

## SPRINT – 1

Date	29/10/2022
Team ID	PNT2022TMID32597
Project Name	Personal Expense Tracker

### LOGIN :

```
import React from "react";
import Swal from 'sweetalert2';
import './style.css'
import CustomButton from "../../components/Button/button";
import Text from "../../components/Text/text";
import Input from "../../components/Input/input";
import { createSearchParams } from "react-router-dom";
export default class Login extends React.Component{
  render(){
    const notification = {
      toast:true,
      position: 'top-end',
      showConfirmButton: false,
      icon: 'error',
      timer:4000,
      timerProgressBar:true,
    }

    const navigation = this.props.navigation
    const login = async() =>{
      const email = document.getElementById('emailInput_login').value;
      const password =
document.getElementById('passwordInput_login').value;
      let url= new URL("http://localhost:5000/login")
      url.searchParams.set('email',email)
      url.searchParams.set('password',password)
      fetch(url).then((res)=>{
        res.json().then((data)=>{
          if(data.value===0){
            notification.title="Not a user"
            notification.text="Email id is not registered"
            Swal.fire(notification)
          }else if(data.value ===1){
```

```

        navigation({pathname
: '/expensetracker/dashboard', search: createSearchParams({email:email}).toString()})
    )

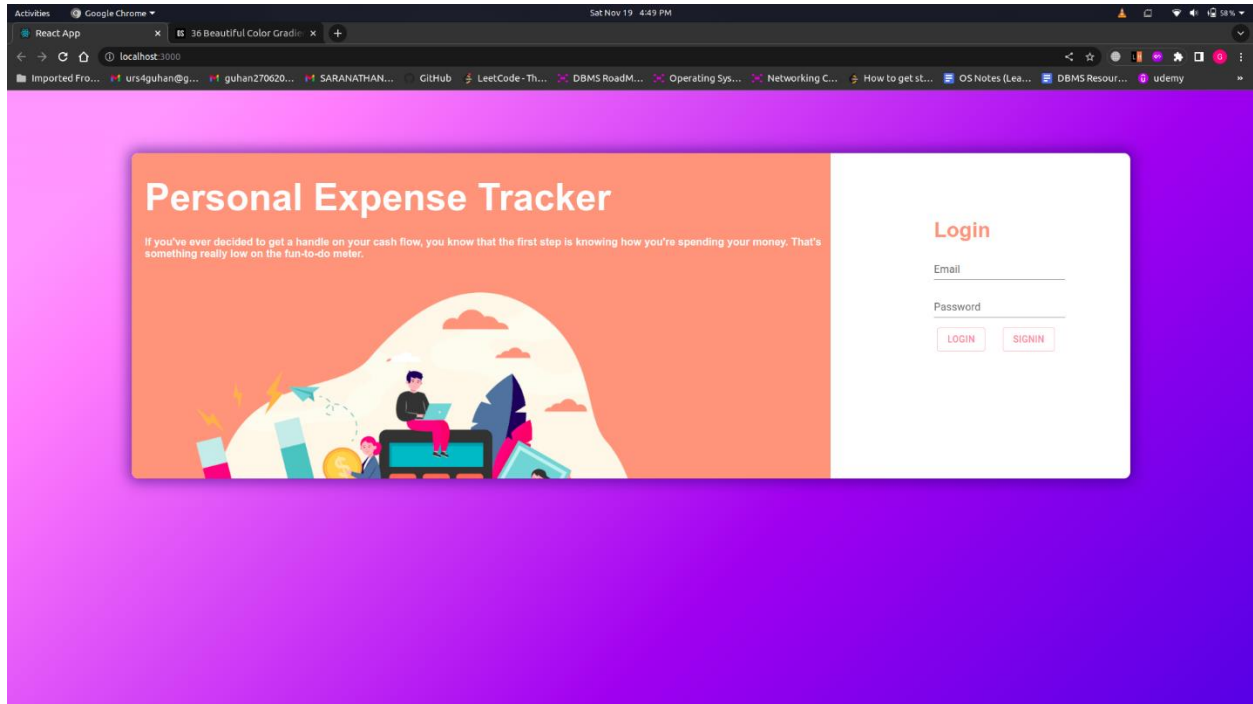
        }else if(data.value===2){
            notification.title="Invalid Password"
            notification.text="Please provide correct password"
            Swal.fire(notification)
        }else{
            notification.title="Connection Error"
            notification.text="Please try again later"
            Swal.fire(notification)
        }
    })
})
}

const signin = ()=>{
    navigation("/register")
}

return(
    <div>
        <div>
            <Text value="Login"/>
        </div>
        <Input id='emailInput_login' value="Email"/>
        <Input id='passwordInput_login' value="Password"/>
        <div className="buttonrow">
            <CustomButton function={login} value="Login"/>
            <CustomButton function={signin} value="Signin"/>
        </div>
    </div>
)
}
}

