

Міністерство освіти і науки України
ДВНЗ «Прикарпатський національний університет імені Василя Стефаника»
Кафедра комп'ютерних наук та інформаційних систем

Лабораторна робота № 4

3 курсу: Програмування та підтримка веб-застосунків

Тема: Застосування AngularJS для реалізації сценаріїв обробки та
передачі даних між контекстами

Виконав ст.гр. КН-4

Юрць Т.В.

(Особистий підпис, дата)

Прийняв викладач

Ізмайлов А. В.

(Особистий підпис, дата)

м.Івано-Франківськ

2020

Мета роботи: Навчитись застосовувати AngularJS для реалізації сценаріїв обробки та передачі даних між контекстами.

Теоретичні відомості:

AngularJS — JavaScript-фреймворк з відкритим програмним кодом, який розробляє Google. Призначений для розробки односторінкових додатків, що складаються з одної HTML сторінки з CSS і JavaScript. Його мета — розширення браузерних застосунків на основі шаблону Модель-вид-контролер (MVC), а також спрощення їх тестування та розробки.

Фреймворк працює зі сторінкою HTML, що містить додаткові атрибути і пов'язує області вводу або виводу сторінки з моделлю, яка є звичайними змінними JavaScript. Значення цих змінних задаються вручну або отримуються зі статичних або динамічних JSON-даних.

За даними служби аналізу JavaScript для Libscore, AngularJS використовується на веб-сайтах Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News та близько 12,000 інших сайтів з 1 мільйона протестованих у жовтні 2016 року. AngularJS наразі входить до трійки проектів, що набрали найбільшу кількість зірок на GitHub.

AngularJS побудований на переконанні, що декларативне програмування слід використовувати для створення користувацьких інтерфейсів та підключення компонентів програмного забезпечення, тоді як імперативне програмування краще підходить для визначення бізнес-логіки додатка. Фреймворк адаптує та розширює традиційний HTML, щоб представити динамічний вміст через двостороннє зв'язування даних, що дозволяє автоматично синхронізувати моделі та перегляди.

Конструктивні цілі AngularJS включають:

- відокремлення DOM-маніпуляцій від логіки додатків, що суттєво впливає на спосіб побудови коду.
- відокремлення клієнтської частини програми від серверної. Це дозволяє розробці працювати паралельно і використовувати повторно обидві сторони.
- проведення розробника через весь шлях створення додатку: від проектування користувацького інтерфейсу, через написання бізнес-логіки, до тестування .
- AngularJS реалізує шаблон MVC для відокремлення представлення, даних та логічних компонентів. Використовуючи впровадження залежності, Angular традиційно постачає сервісні служби, такі як контролери, залежні від вигляду, для клієнт-серверних веб-додатків. Відповідно, зменшується навантаження на сервер.

Результати роботи:

Купити

Апельсин7

Купити

Пачка Макарон6

Купити

Слива31

Купити

Хліб2

Купити

Мандарин18

Купити

Куплено

Ще нічого не куплено

Рисунок 1

Купити

Апельсин7

Купити

Пачка Макарон6

Купити

Хліб2

Купити

Куплено

Куплено31Слива

Куплено18Мандарин

Рисунок 2

Код програми:

index.html

```
<!DOCTYPE html>
<html">
<head>
  <meta charset="utf-8">
  <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.8.0/angular.min.js"></script>
  <script type="text/javascript" src="app.js"></script>
  <link rel="stylesheet" href="style.css">
  <title>Yurts Lab 4</title>
</head>

<body ng-app="myModule">

  <div class="cards">
    <div class='card' ng-controller='buyItemController as buyCntr'>
      <h2>Купити</h2>
      <div ng-if="buyCntr.shopItemsToBuy.length == 0">
        Все куплено
      </div>
      <div ng-if="buyCntr.shopItemsToBuy.length > 0">
        <div class="shop-item-row" ng-repeat="item in buyCntr.shopItemsToBuy">
          <span>{{item.name}}</span>
          <span>{{item.quantity}}</span>
          <button ng-click="buyCntr.addItemToBought($index)">Купити</button>
        </div>
      </div>
    </div>

    <div class='card' ng-controller='boughtItemController as boughtCntr'>
      <h2>Куплено</h2>
      <div ng-if="boughtCntr.boughtShopItems.length == 0">
        Ще нічого не куплено
      </div>
      <div ng-if="boughtCntr.boughtShopItems.length > 0">
```

```

        <div class="shop-item-row" ng-
repeat="item in boughtCntr.boughtShopItems">
            <span>Куплено</span> <span>{{item.quantity}}</span> <span></sp
an>{{item.name}}
        </div>
    </div>
</div>
</div>
</body>
</html>

```

style.css

```

@import url('https://fonts.googleapis.com/css2?family=Play:wght@400;700&display=sw
ap');
html,body{
    height: 100%;
}
body {
    font-family: 'Play', sans-serif;
    margin: 0;
    background-color: #dddddd;
}
.cards{
    display: flex;
    height: 100%;
    justify-content: space-around;
    align-items: center;
    margin: 0 400px;
}
.card{
    border-radius: 20px;
    min-height: 270px;
    min-width: 220px;
    padding: 20px 10px;
    background-color: #f1f1f1;
}
.shop-item-row{
    padding-bottom: 10px;
    display: flex;
    justify-content: space-between}

```

```
(function () {

    'use strict';

    angular.module('myModule', [])
        .controller('buyItemController', BuyItemController)
        .controller('boughtItemController', BoughtItemController)
        .service('shopStorageService', ShopStorageService);

    BuyItemController.$inject = ['shopStorageService'];
    function BuyItemController(service) {
        var controller = this;
        controller.shopItemsToBuy = service.getShopItemsToBuy();

        controller.addItemToBought = function (shopItemIndex) {
            service.addBoughtItem(shopItemIndex);
        }
    };

    BoughtItemController.$inject = ['shopStorageService'];
    function BoughtItemController(service) {
        var controller = this;
        controller.boughtShopItems = service.getBoughtShopItems();
    };

    class ShopItem {
        constructor(name, quantity) {
            this.name = name;
            this.quantity = quantity;
        }
    };

    function ShopStorageService() {
```

```
var service = this;

var shopItemsToBuy = [
    new ShopItem('Апельсин', 7),
    new ShopItem('Пачка Макарон', 6),
    new ShopItem('Слива', 31),
    new ShopItem('Хліб', 2),
    new ShopItem('Мандарин', 18)
];

var boughtShopItems = [];

service.getShopItemsToBuy = function () {
    return shopItemsToBuy;
};

service.getBoughtShopItems = function () {
    return boughtShopItems;
};

service.addBoughtItem = function (shopItemIndex) {
    boughtShopItems.push(shopItemsToBuy[shopItemIndex]);
    shopItemsToBuy.splice(shopItemIndex, 1);
};

});
```

Висновок: під час виконання даної лабораторної роботи я навчився застосовувати AngularJS для реалізації сценаріїв обробки та передачі даних між контекстами.