Start Time _________

End Time _________

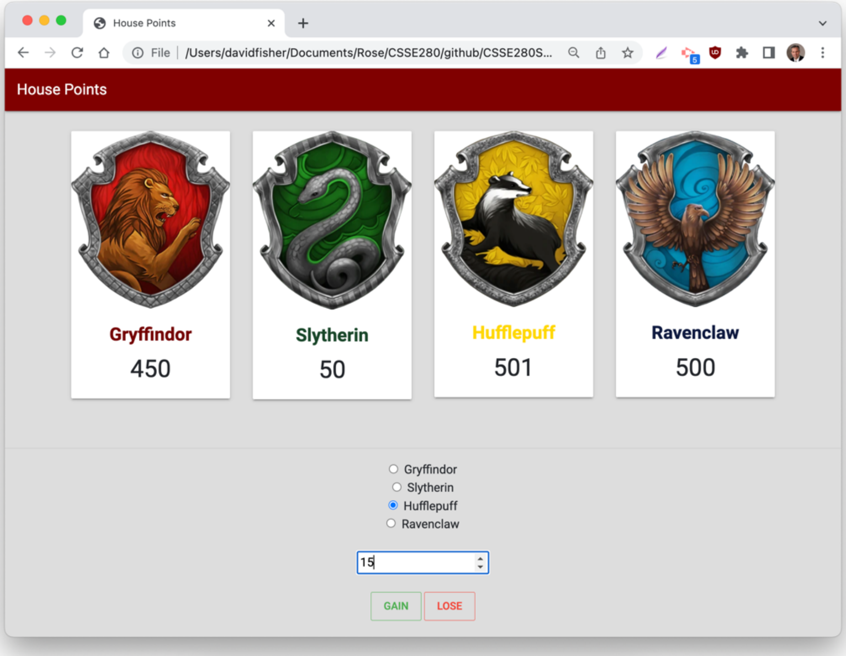
CSSE280 Summer 2022 Exam 1

You may use your computer, any prior notes or programs, the Moodle page, and the internet for general searching on web development. However, ***you must not communicate with anyone*** except your instructors and their assistants, if any. In particular:

- You must not talk with anyone else or exchange information with them during this exam.

	Points
Layout	**___ / 20**
CSS	**___ / 20**
JavaScript	**___ / 60**
Total	**___ / 100**

House Points web app:

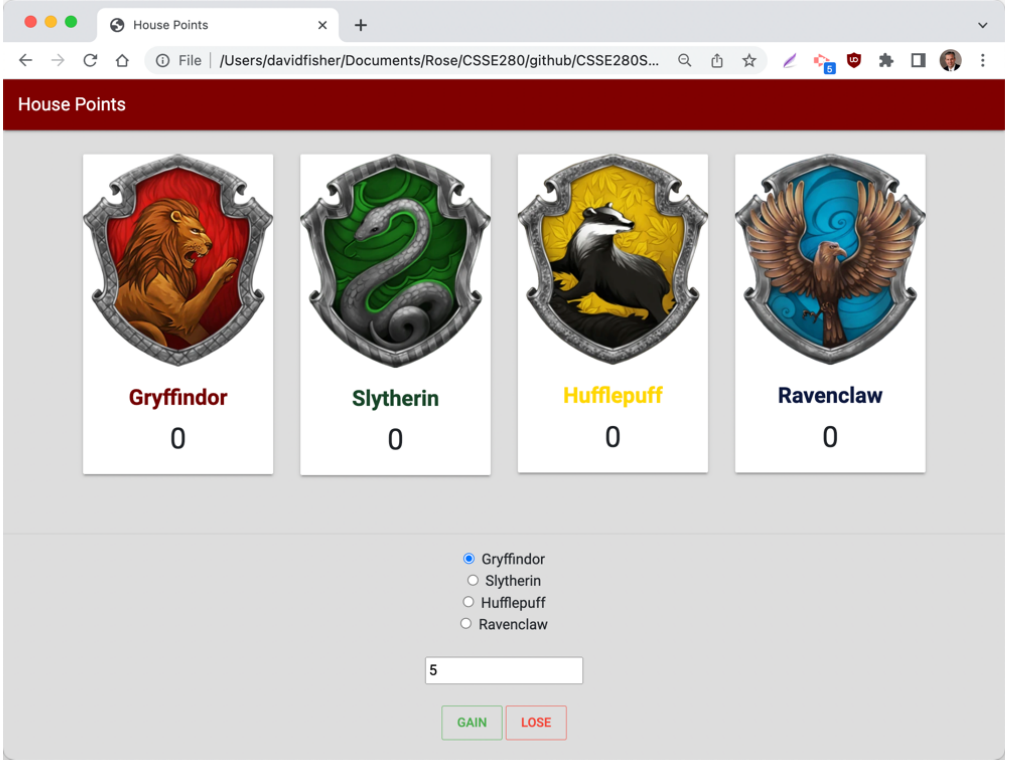


○ Gryffindor
○ Slytherin
● Hufflepuff
○ Ravenclaw

15

Hogwarts House Points

In this coding challenge you will make a House Points leaderboard for the school of Hogwarts. If you happen to know Harry Potter, then great, but if not, no big deal. There are 4 houses: Gryffindor, Slytherin, Hufflepuff, and Ravenclaw, each with a house crest (the four images). Every year Hogwarts has a House Champion based on which house gets the most points. Students can gain or lose points for their house, however it's impossible to go negative (wizards are bad at math). Your job is to make a webpage that starts at 0 points for all houses:



● Gryffindor
○ Slytherin
○ Hufflepuff
○ Ravenclaw

5

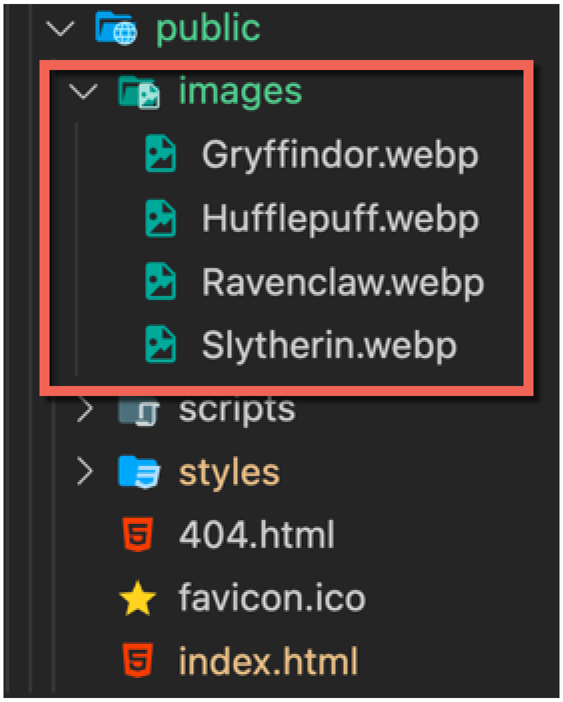
Then has a mechanism to Gain and Lose points for a house. You might notice that the display area with the four panels on top is very pretty, then a thin horizontal line (it's hard to see but it's there), and then a "very ugly" form area below that horizontal line with four radio buttons, an input box, and 2 buttons. That lower area is ugly because we only have points for the HTML and CSS that happen **above the line** (i.e the four pretty panels). Your solution must look EXACTLY like the image above for the area above the pretty area line, but your solution does NOT need to have any HTML or CSS "look" requirements for the area below the line. Below the line does need to be 4 radio buttons, an input box, and two buttons, but it can look terrible and that's fine. Above the line = pretty and functional, below the line = functionality only. More details provided below.

Given images

We will provide the four house crest images. They came from: https://harrypotter.fandom.com/wiki/Gryffindor. You can either download them from here:

House Crest Images .zip

Then put them into your project, into perhaps an images folder:



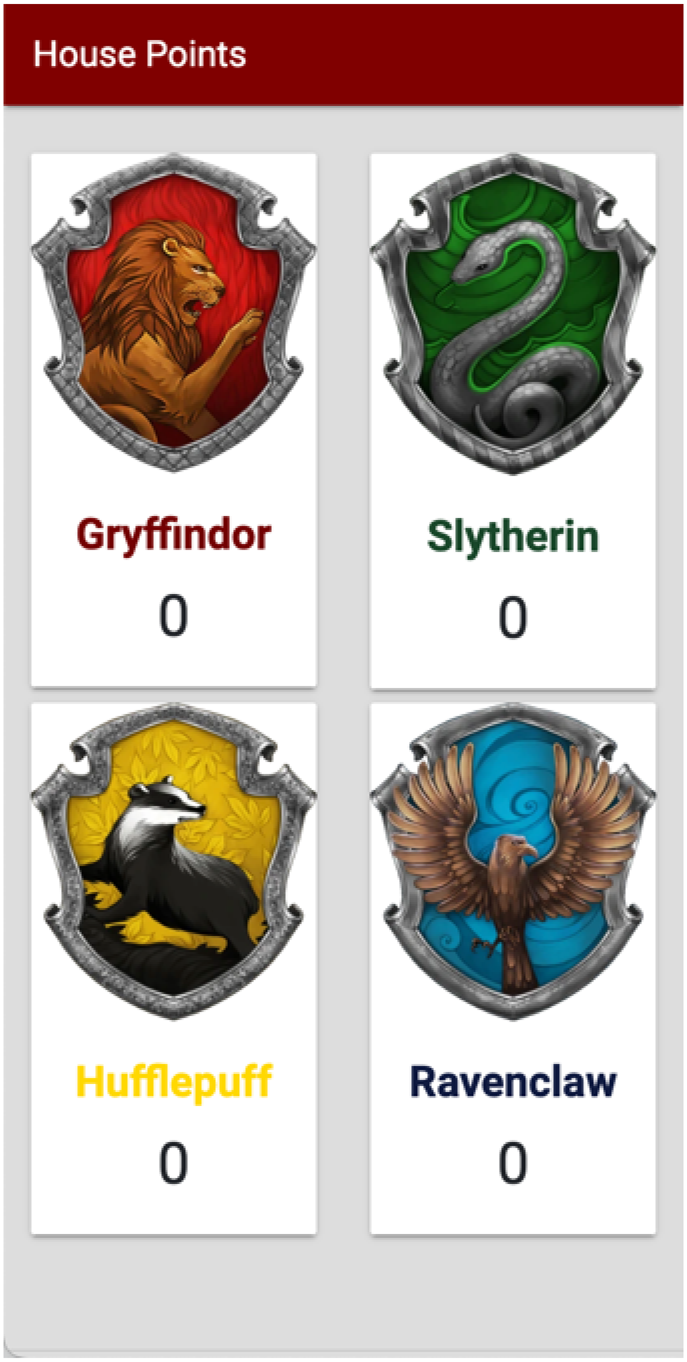
```
public
  images
    Gryffindor.webp
    Hufflepuff.webp
    Ravenclaw.webp
    Slytherin.webp
  scripts
  styles
  404.html
  favicon.ico
  index.html
```

