Unit 8: React: Lab 1 + 2 – Ice Cream Wars Pt. 1 + 2

Assigned: 01/19/23

Due: 01/25/23

Completed: 01/25/23

**LMS:** <https://lms.grandcircus.co/mod/assign/view.php?id=22818>

**Google Doc:** <https://docs.google.com/document/d/1kUz7kOrma1-o2GmC-QB4Sy-yWFnHtAAAej02imEHrd4/preview>

**GitHub:** <https://github.com/clintmsmith/GrandCircusLabs/tree/main/Ice_Cream_Wars_Pt1/ice_cream_wars>

**PART 1**

**Task:** Create the "Ice Cream Wars" website as a Single Page Application using React. The site provides ways to support and vote for your favorite ice cream flavor. (You can pick any three flavors that you want to use in your activity.) It is composed of four components: App, Header, AdDesigner, and Votes.

Graphical user interface, application

Description automatically generated

**COMPONENTS:**

**App**: This includes the other three components and lays them out on the page.

**Header:** This is a simple component that displays the site title: "Ice Cream Wars". Use the <header> semantic tag for the root element. Include an associated CSS file to apply relevant styles to this component, including the font, color, and box shadow.

**AdDesigner**: This component allows the user to design an Ad for their favorite ice cream flavor.

* It displays an "ad" which can be customized in three ways by clicking buttons:
  1. Pick the flavor to show on the ad. Store this as a string in the state.
  2. Pick the color theme: light (dark text on white background) or dark (light text on dark background). Store this as a Boolean in the state.
  3. Pick the font size in pixels. This can be increased or decreased by clicking the "Up" and "Down" buttons. Store this as a number in the state.
* Provide reasonable starting values for the three customizations.
* Disable the flavor selection button that is currently selected.
* Disable the color theme button that is currently selected.

**Votes**: This component allows the user to add votes to each of the ice cream flavors. They can click any flavor as many times as they want. Each time they click, the vote count for that flavor is increased and the graph is updated along with the totals and percentages.

* Hint: Percentage is calculated as [votes for this flavor] / [total votes] \* 100. Also total votes can be calculated by adding together all the votes for the individual flavors. Watch out for the case where there are no votes yet; division by zero gives NaN.
* Hint: The "graph" bars are just <div>s with a percentage width set.
* Do not display information for a flavor if it has not received any votes yet. (For example, this screenshot shows when strawberry has no votes yet.)  
  Graphical user interface, text, application, chat or text message

  Description automatically generated
* If there are no votes at all, display "No votes yet."  
  Graphical user interface, text, application, chat or text message

  Description automatically generated

**Style Guide:**

* Page title: Ice Cream Wars
* Mobile-first development. (See mobile screenshot below, desktop screenshot above.)
* Main font: sans-serif
* Heading and Ad flavor font: Caveat from Google Fonts
* Page header color: #0e3c7e  [  ]
* Button color: #0e3c7e  [  ]
* Button active color: #1e60a9  [  ]
* Button disabled color: gray  [  ]
* Ad border color (light theme): #222  [  ]
* Ad text color (light theme): #000  [  ]
* Ad background color (dark theme): #222  [  ]
* Ad text color (dark theme): #fff
* Ice Cream colors: Pick your own flavors and colors. The example uses #4f2c2c, #cbc0ac, #de8c91

**Mobile Design**

Graphical user interface, application

Description automatically generated with medium confidence

**Grading Criteria**

1. Header component uses a <header> element and displays "Ice Cream Wars". It is included at the top of App.
2. Header component is styled correctly: font, color, shadow.
3. AdDesigner component allows customization of ad text via three flavor buttons and a string state variable.
4. AdDesigner component allows customization of ad color theme via two theme buttons and a Boolean state variable.
5. AdDesigner component allows customization of ad font size via two up/down buttons and a number state variable.
6. AdDesigner flavor buttons and color theme buttons are disabled when that option is already selected.
7. AdDesigner component is styled correctly: buttons; disabled buttons; ad size, layout, and shadow.
8. Votes component displays independent counts for each of three flavors which are increased by clicking the flavor buttons.
9. Votes component shows correctly calculated percentages with the counts.
10. Votes component displays correctly sized graph bars based on percentages.
11. Votes component hides flavors that have no votes yet.
12. Votes component displays "No votes yet." when no votes have been cast yet.
13. Page and App component are styled correctly in mobile: components are stacked vertically, fonts, spacing, alignment.
14. Page and App component are styled correctly in desktop: AdDesigner and Votes side-by-side, spacing, alignment.

When we created the State Hook and gave it the initial value ‘Chocolate’, this is what appears on the screen after we added it our HTML with {flavor} (the name of the variable).

Text

Description automatically generated

Text

Description automatically generated

**PART 2**

**Task**: This activity builds on Ice Cream Wars Pt. 1. Add a few features and refactor the AdDesigner slightly.

**Build Specifications**

* Add a string user prop to the **Header** component. Display this at the right side of the header. Pass in a hard-coded value from the App component. (This screenshot shows the user set to "Chirpus")  
  
* Create an **Ad** component. It has three props, and displays an ad based on these customizations. This should look the same as the ad at the top of the AdDesigner.
  + flavor (a string)
  + fontSize (a number)
  + darkTheme (a Boolean)  
    
* In the App Component, hard-code three **Ad**s at the top, below the header. Choose values to hard-code for their props.  
  
* Refactor the **AdDesigner** component using the new **Ad** component to replace the portion that displays the customized Ad. It should look the same in the end.

**Extended Challenges**

* Use props to set the three flavor names and colors in the Votes component.

continued on the next page...

**Grading Criteria**

1. Header component displays welcome using the name from the user prop.
2. The header is styled correctly: title on left, welcome on right.
3. App component passes a hard-coded name prop to the Header component.
4. Ad component displays customized text from string flavor prop.
5. Ad component displays correct color theme from Boolean darkTheme prop.
6. Ad component displays correct font size from number fontSize prop.
7. Three Ad components with different prop values are displayed at the top of the page, below the header.
8. The AdDesigner component incorporates the Ad component, using the correct prop values from its state.