IST 363

# Lab 11

## This Lab Covers

* React – events and state

## NOTE

1. **Submission:**This week you should only have one App.jsx file so you will submit that. I don’t think that Blackboard will allow you to upload a .jsx file so put it in a zip folder before you upload. On both the Mac and PC you should be able to right click on the App.jsx file and choose compress or create zip file to do this.
2. **Web Server:** You will need to have your web server running to view the files you are working on in this lab. If it’s not running, open the folder we worked on in class this week in VS Code, then go to the terminal in VS Code and type npm run dev

New terminal > type “cd hacker-stories” then “npm run dev”

## React 1

**Goal:** This week we are making a dynamically loading to do list that stores the status of tasks.

Here’s a high-level overview of what the to do list will do:

* List tasks
* Mark tasks completed
* Add tasks to list

1. **Object**  
   Create an object to hold a unique id (needed for list key) for your to do item, the to do item and the status (completed or not completed) for you to do item.   
     
   Put two to do items in the list: Complete Lab 11 and Review JSX Events and State. Both of them are incomplete at this point.
2. **Display List**Use an HTML unordered list and the map function to list your to do items. When you are done it should look like this:  
     
   A close-up of a list

   AI-generated content may be incorrect.
3. **Mark Tasks Completed**For this part, we are going to add a button or graphic to click on to mark a task complete. Task lists don’t delete a task when it is complete, they cross it off. Below is what my list will look like after I press the X button on a task.  
     
   The first thing you need to do when the X button is clicked is change the object so the task is marked as completed. Step two is re-rendering the list so that the completed items are crossed out. Crossing out text is done with CSS: <https://www.w3schools.com/cssref/pr_text_text-decoration.php>.   
     
   A close up of a text

   AI-generated content may be incorrect.
4. **Add Tasks to the List**Add a form element to the code where users can type in new tasks and a button to submit the new task. Once they’ve pressed the button add the task to the object and re-render the list.

|  |  |
| --- | --- |
| Before Task Submitted | After Task Submitted |
|  |  |

**The End**