## HW3: AJAX Web App

**SWE 432 Fall 2016**

**Due: 9/27 before the start of class**

In this HW assignment, you will take a step towards completing the implementation of your web app’s baseline functionality, using AJAX to persist data and interact with a web service. Extending what you’ve built so far, by the end of this HW assignment you should have implemented at least two (2) scenarios where a user can interact with your web app and it behaves correctly. The remaining HW assignments over the semester will build on this functionality and refer to these scenarios.

What do we mean by “scenario”? A scenario describes a specific task a user will perform with your web app and the behavior of your web app as the user executes this task. Each scenario may touch on one or more of your web app’s views or pages. Consider again the sample project description (shown in HW0):

Personalized News Visualization: FLEXibleNews. FLEXible news is a news aggregation and search application that will allow you to view a large number of different types of news articles in one go. Using FLEXible news you can view the latest happenings from around the world, filter them using a variety of different categories. After registration you can log in to the system, allowing it to customize search results. After logging in, FLEXible news shows users an overview of current news, visualized in a treemap. News is grouped by category, and then within each category, stories that are trending appear larger (see screenshot below, but note that screenshots are not expected for the proposal!). Users who are logged in can save their current view, or save searches.

Here are four scenarios that you might be give for this project:

1. Registration

A new user to FLEXible news may create a registration using an email address and password. The application checks to ensure that the email address is unique. After registering, users can log in using these credentials.

1. Browse news articles

Users may get an overview of the current news articles by browsing the homepage view of FLEXible news. News items are grouped by categories, such as World, Sports, Politics, and Entertainment, offering the user to a high-level overview of the types of news available. Using a D3.js tree map visualization, articles are depicted with individual headlines, grouped into the corresponding category. The importance of each headline is depicted through its size, scaled by the number of related stories. Users may toggle the visibility of categories.

1. View news articles

Users can view a news article by hovering over an article header. On hover, a popup is displayed that contains a short snippet of the article, the number of related articles, and a link to view the full article.

1. Search

Users can search for news articles by entering keywords in an input element at the top of the page. As users type, the news article category view updates, filtering the displayed categories and news articles based on the provided keywords.

**Step 1: Define four (4) baseline scenarios**

Working from your project proposal, write down four scenarios that capture the core functionality of your application. Exactly one (1) of these scenarios should use an information visualization that you will later implement in D3.js. You need not, necessarily, cover all of what you initially specified in your project proposal. But your scenarios should not be overly trivial and should reflect a meaningful baseline level of functionality for your web app. If you have questions about what this should be for your project, please contact the TAs or instructor.

**Step 2: Firebase**

Regardless of your specific project, for this step of the homework, you will connect your web app to [Firebase](https://www.firebase.com). Specifically, you must

1. Create and connect to a Firebase project for your group
2. Persist data to Firebase using (a) set and (b) push
3. Read data from Firebase

**Step 3: Web Service**

In this step, you will work to implement one or more of your scenarios by connecting your web app to at least one web service applicable to your project. For this step you should

1. Connect to a web service, either through AJAX or through a wrapper library
2. Retrieve data from the web service and perform computation using the data
3. Use the data retrieved and computed to update the information displayed to the user

**Step 4: Implement two (2) scenarios**

Building on all the work you’ve done in the previous HW assignments and in the previous steps, you should finish implementing two (2) of the four (4) scenarios you defined above. You should **not** implement the scenario requiring the use of an information visualization built in D3.js. In grading your HW assignment, we will test your application to ensure that it functions correctly for these three user scenarios.

**Submission Instructions**

1. Create (or update) your readme file in the Github repo to include descriptions of each of your four (4) scenarios. Separately list the two (2) scenarios that are currently implemented.
2. Include the Submit your changes to the same Github repository that you and your partner used in past HWs. Create a HW3 pull request from gh-pages into hw-submissions. Merge the pull request. Create a HW3 release from gh-pages.