```
// ============== Dynamically List Rendaring ==========
/*
    Array Map use to Rendaring array Elements

    let arr = [1, 2 , 3];
    let newArr = arr.map( (element) => exlement * 3);
    // newArr = [3, 6, 9];

    Here object array are used to List rendearing...
    {
      expense.map((element) =>
          // all components are one by one render [ < Card1 />, < Card2 />]..
          < ExpencesItem
            title = {element.title}
            amount = {element.amount}
            date = {element.date}
            />
      );
    }


    We can add  Update Rendaring by adding element

    old object Arreay or Default Array element and ADD new update element
    and Renderaing.... the component by useState Hooks
    expenseArr old = [
      ...............
      ............
      ......
    ]
    Copy old array and NewAdded element to
    expenseArr = [...newAdded , expenseArr ];


*/

// =============== index.js START =============

import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(<App />, document.getElementById('root'));

//******************** END *********************

//================== App.js START =================
/*
  Dynamic List update
  1) get the child date to parent
  2) one default object Array
  3) useState use the default Array
  4) when the udpate date input change
  5) useState function to call and set the Data
  6) Rendaring the component  to show the data

  expenseArr here default obj Array

   let [expenseUpdate, setExpenseUpdate] = useState(expenseArr);

  function addExpenseHandler(expense){

    expenseUpdate = [...expenseUpdate, expense];
    setExpenseUpdate(expenseUpdate);

  }

  Rendering....

  Best Pratice to useState function call by last componet ( spnaps )
  by call back function way.... useState function call

  function addExpenseHandler(expense){

    setExpenseUpdate((prevExpense) => {
      return [...prevExpense, expense]
    });

  }

  Add ID to help react to identify and stop flaush all component
  but specific element only flush....
  when Component call that time pass only => key={element.id};

*/

import './App.css';
// import the Custom Component file
import React, {useState} from 'react';

import ExpencesItem from './component/Expense/ExpencesItem';
import NewExpense from './component/NewExpense/NewExpense';
import ExpenseFilter from './component/Expense/ExpenseFilter'
function App() {
  let expenseArr = [
    // thsi array contain some object
    {
      id: 'e1',
      title: "Car Buy",
      amount: 300000,
      date: new Date(2020, 3, 5),
    },
    {
      id: 'e2',
      title: "Book Buy",
      amount: 300,
      date: new Date(2021, 4, 6),
    },
    {
      id: 'e3',
      title: "Phone Buy",
      amount: 3000,
      date: new Date(2023, 1, 3),
    },

  ];

  // Update input value
  const [expenseUpdate, setExpenseUpdate] = useState(expenseArr);

  function addExpenseHandler(expense){
    // console.log("App.js run");
    console.log(expense);
    // expenseUpdate = [...expenseUpdate, expense];
    // setExpenseUpdate(expenseUpdate);
    setExpenseUpdate((prevExpense) => {
      return [...prevExpense, expense]
    });
  }
  const [filteredYear, seteFilteredYear] = useState('2020');
  function filterChangeHandler(selectedYear){
      seteFilteredYear(selectedYear);
  }
  return (
    <div >
```

```
     {/* When New Expense component Call that time Excute onSide this pass function */}
     <NewExpense onAddExpense={addExpenseHandler}> </NewExpense>

      <div className="wrap-item">
          <ExpenseFilter selected={filteredYear} onChangeFilter={filterChangeHandler} />

        {/* It component replacement by Dynamically */}

        {/* <ExpencesItem
          title={expenseArr[0].title}
          price={expenseArr[0].price}
          date={expenseArr[0].date}
        ></ExpencesItem> */}

        {expenseUpdate.map( (element) =>
          <ExpencesItem
              key={element.id}
              title={element.title}
              amount={element.amount}
              date = {element.date}
          />

        )}

      </div>

    </div>
  );
}


export default App;


//**************** END ********************


//======================== Expense Filter.js connect ot App.js ==============


import './ExpenseFilter.css'
function ExpenseFilter(props){

    function dropdownChangeHandler(event){

        props.onChangeFilter(event.target.value);
    };

        return(
           <div className="expense-filter">
