

SPRINT 1

Flask API

```

from __future__ import print_function
from flask import Flask, request, jsonify
from flask_json import FlaskJSON, json_response
from flask_cors import CORS
import ibm_db

# Initializing flask app
app = Flask(__name__)
jsonObj = FlaskJSON(app)
cors = CORS(app,resources={r'*':{'origins':'http://localhost:3000'}})

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=0c77d6f2-5da9-48a9-81f8-
86b520b87518.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31198;Securit
y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=txm14386;PWD=SuJ9Kf0eI
JGynsB9;","","")

# Route for seeing a data
@app.route('/login')
def login():
    email = request.args.get('email')
    password = request.args.get('password')
    sql = "SELECT * FROM login where email = '{}'.format(email)"
    out = ibm_db.exec_immediate(conn, sql)
    document = ibm_db.fetch_assoc(out)
    if document == False :
        response = json_response(value=0)
    elif document['PASSWORD'] == password :
        response = json_response(value=1)
    else :
        response = json_response(value=2)
    return response

@app.route('/register',methods=['POST'])
def register():
    if request.method == "POST":
        credentials = json.loads(request.data)
        sql = "INSERT INTO login
VALUES('{}','{}').format(credentials['email'],credentials['password'])"
        out = ibm_db.exec_immediate(conn, sql)
        response = json_response(200)
        return response

# Running app

```

```
if __name__ == '__main__':
    app.run(debug=True)
```

Login Page

```
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";
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('/dashboard',{state:{email:email}})
          }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>
)
}
}

```

Register Page

```

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)
                    }
                })
            })
        }
    })

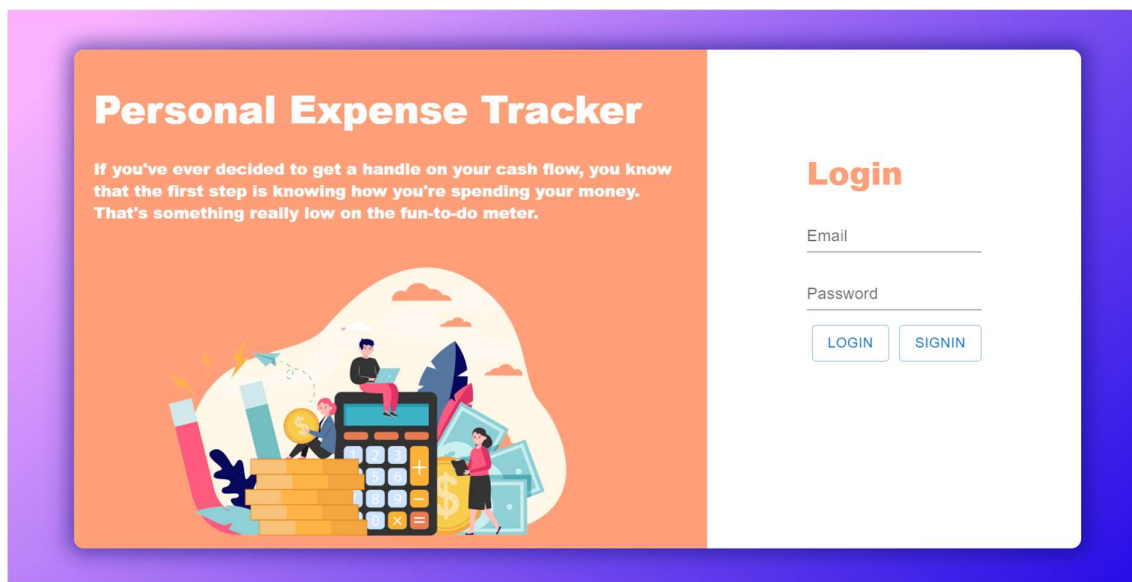
```

```

    })
  }
}
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>
)
}
}

```

Output



Personal Expense Tracker

If you've ever decided to get a handle on your cash flow, you know that the first step is knowing how you're spending your money. That's something really low on the fun-to-do meter.



Register

Name

Email

Password

Confirm Password

[REGISTER](#)