

## 7. Composition (allowing children components or default html elements nested inside custom component)

- Until now, in JSX only default html element can have other elements or components nested inside it as children element, but our custom component cannot be used to wrap other elements
  - Look here

```
ex.js  index.html  App.js  ExpenseItem.js  Expenses.css

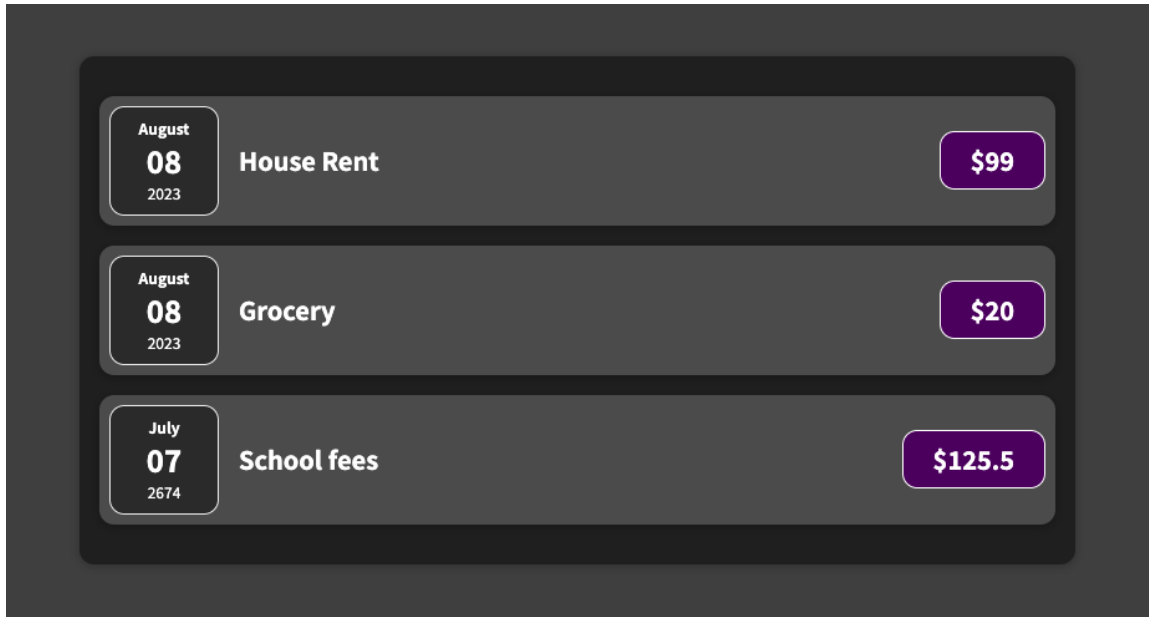
src > components > ExpenseItem.js > ExpenseItem

8
9   return (
10    <div 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'>${props.amount}</div>
15      </div>
16    </div>
17  );
18 }
19
20 export default ExpenseItem;
21
```

- Here the default html tag 'div' with classname expense-item has several elements inside it as child elements
- But whereas, our custom component like ExpenseDate won't support wrapping up of other elements inside it

```
11    <ExpenseDate date={props.date}>
12      <p>hello</p>
13    </ExpenseDate>
```

- It does not have any effect because it is not allowed by default in our custom components
- But we can make our normal custom component to allow wrapping up of other elements and such components are called wrapper components



- Look here, the outer div with dark grey color and the div of expense item have some similar stylings like border radius and box-shadow

```

... ><div class="expense-item">...</div> flex == $0
    ><div class="expanse-item">...</div> flex
html body div#root div div.expenses div.expense-item
Styles Computed Layout Event Listeners DOM Breakpoints >>
Filter :hov .cls + [ [
element.style {
}
.expense-item { ExpenseItem.css:1
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 0.5rem;
  margin: 1rem 0;
  box-shadow: 0 2px 8px rgb(0 0 0 / 25%);
  border-radius: 12px;
  background-color: #4b4b4b;
}

```

```

... <div class="expenses">...</div> == $0
</div>
</div>
html body div#root div div.expenses
Styles Computed Layout Event Listeners DOM Breakpoints >>
Filter :hov .cls + [ [
element.style {
}
.expenses { Expenses.css:1
  padding: 1rem;
  background-color: rgb(31, 31, 31);
  margin: 2rem auto;
  width: 50rem;
  max-width: 95%;
  border-radius: 12px;
  box-shadow: 0 1px 8px rgb(0 0 0 / 25%);
}

