Module 7

From CSE330 Wiki

This module introduces MEAN, a fullstack web framework that uses **M**ongoDB, **E**xpress, **A**ngularJS and **N**ode.js.
You have already learned (some things) about Node.js. In this module, you will be introduced to other three new technologies. Don't frown yet, the goal of this module is to expose you to the concept of web frameworks and prepare you for the creative project. You are not required to be an expert in all of these technologies, however, it's important to understand the basics of each and how they work together. Follow along the tutorials and complete exercises to learn how things work, and you will be ready to create your own web applications using the MEAN stack very soon!

Reading

The following articles on the online class wiki textbook contain information that will help you complete the assignment.

- Preparations
- Getting MEAN
- Git

Individual Assignments

Tutorials

Make sure you go through tutorials for both MongoDB and AngularJS, and understand how Express works.

Install MEAN

The individual and group assignments for this module only require AngularJS and completing the tutorial here (https://docs.angularjs.org/tutorial) . We provide installation instructions for the entire MEAN stack using MEAN.IO as a reference. You do not need to install the MEAN stack for this module (but it is recommended to really understand all of pieces involved).

To install MEAN on your machine, initialize an app and type grunt in your terminal to start the server (make sure your mongoDB server is connected and running). Open up the app in your browser (localhost:3000), and understand what the default app does.

Group Project

- Extended the AngularJS tutorial with additional features.

In the group portion of this module, you will extended the AngularJS tutorial with two additional features. The first is a side-by-side comparison page displaying two products. In the second, you will

provide four additional sorting methods for displaying devices on the main product page.

In the side-by-side comparison page, a user will be able to select two devices and view all of their features in a table. This table will consist of three columns. The first column simply lists all of the features of either device (battery, camera, connectivity, hardware, etc) . The second and third columns will display the information associated with that category for a particular pair of devices, allowing a user to easily compare the same feature across different devices.

The second feature is simply allowing users to sort the devices on additional metrics (in addition to Alphabetical and Newest). Select four different attributes that you want to include to the drop down list on the main page. For example, I may want to sort by by weight or size of battery.

For each feature, create a separate Git branch. Name the first branch SideBySide and the second branch AdditionalSorting. After you complete a feature, merge the branch back to the master branch.

Grading

We will be grading the following aspects of your work. There are 50 points total where 18/50 are extra credit points. To complete this module you must do either the Side-By-Side comparison feature or the Additional Sorting feature. You will receive the points according to the rubric below, and any points over 32/50 will be extra credit.

1. **Completed AngularJS Tutorial (6 Points)**
2. **Completed Side-By-Side comparison feature (18 points)**
    - Users can select two different devices and view all of their attributes in a table format.
    - If information for a particular category is not available on one device, display "Not available" in the table
3. **Completed Additional Sorting feature (8 points)**
    - Users can sort devices based on four different attributes and the attribute values are displayed on the main device list page. (2 points per attribute).
4. **Git Repo (8 points)**
    - Two branches are created, one for each feature (4 points).
    - Both branches are merged back to the master branch (4 points).
5. **Creative Portion (10 Points)**