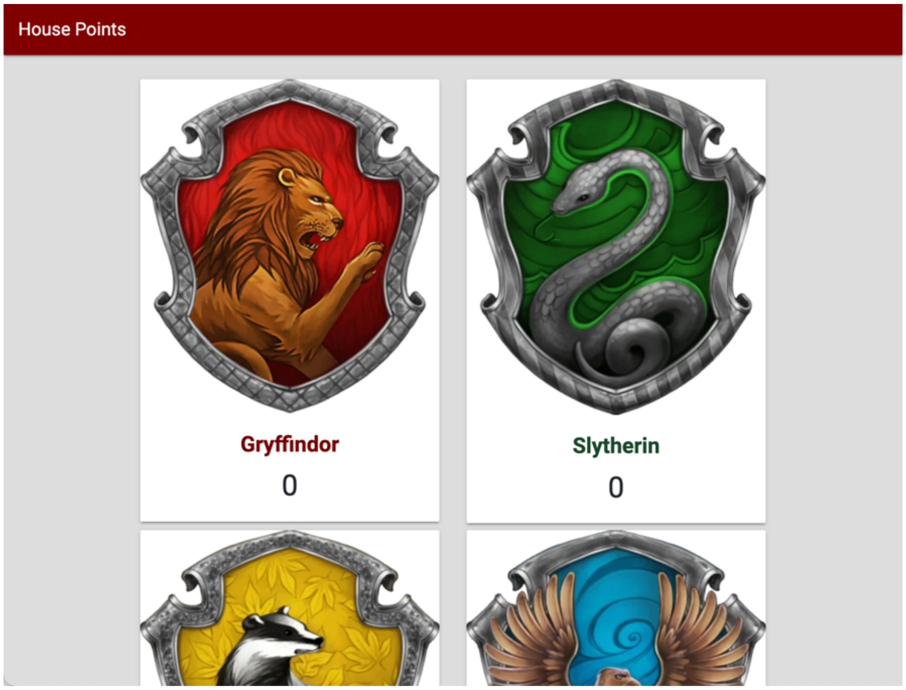
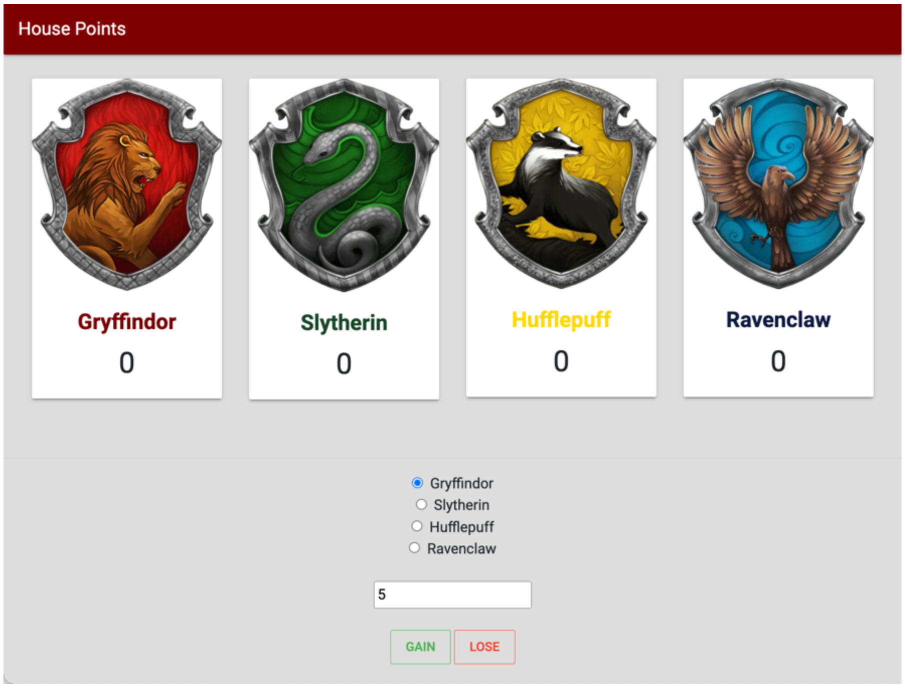
Or you can actually just use the URLs of the images that exist already on the internet. Here are the links to the URLs you could use.

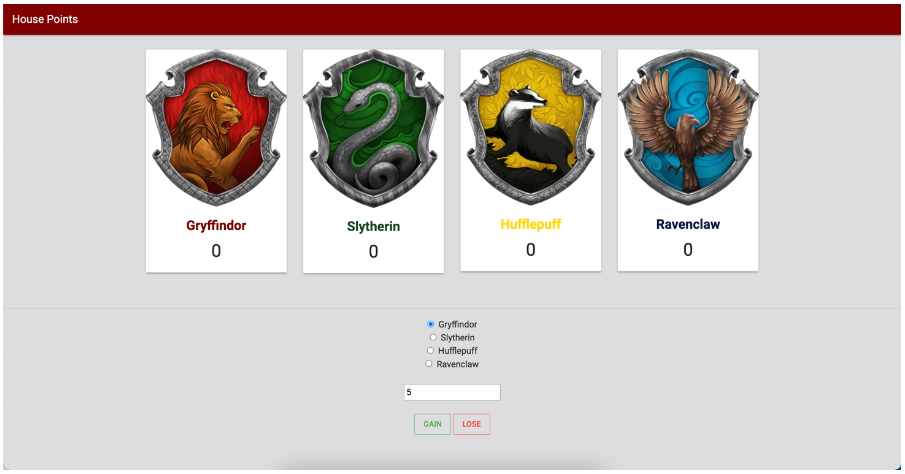
- Gryffindor
- Slytherin
- Hufflepuff
- Ravenclaw

You will use these images to make the four cards. Make sure to use the RHIT version of the documentation to get started with a card. For example: https://rhit-csse280.github.io/bootstrap-material-design/docs/4.0/bootstrap-components/card/

These panels should look different at different screen sizes. Let's look at some.