```

- Instead of rewriting this, we can make a custom wrapper component which can have this common styling and these components can be wrapped into our new component so code rewriting is avoided

```
Card.js — react-complete-guide
Item.js  Expenses.css  Expenses.js  Expens
src > components > Card.js > ...
1  import './Card.css'
2
3  function Card() {
4    return (<div className="Card">
5
6      </div>
7    );
8  }
9
10 export default Card;
11
```

- 
- Now create a custom component
- 

```
Card.css — react-complete-guide
Expenses.css  Expenses.js  ExpenseDate.js  ExpenseDat
src > components > Card.css > .Card
1  .Card{
2    border-radius: 12px;
3    box-shadow: 0 1px 8px rgba(0, 0, 0, 0.25);
4  }
```

- And css for it is also added
- Now we can replace this div by our custom component

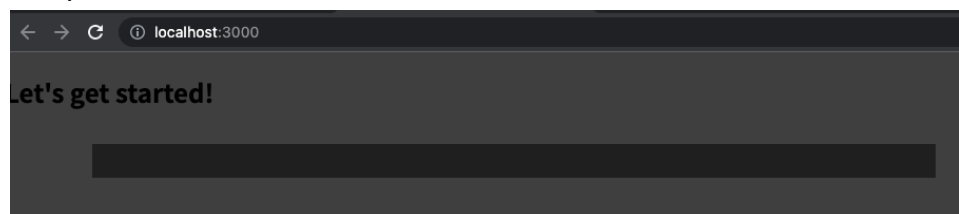
```
Expenseltem.js x Expenses.css Expenses.js Expense
src > components > Expenseltem.js > Expenseltem
3
4 function ExpenseItem(props) {
5   // const expenseDate=new Date(2023, 7, 30);
6   // const expenseItem='House Rent';
7   // const expenseAmount=250.00;
8
9   return (
10    <div className='expense-item'>
11      <ExpenseDate date={props.date}>
12        <p>hello</p>
13      </ExpenseDate>
14      <div className='expense-item__description'>
15        <h2>{props.title}</h2>
16        <div className='expense-item__price'>${props.amou
17      </div>
18    </div>
19  );
20 }
21
22 export default ExpenseItem;
```

○

```
Expenseltem.js x Expenses.css Expenses.js ExpenseDate.js
src > components > Expenseltem.js > Expenseltem
3 import Card from './Card';
4
5 function ExpenseItem(props) {
6   // const expenseDate=new Date(2023, 7, 30);
7   // const expenseItem='House Rent';
8   // const expenseAmount=250.00;
9
10  return (
11    <Card className='expense-item'>
12      <ExpenseDate date={props.date}>
13        <p>hello</p>
14      </ExpenseDate>
15      <div className='expense-item__description'>
16        <h2>{props.title}</h2>
17        <div className='expense-item__price'>${props.amount}</div>
18      </div>
19    </Card>
20  );
21 }
22
```

○

○ Now the output is



■

```
<body data-new-gr-c-s-check-loaded="14.1078.0" installed cz-shortcut-listen="true">
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root">
    <div>
      <h2>Let's get started!</h2>
      <div class="expenses">
        ... <div class="Card"></div> == $0
        <div class="Card"></div>
        <div class="Card"></div>
      </div>
    </div>
  </div>
</body>
```

html body div#root div div.expenses div.Card

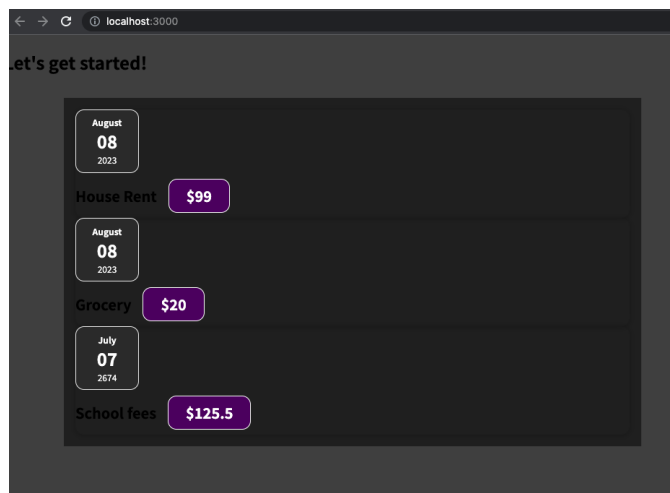
- Now when you see our custom wrapper component, it doesn't have any children inside it.
- Because we know that by default our component doesn't support any children inside it
- To make it support we should do this

```
src > components > Card.js > Card
1 import './Card.css'
2
3 function Card(props) {
4   return (<div className="Card">
5
6     </div>
7   );
8 }
9
10 export default Card;
```

