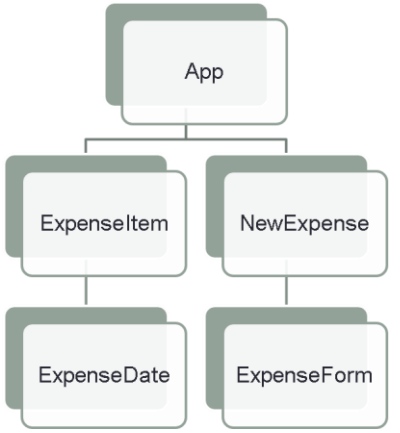
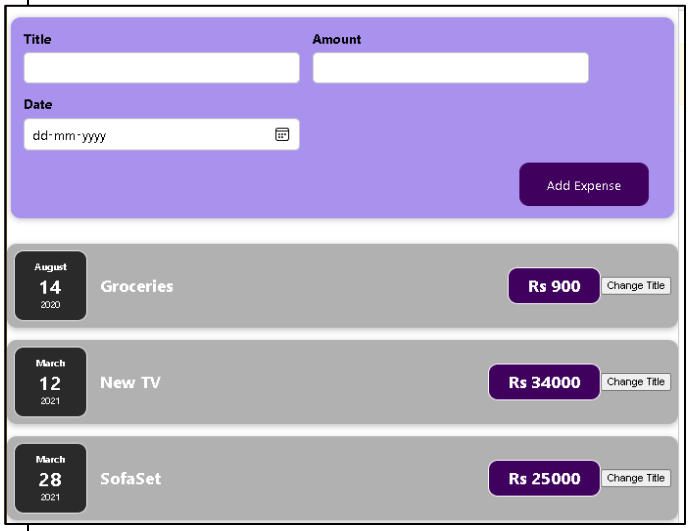
USING FORM FOR INPUT

Adding form inputs

```jsx
import "./ExpenseForm.css"
const ExpenseForm = () => {
    return(
      <form>
          <div className="new-expense__controls">
             <div className="new-expense__control">
                 <label>Title</label>
                 <input />
             </div>
             <div className="new-expense__control">
                 <label>Amount</label>
                 <input type="number"/>
             </div>
             <div className="new-expense__control">
                 <label>Date</label>
                 <input type="date" min="2019-01-01"
                        max="2022-12-31" />
             </div>
          </div>
          <div className="new-expense__actions">
             <button type="submit">Add Expense</button>
          </div>
      </form>
    );
}
export default ExpenseForm;
```

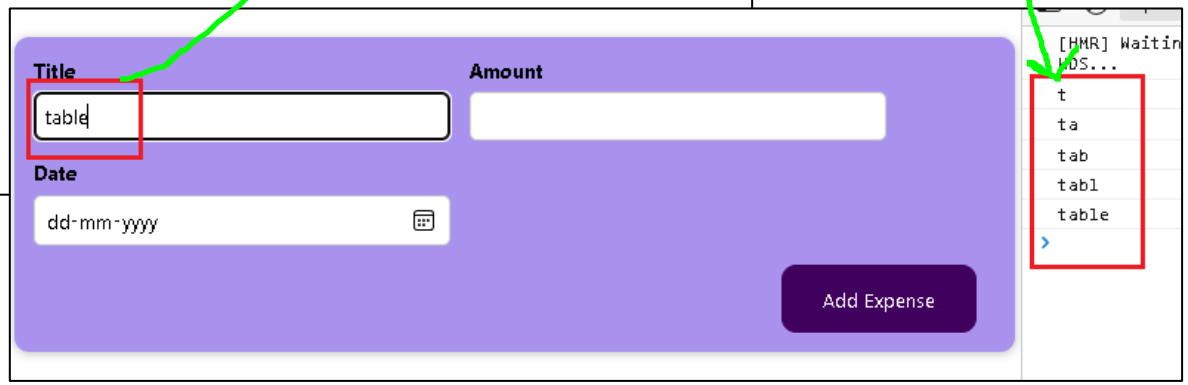
```jsx
const NewExpense = () => {
    return(
        <div className="new-expense">
            <ExpenseForm />
        </div>
    );
}
export default  NewExpense;
```





