

SOURCE CODE

Date	19November 2022
Team ID	PNT2022TMID44305
Project Name	Personal Expense Tracker
Batch No	B8-2A4E

```
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';
import '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>
      <div style={{ display: 'flex', flexDirection: 'row', gap: '5px', marginTop: '10px' }}>
        <Card style={{}} className={title === 'Income' ? "income" : "expense"} >
          {/*<div className="chartButtonContainer" >
            <button className={title === 'Income' ? (doughnatC ? "selectedIn" : "btnIn") :
(doughnatC ? "selectedEx" : "btnEx")} >
              onClick={() => {
                setDoughnatC(true);
                setPolarAreaC(false);
                setPieC(false);
                setRadarC(false);
              }}
            >Doughnat</button>
            <button className={title === 'Income' ? (polarAreaC ? "selectedIn" : "btnIn") :
(polarAreaC ? "selectedEx" : "btnEx")} >
              onClick={() => {
                setDoughnatC(false);
                setPolarAreaC(true);
                setPieC(false);
                setRadarC(false);
              }}
            >PolarArea</button>
            <button className={title === 'Income' ? (pieC ? "selectedIn" : "btnIn") : (pieC ?
"selectedEx" : "btnEx")} >
              onClick={() => {
                setDoughnatC(false);
```

```

        setPolarAreaC(false);
        setPieC(true);
        setRadarC(false);
    }}
    >Pie</button>
    <button className={title === 'Income' ? (radarC ? "selectedIn" : "btnIn") : (radarC ?
"selectedEx" : "btnEx")}
        onClick={() => {
            setDoughnatC(false);
            setPolarAreaC(false);
            setPieC(false);
            setRadarC(true);
        }}
    >Radar</button>
</div>*/}
{ /*<CardHeader style={{ textAlign: "center", }} title={title+": Rs. "+total} />*/}
<CardContent>
    <Doughnut data={chartData} />
</CardContent>
</Card>
<Card className={title === 'Income' ? "income" : "expense"} >
    <CardContent>
        <PolarArea data={chartData} />
    </CardContent>
</Card>
<Card className={title === 'Income' ? "income" : "expense"} >
    <CardContent>
        <Pie data={chartData} />
    </CardContent>
</Card>
<Card className={title === 'Income' ? "income" : "expense"} >
    <CardContent>
        <Radar data={chartData} />
    </CardContent>
</Card>

</div>
</div>
);
};
export default Details;

```

details.css

```

.income {
    border-top: 10px solid rgba(0, 255, 0, 0.7);
    border-bottom: 10px solid rgba(0, 255, 0, 0.7);
}
.expense {

```

```
border-top: 10px solid rgba(255, 0, 0, 0.7);
border-bottom: 10px solid rgba(255, 0, 0, 0.7);
}
```

DetailsTrack.jsx

```
import React from 'react';
import { Line } from 'react-chartjs-2';
import useTransactions from '../useTransactions';
import './detailsTrack.css';
const DetailsTrack = ({ title }) => {
  const { chartDataIncome, chartDataExpense } = useTransactions(title);
  const chartData = (title === 'Income') ? chartDataIncome : chartDataExpense;
  return (
    <div className='chartContainer'>
      <div className={title === 'Income' ? 'incomeChart' : 'expenseChart'} >
        <Line data={chartData} />
      </div>
    </div>
  );
};
export default DetailsTrack;
```

detailsTrack.css

```
.incomeChart {
  background-color: white;
  border: 5px solid rgb(147, 147, 147);
  margin: 20px 200px;
  padding: 50px;
  border-radius: 10px;
}
.expenseChart {
  background-color: white;
  border: 5px solid rgb(147, 147, 147);
  margin: 20px 200px;
  padding: 50px;
  border-radius: 10px;
}
```

