Part 3.

If the user name and password match the information stored in the MongoDB database, Node.js should take this information from the client and display all the user’s details from the database.

If the user name is new, allow the user to create a new record consisting of a user name and password, and store this record in the collection.

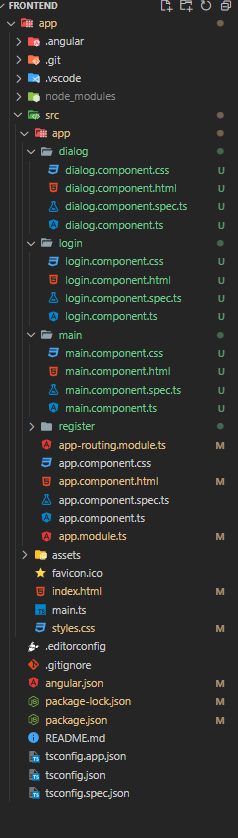
If the user name matches but the password does not, an error message should display

AngularJs:

app/src/app/register – generated component from command: *ng g c register*

app/src/app/main – generated component from command: *ng g c main*

app/src/app/dialog – generated component from command: *ng g c dialog*

**

app/src/app/register/register.component.ts:

import { HttpClient } from '@angular/common/http';

import { Component } from '@angular/core';

import { MatDialog } from '@angular/material/dialog';

import { DialogComponent } from '../dialog/dialog.component';

interface registerResponse {

  message?: string,

  statusCode?: string,

  data?: string

}

@Component({

  selector: 'app-register',

  templateUrl: './register.component.html',

  styleUrls: ['./register.component.css']

})

export class RegisterComponent {

  constructor(private http:HttpClient, private dialog:MatDialog){}

  onRegister(registerData: {username: string, password: string, first\_name: string, last\_name: string, age: number, gender: string, birth\_date: string}) {

    this.http.post("http://localhost:8000/user/register", registerData).subscribe((res) => {

      const response: registerResponse = res

      this.dialog.open(DialogComponent, {

        data: {

          header: response.statusCode == "200" ? "Welcome" : "Ops!",

          message: response.message,

          statusCode: response.statusCode,

          hideRegister: true

        },

        height: '200px',

        width: '500px',

      });

    })

  }

}

app/src/app/register/register.component.html:

<div fxLayoutAlign="center center" fxFlexFill>

  <mat-card fxFlex="25">

    <mat-toolbar color="warn">

      Register

    </mat-toolbar>

    <form class="form" #register="ngForm" (ngSubmit)="onRegister(register.value)" fxLayoutAlign="stretch" fxLayout="column">

      <mat-form-field appearance="fill">

        <mat-label>Username</mat-label>

        <input name="username" id="username" matInput ngModel>

      </mat-form-field>

      <mat-form-field appearance="fill">

        <mat-label>Password</mat-label>

        <input name="password" id="password" type="password" matInput ngModel>

      </mat-form-field>

      <mat-form-field appearance="fill">

        <mat-label>Firstname</mat-label>

        <input name="first\_name" id="first\_name" matInput ngModel>

      </mat-form-field>

      <mat-form-field appearance="fill">

        <mat-label>Lastname</mat-label>

        <input name="last\_name" id="last\_name" type="password" matInput ngModel>

      </mat-form-field>

      <mat-form-field appearance="fill">

        <mat-label>Age</mat-label>

        <input name="age" id="age" type="decimal" matInput ngModel>

      </mat-form-field>

      <mat-form-field appearance="fill">

        <mat-label>Gender</mat-label>

        <select matNativeControl required>

          <option value="" selected disabled>Select</option>

          <option value="male">Male</option>

          <option value="female">Female</option>

        </select>

      </mat-form-field>

      <mat-form-field appearance="fill">

        <mat-label>Birth Date</mat-label>

        <input name="birth\_date" id="birth\_date" type="date" matInput ngModel>

      </mat-form-field>

      <button type="submit" mat-raised-button color="primary">Register</button>

      <mat-divider>or</mat-divider>

      <a mat-raised-button  [routerLink]="['/']" target="">Back to Login</a>

    </form>

  </mat-card>

</div>

app/src/app/main/main.component.ts:

import { Component, OnInit } from '@angular/core';

import { Router } from '@angular/router';

interface dataResponse {

  age?: string,

  birth\_date?:string

  first\_name?:string

  gender?:string

  last\_name?:string

  username?:string

}

@Component({

  selector: 'app-main',

  templateUrl: './main.component.html',

  styleUrls: ['./main.component.css']

