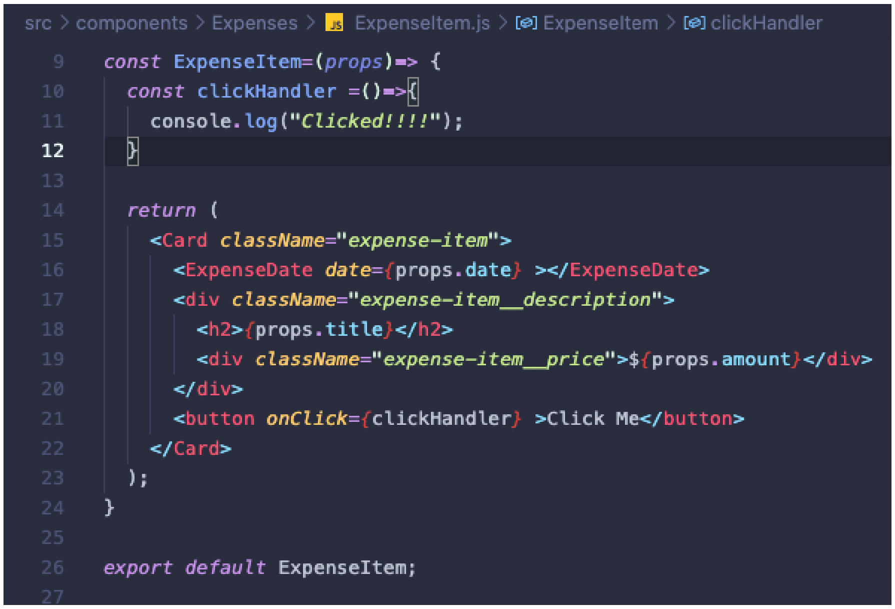
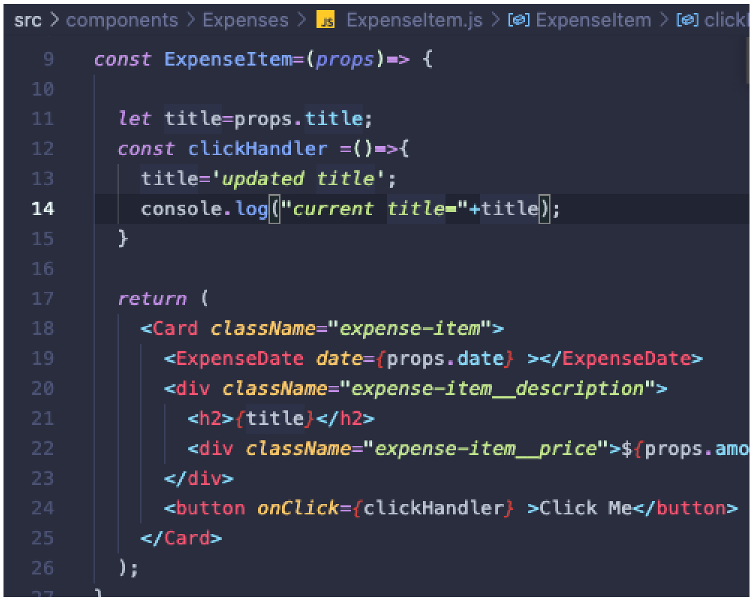
15.state(hook)-part 1

- In the previous lecture we added an eventlistener for button click
- On that event we triggered a function to display something on console log
- Now we have to use this eventlistener to change something on our webpage

INITIAL CODE:

