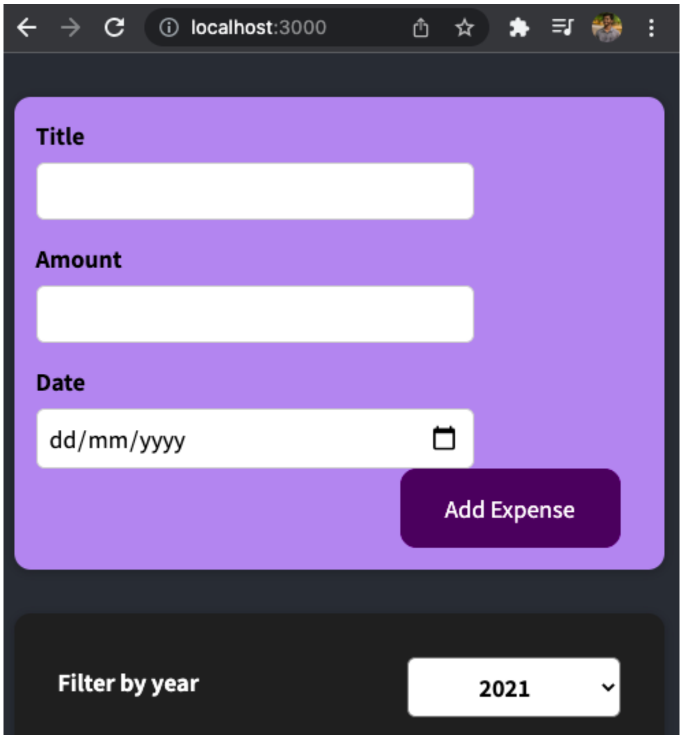
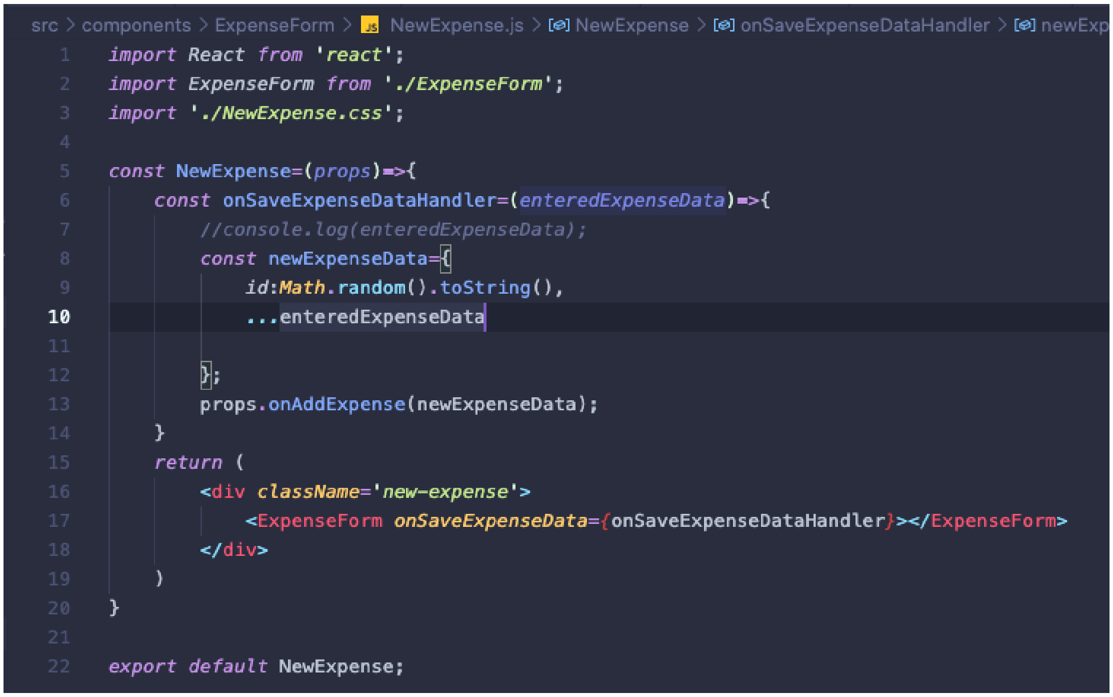
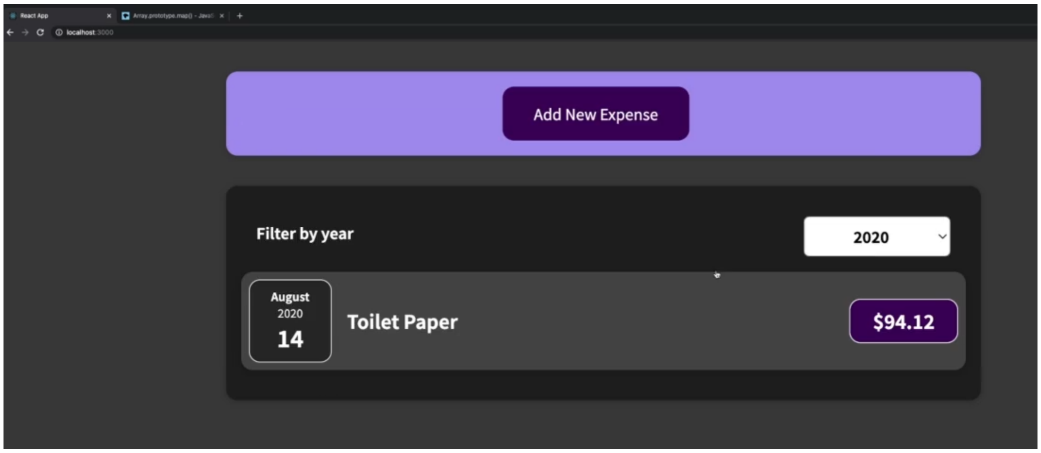
31. Showing form only when user clicks add expense