374px Note, smaller than 374px doesn’t look as good since the words don’t fit on one line. That’s ok. 374px is the smallest we care about.	 House Points Gryffindor 0 Slytherin 0 Hufflepuff 0 Ravenclaw 0

991px still 2 cards per row (sorry the bottom got cut off) Notice there is a small gap between the cards vertically that you must add.	
992px Switches to 4 per row at this breakpoint	 House Points Gryffindor 0 Slytherin 0 Hufflepuff 0 Ravenclaw 0 ● Gryffindor ○ Slytherin ○ Hufflepuff ○ Ravenclaw 5

Really not much exciting changes after 992px. It just does default Bootstrap things.	 House Points Gryffindor 0 Slytherin 0 Hufflepuff 0 Ravenclaw 0 ● Gryffindor ○ Slytherin ○ Hufflepuff ○ Ravenclaw 5 GAIN LOSE

Things you should notice and do with Bootstrap classes at CSS (you can use either).

- The layout is 4x1 for 992px and bigger, 2x2 for 991px and smaller
- Text is all centered (house name and points)
- Text color for the house name uses the following colors:
  - Gryffindor Red. Hex #740001
  - Slytherin Green. Hex #1A472A
  - Hufflepuff Yellow. Hex #FFD800
  - Ravenclaw Blue. Hex #0E1A40
- The points text is just black
- Hints: My house names are just h5 elements in a card-title class. My font weight for my house names is font-weight: 700;
- My house points numbers use a class h2 to set their text size.
- Your images for full credit should match the top area exactly
- Below the line, the “look” doesn’t matter (see functionality below).

Functionality

Once you have the look correct using HTML (Bootstrap classes) and CSS, then it’s time to get the functionality right. There are requirements on WHAT your code does and there are also requirements on HOW your javascript works. You are required to use 2 classes in your code: one for the Model object (called HousePoints) and one for the Controller object (called HpController). The HousePoints class should have 4 instance variables (for the four house point counters). Then exactly 1 method:

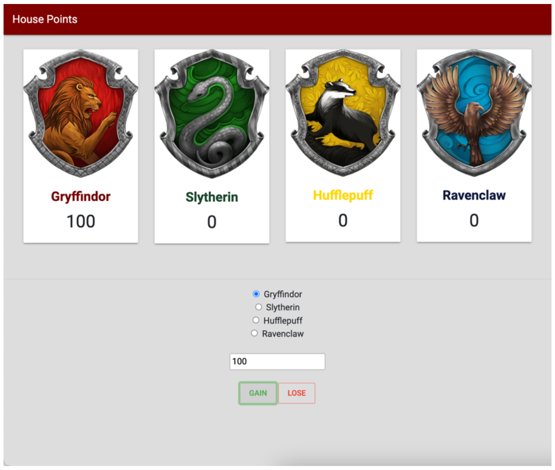
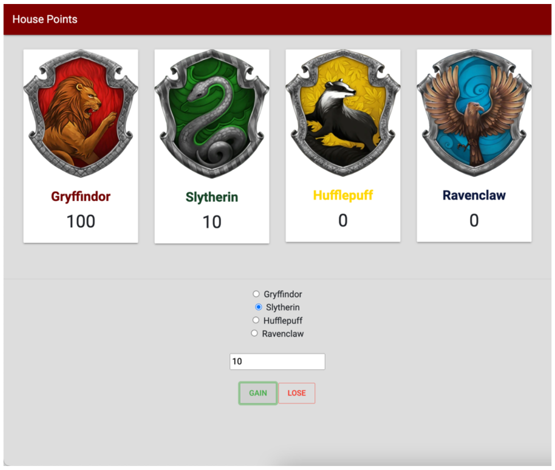
```
change(house, points)
```

