Міністерство освіти і науки України

ВСП «Надвірнянський фаховий коледж НТУ»

Звіт

Для лабораторної роботи №4

З дисципліни «Розробка та програмування веб-застосувань»

На тему: «Маршрутизація на платформі Angular»

Виконав: студент групи КІ-11Б

Гандзюк В.В.

Перевірив: Волинський О.І.

Надвірна 2023р.

**Тема:** Маршрутизація на платформі Angular.

**Мета:** Ознайомлення з маршрутизацією на платформі Angular.

**Варіант:** 3

**Завдання:**

1. Доповнити існуючий застосунок, який розроблявся протягом трьох попередніх лабораторних робіт компонентами і сервісами, додати в кореневий компонент навігаційне меню. Для навігації між компонентами використати механізм маршрутизації.
2. Використати механізм “Охоронців” (guards).
3. В параметрах деяких маршрутів використати властивості CanActivate / CanDeactivate в яких прописати відповідних “Охоронців” (guards).
4. Розробити класи “Охоронців”, які будуть використовуватися в маршрутах.

**Хід роботи:**

В застосунку було змінене меню представлено на рис.1, це було зроблено для зручності подальшої роботи. Було створено ще 1 компонент в якому будуть представлені контакти, зображено на рис.2

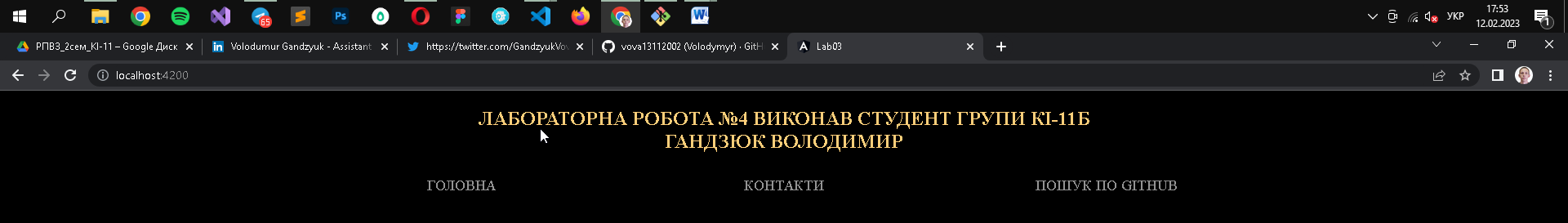


Рисунок 1 –­ Зміни в меню.



Рисунок 2 –­ Компонент «Контакти».

Було створено маршрутизацію для навігації по компонентам меню.

Також було створено двох охоронців представлено на рис.3-4, які мали властивість CanActivate, CanDeactivate.

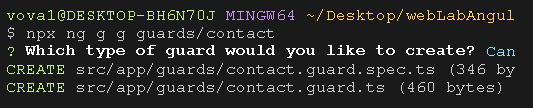


Рисунок 3 –­ Створення охоронця типу CanActivate.



Рисунок 4 –­ Створення охоронця типу CanDeactivate.

Список файлів в яких відбулися зміни і їхній код представлено нижче:

1. contact.guard.ts
2. exit.guard.ts
3. app.module.ts
4. app.component.html
5. app.component.scss
6. contactu.component.html
7. contactu.component.scss

Лістинг коду contact.guard.ts:

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

import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot, UrlTree } from '@angular/router';

import { Observable } from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class ContactGuard implements CanActivate {

canActivate(

route: ActivatedRouteSnapshot,

state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {

return confirm('Перейти?');

}

}

Лістинг коду exit.guard.ts:

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

import { ActivatedRouteSnapshot, CanActivate, CanDeactivate, RouterStateSnapshot, UrlTree } from '@angular/router';

import { Observable } from 'rxjs';

import { MainApiComponent } from '../main-api/main-api.component';

@Injectable({

providedIn: 'root'

