



# .NET Core Sprint Cargo Manager System

Document Revision History

Date	Revision No.	Author	Summary of Changes
13-07-2017	1.0	Pankaj Patil	Case Study Created
30-05-2023	2.0	Rekha Chaudhari	Merged both sprints into single one as per new format.



## Table of Contents

<b>Introduction</b>	<b>4</b>
Setup Checklist	4
Instructions	4
<b>Problem Statement</b>	<b>10</b>
Objective	10
Project structure	14
Implementation	15
Summary of the functionality to be built:	15

## INTRODUCTION

---

This document outlines a project for the .NET Line of Technology (LOT). The project is to develop Cargo Manager System. This document contains the requirements, workflow of the system and gives guidelines on how to build the functionality gradually in each of the course modules of the .NET LOT.

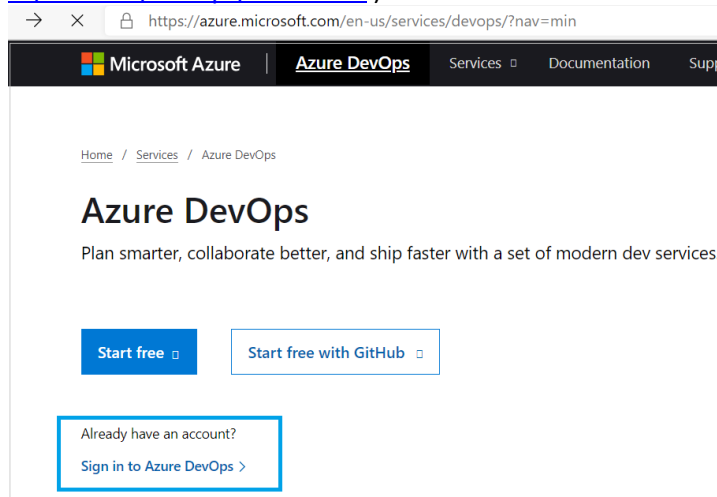
## SETUP CHECKLIST

### Minimum System Requirements

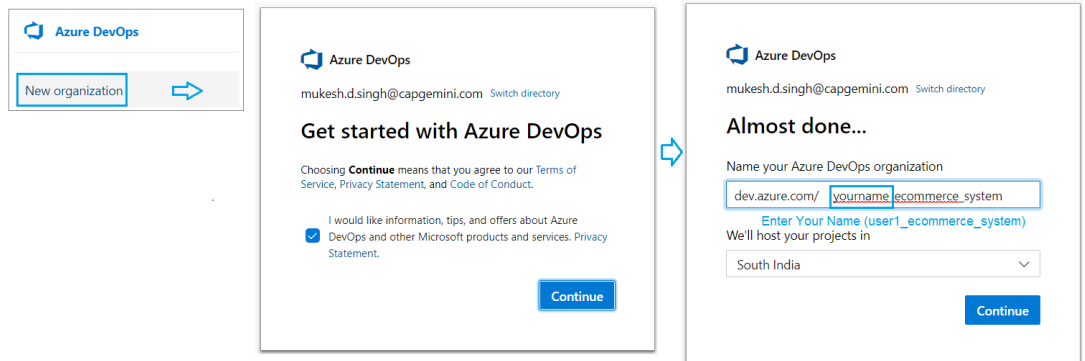
- Intel Pentium 4 and above Windows 7
- Memory 8 GB
- Edge / Chrome browser
- SQL Server 2019 and SQL Server Management Studio
- Visual Studio 2022
- Visual Studio Code
- Git

## INSTRUCTIONS

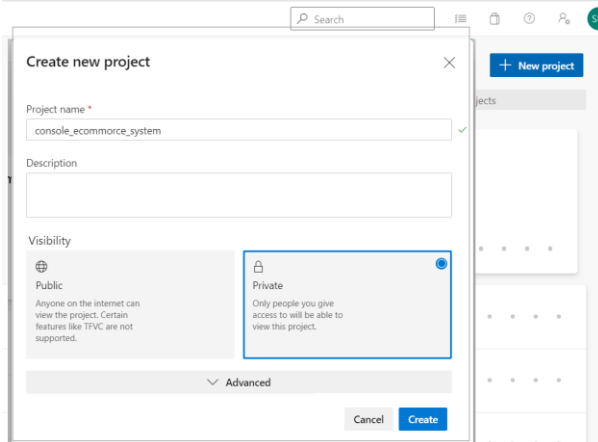
- The code modules in the mini project should follow all the coding standards.
- Sign in to Azure DevOps (link - <https://azure.microsoft.com/en-us/services/devops/?nav=min> )



