# Dashboard Demo Guide

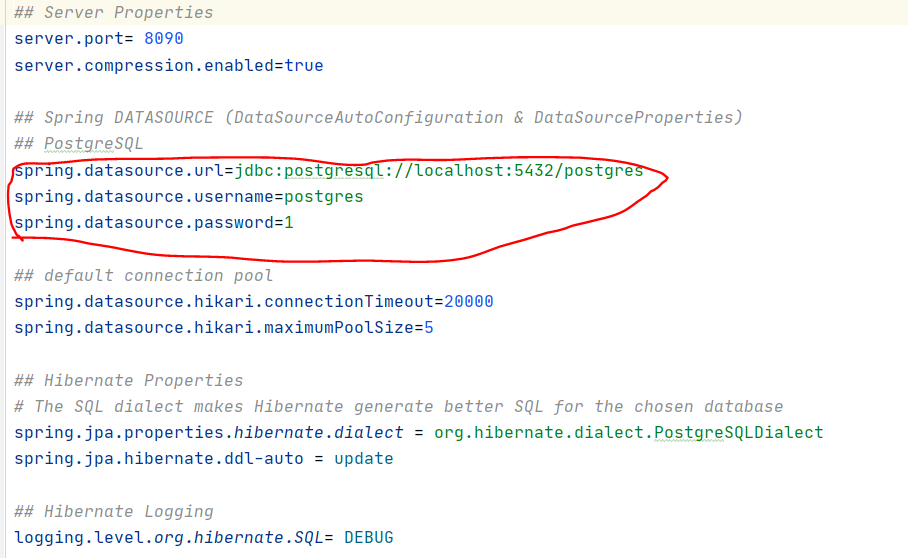
## Dashboard App Server

### Import the dashboard-app-server to your IDE

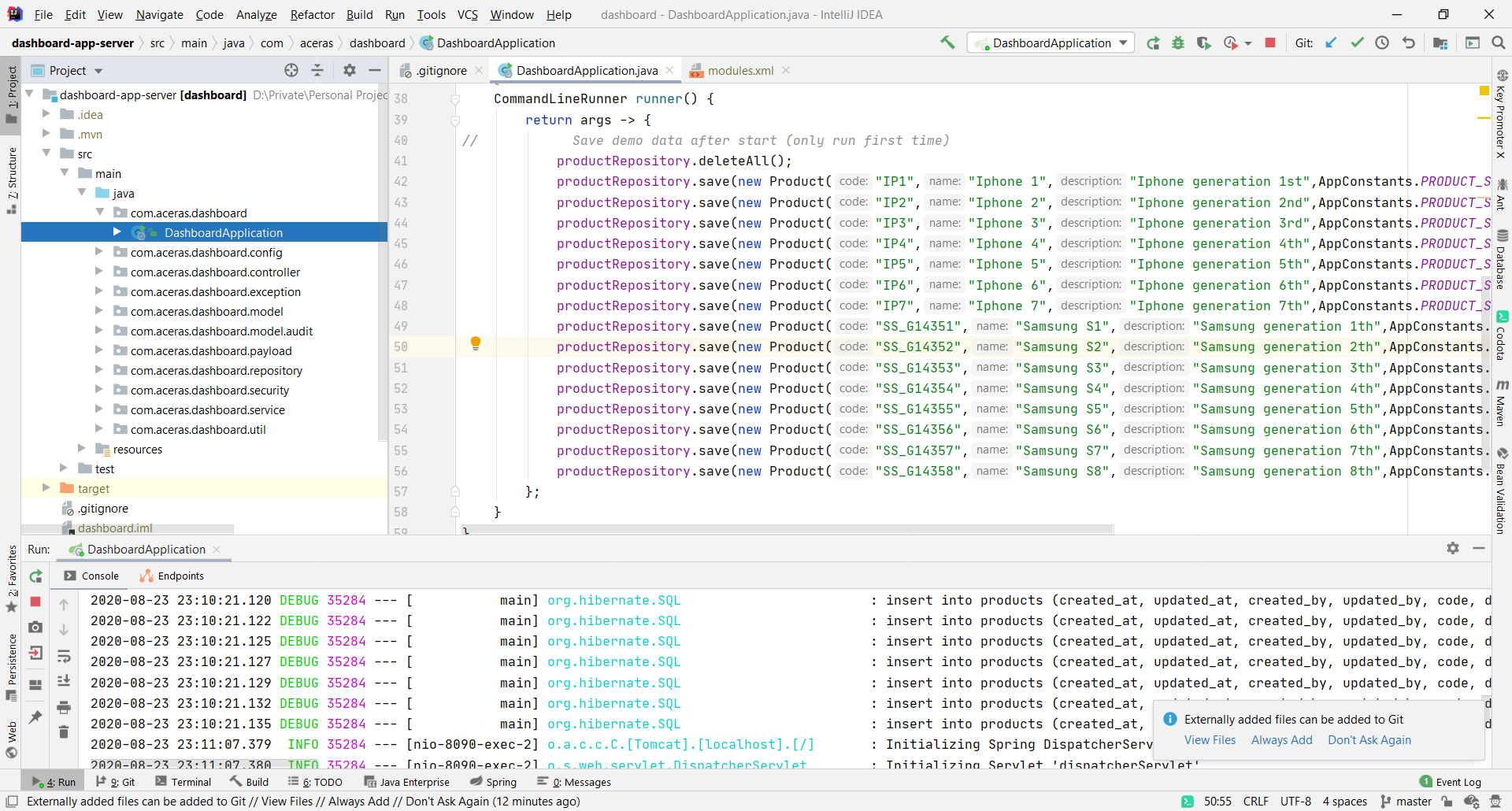
### Setup the JDK 1.8 for that project (if your IDE does not use the JDK 1.8 as default)