})

export class ExitGuard implements CanDeactivate<MainApiComponent> {

canDeactivate(

component: MainApiComponent,

route: ActivatedRouteSnapshot,

state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {

return confirm("На сторінці є не збережені зміни. Продовжити?");

}

}

Лістинг коду 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 { HttpClientModule } from '@angular/common/http';

import { MainApiComponent } from './main-api/main-api.component';

import { NewServiceService } from './new-service.service';

import { FormsModule } from '@angular/forms';

import { ContactuComponent } from './contactu/contactu.component';

import { RouterModule, Routes } from '@angular/router';

import { ContactGuard } from './guards/contact.guard';

import { ExitGuard } from './guards/exit.guard';

const appRoutes: Routes = [

{

path: 'contactu',

component: ContactuComponent,

canActivate: [ContactGuard]

},//путь і компонент до Контактів

{

path: 'git',

component: MainApiComponent,

canDeactivate: [ExitGuard]

}//путь і компонент до Резюме

]

@NgModule({

declarations: [

AppComponent,

MainApiComponent,

ContactuComponent,

],

imports: [

BrowserModule,

AppRoutingModule,

HttpClientModule,

FormsModule,

RouterModule.forRoot(appRoutes)//які адреса будуть відслідовуватися і передаємо змінну з адресами

],

providers: [NewServiceService],

bootstrap: [AppComponent]

})

export class AppModule { }

Лістинг коду app.component.html:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<title>Lab04 Handziul Volodymyr</title>

<base href="/" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="icon" type="image/x-icon" href="favicon.ico" />

</head>

<body>

<div class="wrapper">

<header class="header">

<div class="container">

<div class="header\_\_title">

Лабораторна робота №4 Виконав студент групи КІ-11Б <br />Гандзюк

Володимир

</div>

<nav class="nav" data-nav>

<ul class="menu\_\_list">

<li class="menu\_\_list-item">

<a class="menu\_\_list-link" routerLink="/"> Головна </a>

</li>

<li class="menu\_\_list-item">

<a class="menu\_\_list-link" routerLink="/contactu"> Контакти </a>

</li>

<li class="menu\_\_list-item">

<a class="menu\_\_list-link" routerLink="/git">

Пошук по GitHub

</a>

</li>

</ul>

</nav>

</div>

</header>

</div>

</body>

</html>

<router-outlet></router-outlet>

Лістинг коду app.component.scss:

html {

box-sizing: border-box;

}

\*,

\*::after,

\*::before {

box-sizing: inherit;

}

ul[class],

ol[class] {

padding: 0;

}

body,

h1,

h2,

h3,

h4,

h5,

h6,

p,

ul,

ol,

li,

figure,

figcaption,

blockquote,

dl,

dd {

margin: 0;

}

ul[class] {

list-style: none;

}

img {

max-width: 100%;

display: block;

}

input,

button,

textarea,

select {

font: inherit;

}

a {

text-decoration: none;

}

body {

background-color: #000;

}

.wrapper {

}

.header {

&\_\_title {

padding-top: 20px;

text-align: center;

font-family: "Lab Grotesque";

font-weight: 700;

font-size: 24px;

line-height: 28px;

text-transform: uppercase;

color: #ffcf7a;

}

}

.container {

max-width: 1200px;

margin: 0 auto;

padding: 0 15px;

}

.nav {

padding-top: 30px;

padding-bottom: 30px;

}

.menu {

&\_\_list {

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 0 15px;

}

&\_\_list-item {

margin: 0 auto;

}

&\_\_list-link {

font-family: "Lab Grotesque";

font-weight: 500;

font-size: 18px;

line-height: 21px;

text-align: center;

text-transform: uppercase;

color: #949494;

&:hover {

color: #ffcf7a;

transition: ease 0.66s;

}

}

}

Лістинг коду contactu.component.html:

