НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ

СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт по лабораторній роботі No 4

Створення проекту «Blog».

З дисципліни: «Реактивне програмування»

Студент: Черпак Андрій Вадимович

Група: ІП-01

Дата захисту роботи:

Викладач: доц. Полупан Юлія Вікторівна

Захищено з оцінкою:

Київ, 2023

# Хід роботі

1. Створено проект “Blog” та два компоненти post та post-form.
2. Згідно з завданням лабораторної було виконано операції у проекті “Blog”
3. Розгорнуто додаток на платформі FireBase

# Детальний огляд компоненту Post:

**Створимо файл app.component.ts:**

import { Component } from '@angular/core';  
export interface Post {  
title:string;  
text:string;  
  
  
id?:number;  
}  
@Component({  
selector: 'app-root',  
templateUrl: './app.component.html',  
styleUrls: ['./app.component.css']  
})  
export class AppComponent {  
title = 'BlogComponents';  
posts: Post[]=[{title:'Вивчаю компоненти', text:'Створюю проект "Блог"', id:1},  
{title:'Вивчаю директиви', text:'Все ще створюю "Блог"', id:2}];  
updatePosts(post:Post){  
 this.posts.unshift(post);  
 console.log('Post',post);  
 }  
deletePost(id:number){  
 this.posts=this.posts.filter(p=>p.id!==id)  
 }  
}

Створимо шаблон для даного компоненту:

<div class="container">  
 <app-post-form  
 (onAdd)="updatePosts($event)"  
 ></app-post-form>  
 <hr/>  
 <app-post  
\*ngFor="let p of posts"  
[myPost]="p"  
(onRemove)="deletePost($event)"  
>  
<small \*ngIf="p.text.length>20; else short">Пост довгий</small>  
<ng-template #short>  
<small>Пост короткий</small>  
</ng-template>  
</app-post>  
 </div>

Створимо компонент post.component.ts:

import { Component, Input, OnInit, OnDestroy, Output, EventEmitter } from '@angular/core';  
import { Post } from '../app.component';  
  
@Component({  
 selector: 'app-post',  
 templateUrl: './post.component.html',  
 styleUrls: ['./post.component.css']  
})  
export class PostComponent implements OnInit, OnDestroy {  
 @Input() myPost!: Post;  
 @Output() onRemove=new EventEmitter<number>()  
 constructor() { }  
 removePost(){  
 this.onRemove.emit(this.myPost.id)  
 }  
 ngOnInit(): void {  
 }  
 ngOnDestroy(){  
 console.log('метод ngOnDestroy');  
 }  
}

Він матиме відповідний шаблон:

<div class="card">  
 <h2>{{myPost.title}}</h2>  
 <p>{{myPost.text}}</p>  
 <button class="btn" (click)="removePost()">&times;</button>  
 <ng-content></ng-content>  
</div>

