

```

1 // ===== Dynamically List Rendering =====
2 /*
3   Array Map use to Rendering array Elements
4
5   let arr = [1, 2 , 3];
6   let newArr = arr.map( (element) => element * 3);
7   // newArr = [3, 6, 9];
8
9   Here object array are used to List rendering...
10  {
11    expense.map((element) =>
12      // all components are one by one render [ < Card1 />, < Card2 />]..
13      < ExpensesItem
14        title = {element.title}
15        amount = {element.amount}
16        date = {element.date}
17      />
18    );
19  }
20
21
22  We can add Update Rendering by adding element
23
24  old object Array or Default Array element and ADD new update element
25  and Rendering.... the component by useState Hooks
26  expenseArr old = [
27    .....
28    .....
29    .....
30  ]
31  Copy old array and NewAdded element to
32  expenseArr = [...newAdded , expenseArr ];
33
34
35 */
36
37 // ===== index.js START =====
38
39 import React from 'react';
40 import ReactDOM from 'react-dom';
41 import App from './App';
42
43 ReactDOM.render(<App />, document.getElementById('root'));
44
45 //***** END *****
46
47 //===== App.js START =====
48 /*
49   Dynamic List update
50   1) get the child date to parent
51   2) one default object Array
52   3) useState use the default Array
53   4) when the update date input change
54   5) useState function to call and set the Data
55   6) Rendering the component to show the data
56
57   expenseArr here default obj Array
58
59   let [expenseUpdate, setExpenseUpdate] = useState(expenseArr);
60
61   function addExpenseHandler(expense){
62
63     expenseUpdate = [...expenseUpdate, expense];
64     setExpenseUpdate(expenseUpdate);
65
66   }
67

```

```

68 Rendering....
69
70 Best Praticce to useState function call by last componet ( snaps )
71 by call back function way.... useState function call
72
73 function addExpenseHandler(expense){
74
75     setExpenseUpdate((prevExpense) => {
76         return [...prevExpense, expense]
77     });
78
79 }
80
81 Add ID to help react to identify and stop flaush all component
82 but specific element only flush....
83 when Component call that time pass only => key={element.id};
84
85 */
86
87 import './App.css';
88 // import the Custom Component file
89 import React, {useState} from 'react';
90
91 import ExpencesItem from './component/Expense/ExpencesItem';
92 import NewExpense from './component/NewExpense/NewExpense';
93 import ExpenseFilter from './component/Expense/ExpenseFilter'
94 function App() {
95     let expenseArr = [
96         // thsi array contain some object
97         {
98             id: 'e1',
99             title: "Car Buy",
100             amount: 300000,
101             date: new Date(2020, 3, 5),
102         },
103         {
104             id: 'e2',
105             title: "Book Buy",
106             amount: 300,
107             date: new Date(2021, 4, 6),
108         },
109         {
110             id: 'e3',
111             title: "Phone Buy",
112             amount: 3000,
113             date: new Date(2023, 1, 3),
114         },
115     ];
116
117
118 // Update input value
119 const [expenseUpdate, setExpenseUpdate] = useState(expenseArr);
120
121 function addExpenseHandler(expense){
122     // console.log("App.js run");
123     console.log(expense);
124     // expenseUpdate = [...expenseUpdate, expense];
125     // setExpenseUpdate(expenseUpdate);
126     setExpenseUpdate((prevExpense) => {
127         return [...prevExpense, expense]
128     });
129 }
130 const [filteredYear, seteFilteredYear] = useState('2020');
131 function filterChangeHandler(selectedYear){
132     seteFilteredYear(selectedYear);
133 }
134 return (
135     <div >

```

```

136 { /* When New Expense component Call that time Excute onSide this pass function */}
137 <NewExpense onAddExpense={addExpenseHandler}> </NewExpense>
138
139 <div className="wrap-item">
140   <ExpenseFilter selected={filteredYear} onChangeFilter={filterChangeHandler} />
141
142   { /* It component replacement by Dynamically */}
143
144   { /* <ExpencesItem
145     title={expenseArr[0].title}
146     price={expenseArr[0].price}
147     date={expenseArr[0].date}
148   ></ExpencesItem> */}
149
150   {expenseUpdate.map( (element) =>
151     <ExpencesItem
152       key={element.id}
153       title={element.title}
154       amount={element.amount}
155       date = {element.date}
156     />
157   )}
158
159 </div>
160
161 </div>
162 );
163 }
164
165
166
167 export default App;
168
169
170 //***** END *****
171
172
173 //===== Expense Filter.js connect ot App.js =====
174
175
176 import './ExpenseFilter.css'
177 function ExpenseFilter(props){
178
179   function dropdownChangeHandler(event){
180
181     props.onChangeFilter(event.target.value);
182   };
183
184   return(
185     <div className="expense-filter">
186       <div className="expense-filter_control">
187         <label>Filter by year</label>
188         <select value={props.selected} onChange={dropdownChangeHandler}>
189           <option value='2022'>2022</option>
190           <option value='2023'>2023</option>
191           <option value='2024'>2024</option>
192           <option value='2025'>2025</option>
193
194         </select>
195       </div>
196     </div>
197   );
198 };
199 export default ExpenseFilter;
200
201 //***** END *****
202
203 //===== ExpencesItem.js START =====
204