<div class="wrapper">

<div class="container">

<div class="contact">

<div class="card">

<div class="card-info">

<div class="card-avatar"></div>

<div class="card-title">Гандзюк Володимир</div>

<div class="card-subtitle">KI-11Б</div>

</div>

<ul class="card-social">

<li class="card-social\_\_item">

<svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">

<path></path>

</svg>

</li>

<li class="card-social\_\_item">

<a href="https://twitter.com/GandzyukVova" target="\_blank">

<svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">

<path></path>

</svg>

</a>

</li>

<li class="card-social\_\_item">

<a

href="https://www.linkedin.com/in/volodumur-gandzyuk-196a5422b/"

target="\_blank">

<svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">

<path></path>

</svg>

</a>

</li>

</ul>

</div>

</div>

</div>

</div>

Лістинг коду contactu.component.scss:

.contact {

display: flex;

justify-content: center;

padding-top: 25px;

}

.card {

width: 190px;

height: 254px;

background: #f5f5f5;

padding: 2rem 1.5rem;

transition: box-shadow 0.3s ease, transform 0.2s ease;

}

.card-info {

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

transition: transform 0.2s ease, opacity 0.2s ease;

}

/\*Image\*/

.card-avatar {

--size: 60px;

background: linear-gradient(to top, #f1e1c1 0%, #fcbc97 100%);

width: var(--size);

height: var(--size);

border-radius: 50%;

transition: transform 0.2s ease;

margin-bottom: 1rem;

}

/\*Card footer\*/

.card-social {

transform: translateY(200%);

display: flex;

justify-content: space-around;

width: 100%;

opacity: 0;

transition: transform 0.2s ease, opacity 0.2s ease;

}

.card-social\_\_item {

list-style: none;

}

.card-social\_\_item svg {

display: block;

height: 18px;

width: 18px;

fill: #515f65;

cursor: pointer;

transition: fill 0.2s ease, transform 0.2s ease;

}

/\*Text\*/

.card-title {

color: #333;

font-size: 1.5em;

font-weight: 600;

line-height: 2rem;

text-align: center;

}

.card-subtitle {

color: #859ba8;

font-size: 0.8em;

}

/\*Hover\*/

.card:hover {

box-shadow: 0 8px 50px #23232333;

}

.card:hover .card-info {

transform: translateY(-5%);

}

.card:hover .card-social {

transform: translateY(100%);

opacity: 1;

}

.card-social\_\_item svg:hover {

fill: #232323;

transform: scale(1.1);

}

.card-avatar:hover {

transform: scale(1.1);

}

Наступним кроком за допомогую команди ng serve –o запускає сервер, зображено на рис 5.

