

Logistics Management Application

Project overview:

The **Logistics Management Application** is a web-based platform designed to streamline logistics operations, including shipment tracking, order dispatch, and delivery status monitoring. The system aims to improve supply chain efficiency by enabling easy updates and effective order management.

Objectives:

- Provide an efficient user interface for Admin, Manager, and Viewer roles to track shipment statuses.
- Offer an intuitive and user-friendly interface for logistics personnel.
- Enable real-time shipment tracking.
- Facilitate effective order management.

Technology Stack:

Component	Technology
Frontend Framework	React.js
Routing	React Router DOM
UI Library	CSS
Backend (API)	Dotnet
Database	MSSQL
Authentication	JWT (JSON Web Token)

Key Features:

1. User Authentication with Role-Based Access
 - Secure login using JWT authentication.
 - Users are authenticated based on assigned **roles**:
 - ◆ **Admin**: Full access to all modules (Create, Edit and Delete the shipment orders, override the manager and viewer shipment status).

- ◆ **Manager:** Access to shipments and orders; allowed to modify shipment statuses.
 - ◆ **Viewer:** Read-only access to dashboards and edit shipment status if the status is delayed.
 - Dynamic navigation menus and dashboard content based on user roles.
 - Route protection using Private Routes and Role Guards.
 - Password encryption and account security measures.
2. Dashboard:
- Add new shipments with details.
 - View shipment status (e.g., Pending, In Transit, Delivered).
 - Update delivery status via API integration.
 - Cancel shipment delivery via API integration.

Future Enhancement:

- Interactive charts for shipment trends, delivery timelines, etc.
- Report generation based on delivery time, delays, shipment costs, etc.
- Inventory tracking for warehouse stock with low-inventory alerts.

Steps To Run Logistics Management Application

Project Structure

```

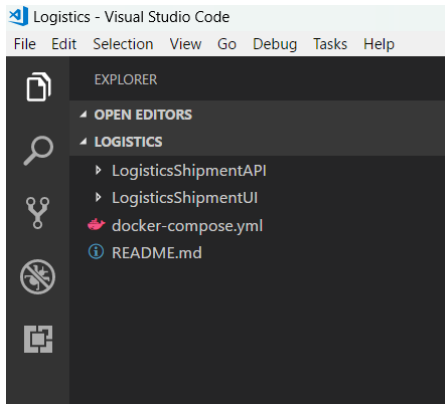
Logistics/
|
├─ LogisticsShipmentUI/    (React + JWT)
|
└─ LogisticsShipmentAPI/   (.NET Core API + MSSQL)

```

Steps to add the project into Visual studio code and run the application

- Open **Visual studio code**.
- Go to File → Open Folder...
- Select the root folder:
Logistics (this will load both LogisticsShipmentUI and LogisticsShipmentAPI).

- You can now see both folders in the VS Code Explorer as below:



- Install the frontend dependencies
 - ❖ Open terminal in VS Code: **cd LogisticsShipmentUI**
 - ❖ Install npm packages: **npm install**

