Міністерство освіти і науки України

ВСП «Надвірнянський фаховий коледж НТУ»

Звіт

Для лабораторної роботи №5

З дисципліни «Розробка та програмування веб-застосувань»

На тему: «HTML-форми на платформі Angular»

Варіант №3

Виконав: студент групи КІ-11Б

Гандзюк В.В.

Перевірив: Волинський О.І.

Надвірна 2023р.

**Тема:** HTML-форми на платформі Angular.

**Мета:** Ознайомлення з HTML-формами на платформі Angular.

**Варіант:** 3.

**Завдання:**

1. Доповнити існуючий застосунок, який розроблявся протягом чотирьох попередніх лабораторних робіт компонентами і сервісами, додати компоненти з використанням форм і валідаторів (синхронних і асинхронних).
2. Спроектувати форму (форми) таким чином, щоб були залежності між елементами управління (наприклад динамічне наповнення контейнера <select> відповідно до значення деякого елемента форми).
3. Створити власні класи валідаторів з особливими правилами для перевірки (порадитись і погодити з викладачем).

**Хід роботи:**

Перед початком роботи з створення форми для зручності, потрібно створити новий компонент ,представлено на рисунку 1 і добавити її в меню і налаштувати роутінг і маршрутизацію.

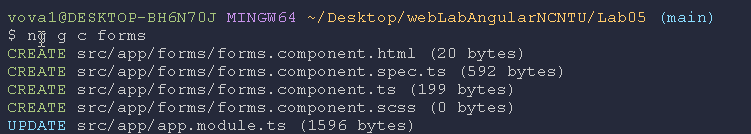


Рисунок 1 ­­­– Створення нового компонента forms.

Далі зробимо розмітку сторінки, і добавимо стилів і функціоналу нижче представлено в лістингу коду:

1. Forms.component.html;
2. Forms.component.scss;
3. Forms.component.ts;
4. App.module.ts;

1.Лістинг коду Forms.component.html:

<form class="form" [formGroup]="form" (ngSubmint)="OnSubmit()">

<p id="heading">Авторизація</p>

<div class="field">

<svg class="input-icon" xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"

viewBox="0 0 16 16">

<path d="M13.106 7.222c0-2.967-2.249-5.032-5.482-5.032-3.35 0-5.646 2.318-5.646 </path>

</svg>

<input autocomplete="off" placeholder="Email:" class="input-field" type="text" formControlName="email">

</div>

<span class="helper-text" \*ngIf="form.get('email').invalid && form.get('email').touched">

<span \*ngIf="form.get('email').errors['required']">Email не повинен бути пустим!</span>

<span \*ngIf="form.get('email').errors['email']">Введіть коректний email адрес!</span>

</span>

<div class="field">

<svg class="input-icon" xmlns="http://www.w3.org/2000/svg" width="16" height="16" fill="currentColor"

viewBox="0 0 16 16">

<path

d="M8 1a2 2 0 0 1 2 2v4H6V3a2 2 0 0 1 2-2zm3 6V3a3 3 0 0 0-6 0v4a2 2 0 0 0-2 2v5a2 2 0 0 0 2 2h6a2 2 0 0 0 2-2V9a2 2 0 0 0-2-2z">

</path>

</svg>

<input placeholder="Password:" class="input-field" type="password" formControlName="password">

</div>

<span class="helper-text" \*ngIf="form.get('password').invalid && form.get('password').touched">

<span \*ngIf="form.get('password').errors['required']">Пароль не повинен бути пустим</span>

<span \*ngIf="form.get('password').errors['minlength']&&form.get('password').errors['minlength']['requiredLength']">Пароль

повинен бути більше {{form.get('password').errors['minlength']['requiredLength']}} символів. Зараз

{{form.get('password').errors['minlength']['actualLength']}}</span>

</span>

<div class="btn">

<buttonclass="button1"type="submit" [disabled]="form.invalid">&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Вхід&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</button>

<button class="button2">Реєстрація</button>

</div>

<button class="button3">Забули пароль</button>

</form>

2.Лістинг коду Forms.component.scss:

.form {

display: flex;

flex-direction: column;

gap: 10px;

padding-left: 2em;

padding-right: 2em;

padding-bottom: 0.4em;

background-color: #171717;

border-radius: 25px;

transition: .4s ease-in-out;

width: 500px;

height: 450px;

display: flex;

justify-content: center;

padding-top: 50px;

margin: 0 auto;

margin-top: 50px;