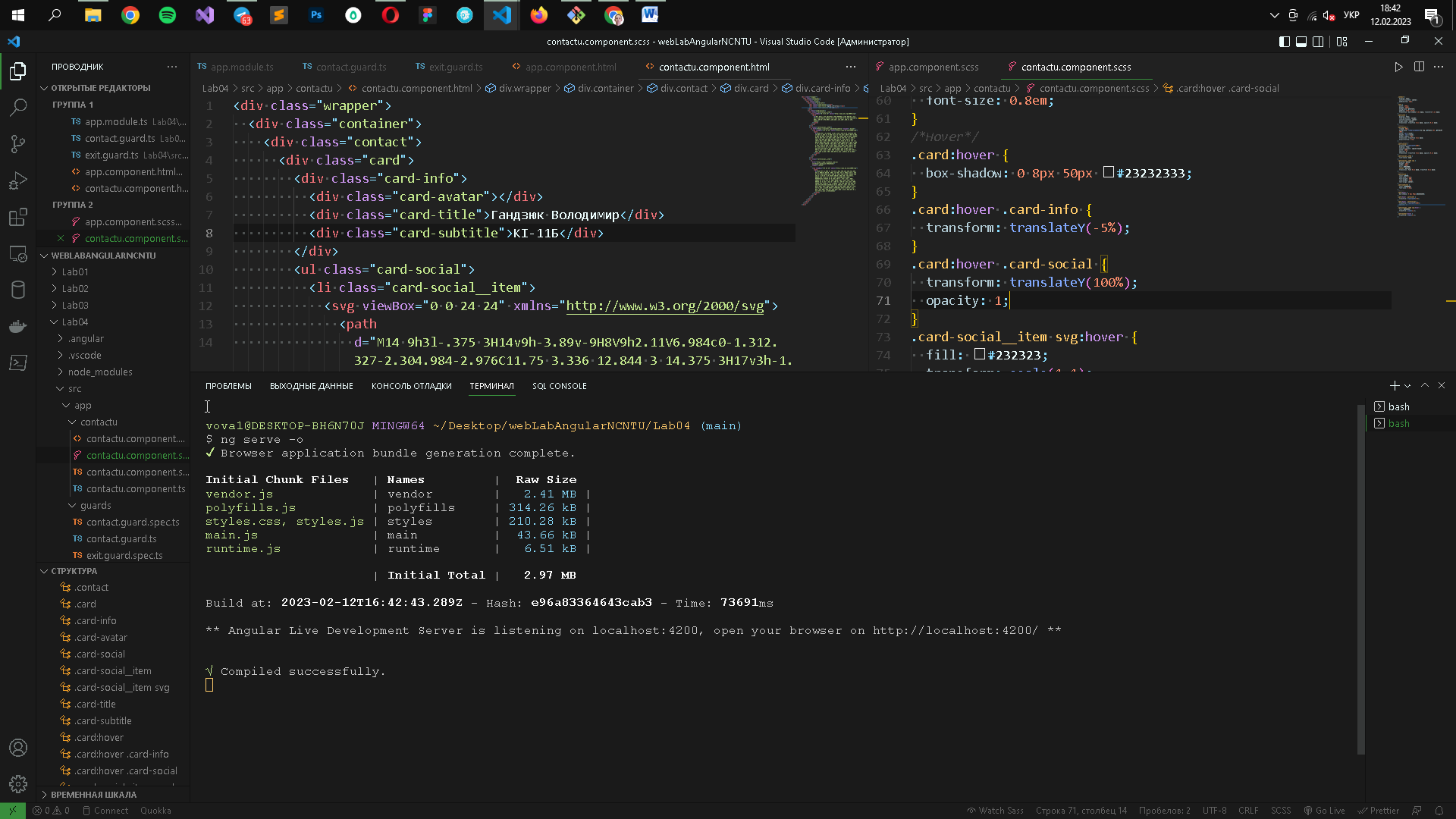


Рисунок 5 –­ Запуск сервера.

Під час того, як користувач захоче перейти на вкладку «Контакти», він побачить перед собою дію функції confirm, зображено на рис.6 з запитанням чи точно він хоче перейти на цю сторінку, цей функціонал ми реалізували за допомогою Охоронця, CanActive.