```

## LOGIN SCREENSHOT :



## REGISTER :

```
import React from "react";
import validator from "validator";
import Swal from 'sweetalert2';

import Text from "../../components/Text/text";
import CustomButton from "../../components/Button/button";
import Input from "../../components/Input/input";

export default class Register extends React.Component{
  render(){
    const navigation = this.props.navigation
    const register = async()=>{
      let notification = {
        toast:true,
        position: 'top-end',
        showConfirmButton: false,
        icon: 'error',
        timer:4000,
        timerProgressBar:true,
```

```

    }
    let fullname=document.getElementById("nameInput_register").value
    let validName = fullname.replace(/\s/g, '');
    let email=document.getElementById('emailInput_register').value
    let password=document.getElementById('passwordInput_register').value
    let
confirmPassword=document.getElementById('confirmPasswordInput_register').value
    if(!validator.isAlpha(validName)){
        notification.title = 'Invalid Name';
        notification.text= 'Please just provide alphabets';
        Swal.fire(notification)
    }else if(!validator.isEmail(email)){
        notification.title= 'Invalid Email';
        notification.text= 'Please just provide a proper email';
        Swal.fire(notification)
    }else if(!validator.isStrongPassword(password)){
        notification.title= 'Invalid Password';
        notification.text= 'Should have number, symbol, uppercase, length
> 8';

        Swal.fire(notification)
    }else if(password!==confirmPassword){
        notification.title= 'Password Mismatch';
        notification.text= 'Password must match';
        Swal.fire(notification)
    }else{
        let credentials = {
            name : fullname,
            email: email,
            password: password,
        }
        let url= new URL("http://localhost:5000/register")
        fetch(url,{
            method: 'POST',
            headers: {
                'Content-Type': 'application/json'
            },
            body: JSON.stringify(credentials)
        }).then((res)=>{
            res.json().then((data)=>{
                if(data.status === 200){
                    navigation('/')
                }else{
                    notification.toast=false
                    notification.position='center'
                    notification.title="Error "+200

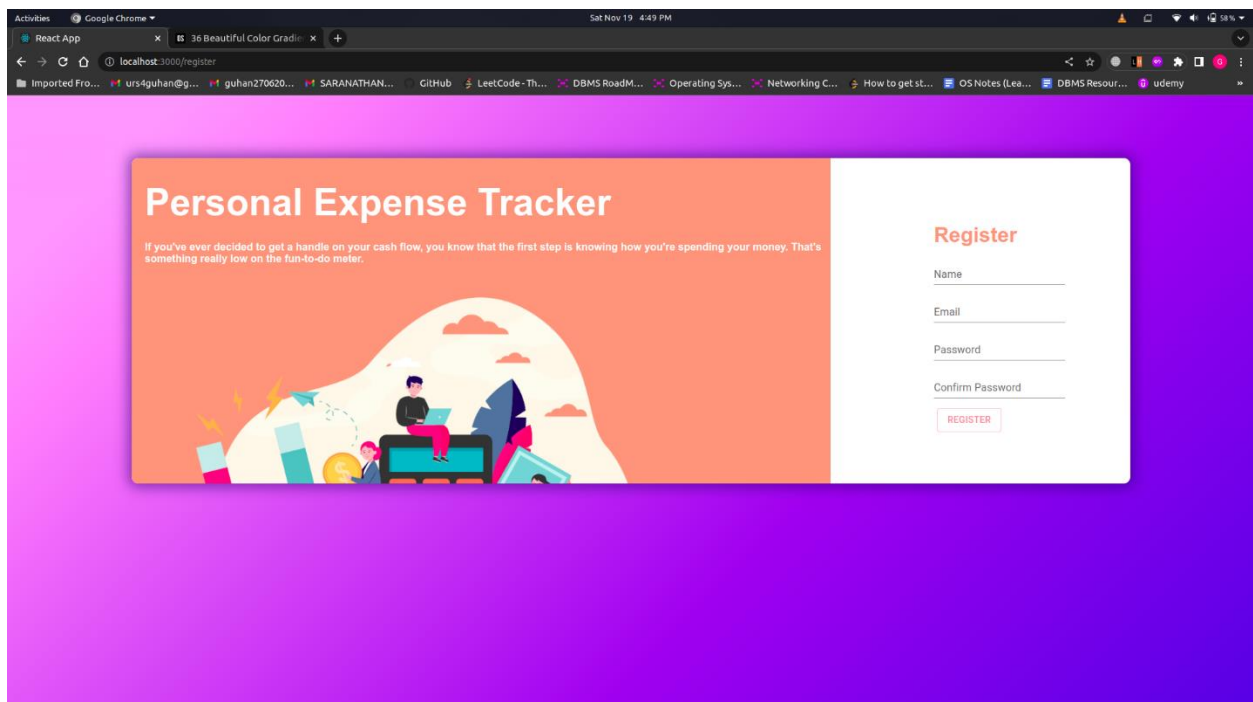
```

```

        notification.text="Please do retry"
        Swal.fire(notification)
    }
    })
  })
}
return(
  <div>
    <div>
      <Text value="Register"/>
    </div>
    <Input id="nameInput_register" value="Name"/>
    <Input id="emailInput_register" value="Email"/>
    <Input id="passwordInput_register" value="Password"/>
    <Input id="confirmPasswordInput_register" value="Confirm
Password"/>
    <CustomButton function={register} value="Register"/>
  </div>
)
}
}

