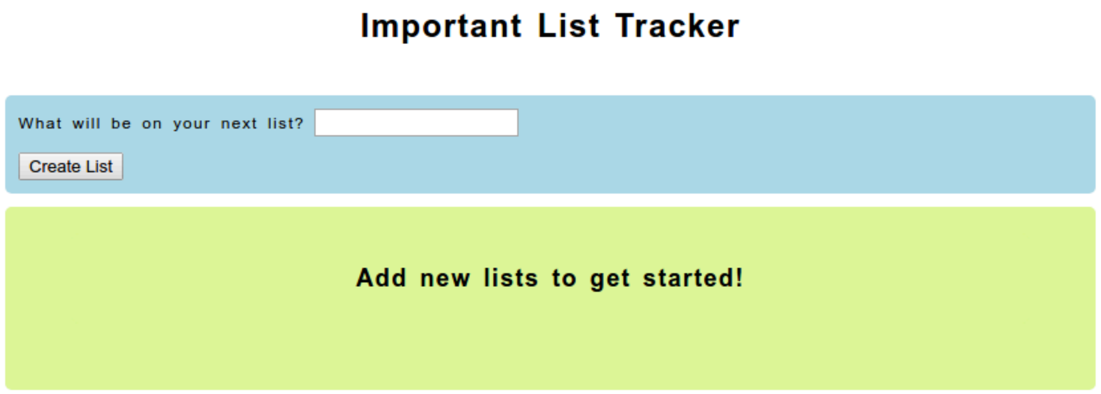
Lab 07 - React App - List Tracker

In this lab, you will develop a simple application to track lists of items using React.
The application you will be building has many components that work together in order to organize multiple lists. Such an app could theoretically be used to create a shopping list, a set of daily tasks, or so forth. You can use this app in your project, such as creating an app for admins to add items or a shopping cart.
In completing this assignment, you will:

- Gain experience understanding and modifying an existing React app
- Use the Node framework to create and deploy a React app with multiple components defined in separate files
- Gain more experience in implementing callback functions in React components that affect the components' appearance
- Apply what you have learned in the lessons about the relationships between React components and how components interact

Background

For this assignment, we will create a React app that initially looks like this:

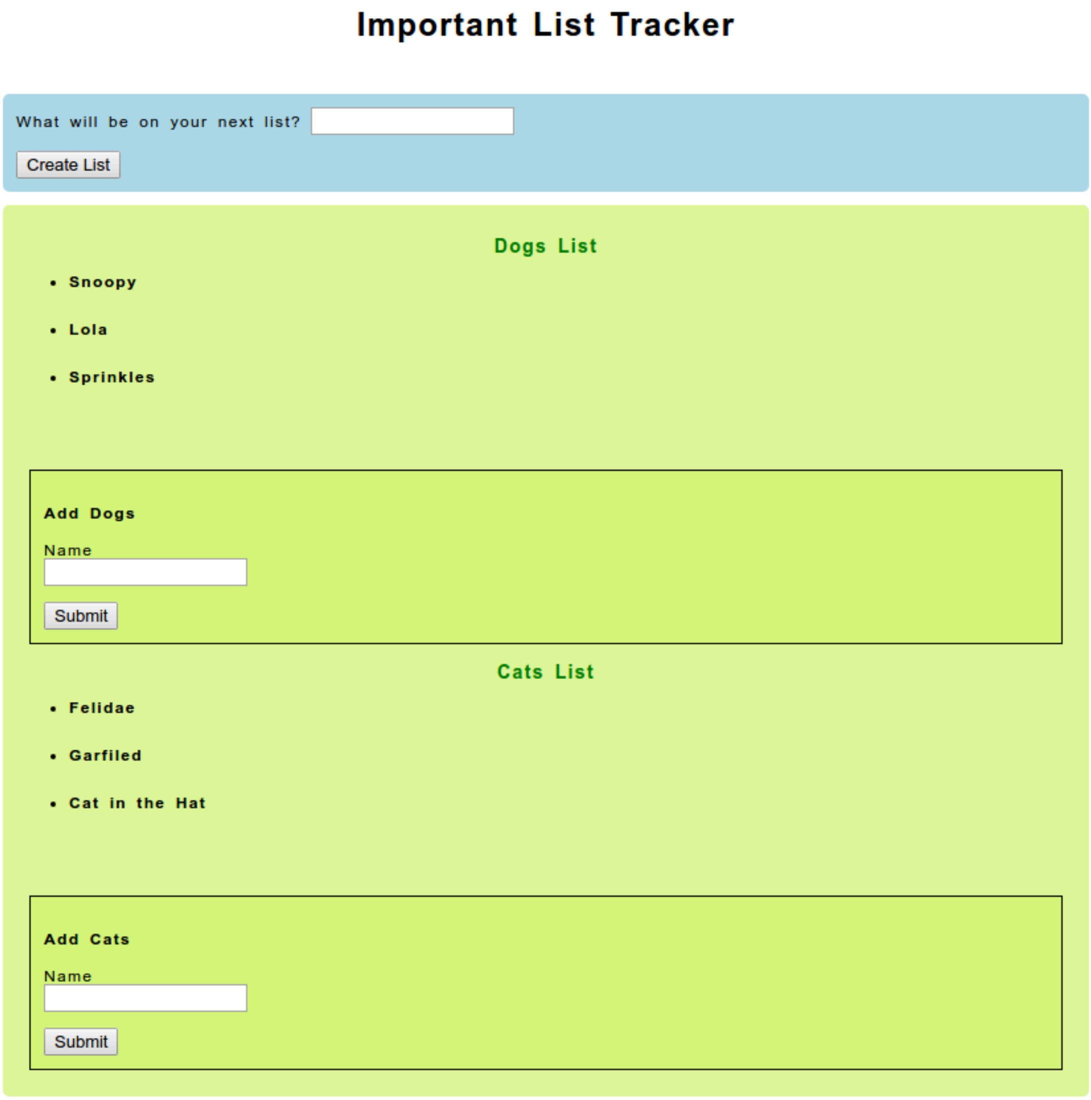


Important List Tracker

What will be on your next list?

Add new lists to get started!

And the user can create new lists, and then add items to those lists, so that it can for instance look like this:



Important List Tracker

Dogs List

- Snoopy
- Lola
- Sprinkles

Cats List

- Felidae
- Garfiled
- Cat in the Hat

Getting Started

To complete this assignment, set up a development environment (https://react.dev/learn/build-a-react-app-from-scratch#vite) that uses Node.js and the Vite to create react app using this command:

npm create vite@latest my-app -- --template react

Name your project lists.

