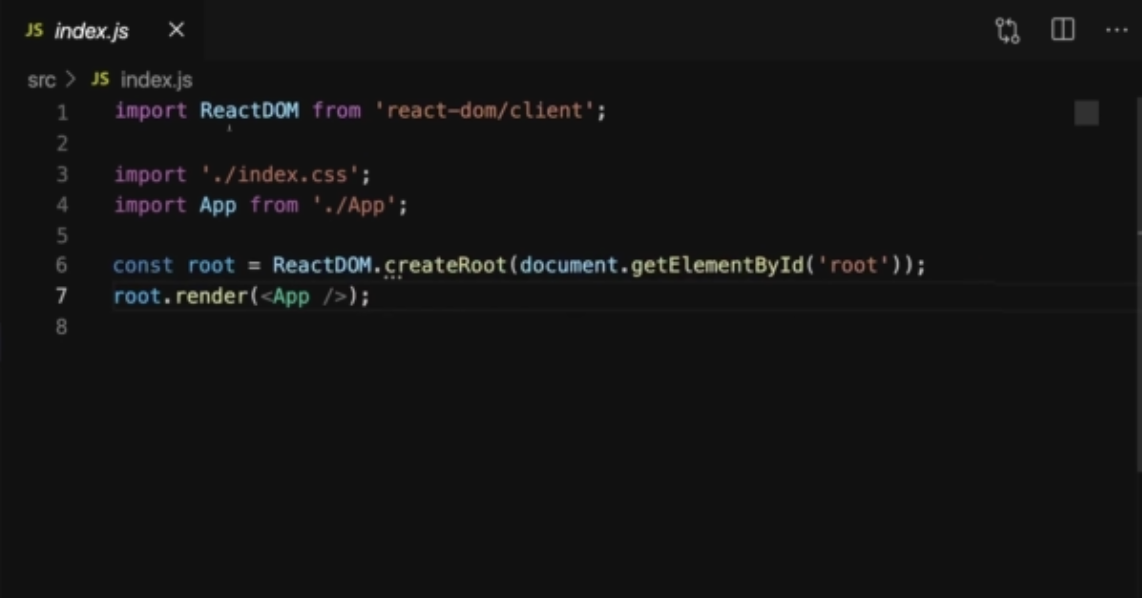
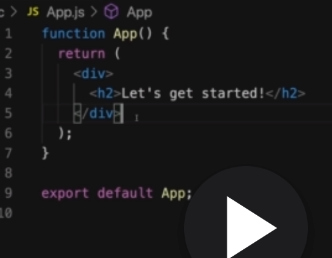
