

أفضل مشروع نعمله مع بعض

مشروع إدارة النقل والشحن (Transportation Management System - TMS)

لأنه فيه Backend المشروع نفسه مناسب جداً لتعلم

- Users + Roles
 - Vehicles
 - Drivers
- Trips (الرحلات)
- Loads / Shipments
 - Tracking
 - Reports
- Dashboard





Transport Management System

TM Service

Dashboard Overview
10th April, 2021

Shipment Metrics ● Complete ● Pending dispatch Last 30 days

Active Shipments

48 Complete CRS/Total: 0.00 % Trending: N/A

14 On Time Delivery OTR/Total: 0.00 % Trending: N/A

80 Pending Dispatch CRS/Total: 0.00 % Trending: N/A

17 On Time Pickup OTP/Total: 0.00 % Trending: N/A

Shipments By Countries

Country	Percentage
USA	64%
Italy	18.20%
Ukraine	9.13%
Russia	11.26%
China	5.20%
Other	0.20%

Transaction History

Purchaser	Order	Date	Amount
Tesco Market	#02942	13 Apr 2021	\$1075.67
ElectroMen Market	#02231	14 Apr 2021	\$1250.00
Fiorgio Restaurant	#07241	07 Apr 2021	\$319.50
John Mathew Kayne	#04232	06 Apr 2021	\$1350
Ann Morlin	#02461	31 Apr 2021	\$7430
Ember Cafe	#02948	31 Apr 2021	\$2430.20

Create New Shipment +

Need help?

Shipping Status Dashboard

Klipfolio



4

أجزاء المشروع اللي هنتعلمها ⚙️

1) تصميم الـ Domain

1. Vehicle
2. Driver
3. Trip
4. Shipment
5. FuelRecord
6. Maintenance

2) إنشاء API بـ .NET 8

- Minimal APIs أو Controllers (اختيارك)
- DTOs

- Mapping
- Validation

3) Database + EF Core

- Migrations
- Seeding Data
- Relations (One-to-Many / Many-to-Many)

4) Authentication

تسجيل دخول
JWT Token
Authorization (Roles + Policies)

5) CQRS

Commands
Queries
MediatR

6) Services

Trip assignment
Tracking updates

7) Dashboard API

Daily loads count
Driver performance
Vehicle status

فكرة المشروع

Transport & Finance Management System

يشمل:

- إداره:
 - Companies / Customers
 - Vehicles
 - Drivers
 - Trips (رحلات)
 - Shipments (الشحنات)
 - الجزء المالي
 - Invoices (فواتير)
 - Payments (مدفعات)
 - Expenses (... مصروفات تشغيل: وقود، صيانة)
 - حساب الربح / الخسارة لكل رحلة أو فترة

brick icon المعمارية اللي هنشتغل بيها (Clean Architecture)

فيها 4 طبقات رئيسية Solution هنعمل

- 7. Domain
 - a. Entities
 - b. Value Objects
 - c. Enums
 - d. Domain Events (لاحقاً)
- 8. Application
 - a. DTOs / ViewModels
 - b. Interfaces (Repositories, Services)
 - c. Use Cases (Commands / Queries – CQRS)
 - d. Business Rules
- 9. Infrastructure

- a. EF Core DbContext
- b. Repositories (Repository + Unit Of Work)
- c. Implementations للتعامل مع SQL Server

10. API (Presentation)

- a. Controllers
- b. Endpoints
- c. Dependency Injection

اللى هستخدمهم Design Patterns

: خلال الشغل هنطبق

- Repository Pattern
- Unit of Work
 - CQRS (Commands / Queries)
 - Specification Pattern (فلترة البيانات)
- Strategy Pattern (مثلاً لحساب تكلفة الرحلة أو الضريبة بأكثر من طريقة)
 - Factory Pattern (إنشاء أنواع مختلفة من الدفعات)
- Domain Events (عشان نصدر فاتورة Event مثلاً لマरحلة تخلص نطلق)

المرحلة القادمة هي اللي هترافقك عن Junior:

- CQRS
- MediatR
- Validation
- Authentication
- Caching
- Logging
- SOLID
- Domain Events
- Specifications Pattern
- DDD Concepts

فلاً بذن الله Senior وهمشى فيهم واحدة واحدة لحد ما توصل 🔥🔥

JWT TOKEN وعاوز اضيف عليهم تص

(خريطة المشروع (من البداية للنهاية

تم إنجازه ✓:

- **Domain Layer**
 - Entities (Vehicle, Driver, Trip, Invoice)
 - قواعد الدومين + العلاقات
- **Infrastructure Layer**
 - TransportDbContext
 - Repositories
 - Unit of Work
 - مثال عملي للتجربة

➡ مهم جدًا) المطلوب إضافته قبل **soon** (CQRS

أعلى Layer هذه الأشياء لازم ننهيها قبل ما نطلع لأي



■ ① **Repositories** توحيد الـ **Aggregate**

- IVehicleRepository
- IDriverRepository
- ITripRepository
- IIInvoiceRepository

مختلفة لكل واحد Queries لأنك تحتاج

■ ② **Configuration** إعداد **Fluent API** كامل للعلاقات بـ

- One-to-One: Vehicle ↔ Driver ✓ بدأناه
- One-to-Many: Vehicle → Trips ✗ نكمله

- في الجداول Fields خصائص الـ:
 - طول String
 - Required
 - Default values

:مثال

```
builder.Entity<Vehicle>(b =>
{
    b.Property(v => v.PlateNumber)
        .IsRequired()
        .HasMaxLength(50);
});
```

3 Seed Data (اختياري)

تضيف بيانات اختبار للمركبات والسائلين
عشن نختبر APIs

4 للعمليات الكبيرة UoW داخل Transactions تفعيل الـ

لبو عايز نعمل:

- BeginTransaction
- Commit
- Rollback

5 DI تسجيل الـ Repositories في

في Program.cs:

```
services.AddScoped<IVehicleRepository, VehicleRepository>();
services.AddScoped<IDriverRepository, DriverRepository>();
```

```
services.AddScoped<ITripRepository, TripRepository>();  
services.AddScoped<IInvoiceRepository, InvoiceRepository>();
```

6 تنظيم Namespaces والملفات