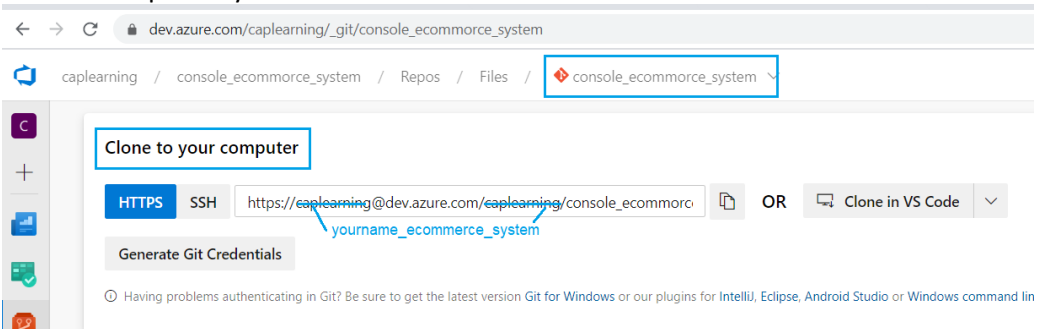
- After creating account, login to <https://dev.azure.com/>  
**Note:** You can select organization name and project name according to your choice and availability.



- Create New Project. Give a name according to your project. ( Here, just for the sake of a demo the name - ecommerce is given)