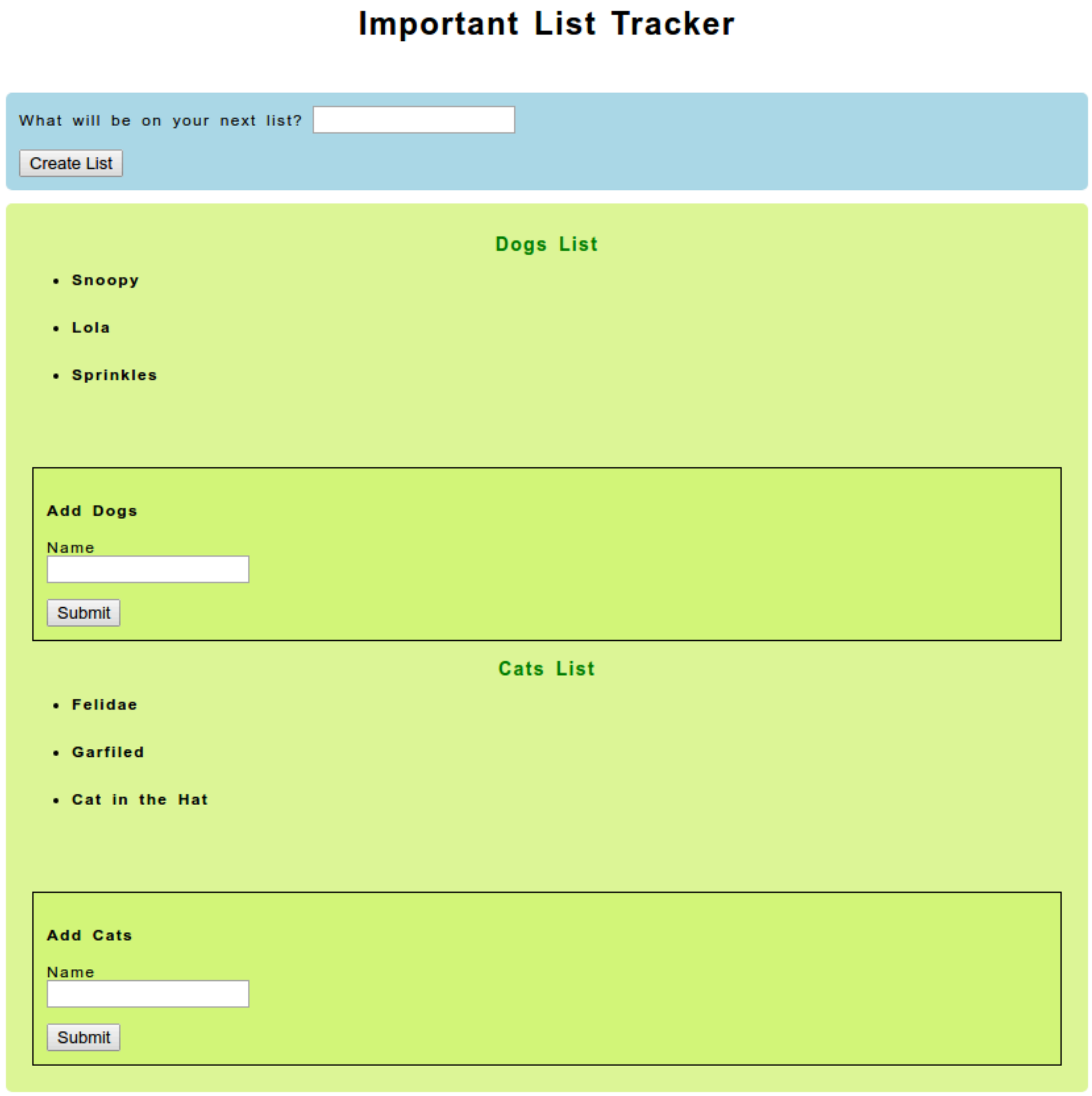
Activity

The goal of this assignment is to build an app that the user can:

1. Create a new list (40 Marks)

2. Add items to that list (40 Marks)
3. Mark items by changing their color (20 Marks)

When you are finished, the app should look something like this, depending on the user inputs:



Important List Tracker

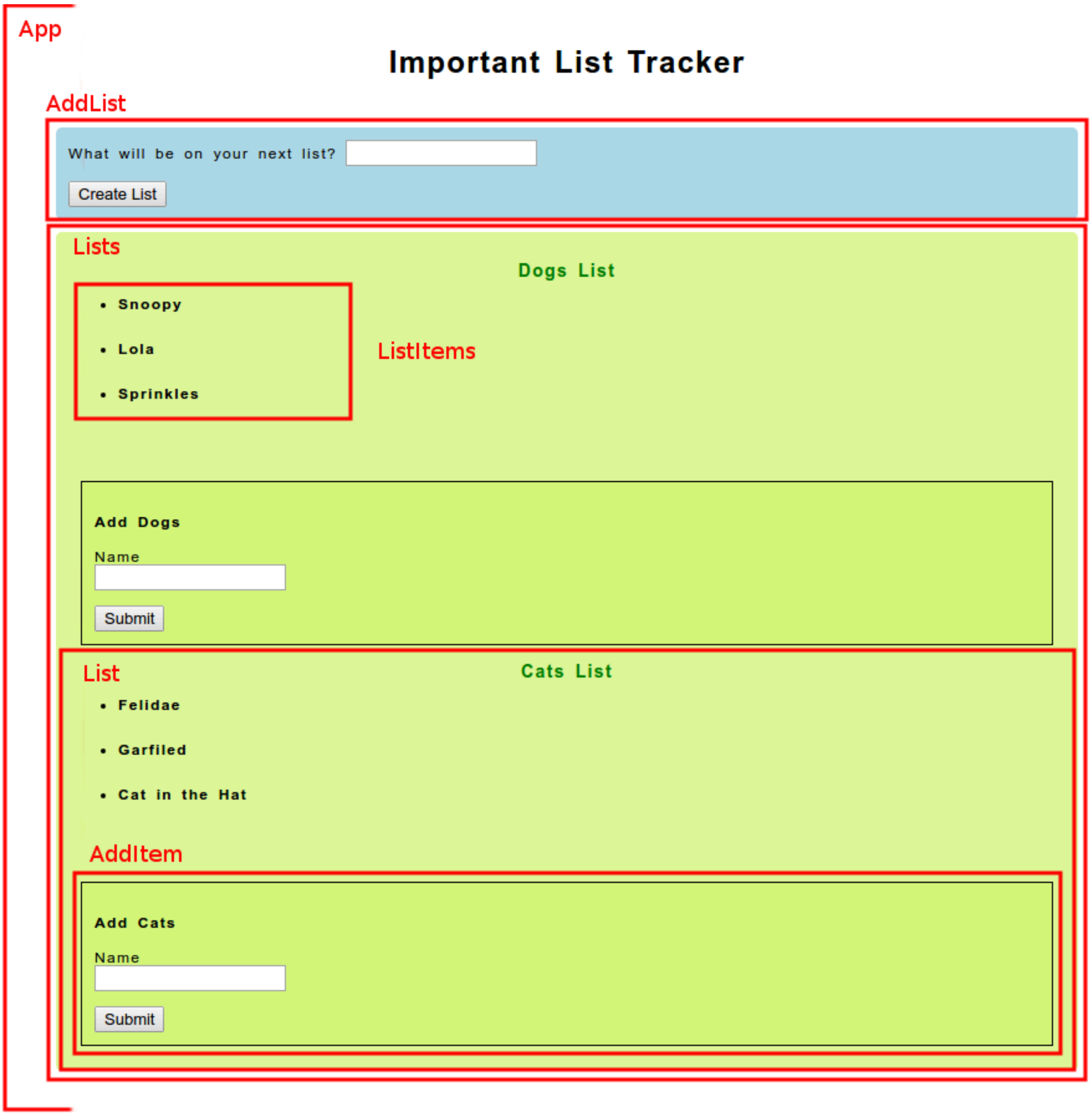
Dogs List

- Snoopy
- Lola
- Sprinkles

Cats List

- Felidae
- Garfiled
- Cat in the Hat

To accomplish this, you can use six React components that are organized like this:



Important List Tracker

What will be on your next list?

Dogs List

- Snoopy
- Lola
- Sprinkles

Add Dogs

Name

Cats List

- Felidae
- Garfiled
- Cat in the Hat

Add Cats

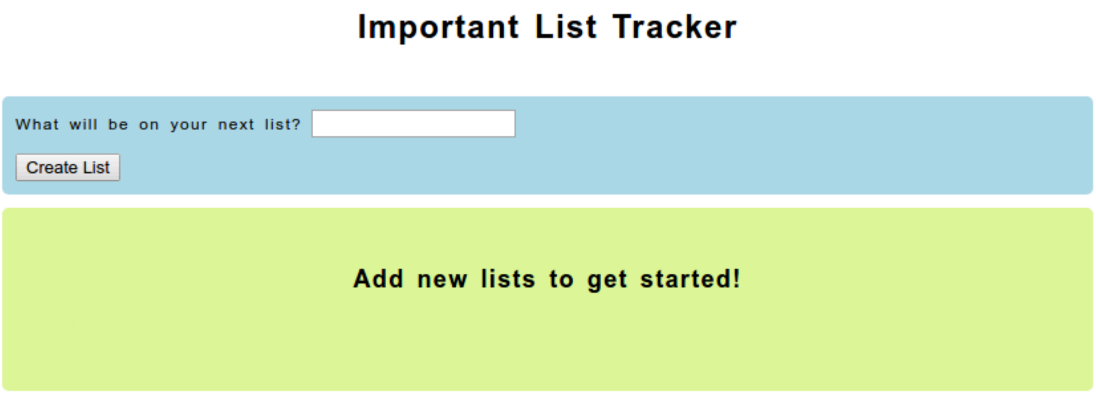
Name

The App component is the main component of the app and its render function creates two additional components:

- AddList, which is responsible for taking user input and passing information back to the App component in order to create a new list
- Lists, which is empty when the App is first rendered, but contains all instances of individual List components, each of which contains an AddItem component for adding items and ListItem components for all items in the list

Before proceeding, be sure you understand the role of each component in the app and how the components are related.

