Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

ЗВІТ

з дисципліни "Аналіз та рефакторинг коду"

до лабораторної роботи №2

на тему: "**РОЗРОБКА БАЗИ ДАНИХ ДЛЯ СЕРВЕРНОЇ ЧАСТИНИ ПРОГРАМНОЇ СИСТЕМИ ТА ПРИКЛАДНОГО ПРОГРАМНОГО ІНТЕРФЕЙСУ (API)**"

Виконав ст. гр. ПЗПІ-22-2

Терновий Данііл Павлович

Ст. викл. кафедри ПІ

Сокорчук Ігор Петрович

Харків 2024

1. **МЕТА РОБОТИ**

Метою роботи є розробити базу даних для серверної частини та прикладного програмного інтерфейсу для програмної системи оренди малих складів.

1. **ХІД РОБОТИ**

Робота над лабораторним проектом розпочалася зі створення детального проекту системи. Першим етапом стало проектування UML-діаграми прецедентів для системи оренди складів, яка охоплює основні та спеціалізовані функції. Для забезпечення чіткості та деталізації, діаграма була розбита на кілька частин, що наведені у Додатку А.

На другому етапі було створено ER-діаграму, яка демонструє зв’язки між сутностями системи та слугує основою для моделювання даних. Побудована ER-діаграма представлена у Додатку Б, а на її основі розроблена структура бази даних, відображена у вигляді діаграми в Додатку В.

Наступним кроком стала розробка API для забезпечення взаємодії між серверною частиною системи та клієнтами. Для документування API підготовлено Swagger специфікацію у Додатку Г, яка містить детальний опис усіх ендпоінтів, їх параметрів, типів даних та очікуваних відповідей. Це спрощує інтеграцію з клієнтськими додатками.

Приклади коду серверної частини, що демонструють ключові аспекти реалізації системи, наведені у Додатку Д. Повний вихідний код проекту завантажено на платформу GitHub.

Під час розробки API були прийняті наступні інженерні рішення для забезпечення стабільності та гнучкості системи:

Серверна частина побудована на базі ASP.NET з використанням Entity Framework. Кожен компонент системи є незалежним, що включає контролер та сервіс для реалізації конкретних функцій. Це дозволяє спрощено масштабувати систему, імпортувати необхідні моделі для взаємодії з базою даних та експортувати сервіси для використання в інших модулях.

Для роботи з базою даних використано Entity Framework в поєднанні з MSSQL, що забезпечує зручне управління даними, ефективність внесення змін та оптимізацію запитів.

Системою управління базами даних обрано MSSQL, розміщену в хмарному середовищі. Це забезпечує віддалене зберігання даних, підвищуючи продуктивність проекту та зручність його використання.

**3 ВИСНОВКИ**

Під час виконання лабораторної роботи вдалося успішно досягти основної мети — розробити серверну частину та прикладний програмний інтерфейс (API) для системи оренди складів, яка відповідає сучасним стандартам якості та функціональності. Виконання послідовних етапів, від створення UML-діаграми прецедентів до розробки та документування API, забезпечило структурованість і ефективність процесу розробки.

Використання ASP.NET з Entity Framework та модульною архітектурою надало системі високу гнучкість і легкість масштабування. Впровадження Entity Framework сприяло ефективному управлінню даними в MSSQL, обраній як основна система управління базами даних. Розміщення бази даних у хмарі забезпечило зручний віддалений доступ і підвищило продуктивність, що є критично важливим для сучасних програмних рішень.

Загалом, лабораторна робота продемонструвала використання сучасних підходів до проєктування програмних систем, що сприяють створенню масштабованих, надійних і ефективних рішень. Використані технології та інструменти, такі як ASP.NET, Entity Framework, MSSQL, дозволили створити високоякісну систему, готову до реального використання та подальшого розширення.

Публікація коду на платформі GitHub підвищила доступність проєкту та його готовність до подальшого розвитку.

