

## Лабораторна робота № 3

### Знайомство з Ionic Framework.

Ionic - це завершений SDK з відкритим кодом для розробки гібридних мобільних застосунків. Оригінальна версія була побудована на вершині AngularJS та Apache Cordova. Однак остання версія була випущена як набір веб-компонентів, що дозволяють користувачеві вибирати будь-який фреймворк розробки інтерфейсу користувача, наприклад Angular, React або Vue.js. Також можливо використовувати компоненти Ionic окремо, без використання жодного фреймворку.

Ionic надає інструменти та сервіси для розробки гібридних мобільних, настільних та прогресивних веб-застосунків на основі сучасних технологій веб-розробки, використовуючи такі веб-технології, як CSS, HTML5 та SASS. Зокрема, мобільні застосунки можуть бути побудовані за допомогою даних веб-технологій, а потім встановлені на пристрої, використовуючи Cordova або Capacitor.

У даній роботі розглянемо використання компонентів Ionic для створення застосунку обліку студентських груп. Даний застосунок буде представлений у вигляді веб-сторінки, тому встановлення фреймворку поки не обов'язкове. Виконаємо підключення необхідних компонент через cdn-посилання (рис. 1).

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <script type="module" src="https://cdn.jsdelivr.net/npm/@ionic/core/dist/ionic/
      ionic.esm.js"></script>
8      <script nomodule src="https://cdn.jsdelivr.net/npm/@ionic/core/dist/ionic/ionic.js"></
      script>
9      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@ionic/core/css/
      ionic.bundle.css"/>
10     <title>Students groups</title>
11 </head>
```

Рис. 1. Підключення ionic через cdn.

Далі реалізуємо зовнішній вигляд головної сторінки, додавши у тіло header та сітку із елементом список (рис. 2).

```
12 <body>
13   <ion-header>
14     <ion-toolbar>
15       <ion-title>Groups</ion-title>
16     </ion-toolbar>
17   </ion-header>
18   <section id="main">
19     <ion-grid>
20       <ion-row>
21         <ion-col size="12" size-md="6" offset-md="3">
22           <ion-button color="success" size="large">
23             <ion-icon name="add-circle-outline"></ion-icon>
24           </ion-button>
25         </ion-col>
26       </ion-row>
27       <ion-row>
28         <ion-col size="12" size-md="6" offset-md="3">
29           <ion-list>
30             <ion-list-header>
31               <ion-label>Group</ion-label>
32             </ion-list-header>
33             <ion-item>
34               <ion-label>301</ion-label>
35               <ion-button color="primary" size="medium">
36                 <ion-icon name="create"></ion-icon>
37               </ion-button>
38               <ion-button color="danger" size="medium">
39                 <ion-icon name="trash"></ion-icon>
40               </ion-button>
41             </ion-item>
42           </ion-list>
43         </ion-col>
44       </ion-row>
45     </ion-grid>
46   </section>
47 </body>
48 </html>
```

Рис. 2. Шаблон головної сторінки.

Завантажимо застосунок у браузері та переглянемо зовнішній вигляд сторінки (рис. 3).

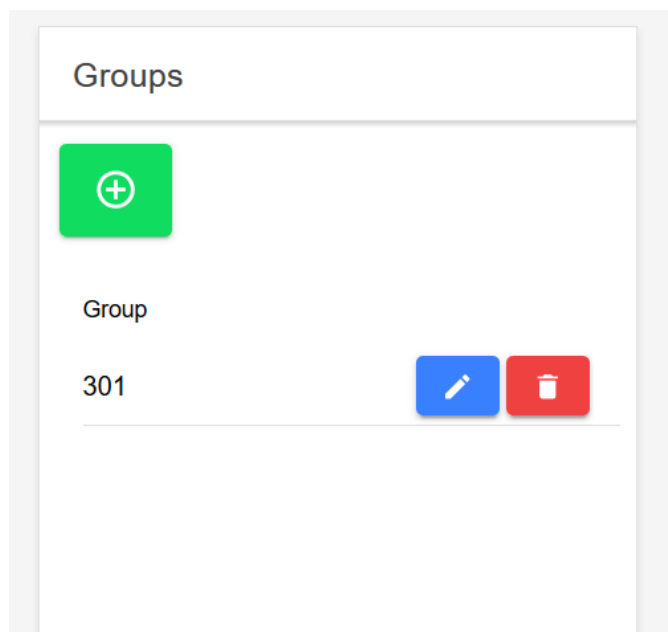


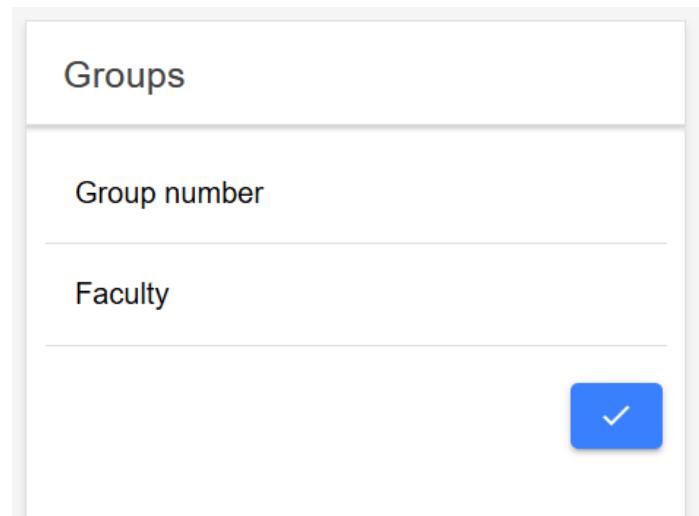
Рис. 3. Зовнішній вигляд головної сторінки.

У цей раз реалізуємо наш застосунок за принципом односторінкового застосунку (SPA), тому форму додавання та редагування елемента реалізуємо в окремій секції файлу index.html (рис. 4).