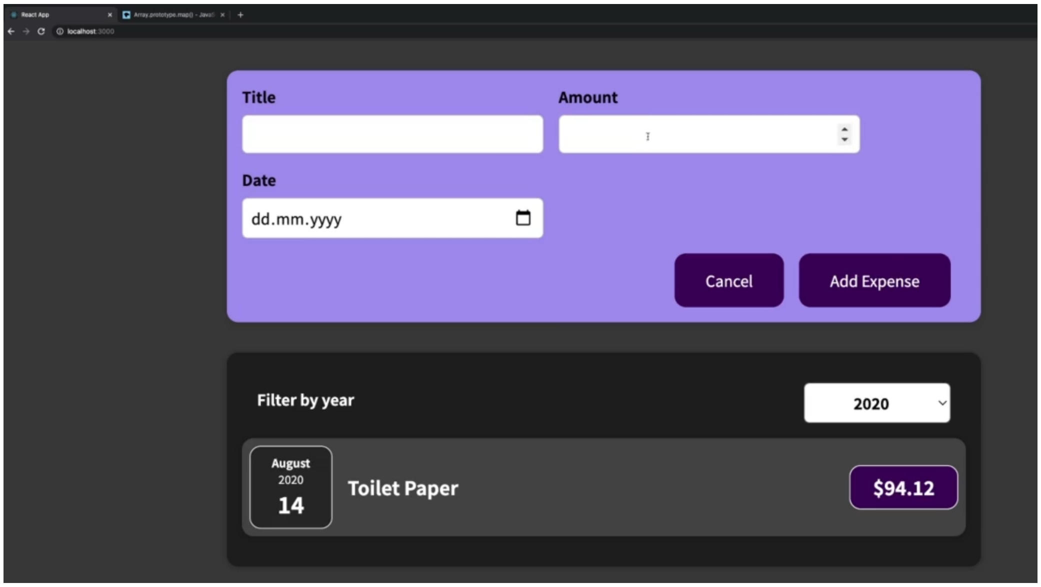
- Initially
    - 

- Initially NewExpenses.js
    - 

src > components > ExpenseForm > NewExpense.js > NewExpense > onSaveExpenseDataHandler > newExp