**ДОДАТОК А**

UML – діаграма прецедентів

A diagram of a company

Description automatically generated

Рисунок А.1 – UML – діаграма прецедентів

ДОДАТОК Б

ER – діаграма бази даних

A screenshot of a computer

Description automatically generated

Рисунок Б.1 – ER-діаграма бази даних

ДОДАТОК В

Діаграма структури бази даних

A screenshot of a computer

Description automatically generated

Рисунок В.1 – Діаграма структури бази даних

**ДОДАТОК Г**

Swagger специфікація API

{

"openapi": "3.0.1",

"info": {

"title": "ChefMate",

"version": "v1"

},

"paths": {

"/api/Auth/register": {

"post": {

"tags": [

"Auth"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/RegisterRequest"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/RegisterRequest"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/RegisterRequest"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Auth/login": {

"post": {

"tags": [

"Auth"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/LoginRequest"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/LoginRequest"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/LoginRequest"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Menu": {

"get": {

"tags": [

"Menu"

],

"responses": {

"200": {

"description": "OK"

}

}

},

"post": {

"tags": [

"Menu"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

},

"delete": {

"tags": [

"Menu"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Menu/{id}": {

"get": {

"tags": [

"Menu"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

},

"put": {

"tags": [

"Menu"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/MenuDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

},

"delete": {

"tags": [

"Menu"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/MenuItem": {

"get": {

"tags": [

"MenuItem"

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/MenuItem/{id}": {

"get": {

"tags": [

"MenuItem"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/MenuItem/post": {

"post": {

"tags": [

"MenuItem"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/MenuItem/update": {

"put": {

"tags": [

"MenuItem"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/MenuItem/delete/{id}": {

"delete": {

"tags": [

"MenuItem"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/MenuItem/delete": {

"delete": {

"tags": [

"MenuItem"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/MenuItemDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Order": {

"get": {

"tags": [

"Order"

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Order/{id}": {

"get": {

"tags": [

"Order"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Order/post": {

"post": {

"tags": [

"Order"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Order/update": {

"put": {

"tags": [

"Order"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Order/delete/{id}": {

"delete": {

"tags": [

"Order"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Order/delete": {

"delete": {

"tags": [

"Order"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/OrderDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/OrderItem": {

"get": {

"tags": [

"OrderItem"

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/OrderItem/{id}": {

"get": {

"tags": [

"OrderItem"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/OrderItem/post": {

"post": {

"tags": [

"OrderItem"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/OrderItem/update": {

"put": {

"tags": [

"OrderItem"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/OrderItem/delete/{id}": {

"delete": {

"tags": [

"OrderItem"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/OrderItem/delete": {

"delete": {

"tags": [

"OrderItem"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/OrderItemDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Review": {

"get": {

"tags": [

"Review"

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Review/{id}": {

"get": {

"tags": [

"Review"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Review/post": {

"post": {

"tags": [

"Review"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Review/update": {

"put": {

"tags": [

"Review"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Review/delete/{id}": {

"delete": {

"tags": [

"Review"

],

"parameters": [

{

"name": "id",

"in": "path",

"required": true,

"schema": {

"type": "string",

"format": "uuid"

}

}

],

"responses": {

"200": {

"description": "OK"

}

}

}

},

"/api/Review/delete": {

"delete": {

"tags": [

"Review"

],

"requestBody": {

"content": {

"application/json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

},

"text/json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

},

"application/\*+json": {

"schema": {

"$ref": "#/components/schemas/ReviewDto"

}

}

}

},

"responses": {

"200": {

"description": "OK"

}

}

}

}

},

"components": {

"schemas": {

"KitchenTaskStatus": {

"enum": [

0,

1,

2,

3

],

"type": "integer",

"format": "int32"

},

"LoginRequest": {

"type": "object",

"properties": {

"email": {

"type": "string",

"nullable": true

},

"password": {

"type": "string",

"nullable": true

}

},

"additionalProperties": false

},

"MenuDto": {

"type": "object",

"properties": {

"id": {

"type": "string",

"format": "uuid",

"nullable": true

},

"ownerId": {

"type": "string",

"nullable": true

},

"name": {

"type": "string",

"nullable": true

}

},

"additionalProperties": false

},

"MenuItemDto": {

"type": "object",

"properties": {

"id": {

"type": "string",

"format": "uuid",

"nullable": true

},

"name": {

"type": "string",

"nullable": true

},

"description": {

"type": "string",

"nullable": true

},

"price": {

"type": "number",

"format": "double"