```
40 <section id="edit">
41   <ion-grid>
42     <ion-row>
43       <ion-col size="12" size-md="6" offset-md="3">
44         <ion-item>
45           <ion-label position="floating">Group number</ion-label>
46           <ion-input id="number"></ion-input>
47         </ion-item>
48         <ion-item>
49           <ion-label position="floating">Faculty</ion-label>
50           <ion-input id="faculty"></ion-input>
51         </ion-item>
52         <ion-button id="save" color="primary" size="medium" float-right
53           style="margin-top: 20px;">
54           <ion-icon name="checkmark"></ion-icon>
55         </ion-button>
56       </ion-col>
57     </ion-row>
58   </ion-grid>
</section>
```

Рис. 4. Шаблон сторінки редагування елемента.

Зовнішній вигляд форми редагування наведено на рис. 5.



The image shows a web form for editing a group. It has a title 'Groups' at the top. Below the title are two input fields: 'Group number' and 'Faculty'. At the bottom right of the form is a blue button with a white checkmark, indicating a 'save' or 'confirm' action.

Рис. 5. Зовнішній вигляд форми редагування.

Тепер реалізуємо js-код для представлення даних та обробки дій користувача. Файл data.js у порівнянні із попередніми роботами фактично залишається без змін (рис. 6).

```
1 function saveGroups(groups) {  
2   |   localStorage.setItem("groups_data", JSON.stringify(groups));  
3   }  
4  
5   var groups = localStorage.getItem("groups_data");  
6   if (localStorage.getItem("groups_data") === null) {  
7     |   groups = [  
8     |   |   {  
9     |   |   |   number: 301,  
10    |   |   |   faculty: 'Computer science'  
11    |   |   |   },  
12    |   |   |   {  
13    |   |   |   number: 302,  
14    |   |   |   faculty: 'Computer science'
```

```

15     }
16   ];
17   saveGroups(groups);
18 } else {
19   groups = JSON.parse(groups);
20 }

```

Рис. 6. Файл data.js.

Далі наведемо реалізацію обробки виводу даних та дій користувача у файлі my.js. При реалізації функціоналу не будемо використовувати додаткових js бібліотек. На початку при завантаженні документу будемо виводити список груп до елементу ion-list. Також сховаємо секцію редагування даних. Та додамо події для кнопок додавання, зміни та видалення даних (рис. 7).