main.jsx

```
import React from 'react';
import { Card, CardContent, Typography, Grid, Divider } from '@mui/material';
import './main.css';
import Form from './Form/Form';
import List from './List/List';
import { useContext } from 'react';
import { ExpenseTrackerContext } from '../context/context';
import InfoCard from './InfoCard';
```

```

import useTransactions from '../useTransactions';
const Main = () => {
  const { balance } = useContext(ExpenseTrackerContext);
  const { total: incomeTotal } = useTransactions("Income");
  const { total: expenseTotal } = useTransactions("Expense");
  return (
    <Card style={{width: '370px', padding: '10px', backgroundColor: "transparent" }}>
      {/*<CardHeader className="cardHeader" title="Personal Expense Tracker"
subheader="speak to track" />*/}
      <CardContent className="cardContent1">
        <Typography style={{ color: 'rgb(144 122 0)' }} align="center" variant="h6">
          Total Balance: <span style={{ color: 'goldenrod' }}>Rs.{balance}</span>
        </Typography>
        <Typography style={{ color: 'green' }} align="center" variant="h6">
          Income: <span style={{ color: 'rgba(0, 255, 0, 0.9)' }}>Rs.{incomeTotal}</span>
        </Typography>
        <Typography style={{ color: 'red' }} align="center" variant="h6">
          Expense: <span style={{ color: 'rgba(255, 0, 0, 0.8)' }}>Rs.{expenseTotal}</span>
        </Typography>
        <Typography variant="subtitle1" style={{ lineHeight: '1.5em', marginTop: '20px' }} >
          <InfoCard />
        </Typography>
        <Divider style={{ borderTop: '1px dashed gray', backgroundColor: 'white', paddingTop:
'15px' }} />
        <Form />
      </CardContent>
      <CardContent className="cardContent2">
        <Grid container spacing={2}>
          <Grid item xs={12}>
            <List />
          </Grid>
        </Grid>
      </CardContent>
    </Card>
  );
};
export default Main;

```

main.css

```

.cardHeader {
  text-align: center;
  border: 2px solid rgb(144 122 0);
  color: rgb(144 122 0);
  border-radius: 5px;
  background-color: white;
}
.cardContent1 {
  border: 2px solid rgb(144 122 0);
  border-radius: 5px;

```

```

    background-color: white;
  }
  .cardContent2 {
    border: 2px solid rgb(144 122 0);
    border-radius: 5px;
    background-color: white;
  }
}

```

List.jsx

```

import React from 'react';
import { List as MUIList, ListItem, ListItemAvatar, ListItemText, Avatar, ListItemSecondaryAction,
IconButton, Slide } from '@mui/material';
import CurrencyRupeeIcon from '@mui/icons-material/CurrencyRupee';
import DeleteIcon from '@mui/icons-material/Delete';
import { useContext } from 'react';
import { ExpenseTrackerContext } from '../..../context/context';
import { useState } from 'react';
import CustomizedSnackbar from '../..../Snackbar/Snackbar';
import './list.css';
const List = () => {
  const { deleteTransaction, transactions } = useContext(ExpenseTrackerContext);
  const [open, setOpen] = useState(false);
  return (
    <MUIList dense={false} className='listContainer'>
      <CustomizedSnackbar title="list" open={open} setOpen={setOpen} />

      {transactions.map((transaction) => (
        <Slide direction="down" in mountOnEnter unmountOnExit key={transaction.id}>
          <ListItem>
            <ListItemAvatar>
              <Avatar style={{backgroundColor: transaction.type === 'Income' ? 'green' : 'red'}}>
                <CurrencyRupeeIcon />
              </Avatar>
            </ListItemAvatar>
            <ListItemText primary={transaction.category}
secondary={`Rs.${transaction.amount} - ${transaction.date}`} />
            <ListItemSecondaryAction>
              <IconButton edge="end" aria-label="delete" onClick={() => { setOpen(true);
deleteTransaction(transaction.id) }} >
                <DeleteIcon />
              </IconButton>
            </ListItemSecondaryAction>
          </ListItem>
        </Slide>
      )))
    </MUIList>
  );
}

```

```
export default List;
```

ListStyles.js

```
import { makeStyles } from '@material-ui/core/styles';
import { red, green } from '@material-ui/core/colors';
export default makeStyles((theme) => ({
  avatarIncome: {
    color: '#fff',
    backgroundColor: green[500],
  },
  avatarExpense: {
    color: theme.palette.getContrastText(red[500]),
    backgroundColor: red[500],
  },
  list: {
    maxHeight: '240px',
    overflow: 'auto',
  },
})));
```

list.css

```
.avatarIncome {
  color: #fff;
  background-color: green;
}
.avatarExpense {
  color: #fff;
  background-color: red;
}
.listContainer {
  max-height: 240px;
  overflow: auto;
}
```

form.css

```
.createBtn {
  margin-top: 50px;
  margin-bottom: 20px;
  margin-left: 16px;
  width: 100%;
  border-radius: 10px;
  padding: 15px;
  border: 2px solid rgb(111 95 0);
  background-color: rgb(231 229 175);
  cursor: pointer;
  color: rgb(111 95 0);
  font-weight: 600;
```

```
}
```