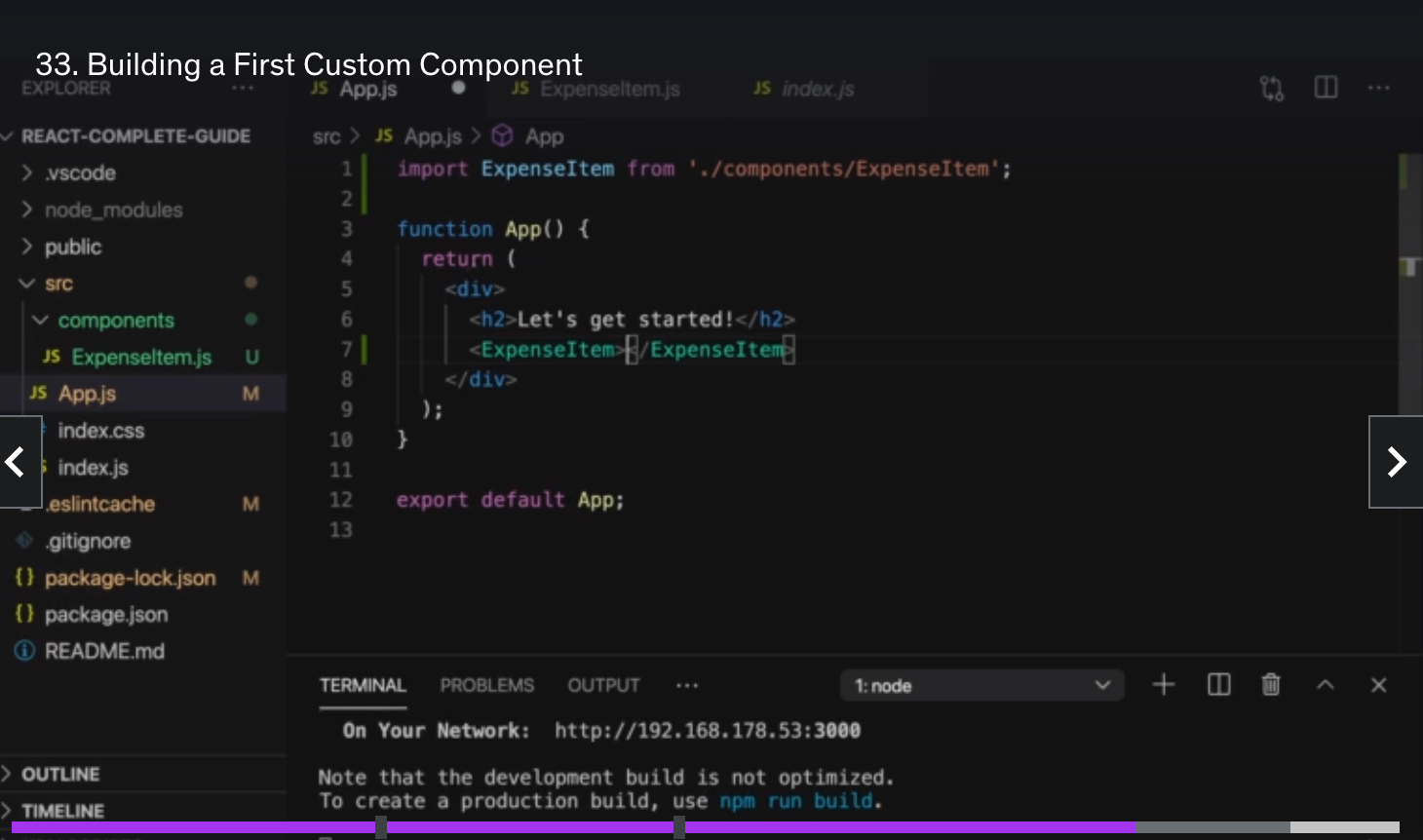
**React JS Notes:-**