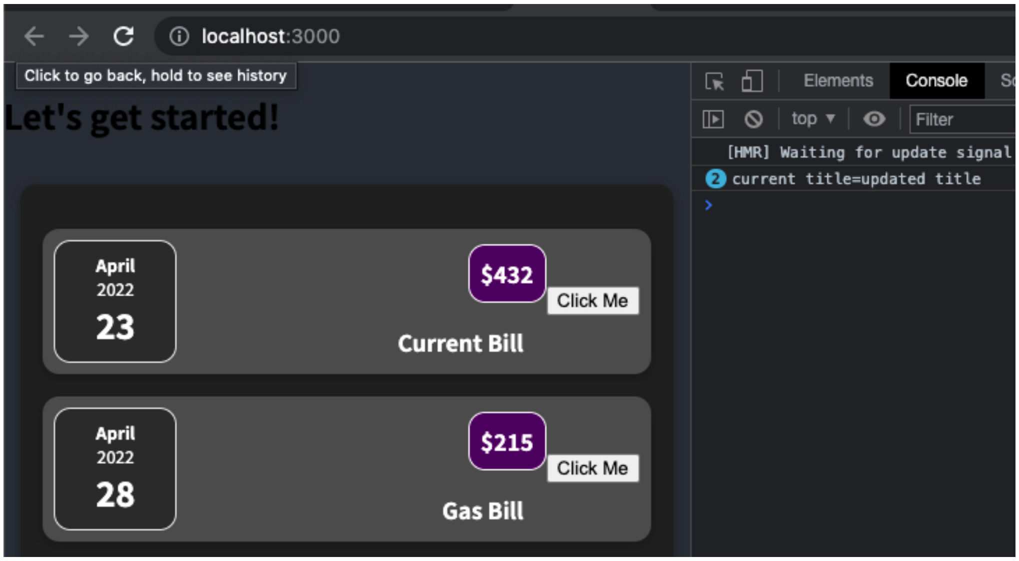
- 
src > components > Expenses > ExpenseItem.js > ExpenseItem > clickHandler

```
const ExpenseItem=(props)=> {
  const clickHandler =()=>{
    console.log("Clicked!!!!");
  }

  return (
    <Card className="expense-item">
      <ExpenseDate date={props.date} ></ExpenseDate>
      <div className="expense-item__description">
        <h2>{props.title}</h2>
        <div className="expense-item__price">${props.amount}</div>
      </div>
      <button onClick={clickHandler} >Click Me</button>
    </Card>
  );
}

export default ExpenseItem;
```

- Let us change the title of the expense on click
- 
src > components > Expenses > ExpenseItem.js > ExpenseItem > click

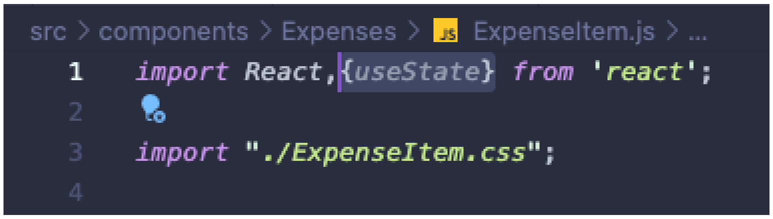
```
const ExpenseItem=(props)=> {

  let title=props.title;
  const clickHandler =()=>{
    title='updated title';
    console.log("current title="+title);
  }

  return (
    <Card className="expense-item">
      <ExpenseDate date={props.date} ></ExpenseDate>
      <div className="expense-item__description">
        <h2>{title}</h2>
        <div className="expense-item__price">${props.amo
      </div>
      <button onClick={clickHandler} >Click Me</button>
    </Card>
  );
}
```

