Hello its adarsh and for my CSCI 3155 extra credit project I decided to investigate the react framework within JAvascript, this framework is usually used to make webpages utlizing the teamwork of HTML and Javascript. Through react, we can create these components for our webpage and we can build our webserver using these components. Through my investigation of this langugae mode, I found various concepts that connect back to our class and what we have learned.

Lets first dive into how functions can be used as values in react. This example, we created a greeting function that returns a jsx element that contains the text hello world, and we use this function in another component called app, and we were able to output as a value. We were able to use the greeting function as a value, as we create a JSX element and we export that into our App component. This demonstrates the concept of functions as values, because were using a function more as a value than an expression, we are using a function as value in a large expressions and this very useful for react because it allows to create very complex webservers and applications.

The next topic I will dive into is callbacks and higher order functions. In this example were showing how the button component receives a callback function, onlclock, as a prop to then hand the button click event. The callback function is practically a function that is passes to another function as a function which are executed at relatively the same time. This demonstrates the concept of callbacks because it shows how a function can be passed as an argument to another function that can be executed at a different time. This is once again very useful for react because it allows use to create more complex functionswhich results in higher complexity webservers/applications. App is a higher-order function in itself because it takes the handleclick function and passes it to the Button function which, is an example of a higher order function because it takes another function and executes those functions inside first.

Finally, I will showcase abstract data types in react. Abstract data types, are a way of represrning data that is independent of its underlying implementation. A data structure that is defined in terms of its behavior and interface rather than its internal implementation. This example, is an ADT because were using the useState hook to create an abstract data type by encapsulating the state variable count and the setcount function. The count variable is the data and the setCount function is the interface that kind of allows us to interact with that data. When we use the useState hook we dont; need to worry about how the state is implemented internally, we can simply use the count variable and the setCount function, we dont have to know how useState is exactly working or its underlying implementation. Super helpful because its much more modular in a away, it makes our code more modular and reusable and we can use this counter component in multiple places without having to worry how state is implemented. Makes it easier to change the implementation without having to change alot of the outside code. Think of it like a box that kind of holds data, but we dont need to know how the box works inside, we can use the box by interacting with its outside interface, for example a magic 8 ball.