### Update the server.port

### Setup postgre then set the postgre server information into application.properties file



### Run the file DashboardApplication



## Dashboard App Client

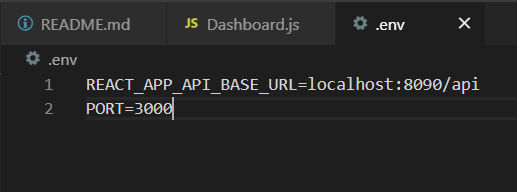
### Import the dashboard-app-client to Visual Studio Code.

### Run “npm install” in terminal

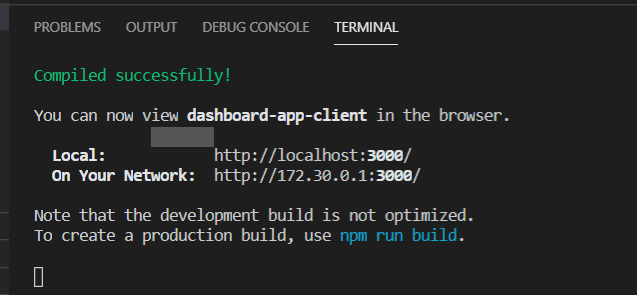
### Update the environment in .env file

#### Dashboard app Server address

#### Client server port



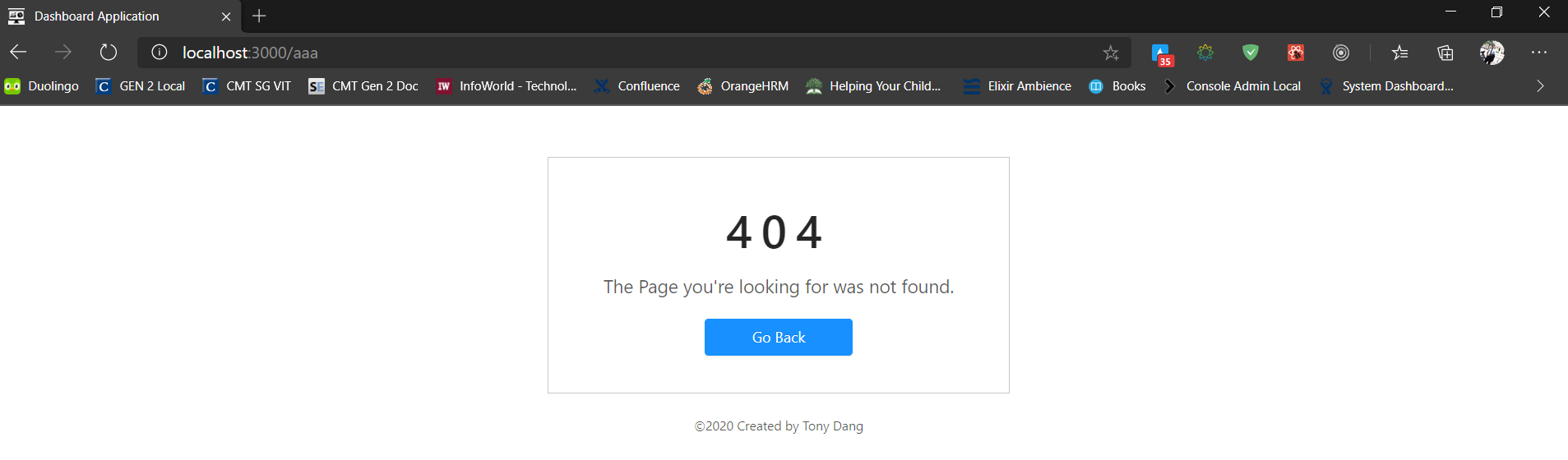