Listening to user input

```
const ExpenseForm = () => {

    const titleChangeHandler = (event) => {
        console.log(event.target.value)
    }
    return(
      <form>
          <div className="new-expense__controls">
             <div className="new-expense__control">
                 <label>Title</label>
                 <input onChange={titleChangeHandler}/>
             </div>
             ...
          </div>
          <div className="new-expense__actions">
              <button type="submit">Add Expense</button>
          </div>
      </form>
    );
}
export default ExpenseForm;
```



Storing input into state – working with multiple states

```
import React, {useState} from 'react';
const ExpenseForm = () => {

    const [inputTitle, setInputTitle] = useState('')
    const [inputAmount, setInputAmount] = useState('')
    const [inputDate, setInputDate] = useState('')

    const titleChangeHandler = (event) => {
        setInputTitle(event.target.value)
    }
    const amountChangeHandler = (event) => {
        setInputAmount(event.target.value)
    }
    const dateChangeHandler = (event) => {
        setInputDate(event.target.value)
    }
    return(
      <form>
          <div className="new-expense__controls">
             <div className="new-expense__control">
                 <label>Title</label> <input onChange={titleChangeHandler}/>
             </div>
             <div className="new-expense__control">
                 <label>Amount</label>
                 <input type="number" onChange={amountChangeHandler}/>
             </div> ...
```

1

2

3

Form submission

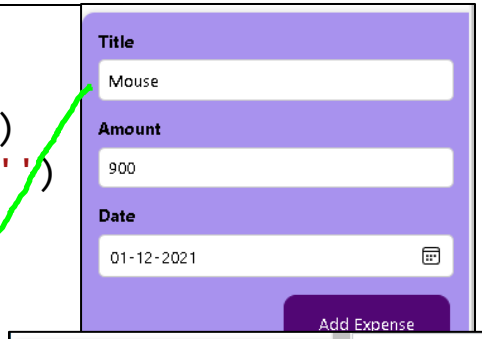
```
const ExpenseForm = () => {

    const [inputTitle, setInputTitle] = useState('')
    const [inputAmount, setInputAmount] = useState('')
    const [inputDate, setInputDate] = useState('')

    const titleChangeHandler = (event) => {···
    }
    const amountChangeHandler = (event) => {···
    }
    const dateChangeHandler = (event) => {···
    }
    const submitHandler = (event) => {
        event.preventDefault();
    }
    return(
      <form onSubmit={submitHandler}>
            <div className="new-expense__controls">···
            </div>
            <div className="new-expense__actions">
                <button type="submit">Add Expense</button>
            </div>
      </form>
    );
}
```