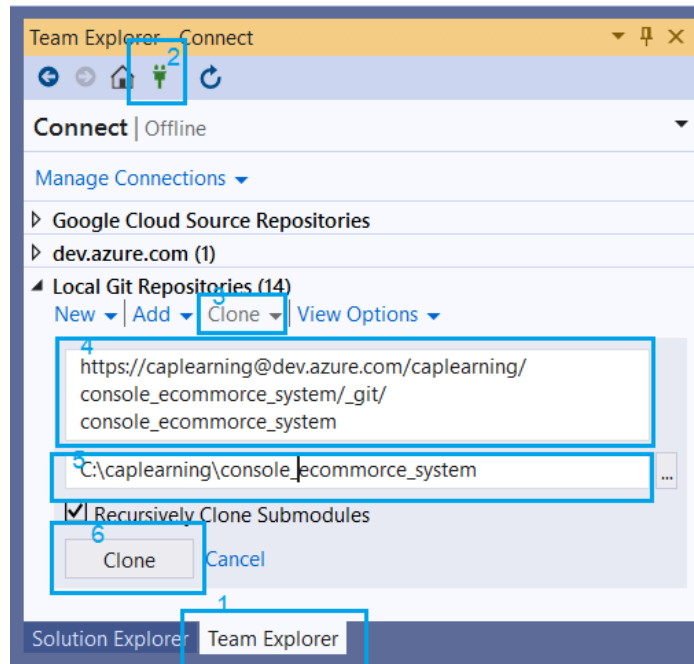
- Clone the repository



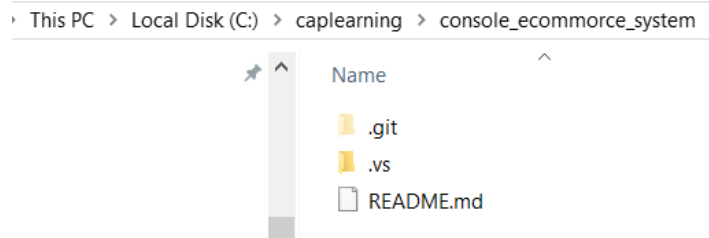
Copy the clone URL

Click on Initialize button at the bottom of screen

- Clone in Visual Studio



After Clone, below folders get created in your local machine



- Create New Blank Solution at your above local machine folder

### Configure your new project

**Blank Solution** Other

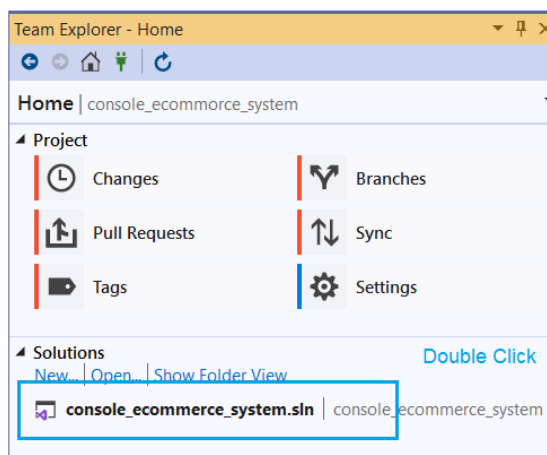
Project name  
console\_ecommerce\_system