```

## REGISTER SCREENSHOT :



## DASHBOARD :

```
import React, { useEffect, useState } from "react";
import { Bar, Doughnut } from 'react-chartjs-2';
// import LoadingButton from '@mui/lab/LoadingButton';
import {
  Chart as ChartJS,
  CategoryScale,
  LinearScale,
  BarElement,
  Title,
  Tooltip,
  ArcElement,
  Legend,
} from 'chart.js';
import { Paper, Grid, Box, Avatar, Divider, Typography, IconButton } from
"@mui/material";
import ShoppingCartCheckoutRoundedIcon from "@mui/icons-
material/ShoppingCartCheckoutRounded"
import AccountBalanceRoundedIcon from "@mui/icons-material/AccountBalanceRounded"
import PriceCheckRoundedIcon from "@mui/icons-material/PriceCheckRounded"
import CurrencyRupeeOutlinedIcon from "@mui/icons-material/CurrencyRupeeOutlined"
import AddIcon from '@mui/icons-material/Add';
import Expense from "../expense_page/expense_page";
import expensePage from "../expense_page/expense_page";
import Lottie from 'react-lottie';
import animationData from '../assets/loading.json';
import './style.css'
export default function Dashboard() {
  // constructor(props) {
  //   super(props);
  //   this.state = {
  //     bargraphdata: [],
  //     piegraphdata: [],
  //     totalExpense: 0,
  //     balance: 0,
  //     dailyExpense: 0,
  //     weeklyExpense: 0,
  //     isDataLoaded: false
  //   };
  // };
  // };
  const [bargraphdata, setbargraphdata] = useState([])
  const [bargraphlabel, setbargraphlabel] = useState([])
  const [piegraphdata, setpiegraphdata] = useState([])
```

```

const [piegraphlabel, setpiegraphlabel] = useState([])
const [totalExpense, settotalExpense] = useState(0)
const [balance, setbalance] = useState(0)
const [dailyExpense, setdailyExpense] = useState(0)
// const [weeklyExpense, setweeklyExpense] = useState(0)
const [isDataLoaded, setisDataLoaded] = useState(false)

const loadData = async () => {
  setisDataLoaded(false)
  let url = new URL("http://localhost:5000/loadData")
  url.searchParams.set('email', email)
  fetch(url).then((res) => {
    res.json().then((data) => {
      console.log(data.resultData)
      // this.state.bargraphdata = data.bargraphdata
      // this.state.piegraphdata = data.piegraphdata
      if (data.resultData.totalExpense < 1000)
        settotalExpense(data.resultData.totalExpense)
      else
        settotalExpense(Math.round(data.resultData.totalExpense /
1000) + "k")
      // this.state.balance = data.balance
      // this.state.dailyExpense = data.dailyExpense
      // this.state.weeklyExpense = data.weeklyExpense
      // setBargraphdata(data.bargraphdata)
      setpiegraphdata(data.resultData.piegraphdata)
      setpiegraphlabel(data.resultData.piegraphlabel)
      setbargraphdata(data.resultData.bargraphdata)
      setbargraphlabel(data.resultData.bargraphlabel)
      setdailyExpense(data.resultData.dailyExpense)
      setbalance(data.resultData.balance)
      setisDataLoaded(true)
    })
  })
}

useEffect(() => {
  //Runs only on the first render
  loadData()
}, []);
// state = {
//   bargraphdata: [],
//   piegraphdata: [],
//   totalExpense: 0,

