Flaunt Web Application

**Source Code**

* 1. Creating a user account

01.03 Use Account Authentication

**01.01 Creating a user account**

Front end code

import React, { Component } from 'react';

import useForm from './useForm';

import validate from './validateInfo';

import './UserRegistration.css'

import { NavLink } from 'react-router-dom';

import { Redirect } from 'react-router-dom';

const UserSignUp = () => {

    const { handleChange, values,handleSignUpSubmit, errors,email,status } = useForm(validate);

    return (

        <div>

            <section style={{ float: 'left' }}>

                <img src='/images/banner2.png' alt='BannerImg'></img>

            </section>

            <section>

                <h2>User Registration</h2>

                <form className="Form" onSubmit={handleSignUpSubmit}>

                    <section className=" bg-light" style={{ minHeight: '90vh', marginTop: '-90px' }}>

                        <div className="card shadow mx-auto" style={{ maxWidth: "400px", marginTop: "40px" }}>

                            <div className="card-body">

                                <div className="row gx-2 formInputs">

                                    <div className="col-6 mb-3">

                                        <label htmlFor='firstName' className="form-label">First name</label>

                                        <input

                                            type="text"

                                            id='firstName'

                                            className="form-control"

                                            name="firstName"

                                            placeholder='First Name'

                                            value={values.firstName}

                                            onChange={handleChange}

                                        />

                                        {errors.firstName && <p>{errors.firstName}</p>}

                                    </div>

                                    <div className="col-6 mb-3 formInputs">

                                        <label htmlFor='lastName' className="form-label">Last name</label>

                                        <input

                                            type="text"

                                            id='lastName'

                                            className="form-control"

                                            name="lastName"

                                            placeholder='Last Name'

                                            value={values.lastName}

                                            onChange={handleChange}

                                        />

                                        {errors.lastName && <p>{errors.lastName}</p>}

                                    </div>

                                </div>

                                <div className="mb-3 formInputs">

                                    <label htmlFor='email' className="form-label">Email</label>

                                    <input

                                        id='email'

                                        type="email"

                                        className="form-control"

                                        name="email"

                                        placeholder="Email"

                                        value={values.email}

                                        onChange={handleChange}

                                    />

                                    {errors.email && <p>{errors.email}</p>}

                                </div>

                                <div className="mb-3 formInputs">

                                    <label htmlFor='phone' className="form-label">Phone</label>

                                    <div className="row">

                                        <div> <input

                                            id='phone'

                                            name='phone'

                                            className="form-control"

                                            placeholder="Phone"

                                            type="number"

                                            value={values.phone}

                                            onChange={handleChange}

                                        />

                                            {errors.phone && <p>{errors.phone}</p>}

                                        </div>

                                    </div>

                                </div>

                                <div className="mb-3 formInputs">

                                    <label htmlFor='password' className="form-label">Password</label>

                                    <input

                                        id='password'

                                        type="password"

                                        className="form-control"

                                        name="password"

                                        placeholder="Password"

                                        value={values.password}

                                        onChange={handleChange}

                                    />

                                    {errors.password && <p>{errors.password}</p>}

                                </div>

                                <div className="mb-3 formInputs">

                                    <label htmlFor='confirmPassword' className="form-label">Confirm Password</label>

                                    <input

                                        id='confirmPassword'

                                        type="password"

                                        className="form-control"

                                        name="confirmPassword"

                                        placeholder="Confirm Password"

                                        value={values.confirmPassword}

                                        onChange={handleChange}

                                    />

                                    {errors.confirmPassword && <p>{errors.confirmPassword}</p>}

                                </div>

                                <div className="mb-4">

                                    {

                                        status

                                        ? <Redirect to={`/login`}> <button type="submit" className="btn btn-primary w-100"> Sign Up </button> </Redirect>

                                        :<button type="submit" className="btn btn-primary w-100"> Sign Up </button>

                                    }

                                </div>

                                <hr />

                                <p className="text-center mb-2">Already have account? <a href='#'>Sign in</a></p>

                            </div>

                        </div>

                    </section>

                </form>

            </section>

        </div>

    )

}

export default UserSignUp

Backend code

package com.flaunt.rest.controller;

import java.util.ArrayList;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RestController;

import com.flaunt.bean.UserRegistration;

import com.flaunt.service.UserRegistrationService;

import com.flaunt.util.common.Util;

//@CrossOrigin(origins = {"http://localhost:8080"})

@RestController

public class UserRegistrationController {

private static Logger flauntAPILog = LoggerFactory.getLogger("flauntAPI");

@Autowired

private UserRegistrationService userRegistrationService;

@PostMapping(path = "/registerUser")

public ResponseEntity<List<Object>> registerUser(@RequestBody UserRegistration userRegistration) {

flauntAPILog.info("Inside UserRegistrationController -> registerUser");

List<Object> result = new ArrayList<Object>();

result = userRegistrationService.registerUser(userRegistration);

flauntAPILog.info("Inside UserRegistrationController -> registerUser");

return new ResponseEntity<List<Object>>(result, HttpStatus.OK);