}

.form:hover {

transform: scale(1.05);

border: 1px solid black;

}

#heading {

text-align: center;

margin: 2em;

color: rgb(255, 255, 255);

font-size: 1.2em;

}

.field {

display: flex;

align-items: center;

justify-content: center;

gap: 0.5em;

border-radius: 25px;

padding: 0.6em;

border: none;

outline: none;

color: white;

background-color: #171717;

box-shadow: inset 2px 5px 10px rgb(5, 5, 5);

}

.input-icon {

height: 1.3em;

width: 1.3em;

fill: white;

}

.input-field {

background: none;

border: none;

outline: none;

width: 100%;

color: #d3d3d3;

}

.form .btn {

display: flex;

justify-content: center;

flex-direction: row;

margin-top: 2.5em;

}

.button1 {

padding: 0.5em;

padding-left: 1.1em;

padding-right: 1.1em;

border-radius: 5px;

margin-right: 0.5em;

border: none;

outline: none;

transition: .4s ease-in-out;

background-color: #252525;

color: white;

}

.button1:hover {

background-color: black;

color: white;

}

.button2 {

padding: 0.5em;

padding-left: 2.3em;

padding-right: 2.3em;

border-radius: 5px;

border: none;

outline: none;

transition: .4s ease-in-out;

background-color: #252525;

color: white;

}

.button2:hover {

background-color: black;

color: white;

}

.button3 {

margin-bottom: 3em;

padding: 0.5em;

border-radius: 5px;

border: none;

outline: none;

transition: .4s ease-in-out;

background-color: #252525;

color: white;

}

.button3:hover {

background-color: red;

color: white;

}

.helper-text {

color: red;

padding-left: 15px;

}

3.Лістинг коду Forms.component.ts:

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

import { FormControl, FormGroup, Validators } from '@angular/forms';

@Component({

selector: 'app-forms',

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

styleUrls: ['./forms.component.scss']

})

export class FormsComponent implements OnInit {

form: FormGroup

ngOnInit() {

this.form = new FormGroup({

email: new FormControl(null, [Validators.required, Validators.email]),

password: new FormControl(null, [Validators.required, Validators.minLength(8)])

})

}

OnSubmit() {

}

}

4.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, ReactiveFormsModule } 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';

import { FormsComponent } from './forms/forms.component';

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

{

path: 'forms',

component: FormsComponent

}

]

@NgModule({

declarations: [

AppComponent,

MainApiComponent,

ContactuComponent,

FormsComponent,

],

imports: [

BrowserModule,

AppRoutingModule,

HttpClientModule,

FormsModule,

ReactiveFormsModule,

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

],

providers: [NewServiceService],

bootstrap: [AppComponent]

})

export class AppModule { }

За допомогою команди ng serve –o збираємо і запускаємо сервер, щоб провірити роботу за стосунку. Представлено на рисунку 2.

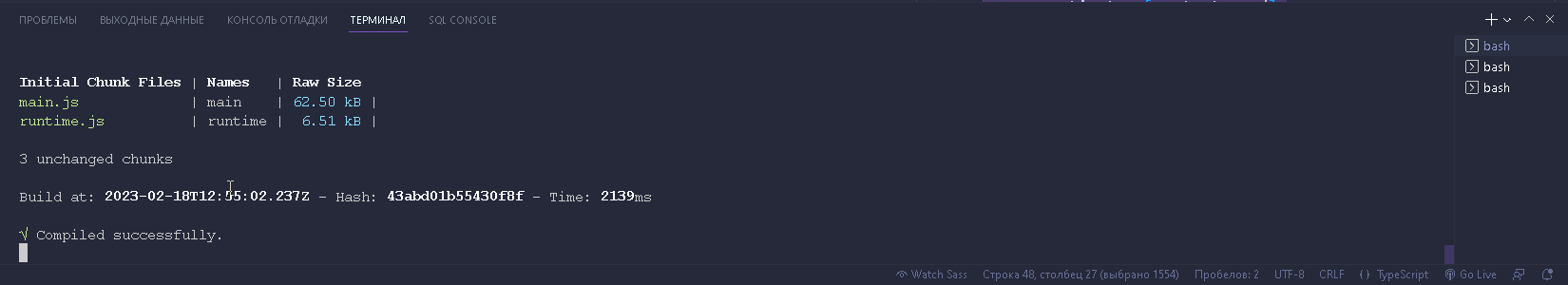


Рисунок 2 ­­­– Запуск сервера.

