Міністерство освіти і науки України

ВСП «Надвірнянський фаховий коледж НТУ»

Звіт

Для лабораторної роботи №3

З дисципліни «Розробка та програмування веб-застосувань»

На тему: «Сервіси, мережева комунікація на платформі Angular»

Варіант 3

Виконав: студент групи КІ-11Б

Гандзюк В.В.

Перевірив: Волинський О. І.

Надвірна 2023р.

**Тема:** Сервіси, мережева комунікація на платформі Angular

**Мета:** Ознайомити з сервісами та мережавими комунікаціями на платформі Angular.

**Завдання:** Виконати ознайомлення з сервісами та мережавими комунікаціями на платформі Angular.

**Варіант:** 3.

**Хід роботи:**

Для виконання даної роботи, було потрібно створити новий проект за допомогою команди *ng new Lab03* і перенести файли з лабораторної роботи №2. Також, створити новий компонент за допомогою команди *ng g c main-api*. Створення файлу зображено на рисунку 1.

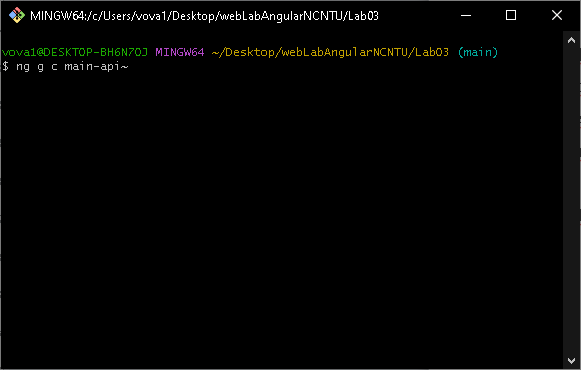


Рисунок 1 ­– Створення нового компонент.

Виконуючи доповнення застосунку розробленого протягом двох попередніх лабораторних робіт потрібно створити сервіси, які будуть включати в себе мережеву комунікацію по протоколу HTTP, для цього використовуємо знайдений, безплатний «API GitHub». З допомогою даного сервісу реалізовуємо пошук користувачів за ім’ям.

Посилання на API : [*https://api.github.com/search/users/*](https://api.github.com/search/users/)

Також, потрібно створити сервіс, який буде включати в себе мережеву комунікацію. Для його створення використовуємо команду *ng g service newService*. Створення зображено на рисунку 2.



Рисунок 2 ­– створення сервісу.

Лістинг коду з таких файлів: «app.module.ts», «new-servise.service», «main-api.component.ts», «main-api.component.html», «main-api.component.scss», «app.component.html» має наступний вигляд:

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 { NewComponentComponent } from './new-component/new-component.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';

@NgModule({

declarations: [

AppComponent,

NewComponentComponent,

MainApiComponent

],

imports: [

BrowserModule,

AppRoutingModule,

HttpClientModule,

FormsModule

],

providers: [NewServiceService],

bootstrap: [AppComponent]

})

export class AppModule { }

Лістинг коду new-service.service.ts:

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

@Injectable({

providedIn: 'root'

})

export class NewServiceService {

}

Лістинг коду main-api.component.ts:

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

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

@Component({

selector: 'app-main-api',

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

styleUrls: ['./main-api.component.scss']

})

export class MainApiComponent {

userName: string = "";

response: any;

constructor(private http: HttpClient) {

}

search() {

this.http.get('https://api.github.com/users/' + this.userName)

.subscribe((response) => {

this.response = response;

console.log(this.response);

})

}

}

Лістинг коду main.api-component.html:

<h3 class="search\_\_git-title">Пошук користувача GitHub</h3>

<div class="Username">

<input

class="input"

placeholder="Username"

type="text"

[(ngModel)]="userName"/>

<button class="Username\_\_button" (click)="search()">Пошук</button>

</div>

<div class="card">

<div class="card\_\_item">

<div class="card\_\_border-top"></div>

<div \*ngIf="response" class="card\_\_img">

<img src="{{ response.avatar\_url }}" alt="avatar" />

</div>

<span class="card\_\_person"> Ім'я: {{ response.login }}</span>

<p class="card\_\_job">ID: {{ response.id }}</p>

<button class="card\_\_button">

<a

class="card\_\_button-link"

href="{{ response.html\_url }}"

target="\_blank"

