Installation Guide of Fleet Service Application



Spartanburg Community College - 107 Community College Drive, Spartanburg, SC 29303- (864) 592-4600 - <https://www.sccsc.edu/>

Table of Contents

[1. Introduction 2](#_Toc140675085)

[2. Visual Studio Requirements 2](#_Toc140675086)

[3. Downloading and First Start of the application 3](#_Toc140675087)

[4. Setting up a new database to run the application on 5](#_Toc140675088)

[5. Schema Script 7](#_Toc140675089)

[6. SMTP Modification (Sender Email) 11](#_Toc140675090)

[7. Publish the project and promoting on the server 12](#_Toc140675091)

[Promoting to the server 15](#_Toc140675092)

[8. Setting up an Event Viewer Log 17](#_Toc140675093)

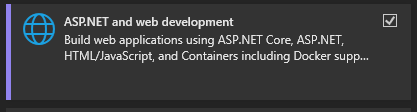
# Introduction

Welcome to the installation guide for the Fleet Service program! This guide provides step-by-step instructions to help you install and set up the Fleet Service program on your system. The Fleet Service program is designed to streamline and manage the operations of a fleet management service, allowing you to efficiently track and maintain vehicles, manage inspections, and generate insightful reports.

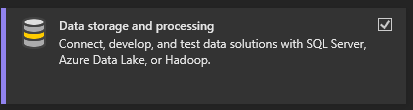
# Visual Studio Requirements

To run the application in the visual studio you need to have the following Workloads installed in the visual studio.

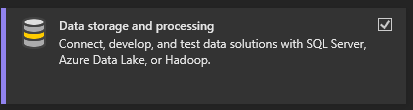
* ASP.NET and web development



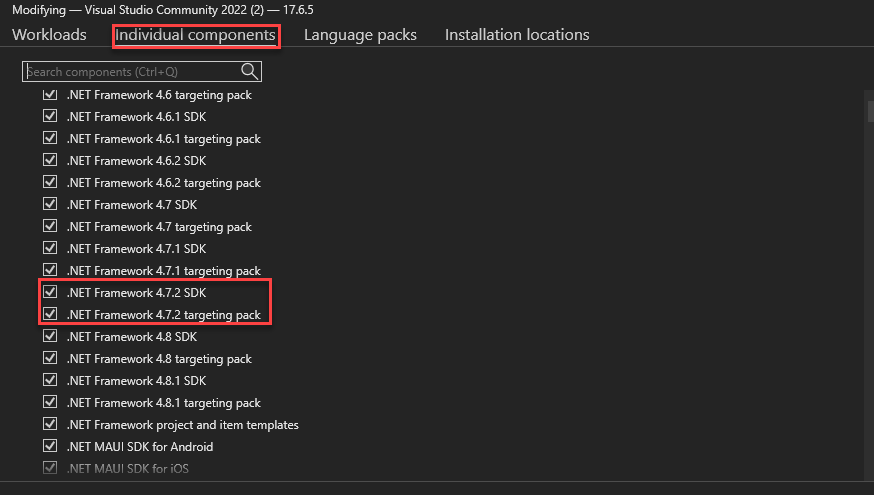
* Data storage and processing



* Data science and analytical application



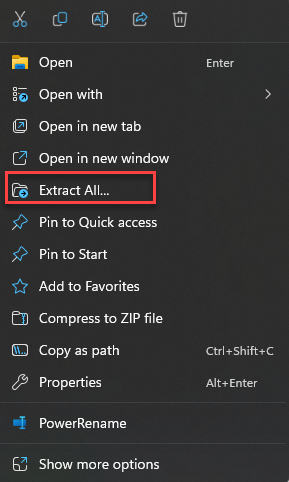
* .Net Framework 4.7.2 needs to be installed in visual studio to troubleshoot and publish the application through the Microsoft Visual Studio