```

```

204
205
206 // here useState hooks use to update change of variable
207
208 import React, { useState } from 'react';
209
210 // import CSS file
211 import './ExpencesItem.css';
212 // import the Date component file
213 import ExpenseDate from './ExpenseDate';
214
215 function ExpencesItem(props) {
216   // Here the onClick function Body
217
218   // let title = props.title;
219
220   // Here Array [variable, fuction ] = useState(take_default_variable)
221
222   const [title, setTitle] = useState(props.title);
223
224   const clickHandler = () => {
225
226     //title = "Update Title";
227
228     // Here value change by function
229     setTitle("Update");
230
231     /* only Set the title will not show to website
232     When the Component Call That time Read At a time but => Title varibale
233     After button Click How we can Set and Update value it is Our are concern
234     */
235     console.log(title);
236   }
237
238   return (
239     <div className="container">
240       <div className="date-item">
241         <ExpenseDate date={props.date}> </ExpenseDate>
242       </div>
243       <div className="name-item">{title /* set vaibale*/}</div>
244       <div className="price-item">${props.amount}</div>
245       /* <button onClick = {} => { console.log("Clicked")}> Clicked Me</button> */
246       <button onClick={clickHandler} > Change title </button>
247       /*
248         1) Calling not use the ()
249         2) COZ we just point the function Only
250         3) if We use (), When the this Item component call the the line fun excute that time instatly
251         4) But we want to excute when the user click the button got it??
252       */
253     </div>
254   );
255 }
256
257 export default ExpencesItem;
258
259 //***** END *****
260
261 //===== ExpencesDate.js START =====
262
263
264 // CSS file import for use this....
265 import './ExpenseDate.css';
266 // we can use CSS on this file
267
268 function ExpenseDate(props) {
269   // use some javascript code..
270   // date varibale should be base on the given (props vaibale) name
271
272   const month = props.date.toLocaleString('en-US', { month: 'long' });

```

```

272 const month = props.date.toLocaleString('en-US', { month: 'long' });
273 const day = props.date.toLocaleString('en-US', { day: '2-digit' });
274 const Year = props.date.getFullYear();
275
276 return (
277     <div>
278         <div> {month} </div>
279         <div>{Year}</div>
280         <div>{day}</div>
281     </div>
282 );
283
284 }
285
286 export default ExpenseDate;
287
288 //***** END *****
289
290
291 //===== NewExpense.js =====
292
293 import React from 'react';
294 import './NewExpense.css'
295 import NewExpenseForm from './NewExpenseForm'
296
297 const NewExpense = (props) => {
298     // this prop send data to App.js means parent componenet
299
300
301     // Collect the Expense data from child components NewExpenseFrom.js
302
303     function saveExpenseDataHandler(enteredExpenseData){
304         const ExpenseData = {
305             ...enteredExpenseData,
306             id: Math.random().toString(),
307         }
308         // console.log(ExpenseData);
309         // Data pass to Parent App.js component by Executing Function...
310         props.onAddExpense(ExpenseData);
311     }
312
313     return (
314         <div className="form-section" >
315
316             <NewExpenseForm onSaveExpenseDate={saveExpenseDataHandler}/>
317
318         </div>
319
320     );
321 };
322
323 export default NewExpense;
324
325
326 //***** End *****
327
328 //===== NewExpenseFrom.js =====
329
330 /*
331     How we can get the input value <= FORM
332
333     1) inside the < input set  onChange = {function_ponnter} >
334     2) inside the component function create Handler_body(Take_event)
335     3) The (event) Help to get the input values from
336     4) console.log(event.target.value);
337     5) this event.target.value gives Always String ...
338
339     After Getting values of input to Store
340

