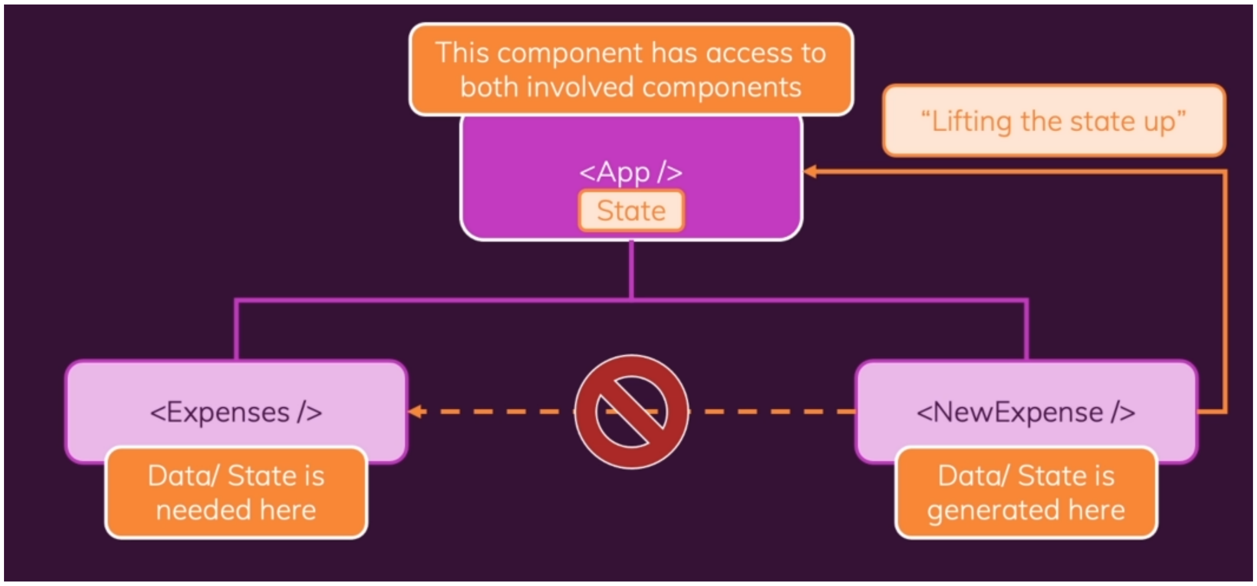
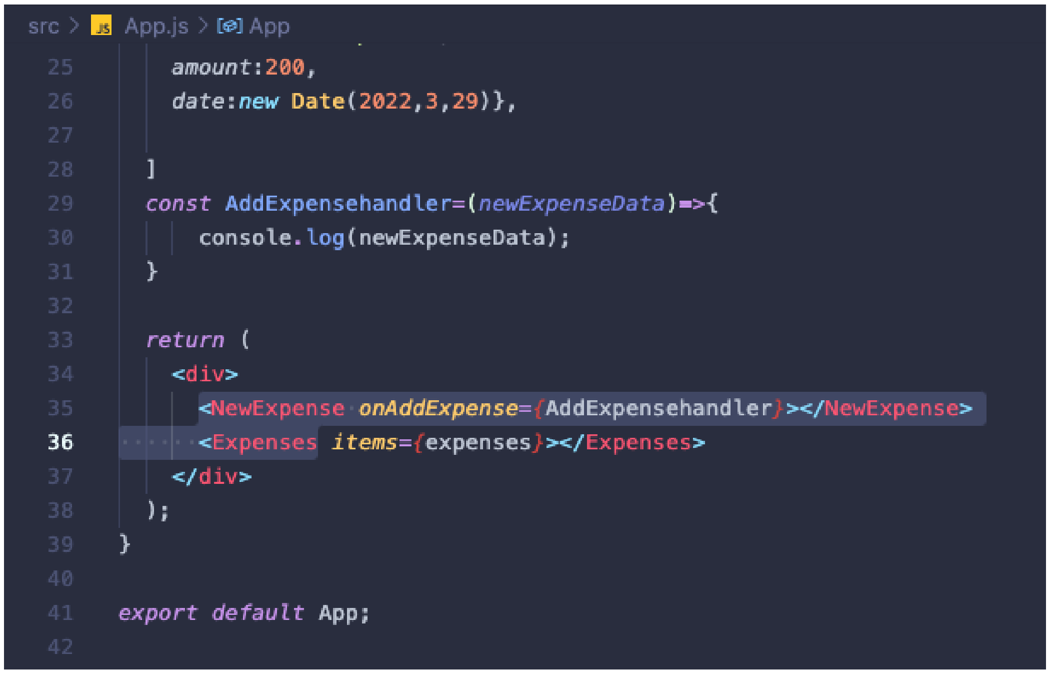
24. Lifting the state up

- Until now, we know how to pass data/state from parent to child and child to parent
- How can we pass data from one component to another component which has no direct relationship with the sending component(i.e no parent child relationship)
- Then a this case, we can consider it both as a sibling
- Here we want to find the component which has direct relationship with both the components
- Here we could see that Expenses and NewExpense has no direct access, but through app we can communicate with Expense from Newexpense and vice versa
- 
This component has access to both involved components
"Lifting the state up"
<App />
State
<Expenses />
Data/ State is needed here
<NewExpense />
Data/ State is generated here
- 

```
src > App.js > App
        amount:200,
        date:new Date(2022,3,29)},

    ]
    const AddExpensehandler=(newExpenseData)=>{
        console.log(newExpenseData);
    }

    return (
      <div>
        <NewExpense onAddExpense={AddExpensehandler}></NewExpense>
        <Expenses items={expenses}></Expenses>
      </div>
    );
  }

  export default App;

```

    - Here in app.js we are getting the newly added data in newexpense.js and later in the course section we are going to append it with the list of expenses taht are already existing
    - And now if you see it, we are passing down the expenses list to Expenses.js
    - This is how we can communicate between two siblings components.

Now what is the tem ‘Lifting the state up”?

- Here in our ExpenseForm we have lifted our state(enteredExpenseData) to newExpense Component and then again we are lifting it further to the app component which has access to the target component where we need that enteredData and now we are passing down the received state to the Expenses component.
- In React, sharing state is accomplished by moving it up to the closest common ancestor of the components that need it. This is called “lifting state up”.
- Here the closest ancestor is our app.js which has direct access to both the components(ExpenseForm and Expenses)