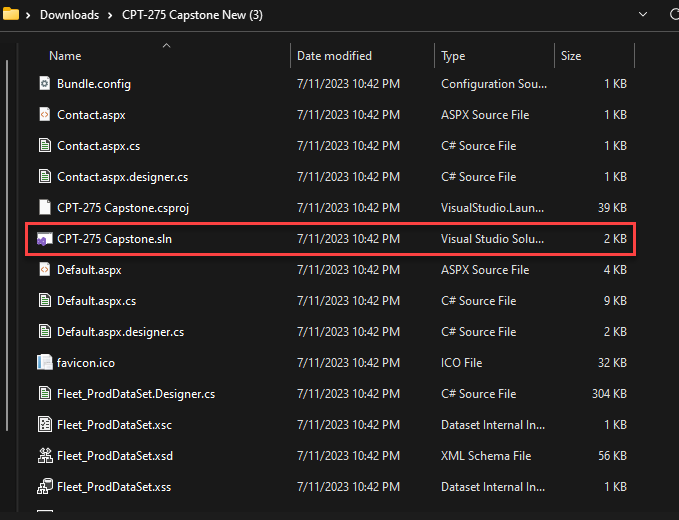
# Downloading and First Start of the application

Fleet Service application will be provided to the IT department by the SCC students from the Group 2 as a zip package of the application.

1. Once the application is provided to the IT as a zip folder, then the application needs to be unzipped



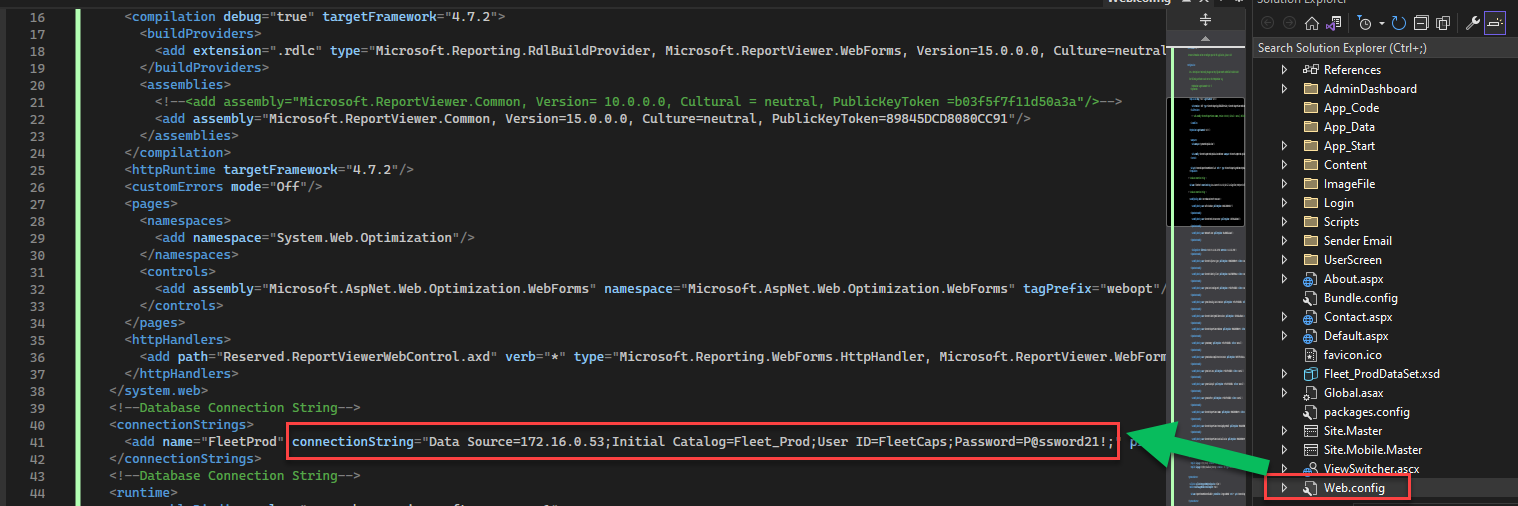
1. Once the application is unzipped, it can be started by clicking the “CPT-275 Capstone.sln”