### Run “npm start”



## Demo functions

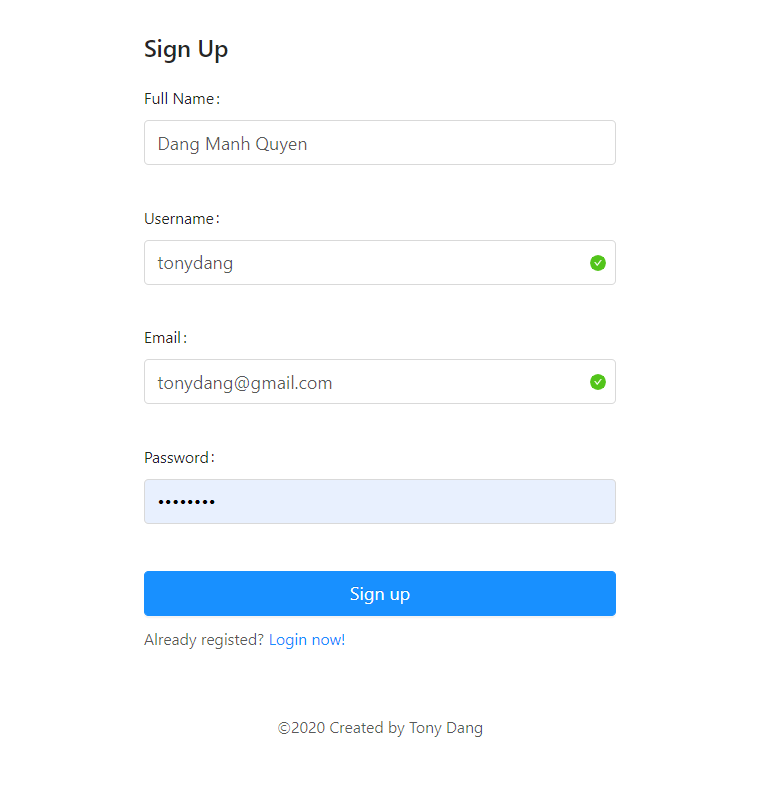
I protected the dashboard page as only login and sign up screen could be accessed with a guest.

* If a guest access to protected URL (this case is localhost:3000/dashboard), the application will redirect to /login page.
* If a guest access to unknow URI, the Not page found will be displayed.