>GitHub</a>

</button>

</div>

</div>

Лістинг коду main.api-component.scss:

.Username{

text-align: center;

&\_\_button{

--glow-color: rgb(217, 176, 255);

--glow-spread-color: rgba(191, 123, 255, 0.781);

--enhanced-glow-color: rgb(231, 206, 255);

--btn-color: rgb(100, 61, 136);

border: .25em solid var(--glow-color);

padding: 1em 3em;

color: var(--glow-color);

font-size: 15px;

font-weight: bold;

background-color: var(--btn-color);

border-radius: 1em;

outline: none;

box-shadow: 0 0 1em .25em var(--glow-color),

0 0 4em 1em var(--glow-spread-color),

inset 0 0 .75em .25em var(--glow-color);

text-shadow: 0 0 .5em var(--glow-color);

position: relative;

transition: all 0.3s;

&:after {

pointer-events: none;

content: "";

position: absolute;

top: 120%;

left: 0;

height: 100%;

width: 100%;

background-color: var(--glow-spread-color);

filter: blur(1em);

opacity: .7;

transform: perspective(1.5em) rotateX(35deg) scale(1, .6);

}

&:hover {

color: var(--btn-color);

background-color: var(--glow-color);

box-shadow: 0 0 1em .25em var(--glow-color),

0 0 4em 2em var(--glow-spread-color),

inset 0 0 .75em .25em var(--glow-color);

}

&:active {

box-shadow: 0 0 0.6em .15em var(--glow-color),

0 0 1.5em 1em var(--glow-spread-color),

inset 0 0 .5em .15em var(--glow-color);

}

}

}

.input {

margin: 30px;

background: none;

border: none;

outline: none;

// max-width: 300px;

padding: 15px 200px 15px 25px;

font-size: 16px;

border-radius: 9999px;

box-shadow: 5px 10px 60px 30px #100a886b;

color: #fff;

}

.search\_\_git-title{

margin-top: 20px;

text-align: center;

font-family: 'Lab Grotesque';

font-weight: 700;

font-size: 24px;

line-height: 28px;

letter-spacing: 0.05em;

text-transform: uppercase;

color: #FFCF7A;

}

.card{

&\_\_item {

width: 190px;

height: 254px;

background: #3405a3;

border-radius: 15px;

margin: 0 auto;

margin-top: 30px;

}

&\_\_border-top {

width: 60%;

height: 3%;

background: #6b64f3;

margin: auto;

border-radius: 0px 0px 15px 15px;

}

&\_\_person {

font-weight: 600;

color: white;

text-align: center;

display: block;

padding-top: 10px;

font-size: 16px;

}

&\_\_job {

font-weight: 400;

color: white;

display: block;

text-align: center;

padding-top: 3px;

font-size: 12px;

}

&\_\_img {

width: 70px;

height: 80px;

background: #6b64f3;

border-radius: 15px;

margin: auto;

margin-top: 25px;

}

&\_\_button {

padding: 8px 25px;

display: block;

margin: auto;

border-radius: 8px;

border: none;

margin-top: 30px;

background: #6b64f3;

color: white;

font-weight: 600;

}

&\_\_button:hover {

background: #534bf3;

}

}

За допомогою команди *ng serve*, запускаємо сервер. Використання команди зображено на рисунку 3.