})

export class MainComponent implements OnInit{

  firstname: string | null | undefined = ""

  lastname: string | null | undefined= ""

  gender: string | null | undefined= ""

  age: string | null | undefined= ""

  bdate: string | null| undefined = ""

  constructor(private \_router: Router){}

  ngOnInit(){

    try {

      const data: dataResponse|null = JSON.parse(sessionStorage.getItem("data")||"");

      console.log(data)

      if (!data) {

        this.\_router.navigateByUrl("/")

      }

      this.firstname = data?.first\_name;

      this.lastname = data?.last\_name;

      this.gender = data?.gender;

      this.age = data?.age;

      this.bdate = data?.birth\_date;

    } catch (error) {

      this.\_router.navigateByUrl("/")

    }

  }

}

app/src/app/main/main.component.html :

<div fxLayoutAlign="center center" fxFlexFill>

  <mat-card fxFlex="25">

    <mat-toolbar color="warn">

      User Information

    </mat-toolbar>

    <form class="form" fxLayoutAlign="stretch" fxLayout="column">

        <p>Firstname: {{firstname}}</p>

        <p>Lastname: {{lastname}}</p>

        <p>age: {{age}}</p>

        <p>gender: {{gender}}</p>

        <p>Birth Date: {{bdate}}</p>

    </form>

  </mat-card>

</div>

app/src/app/dialog/dialog.component.ts :

import { Component, OnInit, Inject } from '@angular/core';

import {MAT\_DIALOG\_DATA} from '@angular/material/dialog';

@Component({

  selector: 'app-dialog',

  templateUrl: './dialog.component.html',

  styleUrls: ['./dialog.component.css']

})

export class DialogComponent implements OnInit {

  register: boolean = false;

  constructor(@Inject(MAT\_DIALOG\_DATA) public data: any){

    this.register = data.hideRegister;

  }

  ngOnInit(){

  }

}

app/src/app/dialog/dialog.component.html :