},

"imageUrl": {

"type": "string",

"nullable": true

},

"isAvailable": {

"type": "boolean"

},

"menuId": {

"type": "string",

"format": "uuid"

}

},

"additionalProperties": false

},

"OrderDto": {

"type": "object",

"properties": {

"id": {

"type": "string",

"format": "uuid"

},

"customerId": {

"type": "string",

"nullable": true

},

"waiterId": {

"type": "string",

"nullable": true

},

"chefId": {

"type": "string",

"nullable": true

},

"orderDate": {

"type": "string",

"format": "date-time"

},

"status": {

"$ref": "#/components/schemas/OrderStatus"

},

"totalAmount": {

"type": "number",

"format": "double"

}

},

"additionalProperties": false

},

"OrderItemDto": {

"type": "object",

"properties": {

"id": {

"type": "string",

"format": "uuid"

},

"orderId": {

"type": "string",

"format": "uuid"

},

"menuItemId": {

"type": "string",

"format": "uuid"

},

"priority": {

"type": "integer",

"format": "int32"

},

"status": {

"$ref": "#/components/schemas/KitchenTaskStatus"

},

"quantity": {

"type": "integer",

"format": "int32"

},

"price": {

"type": "number",

"format": "double"

}

},

"additionalProperties": false

},

"OrderStatus": {

"type": "integer",

"format": "int32"

},

"RegisterRequest": {

"type": "object",

"properties": {

"email": {

"type": "string",

"nullable": true

},

"password": {

"type": "string",

"nullable": true

},

"role": {

"type": "string",

"nullable": true

}

},

"additionalProperties": false

},

"ReviewDto": {

"type": "object",

"properties": {

"id": {

"type": "string",

"format": "uuid"

},

"customerId": {

"type": "string",

"nullable": true

},

"orderId": {

"type": "string",

"format": "uuid"

},

"feedback": {

"type": "string",

"nullable": true

},

"rating": {

"type": "integer",

"format": "int32"

},

"date": {

"type": "string",

"format": "date-time"

}

},

"additionalProperties": false

}

},

"securitySchemes": {

"Bearer": {

"type": "http",

"description": "JWT Authorization header using the Bearer scheme. Example: \"Bearer {token}\"",

"scheme": "bearer",

"bearerFormat": "JWT"

}

}

},

"security": [

{

"Bearer": [ ]

}

]

}

**ДОДАТОК Д**

Приклад програмного коду

using AutoMapper;

using ChefMate\_backend;

using ChefMate\_backend.Models;

using ChefMate\_backend.Repositories;

using ChefMate\_backend.Services;

using Microsoft.AspNetCore.Authentication.JwtBearer;

using Microsoft.AspNetCore.Identity;

using Microsoft.EntityFrameworkCore;

using Microsoft.IdentityModel.Tokens;

using Microsoft.OpenApi.Models;

using System.Text;

var builder = WebApplication.CreateBuilder(args);

//Connect database context to MSSQL

builder.Services.AddDbContext<ApplicationDbContext>(options =>

options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")));

//Registrate IMapper

IMapper mapper = MappingConfig.RegisterMaps().CreateMapper();

builder.Services.AddSingleton(mapper);

builder.Services.AddAutoMapper(AppDomain.CurrentDomain.GetAssemblies());

//Registrate JwtService and JwtSettings

builder.Services.Configure<JwtSettings>(builder.Configuration.GetSection("JwtSettings"));

builder.Services.AddScoped<JwtTokenService>();

//Add identity requerements

builder.Services.AddIdentity<ChefMateUser, ChefMateRole>(options =>

{

options.Password.RequireDigit = true;

options.Password.RequiredLength = 6;

options.Password.RequireNonAlphanumeric = false;

options.Password.RequireUppercase = false;

options.Password.RequireLowercase = true;

options.User.RequireUniqueEmail = true;

})

.AddEntityFrameworkStores<ApplicationDbContext>()

