Dillon Beliveau

Elliot DeMatteis

Will Barnwell

CS142 – Advanced Web Design: CSS and Usability

Project Summary

When designing this site, we opted for a design of extreme simplicity that followed the design guidelines set out by Twitter Bootstrap. Each page was to have one very specific purpose and nothing more. All pages were to remain extremely clean and free of unnecessary clutter.

This project was a continuation of a CS 008 final. Since then, we have grown as programmers, and also as the base of the site was already completed, we were free to add on a large amount of features and usability improvements that we had intended to add initially, but simply ran out of time. These changes alone turned into a complete project.

Friend requests were invisible by default. When someone added you as a friend, you would not know it until you went to the “Friends” page and looked at your pending requests. We wrote some JavaScript to query the server to see how many friend requests the user had pending. This information was displayed in a badge in the main site navigation. The ability to delete friends was also added.

Form validation and display was improved throughout the site. In the original project, we didn’t have time to make our forms look good or make sense without explanation. A lot of time was spent carefully aligning and reordering text boxes and adding carefully worded labels to form elements.

It wasn’t possible to do a lot with events after they were created in the original site. We added a wide range of functionality to them. This consisted of the ability to quickly edit, delete and share the events. This was accessible by a standard hamburger menu that was placed on the event (only visible when the owner of the schedule was logged in.)

A frustrating thing about the original site was that once an account was created, you would still have to log in to access the site. Code was written to log the user in automatically after registration, saving precious seconds and providing a good first impression. Additionally, when the user logs out, they are automatically redirected to the main page of the site.

The interface of the schedule view was almost completely altered. Control options were moved to logical places and styled to stand out. Previously, jumping to dates far away required the user to click the “Next week” button many, many times. A small form was added allowing the user to jump instantly to any date. Also previously, whenever a user performed a function on the schedule which took them off and then returned them to the schedule view, they would always find themselves at the current week. Django’s caching system was leveraged to save what time the user was on, and to return them to this time when their function was completed. Indicators were added to allow users to easily see what time was in the future or the past. Past days are shaded in gray, and future ones remain white. The current time is shown as a red horizontal line.

When creating an event, users seemed to have trouble entering times in the correct format. JavaScript was written to translate time written in any format to correct 24 hour time.