Form.jsx

```
import React, { useContext, useEffect, useState } from 'react';
import { TextField, Typography, Grid, FormControl, InputLabel, Select, MenuItem, OutlinedInput }
from '@mui/material';
import { v4 as uuidv4 } from 'uuid';
import './form.css';
import { ExpenseTrackerContext } from '../..../context/context';
import { expenseCategories, incomeCategories } from '../..../constants/categories';
import formatDate from '../..../utils/formatDate';
import { useSpeechContext } from '@speechly/react-client';
import CustomizedSnackbar from '../..../Snackbar/Snackbar';
const initialState = {
  amount: "",
  category: "",
  type: 'Income',
  date: formatDate(new Date()),
}
const Form = () => {
  const [formData, setFormData] = useState(initialState);
  const { addTransaction } = useContext(ExpenseTrackerContext);
  const { segment } = useSpeechContext();
  const [open, setOpen] = useState(false);
  const [unfill, setUnfill] = useState(false);
  const createTransaction = () => {
    if (formData.category && formData.amount) {
      const transaction = { ...formData, amount: Number(formData.amount), id: uuidv4() }
      addTransaction(transaction);
      setFormData(initialState);
      setOpen(true);
    }
    else {
      setUnfill(true);
      /*swal("Unfulfilled Inputs", "Please fill all the fields", "warning");*/
    }
  }
  useEffect(() => {
    if (segment) {
      if (segment.intent.intent === 'add_expense') {
        setFormData({ ...formData, type: 'Expense' });
      }
      else if (segment.intent.intent === 'add_income') {
        setFormData({ ...formData, type: 'Income' });
      }
      else if (segment.isFinal && segment.intent.intent === 'create_transaction') {
        return createTransaction();
      }
      else if (segment.isFinal && segment.intent.intent === 'cancel_transaction') {
```

```

        return setFormData(initialState);
    }
    segment.entities.forEach((e) => {
        const category = `${e.value.charAt(0)}${e.value.slice(1).toLowerCase()}`;
        switch (e.type) {
            case 'amount':
                setFormData({ ...formData, amount: e.value });
                break;
            case 'category':
                if (incomeCategories.map((ic) => ic.type).includes(category)) {
                    setFormData({ ...formData, type: 'Income', category });
                }
                else if (expenseCategories.map((ec) => ec.type).includes(category)) {
                    setFormData({ ...formData, type: 'Expense', category });
                }
                break;
            case 'date':
                setFormData({ ...formData, date: e.value });
                break;
            default:
                break;
        }
    });
    if (segment.isFinal && formData.type && formData.category && formData.amount &&
formData.date) {
        createTransaction();
    }
}
// eslint-disable-next-line
}, [segment]);
const selectedCategories = formData.type === 'Income' ? incomeCategories : expenseCategories;
return (
    <Grid container spacing={2}>
        <CustomizedSnackbar title="form" open={open} setOpen={setOpen} />
        <CustomizedSnackbar title="unfilled" open={unfill} setOpen={setUnfill} />
        <Grid item xs={12}>
            <Typography align="center" variant="subtitle2" gutterBottom>
                {segment &&
                    segment.words.map((w) => w.value).join(" ")
                }
            </Typography>
        </Grid>
        <Grid item xs={6}>
            <FormControl fullWidth>
                <InputLabel>Type</InputLabel>
                <Select
                    value={formData.type}
                    onChange={(e) => setFormData({ ...formData, type: e.target.value })}
                    input={<OutlinedInput label="Type" />}
                >

```



```

        <MenuItem value="Income">Income</MenuItem>
        <MenuItem value="Expense">Expense</MenuItem>
      </Select>
    </FormControl>
  </Grid>
  <Grid item xs={6}>
    <FormControl fullWidth>
      <InputLabel>Category</InputLabel>
      <Select
        value={formData.category}
        onChange={(e) => setFormData({ ...formData, category: e.target.value })}
        input={<OutlinedInput label="Category" />}
      >
        {
          selectedCategories.map((c) =>
            <MenuItem key={c.type} value={c.type}>
              {c.type}
            </MenuItem>
          )
        }
      </Select>
    </FormControl>
  </Grid>
  <Grid item xs={6}>
    <TextField type="number" onKeyDown={(e) => { e.key === 'e' && e.preventDefault();
e.key === '-' && e.preventDefault(); e.key === '+' && e.preventDefault(); }} label="Amount"
fullWidth value={formData.amount} onChange={(e) => setFormData({ ...formData, amount:
e.target.value })} />
  </Grid>
  <Grid item xs={6}>
    <TextField type="date" label="Date" fullWidth value={formData.date} onChange={(e) =>
setFormData({ ...formData, date: formatDate(e.target.value) })} />
  </Grid>
  <button className="createBtn" onClick={createTransaction} >Create</button>
</Grid>
);
};
export default Form;