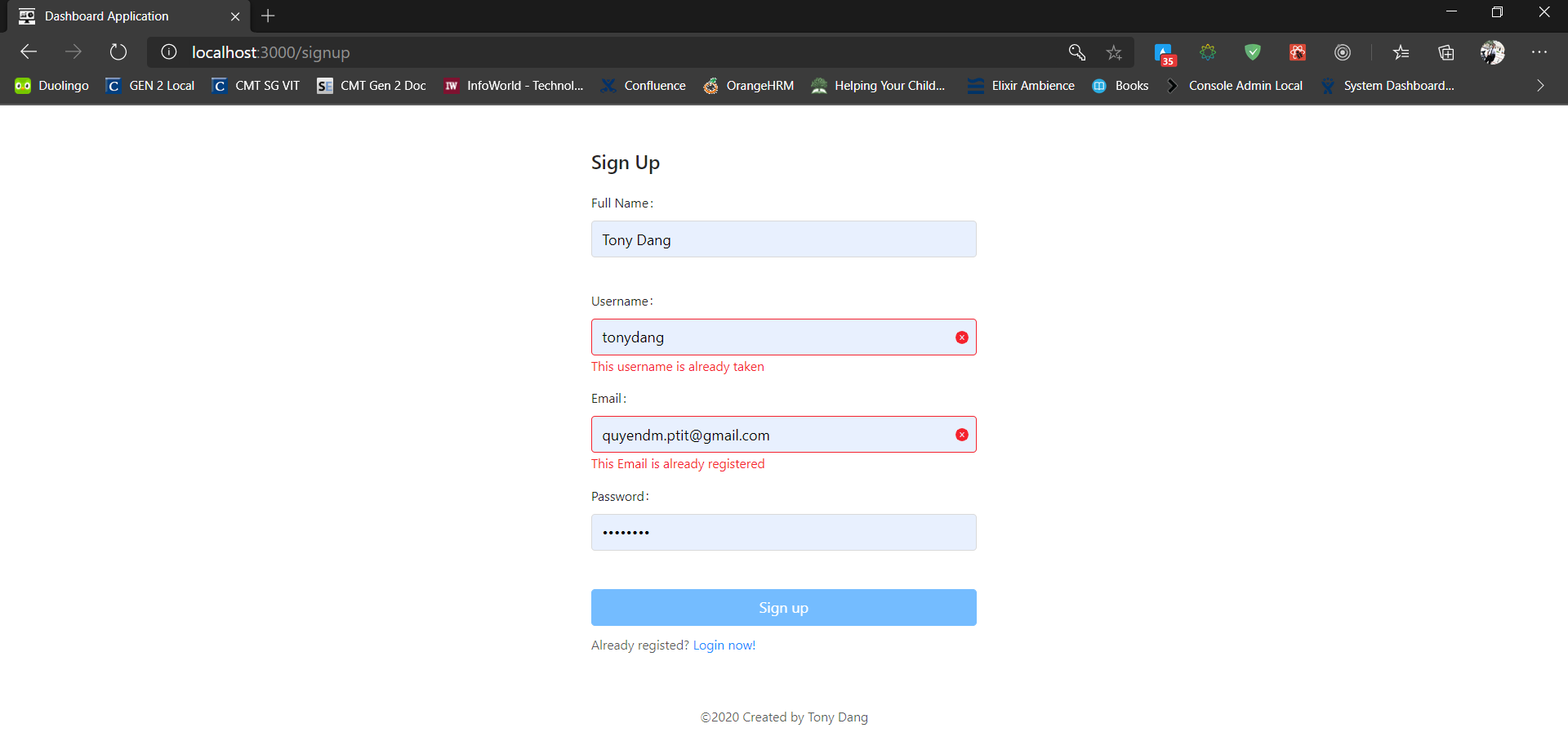


## Register page

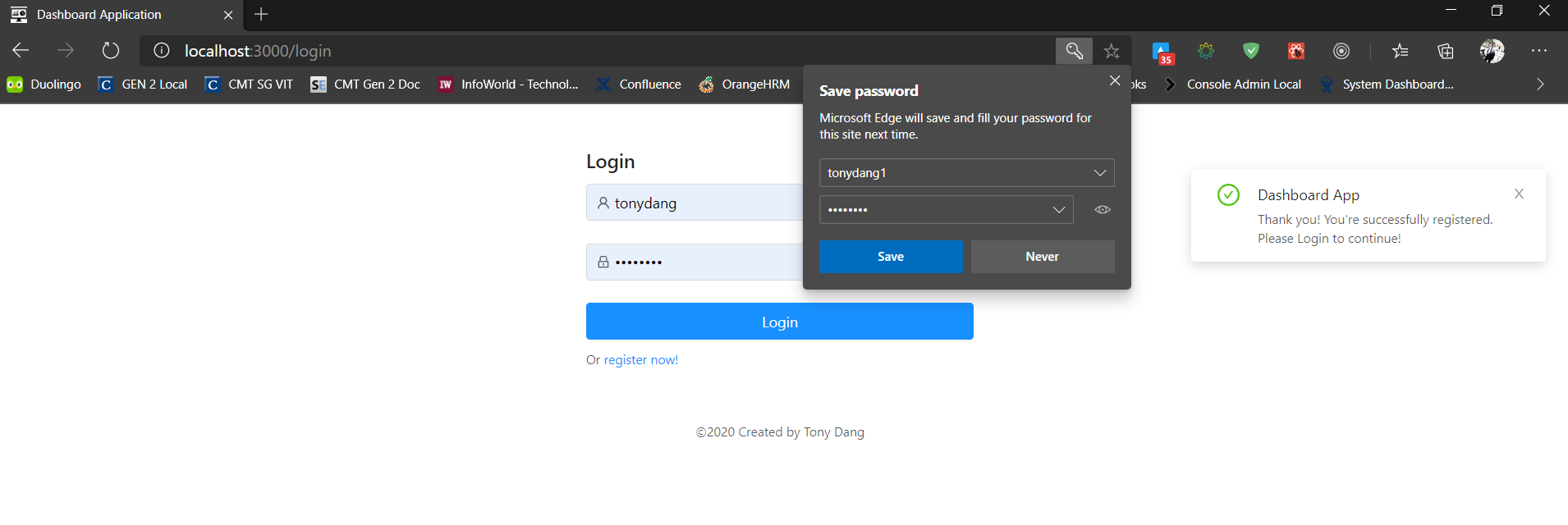
I added new function register to register new user to server