<h2 mat-dialog-title>{{data.header}}</h2>

<mat-dialog-content>

  {{data.message}}

</mat-dialog-content>

<mat-dialog-actions>

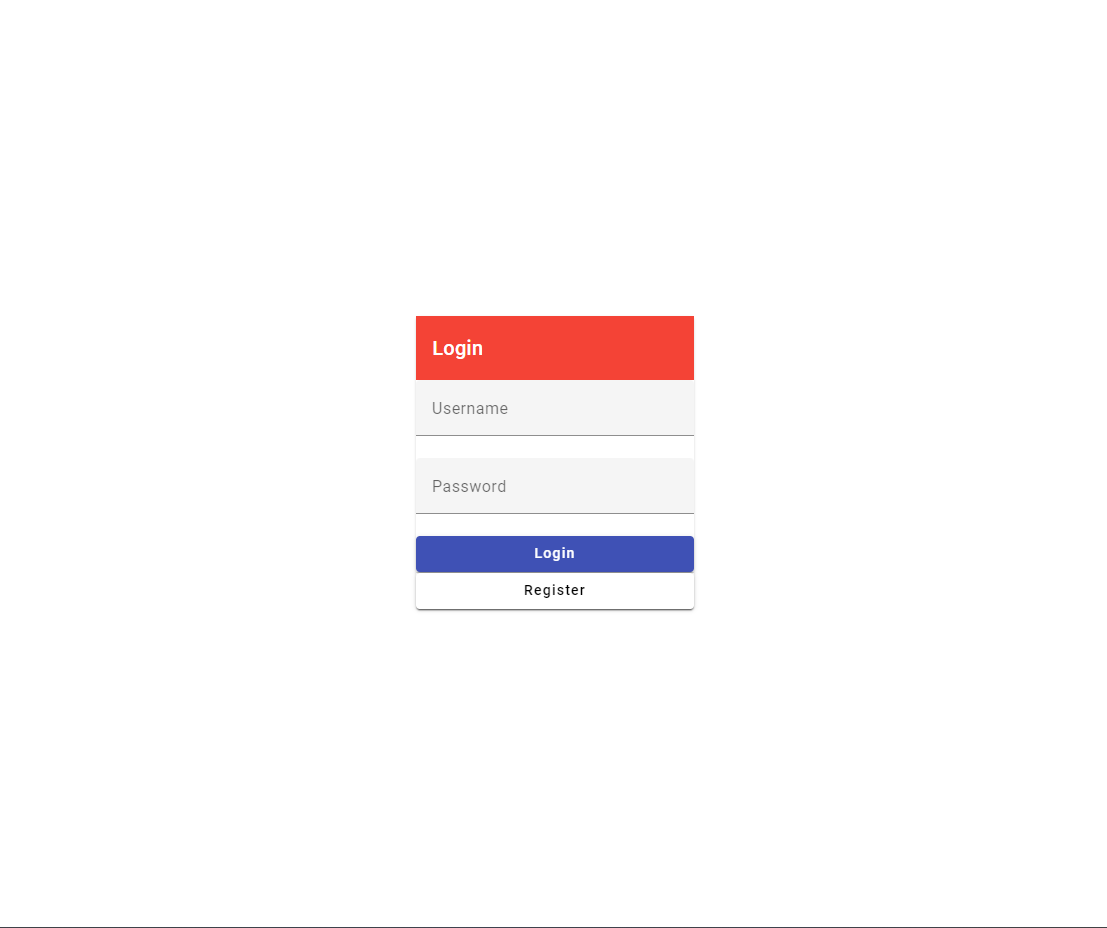
  <button  color="danger" mat-button mat-dialog-close>Close</button>

  <button color="primary" [hidden]="register"  mat-button [routerLink]="['/register']" mat-dialog-close>Register</button>