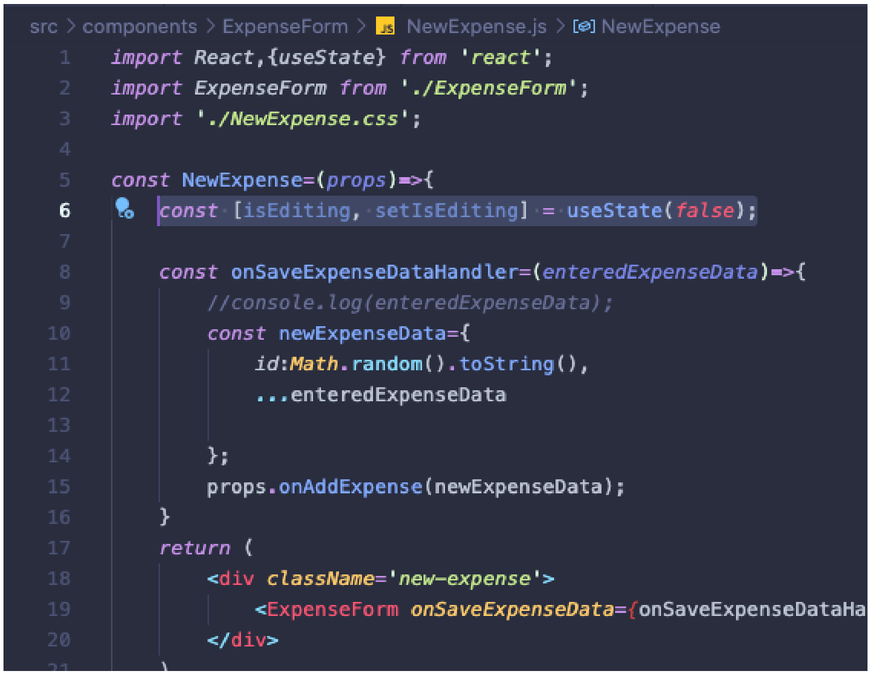
```
import React from 'react';
import ExpenseForm from './ExpenseForm';
import './NewExpense.css';

const NewExpense=(props)=>{
    const onSaveExpenseDataHandler=(enteredExpenseData)=>{
        //console.log(enteredExpenseData);
        const newExpenseData={
            id:Math.random().toString(),
            ...enteredExpenseData

        };
        props.onAddExpense(newExpenseData);
    }
    return (
        <div className='new-expense'>
            <ExpenseForm onSaveExpenseData={onSaveExpenseDataHandler}></ExpenseForm>
        </div>
    )
}

export default NewExpense;
```

- Aim:
  - 

  - When we click this Add new Expense button then it should display like this in the below picture
  - 