```

navbar.css

```

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

```

```

}
.navTitle {
  font-size: 30px;
  margin-left: 15%;
  text-align: center;
  color: rgb(187, 134, 0);
  font-weight: bold;
  letter-spacing: 1px;
}
.navSubTitle {
  letter-spacing: normal;
  font-size: 20px;
  color: grey;
  margin-left: 75px;
}
.navBtn {
  height: 70%;
  margin-right: 220px;
  background-color: rgb(228, 227, 194);
  border-radius: 5px;
  border: 2px solid rgb(111, 95, 0);
  font-weight: 600;
  color: rgb(194, 128, 6);
  padding: 10px 20px;
  cursor: pointer;
}
/*
.navContainer {
  display: flex;
  flex-direction: row;
}
.navItems {
  display: flex;
  flex-direction: row;
}
.logoutBtn, .profileBtn {
  border-radius: 10px;
  padding: 0px 20px 0px 20px;
  border: 2px solid rgb(111 95 0);
  background-color: rgb(231 229 175);
  cursor: pointer;
  color: rgb(111 95 0);
  font-weight: 600;
}
.navHeader {
  text-align: left;
  border: 2px solid rgb(144 122 0);
  color: rgb(144 122 0);
  border-radius: 5px;
  background-color: white;

```

```

    width: 100%;
  }
  .navTitle {
  } */

```

Navbar.jsx

```

import React from 'react';
import { useNavigate } from 'react-router';
import './navbar.css';
const Navbar = () => {
  const navigate = useNavigate();
  const handleLogout = () => {
    navigate('/');
  }
  return (
    <div className="navbar">
      <div className="navTitle">Personal Expense Tracker
        <span className='navSubTitle'>speak to track</span>
      </div>
      <div>
        <button className='navBtn' onClick={() => handleLogout()}>Logout</button>
      </div>
    </div>
  );
}
export default Navbar;

```

snackbar.css

```

.root {
  width: 100%;
}

```

Snackbar.jsx

```

import React from 'react'
import Snackbar from '@mui/material/Snackbar';
import MuiAlert from '@mui/material/Alert';
const CustomizedSnackbar = ({ title, open, setOpen }) => {
  const handleClose = (event, reason) => {
    if (reason === 'clickaway') return;

    setOpen(false);
  }
  return (
    <div>
      {title === 'list' &&
        <div className='root'>
          <Snackbar

```

```

        anchorOrigin={{ vertical: 'top', horizontal: 'right' }}
        open={open}
        autoHideDuration={3000}
        onClose={handleClose}
      >
        <MuiAlert onClose={handleClose} severity="success" elevation={6} variant="filled"
    >
      Category successfully Removed
    </MuiAlert>
  </Snackbar>
</div>
}
{title === 'form' &&
  <div className='root'>
    <Snackbar
      anchorOrigin={{ vertical: 'top', horizontal: 'right' }}
      open={open}
      autoHideDuration={3000}
      onClose={handleClose}
    >
      <MuiAlert onClose={handleClose} severity="success" elevation={6} variant="filled"
    >
      Category successfully Added
    </MuiAlert>
  </Snackbar>
</div>
}
{title === 'unfilled' &&
  <div className='root'>
    <Snackbar
      anchorOrigin={{ vertical: 'top', horizontal: 'right' }}
      open={open}
      autoHideDuration={3000}
      onClose={handleClose}
    >
      <MuiAlert onClose={handleClose} severity="warning" elevation={6}
variant="filled" >
        Unfulfilled fields..!!
      </MuiAlert>
    </Snackbar>
  </div>
}
</div>
);
};
export default CustomizedSnackbar;

```

SnackbarStyles.js

```
import { makeStyles } from '@material-ui/core/styles';
```

```
export default makeStyles((theme) => ({
  root: {
    width: '100%',
    '& > * + *': {
      marginTop: theme.spacing(2),
    },
  },
}));
```

index.js

```
export { default as Details } from './Details/Details';
export { default as Main } from './Main/Main';
export { default as SnackBar } from './SnackBar/SnackBar';
export { default as InfoCard } from './InfoCard';
```