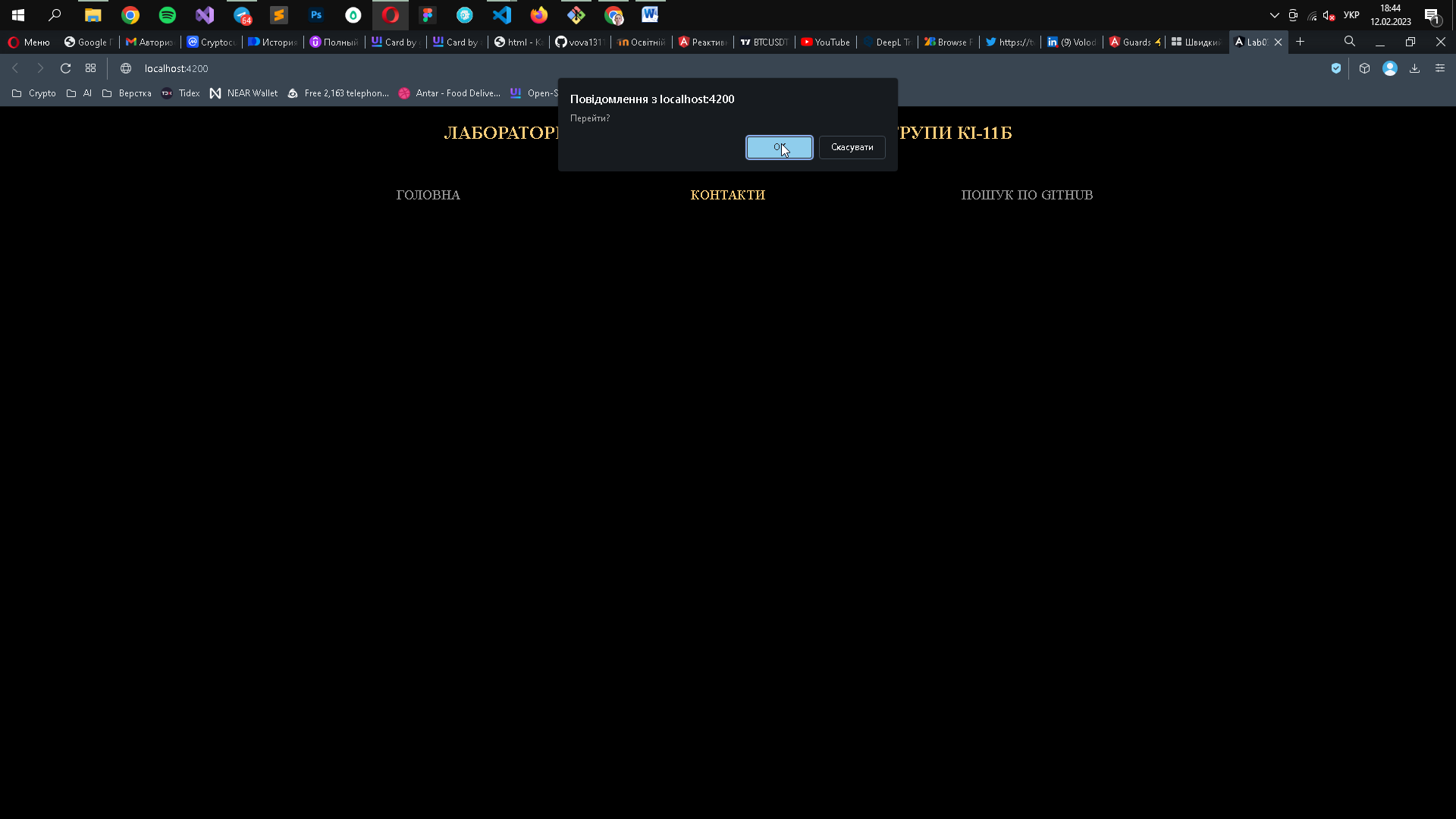


Рисунок 6 – ­Робота Guardian CanActive.

Після підтвердження дії, користувач буде пере направлений на сторіку «Контакти» , зображено на рис.7