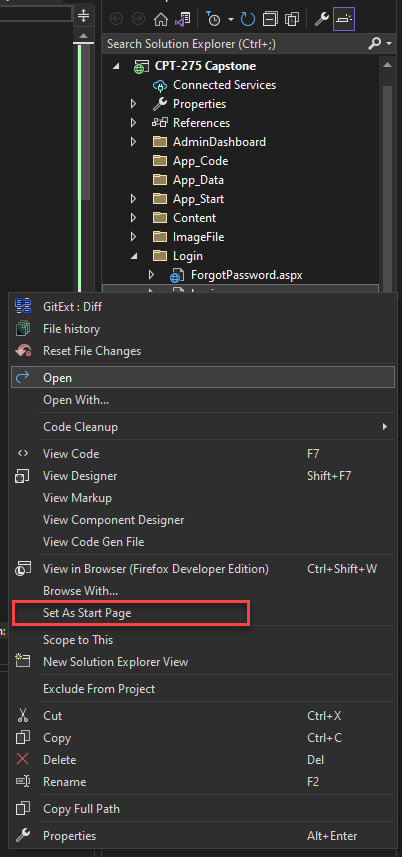
1. Once the application is opened in the Microsoft Visual Studio, the first thing that needs to be done in order to start the application is to point the application the new SQL server, with the new SQL database.

Fields that needs to be modified are: ConnectionString, Initial Catalog, User ID, Password

These are the credentials that are used to access the SQL server and the database that is on.



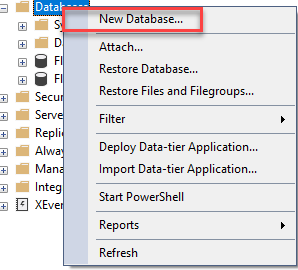
Second, set the Login page under the Login folder as a Start Page.



# Setting up a new database to run the application on

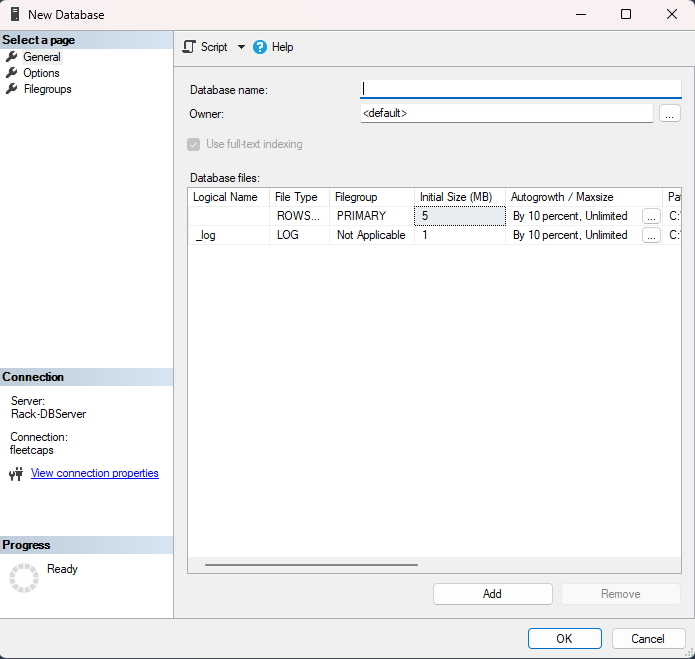
In order to run the application on the server the SQL database needs to be created with the correct table structure and reports to use to run the application.

To create a new database in the Microsoft SQL Server Management Studio, you would need to connect to the Database server and right click the “Databases” folder.



On the New Database prompt, you would need to enter a database name that will be used and give an access to the users that needs to have access to the database.

\*\*Application does not require any accounts to have access to the database\*\*



Once the database is created you would need to use the Schema script that was provided by the Lead Programmer of the application.