Стартовий вигляд застосунку представлено на рисунку 3.

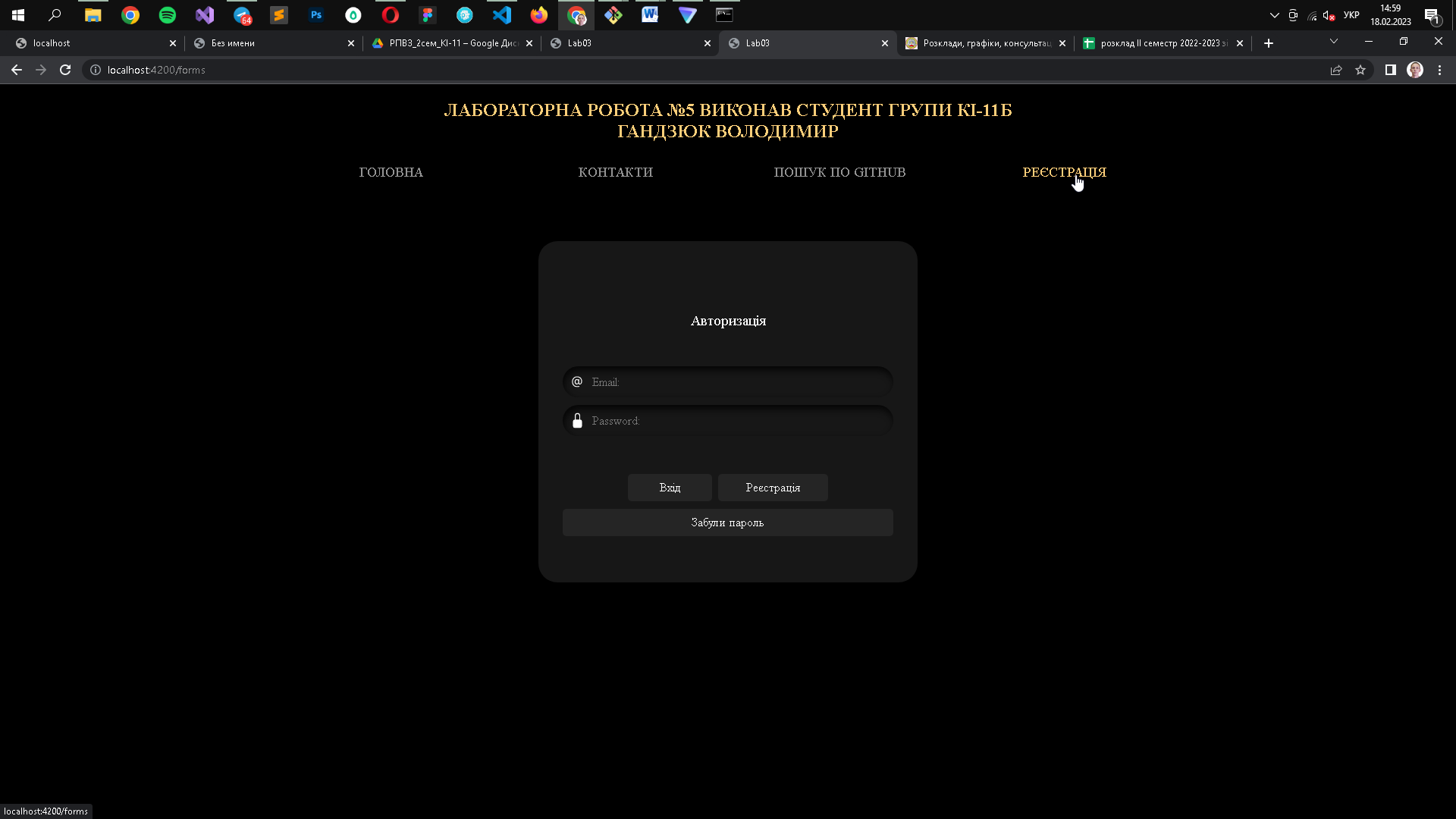


Рисунок 3 ­­­– Вигляд стартової сторінки застосунку.