```

```

//    balance: 0,
//    dailyExpense: 0,
//    weeklyExpense: 0,
//    isDataLoaded: false
// }
const inputurl = new URL(window.location.href)
const email = inputurl.searchParams.get('email')

// const expensePage = () => {
//    // return <>
//    //    // <Expense email="Hiiiiii"/>
//    //    // </>
//    //    // alert("HIIIIIIIIIIII")
// }
// const loadData = async () => {
//    let url = new URL("http://localhost:5000/loadData")
//    url.searchParams.set('email',this.email)
//    fetch(url).then((res) => {
//        res.json().then((data) => {
//            console.log(data)
//            // this.state.bargraphdata = data.bargraphdata
//            // this.state.piegraphdata = data.piegraphdata
//            // this.state.totalExpense = data.totalExpense
//            // this.state.balance = data.balance
//            // this.state.dailyExpense = data.dailyExpense
//            // this.state.weeklyExpense = data.weeklyExpense
//            this.setState({ isDataLoaded: !this.state.isDataLoaded })
//        })
//    })
// }
const expense_Page = async () => {
    expensePage(email, balance, loadData)
}

if (balance < 0) {

}

ChartJS.register(
    CategoryScale,
    LinearScale,
    BarElement,
    ArcElement,
    Title,
    Tooltip,

```



```

    Legend
  );

  const baroptions = {
    responsive: true,
    plugins: {
      legend: {
        position: 'bottom',
      },
      title: {
        display: false,
        // text: 'Chart.js Bar Chart',
      },
    },
    maintainAspectRatio: false,
    barThickness: 10,
    borderRadius: 5,
    // barPercen
  }

  const pieoptions = {
    plugins: {
      legend: {
        position: 'right',
      },
    },
    maintainAspectRatio: false
  }
  const piedata = {
    labels: piegraphlabel,
    datasets: [
      {
        label: '# of Votes',
        data: piegraphdata,
        backgroundColor: [
          'rgba(255, 99, 132, 0.2)',
          'rgba(54, 162, 235, 0.2)',
          'rgba(255, 206, 86, 0.2)',
          'rgba(75, 192, 192, 0.2)',
          'rgba(153, 102, 255, 0.2)',
          'rgba(255, 159, 64, 0.2)',
          'rgba(255, 159, 64, 0.2)',
          'rgba(255, 206, 86, 0.2)',
          'rgba(75, 192, 192, 0.2)',
        ],
      }
    ]
  }

```

```

        borderColor: [
            'rgba(255, 99, 132, 1)',
            'rgba(54, 162, 235, 1)',
            'rgba(54, 162, 235, 1)',
            'rgba(255, 206, 86, 1)',
            'rgba(75, 192, 192, 1)',
            'rgba(153, 102, 255, 1)',
            'rgba(255, 159, 64, 1)',
            'rgba(255, 206, 86, 1)',
            'rgba(75, 192, 192, 1)',
        ],
        borderWidth: 2,
    },
],
}

const bardata = {
    labels: bargraphlabel,
    datasets: [
        {
            label: 'Expense',
            data: bargraphdata,
            backgroundColor: 'rgba(53, 162, 235, 0.5)',
        },
    ],
};

window.watsonAssistantChatOptions = {
    integrationID: "0ce1b583-c49d-4981-89ce-01bc6c66bd9c", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "92fe30d2-43a9-4e1a-b852-8d5f379c97a2", // The ID of
your service instance.
    onLoad: function (instance) { instance.render(); }
};
setTimeout(function () {
    const t = document.createElement('script');
    t.src = "https://web-
chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
});

return (<>