Part 1. Creating a new List (40 Marks)

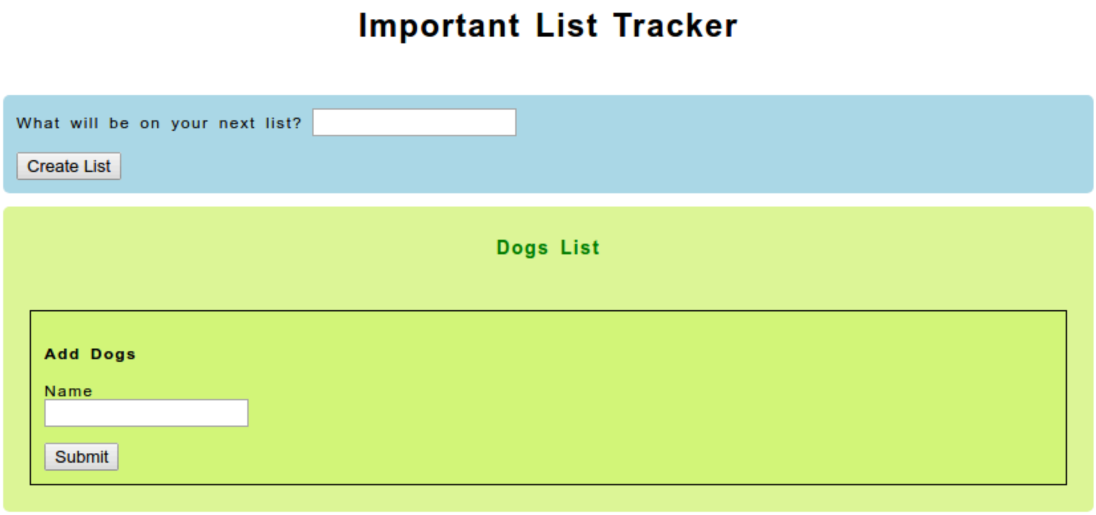


Important List Tracker

Add new lists to get started!

In the image above, we see the App component render two child components: AddList and Lists. The AddList component contains the text input field next to the words “What will be on your next list?” and a button labeled “Create List”. The Lists component by default reads “Add new lists to get started!”, but will eventually display each individual List component, which consists of an AddItem component and ListItem components.

Implement the AddList.handleSubmit and App.handleAddList functions and modify the components as necessary so that it is possible for the user to create a new list and see the List and AddItem components rendered in the page. The image below shows what the app should now look like when the user adds a “Dogs” list.