InfoCard.jsx

```
import React from 'react';
const isIncome = Math.round(Math.random());
const InfoCard = () => {
  return (
    <div style={{ textAlign: 'center', padding: '0 10%', paddingBottom: '5%' }}>
      Try saying: <br />
      Add {isIncome ? 'Income ' : 'Expense '}
      for {isIncome ? 'Rs.100 ' : 'Rs.50 '}
      in Category {isIncome ? 'Business ' : 'House '}
      for {isIncome ? 'Monday ' : 'Friday '}
    </div>
  );
};
export default InfoCard;
```

categories.js

```
const incomeColors = ['#123123', '#154731', '#165f40', '#16784f', '#14915f', '#10ac6e', '#0bc77e', '#04e38d', '#00ff9d'];
const expenseColors = ['#b50d12', '#bf2f1f', '#c9452c', '#d3583a', '#dc6a48', '#e57c58', '#ee8d68', '#f79d79', '#ffae8a', '#cc474b', '#f55b5f'];
export const incomeCategories = [
  { type: 'Business', amount: 0, color: incomeColors[0] },
  { type: 'Investments', amount: 0, color: incomeColors[1] },
  { type: 'Extra income', amount: 0, color: incomeColors[2] },
  { type: 'Deposits', amount: 0, color: incomeColors[3] },
  { type: 'Lottery', amount: 0, color: incomeColors[4] },
  { type: 'Gifts', amount: 0, color: incomeColors[5] },
  { type: 'Salary', amount: 0, color: incomeColors[6] },
  { type: 'Savings', amount: 0, color: incomeColors[7] },
  { type: 'Rental income', amount: 0, color: incomeColors[8] },
];
```

```

export const expenseCategories = [
  { type: 'Bills', amount: 0, color: expenseColors[0] },
  { type: 'Car', amount: 0, color: expenseColors[1] },
  { type: 'Clothes', amount: 0, color: expenseColors[2] },
  { type: 'Travel', amount: 0, color: expenseColors[3] },
  { type: 'Food', amount: 0, color: expenseColors[4] },
  { type: 'Shopping', amount: 0, color: expenseColors[5] },
  { type: 'House', amount: 0, color: expenseColors[6] },
  { type: 'Entertainment', amount: 0, color: expenseColors[7] },
  { type: 'Phone', amount: 0, color: expenseColors[8] },
  { type: 'Pets', amount: 0, color: expenseColors[9] },
  { type: 'Other', amount: 0, color: expenseColors[10] },
];
export const resetCategories = () => {
  incomeCategories.forEach((c) => c.amount = 0);
  expenseCategories.forEach((c) => c.amount = 0);
};

```

context.js

```

import React, { useReducer, createContext } from 'react';
import contextReducer from './contextReducer';
const initialState = JSON.parse(localStorage.getItem('transaction')) || [];
export const ExpenseTrackerContext = createContext(initialState);
export const Provider = ({ children }) => {
  const [transactions, dispatch] = useReducer(contextReducer, initialState);
  const deleteTransaction = (id) => dispatch({ type: 'DELETE_TRANSACTION', payload: id });
  const addTransaction = (transaction) => dispatch({ type: 'ADD_TRANSACTION', payload:
transaction });
  const balance = transactions.reduce((acc, currVal) => (currVal.type === 'Expense' ? acc -
currVal.amount : acc + currVal.amount), 0);
  return (
    <ExpenseTrackerContext.Provider value={{
      deleteTransaction,
      addTransaction,
      transactions,
      balance,
    }}>
      {children}
    </ExpenseTrackerContext.Provider>
  );
};

```

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;

```

Dashboard.jsx

```

import { PushToTalkButton, PushToTalkButtonContainer } from '@speechly/react-ui';
import React from 'react';
import { Details, Main } from '../components';
import DetailsTrack from '../components/DetailsTrack/DetailsTrack';
import Navbar from '../components/Navbar/Navbar';
const Dashboard = () => {
    return (
        <div>
            <div style={{ display: 'flex', flexDirection: 'row' }}>
                { /* <Grid className={classes.grid} container spacing={0} alignItems="center"
                justifyContent="center" style={{ height: '100vh' }} >
                    <Grid item xs={12} sm={4} className={classes.main} >
                        <Main />
                    </Grid>
                    <Grid item xs={12} sm={3} className={classes.last} >
                        <Details title="Income" />
                        <Details title="Expense" />
                    </Grid>
                    <Grid item xs={12} sm={3} className={classes.desktop} >
                        <Details title="Income" />
                    </Grid>
                    <Grid item xs={12} sm={3} className={classes.last} >
                        <Details title="Expense" />
                    </Grid>
                </Grid> */ }
            <div>
                <Main />
            </div>
            <div>
                <div>
                    <Navbar />
                </div>
                <span>
                    <Details title="Income" />
                </span>
                <span>

```