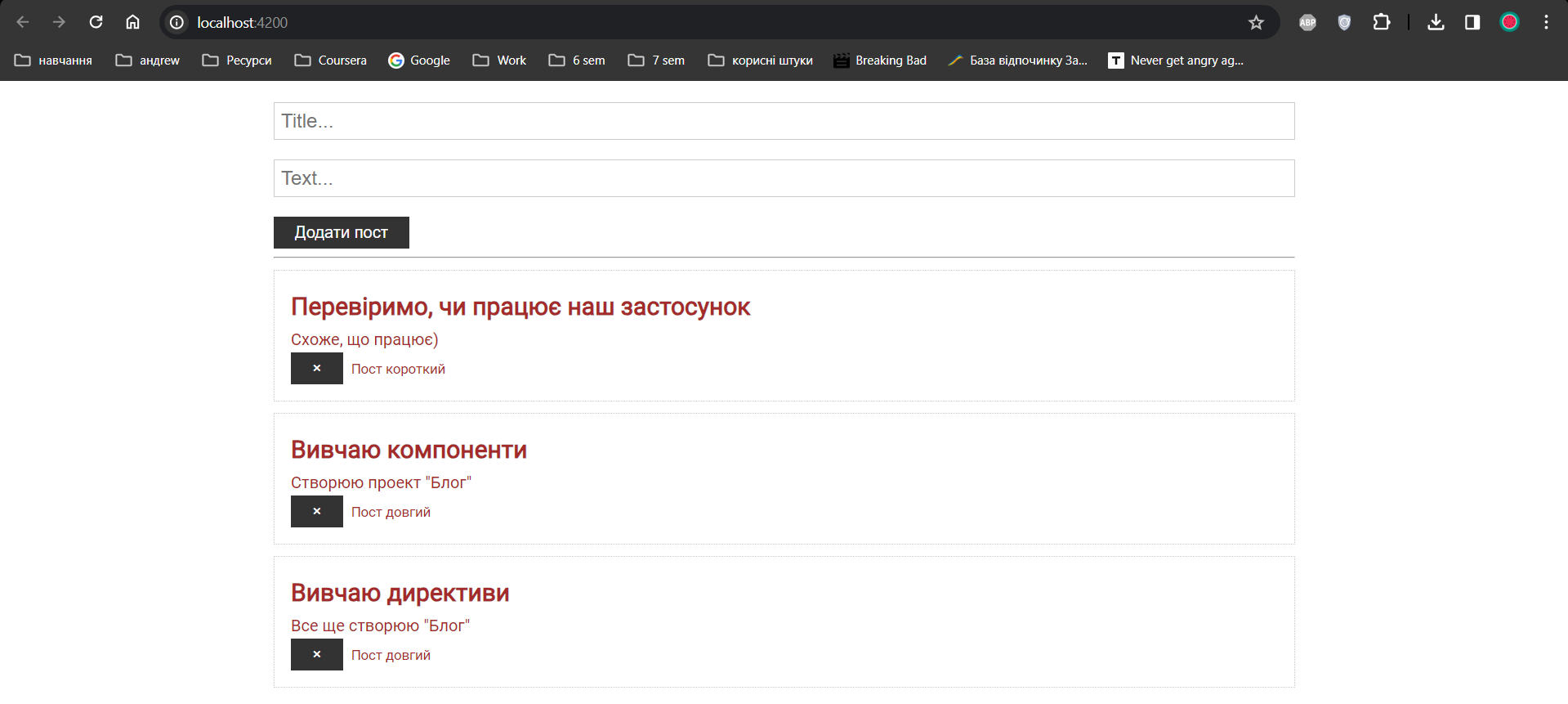
Також створимо компоненту для створення постів – post-form.component.ts:

import { Component, EventEmitter, OnInit, Output} from '@angular/core';  
import { Post } from '../app.component';  
@Component({  
 selector: 'app-post-form',  
 templateUrl: './post-form.component.html',  
 styleUrls: ['./post-form.component.css']  
})  
export class PostFormComponent implements OnInit {  
 @Output() onAdd:EventEmitter<Post> = new EventEmitter<Post>()  
 title='';  
 text='';  
 constructor() { }  
 ngOnInit(): void {  
 }  
 addPost(){  
 if (this.title.trim()&&this.text.trim()){  
 const post: Post={  
 title:this.title,  
 text:this.text  
 }  
 this.onAdd.emit(post);  
 console.log('New post',post);  
 this.title=this.text=''; *// очищення полів* }  
 }  
}