Вводимо будь-яке значення в поля email і password. Як бачимо появляється повідомлення про те що введено не коректні дані зображено на рисунку 4-5, оскільки асинхронний валідатор працює. Крім того, текст під полем пов’язаний зі значенням поля:

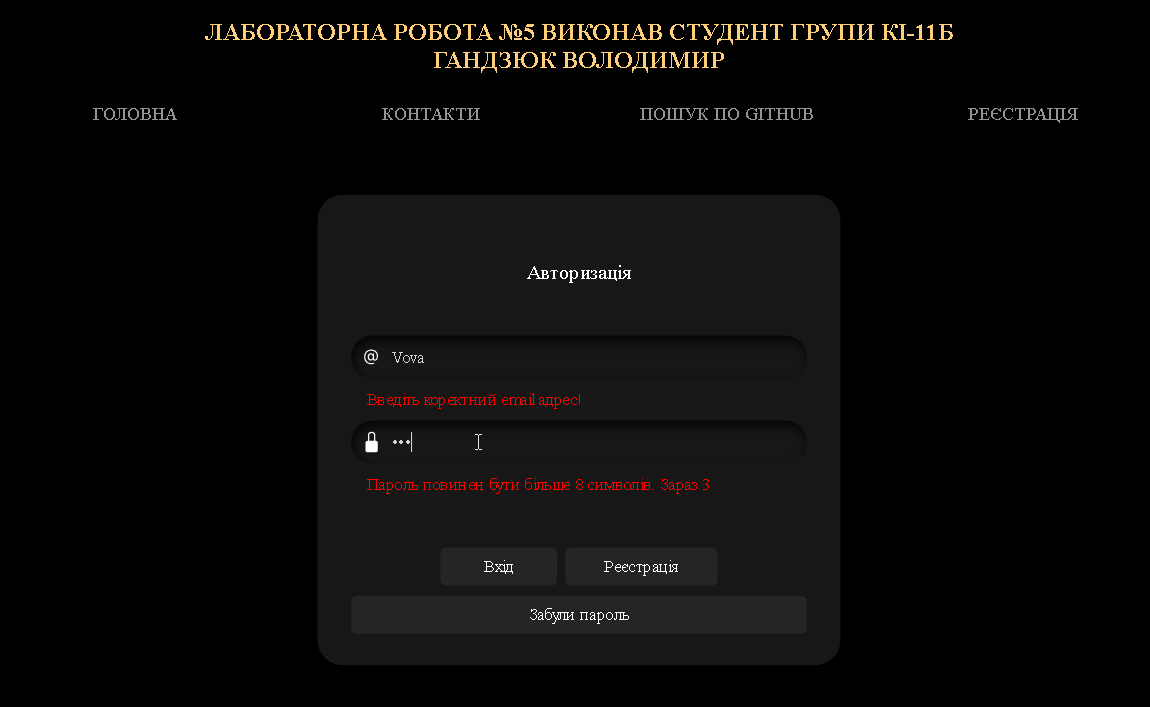


Рисунок 4 ­­­– Введені не коректні дані в input(email&password).

