

## Лабораторна робота № 7

### Реалізація роботи зі списком студентських груп.

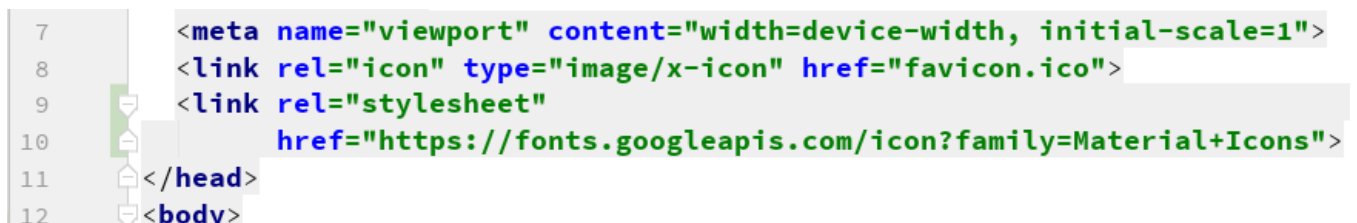
Підключимо до нашого проекту стилі materialize. Встановити materialize можна використавши менеджер пакетів npm за допомогою команди `npm install materialize-css@next`, але ми використаємо інший спосіб - завантажимо необхідні бібліотеки у папку `src/assets`, та виконаємо їх підключення у `angular.json` (рис. 1).



```
29 ],  
30  
31 "styles": [  
32   "src/assets/materialize.min.css",  
33   "src/styles.scss"  
34 ],  
35 "scripts": [  
36   "src/assets/materialize.min.js"  
37 ]
```

Рис. 1. Підключення materialize.

Також за cdn посиланням виконаємо підключення material icons в `index.html` (рис. 2).



```
7 <meta name="viewport" content="width=device-width, initial-scale=1">  
8 <link rel="icon" type="image/x-icon" href="favicon.ico">  
9 <link rel="stylesheet"  
10 href="https://fonts.googleapis.com/icon?family=Material+Icons">  
11 </head>  
12 <body>
```

Рис. 2. Підключення material icons.

Далі змінимо код класів та шаблонів для компонент `app` та `group`, приберемо зайві частини коду та адаптуємо `html` під `materialize css`. Почнімо з `app.component.ts` – тут поки що залишається тільки масив студентських груп (рис. 3).

```

8   export class AppComponent {
9       groups = [
10          {
11              number: 201,
12              faculty: `факультет комп'ютерних наук`,
13              specialty: `комп'ютерні науки`,
14              studentsQuantity: 22
15          },
16          {
17              number: 308,
18              faculty: `факультет комп'ютерних наук`,
19              specialty: `інженерія програмного забезпечення`,
20              studentsQuantity: 24
21          },
22      ];
23  }

```

Рис. 3. app.component.ts.

Далі наведемо код шаблону для app.component.html (рис. 4).

```

1   <nav>
2       <div class="nav-wrapper">
3           <a class="brand-logo">Групи</a>
4       </div>
5   </nav>
6   <div class="container">
7       <div class="row" style="margin-top: 20px;">
8           <div class="col m6 offset-m3 s12">
9               <a class="btn-floating btn-large waves-effect waves-light green">
10                  <i class="material-icons">add</i>
11              </a>
12          </div>
13      </div>
14      <app-group
15          *ngFor="let group of groups"
16          [group]="group">
17      </app-group>
18  </div>

```

Рис. 4. app.component.html.

Переходимо до компоненти group, та почнімо також із класу. Тут поки залишається тільки поле showInfo та вхідний параметр group (рис. 5).

```

9   export class GroupComponent implements OnInit {
10     @Input() group;
11     showInfo = false;
12
13     constructor() { }
14
15     ngOnInit() {
16     }
17 }

```

Рис. 5. group.component.ts.

Шаблон group.component.html також адаптуємо під materialize css (рис. 6).

```

1   <div class="row valign-wrapper">
2     <div class="col m3 offset-m3 s6 center-align">
3       <h5>{{group.number}}</h5>
4     </div>
5     <div class="col m3 s6 center-align">
6       <a class="btn-floating waves-effect waves-light blue"
7         (click)="showInfo=!showInfo">
8         <i *ngIf="showInfo" class="material-icons">
9           keyboard_arrow_up
10         </i>
11         <i *ngIf="!showInfo" class="material-icons">
12           keyboard_arrow_down
13         </i>
14       </a>
15     </div>
16   </div>
17
18   <div class="info" *ngIf="showInfo">
19     <div class="input-field">
20       <input id="number" [(ngModel)]="group.number"
21         type="text" class="validate">
22       <label for="number" class="active">Номер групи</label>
23     </div>
24     <div class="input-field">
25       <input id="faculty" [(ngModel)]="group.faculty"
26         type="text" class="validate">
27       <label for="faculty" class="active">Факультет</label>
28     </div>
29     <div class="input-field">
30       <input id="specialty" [(ngModel)]="group.specialty"
31         type="text" class="validate">
32       <label for="specialty" class="active">Спеціальність</label>
33     </div>

