UI-101: Week 4 Assignments

This week assignments focus on 2 main topics that are covered during the training section including Git and AngularJS MVC framework. By completing the assignments, candidates would be able to work on applications that utilizing Git version control and to develop Single Page Applications using AngularJS framework

Before beginning the assignments, please complete the prerequisites:

1. Navigate to the following URL to prepare cloning the needed repository
   https://github.com/tachymetre?tab=repositories

2. Clone the pre-built ”dashboard” project
   - Using Sourcetree:

   Add the following URL to the ”New Repository” field
   https://github.com/tachymetre/{{repo_name}}.git

   - Using CLI:
   Type the following command into the CLI of your choice

   ```
   git clone https://github.com/tachymetre/{{repo_name}}.git
   ```

3. Navigate to the application folder to read up on steps (‘README.md‘) to build Dashboard application and start Dashboard server

1 Assignment 1: dashboard_app

Create the Dashboard application given the provided files in the project folder from the remote repository. The application will be a Single Page Application with multiple pages all mapped to one ‘index.html‘ file only. The application must be able to have multiple controllers for each single page, multiple services to retrieve appropriate APIs from the server, and a single directive to perform certain DOM manipulation. The application layout would also be completed using the known technology of HTML and CSS. This application will be completed as a group, not individual so be prepare to create branches for each individual as well as merge them later into one main branch as the final copy

- Folder structure

```
dashboard/
  |  app/
  |  |   assets/
  |  |   |   images/
  |  |   |   |   src/
  |  |   |   |   |   app/
  |  |   |   |   |   |   background.jpg
  |  |   |   |   |   |   favicon.ico
  |  |   |   |   |   |   logo.png
  |  |   |   |   |   profile/
  |  |   |   |   |   |   darth.png
  |  |   |   |   |   |   minh.png
  |  |   |   |   |   work/
  |  |   |   |   |   |   1.jpg
  |  |   |   |   |   |   2.jpg
  |  |   |   |   |   |   3.jpg
  |  |   |   |   |   |   4.jpg
  |  |   |   |   |   |   5.jpg
  |  |   |   |   |   |   6.jpg
  |  |   |   |   |   |   7.jpg
  |  |   |   |   |   |   8.jpg
  |  |   |   |   |   |   9.jpg
  |  |   |   |   |   |   10.jpg
  |  |   |   styles/
  |  |   scripts/
  |  |   |   controllers/
  |  |   |   directives/
  |  |   |   services/
  |  |   |   app.js
  |  |   templates/
  |  |   |   404.html
  |  |   favicon.ico
  |  |   index.html
  |  dashboardServer/
  |  |  bin/
  |  |  |  www
  |  |   routes/
  |  |   |   category.js
  |  |   |   index.js
  |  |   |   location.js
  |  |   |   login.js
|   |   |   profile.js
|   |   |   users.js
|   |   |   work.js
|   |   views/
|   |   |   error.jade
|   |   |   index.jade
|   |   |   layout.jade
|   |   app.js
|   |   package.json
|   .gitignore
|   bower.json
|   Gruntfile.js
|   package.json
|   README.md
|   robots.txt
```

All the file names mentioned in the folder structure will be provided upfront. Custom script and style file(s) must be placed in the appropriate folder within the structure

- Instructions
  - Branches
    - Each individual working on the application will have to create his/her own branch with the following syntax: ”dashboard@0.1.0-{{functionality}}”. For example, if a person is working on a ”login” functionality, the branch that would need to be created will be: ”dashboard@0.1.0-login”
    - The main branch that includes the final working application will be named: ”dashboard@0.1.0”. Please only use this branch to merge the complete implementation from other branches and to deliver the final product. Each developer is resposible for the changes he/she made to the main branch. When getting merged, this main branch should work correctly with the given codes and the reposibility to maintain this branch will be on the whole team
    - There is no limit on how many branches need to be created as long as the final branch ”dashboard@0.1.0” is working properly. Please do not work on the ”master” branch since the original state of the application when started will be preserved within this location
- Requirements
  - HTML
    - The document must be declared using HTML5 syntax
    - The document must have a title tag ‘Dashboard’ that describes the title for the web page

    - The document must have the external stylesheets included in the head section
    - The document must have the script file(s) included in the bottom of the document, before closing the body tag
    - For template document files, it would not need HTML5 declarations and proper document elements
  - CSS
    - Use external stylesheets for the application, not inside the document file. This applies to both vendor stylesheets as well as custom stylesheets
    - The application must be responsive on multiple devices of different widths. This can be done using Bootstrap and media querries
    - Be sure to provide comments to the stylesheet(s) if needed
    - The stylesheet should have appropriate prefixers so that when viewing on modern browsers (i.e Chrome, Firefox, Safari, IE 11+), the user interface should be the same
  - AngularJS
    - The application must have a working login page. When logging in the incorrect users, the login attempt would be unsuccessful. Otherwise, navigate to the work section
    - When logging in, the app must have the side bar navigation with links mapped to each section of dashboard app
    - The application must have a header that displays information about the company logo, username, date logged in and a ”Log Out” button that can sign out a login user
    - For the main view of the application, it would be done as subviews using the routing architecture to map the correct views to the correct paths
    - The first phase of the application would be to complete the ”work” section by displaying the items with 2 designs: list and card
    - The application must have filter functionality based on the user inputs for ”work” section
    - The application must have 2 buttons: ”Select design” to switch between different design and ”Add item” to display a dialog for users to add new item
    - Each item displayed in the app must have the title, author, number of likes and comments grabbed from the service APIs
    - For each of the item in the application, it must be accompanied with 2 links: ”Edit” and ”Delete”. These links would do the appropriate functionalities based on their names
    - Controllers must be used for each section and the single directive must be sed for the ”Add” and ”Edit” functionality. ”Delete” functionality would be taken care by controller functions

    - ∗ Routing architecture of the application must be taken care in the ”app.js” and each route must be mapped to the corresponding views they are associated with
    - ∗ If the file(s) has multiple functions within, be sure to include comments for descriptive purpose

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