1. Setting Up Course Dev Environment:- Classes in Javascript Classes can have properties and methods.
2. React uses virtual DOM . React uses component driven user interfaces.
3. React is a javascript library for building user interfaces. React is all about components.
4. All user interfaces are made of components.
5. Divide every page into small components.
6. React uses declarative approach for creating components.
7. Create a new react project :- Use create react app.
8. Command to create react app:- npx create-react-app my-app
9. cd my-app and then npm start to create react app.
10. First file that gets executed is index.js .
11. 

ReactDOM is a third party library imported from react.

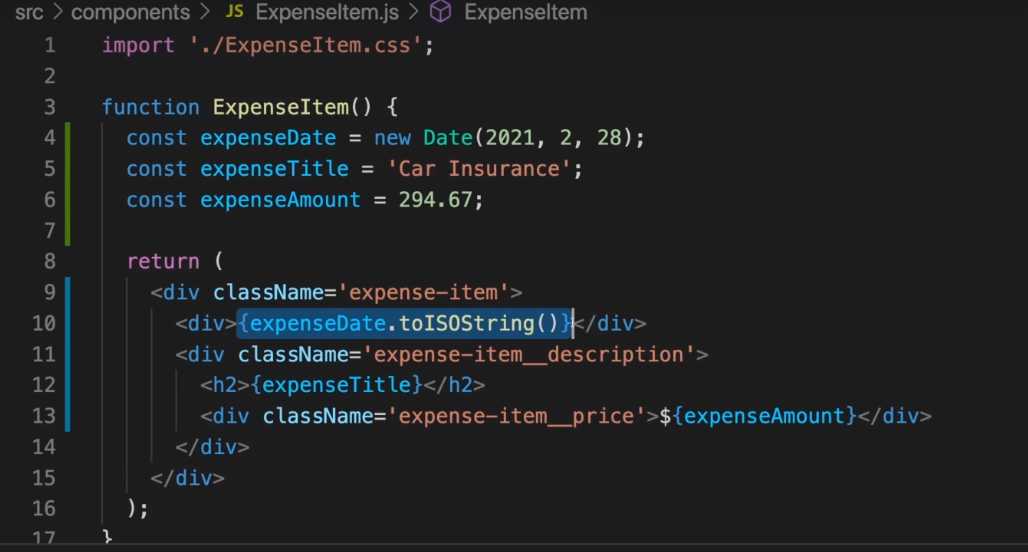
1. Index.html is the single page main HTML file that is rendered.
2. ReactDOM.createRoot() is the main root of the react application where all the components would be rendered.
3. Root.render(<App />) is for the main app component of the react application.



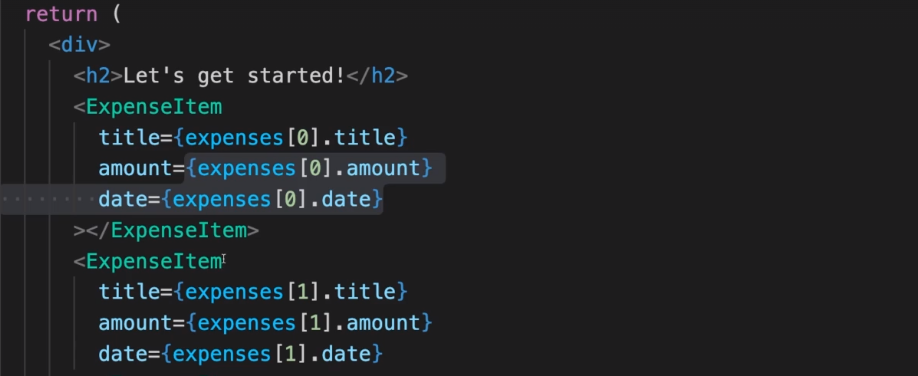
1. JSX stands for Javascript XML. React embraces.
2. App component is usually the root component. We build a a tree of components. Every component in react is a just a javascript function.
3. 

Lower case HTML elements are default elements and upper case HTML are custom elements.

1. We use className instead of class in react for assigning class.
2. We can use dynamic data for reusable components. This can be done with {} brackets.

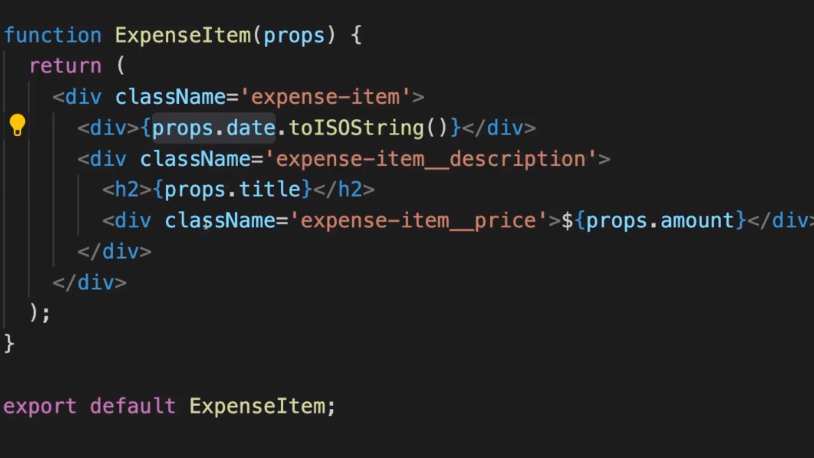


1. We can pass data via props as follows:- We can pass data to child components via props.



Data is passed in the form of props. In the child component.

