

TOUCHE: Coding Competition Software

Caleb Stevenson

Department of Computer Science and Engineering, Taylor University, Upland, Indiana 46989

Introduction

The main two parts that I worked on for this project were the continuation of the multi – site functionality and the creation of a test team for a test start. There have been potential users of Touche that have wanted to be able to host two or more contests simultaneously. With the current version of Touche at the time, that was not possible. Some code was quickly thrown together to try and accommodate the multi – site functionality, but ultimately ended up failing. Being able to work off of this code was very helpful to complete the multi – site functionality.

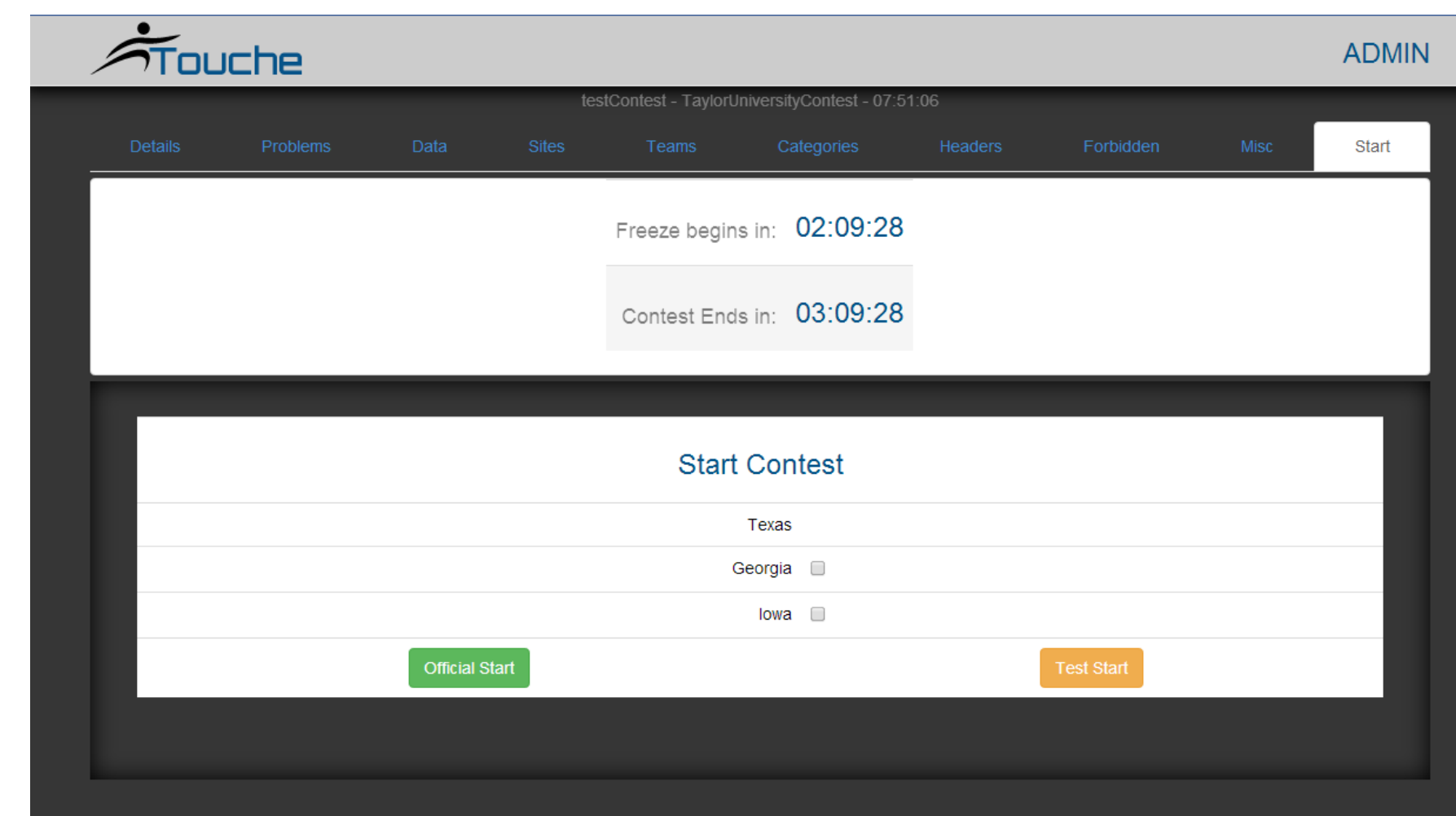
The whole point of having a test start is that admins may want to run a fake contest to make sure everything is in order before running the real thing. But if an actual team logged in during this time, they could see the problems, creating a security leak. Thus we wanted to have test teams that could log in to the test start to test the functionality, but not have real teams log in during this time. There was nothing created to support the idea of a test team, however, so I had to come up with the code for that entirely on my own.