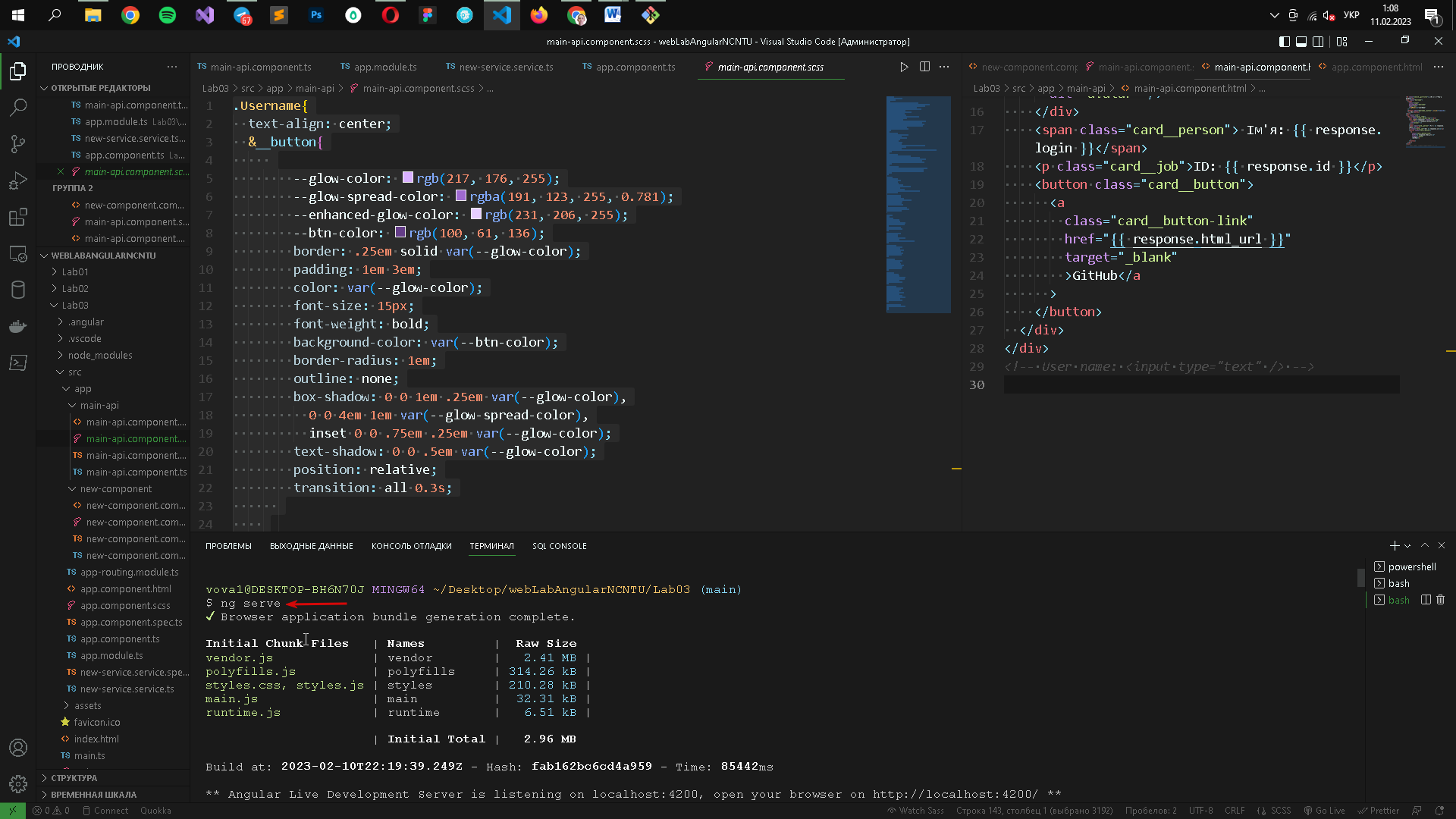


Рисунок 3 ­– запуск сервера.

Ввівши в строку пошуку <http://localhost:4200/>, виконається перехід на сайт. Вікно з сайтом зображено на рисунку 4.