```

        <Details title="Expense" />
      </span>
    </div>
    <PushToTalkButtonContainer>
      <PushToTalkButton />
    </PushToTalkButtonContainer>
  </div>
  <div>
    <DetailsTrack title="Income" />
    <DetailsTrack title="Expense" />
  </div>
</div>
);
};
export default Dashboard;

```

DashboardStyles.js

```

import { makeStyles } from '@material-ui/core/styles';
export default makeStyles((theme) => ({
  desktop: {
    [theme.breakpoints.up('sm')]: {
      display: 'none',
    },
  },
  mobile: {
    [theme.breakpoints.down('sm')]: {
      display: 'none',
    },
  },
  last: {
    [theme.breakpoints.down('sm')]: {
      marginBottom: theme.spacing(3),
      paddingBottom: '200px',
    },
  },
  grid: {
    '& > *': {
      margin: theme.spacing(2),
    },
  },
})));

```

forminput.css

```

.inputField {
  padding: 15px;
  margin: 10px 0px;
  border-radius: 10px;
  border: 1px solid rgb(122 103 0);
}

```



```

}
.formInput {
  display: flex;
  flex-direction: column;
  width: 280px;
}
.labelField {
  font-size: 12px;
  color: rgb(50, 50, 0);
}
.errorMessage {
  font-size: 10px;
  padding: 3px;
  color: red;
  display: none;
}
.inputField:invalid[focused="true"] {
  border: 1px solid red;
}
.inputField:invalid[focused="true"] ~ span {
  display: block;
}

```

FormInput.jsx

```

import React, { useState } from 'react';
import './formInput.css';
const FormInput = (props) => {
  const [focused, setFocused] = useState(false);

  const { label, errorMessage, onChange, id, ...inputProps } = props;
  const handleFocus = (e) => {
    setFocused(true);
  };
  return (
    <div className="formInput">
      <label className="labelField">{label}</label>
      <input
        {...inputProps}
        className="inputField"
        onChange={onChange}
        onBlur={handleFocus}
        onFocus={() => inputProps.name === "confirmPassword" && setFocused(true)}
        focused={focused.toString()} />
      <span className="errorMessage">{errorMessage}</span>
    </div>
  );
};
export default FormInput;

```

login.css

```
.login {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
}
.loginForm {
  background-color: rgb(244 228 140);
  padding: 0px 60px;
  border-radius: 10px;
}
.loginTitle {
  color: rgb(139 120 2);
  text-align: center;
  margin-top: 30px;
}
.loginBtn {
  width: 100%;
  height: 50px;
  padding: 10px;
  border: none;
  background-color: rgb(139 120 2);
  color: white;
  border-radius: 10px;
  font-weight: 600;
  font-size: 18px;
  cursor: pointer;
  margin-top: 15px;
  margin-bottom: 15px;
}
.loginBtn:hover {
  background-color: rgb(246 235 172);
  color: rgb(139 120 2);
  border: 2px solid rgb(139 120 2);
}
.registerLink {
  text-decoration: none;
  margin-left: 5px;
  color: rgb(72 62 1);
  font-weight: 500;
}
```

Login.jsx

```
import React, { useState } from 'react';
import { Link, useNavigate, } from 'react-router-dom';
import swal from 'sweetalert';
```

```

import FormInput from '../formInput/FormInput';
import './login.css';
const Login = () => {

  const navigate = useNavigate();
  const [values, setValues] = useState({
    username: "",
    password: "",
  });
  const { username, password } = values;
  const inputs = [
    {
      id: 1,
      name: "username",
      type: "text",
      placeholder: "Username",
      label: "Username",
      errorMessage: "Username should be 3-16 characters and shouldn't include any special
characters",
      required: true,
      pattern: "^[A-Za-z0-9]{3,16}$",
    },
    {
      id: 3,
      name: "password",
      type: "password",
      placeholder: "Password",
      label: "Password",
      errorMessage: "Password should be 8-20 characters and include at least 1 letter, 1 number
and 1 special character",
      required: true,
      pattern: "^(?=.*[0-9])(?=.*[A-Za-z])(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^&*]{8,20}$",
    },
  ]
  const handleSubmit = (e) => {
    e.preventDefault();
    if (username === "aaaa" && password === "aaaa@1111") {
      swal("Successfull Login", "Welcome to Dashboard", "success");
      navigate(`/dashboard/${username}`);
    }
    else {
      swal("Wrong Credentials", "Please try again!", "error");
    }
  };
  const onChange = (e) => {
    setValues({ ...values, [e.target.name]: e.target.value });
  };
  return (
    <div className="login">
      <form className="loginForm" onSubmit={handleSubmit}>

```