# Schema Script

--Create a vehicle table

CREATE TABLE vehicle (

    vehicle\_id int identity (1,1) primary key,

    vehicle\_year varchar(50) NOT NULL,

    vehicle\_make varchar(255) not null,

    vehicle\_model varchar (50) not null,

    vehicle\_number char (20),

    vehicle\_mileage char (50),

    vehicle\_plate char (20),

    vehicle\_no\_default char (20),

);

--Users table

CREATE TABLE users(

    users\_last\_name varchar(50) NOT NULL,

    users\_first\_name varchar(50) NOT NULL,

    users\_telephone char(10) NULL,

    users\_email varchar(50) NULL,

    users\_password char(20) NULL,

    users\_DL char(20) NULL,

    users\_DL\_state [char](20) NULL,

    users\_type char(20) NULL,

    vehicle\_id int NULL,

    user\_id int IDENTITY(1,1) NOT NULL,

PRIMARY KEY CLUSTERED

(

    [user\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

--Inspection table

CREATE TABLE inspection(

    inspection\_id int IDENTITY(1,1) NOT NULL,

    inspection\_beginning\_mileage decimal(18, 0) NOT NULL,

    inspection\_ending\_mileage decimal(18, 0) NOT NULL,

    inspection\_total\_mileage\_driven decimal(18, 0) NULL,

    inspection\_last\_oil\_change\_date date NULL,

    inspection\_oil\_change\_due date NULL,

    inspection\_interval varchar(10) NULL,

    inspection\_last\_tire\_rotation date NULL,

    inspection\_tires\_rotation\_due date NULL,

    inspection\_tires\_pressure decimal(18, 0) NULL,

    vehicle\_number int NULL,

    inspection\_additional\_notes varchar(250) NULL,

    trip\_fluid\_level varchar(10) NULL,

    battery\_good varchar(10) NULL,

    gauge\_working varchar(10) NULL,

    clean\_cab varchar(10) NULL,

    clean\_exterior varchar(10) NULL,

    inspection\_date date NULL,

PRIMARY KEY CLUSTERED

(

    [inspection\_id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

--Trip table

CREATE TABLE trip(

    trip\_date date NULL,

    trip\_beginning\_mileage char(20) NOT NULL,

    trip\_destination varchar(100) NOT NULL,

    trip\_purpose varchar(100) NOT NULL,

    trip\_ending\_mileage varchar(20) NOT NULL,

    trip\_total\_miles char(20) NULL,

    vehicle\_id varchar(150) NULL,

    user\_id int NULL

);

--\*\*Reports-----

--Inspection Report--------

/\*\*\*\*\*\* Object:  StoredProcedure [dbo].[inspection\_report]    Script Date: 7/11/2023 9:58:03 PM \*\*\*\*\*\*/

/\*\*\*\*\*\* Object:  StoredProcedure [dbo].[inspection\_report]    Script Date: 7/12/2023 10:57:53 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE procedure inspection\_report

(@beginning\_date date, @ending\_date date, @selected\_vehicle int=null) as

if @selected\_vehicle is null and @beginning\_date is not null and @ending\_date  is not null

begin

select

cast (v.vehicle\_year+ ' '+v.vehicle\_make+ ' ' +v.vehicle\_model+ ' ' +v.vehicle\_plate as varchar(50)) 'Vehicle',

i.inspection\_beginning\_mileage 'Beginning Mileage', i.inspection\_ending\_mileage 'Ending Mileage',

CONVERT(VARCHAR(10),i.inspection\_last\_oil\_change\_date,101) 'Last Oil Change', CONVERT(VARCHAR(10), i.inspection\_oil\_change\_due,101) 'Oil Change Due',

CONVERT(VARCHAR(10),i.inspection\_last\_tire\_rotation,101) 'Last Tire Change', CONVERT(VARCHAR(10),i.inspection\_tires\_rotation\_due,101) 'Rotation Due',

i.inspection\_tires\_pressure 'Tire Pressure', i.inspection\_additional\_notes 'Additional Notes'

from inspection i join vehicle v on i.vehicle\_number=v.vehicle\_id

where inspection\_date between @beginning\_date and @ending\_date

order by i.inspection\_oil\_change\_due, inspection\_date asc

end

else

begin

select

cast (v.vehicle\_year+ ' '+v.vehicle\_make+ ' ' +v.vehicle\_model+ ' ' +v.vehicle\_plate as varchar(50)) 'Vehicle',

i.inspection\_beginning\_mileage 'Beginning Mileage', i.inspection\_ending\_mileage 'Ending Mileage',