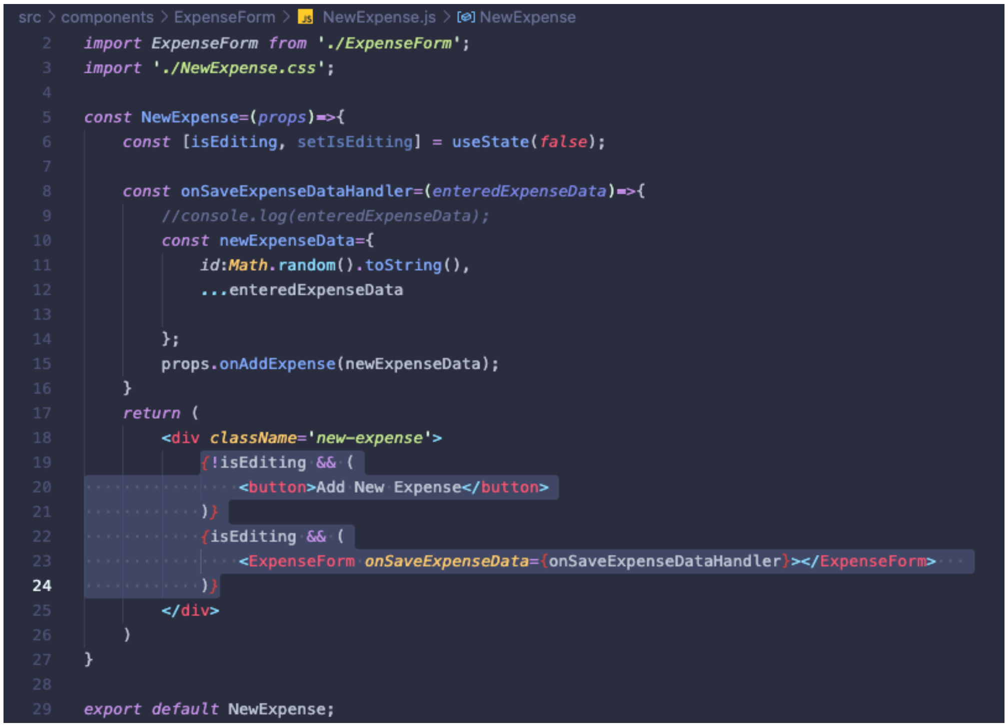
- Now compare this with our initial setup
  - First of all, we need to add the cancel button along with Add Expense
  - Now we need to hide this and show only when a button is clicked
  - So let us have a state variable which will have a boolean value and its value should be true when the button is clicked and its value should be false by default,when the cancel button is clicked and also when after submitting the form

- Now we are creating a state variable with default value false

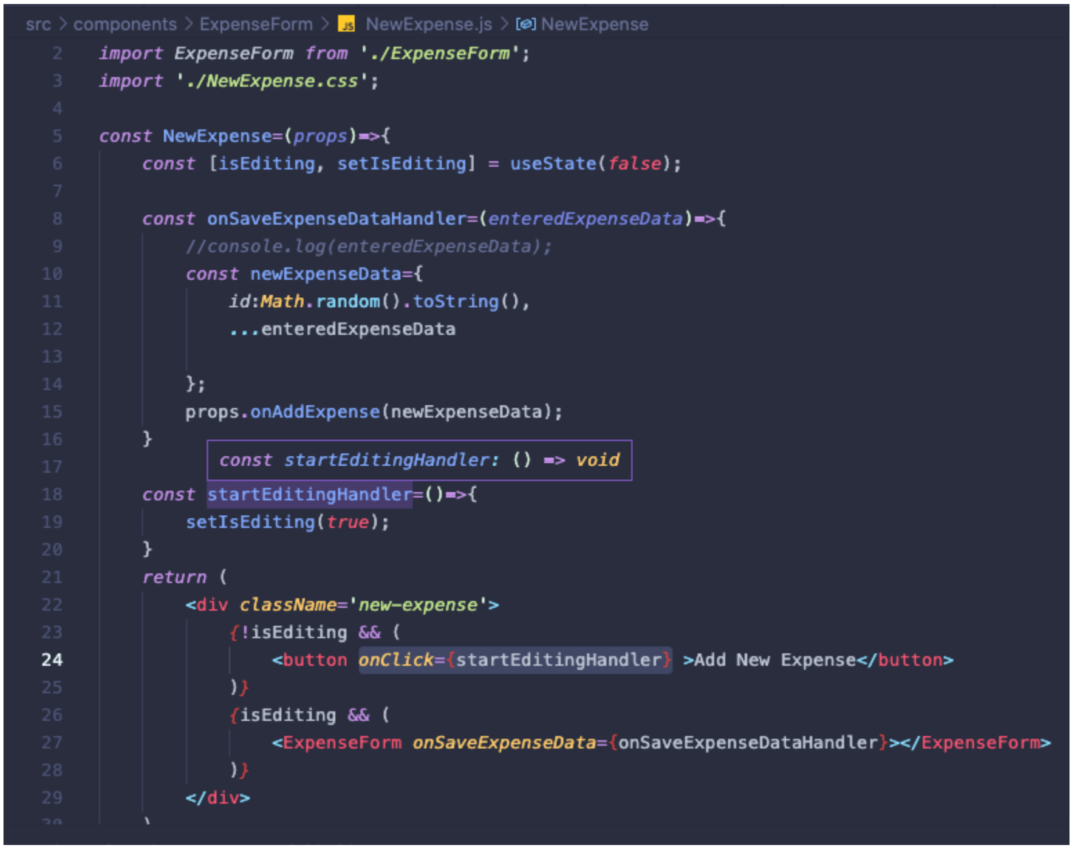


```
import React,{useState} from 'react';
import ExpenseForm from './ExpenseForm';
import './NewExpense.css';

const NewExpense=(props)=>{
    const [isEditing, setIsEditing] = useState(false);

    const onSaveExpenseDataHandler=(enteredExpenseData)=>{
        //console.log(enteredExpenseData);
        const newExpenseData={
            id:Math.random().toString(),
            ...enteredExpenseData

        };
        props.onAddExpense(newExpenseData);
    }
    return (
        <div className='new-expense'>
            <ExpenseForm onSaveExpenseData={onSaveExpenseDataHa
        </div>
    )
```

