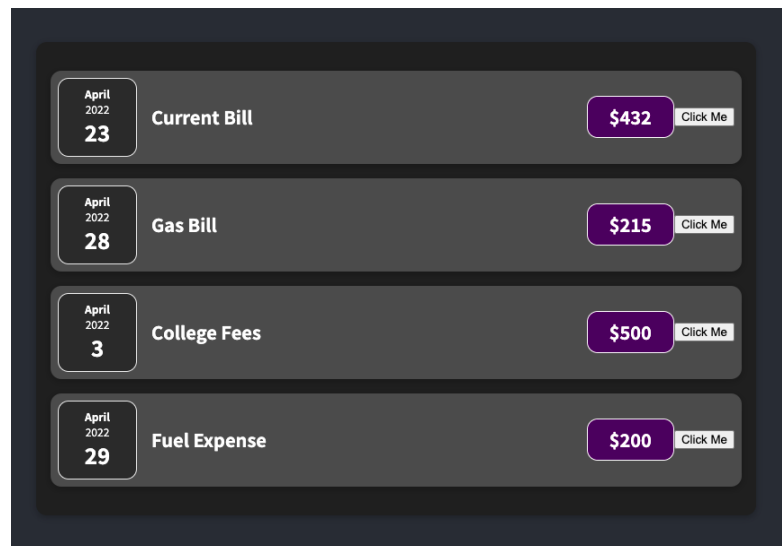


14.Event Handlers in React

- Let us add a button here

```
src > components > Expenses > ExpenseItem.js > [⌕] ExpenseItem.js
7  const ExpenseItem=(props)=> {
8
9    return (
10     <Card className="expense-item">
11       <ExpenseDate date={props.date} ></ExpenseDate>
12       <div className="expense-item__description">
13         <h2>{props.title}</h2>
14         <div className="expense-item__price">
15           {props.price}</div>
16         <button>Click Me</button>
17       </div>
18     </Card>
19   );
20 }
21 export default ExpenseItem;
```



- React supports every dom listeners that are available in vanilla javascript

- <https://developer.mozilla.org/en-US/docs/Web/API/Element>
- Here it listed all the events

▼ Events

```

afterscriptexecute
auxclick
blur
click
compositionend
compositionstart
compositionupdate
contextmenu
copy
cut
dblclick

```

- This is how vanilla js does

```

7  const ExpenseItem = (props) => {
8    document.getElementById('root').addEventListener()
9    return (

```

- Inside the () we could specify which event to be listened for eg(click,blur,etc..)
- But we have react equivalent for all the default events
 - Instead of typing document.getEl.....addEvent...ener(click) we could prefix use onClick,onBlur
 - Here 'on' is prefixed with event we want it to be listened

```

14      <div className="expense-item__price">${props
15    </div>
16    <button onClick={} >Click Me</button>
17  </Card>
18  );
19  }

```

- This onClick takes in a function and when the event happens it will execute or call the function that we have mentioned inside it

```

9  const ExpenseItem=(props)=> {
10    const clickHandler =()=>{
11      console.log("Clicked!!!!");
12    }
13
14    return (
15      <Card className="expense-item">
16        <ExpenseDate date={props.date} ></ExpenseDate>
17        <div className="expense-item__description">
18          <h2>{props.title}</h2>
19          <div className="expense-item__price">${props.price}</div>
20        </div>
21        <button onClick={clickHandler} >Click Me</button>
22      </Card>
23    );
24  }
25
26  export default ExpenseItem;

```

- Here we can use both normal function and arrow function, it is upto the developer
- In line 21, we have specified the function name
- NOTE: in line 21 we have specified just the function name, we didn't call it using () parentheses because we want that function to be executed only when the event occurs, if we specify the () then when the compiler compiles the code, it will call the function when that line gets compiled but we don't want that, we just need to specify which function that we want to be executed then the onClick attribute takes care of calling whenever the event occurs

