**ЗВІТ ВИКОНАННЯ 5 ЗАВДАННЯ**

**Виконали Орищак, Дідух, Мусурівськи,Думайло,Кріпак**

**Реалізація use-case в рамках MVC**

Було реалізовано основні функціональні модулі відповідно до архітектурного шаблону Model-View-Controller.

Контролери (Controllers) обробляють вхідні HTTP-запити, викликають сервіси та повертають відповідні представлення або відповіді API.

Моделі (Models) містять сутності та сервіси для виконання бізнес-логіки, зокрема Order, Product, Client, Transaction, User.

Представлення (Views) реалізовано за допомогою Razor View Engine. Створено окремі сторінки для кожного модуля (Продажі, Інвентар, Фінанси, Клієнти, Користувачі) з урахуванням ролей користувачів.

**Написання контролерів**

Реалізовано наступні контролери:

* OrdersController
* InventoryController
* FinanceController
* ClientsController
* UsersController

Контролери обробляють запити користувача на створення, редагування, перегляд, зміну статусу та видалення даних відповідно до кожного модуля

**Написання класів сутностей та сервісів**

Створено класи сутностей, що відображають основні елементи предметної області: замовлення, товари, клієнти, транзакції, користувачі. Реалізовано сервіси для кожного з модулів: OrderService, InventoryService, FinanceService, ClientService, UserService. Бізнес-логіка розміщена у відповідних сервісах відповідно до принципу єдиної відповідальності (Single Responsibility Principle).

**Написання класів (репозиторіїв) для роботи з базою даних**

Для роботи з базою даних використано шаблон репозиторію. Створено наступні класи:

* OrderRepository
* ProductRepository
* ClientRepository
* TransactionRepository
* UserRepository

Кожен репозиторій реалізує відповідний інтерфейс для забезпечення інкапсуляції доступу до даних. Робота з базою даних виконується через Entity Framework Core.

**Створення шаблонів відображень (Views)**

Реалізовано Razor-шаблони для представлення даних користувачу. Основні шаблони:

* Продажі: таблиця замовлень, сторінка деталей замовлення, створення нового замовлення
* Інвентар: таблиця товарів, форма редагування, додавання товару
* Фінанси: зведені показники, генерація фінансових звітів
* Клієнти: перегляд клієнтів, редагування даних, програма лояльності
* Користувачі: додавання нових користувачів, призначення ролей

Відображення адаптовано до ролей користувачів згідно з функціональними правами доступу.

**Написання юніт-тестів з використанням xUnit**

Тестування реалізовано за допомогою фреймворку xUnit.

* Покрито основні методи, що реалізують бізнес-логіку
* Враховано позитивні та негативні сценарії виконання

**Висновок**

Проєкт реалізовано згідно з усіма вимогами до архітектури, бізнес-логіки, тестування та відповідності стандартам написання коду. Усі функціональні частини розділено відповідно до шаблону MVC. Забезпечено надійність, підтримуваність і тестованість розробленого програмного забезпечення.