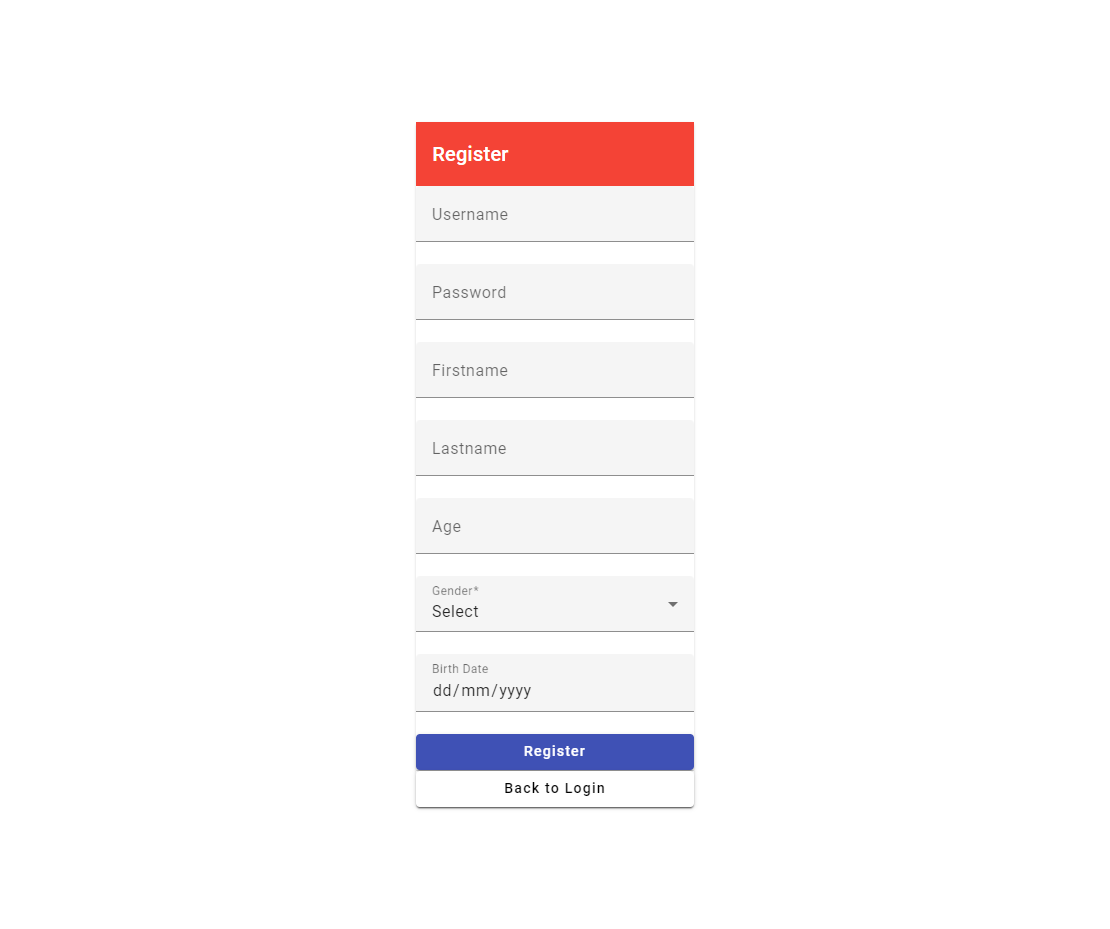
</mat-dialog-actions>

SCREEN SHOTS:

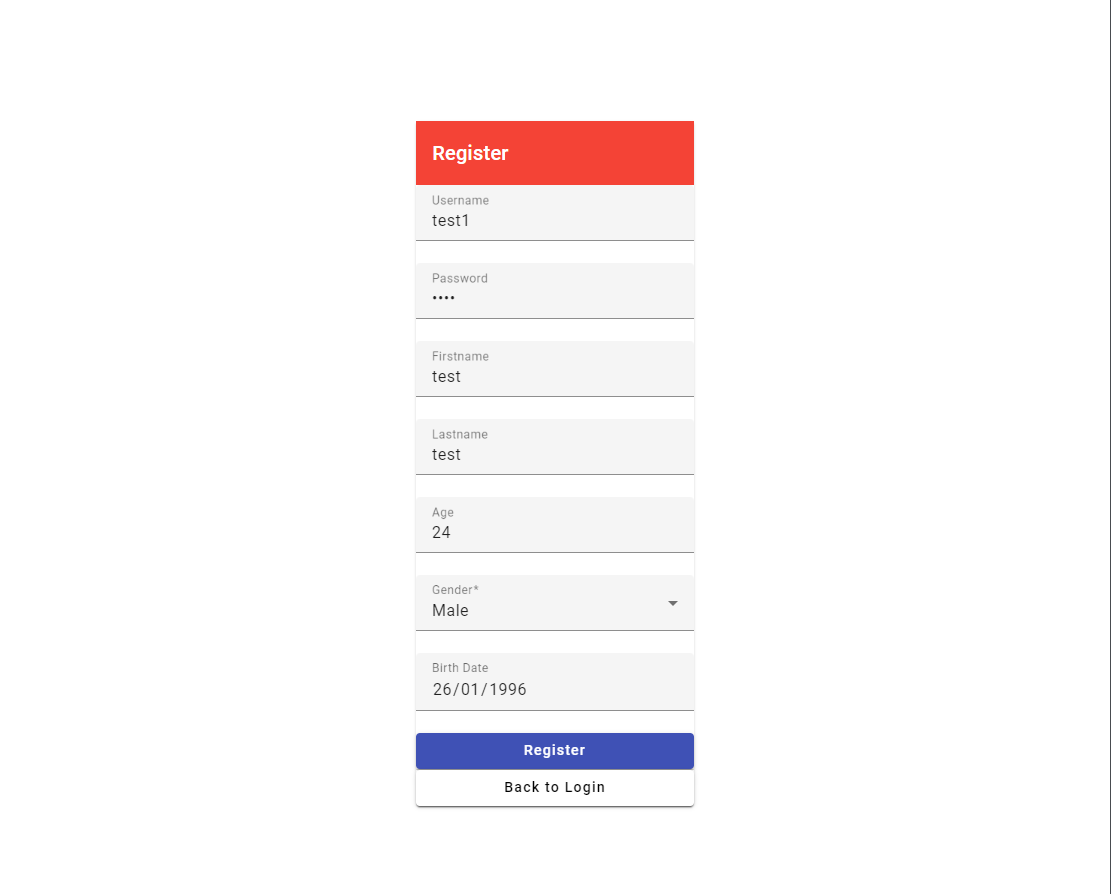
LOGIN FORM

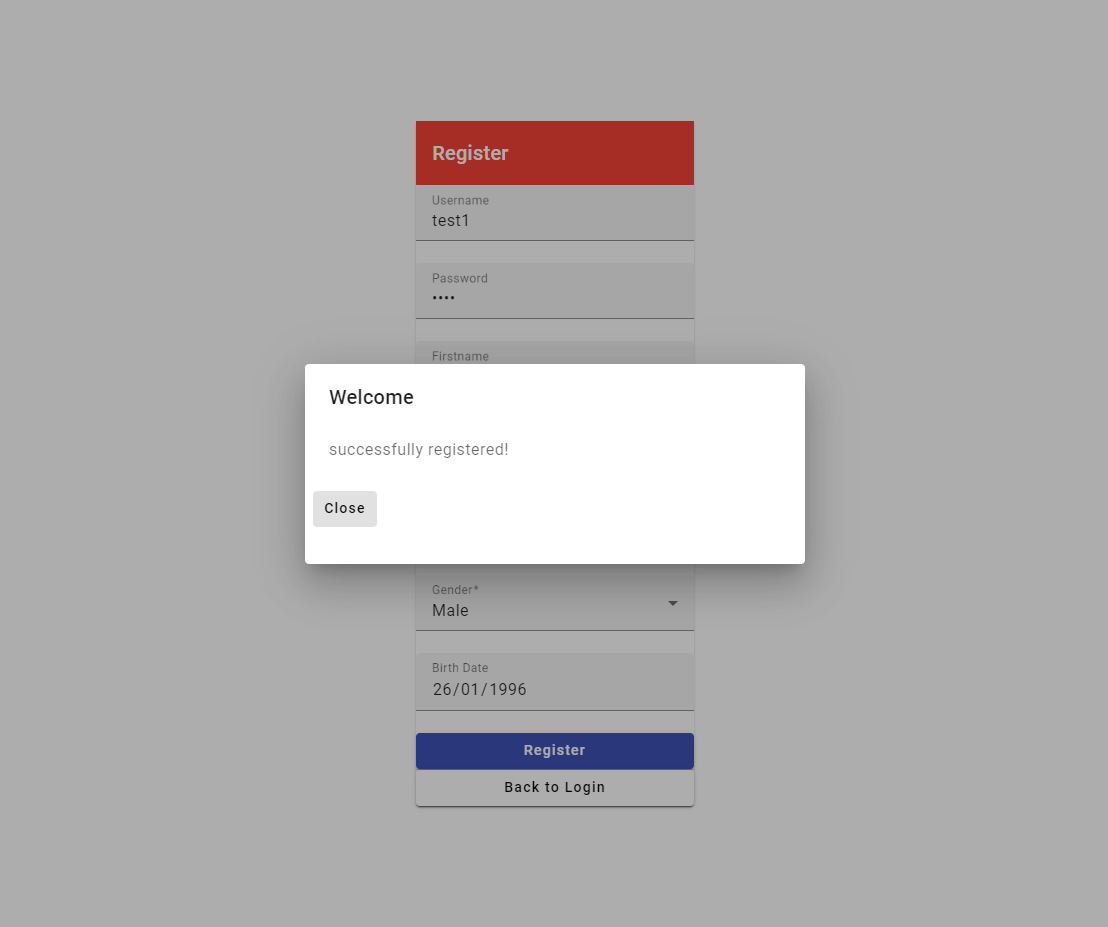


REGISTER FORM

