CEP Final Project

Introduction:

Project Name: sCORE!

Project Team: Bryan Yang, Wang Jianzhi

Statement of the Problem:

A scoreboard is an important and vital component in any competition, friendly or competitive, and thus it plays an essential role in promoting the spirit of competitiveness which urges the contestants on during the course of their competition. Such an element should not be overlooked and it suffices to have a beautiful and multi-featured scoreboard.

However, we feel that most available scoreboards are not able to provide such features. Many scoreboards, either online or offline, have some of the following flaws in them:

- They cannot sort easily according to the formula that the users want. Most online scoreboards only perform sorting based on the total score which is quite inflexible if the user wants another form of sorting method.
- Some of them do not have very nice interface and seems dull to users. Examples of these are Microsoft Excel and Google Forms. They can be used as scoreboards, but they do not have very nice interface which add interest to the competition and the users.
- Almost all the scoreboards online are not public. Some programming websites which host programming competitions (such as Google Code Jam, IOI scoreboards) are specific to the programming competition and the scoreboard cannot be used by other users of public to design their own competition.

Objectives:

The goal of our project is to design a **public** scoreboard which has both a **nice interface** and an **efficient system of sorting**. This means that the application is able to be used by any online user and is able to sort the ranking of contestants with a variety of choices that the user can select.

Most scoreboard encompasses at least one of the above flaws. Having a flaw in such a vital element in a competition is heavily detrimental to the atmosphere of the venue and hence we decide to build a better scoreboard which covers all of the points above. Our product as described above is able to solve all the issues brought out.

Product

Technical Approach:

To effectively solve the problem, we will build a web application in Django. To make the explanations clear, we will define **'user'** as the administrator of the contest and **'contestants'** as the people participating in the contest.

Firstly, the user will have to register an account under our website. After he has registered for an account, he is then able to 'create a scoreboard'. Then, he will be given the choices of scoreboards, such as the Boolean scoreboard or number scoreboard.

After his scoreboard has been set up, the user can add contestants as models and add in their scores. He can also update, edit or delete contestants using their views respectively. When the data has all been tabulated, they will be sorted by total by default; however, the user can then sort the data and obtain the ranking for any sorting method he wants. We will use the Excel's method for ranking due to the vast complexity of Excel's formula system. For interface, we will use green, yellow, orange and red tone to colour the grids based on what the percentage of the score is.

External Scan:

We see a few applications which are similar to our application in terms of idea:

- Microsoft Excel, Google Sheet
  We feel that although their sorting function is strong with different formula available for sorting, their sorting is extremely limited if the user would like to change his sorting method in the middle of the contest, which our application can do in an instant. Also, their interface does not give the contestant an urge to excel in their competition.
- OJ Scoreboard, Code Jam Scoreboard, IOI Scoreboard
  Their interface is extremely pleasant with colouring tones. However, their objective for scoring is quite limited and does not allow external users to use the platform to make a scoreboard for their own event.

Looking at our application, it certainly does have quite an advantage over Excel, Google Sheet and the other contest scoreboards.

Feature Listing:

Our application provides **a beautiful interface**, **public scoreboard** as well as a **variety of ranking methods**.

Some of ranking methods we thought up of are:

- Total score (default)
- Time
- Problems (selective problems)
- Days (for example IOI day 1, IOI day 2)
- Ranging*
- Closest number*
- Specific queries*
- Weighted sum*
- Weighted average*
- Polynomial P(x)*
- Special formula indicated by user*

* indicates ranking methods which are new

We may also think of making this scoreboard a library like crispy-forms so that other websites which wish to use our scoreboard inside their website. We hope that this will make a huge impact in promoting the competitiveness spirit of the world :).

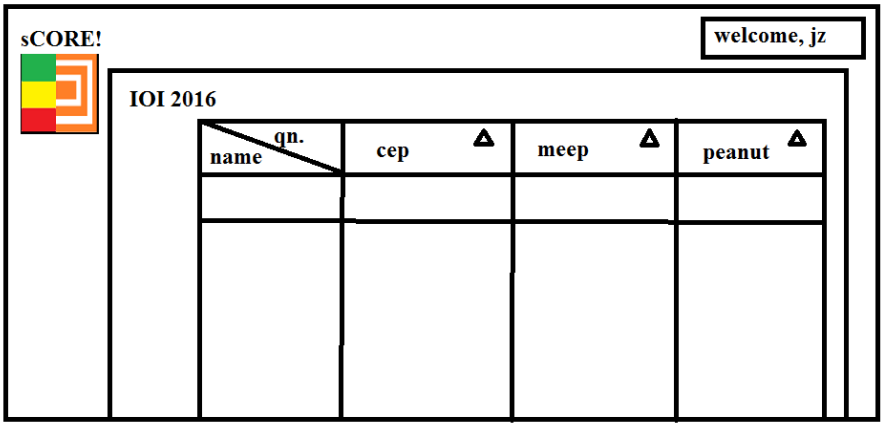
Application

Use Case Scenarios:

Class Rankings --- If a teacher would like to have a ranking of his/her students inside the class (can be public ranking or non-public), then he/she is able to generate the scoreboard which he is able to update or edit as long as he/she remembers the password. The application is also applicable to schools where each assignment has a weight (a percentage of the whole year's assignment) in which the teacher can use the weighted sum to sort the students.

Inter-house Maths/Science Quiz --- this annual event happens every year in the school and many executive committee members of these clubs spend a significant amount of time to sort out the scoreboard problems. With this application, all the competitive quizzes are able to be tallied quickly which saves the members time to plan other activities.

Design Mockups:



Measuring Success:

Our definition of success for our application is when our application improves the competitive spirit of the world when more and more contest organisers use our website/library to make their scoreboard.