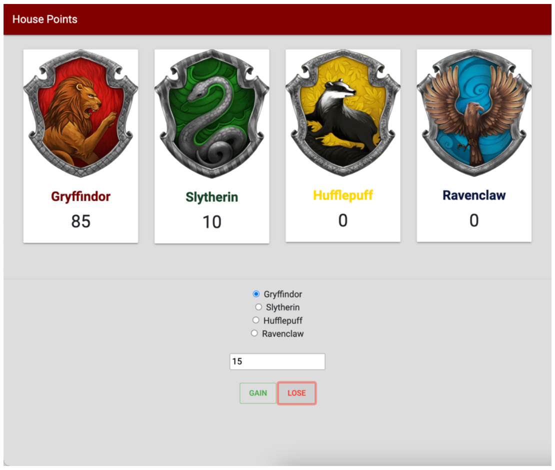
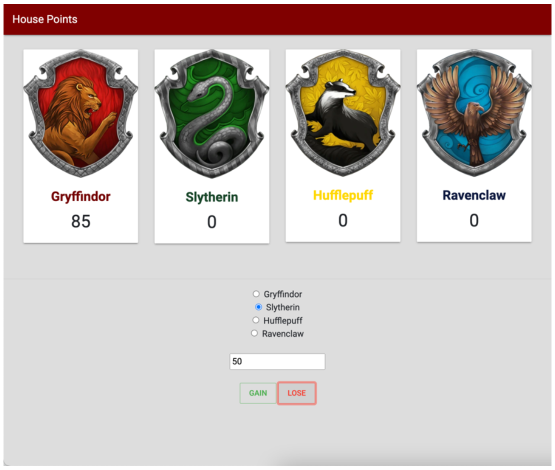
The first argument should be a string that is either `“g”, “s”, “h”, or “r”`. The second argument should be a positive or negative integer.   For example:

```
housePoints = new rhit.HousePoints(); // Any mechanism to make an
instance of the HousePoints class is fine.
housePoints.change("g", 100);
housePoints.change("s", 10);
housePoints.change("g", -15);
housePoints.change("s", -50);
```

At the end of that sample code Gryffindor would be at 85 points and Slytherin would be at 0 points (note, that you can't go negative). Note: you must design your code this way for full credit in the JavaScript part of the exam.

The HpController needs to only have the constructor to setup the onclick listeners and an `updateView` method that is called whenever points change. Let's look at an example of a user, using your app. The page loads with all 0 point values (see images above), then…

1. Select Gryffindor's radio button 2. Type in 100 points 3. Hit Gain Notice Gryffindor is at 100 points.	 House Points Gryffindor 100 Slytherin 0 Hufflepuff 0 Ravenclaw 0 ● Gryffindor ○ Slytherin ○ Hufflepuff ○ Ravenclaw 100
1. Select Slytherin 2. Change the value to 10 3. Hit Gain Notice Gryffindor still has 100, but Slytherin has 10.	 House Points Gryffindor 100 Slytherin 10 Hufflepuff 0 Ravenclaw 0 ○ Gryffindor ● Slytherin ○ Hufflepuff ○ Ravenclaw 10

1. Select Gryffindor again 2. Change the value to 15 (notice it is a positive 15 not -15) 3. Hit Lose Gryffindor is down to 85 points.	 House Points Gryffindor 85 Slytherin 10 Hufflepuff 0 Ravenclaw 0 ● Gryffindor ○ Slytherin ○ Hufflepuff ○ Ravenclaw 15 GAIN LOSE
Select Slytherin again Change the value to 50 Hit Lose Notice that Slytherin is NOT at -40. They are at 0 (you can't go negative). Note that the capping at 0 must happen within the Model object, within the change method. The controller still calls something like: housePoints.change("s", -50); but the model object caps it at 0.	 House Points Gryffindor 85 Slytherin 0 Hufflepuff 0 Ravenclaw 0 ○ Gryffindor ● Slytherin ○ Hufflepuff ○ Ravenclaw 50 GAIN LOSE

Firebase

You can firebase init and firebase serve if you like (not necessary!), but you'll **NEVER** firebase deploy this exam.

Submitting your work

You will submit a .zip of your project folder to Moodle. Make sure to zip the public folder, which includes the images folder. You will be answering some questions on Gradescope as well.