

Code:

1. Login.component.html

```

<div style="width: 100%; display: flex; flex-direction: column; justify-content: center;
align-items: center; padding-top: 5%; ">

    <form (submit)="onSubmit()" id="loginForm" style="display: flex; flex-direction: column;
width: 28%;">

        <div class="dataContainer">

            <label>Email :</label>

            <input id="loginemail" name="email" type="email" [(ngModel)]="userData.email">

        </div>

        <div class="dataContainer">

            <label>Password :</label>

            <input id="loginpassword" name="password" type="password"
[(ngModel)]="userData.password">

        </div>

        <div class="dataContainer" style="gap:25px;margin-top: 20px; display: flex;flex-
direction: row;">

            <button id="submit" >Submit</button>

            <button (click)="Register()" >Register</button>

        </div>

        <p style="text-align: center; color: red">{{message}}</p>

    </form>

</div>

```

2. login.component.ts

```

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

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.css']
})
export class LoginComponent {
  userData = {
    email: '',
    password: '',
  };

  message = "";

  constructor(private router: Router) {

  }

  Register () {
    this.router.navigateByUrl('/register');
  }

  onSubmit() {
    var user = JSON.parse(localStorage.getItem("user") || "{}");

```

```

        this.userData.email = (<HTMLInputElement>document.getElementById("loginemail")).value;
        this.userData.password =
(<HTMLInputElement>document.getElementById("loginpassword")).value;
        if((this.userData.email===user.email) && (this.userData.password===user.password)){
            this.message = "Login Successful!!";
            alert(this.message);
            this.router.navigateByUrl('/profile');
        }
        else{
            this.message = "Invalid User";
        }
    }
}

```

3. profile.component.html

```

<div style="
width: 100%; display: flex; flex-direction: column; justify-content: center; align-items:
flex-start; padding-top: 10%; padding-left: 40%;">
    <div style="border: 3px solid black;padding: 25px;">
        <h1 style="color: red;font-size: 30px;">Your Information</h1>
        <p>Name: {{userData.name}}</p>
        <p>Email: {{userData.email}}</p>
    </div>
</div>

```

4. profile.component.ts

```

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

@Component({
    selector: 'app-profile',
    templateUrl: './profile.component.html',
    styleUrls: ['./profile.component.css']
})
export class ProfileComponent {
    userData = {
        name: '',
        email: '',
        password: '',
    };
    constructor() {
        var user = JSON.parse(localStorage.getItem("user")||"{}");
        this.userData.name = user.name;
        this.userData.email = user.email;
        this.userData.password = user.password;
    }
}

```

5. register.component.html

```

<div style="width: 100%; display: flex; flex-direction: column; justify-content: center;
align-items: center; padding-top: 5%; ">
    <form (submit)="onSubmit()" id="registrationForm" style="display: flex; flex-
direction: column; width: 28%;">
        <div class="dataContainer">
            <label>Name :</label>
            <input id="name" name="name" type="text" [(ngModel)]="userData.name">
        </div>
        <div class="dataContainer">
            <label>Email :</label>
            <input id="email" name="email" type="email" [(ngModel)]="userData.email">
        </div>
    </div>

```

```

        <label>Password :</label>
        <input id="password" name="password" type="password"
    [(ngModel)]="userData.password">
    </div>

    <div class="dataContainer" style="gap:25px;margin-top: 20px; display: flex;flex-
direction: row;">
        <button id="submit" >Register</button>

        <button (click)="Login()" >Login</button>
    </div>
</form>
</div>

```

6. register.component.spec.ts

```

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

@Component({
  selector: 'app-register',
  templateUrl: './register.component.html',
  styleUrls: ['./register.component.css']
})
export class RegisterComponent {
  userData = {
    name: '',
    email: '',
    password: '',
    age: '',
    dob: '',
    city: '',
    pincode: ''
  };

  constructor(private router: Router) {

  }

  Login () {
    this.router.navigateByUrl('/login');
  }

  onSubmit(){
    this.userData.name = (<HTMLInputElement>document.getElementById("name")).value;
    this.userData.email = (<HTMLInputElement>document.getElementById("email")).value;
    this.userData.password =
    (<HTMLInputElement>document.getElementById("password")).value;
    localStorage.setItem("user", JSON.stringify(this.userData));
    this.router.navigateByUrl('/profile');
  }
}

```

7. app-routing.module.ts

```

import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { RegisterComponent } from './register/register.component';
import { LoginComponent } from './login/login.component';
import { ProfileComponent } from './profile/profile.component';
import { HomeComponent } from './home/home.component';

const routes: Routes = [
  { path: '', component: LoginComponent },
  { path: 'register', component: RegisterComponent },
  { path: 'login', component: LoginComponent },

