**Звіт по коду проекту "SmartBiz"**

**Підготували: Дідух Кріпак Мусурівський Думайло Орищак**

**1. Структура проекту**

Проект "SmartBiz" складається з кількох основних частин коду, розділених за функціональністю. Основні компоненти включають:

* Startup.cs
* ApplicationDbContext.cs
* Моделі (Models)
* Контролери (Controllers)
* SQL-скрипти для створення таблиць і заповнення даними

**2. Основні частини коду**

**2.1. Startup.cs**

public class Startup

{

public void ConfigureServices(IServiceCollection services)

{

services.AddDbContext<ApplicationDbContext>(options =>

options.UseNpgsql("Host=localhost;Database=SmartBizDB;Username=postgres;Password=your\_password"));

// Інші сервіси

}

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

// Налаштування middleware

}

}

**Опис**: Цей клас відповідає за конфігурацію сервісів та middleware в додатку. У методі ConfigureServices налаштовується підключення до бази даних за допомогою Entity Framework.

**2.2. ApplicationDbContext.cs**

csharp

using Microsoft.EntityFrameworkCore;

namespace SmartBiz.Infrastructure

{

public class ApplicationDbContext : DbContext

{

public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options)

: base(options)

{

}

public DbSet<Order> Orders { get; set; }

public DbSet<Product> Products { get; set; }

public DbSet<Customer> Customers { get; set; }

public DbSet<FinancialRecord> FinancialRecords { get; set; }

}

}

**Опис**: ApplicationDbContext відповідає за доступ до бази даних. Він містить DbSet для кожної з основних таблиць.

**2.3. Моделі**

**Модель клієнта (Customer)**:

csharp

public class Customer

{

public int Id { get; set; }

public string FirstName { get; set; }

public string LastName { get; set; }

public string Email { get; set; }

public string PhoneNumber { get; set; }

}

**Модель продукту (Product)**:

csharp

Copy

public class Product

{

public int Id { get; set; }

public string Name { get; set; }

public decimal Price { get; set; }

public int Stock { get; set; }

}

**Модель замовлення (Order)**:

csharp

public class Order

{

public int Id { get; set; }

public int CustomerId { get; set; }

public DateTime OrderDate { get; set; }

public decimal TotalAmount { get; set; }

public string Status { get; set; }

}

**Опис**: Моделі представляють структуру даних, що зберігаються в базі даних. Кожна модель відповідає своїй таблиці.

**2.4. Контролери**

**Контролер для управління замовленнями**:

csharp

[ApiController]

[Route("api/[controller]")]

public class OrdersController : ControllerBase

{

private readonly ApplicationDbContext \_context;

public OrdersController(ApplicationDbContext context)

{

\_context = context;

}

[HttpGet]

public ActionResult<IEnumerable<Order>> GetOrders()

{

return \_context.Orders.ToList();

}

}

**Опис**: Контролер відповідає за обробку HTTP-запитів, пов'язаних із замовленнями. У ньому реалізовані методи для отримання даних з бази.

**3. SQL-скрипти**

**3.1. Створення таблиць**

sql

CREATE TABLE Customers (

Id SERIAL PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Email VARCHAR(100) UNIQUE NOT NULL,

PhoneNumber VARCHAR(15)

);

CREATE TABLE Products (

Id SERIAL PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

Stock INT NOT NULL

);

CREATE TABLE Orders (

Id SERIAL PRIMARY KEY,

CustomerId INT REFERENCES Customers(Id),

OrderDate TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

TotalAmount DECIMAL(10, 2) NOT NULL,

Status VARCHAR(20) NOT NULL

);

**3.2. Заповнення таблиць**

sql

INSERT INTO Customers (FirstName, LastName, Email, PhoneNumber) VALUES

('John', 'Doe', 'john.doe@example.com', '123-456-7890'),

('Jane', 'Smith', 'jane.smith@example.com', '098-765-4321');

INSERT INTO Products (Name, Price, Stock) VALUES

('Product A', 19.99, 100),

('Product B', 29.99, 200),

('Product C', 39.99, 150);

INSERT INTO Orders (CustomerId, TotalAmount, Status) VALUES

(1, 59.97, 'Completed'),

(2, 29.99, 'Pending');

**4. Висновок**

**Опис проекту**

Проект "SmartBiz" є веб-додатком, розробленим на основі ASP.NET Core з використанням Entity Framework Core для роботи з базою даних PostgreSQL. Основна мета проекту — створити платформу для управління даними про клієнтів, продукти та замовлення, що дозволяє підприємствам ефективно організовувати свою діяльність.

**Основні функціональні можливості**

* **Управління клієнтами**: Додаток дозволяє зберігати та редагувати інформацію про клієнтів, включаючи ім'я, прізвище, електронну пошту та номер телефону.
* **Управління продуктами**: Можливість додавання, редагування та видалення продуктів, а також управління їхньою ціною та кількістю на складі.
* **Обробка замовлень**: Додаток дозволяє створювати замовлення, пов'язані з клієнтами, фіксуючи дату замовлення, загальну суму та статус.

**Зроблена робота**

**5. Архітектура проекту**

Проект реалізовано за допомогою архітектури ASP.NET Core, що включає в себе три основні компоненти: frontend, backend і базу даних.

**6. Створення бази даних**

Розроблено структуру бази даних з трьома основними таблицями: Customers, Products і Orders. Використано SQL-скрипти для створення таблиць і заповнення їх початковими даними.

**7. Реалізація DbContext**

Створено клас ApplicationDbContext, який відповідає за доступ до бази даних і містить DbSet для кожної таблиці. Це дозволяє виконувати CRUD-операції з даними.

**8. Налаштування сервісів**

У класі Startup налаштовано підключення до бази даних за допомогою Entity Framework. Визначено використовувані сервіси та middleware.

**9. Опис моделей**

Для кожної таблиці створено відповідні моделі: Customer, Product та Order. Ці моделі представляють структуру даних, що зберігаються в базі даних.

**10. Реалізація контролерів**

Створено контролери для обробки HTTP-запитів. Наприклад, контролер OrdersController відповідає за отримання інформації про замовлення.

**11. Тестування**

Проведено тестування функціональності додатку, включаючи перевірку можливостей додавання, редагування та видалення даних.

.