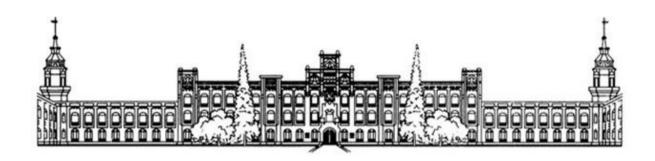
МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»



Лабораторна робота № 3

з дисципліни «Реактивне програмування»

на тему:

«Створення проєкту «Todo»

Перевірила доцент Полупан Ю. В. Виконав студент ФІОТ групи IC-01 Адамов Денис

Зміст

Лабораторне заняття №3: Створення проекту «Todo»	3
Огляд моделі даних у проекті «Todo»	4
Пакет Angular Material	5
Двосторонні прив'язки даних	8
Функція filter y JavaScript	12
Розгорнутий на Firebase застосунок Todo	13

Лабораторне заняття №3: Створення проекту «Todo»

Мета: Знайомство з основним процесом розробки Angular-додатку з використанням редактора Visual Studio Code. Навчитися застосовувати компоненти Angular Material для стилізації контенту та створювати моделі даних. Навчитися створювати двосторонні прив'язки даних.

Завдання:

- I) Встановити Visual Studio Code, пакет Angular Material. Створити каркас додатку «Todo». Додати наступні можливості до проекту «Todo»: відображення списку завдань у вигляді таблиці; додавання нових завдань користувачем; фільтрування завдань користувачем.
- II) Зробити звіт по роботі.
- III) Angular-додаток «Todo» розгорнути на платформі Firebase у проекті з ім'ям «ПрізвищеГрупаLaba3».

Огляд моделі даних у проекті «Todo»

```
lab3 > todo > src > app > TS todoltem.ts > ...
1    export class TodoItem {
2         constructor(public task: string, public complete: boolean = false) {}
3     }
```

todoItem.ts

```
lab3 > todo > src > app > TS todoList.ts > ...

1    import { TodoItem } from './todoItem';

2    export class TodoList {

3        constructor(public user: string, private todoItems: TodoItem[] = []) {}

4        get items(): readonly TodoItem[] {

5            return this.todoItems;

6        }

7        addItem(task: string) {

8             this.todoItems.push(new TodoItem(task));

9        }

10    }
```

todoList.ts

Така модель даних дозволяє відображати список справ наступним чином



Abuba: список справ

2 елементів

Пакет Angular Material

Модулі Angular Material підключаються шляхом імпортування у файлі app.module.ts.

```
lab3 > todo > src > app > TS app.module.ts > ...
      import { NgModule } from '@angular/core';
      import { BrowserModule } from '@angular/platform-browser';
      import { AppComponent } from './app.component';
      import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
      import { FormsModule } from '@angular/forms';
      import { MatButtonModule } from '@angular/material/button';
      import { MatToolbarModule } from '@angular/material/toolbar';
      import { MatIconModule } from '@angular/material/icon';
      import { MatBadgeModule } from '@angular/material/badge';
      import { MatTableModule } from '@angular/material/table';
      import { MatCheckboxModule } from '@angular/material/checkbox';
      import { MatFormFieldModule } from '@angular/material/form-field';
      import { MatInputModule } from '@angular/material/input';
      import { MatSlideToggleModule } from '@angular/material/slide-toggle';
      @NgModule({
        declarations: [AppComponent],
 17
        imports: [
          BrowserModule,
          BrowserAnimationsModule,
          FormsModule,
          MatButtonModule,
          MatToolbarModule,
          MatIconModule,
          MatBadgeModule,
          MatTableModule.
          MatCheckboxModule,
          MatFormFieldModule,
          MatInputModule,
         MatSlideToggleModule,
        providers: [],
        bootstrap: [AppComponent],
      export class AppModule {}
```

Компоненти застосовуються за допомогою елементів і атрибутів HTML у файлі шаблону

```
lab3 > todo > src > app > # app.component.css >
1    .spacer {
2     flex: 1 1 auto;
3  }
```

Із застосуванням такого стилю отримаємо наступний вигляд застосунку:

Abuba's To Do List



Додамо таблицю завдань.

Компонент Angular Material table застосовується шляхом додавання mattable атрибут до стандартного HTML елементу таблиці, а дані, які міститиме таблиця, вказуються за допомогою спеціального dataSource атрибуту.

```
lab3 > todo > src > app > ◆ app.component.html > ...
   Go to component
   <mat-toolbar color="primary" class="mat-elevation-z3">
    <span class="spacer"></span>
    <span>{{ username }}'s To Do List </span>
    <span class="spacer"></span>
    <mat-icon matBadge="{{ itemCount }}" matBadgeColor="accent">
     checklist
   </mat-toolbar>
   <div class="tableContainer">
    <ng-container matColumnDef="id">
12
      №
      {{ i + 1 }}
     </ng-container>
     <ng-container matColumnDef="task">
      {{ item.task }}
     </ng-container>
     <ng-container matColumnDef="done">
      Done
      {{ item.complete }}
     </ng-container>
     26
   </div>
```