CONVERT(VARCHAR(10),i.inspection\_last\_oil\_change\_date,101) 'Last Oil Change',CONVERT(VARCHAR(10), i.inspection\_oil\_change\_due,101) 'Oil Change Due',

CONVERT(VARCHAR(10),i.inspection\_last\_tire\_rotation,101) 'Last Tire Change', CONVERT(VARCHAR(10),i.inspection\_tires\_rotation\_due,101) 'Rotation Due',

i.inspection\_tires\_pressure 'Tire Pressure', i.inspection\_additional\_notes 'Additional Notes'

from inspection i join vehicle v on i.vehicle\_number=v.vehicle\_id

where inspection\_date between @beginning\_date and @ending\_date

and @selected\_vehicle=v.vehicle\_id

order by i.inspection\_oil\_change\_due, inspection\_date asc

end

GO

--Trip Report

/\*\*\*\*\*\* Object:  StoredProcedure [dbo].[trip\_report\_new]    Script Date: 7/12/2023 10:58:37 PM \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

--Trip Date, Trip Destination, Trip Purpose, Trip Total Miles

--ALTER TABLE table\_name

--RENAME COLUMN old\_name TO new\_name;

--ALTER TABLE employee ALTER COLUMN emp\_parking\_space BIGINT;

--EXEC sp\_rename 'trip\_report\_new.trip\_date', 'Trip Date';

CREATE procedure trip\_report\_new

(@trip\_beginning\_date DATETIME, @trip\_ending\_date DATETIME, @selectd\_vehicle INT = null, @select\_user INT = null) AS

if @select\_user is null and @selectd\_vehicle is null            -- SCENARIO 1: Date parameters included but No User and no Vehicle parameter

    begin

        print '1'

        select cast(u.users\_last\_name+', '+u.users\_first\_name as char(20)) 'Full Name', CONVERT(VARCHAR(10),t.trip\_date,101) 'Trip Date', t.trip\_destination 'Trip Destination',

        t.trip\_purpose 'Trip Purpose', t.trip\_total\_miles 'Trip Total Miles', cast (v.vehicle\_year+ ' '+v.vehicle\_make+ ' ' +v.vehicle\_model+ ' ' +v.vehicle\_plate as varchar(50)) 'Vehicle'

        from trip t join vehicle v on t.vehicle\_id=v.vehicle\_id join users u on (t.user\_id = u.user\_id)

        where trip\_date between @trip\_beginning\_date and  @trip\_ending\_date

        order by trip\_date asc

    end

else if @select\_user is null and @selectd\_vehicle is NOT null   -- SCENARIO 2: Date and Vehicle parameters included but No User parameter

    begin

        print '2'

        select cast(u.users\_last\_name+', '+u.users\_first\_name as char(20)) 'Full Name',CONVERT(VARCHAR(10), t.trip\_date,101) 'Trip Date', t.trip\_destination 'Trip Destination',

        t.trip\_purpose 'Trip Purpose', t.trip\_total\_miles'Trip Total Miles', cast (v.vehicle\_year+ ' '+v.vehicle\_make+ ' ' +v.vehicle\_model+ ' ' +v.vehicle\_plate as varchar(50)) 'Vehicle'

        from trip t join vehicle v on t.vehicle\_id=v.vehicle\_id join users u on (t.user\_id = u.user\_id)

        where trip\_date between @trip\_beginning\_date and  @trip\_ending\_date and @selectd\_vehicle = t.vehicle\_id

        order by trip\_date asc

    end

else if @select\_user is not null and @selectd\_vehicle is null   --  SCENARIO 3: Date and User parameters included but No Vehicle

    begin

        print '3'

        select cast(u.users\_last\_name+', '+u.users\_first\_name as char(20)) 'Full Name',CONVERT(VARCHAR(10), t.trip\_date,101) 'Trip Date', t.trip\_destination 'Trip Destination',