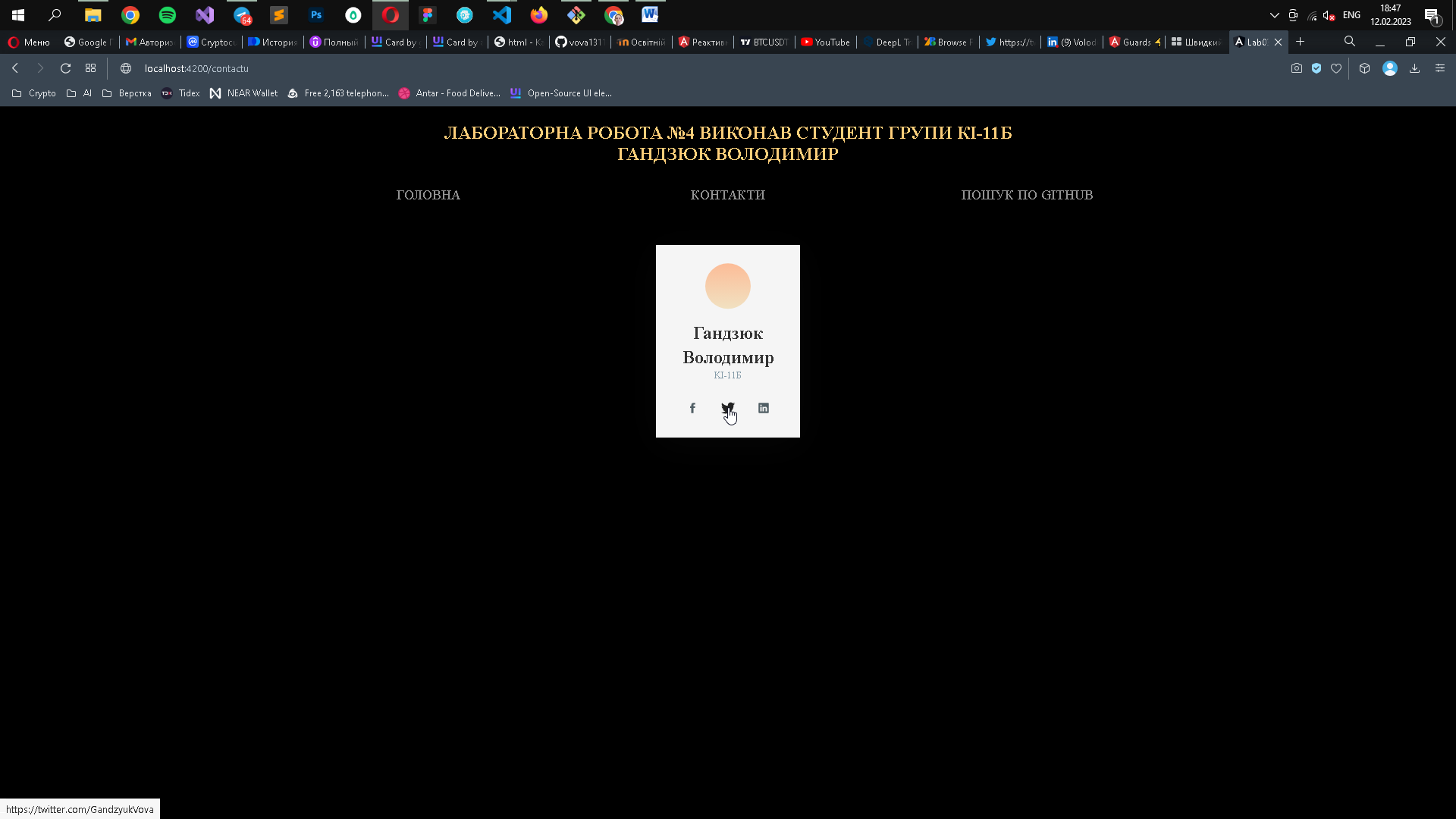


Рисунок 7 – Результат виконання дії

Під час того, як користувач захоче перейти з сторінки «Пошук по GitHub» він побачить перед собою форму зображено на рис.8, в якій буде попереджено користувача про те, що в нього є не збережені зміни і якщо він перейде на іншу сторінку то все очиститься. Цю функцію було зроблено за допомогою створеного охоронця типу CanDeactivate.