.AddDefaultTokenProviders();

//Add auth and Bearer

builder.Services.AddAuthentication(options =>

{

options.DefaultAuthenticateScheme = JwtBearerDefaults.AuthenticationScheme;

options.DefaultChallengeScheme = JwtBearerDefaults.AuthenticationScheme;

}).AddJwtBearer(options =>

{

var jwtSettings = builder.Configuration.GetSection("JwtSettings").Get<JwtSettings>();

options.TokenValidationParameters = new TokenValidationParameters

{

ValidateIssuer = true,

ValidateAudience = true,

ValidateLifetime = true,

ValidateIssuerSigningKey = true,

ValidIssuer = jwtSettings.Issuer,

ValidAudience = jwtSettings.Audience,

IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(jwtSettings.Secret))

};

});

builder.Services.AddControllersWithViews();

builder.Services.ConfigureApplicationCookie(options =>

{

options.ExpireTimeSpan = TimeSpan.FromDays(14);

options.SlidingExpiration = true;

});

//Registrate services in DI container

builder.Services.AddScoped<IOrderRepository, OrderRepository>();

builder.Services.AddScoped<IOrderItemRepository, OrderItemRepository>();

builder.Services.AddScoped<IReviewRepository, ReviewRepository>();

builder.Services.AddScoped<IMenuItemRepository, MenuItemRepository>();

builder.Services.AddScoped<IMenuRepository, MenuRepository>();

builder.Services.AddControllers()

.AddJsonOptions(options =>

{

options.JsonSerializerOptions.ReferenceHandler = System.Text.Json.Serialization.ReferenceHandler.IgnoreCycles;

});

//Add swagger

builder.Services.AddSwaggerGen(options =>

{

options.SwaggerDoc("v1", new OpenApiInfo { Title = "ChefMate", Version = "v1" });

options.AddSecurityDefinition("Bearer", new OpenApiSecurityScheme

{

Name = "Authorization",

Type = SecuritySchemeType.Http,

Scheme = "bearer",

BearerFormat = "JWT",

In = ParameterLocation.Header,

Description = "JWT Authorization header using the Bearer scheme. Example: \"Bearer {token}\""

});

//Add security scheme to Bearer

options.AddSecurityRequirement(new OpenApiSecurityRequirement

{

{

new OpenApiSecurityScheme

{

Reference = new OpenApiReference

{

Type = ReferenceType.SecurityScheme,

Id = "Bearer"

}

},

new string[] {}

}

});

});

var app = builder.Build();

// Code for automigration and roles creating

using (var scope = app.Services.CreateScope())

{

var services = scope.ServiceProvider;

try

{

var context = services.GetRequiredService<ApplicationDbContext>();

context.Database.Migrate();

}

catch (Exception ex)

{

var logger = services.GetRequiredService<ILogger<Program>>();

logger.LogError(ex, "Error occurred while migrating the database");

}

try

{

var roleManager = services.GetRequiredService<RoleManager<ChefMateRole>>();

var roles = new List<ChefMateRole>

{

new ChefMateRole { Name = "Admin" },

new ChefMateRole { Name = "Customer" },

new ChefMateRole { Name = "Chef" },

new ChefMateRole { Name = "Waiter" }

};

foreach (var role in roles)

{

if (!roleManager.RoleExistsAsync(role.Name).Result)

{

roleManager.CreateAsync(role).Wait();

}

}

}

catch (Exception ex)

{

var logger = services.GetRequiredService<ILogger<Program>>();

logger.LogError(ex, "An error occurred creating the DB.");

}

}

if (app.Environment.IsDevelopment())

{

app.UseDeveloperExceptionPage();

app.UseSwagger();

//Configurate endpoint for swagger in production

app.UseSwaggerUI(c =>

{

c.SwaggerEndpoint("/swagger/v1/swagger.json", "ChefMate V1");

c.RoutePrefix = "swagger";

});

}

else

{

app.UseExceptionHandler("/Home/Error");

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

app.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

**ДОДАТОК Е**

Відео-посилання на Youtube

https://youtu.be/AWRQZ0rmSO8