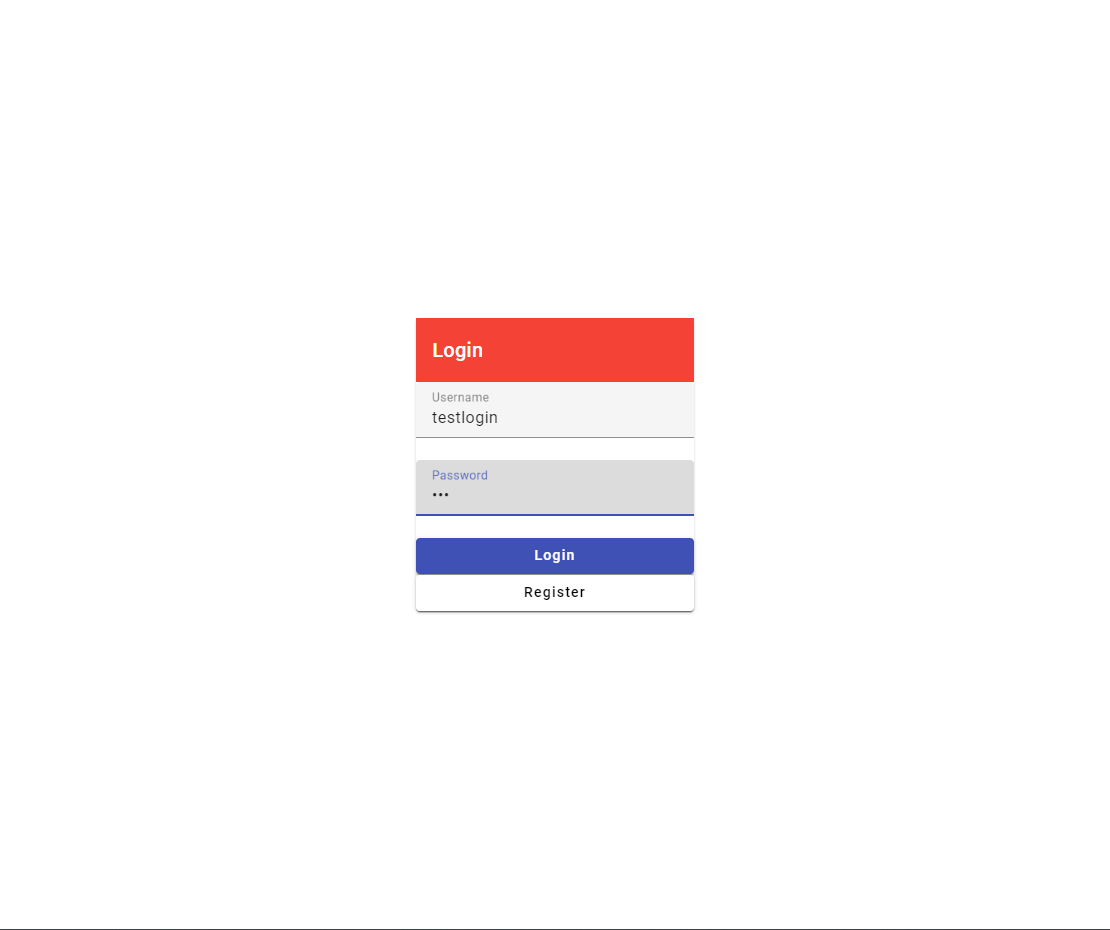


REGISTRATION TEST DATA

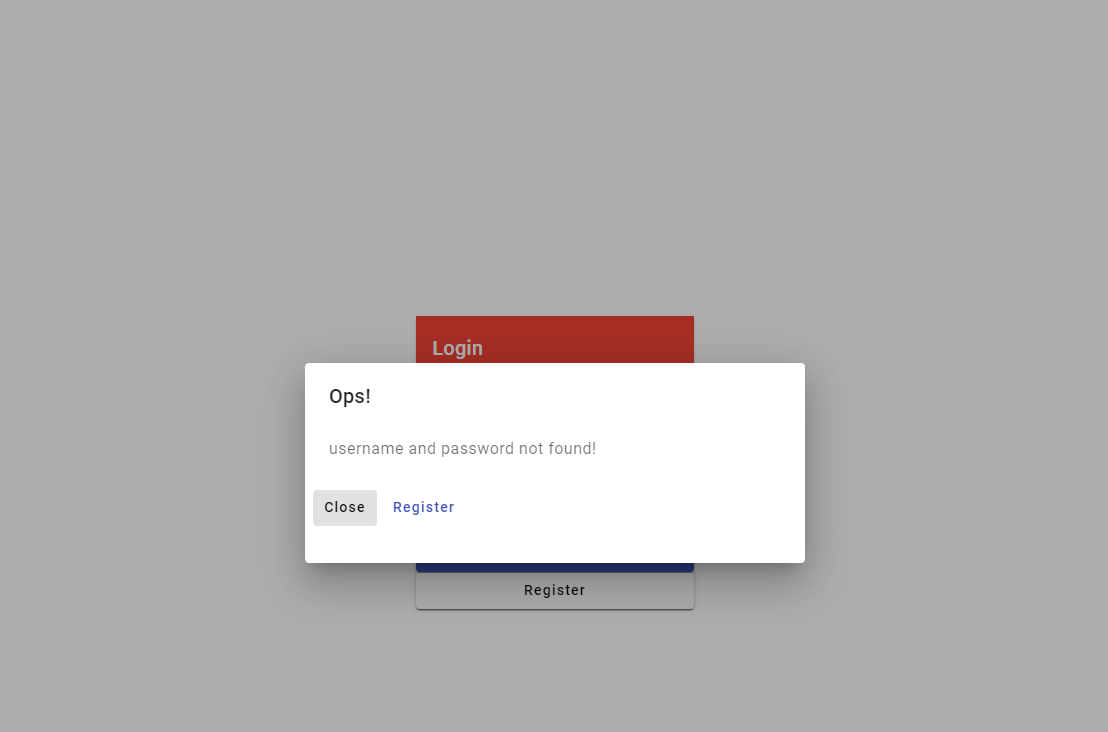


