

Good job. There are still some writing and presentation issues, but all-in-all a good effort.

G 4
F 8 (A)
C 5 (B)
U 4
I 3 (C)
S 4
A 4
P 6

38

Red, White, and Youth

Dalton James
Jared Perreault
Kyle White

UMass Lowell
91.462 – GUI Programming II
February 10, 2015

- (A) some things in this section were actually found in your components section.
- (B) likewise, some things that should go here only appear in your list of issues
- (C) I think there are more issues related to the front-end, too

Contents

Goal Statement	2
Target Audience	2
Feature Descriptions.....	2
Component Details	3
Website Mock-up	4
Home Page	4
Register Account Page	5
Topics Page	6
View Topic Page	7
Create New Topic.....	8
Discussion of Issues	9
Schedule	10
Sources	11

Goal Statement

Our mission is to provide a means of political discussions for the young adults of America. Using a clean user interface, we hope to facilitate these debates by giving users a bias-free environment to voice their opinions and to reflect on the opinions of others.

Target Audience

Red, White, and Youth is an organization that caters to young adults. It is no secret that voters between the ages of 18-24 have never consistently been a large part of elections.

In fact, based on statistics taken from the U.S. Census Bureau, young voters are voting even less than in the past. "Since 1964, young voters between the ages of 18 through 24 have consistently voted at lower rates than all other age groups".¹ In 1964, 50.9 percent of eligible voters in this age group voted in elections. By 2012, only 38.0 percent of voters were part of these same elections. These very statistics are evidence of issues that Red, White, and Youth hope to change.

There is a huge untapped market that candidates are neglecting. Red, White, and Youth is determined to change that. We offer an interactive and modern way to talk about controversial topics to get young voters involved in the discussion. Not only does our site offer a place for voters to debate, it offers a place for candidates to see what users between the ages of 18-24 actually care about. With this information, candidates can finally tap into a market that is currently underserved in politics. Red, White, and Youth will change elections forever by pioneering a place that will get young voters motivated about politics.

Feature Descriptions

Our website will feature a user account system. Users will have to register with an email and username before they are able to use the site. The website will have a topics page

The footnote that goes with this reference seems to have been deleted.

that will be a list of user-created discussions. This page allows users to see current topics of discussion and create new topics of their own.

Each topic will consist of

- a title
- description
- all user comments
- the user's information of each comment
- topic tags
- number of views
- the time each post or comment was made

Time permitting; we also want to add more account features such as private messages

between users, a user profile page, and an alert system to notify users that a topic they are interested in has had comments added.

Component Details

Our website will have five general pages. One will be the basic home page and the other four will have heavy server communications. We plan to use a MySQL database to store all user account information and user-created topic data. This database will be accessed by PHP scripts¹, which will relay all the desired data to the client-side JavaScript code. We plan to have two tables, one for account information and the other will be a topics reservoir. We plan to use the AngularJS and jQuery JavaScript frameworks on all of our HTML pages. AngularJS will be used for all dynamically displayed content² such as the topic data from the

¹ <http://php.net/manual/en/ref.mysql.php>

² <http://stackoverflow.com/questions/22536477/rendering-dynamic-htmlangularjs-content-content-after-ajax-call-in-angularjs>

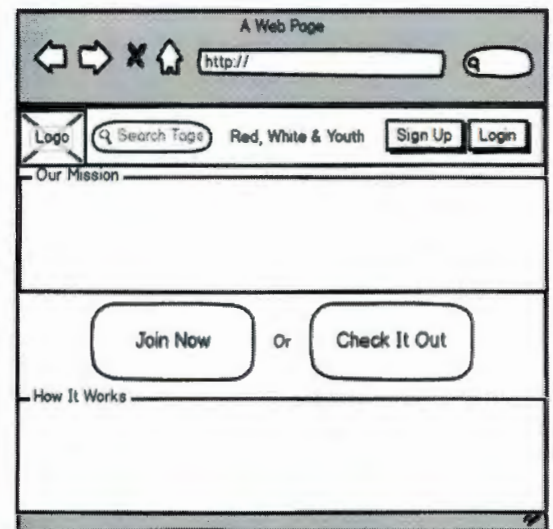
MySQL database. We will use jQuery as a framework for the form submission utilities³. We will also be using Bootstrap 3 as a CSS framework, due to its ease of use and excellent documentation.