        t.trip\_purpose 'Trip Purpose', t.trip\_total\_miles 'Trip Total Miles', cast (v.vehicle\_year+ ' '+v.vehicle\_make+ ' ' +v.vehicle\_model+ ' ' +v.vehicle\_plate as varchar(50)) 'Vehicle'

        from trip t join vehicle v on t.vehicle\_id=v.vehicle\_id join users u on (t.user\_id = u.user\_id)

        where trip\_date between @trip\_beginning\_date and  @trip\_ending\_date and @select\_user = u.user\_id

        order by trip\_date asc

    end

    else   -- SCENARIO 4: Both User and Vehicle parameters included with Date parameters

        begin

        print '4'

        select cast(u.users\_last\_name+', '+u.users\_first\_name as char(20)) 'Full Name',CONVERT(VARCHAR(10), t.trip\_date,101) 'Trip Date', t.trip\_destination 'Trip Destination',

        t.trip\_purpose 'Trip Purpose', t.trip\_total\_miles 'Trip Total Miles', cast (v.vehicle\_year+ ' '+v.vehicle\_make+ ' ' +v.vehicle\_model+ ' ' +v.vehicle\_plate as varchar(50)) 'Vehicle'

        from trip t join vehicle v on t.vehicle\_id=v.vehicle\_id join users u on (t.user\_id = u.user\_id)

        where @select\_user = u.user\_id and @selectd\_vehicle = t.vehicle\_id and trip\_date between @trip\_beginning\_date and  @trip\_ending\_date

        order by trip\_date asc

end

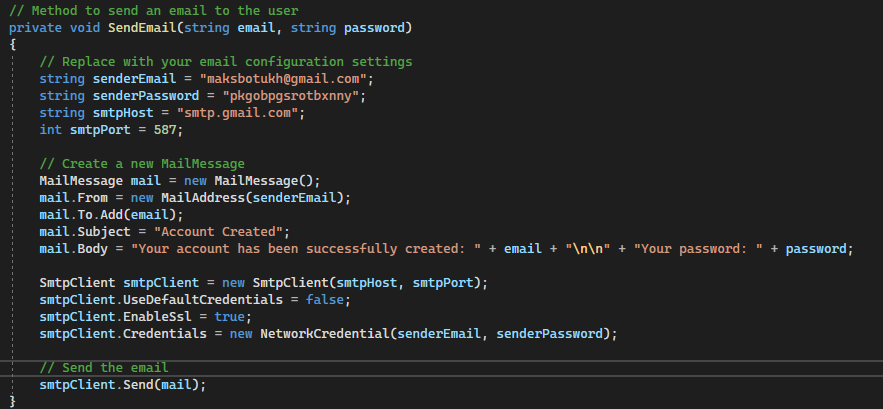
GO

# SMTP Modification (Sender Email)

To modify the sender email from the application, you would need to find the SendEmail statements through the application, which are located in the following screens of the application:

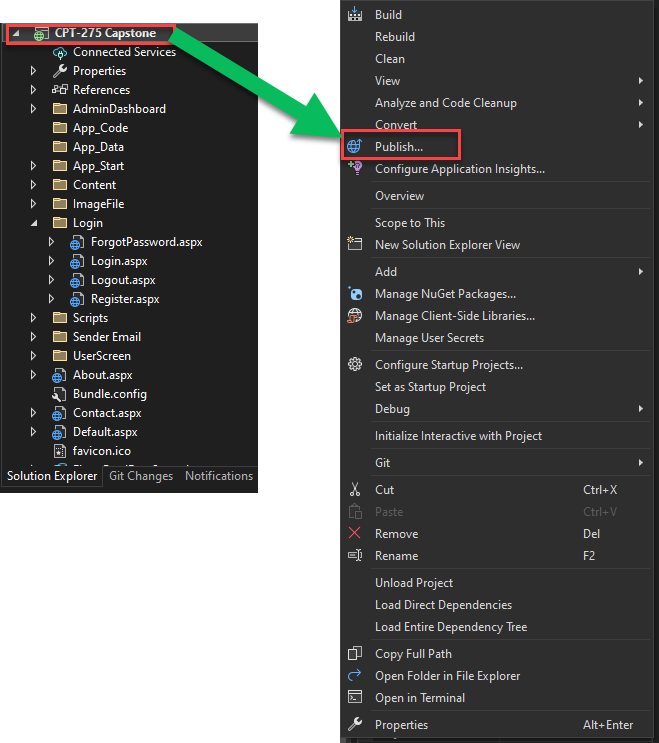
* AdminDashboard Folder:
  + AddUser.aspx.cs
* Login Folder
  + ForgotPassword.aspx.cs
  + Register.aspx.cs