- Now we are using condition content rendering



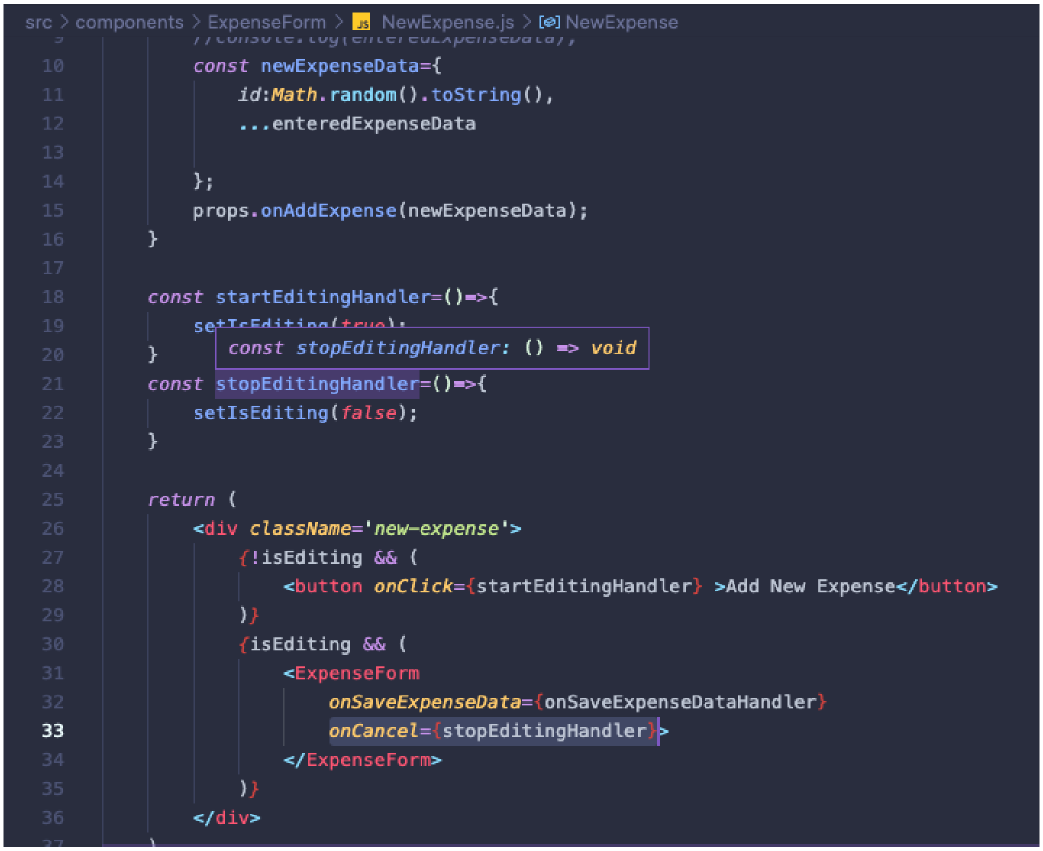
```
import ExpenseForm from './ExpenseForm';
import './NewExpense.css';

const NewExpense=(props)=>{
    const [isEditing, setIsEditing] = useState(false);

    const onSaveExpenseDataHandler=(enteredExpenseData)=>{
        //console.log(enteredExpenseData);
        const newExpenseData={
            id:Math.random().toString(),
            ...enteredExpenseData

        };
        props.onAddExpense(newExpenseData);
    }
    return (
        <div className='new-expense'>
            {!isEditing && (
                <button>Add New Expense</button>
            )}
            {isEditing && (
                <ExpenseForm onSaveExpenseData={onSaveExpenseDataHandler}></ExpenseForm>
            )}
        </div>
    )
}

export default NewExpense;
```