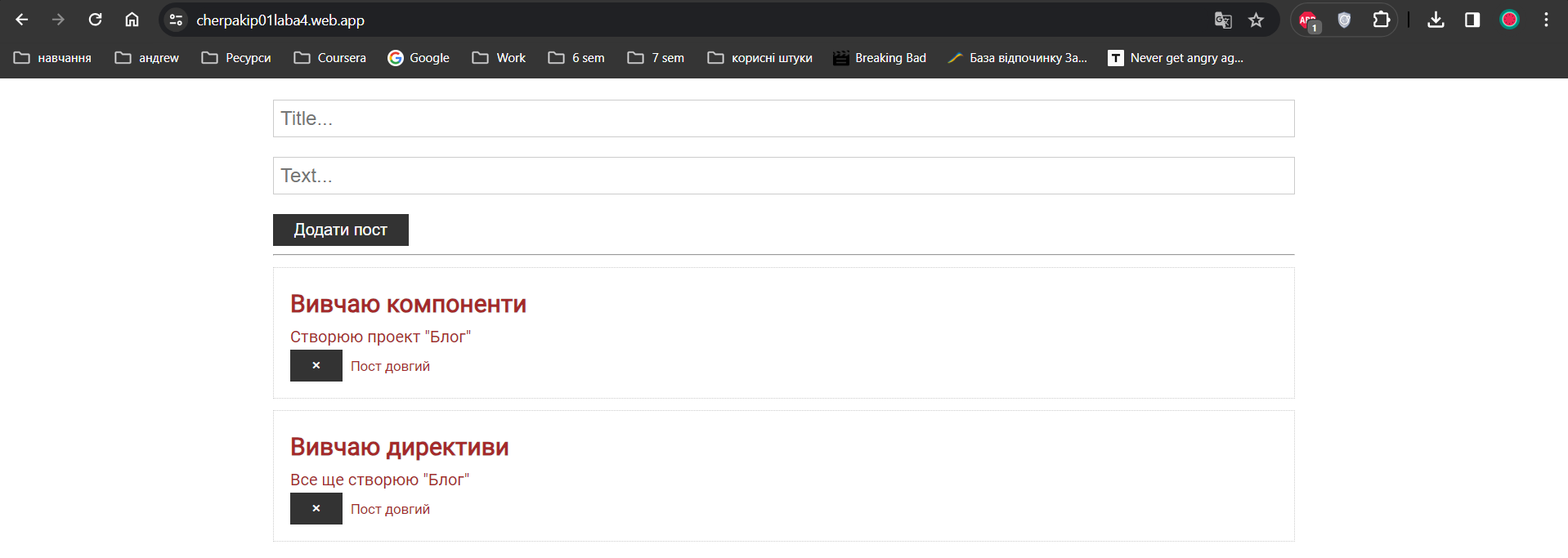
Вона використовуватиме такий шаблон:

<div>  
 <input  
 type="text"  
 class="form-control"  
 placeholder="Title..."  
 [(ngModel)]="title">  
 <input  
 type="text"  
 class="form-control"  
 placeholder="Text..."  
 [(ngModel)]="text">  
 <button class="btn" (click)="addPost()">Додати пост</button>  
</div>

Розгорнемо наш застосунок на локальній мережі та перевіримо, чи він працює:



Як бачимо, все відпрацьовує коректно, тому можемо розгортати за стосунок на платформі firebase:



Посилання на результат: https://cherpakip01laba4.web.app/

# Висновок

У цьому завданні ми розглянули додаток під назвою "Blog". Ми досліджували використання Angular, створили просту модель даних для нашого додатка та вивчили, як працює двостороння прив'язка даних на практиці. Також ми освоїли функціонал нестрогих змінних, передачу даних між компонентами за допомогою ngModel, а також використання методів ngOnInit() та ngOnDestroy(). У процесі роботи над проектом ми використовували EventEmitter для ефективної комунікації між компонентами та сповіщення батьківського елементу про додавання нових даних.

# Список літератури

* + 1. Introduction to the Angular docs [Електронний ресурс] – Режим доступу до ресурсу: <https://angular.io/docs>.
    2. Angular. Angular tutorials [Електронний ресурс] / Angular – Режим доступу до ресурсу: <https://angular.io/tutorial>.
    3. The Angular Book [Електронний ресурс] – Режим доступу до ресурсу: <https://angular-book.dev/>.