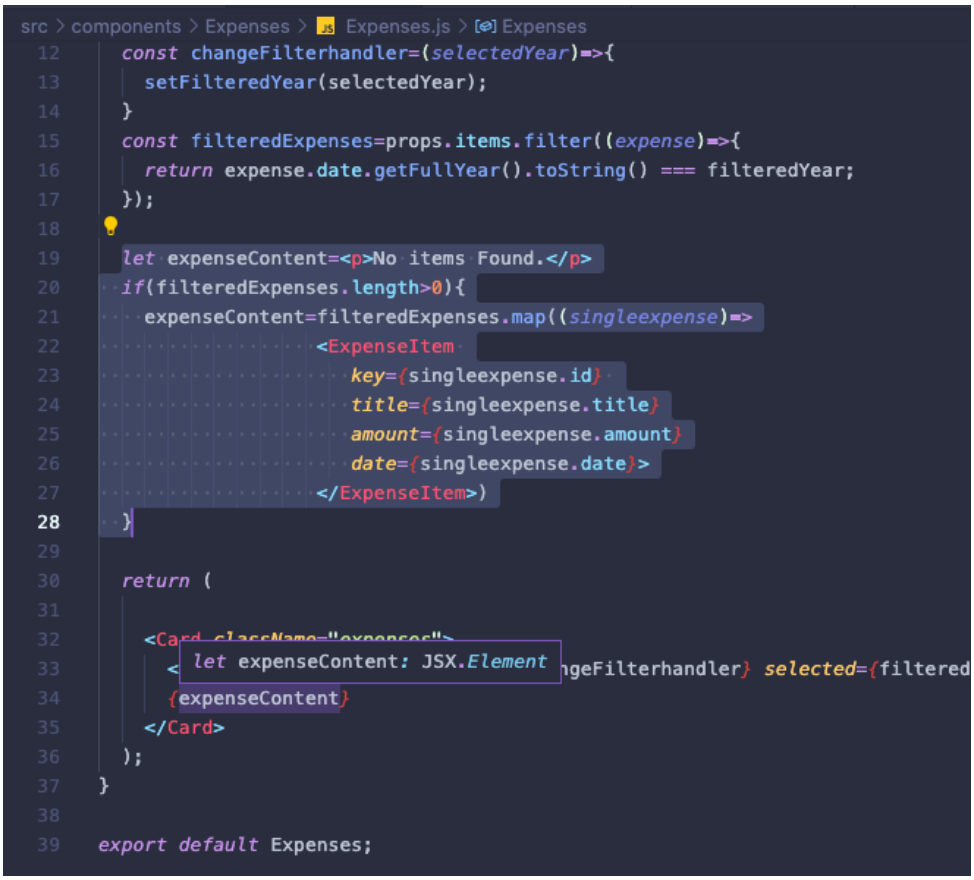
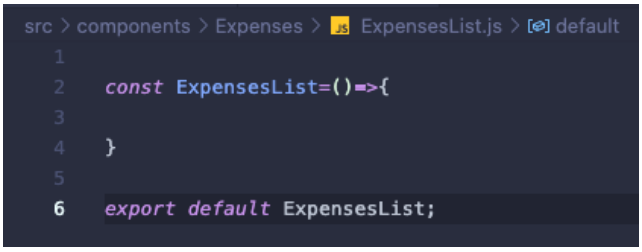
30. Conditional return statements

- In the previous lecture, we learned how to render conditional contents inside JSX code using a ternary operator and a trick
- Now suppose if our entire statement changes then we can use normal if-else condition
- In the previous lecture, only a particular component only changes due to the condition so we use ternary operator because it is inside the JSX code, but now the entire return statement is going to be different based on the condition so we can use if else block to return the entire JSX based on the condition
- In Expenses.js
  - 

  - This component is still big so let us create a new separate component
    - 

  - Now we are moving the lines 19-28 to ExpensesList.js and we will be modifying it

- Now in Expenses.js we are using the ExpensesList component by passing this prop

src > components > Expenses > Expenses.js > Expenses

```
8
9   const Expenses=(props)=> {
10    const [filteredYear,setFilteredYear]=useState('2021');
11
12    const changeFilterhandler=(selectedYear)=>{
13      setFilteredYear(selectedYear);
14    }
15    const filteredExpenses=props.items.filter((expense)=>{
16      return expense.date.getFullYear().toString() === filteredYear;
17    });
18
19
20
21    return (
22
23      <Card className="expenses">
24        <ExpenseFilter onChangeFilter={changeFilterhandler} selected={filteredYear} ></ExpenseFilter>
25        <ExpensesList items={filteredExpenses} ></ExpensesList>
26      </Card>
27    );
28  }
29
30  export default Expenses;
```

-
    - This is because we haven't moved the filtered list to that new component because we have the state variable in Expenses.js, so it must be here so we are passing it using the props to the ExpensesList component

src > components > Expenses > ExpensesList.js > ExpensesList

```
1   import ExpenseItem from "./ExpenseItem";
2
3
4   const ExpensesList=(props)=>{
5     if(props.items.length>0){
6       return(props.items.map((singleexpense)=>
7                       <ExpenseItem
8                         key={singleexpense.id}
9                         title={singleexpense.title}
10                        amount={singleexpense.amount}
11                        date={singleexpense.date}>
12                      </ExpenseItem>))
13      }
14      else{
15        return(
16          <p>No items Found.</p>
17        )
18      }
19  }
20
21  export default ExpensesList;
```

-

- Here we modified filteredList into props.items because here we don’t have filteredList instead we are getting it through props
- And here we are checking for the length and we are returning based on it
- This is called condition return statements