```

```

    { path: 'profile', component: ProfileComponent }
  ];

  @NgModule({
    imports: [RouterModule.forRoot(routes)],
    exports: [RouterModule]
  })
  export class AppRoutingModule { }

```

8. app.component.html.

```
<router-outlet></router-outlet>
```

9. app.component.spec.ts

```

import { TestBed } from '@angular/core/testing';
import { RouterTestingModule } from '@angular/router/testing';
import { AppComponent } from './app.component';

describe('AppComponent', () => {
  beforeEach(() => TestBed.configureTestingModule({
    imports: [RouterTestingModule],
    declarations: [AppComponent]
  }));

  it('should create the app', () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app).toBeTruthy();
  });

  it(`should have as title 'angular'`, () => {
    const fixture = TestBed.createComponent(AppComponent);
    const app = fixture.componentInstance;
    expect(app.title).toEqual('angular');
  });

  it('should render title', () => {
    const fixture = TestBed.createComponent(AppComponent);
    fixture.detectChanges();
    const compiled = fixture.nativeElement as HTMLElement;
    expect(compiled.querySelector('.content span')?.textContent).toContain('angular app is running!');
  });
}

```

10.app.component.ts

```

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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'Login';
}

```

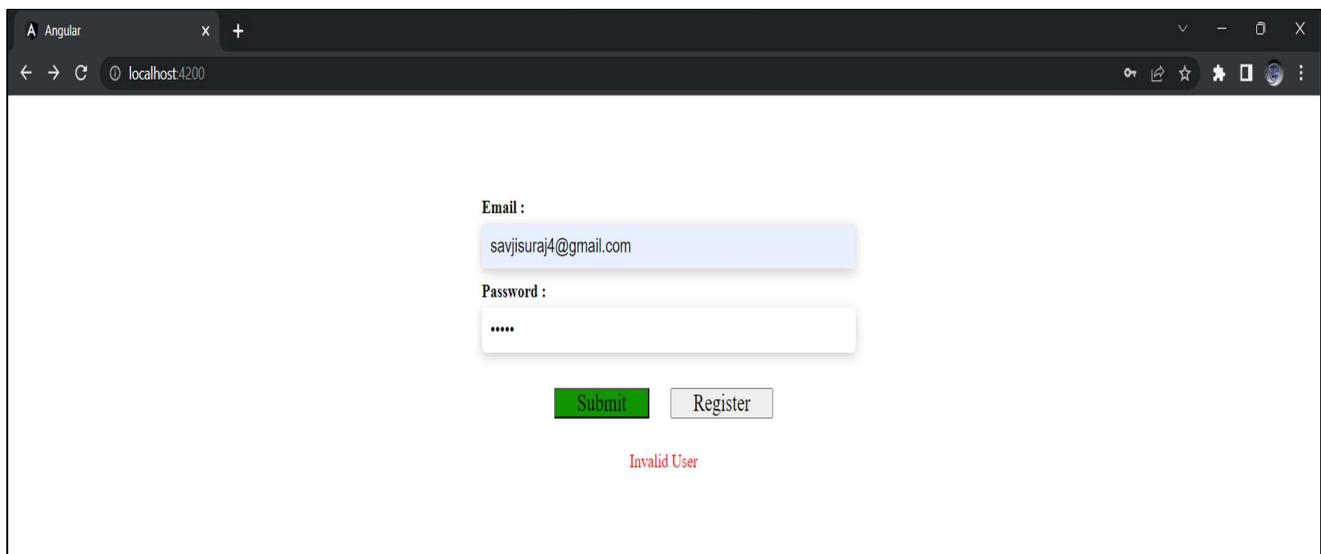
11.app.module.ts

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { RegisterComponent } from './register/register.component';
import { LoginComponent } from './login/login.component';
import { ProfileComponent } from './profile/profile.component';
import { FormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent,
    RegisterComponent,
    LoginComponent,
    ProfileComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

OUTPUT:



The screenshot shows a web browser window with the title 'Angular' and the address bar displaying 'localhost:4200'. The browser window contains a registration form with the following elements:

- Email :** A text input field containing the email address 'savjisoraj4@gmail.com'.
- Password :** A password input field with masked characters '.....'.
- Buttons:** Two buttons are located below the input fields: a green 'Submit' button and a grey 'Register' button.
- Error Message:** Below the buttons, the text 'Invalid User' is displayed in red.

Angular x +

localhost:4200/register

localhost:4200 says
User Added Successfully

OK

Name :
Suraj Santosh Savji

Email :
savjisuraj4@gmail.com

Password :

Register Login

Angular x +

localhost:4200/login

localhost:4200 says
Login Successful!!

OK

Email :
savjisuraj4@gmail.com

Password :

Submit Register

Angular x +

localhost:4200/profile

Your Information

Name: Suraj Santosh Savji
Email: savjisuraj4@gmail.com