- Now we must show the form(i.e turn isEditing to true) whenever the Add new Expense is clicked
    - 
src > components > ExpenseForm > NewExpense.js > NewExpense
const startEditingHandler: () => void

```
import ExpenseForm from './ExpenseForm';
import './NewExpense.css';

const NewExpense=(props)=>{
    const [isEditing, setIsEditing] = useState(false);

    const onSaveExpenseDataHandler=(enteredExpenseData)=>{
        //console.log(enteredExpenseData);
        const newExpenseData={
            id:Math.random().toString(),
            ...enteredExpenseData

        };
        props.onAddExpense(newExpenseData);
    }

    const startEditingHandler=()=>{
        setIsEditing(true);
    }
    return (
        <div className='new-expense'>
            {!isEditing && (
                <button onClick={startEditingHandler} >Add New Expense</button>
            )}
            {isEditing && (
                <ExpenseForm onSaveExpenseData={onSaveExpenseDataHandler}></ExpenseForm>
            )}
        </div>
    )
```

        - Now if the button is clicked then the statevariable isediting will be turned to true so that the form will be rendered

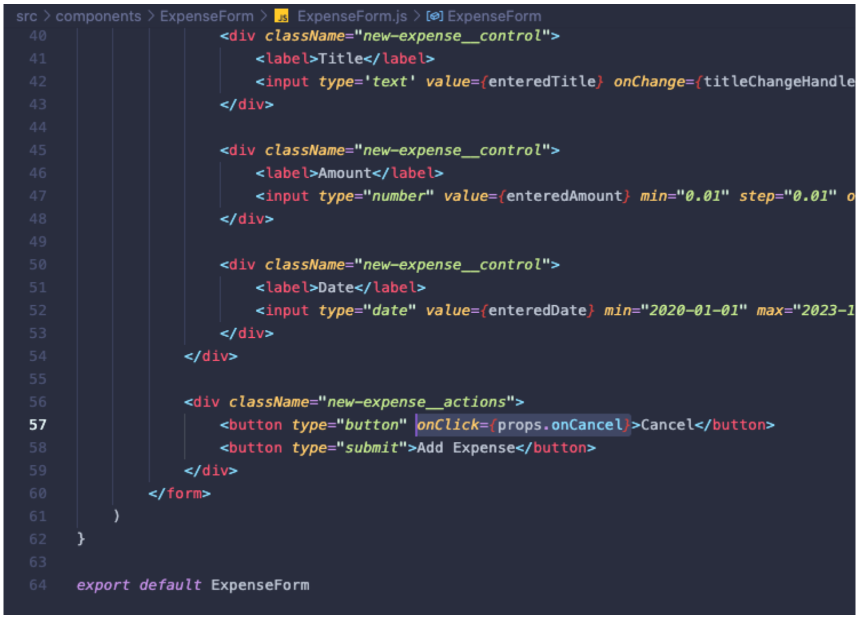
- Okay, now the form will be displayed, in the form we have two buttons one is cancel and another is submit, whenever the user clicks cancel again the state variable alue should be changed to false
    - But here the cancel button is in the child component, so
    - In newExpenses.js(parent)



