План:

Створення grid

Розмітка

Підготовка графіки

Оформлення елементів

Оформлення контенту

Анімація інтерактивних елементів

Оптимізація

Захист перегляду сторінки через пароль що надається лише в резюме (доопрацювання роутингу та модулю вітання)

PageNotFoundComponent (<https://angular.io/guide/router>) path: '\*\*', component: NotFoundComponent

Розгорнуто проект Angular

Підключено Bootstrap, jQuery, popper.js

Інтегровано стилі Bootstrap та додаткові елементи до angular.json:

"styles": [

"src/styles.css",

"./node\_modules/bootstrap/dist/css/bootstrap.min.css"

],

"scripts":[

"./node\_modules/jquery/dist/jquery.min.js",

"./node\_modules/bootstrap/dist/js/bootstrap.min.js"

]

Перевірено коректність роботи

Створено модулі додатку: header, content, footer

Створено складові компоненти модулів

У модулі content створюємо content-routing.module, прописані шляхи переходів до окремих компонент, внесені данні до глобального app-routing.module

Створено модуль Sequre, додано secure-routing.module, та компоненту access-check

В компоненті access-check створено базову розмітку на AngularMaterial (встановлено та підключено усі його складові ) з формою уведення коду

До класу компоненти access-check додано значення паролю з environments, створено реактивну форму, додано ErrorStateMatcher (**працює лише з AngularMaterial** ), додано метод getAccess що здійснює порівняння базового паролю із введеним.

Метод getAccess у випадку коректного ведення паролю зберігає до LocalStorage повідомлення, якщо введено невірний пароль обмежується кількість спроб введення.

В розмітку додано поля що інформують про некоректність введення не валідного паролю, та про обов’язковість заповнення поля, спрацьовують по даним валідатора форми:

*\*ngIf*="passwordForm.get('password').errors?.required"

Для виведення інтерактивних повідомлень підключено **primeng** підключено стилів angular.json, **(**<https://www.primefaces.org/primeng/#/toast>**),** налаштовано функціонал інтерактивний повідомлень.

Створено глобальний сервіс перевірки доступу, після підтвердження доступу видається дозвіл (true/false)

У класі сервісу створено два методи перший для отримання значення з локального сховища через get

get getAccess() {

*return* localStorage.getItem('mlp\_client\_token');

}

Та другий метод що надає данні про доступ

Додаємо захист роутигну: ng g guard guards/auth -> CanActive

Модуль захисту:

*import* { Injectable } *from* '@angular/core';

*import* { ActivatedRouteSnapshot, RouterStateSnapshot, UrlTree, CanActivate, Router } *from* '@angular/router';

*import* { ControAccessService } *from* '../global-services/contro-access.service';

@Injectable({

providedIn: 'root'

})

*export* class AuthGuard implements CanActivate {

constructor(

private access: ControAccessService,

private router: Router

) {}

canActivate(route: ActivatedRouteSnapshot, state: RouterStateSnapshot): boolean {

*if* (*this*.access.keeper()) {

*return* true;

} *else* {

*this*.router.navigate([`/secure`]);

*return* false;

}

}

}

Написання Хедеру:

В модулі хедеру прописуємо експорт компоненти хедеру

*import* { SiteHeaderComponent } *from* './site-header/site-header.component';

@NgModule({

declarations: [SiteHeaderComponent],

imports: [

CommonModule

],

exports: [

SiteHeaderComponent

]

Створюємо базову розмітку хедеру

Додано навігацію з routerLink