

# Transport & Management System

## Project Location

Head Office: Muğla, Turkey

---

## Project Overview

This project is a **comprehensive web-based management system** for a transportation company that ships goods from **Muğla** to destinations worldwide.

It allows customers to request shipments, automatically calculates prices, optimizes container usage, manages ships and trucks, tracks financial performance, monitors inventory, and generates full business reports.

---

## Project Objectives

1. Streamline transportation operations from order creation to delivery.
  2. Automate cost, time, and distance calculations.
  3. Optimize container capacity to reduce empty space and increase profit.
  4. Manage the company's ships and trucks, including travel expenses.
  5. Track financial performance (revenue, expenses, tax, profit).
  6. Monitor and control product inventory (e.g., blueberries by category).
  7. Provide detailed analytics and comprehensive company reports.
- 

## System Users

User Type	Description
Customer	Creates shipment requests, views price estimates, tracks shipments
Admin (Company User)	Manages shipments, containers, vehicles, finances, inventory, and reports

---

# System Modules Overview

## 1. Customer Interface

- **Home Page** – company introduction, “Create Shipment” button.
- **Product Upload Page** – customer enters product details:
  - Product name, category, weight (kg)
  - Container type (Small, Medium, Large)
  - Destination (city, country)
  - Upload product image or details
- **Price Calculation**
  - Base location: Muğla
  - Price depends on distance and container type.
  - Formula:  
$$\text{Total Price} = \text{Distance} \times \text{Rate per km} \text{ (based on container type)}$$
- **Example Rates**

Container Type	Capacity (kg)	Price (₺/km)
Small	2,000	₺5
Medium	5,000	₺8
Large	10,000	₺12
- **Result Page:** Shows calculated price, estimated delivery time, and order summary.
- **Tracking Page:** Customers can check shipment status using an order ID.
- **Capacity:** If capacity is full, there should be alert(ex: There is no enough space at small container)

---

## 2. Container Optimization System

The company has limited containers. When multiple customers request smaller shipments, the system groups them into the same container according to total capacity.

### Algorithm Concept

- Sort shipments by weight (largest first).
- Place each shipment into the first container that has enough remaining capacity.
- When a container is full, it is marked as “Ready for Transport.”

### Example:

You have a **Large Container (10,000 kg)** and customer orders:

- A: 2,000 kg
  - B: 500 kg
  - C: 200 kg
- These three orders (2,700 kg total) are packed together; space left = 7,300 kg.  
The system continues filling until optimal utilization is achieved.
- 

### 3. Fleet Management

The company owns:

- **3 Ships** for international deliveries
- **4 Trucks** for domestic deliveries and port transfers

Each vehicle has defined **capacity** and **operating costs**.

Type	Name	Capacity (kg)	Fuel Cost/km	Crew/Driver Cost	Maintenance	Total Expense
Ship	BlueSea	100,000	₦40	₦20,000	₦10,000	₦70,000
Ship	OceanStar	120,000	₦50	₦25,000	₦12,000	₦87,000
Ship	AegeanWind	90,000	₦35	₦18,000	₦8,000	₦61,000
Truck	RoadKing	10,000	₦8	₦3,000	₦2,000	₦8,000
Truck	FastMove	12,000	₦9	₦3,500	₦2,500	₦9,000
Truck	CargoPro	9,000	₦7	₦2,800	₦2,000	₦7,800
Truck	HeavyLoad	15,000	₦10	₦4,000	₦3,000	₦10,500

#### Vehicle Expense Formula

Trip Expense = (Fuel Cost/km × Distance) + Crew/Driver + Maintenance

#### Example:

Ship “BlueSea” travels 2,000 km

$$\rightarrow \text{Expense} = (\text{₦}40 \times 2,000) + \text{₦}20,000 + \text{₦}10,000 = \text{₦}110,000$$


---

### 4. Financial Calculations

Metric	Formula
Revenue	Total of all completed shipment payments
Expenses	Sum of all fleet costs + maintenance + other costs
Net Income	Revenue – Expenses
Tax	20 % × Net Income
Profit After Tax	Net Income – Tax