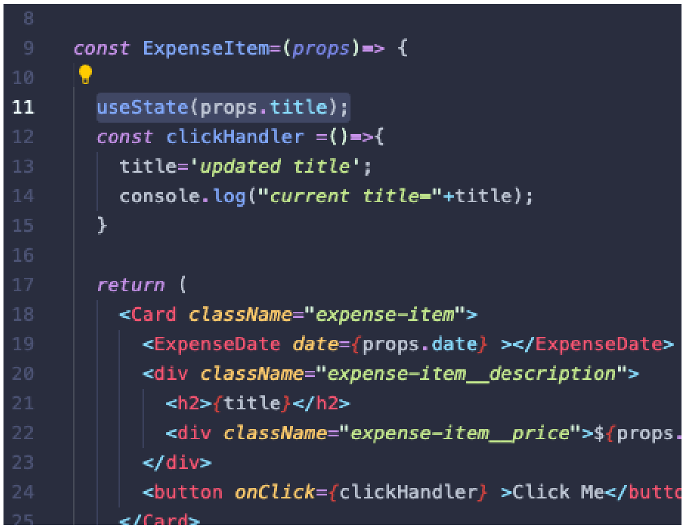
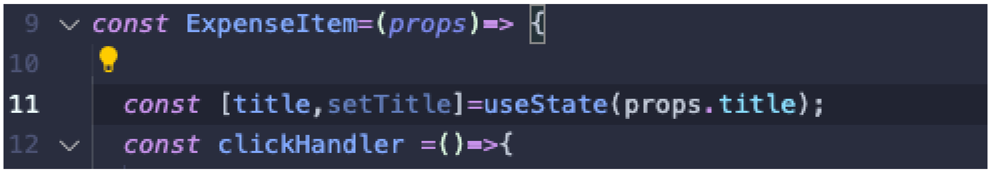
  - Now what do you think? will it change or not
  - And the answer is, yes it changes BUT it will not be reflected in our page
  - This is because:

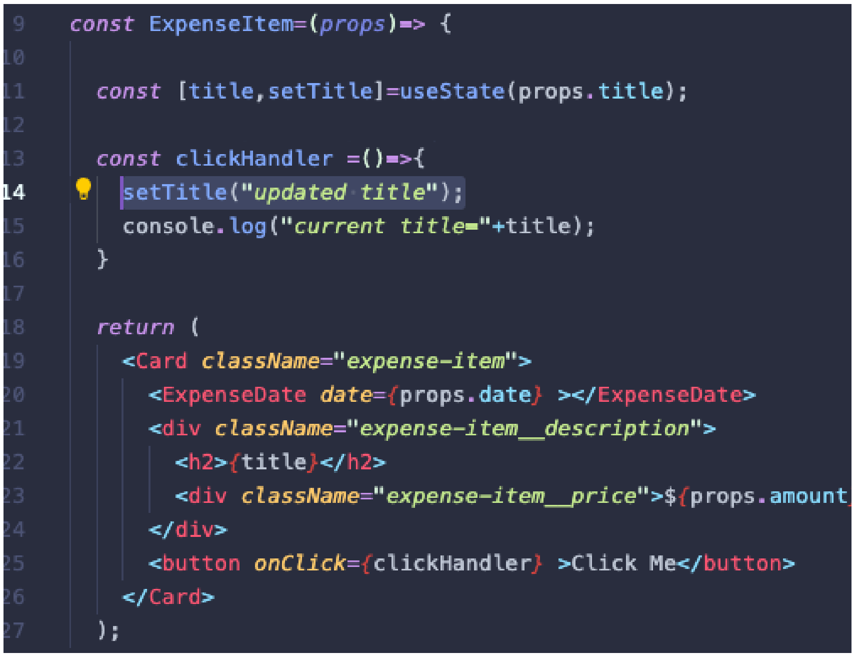
        - Firstly the app component calls the expenses component, then the expenses component calls the expenseitems component expenseitem component, which will be rendered and displayed
        - Initially in expenseitem component, the button doesnot trigger the click handler function because initially while rendering, the event of click can’t be occurred, so the clickhandler function will not be triggered initially
        - Now the webpage is displayed, now when we click the button it triggers the clickhandler function which will do the tasks specified in it(here changing the title) but after doing that it won’t again call the expenseitem component so that the new updated part of that component will not be evaluated
        - But when we print the title in console it gives us the updated title but it is not reflected in our page because our expense item component is evaluated only once when we open our page
            - 

```
[HMR] Waiting for update signal
2 current title=updated title
```

