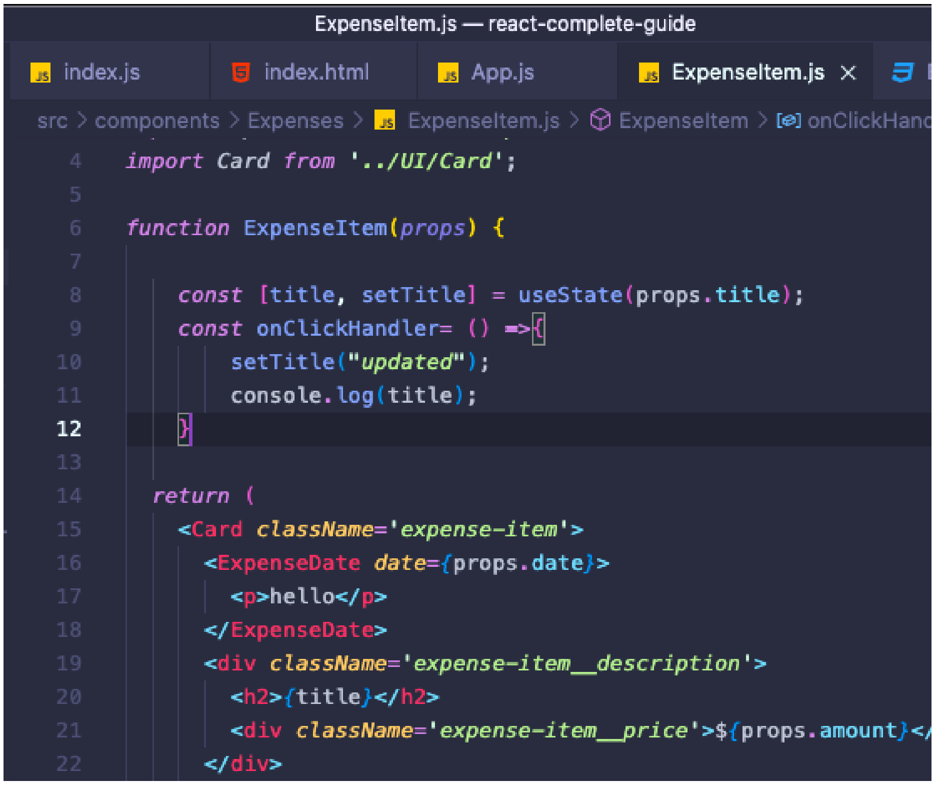
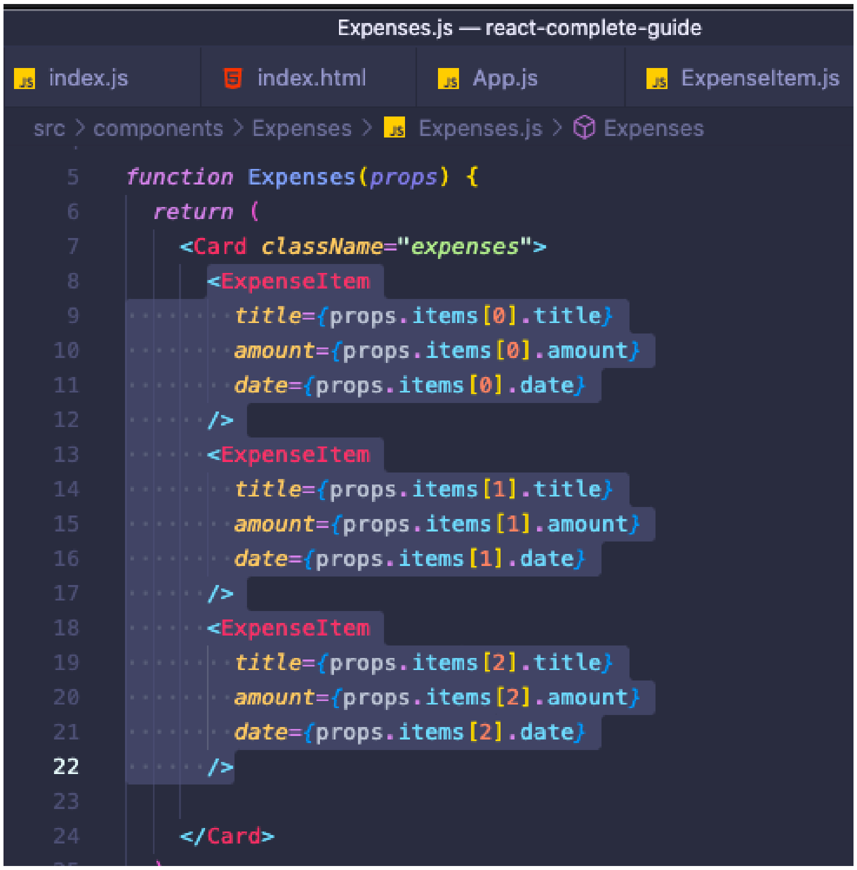
11. Closer look at useState