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

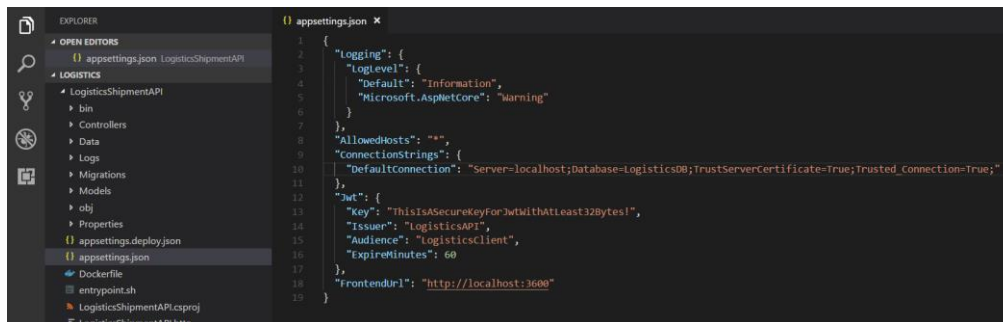
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\A525011\OneDrive - Volvo Group\Logistics> cd .\LogisticsShipmentUI\
PS C:\Users\A525011\OneDrive - Volvo Group\Logistics\LogisticsShipmentUI> npm install

```

- Install the backend dependences
 - ❖ Make sure to have a MSSQL server (localhost), Dotnet has been installed.
 - ❖ Make change for connection string of your server's name in the LogisticsShipmentAPI => appsettings.json as below



- ❖ Open another terminal in VS Code: **cd LogisticsShipmentAPI**
- ❖ Restore .NET dependencies: **dotnet restore**
- ❖ Apply migrations:
 - dotnet tool install --global dotnet-ef** (if ef tool is not installed)

dotnet ef migrations add InitialCreate

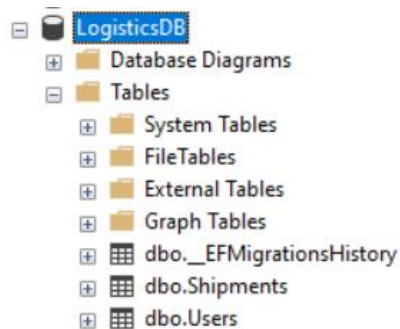
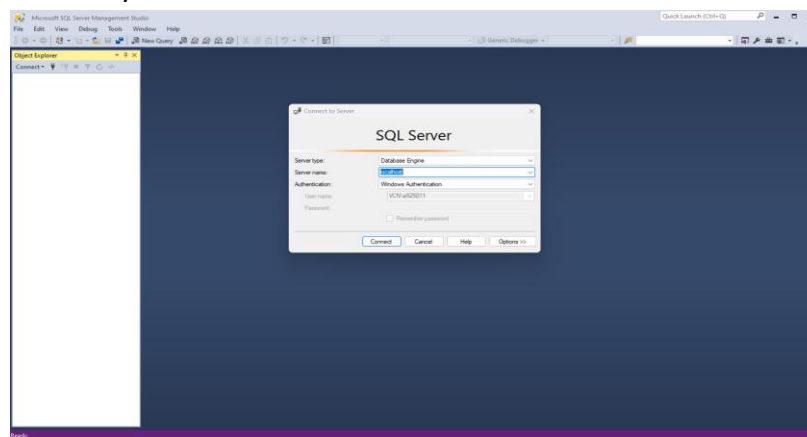
dotnet ef database update

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: powershell
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\VA525011\OneDrive - Volvo Group\Logistics> cd .\LogisticsShipmentAPI\
PS C:\Users\VA525011\OneDrive - Volvo Group\Logistics\LogisticsShipmentAPI> dotnet ef migrations add InitialCreate
Build started...
Build succeeded.
Done. To undo this action, use 'ef migrations remove'
PS C:\Users\VA525011\OneDrive - Volvo Group\Logistics\LogisticsShipmentAPI> dotnet ef database update
Build started...
Build succeeded.
[20:23:15 INF] Executed DbCommand (403ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
CREATE DATABASE [LogisticsDB];
[20:23:15 INF] Executed DbCommand (145ms) [Parameters=[], CommandType='Text', CommandTimeout='60']
IF SERVERPROPERTY('ServerEdition') < 5
```

- Verify the database and tables are created in the defined server name as below:



- Insert some values to the Shipments table (if required)

INSERT INTO Shipments (TrackingNumber, Product, Origin, Destination, Status, PackedOn, ExpectedDelivery, CreatedAt)