Location  
C:\caplearning\console\_ecommerce\_system\

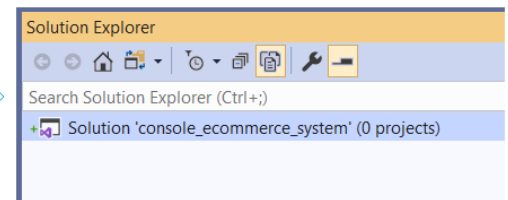
Solution  
Create new solution

Solution name ⓘ  
console\_ecommerce\_system

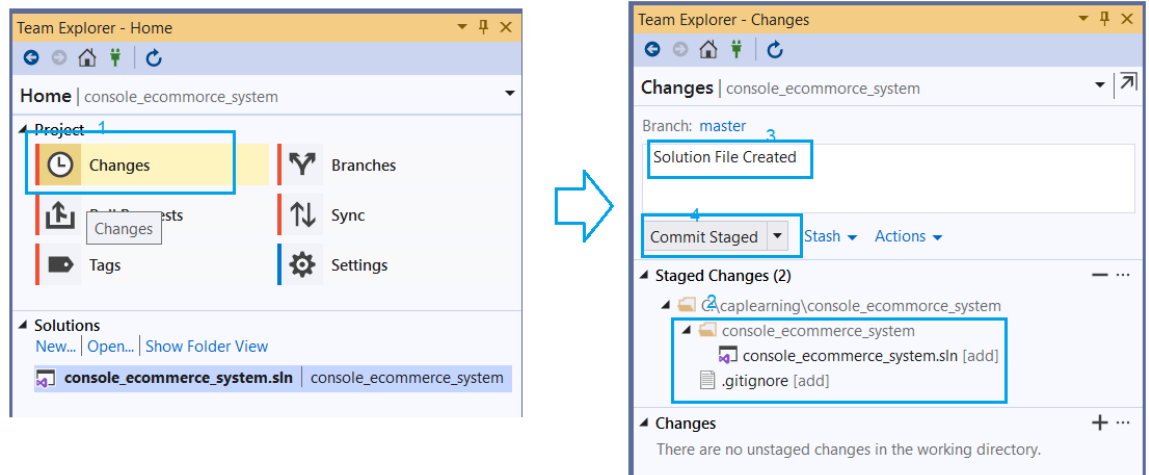
Back Create



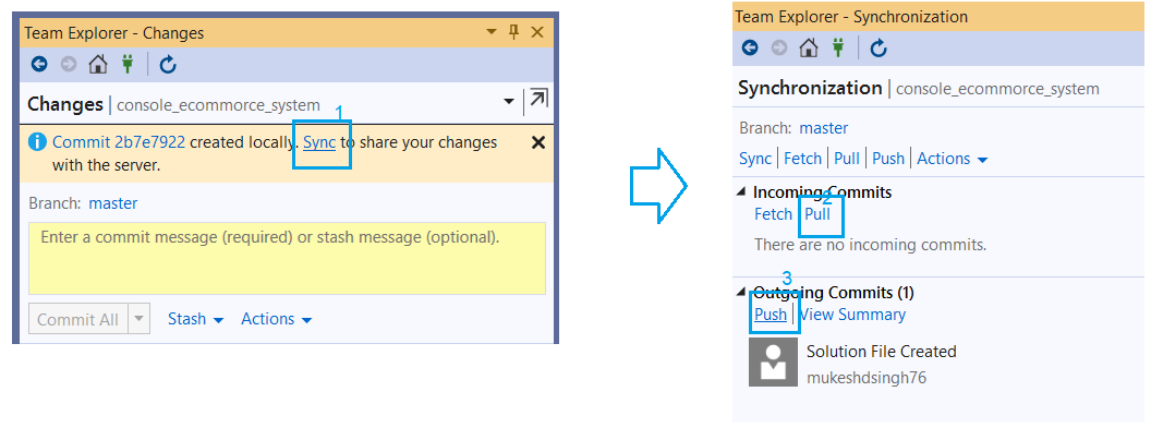
Open Solution



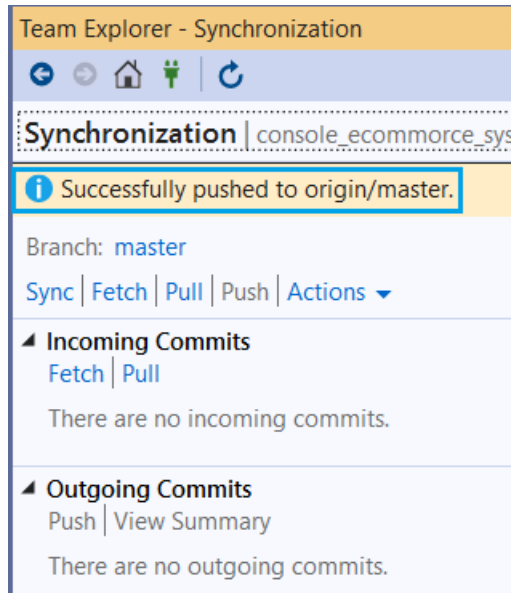
- Commit blank solution changes into repository



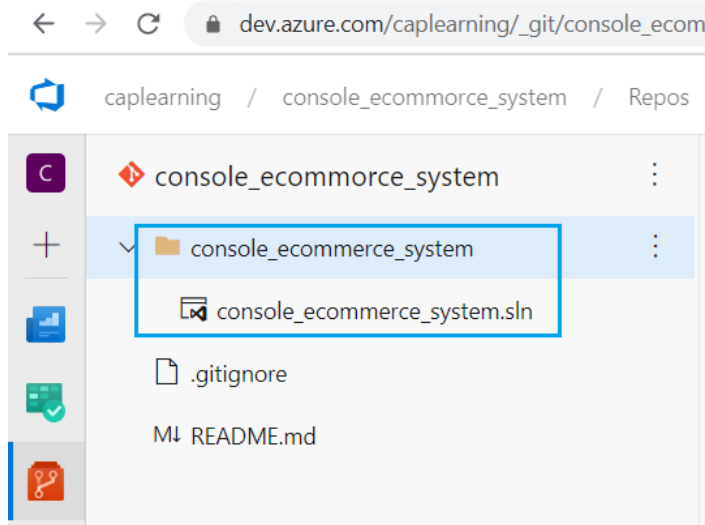
Push changes to devops repository







- Confirm changes are pushed to your DevOps repository



- Create Feature Branch  
Create New Branch from master.  
Feature Branch name: feature/cargomgr-system

## PROBLEM STATEMENT

---

### OBJECTIVE

Need to develop Cargo Manager System.

#### **Abstract of the project:**

**Cargo Manager** is a comprehensive cargo management module, designed for addressing the areas of General cargo, Bulk cargo operations through Aviation. All the aspects of cargo like documentation handling, movement, and storage are addressed in this module.

Cargo Manager system will have following features:

- General Cargo Booking (Manifestation of cargo)
- Storing of Cargo (actual cargo weight, volume etc. are recorded)
- Generating the gate pass for bringing the cargo into the warehouse
- Marking the truck exit
- Delivery order for rejected or damaged cargo
- Generating gate pass for redelivery of cargo
- Housekeeping movements for maintenance i.e., moving the cargo from one place to another
- Extensive report and querying facilities

Users of the System will be Cargo Manager Administrator.