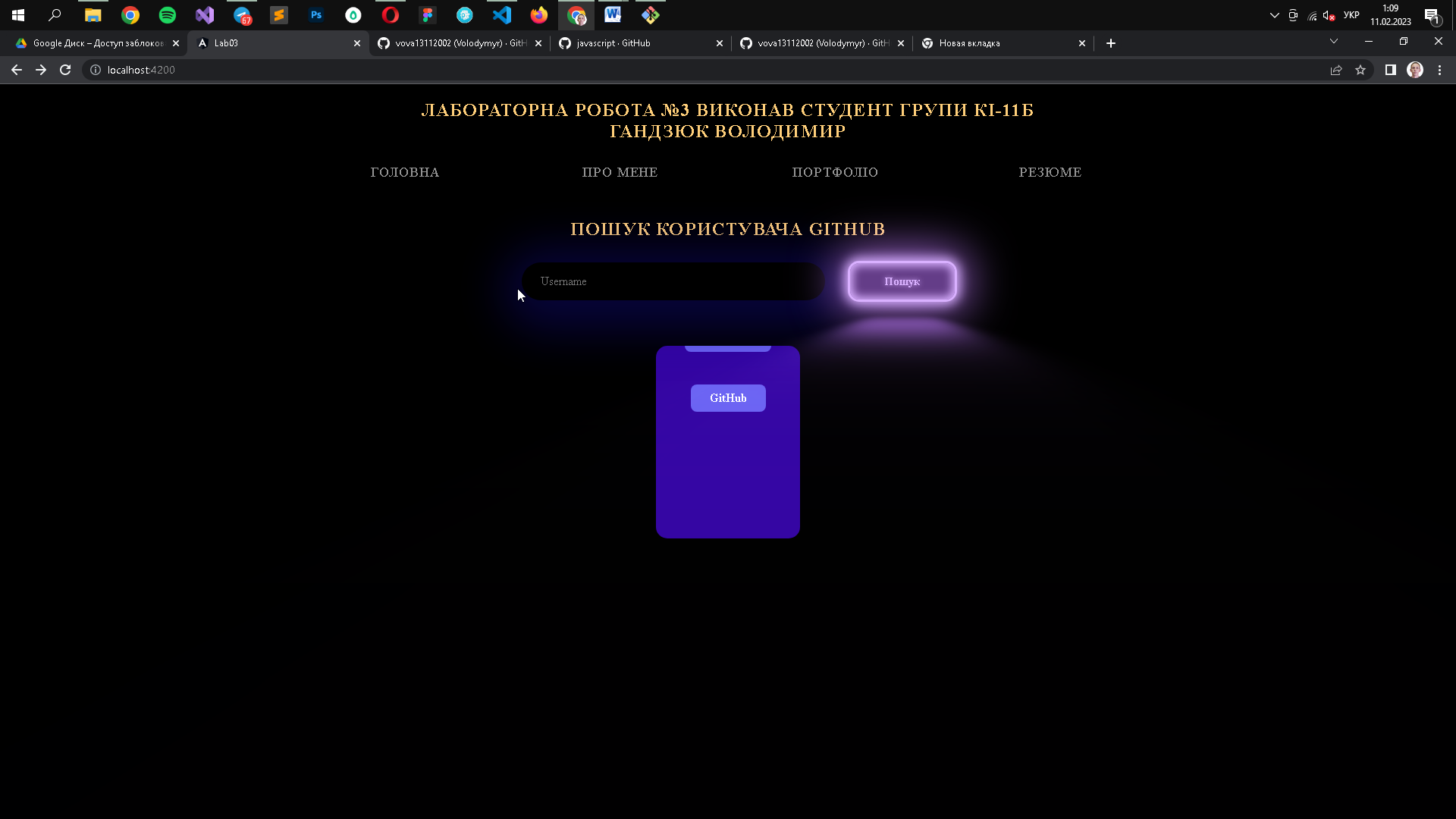


Рисунок 4 ­– Результат запуску сервера.

Щоб знайти користувача на «GitHub», потрібно ввести його ім’я в «input» та натиснути кнопку «Пошук». Вікна з виконанням пошуку зображено на рисунках 5 і 6.