            - We could see there that in the console it is reflected but not in our page
    - We can change this by using state hooks
        - This tells react that, something changed so that certain component should be re-evaluated
        - So if there is any changes in components, then that particular component will be re-evaluated and it will be rendered
        - In react, the normal variables(in previous code screenshot) line 11 we have a normal variable, when it changes,it will not tell react to re-evaluate that component,but a state variable can do

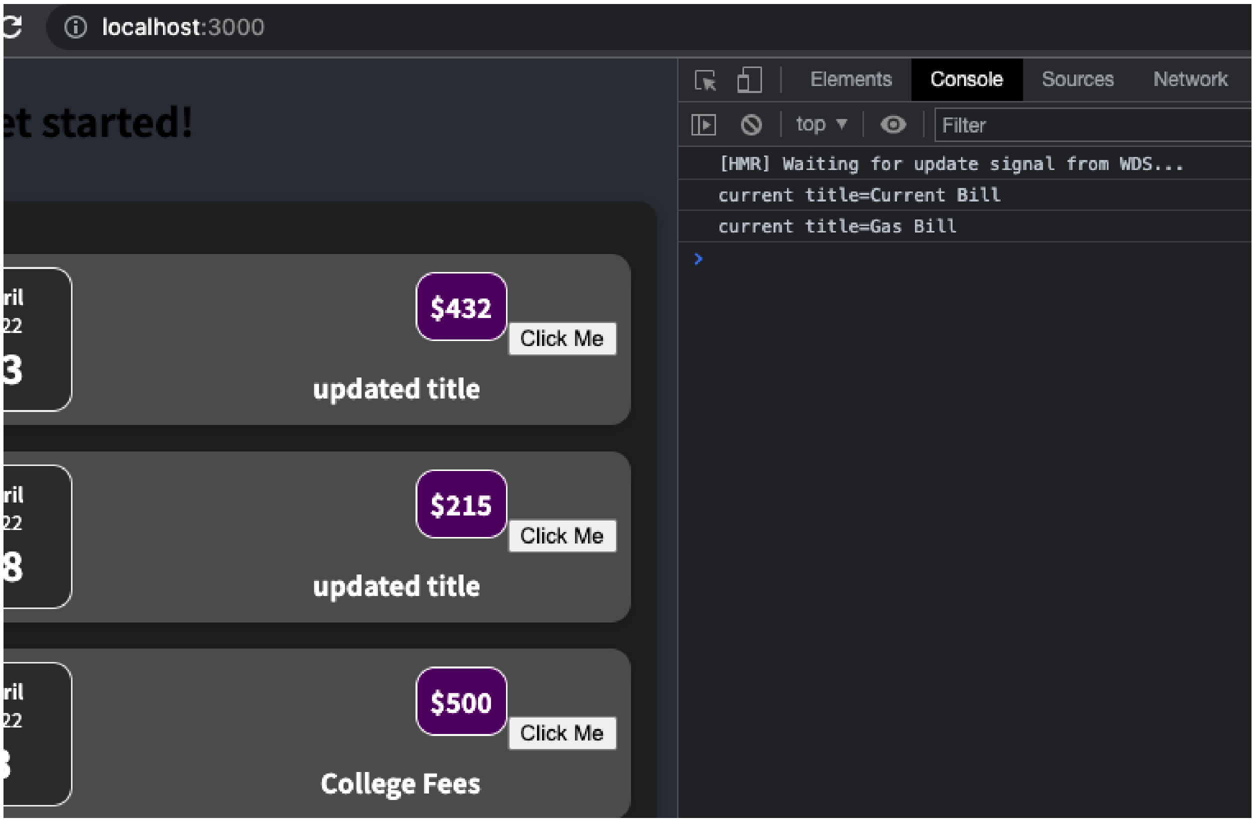
- To use that import:
    - 

      ```
      import React,{useState} from 'react';

      import "./ExpenseItem.css";