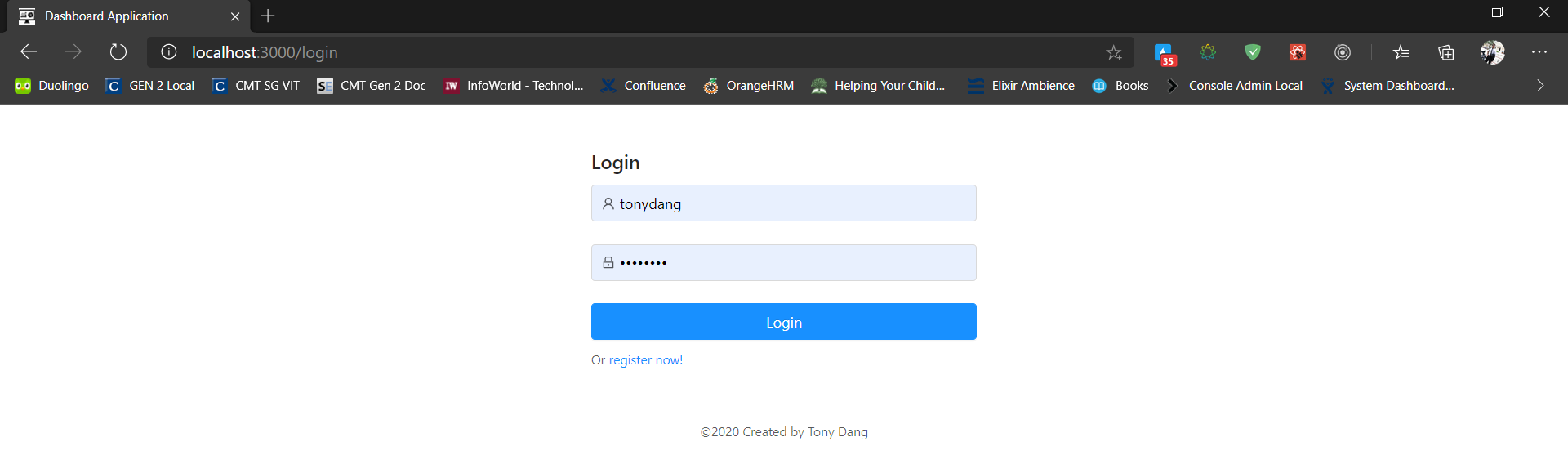
In register page, I added 2 verifications for username and email as unique key for an user.