```

1  let editType = '';
2  let number = '';
3  document.addEventListener("DOMContentLoaded", function(event) {
4    var rowText;
5    var list = document.getElementById('list');
6
7    for(var row of groups) {
8      var rowText = document.createElement('ion-item');
9
10     rowText.innerHTML = `<ion-label>${row.number}</ion-label>
11     <ion-button class="edit" color="primary" size="medium" rowid="${row.number}"
12     ">
13     <ion-icon name="create"></ion-icon>
14     </ion-button>
15     <ion-button class="delete" color="danger" size="medium" rowid="${
16     {row.number}}">
17     <ion-icon name="trash"></ion-icon>
18     </ion-button>`;
19     list.appendChild(rowText);
20
21
22   }
23
24   document.querySelector('#main').style.display = "";
25   document.querySelector('#edit').style.display = 'none';
26
27   document.querySelectorAll('.edit')
28     .forEach(input => input.addEventListener('click', ({target}) => {
29     editType = "edit";
30     number = target.getAttribute('rowid');
31     let group = groups.find((g)=>g.number == number);

```

```

27 |
28 |     document.getElementById('number').value = group.number;
29 |     document.getElementById('faculty').value = group.faculty;
30 |     document.querySelector('#main').style.display = "none";
31 |     document.querySelector('#edit').style.display = "";
32 | });
33 |
34 |
35 | document.querySelectorAll('.delete')
36 |     .forEach(input => input.addEventListener('click', ({target}) => {
37 |         number = target.getAttribute('rowid');
38 |         saveGroups(groups.filter((g)=>g.number != number));
39 |         location.reload();
40 |     }));
41 |
42 | document.querySelector('#save').addEventListener('click', () => {
43 |     if (editType == "add") {
44 |         groups.push({
45 |             number: document.getElementById('number').value,
46 |             faculty: document.getElementById('faculty').value
47 |         });
48 |     } else {
49 |         let group = groups.find((g)=>g.number == number);
50 |         group.number = document.getElementById('number').value;
51 |         group.faculty = document.getElementById('faculty').value;
52 |     }
53 |     saveGroups(groups);
54 |     location.reload();
55 | });
56 |
57 | document.querySelector('#add').addEventListener('click', () => {
58 |     document.querySelector('#main').style.display = "none";
59 |     document.querySelector('#edit').style.display = "";
60 |     document.getElementById('number').value = "";
61 |     document.getElementById('faculty').value = "";
62 |     editType = "add";
63 | });

```

Рис. 7. Файл my.js.

Повернімось до файлу index.html та виконаємо невеликі зміни. Підключимо створені data.js та my.js, видалимо тестовий рядок із групою 301 та додамо ідентифікатори та класи для декількох елементів для можливості подальшого звернення до них із коду js. Повний код index.html після вказаних змін наведено на рис. 8.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <script type="module" src="https://cdn.jsdelivr.net/npm/@ionic/core/dist/ionic/
      ionic.esm.js"></script>
8      <script nomodule src="https://cdn.jsdelivr.net/npm/@ionic/core/dist/ionic/ionic.js"></
      script>

9      <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@ionic/core/css/
      ionic.bundle.css"/>
10     <script src="data.js"></script>
11     <script src="my.js"></script>
12     <title>Students groups</title>
13 </head>
14 <body>
15     <ion-header>
16         <ion-toolbar>
17             <ion-title>Groups</ion-title>
18         </ion-toolbar>
19     </ion-header>

20     <section id="main">
21         <ion-grid>
22             <ion-row>
23                 <ion-col size="12" size-md="6" offset-md="3">
24                     <ion-button id="add" color="success" size="large">
25                         <ion-icon name="add-circle-outline"></ion-icon>
26                     </ion-button>
27                 </ion-col>
28             </ion-row>

29             <ion-row>
30                 <ion-col size="12" size-md="6" offset-md="3">
31                     <ion-list id="list">
32                         <ion-list-header>
33                             <ion-label>Group</ion-label>
34                         </ion-list-header>
35                     </ion-list>
36                 </ion-col>
37             </ion-row>
38         </ion-grid>
39     </section>

40     <section id="edit">
41         <ion-grid>
42             <ion-row>
43                 <ion-col size="12" size-md="6" offset-md="3">
44                     <ion-item>
45                         <ion-label position="floating">Group number</ion-label>
46                         <ion-input id="number"></ion-input>
47                     </ion-item>
48                     <ion-item>
49                         <ion-label position="floating">Faculty</ion-label>
50                         <ion-input id="faculty"></ion-input>
51                     </ion-item>

