

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**«ЗАПОРІЗЬКИЙ ІНСТИТУТ ЕКОНОМІКИ ТА ІНФОРМАЦІЙНИХ**  
**ТЕХНОЛОГІЙ»**

**Кафедра економічної кібернетики та інженерії програмного  
забезпечення**

**ЗВІТ**

**З виконання ПРАКТИЧНОЇ роботи №2**

**“РОБОТА З КОМПОНЕНТАМИ ”**

**з предмету**

**Розробка програм за допомогою скриптів**

**Виконав**

**студ. гр.    ІІЗ-129**

\_\_\_\_\_

**Деркач В. А.**

**Перевішив**

\_\_\_\_\_

**Мержинський Є.К.**

**Запоріжжя**

**2022**

Завдання. Оживити форму додавання постів. Навчитися передавати дані з дочірнього компонента до батьківського. При додаванні посту відображати одразу його на сайті та очищати форму додавання. Організувати доступ до HTML елементів.

```
4 import { AppRoutingModuleModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { PostComponent } from './post/post.component';
7 import { PostFormComponent } from './post-form/post-form.component';
8 import { FormsModule } from '@angular/forms';
9
10 @NgModule({
11   declarations: [
12     AppComponent,
13     PostComponent,
14     PostFormComponent
15   ],
16   imports: [
17     BrowserModule,
18     AppRoutingModuleModule,
19     FormsModule
20   ],
21   providers: [],
22   bootstrap: [AppComponent]
23 })
24 export class AppModule { }
25
```

Рис. 1 – Імпортував модуль FormsModule

Створив дві перемінні title, text та створив метод addPost() (Рис. 2).

```
src > app > post-form > TS post-form.component.ts > PostFormComponent
1  import { Component, OnInit } from '@angular/core';
2  import { Post } from '../app.component';
3
4  @Component({
5    selector: 'app-post-form',
6    templateUrl: './post-form.component.html',
7    styleUrls: ['./post-form.component.css']
8  })
9  export class PostFormComponent implements OnInit {
10
11    title=""
12    text=""
13    constructor() { }
14
15    ngOnInit(): void {
16    }
17
18    addPost() {
19      if(this.text.trim() && this.title.trim()) {
20        const post : Post = {
21          title: this.title,
22          text: this.text
23        }
24        console.log('Add post: ', post)
25        this.text=''
26        this.title=''
27      }
28    }
29  }
30
```

Рис. 2 – Додав код

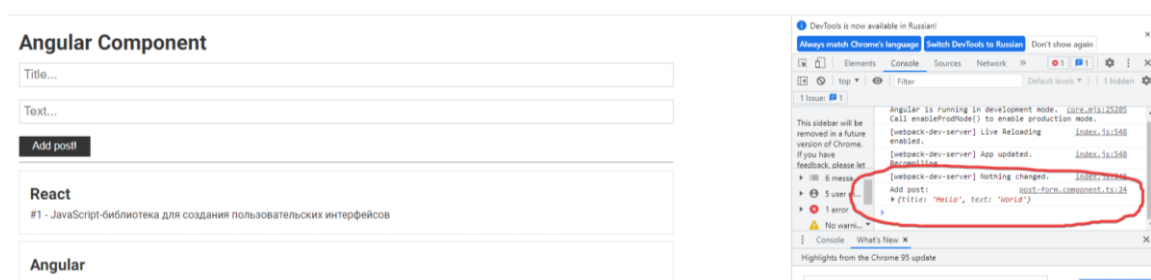


Рис. 3 – Тест доданого коду

```

import { Component, OnInit, Output, EventEmitter } 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 {

  title=""
  text=""
  @Output() addPostUser: EventEmitter<Post> = new EventEmitter<Post>()
  constructor() { }

  ngOnInit(): void {
  }

  addPost() {
    if(this.text.trim() && this.title.trim()) {
      const post : Post ={
        title: this.title,
        text: this.text
      }
      this.addPostUser.emit(post)
      this.text=''
      this.title=''
    }
  }
}

```

Рис. 4 – Додав код

## Angular Component

**Заголовок - тест**  
# - Текст - тест

**React**  
#1 - JavaScript-библиотека для создания пользовательских интерфейсов

**Angular**  
#2 - Angular is an app-design framework and development platform for creating efficient and sophisticated single-page apps.

**Vue**  
#3 - Vue (pronounced /vju:/, like view) is a progressive framework for building user interfaces

**Node.js**  
#4 - Node.js® — це JavaScript-оточення побудоване на JavaScript-рушієві Chrome V8.

Рис. 5 – Тест доданого коду

## Angular Component

**Заголовок - тест**  
#1 - Текст - тест

**React**  
#2 - JavaScript-библиотека для создания пользовательских интерфейсов

**Angular**  
#3 - Angular is an app-design framework and development platform for creating efficient and sophisticated single-page apps.

**Vue**  
#4 - Vue (pronounced /vju:/, like view) is a progressive framework for building user interfaces

**Node.js**  
#5 - Node.js® — це JavaScript-оточення побудоване на JavaScript-рушієві Chrome V8.

Рис. 6 – Тест доданого коду (Самостійна робота)

Код для кнопки, при натисканні на яку в поля title, text завантажуватимуться різні стилі. Встановлено спеціальний декоратор @ViewChild для доступу до DOM елементів і логіка onLoadDefault().

```
export class PostFormComponent implements OnInit {

  title=""
  text=""
  styleToggle=false
  @Output() addPostUser: EventEmitter<Post> = new EventEmitter<Post>()
  @ViewChild('myInputText',{static: false}) myinputText!: ElementRef
  @ViewChild('myInputTitle',{static: false}) myinputTitle!: ElementRef
  constructor() { }

  ngOnInit(): void {
  }

  addPost() {
    if(this.text.trim() && this.title.trim()) {
      const post : Post ={
        title: this.title,
        text: this.text
      }
      this.addPostUser.emit(post)
      this.text=''
      this.title=''
    }
  }

  onLoadDefault () {
    this.styleToggle=!this.styleToggle
    if(this.styleToggle) {
      this.myinputText.nativeElement.style.color = "red"
      this.myinputTitle.nativeElement.style.fontWeight = "bold"
    } else {
      this.myinputText.nativeElement.style.color = 'black'
      this.myinputTitle.nativeElement.style.fontWeight = "normal"
    }
  }
}
```

Рис. 7 – Додав код

# Angular Component

Add post!


Default style


Рис. 8 – Тест доданого коду


## Angular Component

Add post!

Default style

**React**  
#1 - JavaScript-библиотека для создания пользовательских интерфейсов  
Post is long 

**Angular**  
#2 - Angular is an app-design framework and development platform for creating efficient and sophisticated single-page apps.  
Post is long 

**Vue**  
#3 - Vue (pronounced /vju:/, like view) is a progressive framework for building user interfaces  
Post is long 


**Node.js**  
#4 - Node.js® — це JavaScript-оточення побудоване на JavaScript-рушієві Chrome V8.  
Post is long 

Рис. 9 – Тест доданого коду