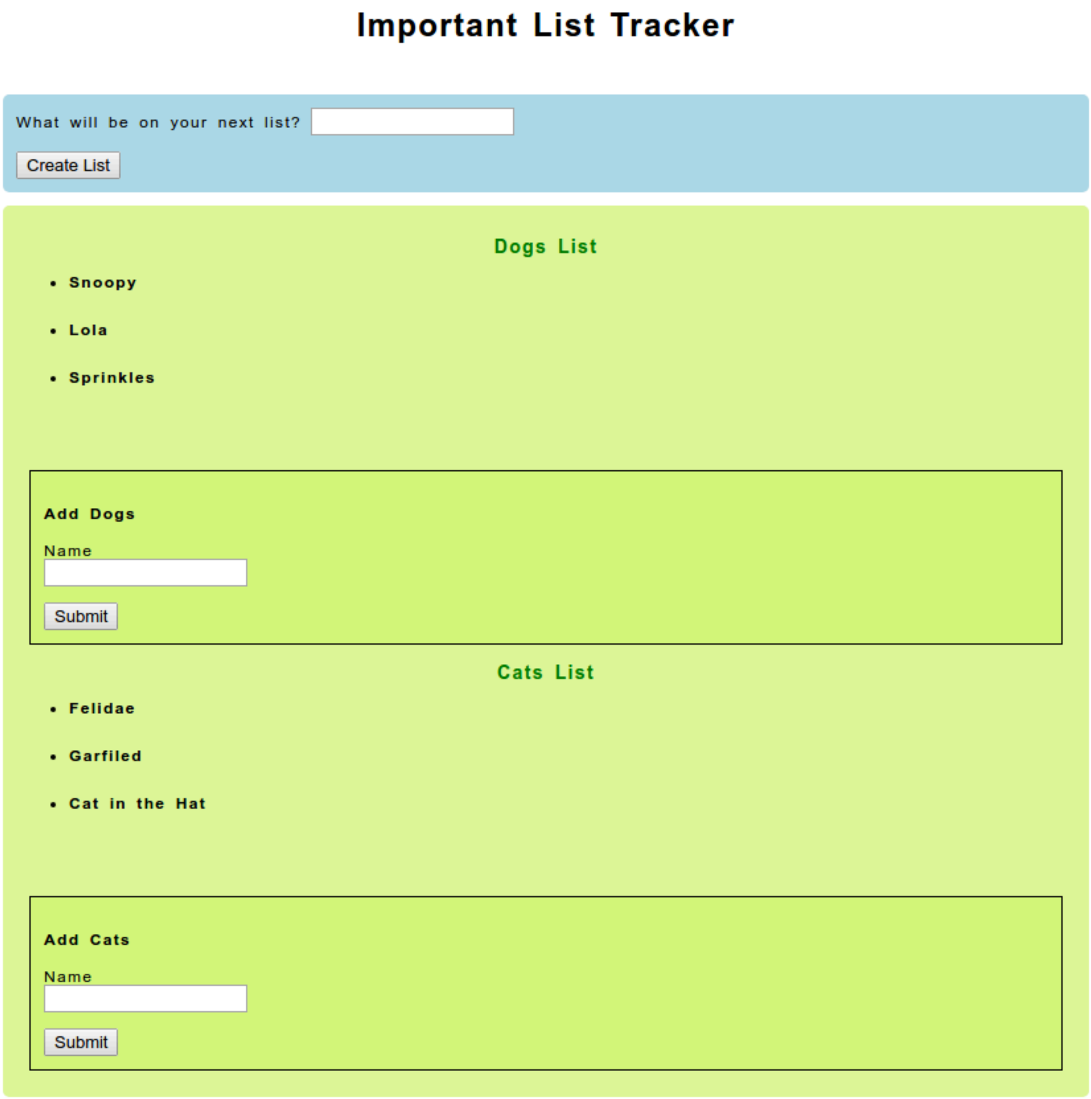
Important List Tracker

At this point, the App’s state should look like this:

```
{
items: {Dogs: []},
lists: ["Dogs"]
}
```

Part 2. Adding ListItems to a List (40 Marks)

Now implement the components as necessary so that it is possible for the user to add an item to a list and see the ListItem component rendered in the page. Below is an image depicting the results of adding two lists – one for Dogs and one for Cats – and adding three items to each.



Important List Tracker

What will be on your next list?

Dogs List

- Snoopy
- Lola
- Sprinkles

Add Dogs

Name

Cats List

- Felidae
- Garfiled
- Cat in the Hat

Add Cats

Name

At this point, the App's state should look like this:

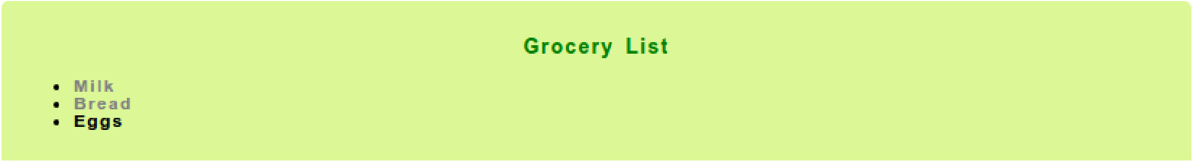
```
{
items:
    {Dogs: [{name: "Snoopy"}, {name: "Lola"}, {name: "Sprinkles"}],
    Cats: [{name: "Felidae"}, {name: "Garfiled"}, {name: "Cat in the Hat"}]
},
lists: ["Dogs", "Cats"]
}
```

Part 3. Marking a ListItem (20 Marks)

Certain lists, such as To-Dos or grocery lists, would be greatly improved by the ability to “mark” an item as complete or purchased.

Implement a function so that it is invoked when the user clicks on the content in the ListItem, i.e., within the <span> element. The function should alternate the color of the text in the ListItem between black and gray.

The image below depicts what would happen if you added Milk, Bread, and Eggs to a Grocery list and clicked on the Milk and Bread ListItems:



Grocery List

- Milk
- Bread
- Eggs

Submission: Please submit only the files you created, along with a manual. The manual should include the commands needed to create the app and specify where to replace the original files to see your results. Clearly explain how to run your app. Do not submit the entire library folder.