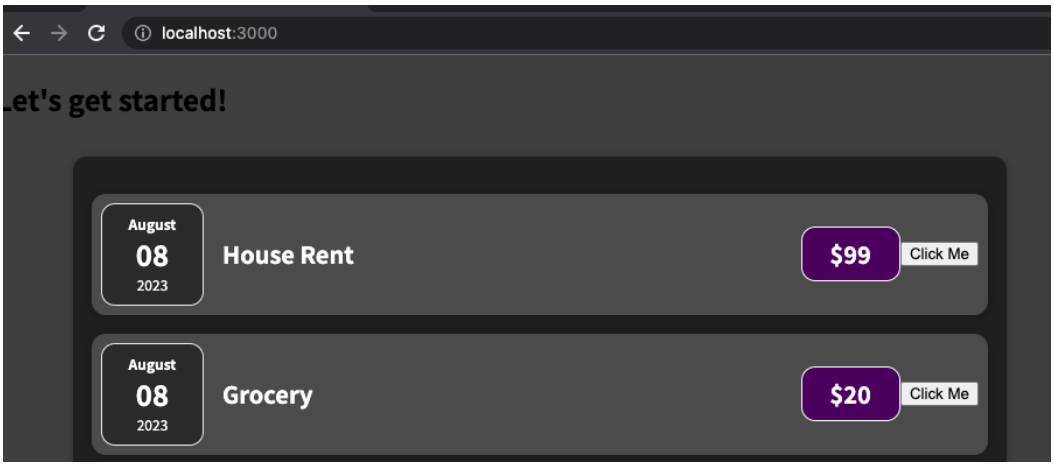
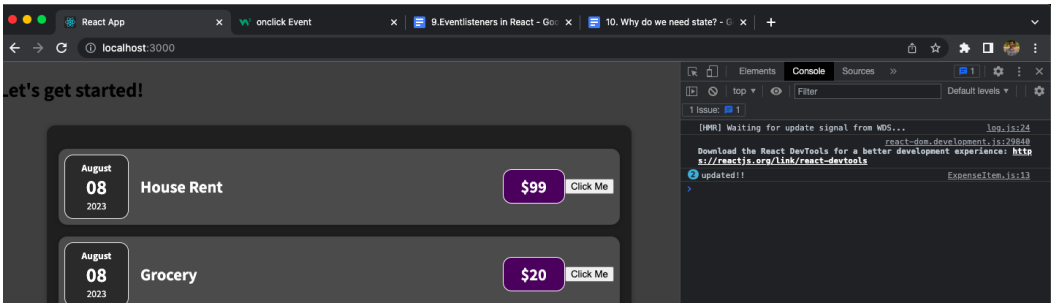
10. Why do we need state?

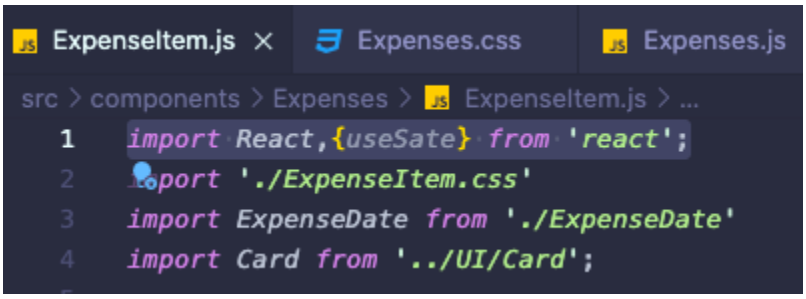
- Now let us try to change the name of the expense item when we click the button here
  - 

  - In expenseitem component, let us initialize a variable named title with props.title
    - 

    - And use this variable to display the title instead of directly using props.title