REGISTRATION RESPONSE

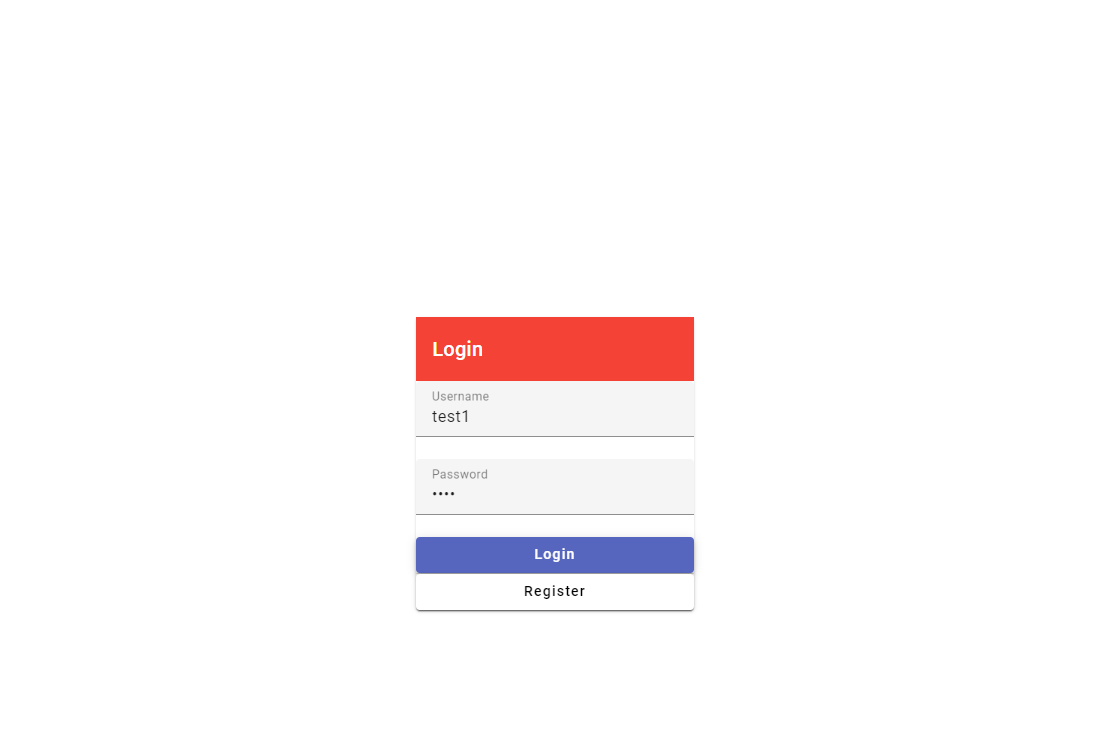
LOGIN TEST DATA WITH INVALID USERNAME and PASSWORD