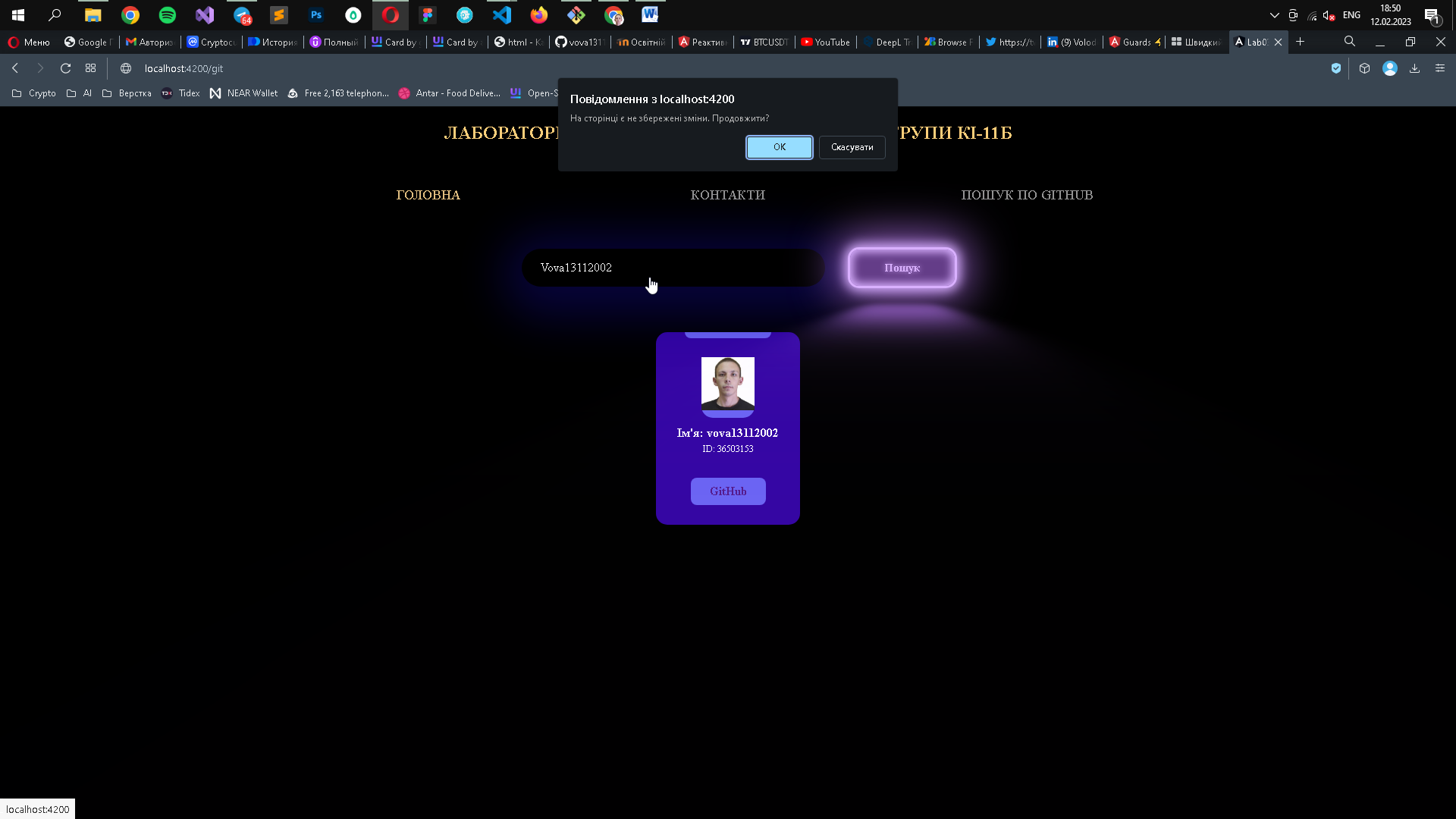


Рисунок 8 – Результат виконання дії

**Висновок:** В процесі виконання даної лабораторної роботи був досліджений процес маршрутизації на платформі Angular, а також були досліджені “охоронці” (guards) з властивостями CanActivate і CanDeactivate.