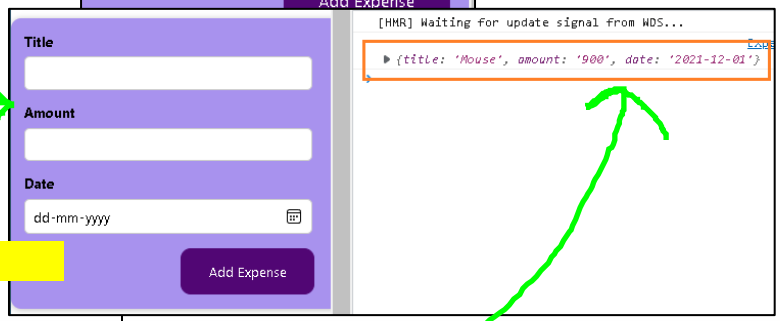
OnSubmit default behaviour is to reload

Form submission – extracting data, 2-way binding

```
const ExpenseForm = () => {

    const [inputTitle, setInputTitle] = useState('')
    const [inputAmount, setInputAmount] = useState('')
    const [inputDate, setInputDate] = useState('')

    const titleChangeHandler = (event) => {...}
    const amountChangeHandler = (event) => {...}
    const dateChangeHandler = (event) => {...}

    const submitHandler = (event) => {
        event.preventDefault();
        const expenseData = {
            title:inputTitle,
            amount:inputAmount,
            date:inputDate
        }
        console.log(expenseData)
        setInputAmount('')
        setInputDate('')
        setInputTitle('')
    }
```

```
return(
    <form onSubmit={submitHandler}>
        <div className="new-expense__controls">
            <div className="new-expense__control">
                <label>Title</label>
                <input value={inputTitle}
                       onChange={titleChangeHandler}/>
            </div>
            ...
        </form>
    );
}
```





1

3 : setting values onchange

4 : on submit passing values to object

2 : Default value

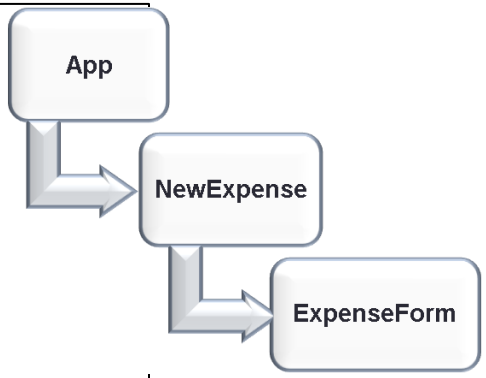
7 : reset again a

5

6

Passing data from child to parent component

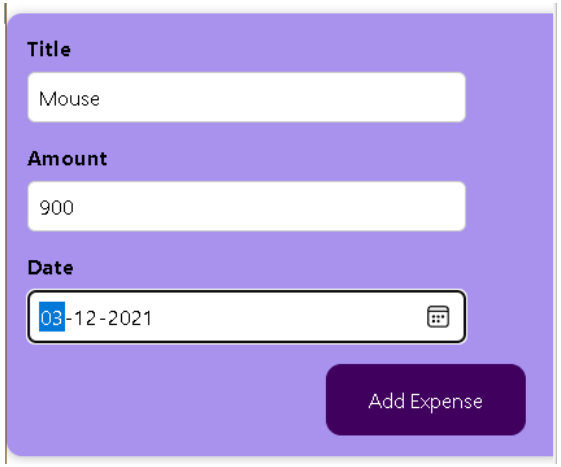
```
const NewExpense = () => {

    const saveExpenseDataHandler = (inputExpenseData) => {
        const expenseData = {
            ...inputExpenseData,
            id:Math.random().toString()
        }
        console.log("In NewExpense ",expenseData)
    }
    return(
        <div className="new-expense">
            <ExpenseForm onSaveExpenseData={saveExpenseDataHandler}/>
        </div>
    );
}

export default  NewExpense;
```




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

    ...
    const submitHandler = (event) => {
        event.preventDefault();
        const expenseData = {
            title:inputTitle,
            amount:inputAmount,
            date: new Date(inputDate)
        }
        //console.log(expenseData)
        props.onSaveExpenseData(expenseData);
        ...
    }
```

Passing data from child to parent

```
function App() {
  const expenses = [...];

  const addExpenseHandler = expense => {
    console.log("In App component ", expense)
  }

  return (
    <div className="App">
      <h2>Welcome to React!</h2>
      <NewExpense onAddExpense={addExpenseHandler} />
      ...
  );
}
export default App;
```




```
In App component                                    App.js:20
▶ {title: 'Mouse', amount: '900', date: '2021-12-03',
   id: '0.25759054649698765'}
```

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

    const saveExpenseDataHandler = (inputExpenseData) => {
        const expenseData = {
            ...inputExpenseData,
            id:Math.random().toString()
        }
        //console.log("In NewExpense ",expenseData)
        props.onAddExpense(expenseData)
    }
    return(...);
}
export default  NewExpense;
```