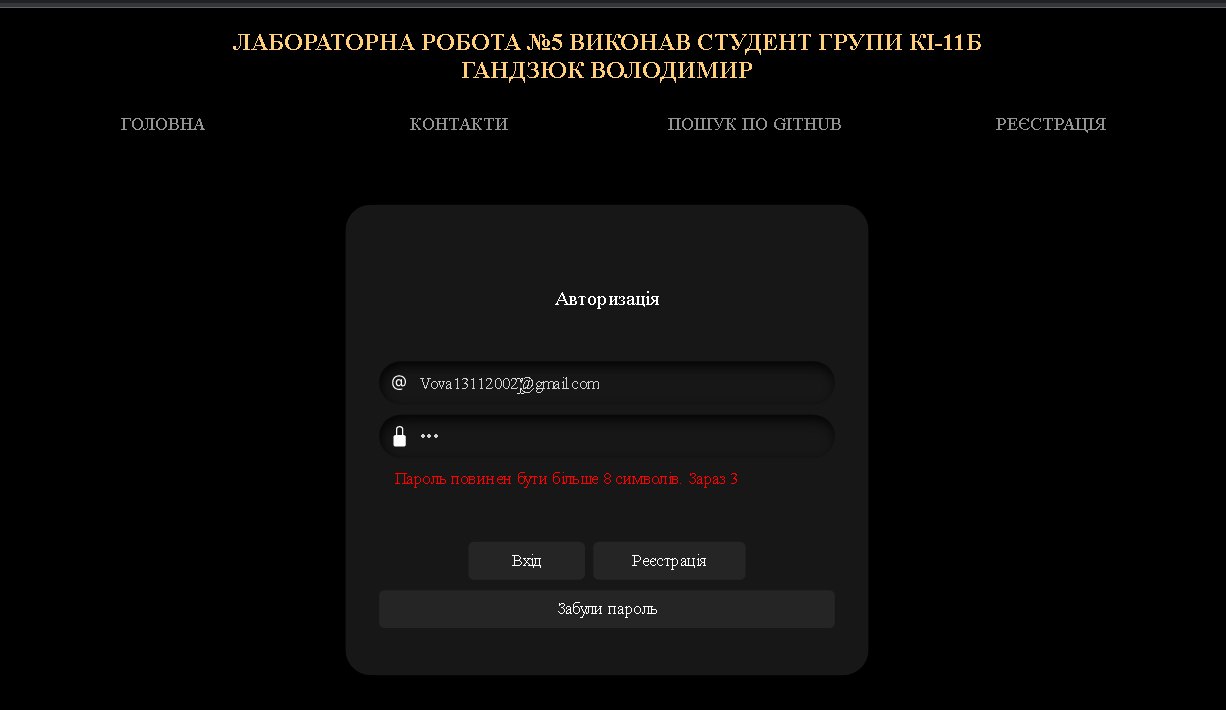


Рисунок 5 ­­­– Введені не коректні дані в input(password).

Якщо користувач введе коректні дані він, не побачити жодної помилки перед собою, представлено на рисунку 6.

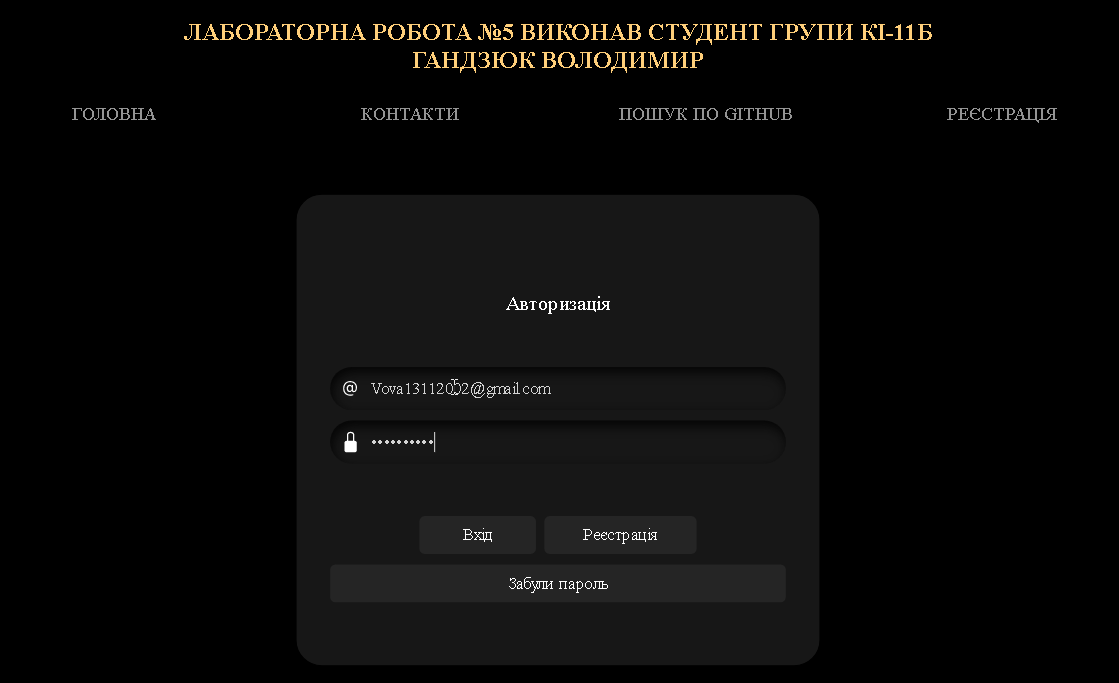


Рисунок 6 ­­­– Введені коректні дані в всі поля вводу

**Висновок:** У процесі виконання лабораторної роботи ми навчилися створювати реактивні форми, був досліджений процеси створення форми, синхронних та асинхронних валідаторів до форми, а також прив’язку значень до значення іншого елементу на платформі Angular.