

Лабораторна робота 17.

Використання Firebase для збереження даних застосунку.

Частина 2.

Реалізуємо збереження списку студентів у нашій БД. Для цього відкриємо консоль firebase (<https://console.firebase.google.com>), перейдемо до нашої БД на вкладці Database та додамо колекцію в одну з студентських груп (рис. 1).

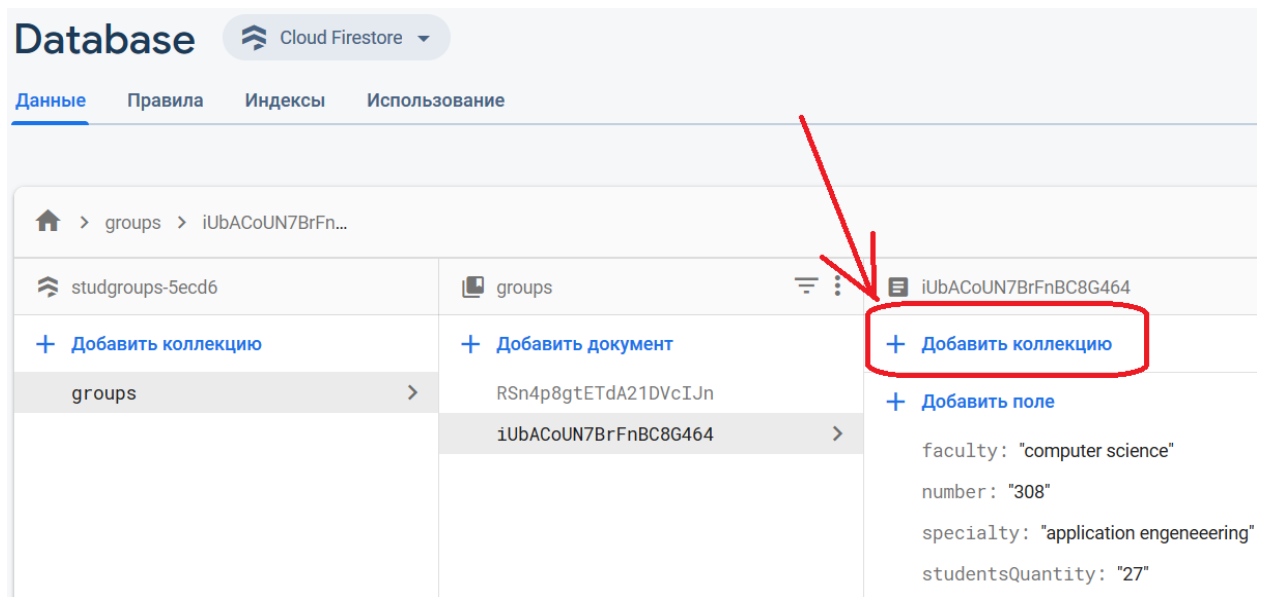


Рис. 1. Додавання колекції у документ.

Вказуємо ідентифікатор для колекції (рис. 2) та додаємо перший елемент колекції (рис. 3).

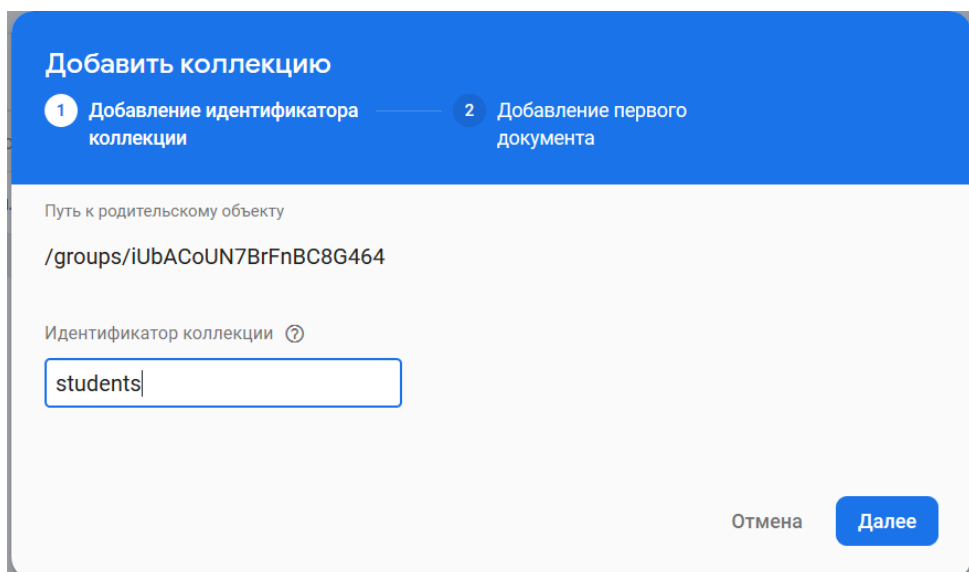


Рис. 2. Ідентифікатор колекції.