src > components > ExpenseForm > NewExpense.js > NewExpense

```
        //console.log(enteredExpenseData);
        const newExpenseData={
            id:Math.random().toString(),
            ...enteredExpenseData

        };
        props.onAddExpense(newExpenseData);
    }

    const startEditingHandler=()=>{
        setIsEditing(true);
    }
    const stopEditingHandler=()=>{
        setIsEditing(false);
    }

    return (
        <div className='new-expense'>
            {!isEditing && (
                <button onClick={startEditingHandler} >Add New Expense</button>
            )}
            {isEditing && (
                <ExpenseForm
                    onSaveExpenseData={onSaveExpenseDataHandler}
                    onCancel={stopEditingHandler}>
                </ExpenseForm>
            )}
        </div>
    )
```

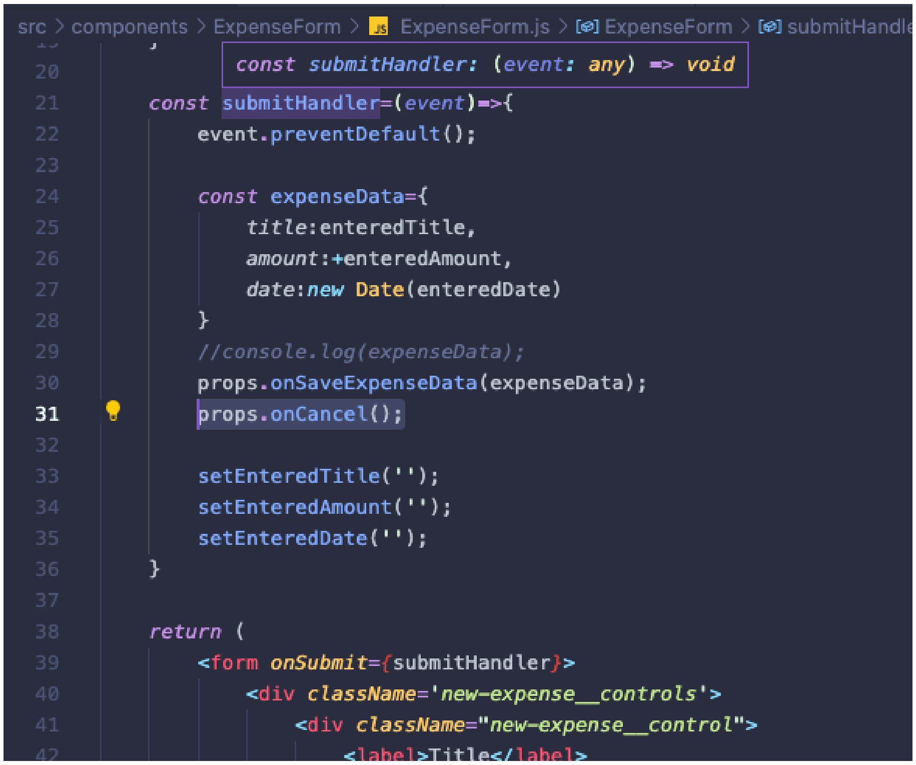
    -
    - Now in the child



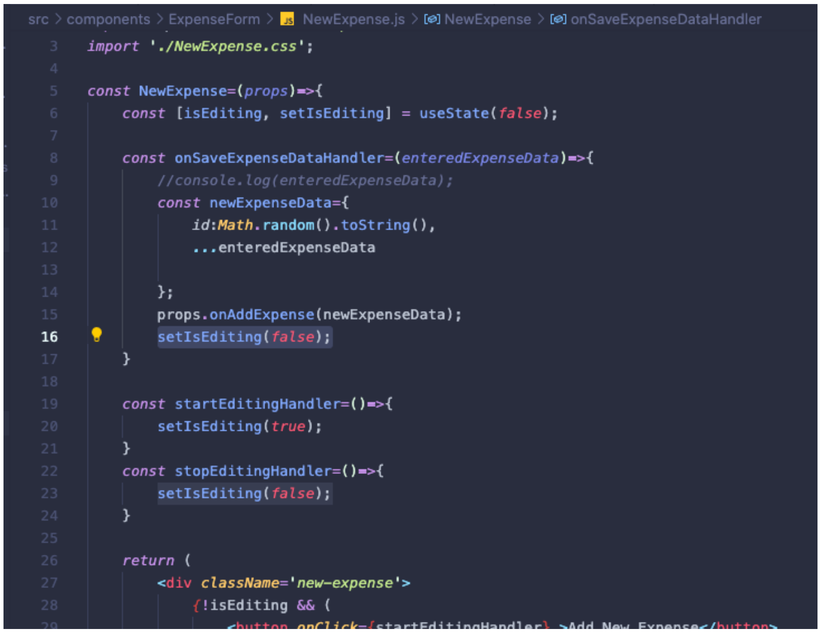
src > components > ExpenseForm > ExpenseForm.js > ExpenseForm

```
                <div className="new-expense__control">
                    <label>Title</label>
                    <input type='text' value={enteredTitle} onChange={titleChangeHandle
                </div>

                <div className="new-expense__control">
                    <label>Amount</label>
                    <input type="number" value={enteredAmount} min="0.01" step="0.01" o
                </div>

                <div className="new-expense__control">
                    <label>Date</label>
                    <input type="date" value={enteredDate} min="2020-01-01" max="2023-1
                </div>
            </div>

            <div className="new-expense__actions">
                <button type="button" onClick={props.onCancel}>Cancel</button>
                <button type="submit">Add Expense</button>
            </div>
        </form>
    )
}

export default ExpenseForm
```

        -