               <div className="expense-filter_control">
                    <label>Filter by year</label>
                    <select value={props.selected} onChange={dropdownChangeHandler}>
                        <option value='2022'>2022</option>
                        <option value='2023'>2023</option>
                        <option value='2024'>2024</option>
                        <option value='2025'>2025</option>

                    </select>
               </div>
            </div>
        );
};
export default ExpenseFilter;

//***************** END ************************

//================ ExpencesItem.js START ============
// here useState hooks use to update change of variable

import React, { useState } from 'react';

// import CSS file
import './ExpencesItem.css';
// import the Date component file
import ExpenseDate from './ExpenseDate';

function ExpencesItem(props) {
    // Here the onClick function Body

    // let title = props.title;

    // Here Array [variable, fucntion ] = useState(take_default_variable)

    const [title, setTitle] = useState(props.title);

    const clickHandler = () => {

        //title = "Update Title";

        // Here value change by function
        setTitle("Update");

        /* only Set the title will not show to website
         When the Component Call That time Read At a time but => Title varibale
        After button Click How we can Set and Update value it is Our are consern
        */
        console.log(title);
    }

    return (
        <div className="container">
            <div className="date-item">
                <ExpenseDate date={props.date}> </ExpenseDate>
            </div>
            <div className="name-item">{title /* set vaibale*/}</div>
            <div className="price-item">${props.amount}</div>
            {/* <button onClick = {() => { console.log("Clicked")}}> Clicked Me</button> */}
            <button onClick={clickHandler} > Change title </button>
            {/*
                1) Calling not use the ()
                2) COZ we just point the function Only
                3) if We use (), When the this Item component call the the line fun excute that time instatly
                4) But we want to excute when the user click the button got it??
             */}
        </div>
    );
}

export default ExpencesItem;

//****************** END **************************

//================ ExpencesDate.js START =============


// CSS file import for use this....
import './ExpenseDate.css';
// we can use CSS on this file

function ExpenseDate(props) {
    // use some javascript code..
    // date varibale should be base on the given (props vaibale) name

    const month = props.date.toLocaleString('en-US', { month: 'long' });
    const month = props.date.toLocaleString('en-US', { month: 'long' });
    const day = props.date.toLocaleString('en-US', { day: '2-digit' });
    const Year = props.date.getFullYear();

    return (

        <div>
            <div> {month} </div>
            <div>{Year}</div>
            <div>{day}</div>
        </div>
    );

}
export default ExpenseDate;

//****************** END **************************


//====================== NewExpense.js =================

import React from 'react';
import './NewExpense.css'
import NewExpenseFrom from './NewExpenseForm'

const NewExpense = (props) => {
    // this prosp send data to App.js means parent componenet


// Collect the Expense data from child components NewExpenseFrom.js

    function saveExpenseDataHandler(enteredExpenseData){
        const ExpenseData = {
            ...enteredExpenseData,
            id: Math.random().toString(),
        }
        // console.log(ExpenseData);
        // Data pass to Parent App.js component by Executing Function...
        props.onAddExpense(ExpenseData);
    }

        return (
            <div className="form-section" >

                <NewExpenseFrom onSaveExpenseDate={saveExpenseDataHandler}/>

            </div>

        );
};

export default NewExpense;


//******************* End ******************************

//====================== NewExpenseFrom.js ===================

/*
    How we can get the input value <= FORM

    1) inside the < input set  onChange = {function_ponnter} >
    2) inside the component function create Handler_body(Take_event)
    3) The (event) Help to get the input values from
    4) console.log(event.target.value);
    5) this event.target.value gives Always String ...