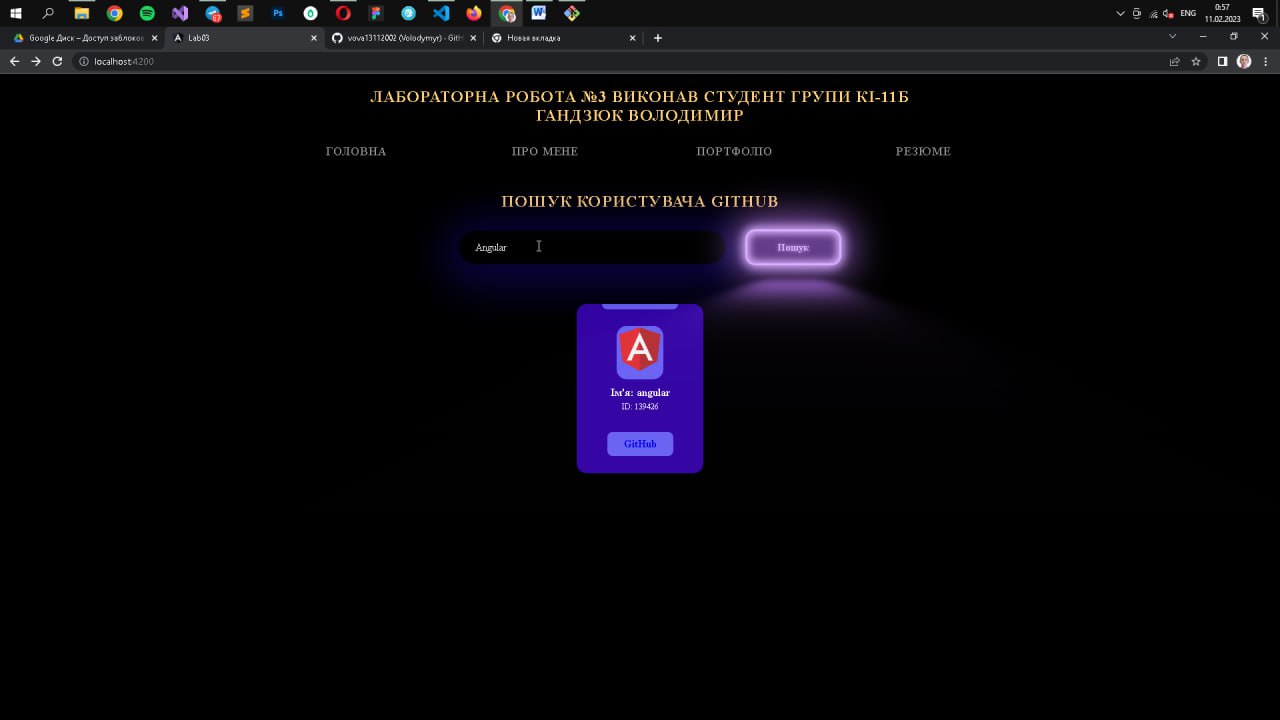


Рисунок 5 ­– Вікно з виконанням пошуку.

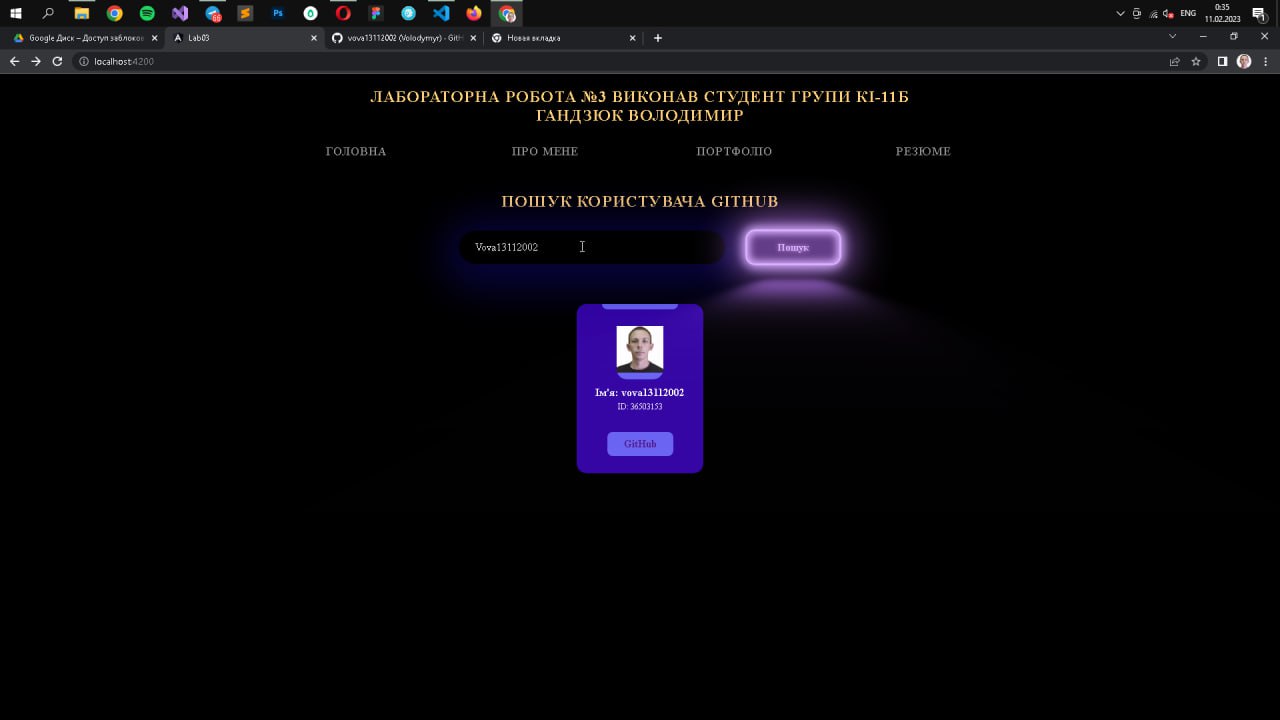


Рисунок 6 ­– Результат пошуку.