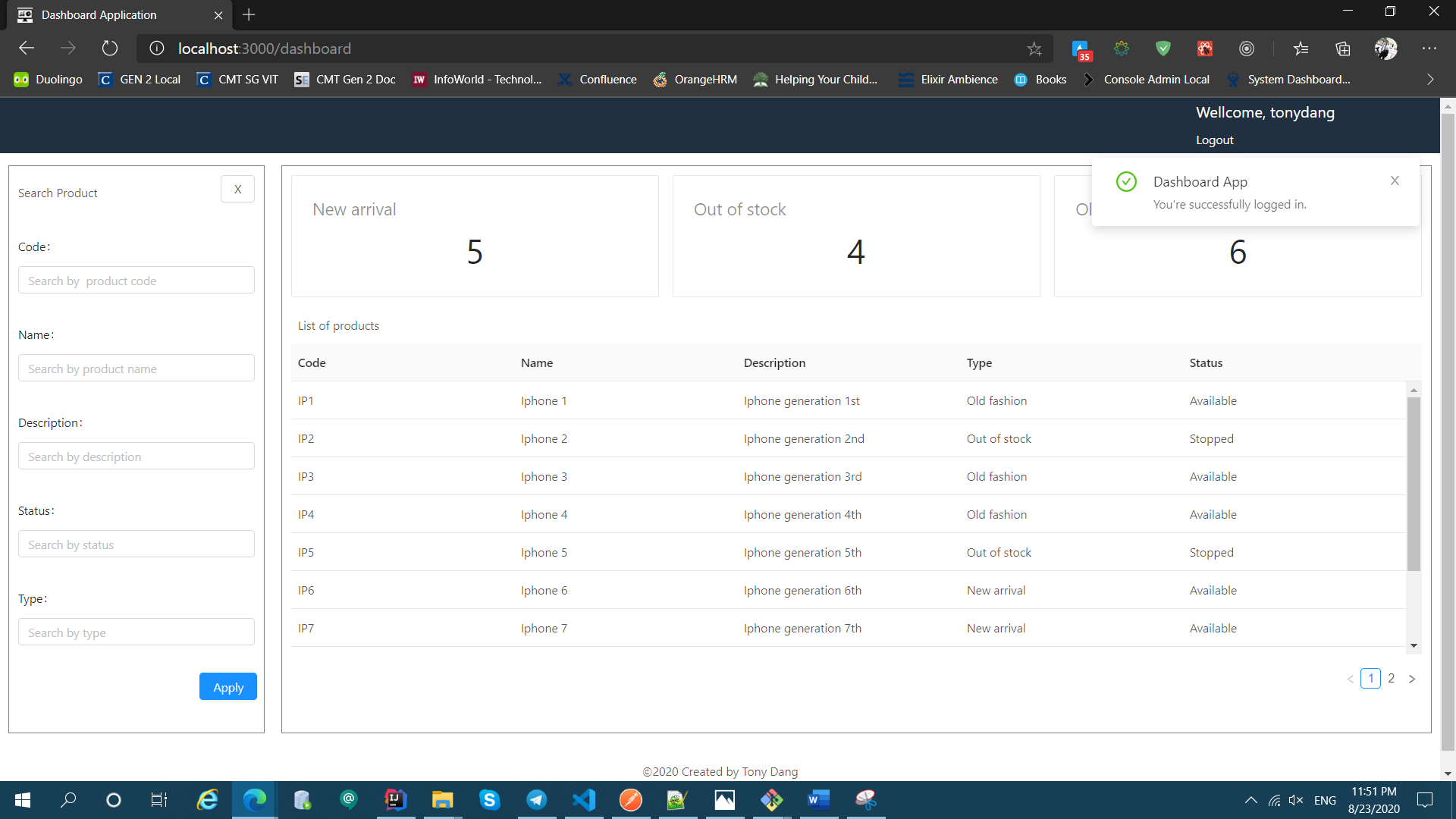
After register finished, a notification will be displayed as a welcome massage to new user and you will be redirect to login page.