```

```

52 |         <ion-button id="save" color="primary" size="medium" float-right
53 |           style="margin-top: 20px;"
54 |           <ion-icon name="checkmark"></ion-icon>
55 |         </ion-button>
56 |       </ion-col>
57 |     </ion-row>
58 |   </ion-grid>
59 | </section>
60 </body>
</html>

```

Рис. 8. Index.html після виконання змін.

На рис. 9 наведемо вигляд користувацького інтерфейсу отриманого застосунку.

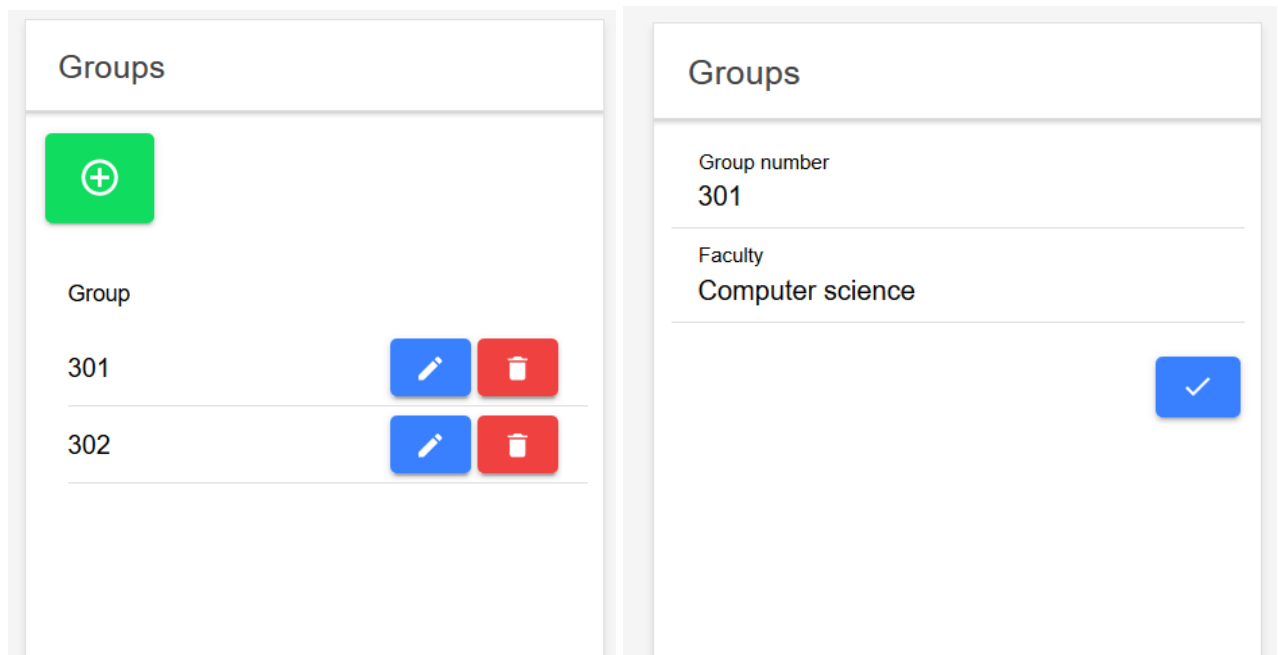


Рис. 9. Головна сторінка та форма редагування елементу.

### Індивідуальне завдання.

На базі іоніс та JS без використання додаткових фреймворків реалізувати застосунок згідно варіанту індивідуального завдання.