    After Getting values of input to Store

    1) we have to use useState
    2) useState help to track the change value

    Handelling Form submission
    1) best pratice is don't use onClick => inside button is COZ
    2) when submit button click that time from total subbmit
    3) <from onSubmit = { submitHandler } > .........
    4) create function submintHandler() { body } ... give instraction ....
    5) submitHandeler(event){event.preventDefault();} stop to page relod
    6) inside the submitHandler create object and group of the form date and work with that.

    Two way bunding .. menas reset the from
    1) inside the < input value={enteredTitle} />
    2) inside the submintHandler() After the object{}
    3) new hooks value call to reset the input Area...

    App.js => ExpenseItem.js => NewExpense.js => NewExpenseFrom.js

    Now to we can Data pass parent to child by Props NewExpense.js => NewExpenseFrom.js

    But How we can Data pass Child to Parent NewExpense.js <= NewExpenseFrom
    1) Up word Data pass..
    2) When Component Call that time Pass a Function pointer to Execute the Parent function
    /
    ComponentParetn(){
        Parent(){
            ......Task
        }
        return(
            < ComponentChild onParentTask = {Parent} />
        );
    }

    ComponentChild(props){

        ....
        ...
        ..Task
        ....
        props.Parent(DataPassToParent);

        return(
            ....
        );
    }
    3) Child component excute the Parent fucntion and recerive Data...





*/
import Ract, { useState } from 'react';
import "./NewExpenseForm.css";

function NewExpenseFrom(props) {
    // here props only use to send data to parent component NewExpense.js

    // use the useState React hooks and value seperate like
    // [variable, function] = useState(default_value);
    // we can use seperate useState for each input changees..

    const [enteredTitle, setEnteredTitle] = useState('');
    const [enteredAmount, setEnteredAmount] = useState('');
    const [enteredDate, setEnteredDate] = useState('');

    const titleChangeHandler = (event) => {

        // console.log(event.target.value);

        // When the titleChangeHandler call that time
        // call the setEnteredTitle function initializes the Update input
        setEnteredTitle(event.target.value);
    }
    const amoutChangeHandler = (event) =>{

        setEnteredAmount(event.target.value);
    }

    const dateChangeHandler = (event) =>{

        setEnteredDate(event.target.value);
    }
    const submitHandler = (event) =>{
        event.preventDefault(); // to stop page realod
        // create object
        // console.log("Mehedi Hasna");
        const expenseData = {
            title: enteredTitle,
            amout: enteredAmount,
            date: new Date(enteredDate),
        };

        // console.log(expenseData);

        // props only use here to Send the data to Parent NewExpense.js <= NewExpenseFrom.js
        props.onSaveExpenseDate(expenseData);

        // New call to reset value set...with the help of <input vlaue={} />
        setEnteredTitle('');
        setEnteredAmount('');
        setEnteredDate('');
    }
    return (


            /*onSubmit={submitHandler}*/
            <from  >
            <div className="new-expense_controls">
                <div className="new-expense_control">
                    <label>Title</label>
                    <input type="text" value={enteredTitle} onChange={titleChangeHandler /* funcation call */} />
                </div>
                <div className="new-expense_control">
                    <label>Amount</label>
                    <input type="number" min="0.01" step="0.01"  value={enteredAmount} onChange = {amoutChangeHandler} />
                </div>
                <div className="new-expense_control">
                    <label>Date</label>
                    <input type="date" min="2022-01-01" max="2025-12-31" value={enteredDate} onChange = {dateChangeHandler} />
                </div>
            </div>
            <div className="new-expense_actions">
                <button type='submit' onClick={submitHandler} >Add Expense</button>
            </div>


        </from>

    );
};

export default NewExpenseFrom;

//************************** END **********************
```

```


//******************* EXpencesItem.css ***********
    /*
    *{
        margin: 0;
        padding: 0;
    }
    .container{
        width: 50%;
        height: 300px;
        margin: 0 auto;
    }
    .heading{
        font-size: 25px;
        background-color: gold;
    }

    /* we can use media query for verity devices */

//********************* END *********************
```