Website Mock-up

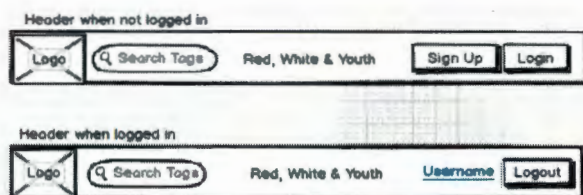
Our website will require a large amount of database communication via PHP scripting, but we feel that all the scripts will be fairly similar. Each script will either retrieve data from the database or add new content to it. Developing the PHP scripts will be our most difficult task. However, due to the similarities in the scripts we feel that after we complete the first one the rest will be much easier and faster. Other than the PHP scripts, the website will mostly be a large HTML skeleton that will be filled in by AngularJS and jQuery.

Home Page

This is the first page the user will see when going to our URL. *(see Figure 1).* Much like GitHub's home page, this page will feature a large advertisement for our site encouraging users to create accounts. It will also include our mission statement and a short description of how the website works. This page is very simple and will be using only HTML, CSS and jQuery. The



(Figure 1) An Example of the Home Page



(Figure 2) An Example of a Header

header of this page will remain the same for all the other pages on our site but the Sign-Up and Login buttons will change whether or not the user is logged into his or her account as shown

in Figure 2.

³ <http://api.jquery.com/category/forms/>

Refer to figures by number in your text.

Page Features

- Join Now & Sign Up buttons will both be <href>s which will direct users to the Register Account Page.
- The Login button will trigger Bootstrap Modal and prompt users to log in. If successful it will then navigate the users to the topics page.
- **Time Permitting - A “Check It Out” button** that will navigate users to the topics page. They would not be logged in, so they will be able to view the site, but posting topics and comments would be prohibited.

Acceptability Criteria

This page will need to be completed with everything except the “Check It Out” button. ✓

Register Account Page

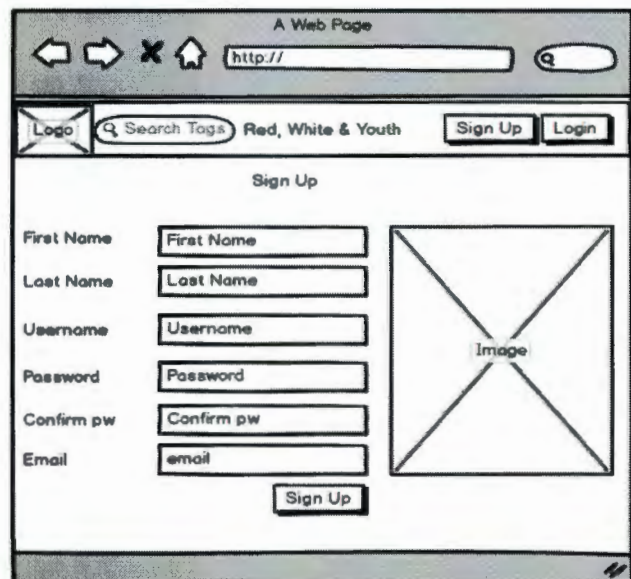
This page will have a simple form

which the user will fill out to create an account. *(see Figure 3).*

We plan to add some patriotic graphics to the page to help cultivate a sense of American pride within the user.

This page will be built with HTML/CSS for formatting and styling. jQuery will be used for its form utilities to easily create a

serialized JSON object to pass to the PHP script.