## Login page



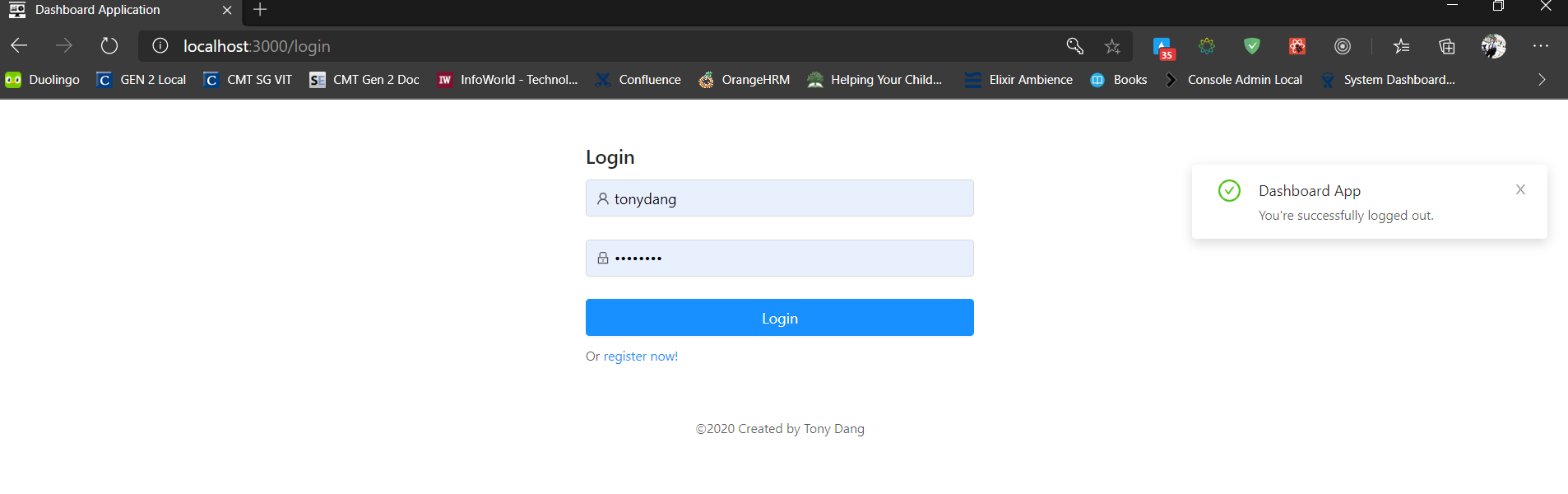
After logged in, the application will redirect to Dashboard page with a notification massage in the left corner of the page.



## Logout

I used the JWT token to verify the request, after user login succeed, the server will return an JWT token to client application, the client application will store that key to localStorage as an item. When user logout or re-login, the token will be deleted from localStorage.

When user logout, the user will receive a notification and back to login page.

