Putting it All Together: State and Events (CodeGrade)

- Due Mar 11 by 11:59pm
- Points 1
- Submitting an external tool
- Available until Mar 11 at 11:59pm

This assignment was locked Mar 11 at 11:59pm.

 (https://github.com/learn-co-curriculum/react-hooks-state-events-mini-project)  (https://github.com/learn-co-curriculum/react-hooks-state-events-mini-project/issues/new)

Learning Goals

- Use state and events to make components dynamic
- Implement controlled components

Introduction

To build on what you've learned over the course of this section, we'll be building out a simple task list app to practice working with state and events, focusing in particular on working with arrays.

Deliverables

There is some starter code built out for all of the components you'll need. The data for the application is imported in `App`, so you'll need to pass that data down to the components that need it as props.

Run `npm install` and `npm start`, then check out the starter code in the browser. You'll see a console message with the `TASK` and `CATEGORY` data you'll need to pass down from `App`.

TaskList

First, we'll want to display all the tasks in our app. Pass down the task data from `App` to `TaskList`, and display each task using the `Task` component. Make sure to use a `key` prop!

Task

Update the `Task` component so that it shows the task's text and category.

When the delete button is clicked, the task should be removed from the list.

CategoryFilter

Pass the list of categories to this component from `App`. Then, update this component to display `<button>` elements for each category. In order to pass the test, the buttons will need a key prop equal to the category.

When a button is clicked, the following should happen:

- Whichever button was clicked should have a class of `selected`. The other buttons should not have any class assigned.
- The list of tasks being displayed should be filtered, so that only tasks that match the category that was clicked are displayed.
- If the button for "All" is selected, all the tasks should be displayed.

NewTaskForm

Pass the list of categories to this component from `App`. Then, update this component to display `<option>` elements for each category inside of the `<select>` element **except** the "All" category, so that the user can select a category when adding a new task.

Next, update this form to be a *controlled component*, so that all form inputs are captured in state.

When the form is submitted, add a new task to the list with the text and category from the form. For the tests for this feature to pass, you'll need a callback prop named `onTaskFormSubmit` that takes a task object as an argument.