}

**01.03 Use Account Authentication**

Front end code

import React from 'react';

import { Link } from 'react-router-dom';

import useForm from './useForm';

import validate from './validateInfo';

import './UserLogin.css';

import axios, { Axios } from 'axios';

import { useState } from 'react';

import { Redirect } from 'react-router-dom';

export const UserLogin = () => {

    const { handleChange, values, handleSubmit, errors, message,email,status } = useForm(validate);

    const [getApiData, setApiData] = useState();

  return (

    <form className="Form" onSubmit={handleSubmit}>

                <section className="padding-y bg-light" style={{ minHeight: "90vh" }}>

                    <div className="container">

                        <div className="panel panel-default">

                        </div>

                        <div className="card shadow mx-auto" style={{ maxWidth: "400px", marginTop: "40px" }}>

                            <div className="card-body">

                                <h4 className="card-title mb-4">Login</h4>

                                {/\* <form> \*/}

                                <div className="mb-3">

                                    <label className="form-label">Email</label>

                                    <input type="email"

                                        className="form-control"

                                        name="email"

                                        placeholder="Email"

                                        value={values.email}

                                        onChange={handleChange}

                                    />

                                    {errors.email && <p style={{color:'red'}}>{errors.email}</p>}

                                    {/\* <label className="form-label" style={{color:'red'}}>Email Field is required</label> \*/}

                                </div>

                                <div className="mb-3">

                                    <label className="form-label">Password</label>

                                    <input type="password"

                                        className="form-control"

                                        name="password"

                                        placeholder="Password"

                                        value={values.password}

                                        onChange={handleChange}

                                    />

                                    {errors.password && <p style={{color:'red'}}>{errors.password}</p>}

                                    {/\* <label className="form-label" style={{ color: 'red' }}>(Password should atleast be 8 characters)</label> \*/}

                                </div>

                                <div className="mb-4">

                                    {

                                        status

                                        ? <Redirect to={`/home/${email}`}> <button type="submit" className="btn btn-primary w-100"> Login </button> </Redirect>

                                        :<button type="submit" className="btn btn-primary w-100"> Login </button>

                                    }

                                </div>

                                <div className="mb-3">

                                    <a href='#'>Forgot Password</a>

                                </div>

                                {/\* </form> \*/}

                                <hr />

                                <p className="text-center mb-2">Don't have an account? <Link to="/register"> Sign Up </Link></p>

                            </div>

                        </div>

                    </div>

                    <div className="message">{message ? <p>{message}</p> : null}</div>

                </section>

            </form>

  )

}

export default UserLogin

Backend code

package com.flaunt.rest.controller;

import java.util.ArrayList;

import java.util.List;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.http.HttpStatus;

import org.springframework.http.MediaType;

import org.springframework.http.ResponseEntity;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.flaunt.bean.Login;

import com.flaunt.service.LoginService;

//@CrossOrigin(origins = {"http://localhost:8082"})

@RestController

public class LoginController {

private static Logger flauntAPILog = LoggerFactory.getLogger("flauntAPI");

@Autowired

private LoginService loginService;

@PostMapping(value = "/login", consumes = {"application/json"})

public ResponseEntity<List<Object>> login(@RequestBody Login login) {

flauntAPILog.info("Inside LoginController -> login");

List<Object> result = new ArrayList<Object>();

result = loginService.login(login);

flauntAPILog.info("Inside LoginController -> login");

return new ResponseEntity<List<Object>>(result, HttpStatus.OK);

}

}

Front End Validation file

export default function validateInfo(values) {

    let errors = {}

    let validCharactersRegex = /^[a-z\u00C0-\u02AB'´`]+\.?\s([a-z\u00C0-\u02AB'´`]+\.?\s?)+$/;

    //Firstname

    if (!values.firstName.trim()) {

        errors.firstName = "Firstname required"

    }

    else if(!validCharactersRegex){

        errors.firstName ="Enter a valid Firstname"

    }

    //Lastname

    if (!values.lastName.trim()) {

        errors.lastName = "Lastname required"

    }

    //Email

    if (!values.email) {

        errors.email = "Email required"

    }

    else if (!/^([\w.%+-]+)@([\w-]+\.)+([\w]{2,})$/i.test(values.email)) {

        errors.email = "Email address is invalid"

    }

    //Phone

    if (!values.phone) {

        errors.phone = "Phone number is required"

    }

    // else if(values.phone.length < 10){

    //     errors.phone = "Number should be 10 digit"

    // }

    //Password

    if(!values.password){

        errors.password = "Password is required"

    }

    else if(values.password.length < 8){

        errors.password = "Password needs to be 8 characters or more"

    }

    //Confirm Password

    if(!values.confirmPassword){

        errors.confirmPassword = "Password is required"

    }

    else if(values.confirmPassword !== values.password){

        errors.confirmPassword = "Passwords do not match"

    }

    return errors;

}

Backend Spring boot Pom File

<project xmlns=*"http://maven.apache.org/POM/4.0.0"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>com.flaunt</groupId>

<artifactId>flauntAPI</artifactId>

<version>0.0.1-SNAPSHOT</version>

<properties>

<java.version>1.8</java.version>

</properties>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.2.6.RELEASE</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-jdbc</artifactId>

</dependency>

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.28</version>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-configuration-processor</artifactId>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.mchange</groupId>

<artifactId>c3p0</artifactId>

<version>0.9.5.2</version>

</dependency>

<dependency>

<groupId>commons-validator</groupId>

<artifactId>commons-validator</artifactId>

<version>1.7</version>

</dependency>

<!-- <dependency> <groupId>postgresql</groupId> <artifactId>postgresql</artifactId>

<version>9.1-901.jdbc4</version> </dependency> -->

</dependencies>

</project>