```

    <h1 className="loginTitle">Login</h1>
    {inputs.map((input) => (
      <FormInput
        key={input.id}
        {...input}
        value={values[input.name]} onChange={onChange}
      />
    ))}
    <button className="loginBtn" >Login</button>
  <div>
    <p style={{ fontSize: "14px", marginBottom: "30px" }}>New user? <Link
to="/register" className="registerLink">Click to Register</Link></p>
  </div>
</form>
</div>
);
};
export default Login;

```

register.css

```

.register {
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
}
.registerForm {
  background-color: rgb(244 228 140);
  padding: 0px 60px;
  border-radius: 10px;
}
.registerTitle {
  color: rgb(139 120 2);
  text-align: center;
  margin-top: 30px;
}
.registerBtn {
  width: 100%;
  height: 50px;
  padding: 10px;
  border: none;
  background-color: rgb(139 120 2);
  color: white;
  border-radius: 10px;
  font-weight: bold;
  font-size: 18px;
  cursor: pointer;
  margin-top: 15px;
  margin-bottom: 15px;
}

```

```

}
.registerBtn:hover {
  background-color: rgb(246 235 172);
  color: rgb(139 120 2);
  border: 2px solid rgb(139 120 2);
}
.loginLink {
  text-decoration: none;
  margin-left: 5px;
  color: rgb(72 62 1);
  font-weight: 500;
}

```

Register.jsx

```

import React, { useState } from 'react';
import { Link, useNavigate } from 'react-router-dom';
import FormInput from '../formInput/FormInput';
import './register.css';
const Register = () => {
  const navigate = useNavigate();
  const [values, setValues] = useState({
    username: "",
    email: "",
    password: "",
    confirmPassword: "",
  });
  const { username } = values;
  const inputs = [
    {
      id: 1,
      name: "username",
      type: "text",
      placeholder: "Username",
      label: "Username",
      errorMessage: "Username should be 3-16 characters and shouldn't include any special characters",
      required: true,
      pattern: "^[A-Za-z0-9]{3,16}$",
    },
    {
      id: 2,
      name: "email",
      type: "email",
      placeholder: "Email",
      label: "Email",
      errorMessage: "It should be a valid email address",
      required: true,
    },
  ]
}

```

```

      id: 3,
      name: "password",
      type: "password",
      placeholder: "Password",
      label: "Password",
      errorMessage: "Password should be 8-20 characters and include at least 1 letter, 1 number
and 1 special character",
      required: true,
      pattern: "^(?=.*[0-9])(?=.*[A-Za-z])(?=.*[@#$%^&*])[a-zA-Z0-9!@#$%^&*]{8,20}$",
    },
    {
      id: 4,
      name: "confirmPassword",
      type: "password",
      placeholder: "Confirm Password",
      label: "Confirm Password",
      errorMessage: "Passwords don't match",
      required: true,
      pattern: values.password,
    },
  ],
  const handleSubmit = (e) => {
    e.preventDefault();
    navigate(`/dashboard/${username}`);
  };
  const onChange = (e) => {
    setValues({ ...values, [e.target.name]: e.target.value });
  };
  return (
    <div className="register">
      <form className="registerForm" onSubmit={handleSubmit}>
        <h1 className="registerTitle">Register</h1>
        {inputs.map((input) => (
          <FormInput
            key={input.id}
            {...input}
            value={values[input.name]} onChange={onChange}
          />
        ))}
        <button className="registerBtn">Submit</button>
      </div>
      <p style={{ fontSize: "14px", marginBottom: "30px" }}>Already have an account?
      <Link to="/login" className="loginLink">Click to Login</Link></p>
    </div>
    </form>
  </div>
);
};
export default Register;

```

home.css

```
.homeContainer {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  height: 100vh;
}
.homeBtn {
  width: 280px;
  background-color: deepskyblue;
  border-radius: 10px;
  padding: 20px;
  margin: 15px;
  cursor: pointer;
  color: white;
  border: ridge white;
}
```

Home.jsx

```
import React from 'react';
import { Link } from 'react-router-dom';
import './home.css';
const Home = () => {
  return (
    <div className="homeContainer">
      <Link to="/login"><button className="homeBtn">Login</button></Link>
      <Link to="/register"><button className="homeBtn">Register</button></Link>
    </div>
  );
};
export default Home;
```

formateDate.js