- In ExpenseItem we have used useState
    - 

ExpenseItem.js — react-complete-guide
index.js
index.html
App.js
ExpenseItem.js ×
src > components > Expenses > ExpenseItem.js > ExpenseItem > onClickHanc

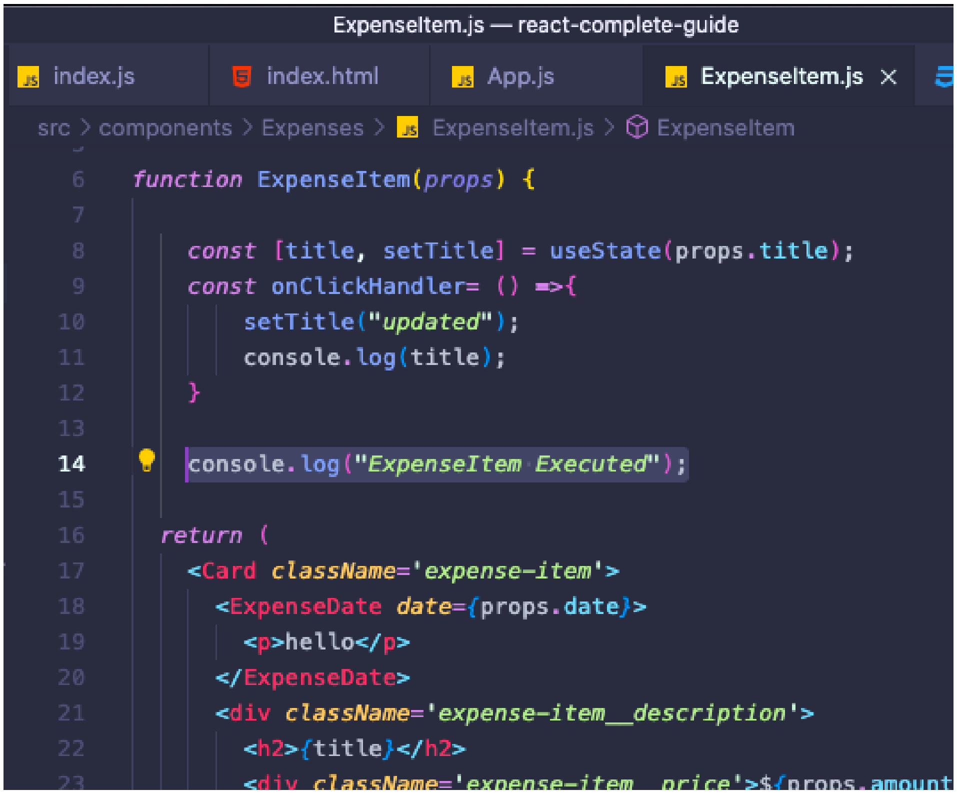
```
import Card from '../UI/Card';

function ExpenseItem(props) {

  const [title, setTitle] = useState(props.title);
  const onClickHandler= () =>{
    setTitle("updated");
    console.log(title);
  }

 return (
  <Card className='expense-item'>
   <ExpenseDate date={props.date}>
    <p>hello</p>
   </ExpenseDate>
   <div className='expense-item__description'>
    <h2>{title}</h2>
    <div className='expense-item__price'>${props.amount}</
   </div>
```