```

```

34 <div class="input-field">
35   <input id="quantity" [(ngModel)]="group.studentsQuantity"
36     type="number" class="validate">
37   <label for="quantity" class="active">Кількість студентів</label>
38 </div>
39 <a class="rem-row btn-floating waves-effect waves-light red">
40   <i class="material-icons">delete</i>
41 </a>
42 </div>

```

Рис. 6. group.component.html.

Також додамо трохи власної стилізації для компонентна group (рис. 7).

```

1 div.info {
2   background: #fff5fd;
3   border-radius: 10px;
4   padding: 10px;
5 }

```

Рис. 7. Group.component.scss.

Переглянемо отриманий результат (рис. 8).

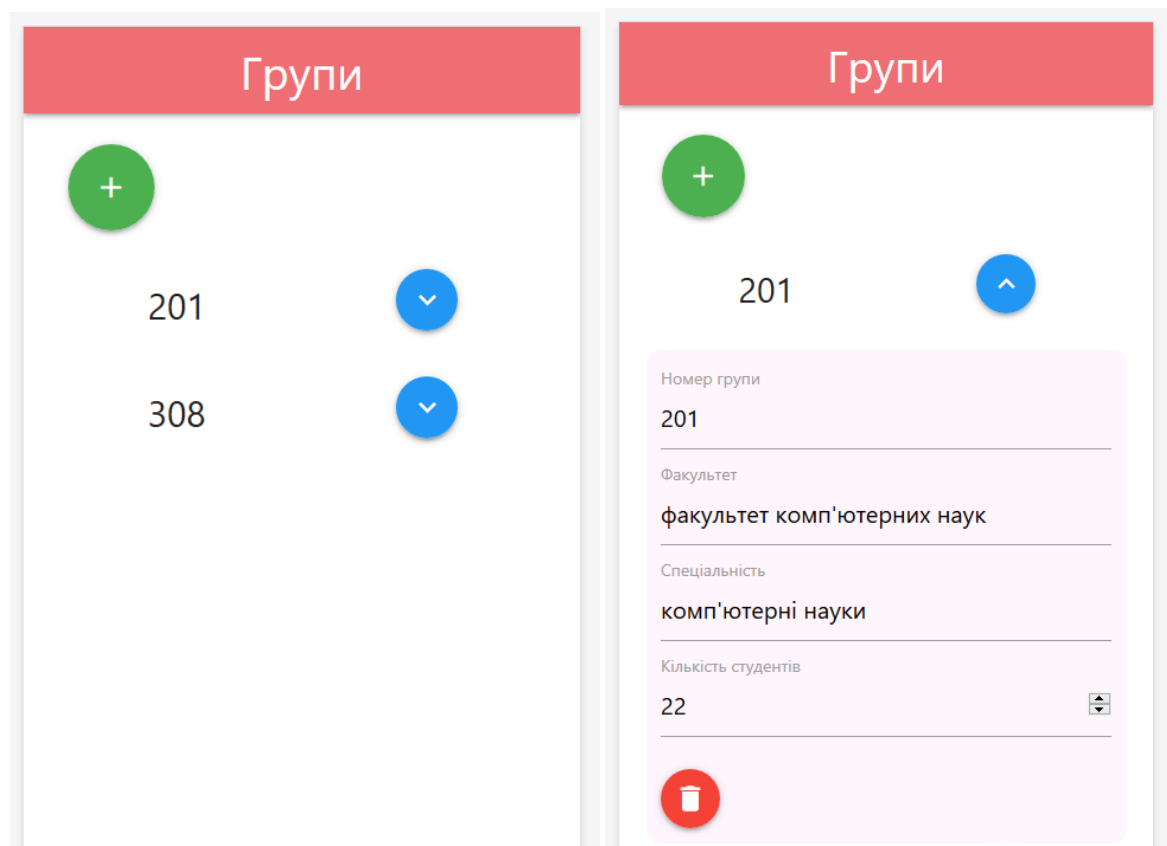


Рис. 8. Перегляд зовнішнього вигляду сторінки.