Идентификатор документа ?

Сгенерировать

Поле	Тип	Значение
name	= string	Sidorov
gender	= string	man
raiting	= number	85

Рис. 3. Додавання елементу колекції.

Аналогічно додаємо ще декілька елементів (студентів) до цієї та іншої групи (рис. 4).

iUbACoUN7BrFnBC8G464	students	OC1CKvN3s8ydBZxuSS6f
+ Додати колекцію	+ Додати документ	+ Додати колекцію
students	OC1CKvN3s8ydBZxuSS6f oIeAogPrzdd6CrnvZRPE	+ Додати поле
+ Додати поле		gender: "woman" name: "Petrova" raitingn: "92"
faculty: "computer science" number: "308" specialty: "application engeneering"		

Рис. 4. Додавання декількох студентів.

Перейдемо ко коду клієнтського застосунку та виконаємо виведення списку студентів. Для цього спочатку додаємо відповідний метод до сервісу fire-data-getter (рис. 5).

```

28
29  getStudents(id) {
30      return this.afs
31          .doc( path: 'groups/' + id)
32          .collection( path: 'students')
33          .valueChanges();
34  }

```

Рис. 5. Додавання методу для виведення списку студентів.

Тепер виконуємо звернення до даного методу при завантаженні сторінки списку студентів (рис. 6).

```
4 import {FireDataGetterService} from '../services/fire-data-getter.service';
5
11 export class StudentsPage implements OnInit {
12     grpId: string;
13     grpnumb: string;
14     students: any[];
15
16     constructor(private dataGetter: DataGetterService,
17                 private route: ActivatedRoute,
18                 private router: Router,
19                 private fireData: FireDataGetterService) {
20         this.grpId = this.route.snapshot.paramMap.get('id');
21     }
22
23     ngOnInit() {
24         this.grpnumb = this.route.snapshot.paramMap.get('grpnumb');
25         this.fireData.getStudents(this.grpId).subscribe(
26             next: data => {
27                 this.students = data;
28             }
29         );
30     }
}
```

Рис. 6. Отримання списку студентів.

Перейдемо до сторінки списку студентів та переглянемо результат (рис. 7).

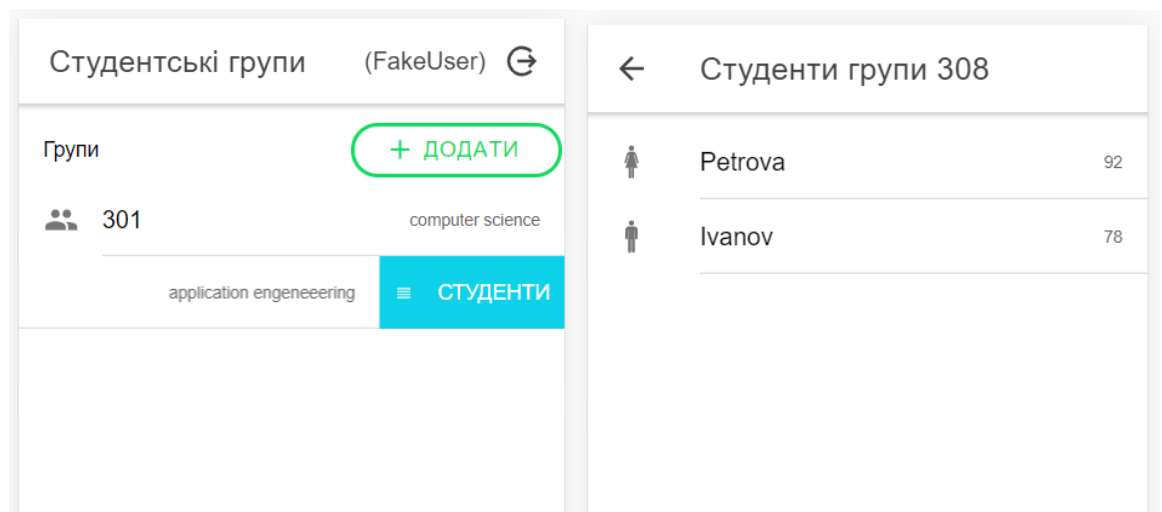


Рис. 7. Виведення списку студентів групи.