VALUES

('TRK-401900', 'Scooter', 'US', 'UK', 'Delivered', '2025-07-13', '2025-07-15', '2025-07-07'),
('TRK-401901', 'Bike', 'US', 'UK', 'Delivered', '2025-07-14', '2025-07-16', '2025-07-08'),
('TRK-401902', 'Laptop', 'US', 'CA', 'In Transit', '2025-07-10', '2025-07-15', '2025-07-08'),
('TRK-401903', 'Phone', 'UK', 'IN', 'Created', '2025-07-12', '2025-07-18', '2025-07-09'),
('TRK-401904', 'Tablet', 'CA', 'AU', 'Delivered', '2025-07-09', '2025-07-13', '2025-07-07'),

```
(
    'TRK-401905', 'Camera', 'IN', 'US', 'In Transit', '2025-07-15', '2025-07-20', '2025-07-08'),
    ('TRK-401906', 'Watch', 'JP', 'UK', 'Delayed', '2025-07-11', '2025-07-14', '2025-07-08'),
    ('TRK-401907', 'Speaker', 'DE', 'FR', 'Pending', '2025-07-13', '2025-07-19', '2025-07-09'),
    ('TRK-401908', 'Headphones', 'US', 'DE', 'Created', '2025-07-10', '2025-07-17', '2025-07-07'),
    ('TRK-401909', 'Monitor', 'UK', 'US', 'In Transit', '2025-07-12', '2025-07-16', '2025-07-08'),
    ('TRK-401910', 'Keyboard', 'FR', 'IN', 'Delivered', '2025-07-14', '2025-07-21', '2025-07-08'),
    ('TRK-401911', 'Mouse', 'CN', 'US', 'Pending', '2025-07-15', '2025-07-22', '2025-07-09'),
    ('TRK-401912', 'Printer', 'US', 'MX', 'Delayed', '2025-07-16', '2025-07-23', '2025-07-10');

```

```
SQLQuery1.sql - loc... (VCN\A525011 (55)) * X
INSERT INTO Shipments (TrackingNumber, Product, Origin, Destination, Status, PackedOn, ExpectedDelivery, CreatedAt)
VALUES
('TRK-401900', 'Scooter', 'US', 'UK', 'Delivered', '2025-07-13', '2025-07-15', '2025-07-07'),
('TRK-401901', 'Bike', 'US', 'UK', 'Delivered', '2025-07-14', '2025-07-16', '2025-07-08'),
('TRK-401902', 'Laptop', 'US', 'CA', 'In Transit', '2025-07-10', '2025-07-15', '2025-07-08'),
('TRK-401903', 'Phone', 'UK', 'IN', 'Created', '2025-07-12', '2025-07-18', '2025-07-09'),
('TRK-401904', 'Tablet', 'CA', 'AU', 'Delivered', '2025-07-09', '2025-07-13', '2025-07-07'),
('TRK-401905', 'Camera', 'IN', 'US', 'In Transit', '2025-07-15', '2025-07-20', '2025-07-08'),
('TRK-401906', 'Watch', 'JP', 'UK', 'Delayed', '2025-07-11', '2025-07-14', '2025-07-08'),
('TRK-401907', 'Speaker', 'DE', 'FR', 'Pending', '2025-07-13', '2025-07-19', '2025-07-09'),
('TRK-401908', 'Headphones', 'US', 'DE', 'Created', '2025-07-10', '2025-07-17', '2025-07-07'),
('TRK-401909', 'Monitor', 'UK', 'US', 'In Transit', '2025-07-12', '2025-07-16', '2025-07-08'),
('TRK-401910', 'Keyboard', 'FR', 'IN', 'Delivered', '2025-07-14', '2025-07-21', '2025-07-08'),
('TRK-401911', 'Mouse', 'CN', 'US', 'Pending', '2025-07-15', '2025-07-22', '2025-07-09'),
('TRK-401912', 'Printer', 'US', 'MX', 'Delayed', '2025-07-16', '2025-07-23', '2025-07-10');

100 %
Messages
(13 rows affected)
```

- Run the application:

