

Теоретичні відомості

Хід роботи:

1) Заходимо на сайт **angular.io**, вибираємо **GetStrarted** і встановлюємо глобально **Angular CLI**

```
npm install -g @angular/cli
```

2) Створюємо на робочому столі папку **Angular**, в цій папці створюємо проект

```
ng new my-app
```

3) В створеній папці проекту **my-app** виконуємо команду **ng serve**, яка робить збірку проекту і запускає сервер на порті **4200**

4) Для перегляду результату набираємо в адресній стрічці браузера **localhost:4200**

5) Для отримання статичного контенту виконаємо команду **ng build --watch**, яка згенерить в нашому проекті папку **dist**.

6) Інсталюємо модулі **express** та **body-parser** і додаємо залежності в файл **package.json**

```
npm i express --save-dev
```

```
npm i body-parser --save-dev
```

7) В корені папки **my-app** створюємо свій сервер - файл **server.js**

```
var express=require('express');
var app=express();

var bodyParser = require('body-parser');
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());
app.use(express.static(__dirname+'/dist/my-app'));
var bodyParser=require('body-parser');
app.use(bodyParser.urlencoded({extended:true}));
app.use(bodyParser.json());

app.get('/',function(req,res){
    res.sendFile(__dirname+'/dist/my-app/index.html');
})

app.listen(process.env.PORT || 8080);
console.log('server is run!');
```

8) Запускаємо сервер в новому терміналі командою **node server**, і набираємо в адресній стрічці браузера **localhost:8080**

9) В папці **src** в файлі **style.css** задаємо глобальні стилі для нашого проекту

```

*{
  margin:0;
  padding: 0;
}
body{
  min-height:100vh;
  box-sizing:border-box;
  display:flex;
}
app-root{
  flex:1;
  display:flex;
  flex-direction:column;
  font-family:Verdana;
  font-size:14px;
}

```

9) В папці **src/app** створюємо новий компонент **header** командою

```
ng g c header
```

10) В файлі **header.component.html** задаємо розмітку компонента

```

<article class="header-phone">User_name</article>
  <section class="header-buttons">
    <article class="header-account">
      <i class="fas fa-user"></i>
      <span class="header-text">My Account</span>
    </article>
    <article class="header-wish">
      <i class="fas fa-heart"></i>
      <span class="header-text">Wish List (0)</span>
    </article>
    <article class="header-login">
      <i class="fas fa-lock"></i>
      <span class="header-text">Login</span>
    </article>
  </section>

```

10) В файлі **header.component.css** задаємо стилі компоненту

```

.header-phone{
  flex:1;
}
.header-buttons{
  display:flex;
}
.header-buttons article{
  border-right:1px white solid;
  padding: 0 10px;
}
.header-buttons article:last-child{
  border:none;
}

```

```

.header-buttons .header-text{
  margin-left:3px;
  display:none;
}
.header-buttons article:hover .header-text{
  display:inline-flex;
}
@media(max-width:640px){
  .header-phone{
    display:none;
  }
}

```

11) Видаляємо вміст файлу **app.component.html** і додаємо створений header-компонент

```
<app-header></app-header>
```

12) В файлі **app.component.css** додаємо селектор **app-header**

```

app-header{
  background-color: black;
  color:white;
  padding:10px 20px;
  display:flex;
}

```

13) По аналогії створюємо компоненти **menu**, **mobile-menu**, **footer**, **products**, **product** (див. практичну частину лабораторної)

14) Створюємо сервіс для роботи з даними командою

```
ng g s data
```

15) В файл **app.module.ts** імпортуємо створений файл **data.service.ts** та стандартний http-модуль

```

import { DataService } from './data.service';
import { HttpClientModule } from '@angular/common/http';

```

В декоратор модуля додаємо **DataService** у властивість **providers**, а **HttpClientModule** у властивість **imports**

```

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    MenuComponent,
    MobileMenuComponent,
    ProductsComponent,
    ProductComponent,
    FooterComponent
  ],
  imports: [
    BrowserModule,

```

```

    BrowserModule,
    HttpClientModule
  ],
  providers: [DataService],
  bootstrap: [AppComponent]
})

```

15) В файл **data.service.ts** реалізуємо логіку роботи з базою даних

```

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';

@Injectable()
export class DataService {

  constructor(private http:HttpClient){

  }

  getProducts(){
    return this.http.get('/getproducts');
  }

  removeProduct(item){
    let obj={id:item._id}
    return this.http.post('/removeproduct',obj,{responseType: 'text'})
  }

  addProduct(item){
    return this.http.post('/addproduct',item,{responseType: 'text'});
  }

}

```

15) В файлі **products.component.ts** інжектуюємо створений сервіс і завантажуюємо продукти в масив **products**

```

import { Component, OnInit } from '@angular/core';
import { DataService } from '../data.service';

@Component({
  selector: 'app-products',
  templateUrl: './products.component.html',
  styleUrls: ['./products.component.css']
})
export class ProductsComponent implements OnInit {
  products;

  constructor(private dataService:DataService) { }

  ngOnInit() {
    this.loadProducts();
  }

}

```

```

loadProducts(){
  this.dataService.getProducts().subscribe(data=>{
    this.products=data;
    console.log(this.products);
  })
}
}

```

16) В файлі **product.component.ts** створюємо з допомогою декоратора **@Input** вхідну властивість **product**, яка прийматиме об'єкт з продуктами і подію **eventProduct** (декоратор **@Output**), яка буде наставати при видаленні продукту

```

import { Component, OnInit, Input, Output, EventEmitter } from '@angular/core';
import { DataService } from '../data.service'
@Component({
  selector: 'app-product',
  templateUrl: './product.component.html',
  styleUrls: ['./product.component.css']
})
export class ProductComponent implements OnInit {
  @Input() product;
  @Output() eventProduct=new EventEmitter();
  constructor(private dataService:DataService) { }

  ngOnInit() {
    console.log(this.product);
  }
  removeProduct(item){
    console.log('removeProduct:');
    console.log(item);
    this.dataService.removeProduct(item).subscribe(data=>{
      console.log(data);
      this.sendEvent();
    });
  }
  sendEvent(){
    this.eventProduct.emit();
  }
}

```

16) В файлі **products.component.html** для кожного елемента масиву **products** формуємо карточку продукту

```

<section class="product-list">
  <app-product *ngFor='let item of products'
    [product]='item'
    (eventProduct)='loadProducts()' >
  </app-product>
</section>

```

17.1) Створити розмітку карточки продукту, реалізувати його видалення з бази даних методом **removeProduct()**

17.2) Реалізувати в компоненті **products** додавання нового продукту в базу даних