Splitting components into smaller components is a good practice.

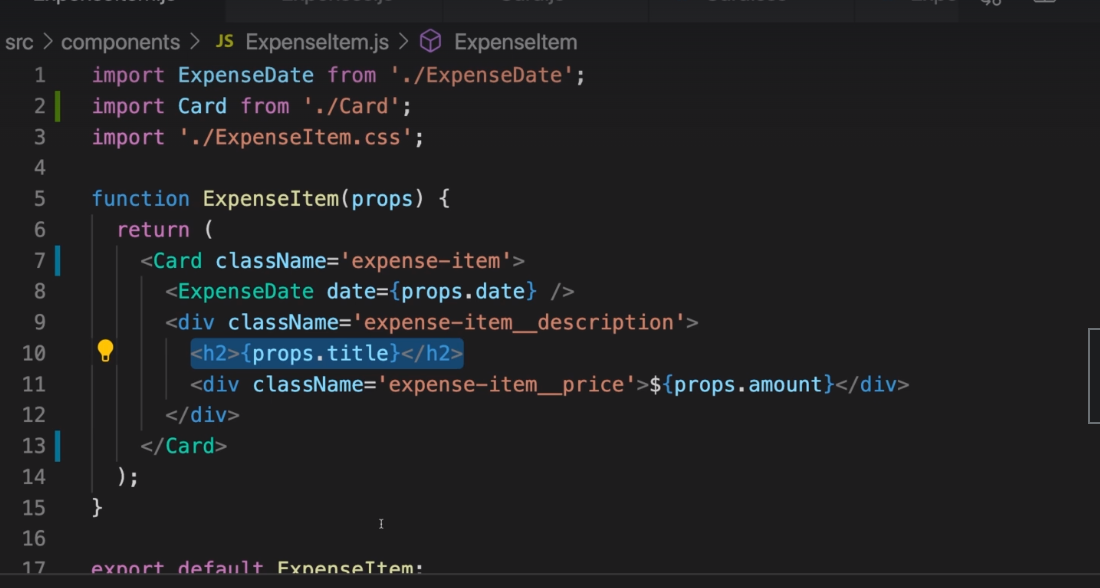


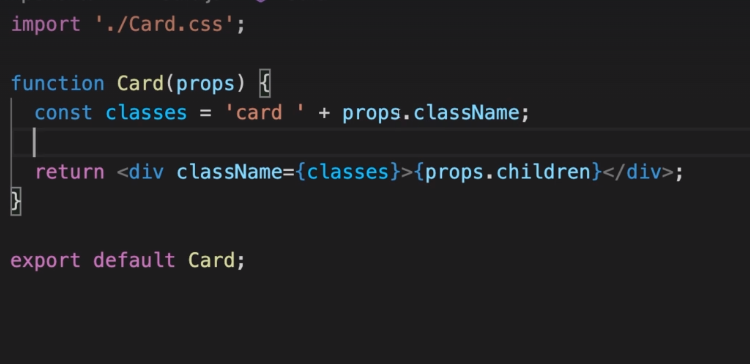
**Children Props:-**

We want create component that creates a shell for all the other components.

We want to pass the content between the opening and closing tags of the component.

This can be done via {props.children}. Wrapper components like cards can be created for reusability.





React and react-dom are dependencies in react package.json. **The react package holds the react source for components, state, props and all the code that is react.** **The react-dom package as the name implies is the glue between React and the DOM.**

Code can be organized based on the commonly used functionality.

Alternative function syntax.

const App = () => {

Return (

)

}