DebAPP:

The first debating app EVER

Caitlin, Sabrina, and Shannon

OVERALL VISION

Who: Voluntary judges (oblivious parents)

What: This application will replace the out of date system of judging that is used in today's New York State debate process. Our application will imitate said paper ballot while sending the inputted information to a database that will be called to when ranking the teams at the end of the tournament

Why: At the current time, judges have to use paper to record debate scores during rounds. These scores are then entered into the computer by a middleman. It takes a very long time for this middleman to calculate the scores and there is a lot of room for error during the process. Also, it can be difficult for debaters to read handwritten comments. We believe that a process that will allow judges to enter scores automatically and type comments will make the process faster, more accurate, and easy to read.

PROPOSAL

A. Problem / Task

Today's debate judging process is not only extremely outdated but also incredibly slow. Debate tournaments take several hours due to the amount of teams that attend and for up to two unnecessary hours just for the *paper* judging forms to be counted and the teams to be ranked is brutal. It wastes the time of all the students attending and the time of the voluntary judges.

B. Solution

We're making a mobile web-based app to use as a judging form that will add scores and wins to a central database. This will improve efficiency, make the whole debate process more environmentally clean, and save money by cutting printing costs.

C. Implementation

We are planning to code in HTML, CSS, JQuery (JavaScript)

We are going to code in these languages because they work best on a mobile platform.

Sabrina will be doing the User Interface using html and css

Caitlin and Shannon will be working together on the database and adding up the scores and wins to add to said database using JavaScript.

D. Monetization

We plan on giving the New York State Forensics League access to our software for a fee (flat rate).

It will also save the schools hosting the debate, printing costs. We would also like to make future adapted versions for other debate leagues, such as the Connecticut Debate Association, in order to make more money.

Use Cases:

To Score a Debate Round-

1. The judge will get onto the homepage which will be a mostly blank page with just the word “debAPP” and a textbox labeled “your debate code:” where the judge will type in the password they were given. The password will be the same for each judge in one debate, but differ depending on the different debates in the state. Currently before each debate the hosts have a judges meeting where they go over rules and regulations and that’s where the parents can get the password and told how important it is not to tell their children.

2. If they type in the correct debate code it will take them to the judging ballet.

3. There they will enter in the team codes, choose what round they are judging from the drop menu, fill in text boxes with each speaker’s name, add comments to each team’s performances, and pick a winner from a different drop down menu.

4. Once the judge hits the submit all the information typed in will be sent to the database. Nobody will have access to the database- it will only be affected by what the judge sends in as their round scores. This will make it impossible for people to hack and change their scores.

Risks:

1. The results might not all add into the database correctly, so that might take us a while.

2.When we want to print out overall stats, it might print out in a hard-to-read format. We will have to work with the UI for the display sheets as well as the score sheets so that each aspect is directed to a predetermined space. As this isn’t as important as having the data display in general, we will have to save this until we get everything else together.

3.It may be difficult to connect everything (UI to backend to database, etc). To deal with this problem we can make a really simple UI, backend, and database to begin with and we’ll just connect those three simple parts. As we continue to code we’ll make those three parts more and more intricate and they’ll already be connected!

4.It could take a long time to figure out how to make certain parts of our site password-only accessed, as well as figure out where these passwords will be generated. This feature isn’t a necessary attribute of our app, it’s just