Role of Cargo Manager Administrator:

- Login to System
- Registration on System
- Search for Address
- Calculating Cargo Price
- Book a Cargo
- View Status of Cargo
- Cancel the Cargo

**Phase 1:** Create Client Application in Angular and Services using ASP.NET Core Web API

## MODULE LIST and MODULE DETAILS

### Modules:

- Customer: - This will contain methods which will allow to manage all Customers
- Employee: - This will contain methods to manage Employee
- CargoOrder :- This will Contain methods which will allow to manage all Order Placed by Customer
- CargoOrderDetails :- This will Contain methods which will allow to manage all Order Placed by Customer
- City: - This will contain methods to manage City details
- Product: This will contain methods to manage Product

i. **Login / Sign In:** Login screen would display asking Cargo Manager user to enter 'User Id' & 'Password'. If the supplied user credentials are valid the Homepage would be displayed, else appropriate Login error message would be displayed

ii. **Home Page:** On successful user authentication (validation of userid/password provided by the user in login screen) the homepage would be displayed. The Homepage would contain below sections/contents:

Header section: The header section would be common across all the pages and would mainly have –

- Sign In Link – On click it would take to Login page
- Search: Search for information

Main content section: It would display generic Welcome message giving overview of the site. Should have menu for Administrator Employee, Customer

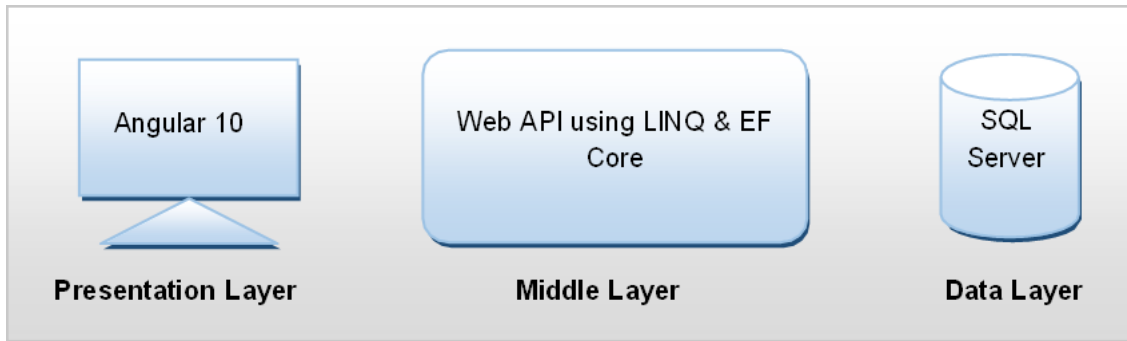
a. For Administrator:

- Register a new Employee on Cargo Manager System
- Delete an existing Employee on Cargo Manager System
- Update an existing Employee on Cargo Manager System
- Search for Employee and Customer.
- Add new Cargo Type.

- vi. Update Cargo Type.
- vii. Update Places and Prices.
- b. For Employee:
  - i. Update self-Record on Cargo Manager System
  - ii. Search for Customer.
  - iii. Search for Cargo fare
  - iv. Search for Order
  - v. Update Cargo Type.
  - vi. Update Places
  - vii. Update Customer
- c. For Customer
  - i. User will get option to add, update its details
  - ii. View the details of all Order Placed

## PROJECT STRUCTURE

The project should have minimum target framework – .NET 6.0 for Core WEB API Services and Angular 10 for UI Design.



## Design guidelines

- All the exceptions/errors to be captured and user-friendly message to be displayed on the Common Error page.
- .NET Web API must be created using Code First Approach.

## IMPLEMENTATION

### SUMMARY OF THE FUNCTIONALITY TO BE BUILT:

The participants need to develop the Cargo Manager System by building the functionality incrementally in each of the course modules of .NET LOT.

Sr. No	Course	Duration	Functionality to be built
		(in PDs)	
1	NET Core 6.0 Entity Framework Core Angular ASP.NET Core Web API	5	Developing Presentation components (Angular Entity Framework Core, ASP.NET Core Web API)