1

Develop a web application using **Javascript to** add behavior to the following page that has a user interface for entering grades on homework assignments.

a. You will compute the percentage of points earned, with an optional **curve**.

i. When "Compute!" is clicked, your JS code should use the values in the text boxes to compute the percentage (**rounded** to the nearest percent).

ii. If the "Curve +5" checkbox is checked, add **+5 percent** up to a **maximum of 100%** total.

b. You should insert the percentage into the page as a new div added to the end (bottom) of the existing page section with the id of resultsarea.

i. If the overall percentage is **60% or more**, give your newly created div a CSS class of pass; otherwise give it a class of fail.

ii. Each time the user clicks "Compute!", you will insert such a new div; this means that several div's would be there after several clicks of "Compute!".

b. You should insert the percentage into the page as a new div added to the end (bottom) of the existing page section with the id of resultsarea.

i. If the overall percentage is **60% or more**, give your newly created div a CSS class of pass; otherwise give it a class of fail.

ii. Each time the user clicks "Compute!", you will insert such a new div; this means that several div's would be there after several clicks of "Compute!".

iii. When **"Clear"** is clicked, all text in all of the input text boxes should be erased.

**Assume valid input**; that is, assume that when "Compute!" is clicked, the user will have already typed valid text into every box that can be interpreted as an integer.

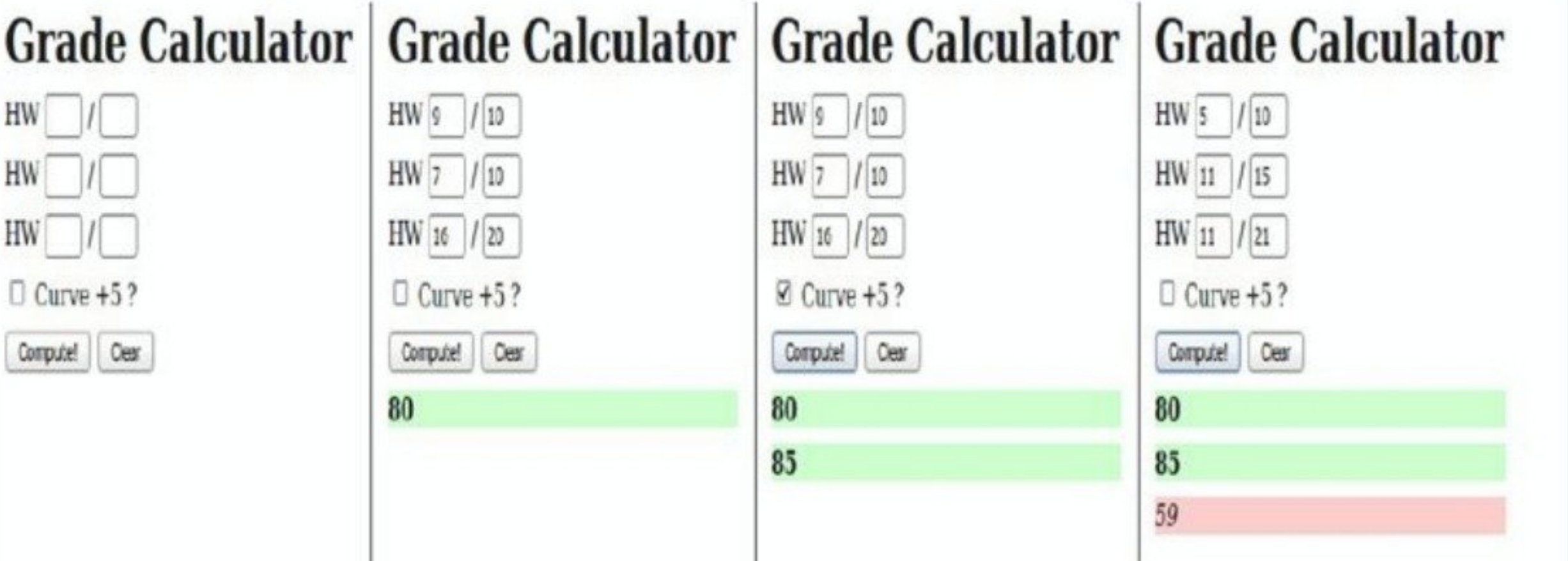
iii. When **"Clear"** is clicked, all text in all of the input text boxes should be erased.

**Assume valid input**; that is, assume that when "Compute!" is clicked, the user will have already typed valid text into every box that can be interpreted as an integer.

These screenshots show the initial state, and state after scores have been typed and "Compute!" has been clicked.