Реалізуємо також команди модифікації даних студентських груп. Почнімо із операції зміни даних. Реалізуємо відповідний метод у сервісі fire-data-getter (рис. 8).

```

36   editGroup(group) {
37       return this.afs
38           .doc( path: 'groups/' + group.id)
39           .update( data: {
40               number: group.number,
41               specialty: group.specialty,
42               faculty: group.faculty,
43               studentsQuantity: group.studentsQuantity
44           });
45   }

```

Рис. 8. Реалізація метода зміни даних студентської групи.

В інтерфейсі StudGroup у data-getter.service.ts змінимо тип поля id з number на string (рис. 9).

```

5   export interface StudGroup {
6       id: string;
7       number: number;
8       faculty: string;
9       specialty: string;
10      studentsQuantity: number;
11  }

```

Рис. 9. Зміна типу поля id у інтерфейсі StudGroup.

Виконаємо звернення до створеного метода у компоненті stud-group.component.ts (рис. 10).

```

3   import {FireDataGetterService} from '../services/fire-data-getter.service';
4
17
18   constructor(private dataGetter: DataGetterService,
19               private fireData: FireDataGetterService) {}
20
45
46   saveGroup() {
47       this.fireData.editGroup(this.studentGroup);
48   }

```

Рис. 10. Виклик метода editGroup по натисканню на кнопку у компоненті.

Перейдемо до застосунку та змінимо номер в одній із студентських груп (рис. 11).

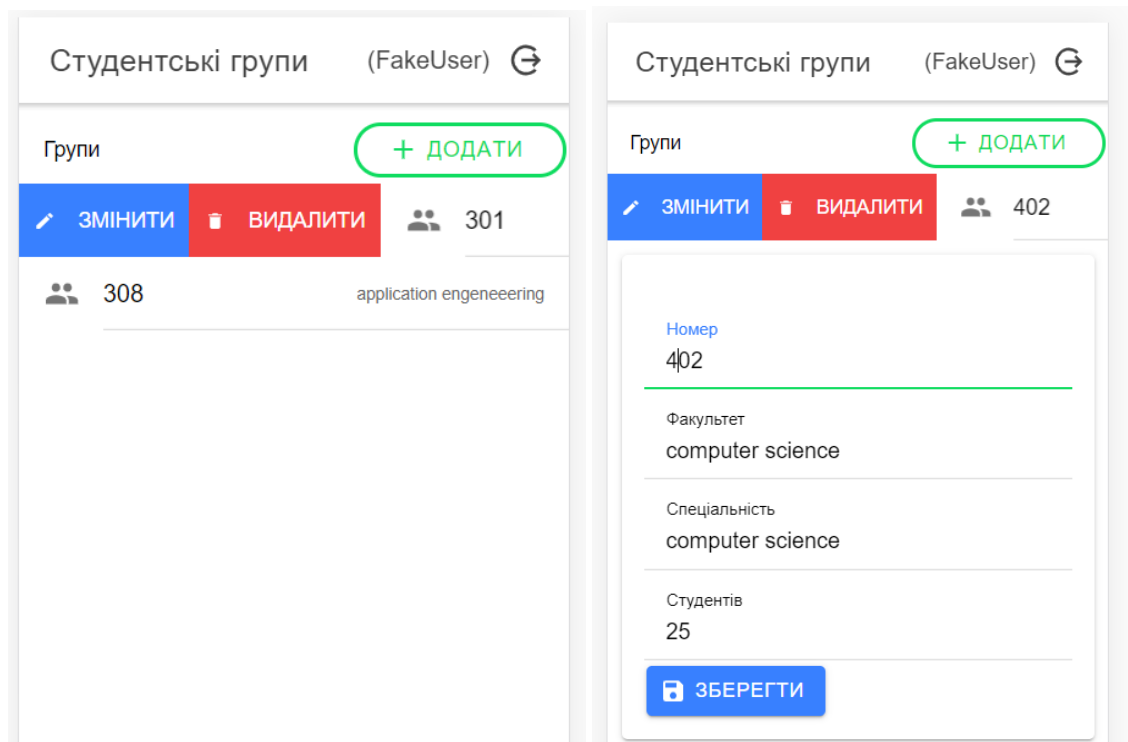


Рис. 11. Зміна номеру групи.

Зберігаємо зміни те перевіряємо їх у консолі firebase (рис. 12).