      ```
    - Here we have used curly braces because we want to import a specific piece from React library.
- useState is a method in React library
- Now let us create a state variable
    - 
      ```

      const ExpenseItem=(props)=> {

        useState(props.title);
        const clickHandler =()=>{
          title='updated title';
          console.log("current title="+title);
        }

        return (
          <Card className="expense-item">
            <ExpenseDate date={props.date} ></ExpenseDate>
            <div className="expense-item__description">
              <h2>{title}</h2>
              <div className="expense-item__price">${props.
            </div>
            <button onClick={clickHandler} >Click Me</butto
          </Card>
      ```
    - Here a new state variable is created which has the value of props.title
    - **useState method should be used only inside the component function and not inside anyother nested function and also not outside component's function**
- **The useState method return an array of 2 elements, one is a state variable which has the current state value of props.title and the other is an function which should be used to change the value of that state variable**
    - 
      ```
      const ExpenseItem=(props)=> {

        const [title,setTitle]=useState(props.title);
        const clickHandler =()=>{
      ```
    - We have used array destructuring to extract the method and variable from the returned array
    - You may ask, we are going to change the title but how can you define it as an const, it will answered later

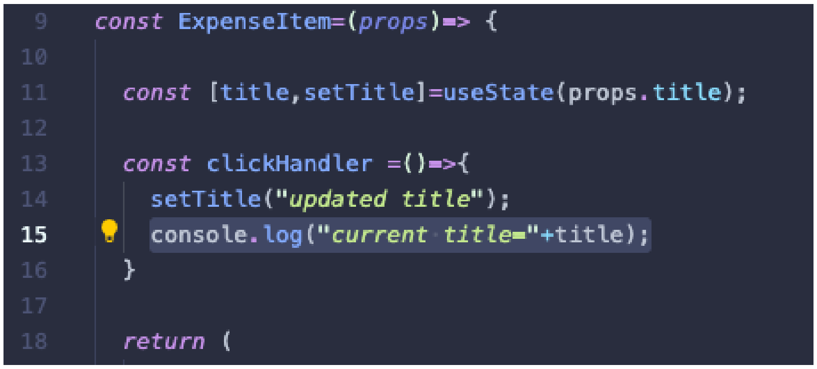
- Now let us use the setTitle method inside the click handler function to change the title
    - 

```
const ExpenseItem=(props)=> {

  const [title,setTitle]=useState(props.title);

  const clickHandler =()=>{
    setTitle("updated title");
    console.log("current title="+title);
  }

  return (
    <Card className="expense-item">
      <ExpenseDate date={props.date} ></ExpenseDate>
      <div className="expense-item__description">
        <h2>{title}</h2>
        <div className="expense-item__price">${props.amount
      </div>
      <button onClick={clickHandler} >Click Me</button>
    </Card>
  );
```

    - Here setTitle is replaced in place of manually changing the title value as we did previously
    - setTitle takes in an argument, it is the the new value for that state variable ‘title’
    - It has two tasks
        - 1. Changing the value of title to a new value
        - 2. Re-evaluating the particular ExpenseItem component so that the changes made here will be updated in the webpage because we are re-evaluating that component again and again it will be rendered

- 

```
[HMR] Waiting for update signal from WDS...
current title=Current Bill
current title=Gas Bill
```

  - If you notice here when we click the button,it changes the title and rednders in our page but in the console,it shows the value before updating
  - 

```
const ExpenseItem=(props)=> {

  const [title,setTitle]=useState(props.title);

  const clickHandler =()=>{
    setTitle("updated title");
    console.log("current title="+title);
  }

  return (
```

    - Eventhough consol.log line is after the setTitle function it prints the old name in the console because the setTitle function doesn’t update the value right away, it just schedules the update of that state variable

CONCLUSION: when an state variable is change it tells react to re-evaluate the component where the change occurred