(Non-anonymous question ⓘ) *

(20 Points)



1

Develop a web application using **AngularJS** to display a list of possible flight prices and carriers using JSON data.

i. The page contains two text boxes where the user can specify the start location and end location, a checkbox that they can check if they want to only see non-stop flights (flights with 0 stops) and a "Go!" button.

ii. When the "Go!" button is clicked, clear previous results, and read the JSON data.

a. Add an h1 to the results div containing the text "Flights from" and then the start and destination locations.

b. Turn each flight's data into a paragraph in the results div.

c. In each paragraph, write the price of the ticket (with a "$") followed by the word "from", the carrier's name, and then the word "with", the number of stops, and the word "stops". If the flight has a price below 1000 display its row in bold.

iii. The JSON data returned has a list of flights associated with it. The list of flights contains lists that each contain a price, a carrier, and the number of stops.

```
{

 "Edinburgh":{
```

```
{
"Edinburgh":{
"start":"Seattle",
"flights":[{"carrier":"Delta","price":812,"stops":2},
{"carrier":"Air France","price":1020,"stops":0},
{"carrier":"Air France","price":1190,"stops":3}]
},
"New York":{
"start":"Seattle",
"flights":[{"carrier":"British Airlines","price":782,"stops":1},
{"carrier":"Delta","price":1562,"stops":2},
{"carrier":"United","price":957,"stops":1},
{"carrier":"KLM","price":687,"stops":3},
{"carrier":"KLM","price":1458,"stops":1}]
}
}
```

The relevant existing HTML in the page is the following:

```
<div>

<label>Start location: <input type="text" id="start" /></label>

<label>End location: <input type="text" id="dest" /></label>

<label>Non-stop? <input type="checkbox" id="stops"></label>

<button id="go">Go!</button>

</div>

<div id="results"><h1></h1></div>
```

(Non-anonymous question) *

(20 Points)

Start location: seattle End location: Edinburgh Non-stop? Go!

# Flights from seattle to Edinburgh

**$812 from Delta with 2 stops**

$1020 from Air France with 0 stops

$1190 from Air France with 3 stops

Close

Start location: seattle End location: Edinburgh Non-stop? Go!

# Flights from seattle to Edinburgh

**$812 from Delta with 2 stops**

$1020 from Air France with 0 stops

$1190 from Air France with 3 stops

---

Start location: seattle End location: Edinburgh Non-stop? Go!

# Flights from seattle to Edinburgh

**$1020 from Air France with 0 stops**

Upload file

File number limit: 1 Single file size limit: 10MB Allowed file types: PDF

Submit

1

Develop web application using **JQuery** to add behavior to the following page for finding palindromes. A *palindrome* is a word that is spelled the same forward as backward, such as "madam" or "Anna".

a. The page UI allows the user to type a phrase into a text box. The user can click a "Find Palindromes" button to find palindrome words in that phrase. Match case-insensitively; You may assume that words in the phrase are separated by single spaces and contain only letters.

b. Each palindrome found should be inserted as a bullet into a list with the id of palindromes. Every other palindrome (the first, third, fifth, etc.) should be given a gray background color of #CC0000. Underneath the list of palindromes you should display text such as "5 total palindrome(s)" in the div with id of count.

c. The user can optionally specify a minimum and maximum word length by typing integer values into two text boxes with id of min and max respectively.

i. If a minimum is specified, you should include only palindrome words that contain at least that many letters inclusive.

ii. If a maximum is specified, you should include only palindrome words that contain at most that many letters inclusive.

ii. If a maximum is specified, you should include only palindrome words that contain at most that many letters inclusive.

iii. If the min or max is left blank, the length is unbounded in that direction. For example, a minimum of 3 and a blank maximum finds all palindromes of at least 3 letters. You may assume that the text typed in these boxes will either be blank or a valid non-negative integer, and that max will be ≥ min.

d. The code should work for multiple clicks of the button. On each click it should clear any previous found information.

These screenshots show the initial state, and after phrases have been typed and "Find Palindromes" is clicked.

(Non-anonymous question ⓘ) *

(20 Points)

# Palindrome Finder!

Phrase:

Length: to

Find Palindromes

# Palindrome Finder!

Phrase: Hi Madam I did see the sTats that both SexEs prefer a Honda CIVIC raceCar

Length: to 3

Find Palindromes

- I
- did
- a

3 total palindrome(s).

# Palindrome Finder!

Phrase: Hi Madam I did see the sTats that both SexEs prefer a Honda CIVIC raceCar

Length: 5 to

Find Palindromes

- Madam
- sTats
- SexEs
- CIVIC
- raceCar

5 total palindrome(s).

# Palindrome Finder!

Phrase: Hi Madam I did see the sTats that both SexEs prefer a Honda CIVIC raceCar

Length: to

Find Palindromes

- Madam
- I
- did
- sTats
- SexEs
- a
- CIVIC
- raceCar

8 total palindrome(s).

# Palindrome Finder!

Phrase: Hi Madam I did see the sTats that both SexEs prefer a Honda CIVIC raceCar

Length: 3 to 5

Find Palindromes

- Madam
- did
- sTats
- SexEs
- CIVIC

5 total palindrome(s).

Upload file