

ProjectDevelopmentPhas

eSprint – 3

SourceCode

Date	11November2022
TeamID	PNT2022TMID23145
ProjectName	PersonalExpenseTracker
BatchNumber	B9-3A5E

Navbar.jsx

```
import React from 'react';
import {useNavigate} from 'react-router';import './navbar.css';

constNavbar=()=>{

  constnavigate=useNavigate();

  consthandleLogout=()=>{navigate('/');
  }

  return(
    <divclassName="navbar">
      <divclassName="navTitle">PersonalExpenseTracker
        <spanclassName='navSubTitle'>speaktotrack</span>
      </div>
      <div>
        <buttonclassName='navBtn'onClick={()=>handleLogout()}>Logout</button>
      </div>
    </div>
  )
}
```

```
);  
}
```

```
export default Navbar;
```

navbar.css

```
.navbar{display:  
  flex;  
  flex-  
    direction:row;height:70  
    px;background-  
    color:white;  
    border: 3px  
    soliddarkgoldenrod;border-  
    radius:5px;  
    margin-  
    top:10px;align-  
    items:center;  
    justify-content:space-between;  
}
```

```
.navTitle{  
  font-  
    size:30px;margi  
    n-left:15%;text-  
    align:center;  
    color:rgb(187,134,0);f  
    ont-weight:bold;letter  
    -spacing:1px;  
}
```

```
.navSubTitle{
```

```

letter-
spacing:normal;font-
size:20px;
color:grey;margi
n-left:75px;
}

.navBtn{height:
  70%;
  margin-right:220px;
  background-
  color:rgb(228,227,194);border-
  radius:5px;
  border:2pxsolidrgb(111,95,0);
  font-weight:600;
  color:rgb(194,128,6);pa
  dding:10px20px;cursor:
  pointer;
}

```

Details.jsx

```

import React from 'react';
import {Card,CardContent} from '@mui/material';
import {Doughnut,Pie,PolarArea,Radar} from 'react-chartjs-
2';import './details.css';
import useTransactions from '../useTransactions';i
mport 'chart.js/auto';

const Details=({title})=>{

  /*const[doughnatC,setDoughnatC]=useState(true);
  const[polarAreaC,setPolarAreaC]=useState(false);

```

```

const[pieC,setPieC]=useState(false);
const[radarC,setRadarC]=useState(false);*/

const{chartData}=useTransactions(title);

console.log(chartData);

return(
  <div>
    <divstyle={{display:'flex',flexDirection:'row',gap:'5px',marginTop:'10px'}}>

      <Cardstyle={{}}className={title==='Income'?"income":"expense"}>
        {/*<divclassName="chartButtonContainer">
          <buttonclassName={title==='Income'?(doughnatC?"selectedIn":"btnIn"):(doughn
atC?"selectedEx":"btnEx")}
            onClick={()=>{setDoughnat
              C(true);setPolarAreaC(fal
                se);setPieC(false);setRad
                  arC(false);
            }}
          >Doughnat</button>
          <buttonclassName={title==='Income'?(polarAreaC?"selectedIn":"btnIn"):(polarAr
eaC?"selectedEx":"btnEx")}
            onClick={()=>{setDoughnat
              C(false);setPolarAreaC(tr
                ue);setPieC(false);setRad
                  arC(false);
            }}
          >PolarArea</button>

```

```

        <buttonclassName={title==='Income'? (pieC?"selectedIn":"btnIn"):(pieC?"selecte
dEx":"btnEx")}
            onClick={()=>{setDoughnat
                C(false);setPolarAreaC(f
                else);setPieC(true);setRa
                darC(false);
            }}
        >Pie</button>

        <buttonclassName={title==='Income'? (radarC?"selectedIn":"btnIn"):(radarC
?"selectedEx":"btnEx")}
            onClick={()=>{setDoughnat
                C(false);setPolarAreaC(f
                else);setPieC(false);setR
                adarC(true);
            }}
        >Radar</button>
    </div>*/}

    {/*<CardHeaderstyle={{textAlign:"center",}}title={title+":Rs."+total}/>*/}
    <CardContent>
        <Doughnutdata={chartData}/>
    </CardContent>
</Card>

<CardclassName={title==='Income'? "income":"expense"}>
    <CardContent>
        <PolarAreadata={chartData}/>
    </CardContent>
</Card>

<CardclassName={title==='Income'? "income":"expense"}>

```

```

    <CardContent>
      <Piedata={chartData}/>
    </CardContent>
  </Card>
  <CardclassName={title==='Income'?"income":"expense"}>
    <CardContent>
      <Radardata={chartData}/>
    </CardContent>
  </Card>

```

```

    { /*{doughnatC&&
      <Doughnutdata={chartData}/>
    }
    {polarAreaC&&
      <PolarAreadata={chartData}/>
    }
    {pieC&&
      <Piedata={chartData}/>
    }
    {radarC&&
      <Radardata={chartData}/>
    } */
  </div>
</div>

```

```

);
};

```

```

export default Details;

