Touché

Senior Project: George Harwood

Introduction

This project has been developed by a few other groups over the years. It was originally written in the late 90's and was last updated in 2005. It was written using PHP, HTML and SQL indicating a web application. At its core the software functions as an application that can run programming contests. During a competition, teams are given problems to work on and try to solve them by writing code. They would then submit their solutions to the problems and a person of authority (or a judge) would then mark there solution as accepted or with a small description of what was wrong and the team would then be able to see that feed back and try again. This software has been used by Taylor and other schools for many years, but there were still some major improvements that we had to make. Quick Fact: The term Touché is used in fencing and means "hit."

Approach

We each had different roles when we were working on Touché. I took the role of Front End Developer/Web Designer. That being the case, I took a strategic approach when I began working on the software. I started off by doing some wireframe and prototype designs for how the software could be laid out. Keep in mind that we had already decided at this point that Touché needed to be completely redesigned. So knowing this, I estimated I would have to touch up over fifty files and strip out a lot of old styling and elements. I decided to work with Twitter Bootstrap and skip a lot of work by having predefined classes. This helped save a lot of time and energy. In the software there are three user interfaces. There is a Admin, Judge and Team interface. I tackled one at a time and was able to complete them all.

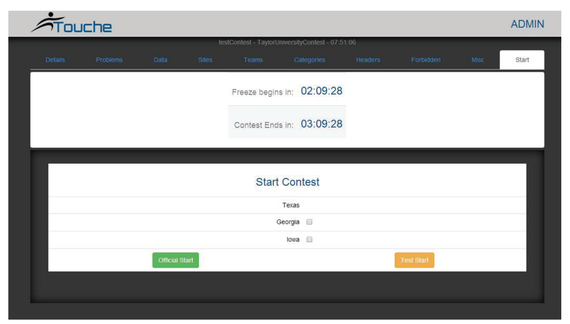
Branding

I wanted to Brand the software. So I used Adobe Illustrator and Adobe Photoshop and came up with the logo. I also decided on a color scheme for the Touché brand. Beyond the actual web interface there wasn't much else that needed to be redesigned. I believe that if everything is consistent and looks good that users will have a better user experience.

Touché

UI Improvements

When we were first shown the software I was pretty horrified at the user experience. I completely redid the navigation between pages. Previously there were no visual cues indicating which page the user was on with the exception of the URL. The clock felt like it needed to be highlighted more within the software. In other competitions like chess, (and basically every other sport invented) the clock is always large and easy to see. So I decided to make it bigger and easier to read.



Challenges

This software was originally built using tables and so I literally found tables within tables within tables. It was hard making sense of it all. Another challenge was working with old school HTML and absolutely no CSS. There were <font> elements and elements. I ended up stripping all of that out and replacing it with CSS. Another thing was the inconsistency of the code. I had a hard time figuring out how it worked from page to page. Sometimes there would be 50 or 60 lines of html all within PHP echo statements. Thankfully the more I worked with it the quicker I was able to work through and figure out how everything came together. Error Handling was inconsistent and all over the place which made styling hard. Sometimes there were no errors written in at all when there should have been. Since the errors were spread all throughout the DOM I decided to wrap them in all in a fixed div, so that they all would be consistent.

Other Tools

Some of the other tools that helped me along the way were Adobe Kuler, Font Squirrel, Sublime Text 2, WinSCP, Putty, the Adobe Creative Suite, Twitter Bootstrap, and Pivotal Tracker.

FONT SQUIRREL
100% Free For Commercial Use.

PivotalTracker

Future Improvements

In terms of User Interface the software still needs a lot of work. I only just began to scratch the surface of what still needs to be accomplished with Touché. I would have liked to do a lot of more animation and work more on making the software more fluid. Besides UI related things there are still two fairly high priority things that we were not able to achieve. The first was that we wanted to allow team's access to the compiler. With the existing system the teams get counted off if their code does not compile. Obviously they would not submit code that they did not think would compile. It does not seem fair to penalize them for differences between their compiler and the contest compiler. The other big thing would have been adding python support for teams.

Conclusions

Overall I got a lot accomplished throughout this project. I edited over fifty files and changed over 5000 lines of code. I was able to completely redesign Touché as well as start the process of branding the software. I made user interface improvements to all areas of the product including the clock, navigation, and forms. I believe that the team I was in was a perfect fit for the job. All of our skills built off of each other and we were able to produce a sleek finished product that I hope others will be able to use and enjoy without a hitch.

Special Thanks to Daniel Sanders, Alexander Wagner, Caleb Stevenson, Tyler Garcia, and Matt Goldsberry for all their hard work!