Залишається реалізувати функціонал додавання та видалення студентських груп. Почнімо із додавання елемента, виділивши кнопку та форму додавання в окремий компонент. Додаємо новий компонент (рис. 9).

```
>ng g c new-group --skipTests
```

Рис. 9. Додавання компонента new-group.

Реалізуємо клас NewGroupComponent, функція якого зводиться до передачі до батьківського компоненту об'єкту нової групи (рис. 10).

```
1  import {Component, EventEmitter, OnInit, Output} from '@angular/core';
2
3  @Component({
4    selector: 'app-new-group',
5    templateUrl: './new-group.component.html',
6    styleUrls: ['./new-group.component.scss']
7  })
8  export class NewGroupComponent implements OnInit {
9
10     @Output() group = new EventEmitter();
11     showForm = false;
12
13     constructor() { }
14
15     ngOnInit() {
16     }
17
18     onSubmit(myForm) {
19         const fields = myForm.form.controls;
20         this.showForm = false;
21         this.group.emit( value: {
22             number: fields.number.value,
23             faculty: fields.faculty.value,
24             specialty: fields.specialty.value,
25             studentsQuantity: fields.studentsQuantity.value
26         });
27     }
28 }
```

Рис. 10. new-group.component.ts.

Переходимо до реалізації макету new-group.component.html (рис. 11).

```

1 <div class="row">
2   <div class="col m6 offset-m3 s12">
3     <a *ngIf="!showForm" (click)="showForm=true"
4       class="btn-floating btn-large waves-effect waves-light green">
5       <i class="material-icons">add</i>
6     </a>
7
8     <form *ngIf="showForm" #myForm="ngForm" novalidate
9       (ngSubmit)="onSubmit(myForm)">
10      <h5>Нова група</h5>
11      <div class="input-field">
12        <input id="number" name="number" ngModel
13          required type="text">
14        <label for="number">Номер групи</label>
15      </div>
16      <div class="input-field">
17        <input id="faculty" name="faculty" ngModel
18          required type="text">
19        <label for="faculty">Факультет</label>
20      </div>
21      <div class="input-field">
22        <input id="specialty" name="specialty" ngModel
23          required type="text">
24        <label for="specialty">Спеціальність</label>
25      </div>
26      <div class="input-field">
27        <input id="quantity" name="studentsQuantity" ngModel
28          required type="number">
29        <label for="quantity">Кількість студентів</label>
30      </div>
31      <div class="form-group">
32        <button type="button" (click)="showForm=false"
33          class="rem-row btn-floating waves-effect
34            waves-light left blue-grey">
35          <i class="material-icons">keyboard_arrow_up</i>
36        </button>
37
38        <button type="submit" [disabled]="myForm.invalid"
39          class="rem-row btn-floating waves-effect
40            waves-light green right">
41          <i class="material-icons">done</i>
42        </button>
43      </div>
44    </form>
45  </div>

```

Рис. 11. new-group.component.html.

Також додаємо трохи власної стилізації (рис. 12).

```

1  div.row {
2    margin-top: 20px;
3  }

```

Рис. 12. new-group.component.scss.

У батьківському компоненті app.component змінимо макет, використавши новий компонент new-group (рис. 13).

```

1  <nav>
2    <div class="nav-wrapper">
3      <a class="brand-logo">Групи</a>
4    </div>
5  </nav>
6  <div class="container">
7    <app-new-group (group)="addGroup($event)"></app-new-group>
8    <app-group
9      *ngFor="let group of groups"
10     [group]="group">
11  </app-group>
12 </div>

```

Рис. 13. Зміни у app.component.html.

У класі AppComponent створимо метод addGroup, що додаватиме новий елемент до масиву (рис. 14).

```

23
24  addGroup(group) {
25    this.groups.push(group);
26  }

```

Рис. 14. Зміни в app.component.ts.

Перевіримо роботу функції додавання нової групи (рис. 15).