```

```

{isLoading ?
  (<Box sx={{ flexGrow: 1 }} className="dashboard-page-container">
    <Grid container spacing={4}>
      <Grid item xs={3}>
        <Paper elevation={5} className="card-container" sx={{
height: 85 }}>
          <Grid container spacing={1}>
            <Grid item xs={8}>
              <Grid item xs={12}>
                <Typography variant="body1" gutterBottom>
                  Balance</Typography></Grid>
              <Grid item xs={12}>
                <Typography variant="h4" gutterBottom>
                  {balance}</Typography></Grid>
            </Grid>
            <Grid item xs={4}>
              <Avatar sx={{ bgcolor: "#5fc", height: 45,
width: 45, marginLeft: 3 }}>
                <CurrencyRupeeOutlinedIcon sx={{ width:
20, height: 20 }} />
              </Avatar>
            </Grid>
          </Grid>
        </Paper>
      </Grid>
      <Grid item xs={3}>
        <Paper elevation={2} className="card-container" sx={{
height: 85 }}>
          <Grid container spacing={1}>
            <Grid item xs={8}>
              <Grid item xs={12}>
                <Typography variant="body1" gutterBottom>
                  Total Expense</Typography></Grid>
              <Grid item xs={12}>
                <Typography variant="h4" gutterBottom>
                  {totalExpense}</Typography></Grid>
            </Grid>
            <Grid item xs={4}>
              <Avatar sx={{ bgcolor: "#5fc", height: 45,
width: 45, marginLeft: 3 }}>
                <PriceCheckRoundedIcon sx={{ width: 20,
height: 20 }} />
              </Avatar>
            </Grid>
          </Grid>
        </Paper>
      </Grid>
    </Grid>
  )
}

```

```

        </Paper>
      </Grid>
      <Grid item xs={3}>
        <Paper elevation={2} className="card-container" sx={{
height: 85 }}>
          <Grid container spacing={1}>
            <Grid item xs={8}>
              <Grid item xs={12}>
                <Typography variant="body1" gutterBottom>
                  Daily Expense</Typography></Grid>
              <Grid item xs={12}>
                <Typography variant="h4" gutterBottom>
                  {dailyExpense}</Typography></Grid>
              </Grid>
            <Grid item xs={4}>
              <Avatar sx={{ bgcolor: "#5fc", height: 45,
width: 45, marginLeft: 3 }}>
                <ShoppingCartCheckoutRoundedIcon sx={{
width: 20, height: 20 }} />
              </Avatar>
            </Grid>
          </Grid>
        </Paper>
      </Grid>
      <Grid item xs={3}>
        <Box onClick={() => expense_Page()}>
          <Paper elevation={2} className="card-container" sx={{
height: 85 }}>
            <Grid container spacing={1}>
              <Grid item xs={8}>
                <Typography variant="h6" gutterBottom>
                  Add Expense
                </Typography>
              </Grid>
              <Grid item xs={4}>
                {/* <Avatar sx={{ height: 75, width: 75,
}}> */}
                {/* <ShoppingCartCheckoutRoundedIcon
sx={{ width: 20, height: 20 }} /> */}
                <AddIcon sx={{ width: 50, height: 50,
color: "#f99", marginTop: 2 }} />
                {/* </Avatar> */}
              </Grid>
            </Grid>
          </Paper>
        </Box>
      </Grid>
    </Grid>
  </div>
)

```

```

        </Paper>
      </Box>
    </Grid>
    <Grid item xs={8}>
      <Paper elevation={2} className="bargraph-container">
        <Typography variant="h5" gutterBottom>
          Weekly Expense
        </Typography>
        <Divider />
        <Box sx={{ marginLeft: 4, width: 650, marginTop: 3
  >>
          <Bar data={bardata} options={baroptions}
height={275} width={10} />
        </Box>
      </Paper>
    </Grid>
    <Grid item xs={4}>
      <Paper elevation={2} className="piegraph-container">
        <Typography variant="h5" gutterBottom>
          Category Expense
        </Typography>
        <Divider />
        <Box sx={{ marginTop: 1 }}>
          <Doughnut data={piedata} options={pieoptions}
height={290} width={50} />
        </Box>
      </Paper>
    </Grid>
  </Grid>
</Box>) : (<Lottie
  options={{
    loop: true,
    autoplay: true,
    animationData: animationData,
    rendererSettings: {
      preserveAspectRatio: "xMidYMid slice"
    }
  }}
  height={400}
  width={400}
/>)}
</>
)

```

```
}
```

## DASEBOARD SCREENSHOT :