#### Example:

Item	Value
Revenue	₦1,200,000
Total Fleet Expense	₦253,300

<b>Item</b>	<b>Value</b>
Other Expenses	฿80,000
<b>Net Income</b>	<b>฿866,700</b>
Tax (20%)	฿173,340
<b>Profit After Tax</b>	<b>฿693,360</b>

---

## 5. Inventory Management

The company stores **blueberries** in multiple categories:

<b>Category</b>	<b>Quantity (kg)</b>	<b>Minimum Stock</b>	<b>Status</b>
Fresh	4,500	2,000	OK
Frozen	1,200	1,000	Low
Organic	8,000	2,500	OK

When stock falls below the minimum, the system alerts:

“Frozen blueberries stock running low — please restock.”

Inventory is automatically updated when shipments are created or fulfilled.

---

## 6. Comprehensive Reporting

The reporting module integrates **financial**, **fleet**, **inventory**, and **shipment** data.

### Reports include:

- Total revenue, expenses, and tax
- Number of shipments
- Container utilization rate
- Most popular routes
- Total distance traveled
- Products sold per category
- Remaining inventory

### Example Report Output

<b>Metric</b>	<b>Value</b>
Total Revenue	฿1,200,000
Total Fleet Expense	฿253,300
Other Expenses	฿80,000
Net Income	฿866,700
Tax (20%)	฿173,340
Profit After Tax	฿693,360
Total Shipments	250
Average Container Utilization	88 %

Metric	Value
Most Popular Route	Muğla → Berlin
Total Distance Covered	340,000 km

---

## Data Information Structure (Simplified)

### Shipments Table

Field	Example
id	1
customer_name	“Ali Yılmaz”
weight	500
destination	“Berlin”
container_id	1
status	“Pending”

### Containers Table

Field	Example
id	1
type	“Large”
capacity	10,000
current_load	7,850
status	“Ready”

### Fleet Table

Field	Example
id	“S001”
type	“Ship”
name	“BlueSea”
capacity	100,000
fuel_cost_per_km	40
crew_cost	20,000
maintenance	10,000
total_expense	70,000

### Inventory Table

Field	Example
category	“Frozen Blueberries”
quantity	1,200
min_stock	1,000
status	“Low”

### Financial Table

Field	Example
total_revenue	1,200,000
total_expenses	333,300

Field	Example
net_income	866,700
tax	173,340
profit_after_tax	693,360

---

## Algorithms (Core Logic)

1. **Distance Calculation:** via Google Maps API (Muğla → Destination).

2. **Price Calculation:**

```
price = base_rate × distance × container_type_factor
```

3. **Container Optimization:**

Bin Packing Algorithm (First-Fit Decreasing).

4. **Fleet Expense:**

```
expense = (fuel_per_km × distance) + crew/driver + maintenance
```

5. **Financial Summary:**

```
revenue_sum, expenses_sum, tax = 0.2 × net_income
```

6. **Inventory Update:**

Subtract shipment quantity from stock after loading.

---

## Admin Dashboard Features

- View all pending and completed shipments
- Run “Optimize Containers”
- Manage ships and trucks, update status
- View financial and tax summaries
- Manage blueberry inventory and receive alerts
- Generate and export PDF reports

### Dashboard Tabs

1. Shipments
2. Container Optimization
3. Fleet Management
4. Financials
5. Inventory
6. Reports

---

## Summary

The **Global Freight Transport & Management System** provides a realistic simulation of how a modern logistics company operates — integrating customers, containers, vehicles, inventory, finances, and analytics into one intelligent platform.

It showcases:

- Business logic (pricing, logistics, optimization)
- Real-world data (fleet costs, inventory tracking)
- Technical depth (databases, algorithms, web interface)

This project demonstrates both **technical skills** and **business understanding**, making it ideal as a professional-level homework or portfolio project.