Під час пошуку користувач отримає: аватарку користувача, ім’я, ID, і також може використати кнопку за допомогою якої зможе перейти на «GitHub» і побачити свої репозиторії і активність. Вікно з «GitHub» користувача зображено на рисунку 7.

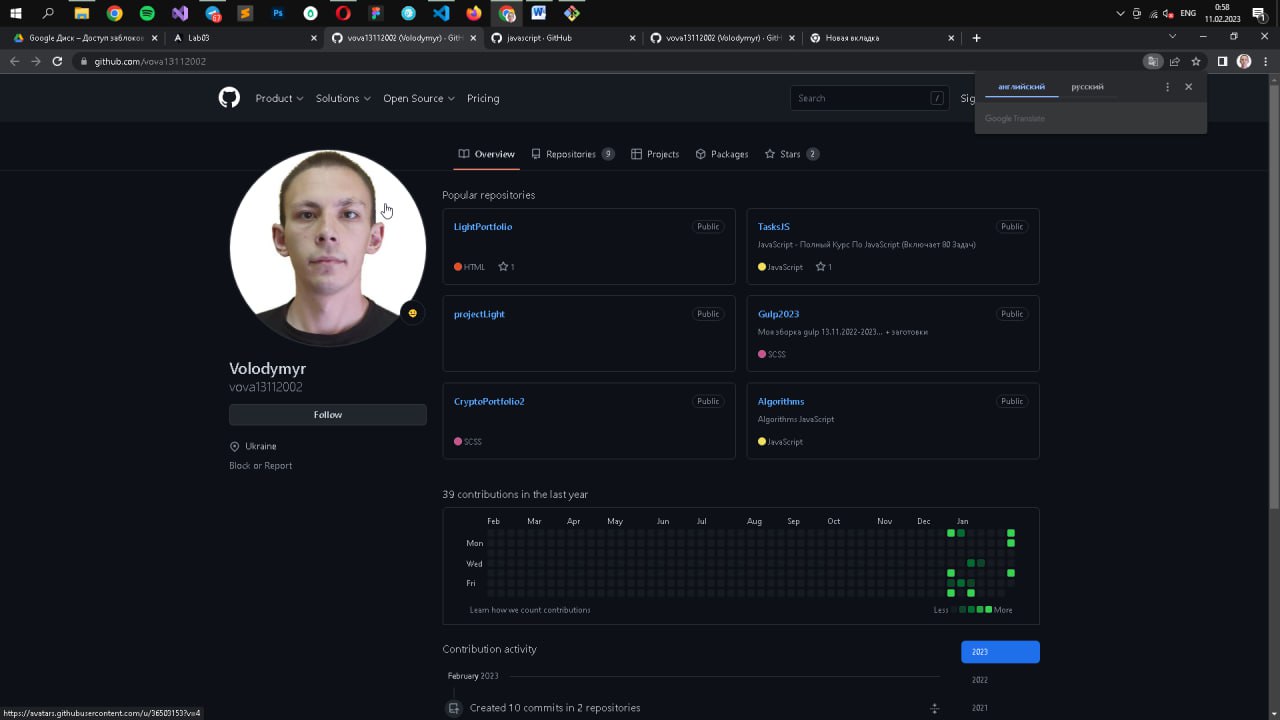


Рисунок 7 ­– «GitHub» користувача.

**Висновок:** В процесі виконання даної лабораторної роботи було проведене ознайомлення з сервісами та мережевими комунікаціями на платформі «Angular», досліджено процес взаємодії з «API», а також розроблено веб-застосунок, який отримує дані через «API» та за допомогою якого можна знайти користувача на «GitHub» і відобразити його дані у зручному для користувача вигляді.