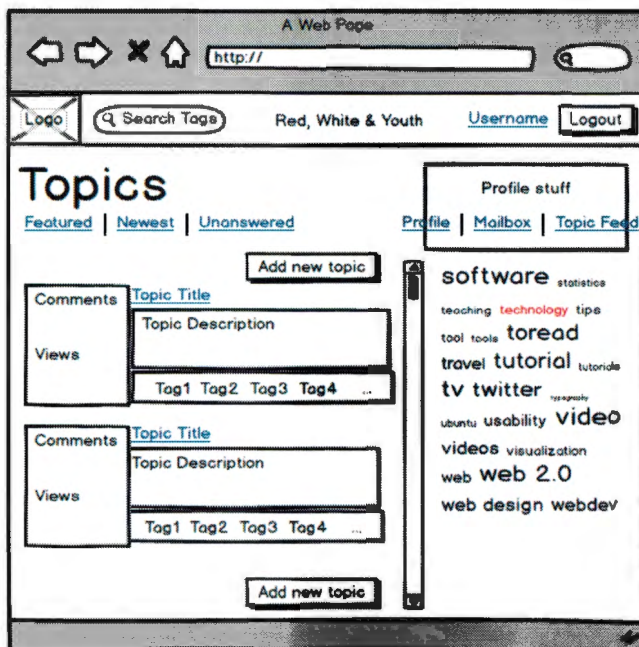
(Figure 3) An Example of a Register Account Page

Page Features

- A jQuery form for users to enter their information to build their profiles.
- The Sign Up button will submit the form content and trigger an AJAX call to a PHP script as explained above. The user will then automatically be redirected to the topics page.

Acceptability Criteria

This page will need to be completed in its entirety. ✓



Topics Page

This is the primary page of our site. From this page users can view all current discussion topics, read a short description of each topic, and view their profile. Only a limited amount of each topic's content will be displayed so the user will have an idea of what the discussion is about. Users will have a few options for sorting topics. They can

(Figure 4) An example Topics page

sort the topics list by selecting an ordering such as featured or newest. They can also use a tag cloud (depicted on the right of figure 4) to see what topics are the most frequently discussed. All topic sorting features will be created using the tags and jQuery filtering. PHP will be used to retrieve the content from the topic reservoir and AngularJS will be used to dynamically display the list of topics.

Page Features

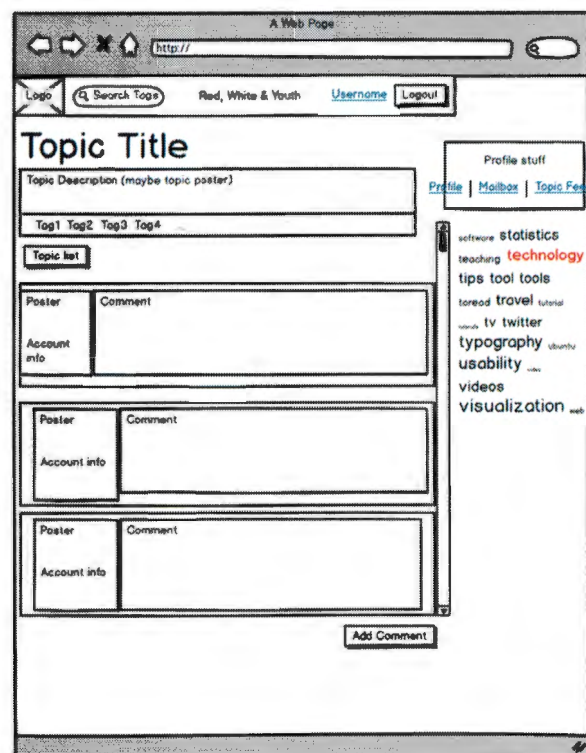
- Dynamically created **topics** list that is populated from a PHP script's retrieval of data from the topics reservoir by AngularJS.
- Each topic will be clickable to allow the user to navigate to the topic's specific View Page.
- Another button will be Create New Topic, which will navigate users to the Create a New Topic page.
- **Time Permitting** - List sorting and tag cloud will be implemented.
- **Time Permitting** - Developing an interactive profile, messaging, and notification system.