```

```

341 1) we have to use useState
342 2) useState help to track the change value
343
344 Handelling Form submission
345 1) best practice is don't use onClick => inside button is COZ
346 2) when submit button click that time from total subbmit
347 3) <from onSubmit = { submitHandler } > .....
348 4) create function submintHandler() { body } ... give instraction ....
349 5) submitHandeler(event){event.preventDefault();} stop to page relod
350 6) inside the submitHandler create object and group of the form date and work with that.
351
352 Two way bunding .. menas reset the from
353 1) inside the < input value={enteredTitle} />
354 2) inside the submintHandler() After the object{}
355 3) new hooks value call to reset the input Area...
356
357 App.js => Expenseltem.js => NewExpense.js => NewExpenseFrom.js
358
359 Now to we can Data pass parent to child by Props NewExpense.js => NewExpenseFrom.js
360
361 But How we can Data pass Child to Parent NewExpense.js <= NewExpenseFrom
362 1) Up word Data pass..
363 2) When Component Call that time Pass a Function pointer to Execute the Parent function
364 /
365 ComponentParetn(){
366   Parent(){
367     .....Task
368   }
369   return(
370     < ComponentChild onParentTask = {Parent} />
371   );
372 }
373
374 ComponentChild(props){
375   ....
376   ...
377   ..Task
378   ....
379   props.Parent(DataPassToParent);
380
381   return(
382     ....
383   );
384 }
385
386 3) Child component excute the Parent fuction and receive Data...
387
388
389
390
391
392 */
393 import Ract, { useState } from 'react';
394 import "./NewExpenseForm.css";
395
396 function NewExpenseFrom(props) {
397   // here props only use to send data to parent component NewExpense.js
398
399   // use the useState React hooks and value separte like
400   // [variable, function] = useState(default_value);
401   // we can use separte useState for each input changees..
402
403   const [enteredTitle, setEnteredTitle] = useState("");
404   const [enteredAmount, setEnteredAmount] = useState("");
405   const [enteredDate, setEnteredDate] = useState("");
406
407   const titleChangeHandler = (event) => {
408

```

```

409 // console.log(event.target.value);
410
411 // When the titleChangeHandler call that time
412 // call the setEnteredTitle function initializes the Update input
413 setEnteredTitle(event.target.value);
414 }
415 const amoutChangeHandler = (event) =>{
416     setEnteredAmount(event.target.value);
417 }
418
419
420 const dateChangeHandler = (event) =>{
421     setEnteredDate(event.target.value);
422 }
423
424 const submitHandler = (event) =>{
425     event.preventDefault(); // to stop page realod
426     // create object
427     // console.log("Mehedi Hasna");
428     const expenseData = {
429         title: enteredTitle,
430         amout: enteredAmount,
431         date: new Date(enteredDate),
432     };
433
434     // console.log(expenseData);
435
436     // props only use here to Send the data to Parent NewExpense.js <= NewExpenseFrom.js
437     props.onSaveExpenseDate(expenseData);
438
439     // New call to reset value set...with the help of <input vlaue={} />
440     setEnteredTitle("");
441     setEnteredAmount("");
442     setEnteredDate("");
443 }
444 return (
445
446
447     /*onSubmit={submitHandler}*/
448     <from >
449     <div className="new-expense_controls">
450         <div className="new-expense_control">
451             <label>Title</label>
452             <input type="text" value={enteredTitle} onChange={titleChangeHandler /* funcation call */} />
453         </div>
454         <div className="new-expense_control">
455             <label>Amount</label>
456             <input type="number" min="0.01" step="0.01" value={enteredAmount} onChange = {amoutChangeHandler} />
457         </div>
458         <div className="new-expense_control">
459             <label>Date</label>
460             <input type="date" min="2022-01-01" max="2025-12-31" value={enteredDate} onChange = {dateChangeHandler} />
461         </div>
462     </div>
463     <div className="new-expense_actions">
464         <button type='submit' onClick={submitHandler} >Add Expense</button>
465     </div>
466
467 </from>
468
469 );
470 };
471
472
473 export default NewExpenseFrom;
474
475 //***** END *****
476

```

```
477
478
479 //***** EXpencesItem.css *****
480 /*
481 *{
482     margin: 0;
483     padding: 0;
484 }
485 .container{
486     width: 50%;
487     height: 300px;
488     margin: 0 auto;
489 }
490 .heading{
491     font-size: 25px;
492     background-color: gold;
493 }
494
495 /* we can use media query for verity devices */
496
497 //***** END *****
```