- ❖ **dotnet restore**
- ❖ **dotnet build**
- ❖ **dotnet run**

```
PS C:\Users\A525011\OneDrive - Volvo Group\Logistics\LogisticsShipmentAPI> dotnet restore
Restore complete (0,3s)

Build succeeded in 0,5s
PS C:\Users\A525011\OneDrive - Volvo Group\Logistics\LogisticsShipmentAPI> dotnet build
Restore complete (0,3s)
LogisticsShipmentAPI succeeded (0,3s) -> bin\Debug\net9.0\LogisticsShipmentAPI.dll

Build succeeded in 1,0s
PS C:\Users\A525011\OneDrive - Volvo Group\Logistics\LogisticsShipmentAPI> dotnet run
Using launch settings from C:\Users\A525011\OneDrive - Volvo Group\Logistics\LogisticsShipmentAPI\Properties\launchSettings.json...
Building...
[20:29:21 INF] Application starting
[20:29:22 INF] Executed DbCommand (9ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
SELECT 1
[20:29:22 INF] Acquiring an exclusive lock for migration application. See https://aka.ms/efcore-docs-migrations-lock for more information.
[20:29:22 INF] Executed DbCommand (12ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
DECLARE @result int;
EXEC @result = sp_getapplock @Resource = '__EFMigrationsLock', @LockOwner = 'Session', @LockMode = 'Exclusive';
SELECT @result
[20:29:22 INF] Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
IF OBJECT_ID(N'[_EFMigrationsHistory]') IS NULL
BEGIN
    CREATE TABLE [_EFMigrationsHistory] (
        [MigrationId] nvarchar(150) NOT NULL,
        [ProductVersion] nvarchar(32) NOT NULL,
        CONSTRAINT [PK__EFMigrationsHistory] PRIMARY KEY ([MigrationId])
    );

```

- ❖ Open another terminal in VS Code: **cd LogisticsShipmentUI**
- ❖ Run React Application: **npm start**

```
PS C:\Users\A525011\OneDrive - Volvo Group\Logistics\LogisticsShipmentUI> npm start

> logistics@0.1.0 start
> react-scripts start

(node:33504) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware'
wares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:33504) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddlewar
lewares' option.
Starting the development server...
Compiled successfully!

You can now view logistics in the browser.

Local:      http://localhost:3600
On Your Network: http://10.144.2.245:3600

Note that the development build is not optimized.
To create a production build, use npm run build.

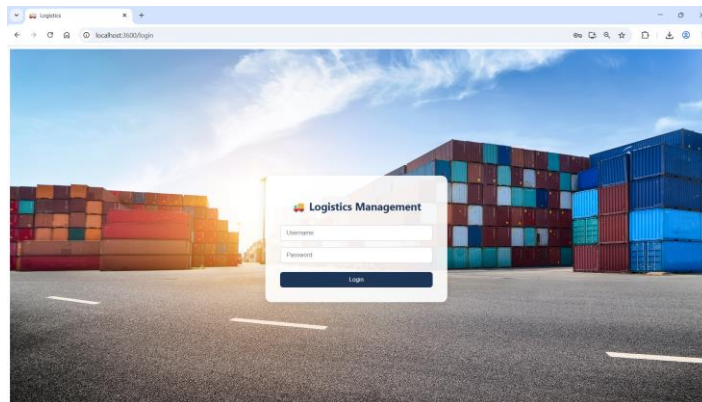
webpack compiled successfully
█
```

Logistics Management Application

Credentials:

Role	Username	Password
Admin	admin	admin123
Manager	manager	manager123
Viewer	viewer	viewer123

Login page:



Dashboard:

Logistics Dashboard (Admin)

Order Number	Tracking Number	Products	Origin	Destination	Status	Tracked On	Expected Delivery	Actions
1	790-401000	Scanner	US	UK	Delivered	2020-07-10	2020-07-10	↗ ✖
2	790-401001	Blow	US	UK	Delivered	2020-07-14	2020-07-16	↗ ✖
3	790-401002	Laptop	US	CA	In Transit	2020-07-16	2020-07-18	↗ ✖
4	790-401003	Phone	UK	IN	Created	2020-07-10	2020-07-16	↗ ✖
5	790-401004	Tablet	CA	AU	Delivered	2020-07-09	2020-07-10	↗ ✖
6	790-401005	Camera	IN	US	In Transit	2020-07-10	2020-07-20	↗ ✖
7	790-401006	Mouse	JP	UK	Delayed	2020-07-11	2020-07-14	↗ ✖
8	790-401007	Speaker	DE	IN	Pending	2020-07-10	2020-07-16	↗ ✖
9	790-401008	Headphones	US	DE	Created	2020-07-10	2020-07-17	↗ ✖
10	790-401009	Monitor	UK	US	In Transit	2020-07-10	2020-07-16	↗ ✖