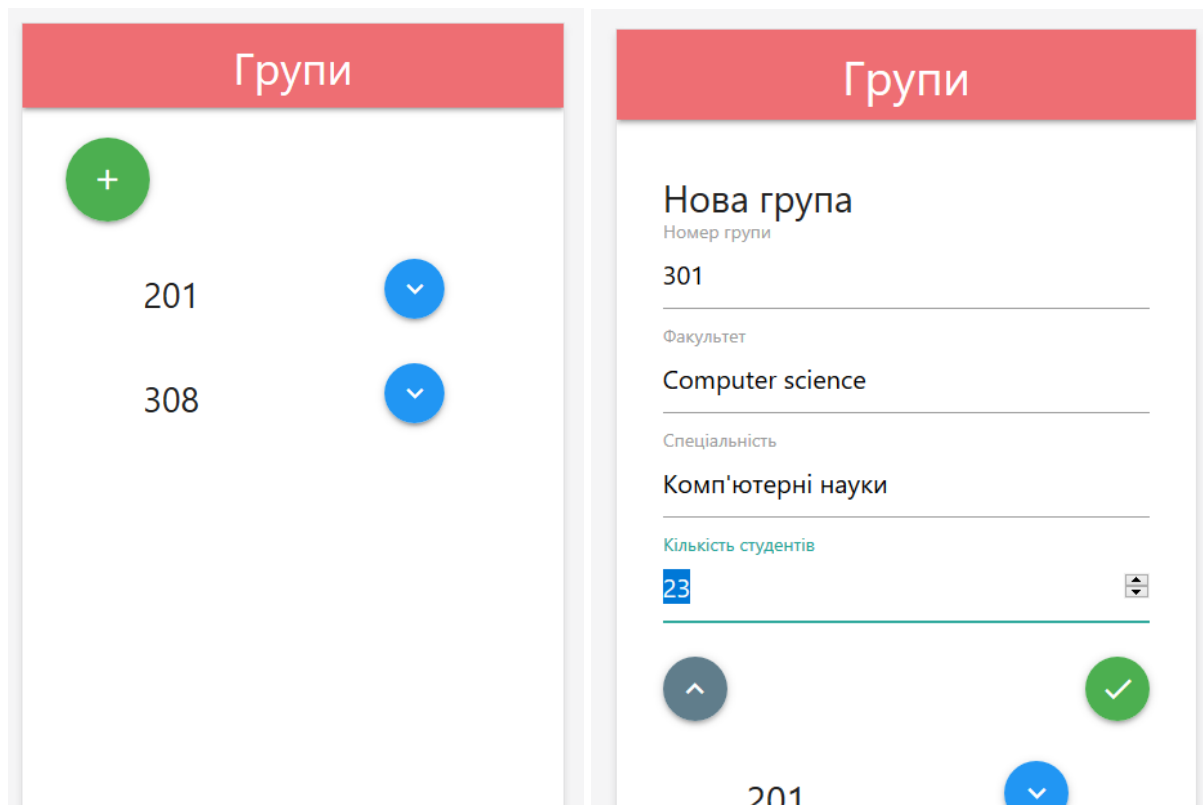


Рис. 15. Додавання нової групи.

Реалізуємо функціонал видалення студентської групи. Додаємо output параметр для компоненти GroupComponent та метод, що ініціюватиме його передачу батьківському компоненту AppComponent. Також додаємо input параметр, у який будемо передавати значення індексу масиву для поточного елемента (рис. 16).

```

9  export class GroupComponent implements OnInit {
10     @Input() group;
11     @Input() grpIndex;
12     @Output() removeGroup = new EventEmitter();
13     showInfo = false;
14
15     ngOnInit() {
16         this.group = this.group;
17     }
18
19     delGroup() {
20         this.removeGroup.emit(this.grpIndex);
21     }
22

```

Рис. 16. Додавання коду до GroupComponent.

У шаблоні group.component.html підключаємо до події натискання на кнопку «видалити» метод delGroup (рис. 17).



```

39     <a class="rem-row btn-floating waves-effect waves-light red"
40       (click)="delGroup()">
41       <i class="material-icons">delete</i>
42     </a>

```

Рис. 17. Натискання на кнопку «видалити».

Також виконаємо доопрацювання компонента AppComponent – у шаблоні при виведенні тегу app-group передаємо туди індекс поточного елемента та приймаємо від нього подію removeGroup (рис. 18).

```

6     <div class="container">
7       <app-new-group (group)="addGroup($event)"></app-new-group>
8       <app-group
9         *ngFor="let group of groups; let i = index"
10        [group]="group" [grpIndex]="i"
11        (removeGroup)="deleteGroup($event)">
12     </app-group>
13 </div>

```

Рис. 18. Зміни у app.component.html.

У класі AppComponent створюємо метод deleteGroup, що відповідатиме за видалення елемента масиву студентських груп (рис. 19).

```

27
28     deleteGroup(index) {
29       this.groups.splice(index, deleteCount: 1);
30     }

```

Рис. 19. Зміни у app.component.ts.

Перевіримо результат, спробувавши видалити одну групу.