```

details.css

```
.income{
  border-top:10pxsolidrgba(0,255,0,0.7);
  border-bottom:10pxsolidrgba(0,255,0,0.7);
}
```

```
.expense{
  border-top:10pxsolidrgba(255,0,0,0.7);
  border-bottom:10pxsolidrgba(255,0,0,0.7);
}
```

useTransactions.js

```
import{useContext}from"react"

import{expenseCategories,incomeCategories,resetCategories}from"./constants/categories";

import{ExpenseTrackerContext}from"./context/context"
```

```
constuseTransactions=(title)=>{r
  esetCategories();
  const{transactions}=useContext(ExpenseTrackerContext);
  consttransactionsPerType=transactions.filter((t)=>t.type===title);
  consttotal=transactionsPerType.reduce((acc,currVal)=>acc+=currVal.amount,0);con
  stcategories=title==='Income'?incomeCategories:expenseCategories;
  varmonth="";
  varmonthIncomeTotal=[
    {m:"January",amount:0},
    {m:"February",amount:0},
    {m:"March",amount:0},
    {m:"April",amount:0},
    {m:"May",amount:0},
```



```

    {m:"June",amount:0},
    {m:"July",amount:0},
    {m:"August",amount:0},
    {m:"September",amount:0},
    {m:"October",amount:0},
    {m:"November",amount:0},
    {m:"December",amount:0},
];

varmonthExpenseTotal=[
    {m:"January",amount:0},
    {m:"February",amount:0},
    {m:"March",amount:0},
    {m:"April",amount:0},
    {m:"May",amount:0},
    {m:"June",amount:0},
    {m:"July",amount:0},
    {m:"August",amount:0},
    {m:"September",amount:0},
    {m:"October",amount:0},
    {m:"November",amount:0},
    {m:"December",amount:0},
];

//transactionsPerType.forEach((t)=>{
//  console.log(t.amount,t.category,t.type);
//})

transactionsPerType.forEach((t)=>{

    month=t.date.slice(5,7);

```

```

if(t.type==='Income'){switch(mont
h){
    case'01':monthIncomeTotal[0].amount+=t.amount;break;
    case'02':monthIncomeTotal[1].amount+=t.amount;break;
    case'03':monthIncomeTotal[2].amount+=t.amount;break;
    case'04':monthIncomeTotal[3].amount+=t.amount;break;
    case'05':monthIncomeTotal[4].amount+=t.amount;break;
    case'06':monthIncomeTotal[5].amount+=t.amount;break;
    case'07':monthIncomeTotal[6].amount+=t.amount;break;
    case'08':monthIncomeTotal[7].amount+=t.amount;break;
    case'09':monthIncomeTotal[8].amount+=t.amount;break;
    case'10':monthIncomeTotal[9].amount+=t.amount;break;
    case'11': monthIncomeTotal[10].amount
    +=t.amount;break;case'12':
    monthIncomeTotal[11].amount
    +=t.amount;break;default:break;
}
}
else if(t.type === 'Expense')
{switch(month){
    case    '01':    monthExpenseTotal[0].amount    +=
    t.amount;break;case '02': monthExpenseTotal[1].amount
    +=
    t.amount;break;case    '03':
    monthExpenseTotal[2].amount  +=  t.amount;break;case
    '04':    monthExpenseTotal[3].amount    +=
    t.amount;break;case '05': monthExpenseTotal[4].amount
    +=
    t.amount;break;case    '06':
    monthExpenseTotal[5].amount  +=  t.amount;break;case
    '07':    monthExpenseTotal[6].amount    +=
    t.amount;break;case '08': monthExpenseTotal[7].amount
    +=
    t.amount;break;case    '09':
    monthExpenseTotal[8].amount  +=  t.amount;break;case
    '10':    monthExpenseTotal[9].amount    +=

```

```
t.amount;break;case'11':monthExpenseTotal[10].amount  
+=t.amount;break;
```

```
      case '12': monthExpenseTotal[11].amount += t.amount; break;
      default: break;
    }
  }
}
```

```
const category = categories.find((c) => c.type === t.category);
```

```
if (category) category.amount += t.amount;
});
```

```
const filteredCategories = categories.filter((c) => c.amount > 0);
```

```
const chartData = {
  datasets: [
    {
      data: filteredCategories.map((c) => c.amount),
      backgroundColor: filteredCategories.map((c) => c.color),
    },
  ],
  labels: filteredCategories.map((c) => c.type)
};
```

```
const chartDataIncome = {
  datasets: [
    {
      label: 'Income',
    },
  ],
};
```

```

        data:monthIncomeTotal.map((m)=>m.amount
        t),borderColor:'#165f40',
        backgroundColor:'#0bc77e',te
        nsion:0.1,
    },

    ],
    labels:monthIncomeTotal.map((m)=>m.m)
};

const chartDataExpense={data
    sets:[
        {
            label:'Expense',
            data:monthExpenseTotal.map((m)=>m.amount),bor
            derColor:'#b50d12',
            backgroundColor:'#e57c58',te
            nsion:0.1,
        },
    ],
    labels:monthExpenseTotal.map((m)=>m.m)
};

    return{total,chartData,chartDataIncome,chartDataExpense};
};

export default useTransactions;

```

contextReducer.js

```
const contextReducer = (state, action) => {  
  let transactions;  
  
  switch (action.type) {  
    case 'DELETE_TRANSACTION':  
      transactions = state.filter((t) => t.id !== action.payload);  
  
      localStorage.setItem('transaction', JSON.stringify(transactions));  
  
      return transactions;  
  
    case 'ADD_TRANSACTION':  
      transactions = [action.payload, ...state];  
  
      localStorage.setItem('transaction', JSON.stringify(transactions));  
  
      return transactions;  
  
    default:  
      return state;  
  }  
}  
  
export default contextReducer;
```