Approach for Multi-Site

The multi – site code was mostly supported, so I quickly learned what was already running and then troubleshooted what was broken. The main problem that I realized early on was that there was a Contest_Config database table that contained all of the highest level data about the overall contest, but at the same time, contained the start time of the contest that was slightly different from the start time of the contest running on the Site database table. I determined that removing the Contest_Config table and put the most important information that was in Contest_Config into the Site table to copy into each individual Site that runs would be the best option. This would involve replacing all of the instances of Contest_Config in the code and replacing it with this master site information, and then changing the old Contest_Config start time information with the correct information from the specific site running the particular contest.

Approach for Test Start

The most basic approach that I could think to take was to simply add another boolean variable to the team table that could be set during the creation of the team to signify if the team was a test team or not and add another state to the Site table’s Has_Started for the creation of a test start to the contest. Then create a separate start that would only allow test teams into the contest in the start page for the admin. Finally I would have to change the log in page itself for the teams to check if the contest running was a test contest and if the team trying to log in was a test team.



Development for Multi-Site

First thing I did was look in the database creation page to change over all of the important data bits inside the Contest_Config table that I now wanted to bring over to the Site table and then transferred those over. Then I located all of the places where Contest_Config was in the file structure. This led to 20 or so hits, mostly for SQL queries. For each one of these, I had to create a new SQL query that would pull the correct information off of the new master site row that I created. Having this information be inside the Site table meant that I now had to hide this master site from the users who would be looking for a site to connect to a team or which site they wanted to start a contest for. By adding a simple if statement inside of those while loops where it gathers a list of all of the sites, I was able to hide this master site from the normal user.

Development for Test Start

I started off by modifying the database to add in the test team boolean value for any team. The only other place that the test team functionality would come into play was the team creation page and the team login page, which made the manhunt for adding in variables much easier than the multi – site. I then added a checkbox in the team creation page to set the variable. Next I added another checkbox that is not selectable next to created teams to show if they are a test team or not. Moving on to the Start page, I created another button for the test start and made it so that site name is re-checkable if on a test start, but permanently checked if on a real start. If there was a test start happening on a site, it would print “Running Test” next to the site to signify that something was happening in that site. I made the new button set the value of Has_Started to 2, where previously it would only be 1 or 0 if a contest had started or not. Finally I moved over the team login, where I changed the session.inc file to check for a test contest, and if it was a test contest, block the real teams from logging in.

Challenges for Multi-Site

Having to go through all of the Contest_Config and figure out exactly what each one was called and for what purpose was quite annoying. I got the call wrong several times trying to figure out the changes in the SQL query structure that I needed to make as I needed to get the data where the Site_Id was 1 instead of looking at the entire table like it was previously implemented with Contest_Config. I also ran out of time during the project while working on this feature, so I could not completely change the start time functionality and clean up the new Site database to only include fields that it absolutely needed. I also was not able to change the parts of the code that references the master start time instead of the individual site’s start time. This is the main reason that the multi – site functionality is not operational currently. There are several other bugs such as time zone differences that might have to be worked out in the future with regards to multi – site interface that I was not able to work on either.

Challenges for Test Start

One of the problems with the way that test teams is currently implemented is that a team is used to always being able to log in, but if there wasn’t a contest running they couldn’t see the problems. Now it looks like they have the wrong username and password if they try to log in during a test start. It would have been nice to keep the same look for the team, but that would involve creating a fake page for the team, which I didn’t have time to create and wouldn’t have been worth the time. Having the teams log in credentials actually take place in the include file instead of the index.php file was quite annoying as I could not form my query directly in the include file and instead had to form the query in the php file and then pass the necessary variables into the include file with \$_SESSION. Creating the SQL query for the test team was quite annoying as it was a very long concatenated string that had way too many parenthesis to make it readable, yet had to be that way for the SQL query to work. Debugging that string took quite a while.

Conclusion

Creating the test start feature was a big target issue in the list of bugs that we started out with. Implementing multi – site was something more off in the future, but I believe that both of these features add value to Touche overall. It would have been very nice to finish off the multi – site feature, but I believe that I made the process much easier for whoever comes around the next time to finish that feature easily. Messing around with the SQL in the database and changing where things were located was a new experience for me and I was quite scared at first with messing with working code like that in order to make it better. This project overall has really opened me up to the idea of taking a back end web development job as this project was quite enjoyable and web development is not nearly as scary as I initially thought it would be.



For More Information contact Dr. Geisler.