separate window with a list of products that the user has reserved and in the case of being an Admin user, this one should see a list by product with the names of those who have reserved it.

The use of the Entity Framework and Owin will be assessed, as will the error controls, data security, security improvements and functionalities between the Front-End interaction and the API created within the solution. (you are free to use the API for extra things in the Front-End). It will also be deterministic the layout and usability of the application, feel free to use the Front-End tool or library that you consider relevant. Feel free to create the database model that you consider optima.