Abuba's To Do List



Nº	Task	Done
1	Зробити пробіжку	true
2	Купити квіти	false
3	Забрати квитки	false

Двосторонні прив'язки даних

Замінимо простий текст позначення статусу завдання на чекбокс і виконаємо двосторонню прив'язку.

```
lab3 > todo > src > app > 💠 app.component.html > 🕏 div.tableContainer > 🕏 table.mat-elevation-z3.fullWidth >
    Go to component
   <mat-toolbar color="primary" class="mat-elevation-z3">
    <span class="spacer"></span>
    <span>{{ username }}'s To Do List </span>
    <span class="spacer"></span>
    <mat-icon matBadge="{{ itemCount }}" matBadgeColor="accent">
    checklist
    <div class="tableContainer">
    <ng-container matColumnDef="id">
      \mathbb{\text{\text{M}}}
{{ i + 1 }}

      </ng-container>
      <ng-container matColumnDef="task">
      Task
{{ item.task }}
      </ng-container>
      <ng-container matColumnDef="done">
       <mat-checkbox [(ngModel)]="item.complete" color="primary">
22
         {{ item.complete }}
      </ng-container>
```

Abuba's To Do List

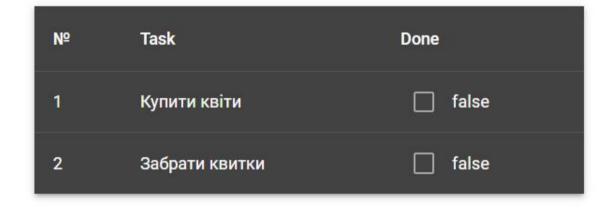


Nº	Task	Done
1	Зробити пробіжку	✓ true
2	Купити квіти	false
3	Забрати квитки	false

Виконаємо прив'язку до події для додавання завдань

```
addItem(newItem: string) {
   if (newItem != '') {
      this.list.addItem(newItem);
   }
}
```

Abuba's To Do List Hoва справа Нічого не робити Додати



Abuba's To Do List



Нова справа	
Enter to-do description	Додати
Litter to do decomption	

Nº	Task	Done
1	Купити квіти	false
2	Забрати квитки	false
3	Нічого не робити	false

Підключимо перемикач, що змінюватиме тип відображення завдань: показувати виконані чи ні.

```
get items(): readonly TodoItem[] {
    return this.list.items.filter(item => this.showComplete || !item.complete);
}
addItem(newItem: string) {
    if (newItem != '') {
        this.list.addItem(newItem);
    }
}
showComplete: boolean = false;
```

Abuba's To Do List



New To Do Add

#	Task	Done
1	Go for run	∨
2	Get flowers	✓
3	Collect tickets	



Функція filter y JavaScript

Функція filter використовується для маніпуляцій над масивами. У нашому випадку використали для вибору лише невиконаних завдань.

```
lab3 > todo > src > app > TS app.component.ts > 43 AppComponent > 15 (get) items
      import { Component } from '@angular/core';
      import { TodoList } from './todoList';
      import { TodoItem } from './todoItem';
      @Component({
        selector: 'app-root',
        templateUrl: './app.component.html',
        styleUrls: ['./app.component.css'],
       export class AppComponent {
        private list = new TodoList('Abuba', [
          new TodoItem('Зробити пробіжку', true),
 11
          new TodoItem('Купити квіти'),
 12
 13
          new TodoItem('Забрати квитки'),
         1);
 15
        get username(): string {
         return this.list.user;
 17
        get itemCount(): number {
         return this.list.items.filter(item => !item.complete).length;
 21
        get items(): readonly TodoItem[] {
 22
       return this.list.items.filter(item => !item.complete);
 23
 24
 25
```

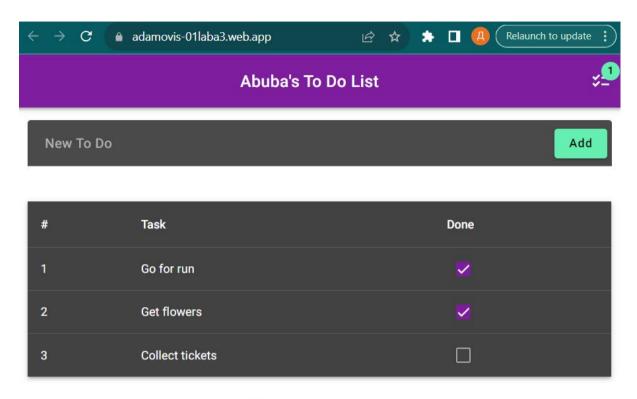
Abuba's To Do List



Nº	Task	Done
1	Купити квіти	false
2	Забрати квитки	false

Розгорнутий на Firebase застосунок Todo

https://adamovis-01laba3.web.app/



Show Completed Items