- In the custom component, we should accept props, because the children elements will also be sent as props
- Now do this

```
Card.js — react-complete-guide
ses.js ExpenseDate.js ExpenseDate.css Card.js x
src > components > Card.js > Card
1 import './Card.css'
2
3 function Card(props) {
4   return (<div className="Card">
5     {props.children}
6   </div>
7   );
8 }
9
10 export default Card;
11
```

- 
- Here the children is a keyword in react, all the children elements will be sent as props with the key name children.
- Now the output is



- 
- Yeah we got it, but this is not the exact thing we need
- This is because

```
ExpenselItem.js x Expenses.css Expenses.js Exp
src > components > ExpenselItem.js > ExpenselItem
3 import Card from './Card';
4
5 function ExpenseItem(props) {
6   // const expenseDate=new Date(2023, 7, 30);
7   // const expenseItem='House Rent';
8   // const expenseAmount=250.00;
9
10  return (
11    <Card className='expense-item'>
12      <ExpenseDate date={props.date}>
13        <p>hello</p>
14      </ExpenseDate>
15      <div className='expense-item_description'>
16        <h2>{props.title}</h2>
17        <div className='expense-item_price'>${props.
18      </div>
19    </Card>
20  );
21 }
```

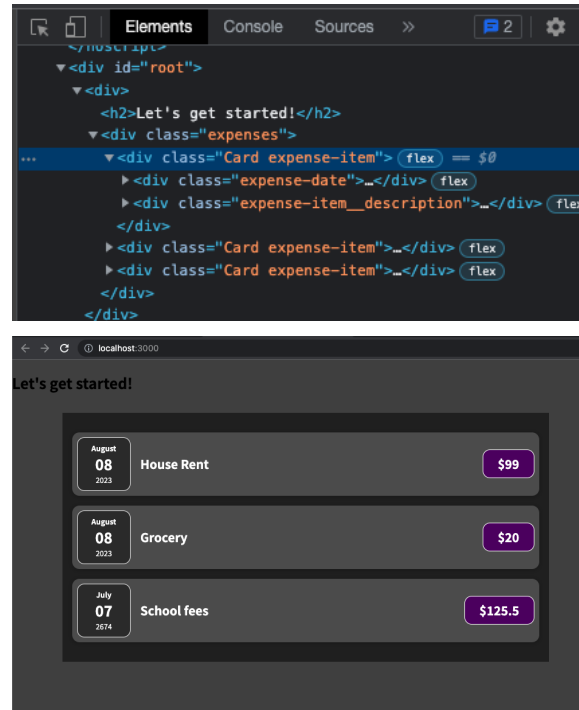
```
Elements Console Sources >> 1
</noscript>
<div id="root">
  <div>
    <h2>Let's get started!</h2>
    <div class="expenses">
      <div class="Card"> == $0
        <div class="expense-date">_</div> flex
        <div class="expense-item_description">_</div> fl
      </div>
      <div class="Card">_</div>
      <div class="Card">_</div>
    </div>
  </div>
</div>
```

- Here if you notice that we are adding expense-item as another classname for card component but while rendering it just has the class card not expense-item.
- To make it work, do this

```
Card.js — react-complete-guide
src > components > Card.js > Card
1 import './Card.css';
2
3 function Card(props) {
4   const classes = "Card " + props.className;
5   return (
6     <div className={classes}>
7       {props.children}
8     </div>);
9 }
10
11 export default Card;
12
```

- We can access the className that is passed while calling the component using props like we do for other attributes, here the output is





- This is what we need
- Similarly we can use this card component for other similar components which needs card styling
  - We can add this for expenses also

```

Expenses.js — react-complete-guide
ExpensesItem.js Expenses.css Expenses.js x
src > components > Expenses.js > Expenses
1 import ExpenseItem from './ExpenseItem';
2 import './Expenses.css';
3
4 function Expenses(props) {
5   return (
6     <div className="expenses">
7       <ExpenseItem
8         title={props.items[0].title}
9         amount={props.items[0].amount}
10        date={props.items[0].date}
11      />
12      <ExpenseItem
13        title={props.items[1].title}
14        amount={props.items[1].amount}
15        date={props.items[1].date}
16      />

```