```
const formatDate = (date) => {
  const d = new Date(date);
  let month = `${d.getMonth() + 1}`;
  let day = `${d.getDate()}`;
  const year = d.getFullYear();
  if (month.length < 2) {
    month = `0${month}`;
  }
  if (day.length < 2) {
    day = `0${day}`;
  }
  return [year, month, day].join('-');
}
export default formatDate;
```

App.css

```
.App {
  text-align: center;
}
.App-logo {
  height: 40vmin;
  pointer-events: none;
}
@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}
.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}
.App-link {
  color: #61dafb;
}
@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

App.js

```
import React from 'react';
import { BrowserRouter, Routes, Route, } from "react-router-dom";
import Dashboard from './routes/dashboard/Dashboard';
import Home from './routes/home/Home';
import Login from './routes/forms/login/Login';
import Register from './routes/forms/register/Register';
const App = () => {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Home />} />
      </Routes>
    </BrowserRouter>
  )
}
```



```

        <Route exact path="/login" element={<Login />} />
        <Route exact path="/register" element={<Register />} />
        <Route exact path="/dashboard/:uname" element={<Dashboard />} />
      </Routes>
    </BrowserRouter>
  );
};
export default App;

```

useTransactions.js

```

import { useContext } from "react"
import { expenseCategories, incomeCategories, resetCategories } from "../constants/categories";
import { ExpenseTrackerContext } from "../context/context"
const useTransactions = (title) => {
  resetCategories();
  const { transactions } = useContext(ExpenseTrackerContext);
  const transactionsPerType = transactions.filter((t) => t.type === title);
  const total = transactionsPerType.reduce((acc, currVal) => acc += currVal.amount, 0);
  const categories = title === 'Income' ? incomeCategories : expenseCategories;
  var month = "";
  var monthIncomeTotal = [
    {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},
  ];
  var monthExpenseTotal = [
    {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(month) {
      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 = {

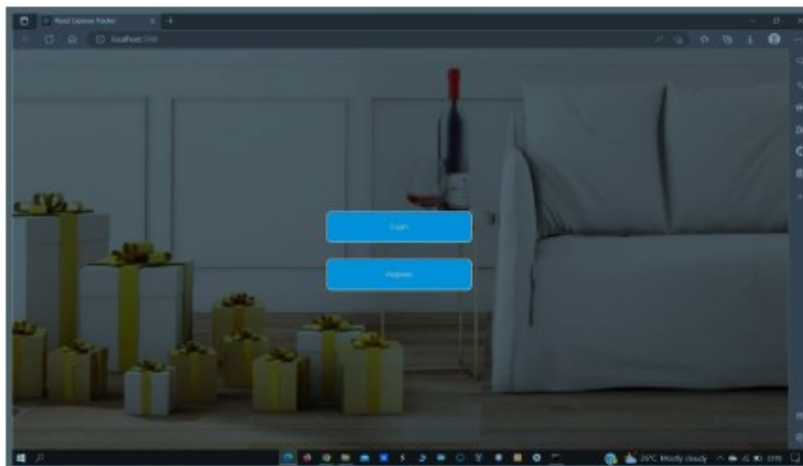
```

```

    datasets: [
      {
        label: 'Income',
        data: monthIncomeTotal.map((m) => m.amount),
        borderColor: '#165f40',
        backgroundColor: '#0bc77e',
        tension: 0.1,
      },
    ],
    labels: monthIncomeTotal.map((m) => m.m)
  };
const chartDataExpense = {
  datasets: [
    {
      label: 'Expense',
      data: monthExpenseTotal.map((m) => m.amount),
      borderColor: '#b50d12',
      backgroundColor: '#e57c58',
      tension: 0.1,
    },
  ],
  labels: monthExpenseTotal.map((m) => m.m)
};
return { total, chartData, chartDataIncome, chartDataExpense };
};
export default useTransactions;

```

OUTPUTS :



Register

Username

Email

Password

Confirm Password

Submit

Already have an account? Click to Login

Login

Username

Password

Login

Not user? Click to Register

Total Balance: **Rs. 13110**
Income: **Rs. 23991**
Expense: **Rs. 10871**

Try seeing:
Add Income for Rs.120 in Category Business for Monday

Type: **Income** Category: **Salary**

Amount: **28000** Date: **06-11-2022**

Create

Gifts **Rs. 23991 - 2022, 75.75**

Shipping **Rs. 120 - 2022, 11**

Car