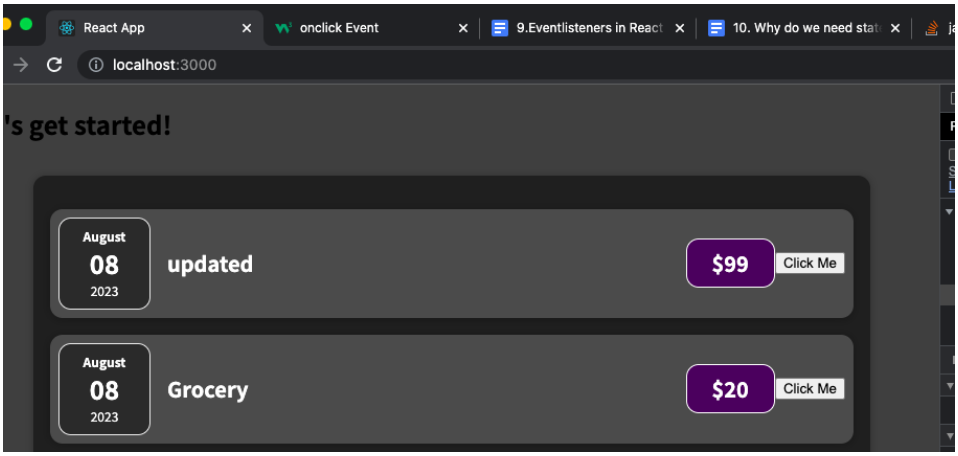
-
```
ExpenseItem.js × | Expenses.css | Expenses.js
src > components > Expenses > ExpenseItem.js > Expe
8   // const expenseAmount=250.00;
9   let title=props.title;
10  const onClickHandler= () =>{
11      // console.log("hello");
12  }
13
14  return (
15    <Card className='expense-item'>
16      <ExpenseDate date={props.date}>
17        <p>hello</p>
18      </ExpenseDate>
19      <div className='expense-item__descri
20        <h2>{title}</h2>
21        <div className='expense-item__pric
22      </div>
23      <button onClick={onClickHandler} >C
24    </Card>
25  );
```
- Now onclickhanler let us try to change the title and see what happens
  -
```
ExpenseItem.js × | Expenses.css | Expenses.js
src > components > Expenses > ExpenseItem.js > Expenselt
8   // const expenseAmount=250.00;
9   let title=props.title;
10  const onClickHandler= () =>{
11      // console.log("hello");
12      title="updated!!";
13      console.log(title);
14  }
15
16  return (
17    <Card className='expense-item'>
18      <ExpenseDate date={props.date}>
19        <p>hello</p>
20      </ExpenseDate>
21      <div className='expense-item__descriptio
22        <h2>{title}</h2>
23        <div className='expense-item__price'>
24      </div>
25      <button onClick={onClickHandler} >Click
26    </Card>
27  );
```
- Now when we clickthe button

- 

  - The title is not changed but the function has logged the updated title in console but not reflecting in our page
- This is because that, our react components are functions, which is called in a hierarchical manner
  - App ->Expenses -> ExpenseItem -> ExpenseDate
  - Here initially the app component is called , which calls Expenses component which calls ExpenseItem components and in which each ExpenseItem component calls its corresponding ExpenseDate component.
  - In our case, initially when the expenseitem component is called, its title is House Rent and now if we click the button the handler function just updated the title variable but the function or component is not refreshed or called again so it just displays the old title which was there when the component is called first.
  - Even if the function is called again, then it again initialises the title variable to props.title which again has no change
  - So the component needs to re-evaluated
- So to re-evaluate or refresh the componen whenever the title is updated, then we need to use state variables
- Now let us begin
  - 

    - Import useSate

-
```
ExpenseItem.js | Expenses.css | Expenses.js | Expense
src > components > Expenses > ExpenseItem.js > ExpenseItem
5
6   function ExpenseItem(props) {
7       // const expenseDate=new Date(2023, 7, 30);
8       // const expenseItem='House Rent';
9       // const expenseAmount=250.00;
10      // let title=props.title;
11      const [title, setTitle] = useSate(props.title);
12      const onClickHandler= () =>{
13          // console.log("hello");
14          // title="updated!!";
15          console.log(title);
16      }
17
```
  - The usestate function returns a array with two elements one is the actual variable and the other is the name of the function that is used to change that variable
  - So we are destructuring the array and while calling useSate, we are passing in the props.title which will be assigned initially to title variable
  -
```
ExpenseItem.js | Expenses.css | Expenses.js | Expe
src > components > Expenses > ExpenseItem.js > ExpenseItem > [e] o
1   import React,{useState} from 'react';
2   import './ExpenseItem.css'
3   import ExpenseDate from './ExpenseDate'
4   import Card from '../UI/Card';
5
6   function ExpenseItem(props) {
7       // const expenseDate=new Date(2023, 7, 30);
8       // const expenseItem='House Rent';
9       // const expenseAmount=250.00;
10      // let title=props.title;
11      const [title, setTitle] = useState(props.title);
12      const onClickHandler= () =>{
13          // console.log("hello");
14          // title="updated!!";
15          setTitle("updated");
16          console.log(title);
17      }
18
19    return (
20      <Card className='expense-item'>
21        <ExpenseDate date={props.date}>
```
  - Now in handler function we are just using the function given by use state to change the value
- Output

- 

  - Now it is updated successfully
- Conclusion:
  - Usestate is one of the hooks provided by react library
  - It keeps the values globally and whenever its value is changed then it re executes the function/component inside where we have that usesate variable, so that the updated value is displayed because the component is re-evaluated