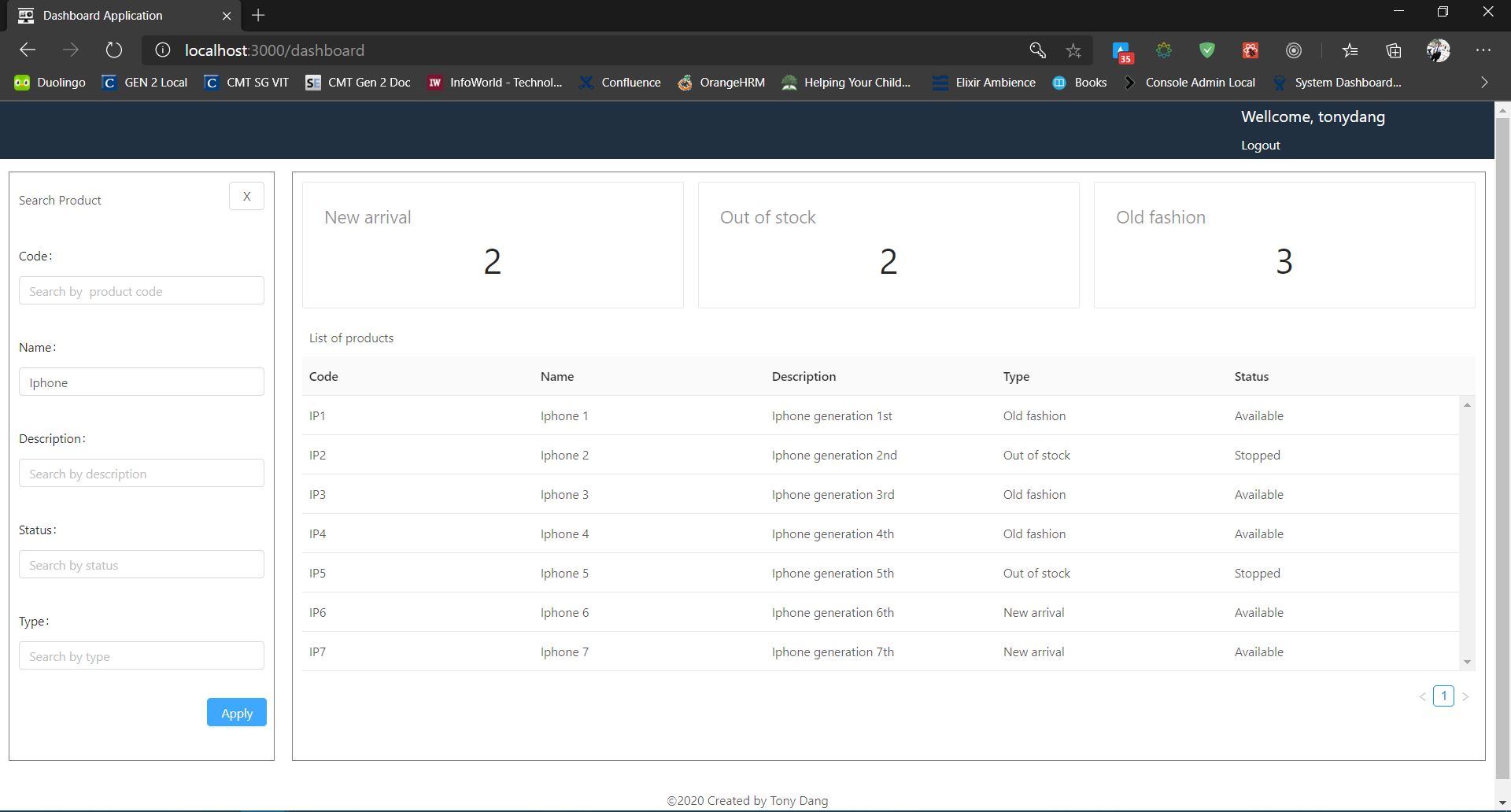


## Dashboard home

1. Description

This dashboard I used the products table to demo. It contains:

* A sidebar as a search product form – with all fields displayed in products list table.
* A summaries section above table which will count products in table by type.
* A table of products.
* By default, after login all products will be fetched from DB and display on screen.
* After user input to search product form then click apply, the data will be fetch following the search conditions in search form then set to the section summaries and table product.



PS: I updated the header with the name will be displayed in welcome note as below