---

Use case:

## Use Case Scenario: “Shipping Blueberries from Muğla to Berlin”

### Actors

1. **Customer:** Ali Yılmaz
  2. **Admin:** Logistics Manager at the company
- 

### Step 1: Customer Creates Shipment

**Goal:** Ali wants to send 500 kg of fresh blueberries to Berlin.

#### Steps:

1. Ali logs into the **Customer Portal**.
2. Navigates to “**Create Shipment**” page.
3. Enters product details:
  - Product: Fresh Blueberries
  - Category: Fresh
  - Weight: 500 kg
  - Container Type: Small
  - Destination: Berlin, Germany
  - Uploads product image
4. The system calculates:
  - Distance from Muğla → Berlin (via Google Maps API): 3,000 km
  - Price = Distance × Rate per km for Small Container =  $3,000 \times \$5 = \$15,000$
5. Ali reviews the **order summary and estimated delivery time**, then confirms shipment.
6. System updates **inventory**: subtracts 500 kg from Fresh Blueberries stock.

**Outcome:** Shipment is created with status “**Pending**”.

---

### Step 2: Admin Optimizes Containers

**Goal:** Make efficient use of containers for multiple shipments.

#### Steps:

1. Admin logs into **Admin Dashboard** → **Container Optimization**.

2. System shows all pending shipments:

- Shipment A: 2,000 kg
- Shipment B: 500 kg (Ali's order)
- Shipment C: 200 kg

3. Admin clicks “**Optimize Containers**”.

4. System runs **First-Fit Decreasing Algorithm**:

- Large container (10,000 kg) packed with A + B + C = 2,700 kg
- Remaining space = 7,300 kg

5. Container status updated to “**Ready for Transport**”.

**Outcome:** Shipments grouped efficiently, no wasted space.

---

### **Step 3: Fleet Management**

**Goal:** Assign vehicles for shipment delivery.

**Steps:**

1. Admin selects **Fleet Management** → **Ships/Trucks**.

2. For Ali's shipment (domestic truck → port → ship → international):

- Truck “RoadKing” carries 500 kg from Muğla to port.
- Ship “BlueSea” carries 2,700 kg to Berlin.

3. System calculates **fleet expense**:

- Truck:  $(\$8 \times 100 \text{ km}) + \$3,000 + \$2,000 = \$5,800$
- Ship:  $(\$40 \times 3,000 \text{ km}) + \$20,000 + \$10,000 = \$150,000$

**Outcome:** Fleet assigned efficiently; trip costs tracked.

---

### **Step 4: Financial Summary**

**Goal:** Update company's revenue, expenses, and profit.

**Steps:**

1. Admin navigates to **Financial Dashboard**.

2. System calculates:

- Revenue =  $\$15,000$  (Ali's shipment) + other shipments
- Expenses = Truck + Ship + maintenance
- Net Income = Revenue – Expenses

- Tax = 20% of Net Income
- Profit After Tax = Net Income – Tax

**Outcome:** Company sees updated financial metrics, including Ali's shipment contribution.

---

## Step 5: Customer Tracks Shipment

**Goal:** Ali wants to check shipment status.

**Steps:**

1. Ali goes to **Customer Portal** → **Tracking**.
2. Enters **Order ID**.
3. System shows:
  - Container: Large-1
  - Status: Ready for Transport → In Transit → Delivered
  - Estimated delivery time

**Outcome:** Customer stays informed; transparency increases satisfaction.

---

## Step 6: Inventory Monitoring

**Goal:** Ensure stock levels are adequate.

**Steps:**

1. Admin checks **Inventory Dashboard**.
2. System shows:
  - Fresh Blueberries: 4,000 kg      OK
  - Frozen Blueberries: 1,200 kg      Low
  - Organic Blueberries: 8,000 kg      OK
3. Low-stock alerts prompt restocking orders.

**Outcome:** Inventory is always monitored; shortages avoided.

---

**Result:**

- Shipment successfully created, optimized, and delivered.
  - Container space maximized, fleet usage tracked, finances updated, inventory monitored.
  - Both **customer** and **admin** have a smooth experience.
-

chatGPT ALLOWED :)