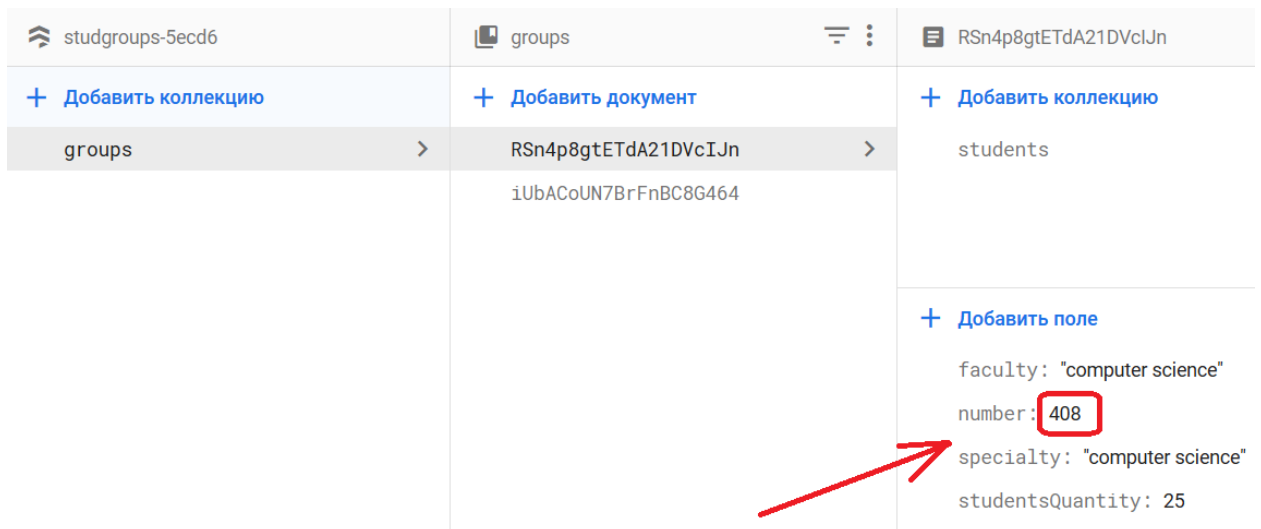


Рис. 12. Перевірка виконаних змін у консолі.

Виконаємо реалізацію функції додавання нової групи. Додаємо метод `addGroup` до нашого сервісу `fire-data-getter` (рис. 13).

```

46
47
48   addGroup(group) {
49     this.afs.collection( path: 'groups')
50       .add({
51         number: group.number,
52         specialty: group.specialty,
53         faculty: group.faculty,
54         studentsQuantity: group.studentsQuantity
55       });
56   }

```

Рис. 13. Метод addGroup.

Та виконуємо звернення до методу на сторінці home.page.ts при отриманні даних від компоненти студентських груп (рис. 14).

```

51
52   addGroup(group) {
53     this.fireData.addGroup(group);
54     this.showNew = false;
55   }

```

Рис. 14. Зміни у home.page.ts.

Додаємо нову групу (рис. 15) та перевіряємо результат у консолі firebase (рис. 16).

The image shows two side-by-side screenshots of a web application interface for managing student groups. Both screenshots have a header with the title 'Студентські групи' and a user profile '(FakeUser)' with a refresh icon.

The left screenshot displays a form titled 'Нова група' (New group) for adding a new entry. It includes a green '+ ДОДАТИ' (Add) button at the top right. The form fields are:

- Номер (Number): 201
- Факультет (Faculty): Комп.наук (Computer Science)
- Спеціальність (Specialty): комп'ютерні науки (Computer Science)
- Студентів (Students): 29

 At the bottom of the form are two buttons: a blue '+ ДОДАТИ' (Add) button and a red '× ВІДМІНА' (Cancel) button.

The right screenshot shows the updated list of groups. It also has the green '+ ДОДАТИ' button. The list contains three entries:

- Group 201: комп'ютерні науки (Computer Science)
- Group 408: computer science
- Group 308: application engeneering

Рис. 15. Додавання нової групи.

groups	DUiHg2pdk3qNc16Z3MYc
+ Добавить документ	+ Добавить коллекцию
DUiHg2pdk3qNc16Z3MYc >	+ Добавить поле
RSn4p8gtETdA21DVcIJn	faculty: "Комп.наук"
iUbACoUN7BrFnBC8G464	number: 201
	specialty: "комп'ютерні науки"
	studentsQuantity: 29

Рис. 16. Перевірка операції додавання групи.

Реалізуємо останню операцію – видалення студентської групи. Також починаємо із відповідного методу сервісу fire-data-getter (рис. 17).

```

56
57   delGroup(group) {
58       return this.afs
59           .doc( path: 'groups/' + group.id)
60           .delete();
61   }

```

Рис. 17. Метод видалення студентської групи.

Звертаємось до реалізованого методу із відповідної події на сторінці home.page.ts (рис. 18).

```

43
44   delete(group) {
45       this.fireData.delGroup(group);
46   }

```

Рис. 18. Звернення до методу видалення студентської групи.

Аналогічно до зміни та додавання, видаляємо одну з груп, та перевіряємо результат у консолі firebase.