- We have called three ExpenseItem component

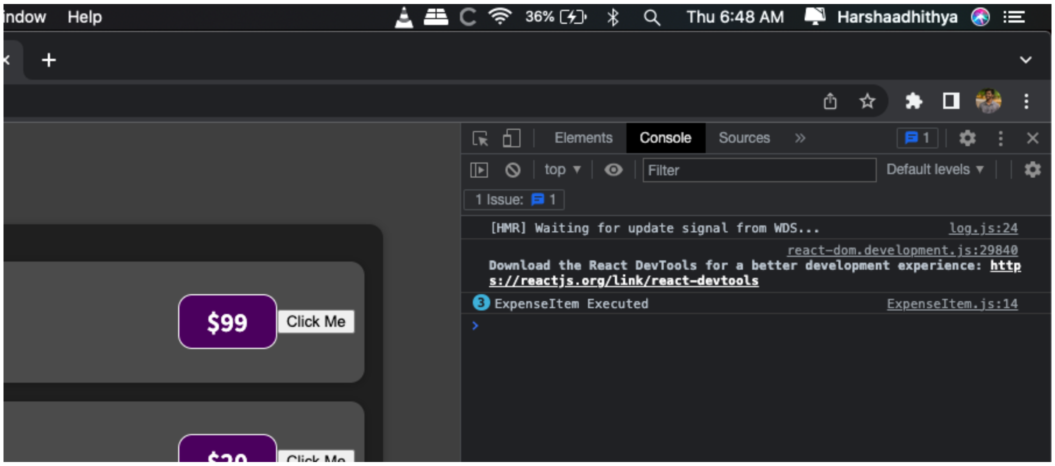
- ○ 
Expenses.js — react-complete-guide
index.js
index.html
App.js
ExpenseItem.js
src > components > Expenses > Expenses.js > Expenses

```
function Expenses(props) {
  return (
    <Card className="expenses">
      <ExpenseItem
        title={props.items[0].title}
        amount={props.items[0].amount}
        date={props.items[0].date}
      />
      <ExpenseItem
        title={props.items[1].title}
        amount={props.items[1].amount}
        date={props.items[1].date}
      />
      <ExpenseItem
        title={props.items[2].title}
        amount={props.items[2].amount}
        date={props.items[2].date}
      />

    </Card>
```

  - Here each of the component creates its own instance of ExpenseItem Component
  - And it also creates separate state variable for each instances.
- Now let us see this

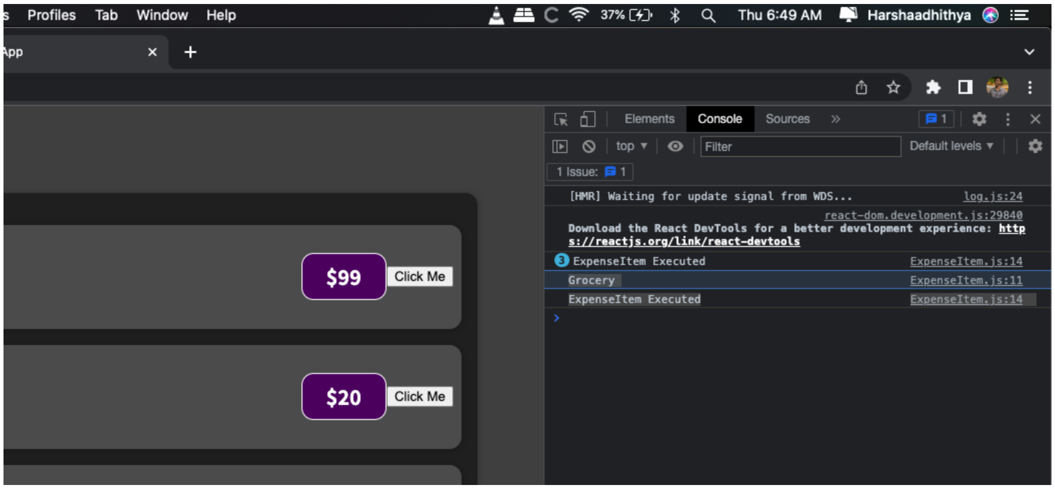
  - 
ExpenseItem.js — react-complete-guide
index.js
index.html
App.js
ExpenseItem.js
src > components > Expenses > ExpenseItem.js > ExpenseItem

```
function ExpenseItem(props) {

    const [title, setTitle] = useState(props.title);
    const onClickHandler= () =>{
      setTitle("updated");
      console.log(title);
    }

    console.log("ExpenseItem Executed");

  return (
    <Card className='expense-item'>
      <ExpenseDate date={props.date}>
        <p>hello</p>
      </ExpenseDate>
      <div className='expense-item__description'>
        <h2>{title}</h2>
        <div className='expense-item__price'>${props.amount
```

