# # TransportApp - A Case Study

**## Problem Statement**

Build a system to search for a Items, get Items statistics and add Items to Tr ansport list.

# ## Requirements

find Items, get Items statistics and add Items to Transport list.

**-** Logout feature should be implemented.

* User can add a Items to Transport list and should be able view the Transport list.
* JWT authentication (Optional)
* Sorting and pagination (optional)

# ## Modules

**### (User interface) - should be able to**

* + Register user, login and logout
  + Search a Items by name
  + View Items statistics
  + Add a Items to Transport list
  + should be able to see Transport Items

# ### UserService

* + Creating a server in Asp.net Core

APIto facilitate user registration and login using SQL Server

* + should be able to manage user accounts.

# ### TransportService

* + should be able to store all the Transport Items for a user

# ## Tech Stack

* + - Asp.net Core Web API
    - Entity Framework Core
    - SQL Server
    - React

# ## Flow of Modules ### Building

* + - * Register/Login
      * Search for a Items
      * Items list - populating from external API

Transport mode - View all transport to a destination from https://developer.transportapi.com/

# ### Building the UserService

1. Creating a server in Asp.net Core

APIto facilitate user registration and login with Sql Server as backend. Upon login.

1. Writing swagger documentation
2. Unit Testing of the entire code which involves the positive and negative scenarios

# ### Building the Transport Service

1. Building a server in Asp.net Core

APIto facilitate CRUD operation over Transport Items stored in Sql Server.

1. Writing Swagger Documentation
2. Write Unit Test Cases and get it executed.

# ### Demonstrate the entire application

* 1. Make sure all the functionalities are implemented
  2. server side (For all layers) codes are unit tested.