Acceptability Criteria

Only the dynamic listing of topics, Create New Topic button, and the abilities to navigate to the View Topic page for any specific topic will be needed. All other features are extra. ✓

View Topic Page

This page will be for viewing the comments on each topic. *(see figure 5).* It will include a short description of the current topic, the tags associated with it, each comment that has been posted and a button for users to add their own comments. Each comment will be stored in a database and a PHP script will retrieve them as the page loads and display them in chronological order. Along



(Figure 5) An example View Topic Page

with the comment, some information about it will also be displayed such as the username of the poster and the time of the post.

Page Features

- Dynamically filled topic information that is generated using PHP scripts.
- Two buttons that **essentially** function as <href>s. One will return users to the Topic list and the other will allow users to add comments to the thread.

Acceptability Criteria

This page will need to be completed in its entirety. ✓

Create New Topic

This page allows users to create a new topic to be discussed. There will be a form for the user to fill out that includes the topic title, a short description of the topic, the tags associated with it, and the user's first comment. It will also have a Post button to submit the form and have the topic added to the list of topics.

Page Features

- Form to be filled out.
- Post button that **sends information** to the database.
- **Time permitting** - A spell check button that will check the form for spelling errors prior to the user submitting the topic and allowing them to fix any mistakes.

(Figure 6) An Example of a Create New Topic Page

Acceptability Criteria

The form and the post button must be functioning, but the spell check does not have to exist. ✓

Discussion of Issues

One potential issue we might have is we are all new to PHP scripting. Our project requires a large amount of **server-side** scripting to talk to the MySQL database. Therefore, if we are unable to write the **necessary** scripts, we won't be able to finish our project. Our solution to this issue would be to use NodeJS⁴ as our backend framework⁵. This would allow us to use JavaScript for our **server-side** scripting, a language that we all know very well.

Another issue would be difficulties for us to meet together. As we have seen, this winter the weather is unpredictable. If we have more heavy snow it will make collaboration on the project increasingly difficult. To work around this, we will have specific parts of the website that each of us will complete so we can do work separately. After, we will all use a group GitHub account to upload the different parts to ensure they can work together. We will also employ the use of an **instant** messenger that we all check on a regular basis.

⁴ <http://stackoverflow.com/questions/5818312/mysql-with-node-js>

⁵ <http://www.codediesel.com/nodejs/querying-mysql-with-node-js/>

Schedule

Week Of	Task	Assigned To	Phase
2/1	Proposal due on 2/10	All	Alpha
	Research MySQL to allow work to begin on database implementation	All	
2/7	Implement Username and Password database -Talks to Sign Up/Sign In pages	Dalton	
	Implement Topics Database -Refers to Create topic and View Topics pages	Jared Kyle	
2/15	Create skeletons for all site pages	All	
	Design UI and implement functionality for Create Topic page -JavaScript code and CSS	Jared Kyle	
2/22	Design UI and implement functionality for Sign in and Sign up pages -JavaScript code and CSS	Dalton	
3/1	Alpha phase presentations	All	
3/15	Communication between databases and site pages -PHP scripts -Ajax requests	All	Beta
3/22	Resolve all residual bugs and code optimization	All	
4/12	Usability Testing -Testing followed by a Test Report	All	
4/26	Final Presentations	All	Release

Much better than in draft.

Sources

jQuery: A JavaScript framework that better fits the Document Object Model than traditional JavaScript.

<http://api.jquery.com/>

AngularJS: A JavaScript **framework** that allows for great flexibility for creating dynamic content.

<https://angularjs.org/>

Bootstrap 3: A flexible, well-documented, and easy-to-use CCS framework.

<http://getbootstrap.com/>

PHP: A server-side scripting language that has excellent MySQL query support.

<http://php.net/>

MySQL: powerful, well-documented open source database

<http://www.mysql.com/>