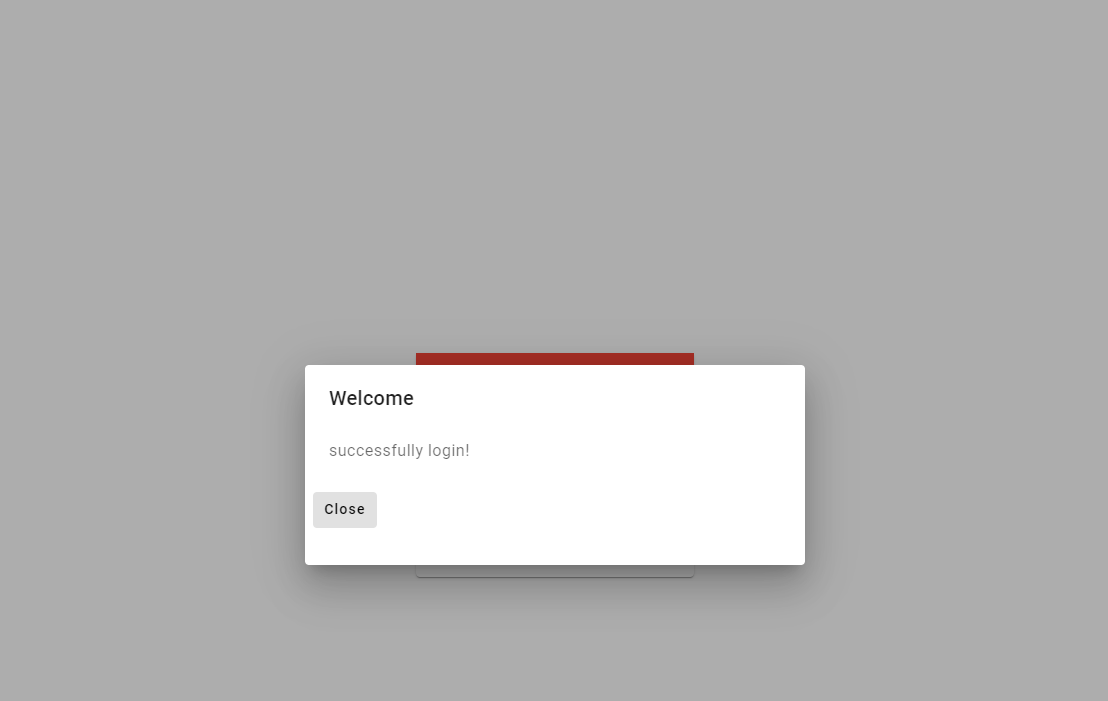
LOGIN TEST DATA WITH INVALID USERNAME and PASSWORD RESPONSE



LOGIN TEST DATA WITH VALID USERNAME and PASSWORD



LOGIN TEST DATA WITH VALID USERNAME and PASSWORD RESPONSE



SUCCESSFUL LOGIN WILL REDIRECT TO MAIN PAGE DISPLAYED THE USER INFORMATION