Page 1 of 2

localhost:3000/shippers

Order Number	Tracking Number	Products	Origin	Destination	Status	Tracked On	Expected Delivery	Actions
1	790-401000	Scanner	US	UK	Delivered	2020-07-10	2020-07-10	↗ ✖
2	790-401001	Blow	US	UK	Delivered	2020-07-14	2020-07-16	↗ ✖
3	790-401002	Laptop	US	CA	In Transit	2020-07-16	2020-07-18	↗ ✖
4	790-401003	Phone	UK	IN	Created	2020-07-10	2020-07-16	↗ ✖
5	790-401004	Tablet	CA	AU	Delivered	2020-07-09	2020-07-10	↗ ✖
6	790-401005	Camera	IN	US	In Transit	2020-07-10	2020-07-20	↗ ✖
7	790-401006	Mouse	JP	UK	Delayed	2020-07-11	2020-07-14	↗ ✖
8	790-401007	Speaker	DE	IN	Pending	2020-07-10	2020-07-16	↗ ✖
9	790-401008	Headphones	US	DE	Created	2020-07-10	2020-07-17	↗ ✖
10	790-401009	Monitor	UK	US	In Transit	2020-07-10	2020-07-16	↗ ✖

Page 1 of 2

New Shipment

Products

Phone

Origin

UK

Destination

IN

Status

Created

Tracked On

4th Jul 2020

Expected Delivery

4th Jul 2021

Submit

Cancel

localhost:3000/shippers

Order Number	Tracking Number	Products	Origin	Destination	Status	Tracked On	Expected Delivery	Actions
1	790-401000	Scanner	US	UK	Delivered	2020-07-10	2020-07-10	↗ ✖
2	790-401001	Blow	US	UK	Delivered	2020-07-14	2020-07-16	↗ ✖
3	790-401002	Laptop	US	CA	In Transit	2020-07-16	2020-07-18	↗ ✖
4	790-401003	Phone	UK	IN	Created	2020-07-10	2020-07-16	↗ ✖
5	790-401004	Tablet	CA	AU	Delivered	2020-07-09	2020-07-10	↗ ✖
6	790-401005	Camera	IN	US	In Transit	2020-07-10	2020-07-20	↗ ✖
7	790-401006	Mouse	JP	UK	Delayed	2020-07-11	2020-07-14	↗ ✖
8	790-401007	Speaker	DE	IN	Pending	2020-07-10	2020-07-16	↗ ✖
9	790-401008	Headphones	US	DE	Created	2020-07-10	2020-07-17	↗ ✖
10	790-401009	Monitor	UK	US	In Transit	2020-07-10	2020-07-16	↗ ✖

Page 1 of 2

Edit Shipment

Products

Phone

Origin

UK

Destination

IN

Status

Created

Tracked On

4th Jul 2020

Expected Delivery

4th Jul 2021

Save Shipment

Cancel

localhost:3000/shippers

Order Number	Tracking Number	Products	Origin	Destination	Status	Tracked On	Expected Delivery	Actions
1	790-401000	Scanner	US	UK	Delivered	2020-07-10	2020-07-10	↗ ✖
2	790-401001	Blow	US	UK	Delivered	2020-07-14	2020-07-16	↗ ✖
3	790-401002	Laptop	US	CA	In Transit	2020-07-16	2020-07-18	↗ ✖
4	790-401003	Phone	UK	IN	Created	2020-07-10	2020-07-16	↗ ✖
5	790-401004	Tablet	CA	AU	Delivered	2020-07-09	2020-07-10	↗ ✖
6	790-401005	Camera	IN	US	In Transit	2020-07-10	2020-07-20	↗ ✖
7	790-401006	Mouse	JP	UK	Delayed	2020-07-11	2020-07-14	↗ ✖
8	790-401007	Speaker	DE	IN	Pending	2020-07-10	2020-07-16	↗ ✖
9	790-401008	Headphones	US	DE	Created	2020-07-10	2020-07-17	↗ ✖
10	790-401009	Monitor	UK	US	In Transit	2020-07-10	2020-07-16	↗ ✖

Page 1 of 2

Confirm Action

Are you sure you want to delete this shipment?

Yes

Cancel