- Now one more thing, if the user submits the form then also we need to hide the form
- So
  - 
src > components > ExpenseForm > ExpenseForm.js > ExpenseForm > submitHandle
const submitHandler: (event: any) => void

```
    }

    const submitHandler=(event)=>{
        event.preventDefault();

        const expenseData={
            title:enteredTitle,
            amount:+enteredAmount,
            date:new Date(enteredDate)
        }
        //console.log(expenseData);
        props.onSaveExpenseData(expenseData);
        props.onCancel();

        setEnteredTitle('');
        setEnteredAmount('');
        setEnteredDate('');
    }

    return (
        <form onSubmit={submitHandler}>
            <div className='new-expense__controls'>
                <div className="new-expense__control">
                    <label>Title</label>
```

  - Inside the submit handler we called the oncancel method of the parent which will internally sets the value as false for state variable isEditing
  - Or we can also do this is in the parent itself because here in this submit handler we are calling the onSaveExpenseData of the parent so there we could implement this directly
  - 
src > components > ExpenseForm > NewExpense.js > NewExpense > onSaveExpenseDataHandler

```
import './NewExpense.css';

const NewExpense=(props)=>{
    const [isEditing, setIsEditing] = useState(false);

    const onSaveExpenseDataHandler=(enteredExpenseData)=>{
        //console.log(enteredExpenseData);
        const newExpenseData={
            id:Math.random().toString(),
            ...enteredExpenseData

        };
        props.onAddExpense(newExpenseData);
        setIsEditing(false);
    }

    const startEditingHandler=()=>{
        setIsEditing(true);
    }
    const stopEditingHandler=()=>{
        setIsEditing(false);
    }

    return (
        <div className='new-expense'>
            {!isEditing && (
                <button onClick={startEditingHandler} >Add New Expense</button>
```