  - Here I have printed this line
  - 

```
[HMR] Waiting for update signal from WDS...                    log.js:24
                                          react-dom.development.js:29840
Download the React DevTools for a better development experience: http
s://reactjs.org/link/react-devtools
3 ExpenseItem Executed                                  ExpenseItem.js:14
>
```

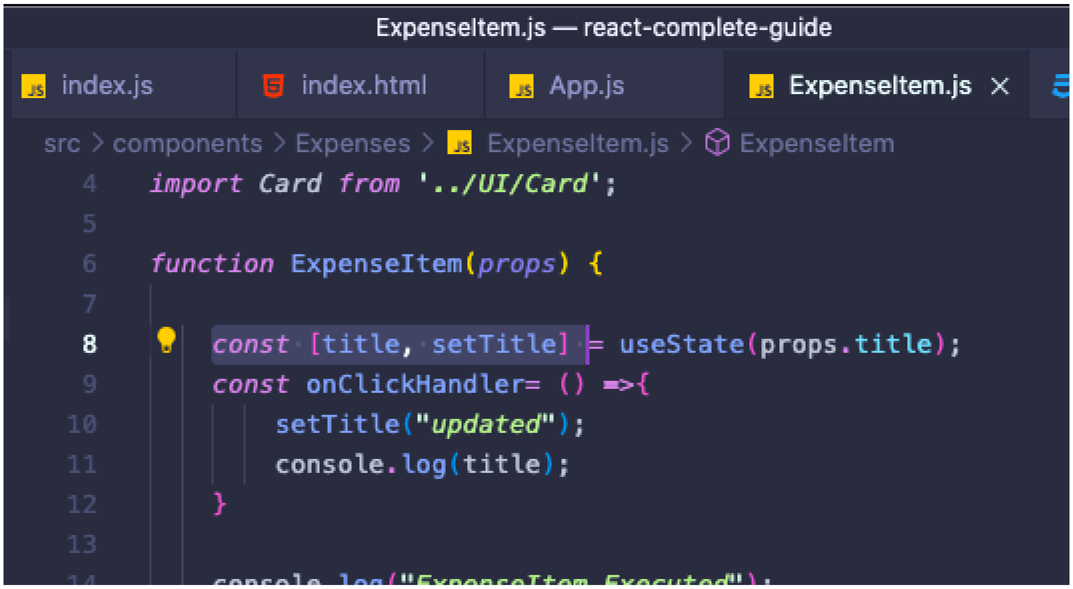
  - Whenever the page is loaded the three instance is executed and so there is three print statements there in console
- Now let me click the button to change the state of one instance

- 

```
[HMR] Waiting for update signal from WDS...                log.js:24
                                        react-dom.development.js:29840
Download the React DevTools for a better development experience: http
s://reactjs.org/link/react-devtools
3 ExpenseItem Executed                                ExpenseItem.js:14
Grocery                                               ExpenseItem.js:11
ExpenseItem Executed                                  ExpenseItem.js:14
>
```

    - ■ Now after clicking one button, the corresponding instance alone is re-evaluated and not other instances of the same component.
- Now you may ask whenever the particular component re-executes, then the title will also get assigned to the initial value of props.title, but useState is intelligent enough to get the latest snapshot that particular state variable, if there is no latest snapshot then only it will be assigned to the default value of props.title.

- —--------------------------


ExpenseItem.js — react-complete-guide
index.js
index.html
App.js
ExpenseItem.js
src > components > Expenses > ExpenseItem.js > ExpenseItem

```
import Card from '../UI/Card';

function ExpenseItem(props) {

    const [title, setTitle] = useState(props.title);
    const onClickHandler= () =>{
        setTitle("updated");
        console.log(title);
    }

    console.log("ExpenseItem Executed");
```

-
    - Why are we using const here? Even though we are updating new value
    - Because we are changing the value without assigning it directly, we are just using the setTitle function for updation.
    - And the useState(props.title), here the state variable title will be initialised with the default value (i.e props.title) only when the instance of the component is loaded for the very first time, whenever the state of title changes, then when we

re-execute the component, the title will have the latest snapshot of that particular state variable.