# Lab 11: Communicating from Child to Parent Component

## Objectives

*  In a child component, accept a function as a prop and invoke it
*  In a parent component, implement a function and pass it as a prop to a child component

## Steps

## In a child component, accept a function as a prop and invoke it

1. In the Prop Types definition, **add** an onEdit **function** that requires a project as a parameter and returns void.

#### src\projects\ProjectCard.js



1. Update the handleEditClick event to invoke the function passed into the onEdit prop and remove the console.log statement.

#### src\projects\ProjectCard.js



### In a parent component, implement a function and pass it as a prop to a child component

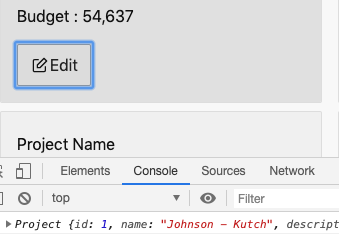
1. **Add** a handleEdit**event handler** toProjectListthat takes a project as an argument and **logs** it to the console.
2. **Wire** up the **onEdit** **event** of the ProjectCard component to the handleEdit event handler.

In VS Code, the code snippet nfn can help create the handleEdit event handler.

#### src\projects\ProjectList.js



1. **Verify** the **application** is working by following these steps:
   1. **Open** the application in your browser and refresh the page.
   2. **Open** the Chrome DevTools to the console (F12 or fn+F12 on laptop).
   3. **Click** the edit **button** for a project.
   4. **Verify** the project is logged to the Chrome DevTools console.



You may remember that logging was happening in a previous lab. In the previous lab, the logging was occuring in the child component. In this lab, we have removed the logging in the child component ProjectCard and are invoking a method on the parent list component ProjectList. The allows the card component to be easily reused in another part of the application.

### ✔ You have completed Lab 11