UI-101: Week 6 Assignments

This week assignments focus on the last couple topics that are covered during the training section including unit tests for AngularJS application, debugging Javascript codes techniques, and application performance analysis. By completing the assignments, candidates would be able to create tests for all the components in an AngularJS application like controllers, services, directives, filters, providers, etc. In addtion, technical skills of identifiying and fixing bugs/issues can be improved together with the skill of analyzing the web application's performance

This is the final continuation of the dashboard application and there are some changes that needed to be done before working so make sure to follow all the steps in order to get everything working

1. Checkout the folder named "test" inside the project directory. This would be where we store all the files needed for the testing
2. Create sub-folders within the folder 'spec'. This will be the folder where the unit tests for all the components are stored. Now the directory for the project would look something like this:

```
dashboard/
| app/
| | same as before...
| dashboardServer/
| | same as before...
| test/
| | spec/
| | | controllers/
| | | | confirmBox.js
| | | | contact.js
| | | | login.js
| | | | overview.js
| | | | producer.js
| | | | root.js
```

```
| | | | work.js
| | | directives/
| | | | dialog.js
| | | services/
| | | | authenticate.js
| | | | getContact.js
| | | | getProfile.js
| | | | getUser.js
| | | | session.js
| | | app.js
```

1 Assignment 1: dashboard_app

The last step in developing the dashboard application would be to add in testing for every single components that has been created or existed in the app. This would include unit tests for elements like controllers, directives, filters, services, etc. The main focus is for the functionalities of the units so other testing methods and workflows like End-to-end tests can be omitted

- Requirements
  - Testing
    - Checkout the file named 'karma.conf.js' to get an idea of the settings for the unit test task
    - Write unit tests for your AngularJS application. Make sure to cover all the components that can be tested such as: controllers, directives, filters, services, providers, and factories, etc.
    - Once the Karma configuration file is created and the unit tests are available, testing task can be run by executing 'grunt test' in the app directory. This task will automatically detect the settings in the Karma config file and boost up the test results to the CLI screen
    - In order to perform the continous testing workflow, executing 'grunt watch' so that every time there are changes happened in the 'test' folder, the 'test' task will run automatically
  - AngularJS
    Add in functionalities for **"producer"** section of the dashboard application regarding the following requirements
    - This section will have custom functionalities that are developed completely by the developers. There are no restrictions on how this section should be created so be sure to get creative on this section

- Sample Demo
  Please check out the 'sample_demo' folder for more details