You will see the following statements in which you will need to enter the senderEmail, Senderpassword(password of the SMTP email to allow send out an emails), SmtpHost, and SMTP Port.

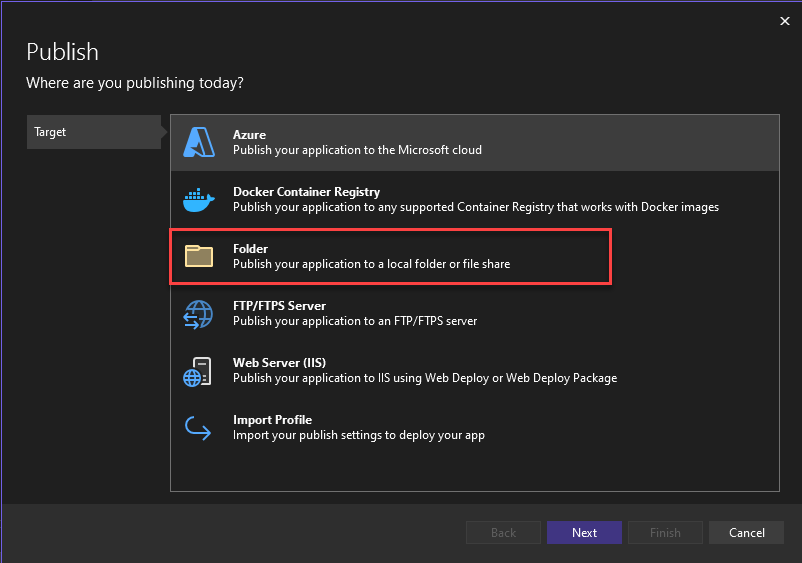


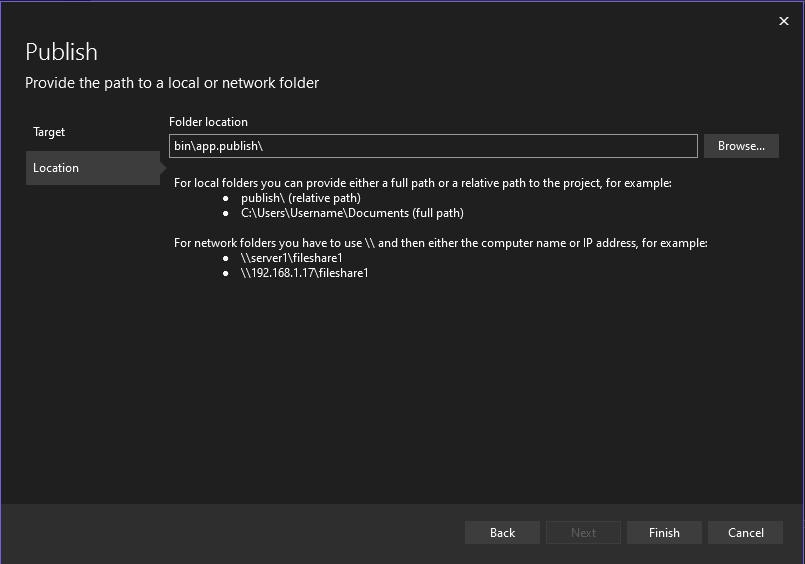
# Publish the project and promoting on the server

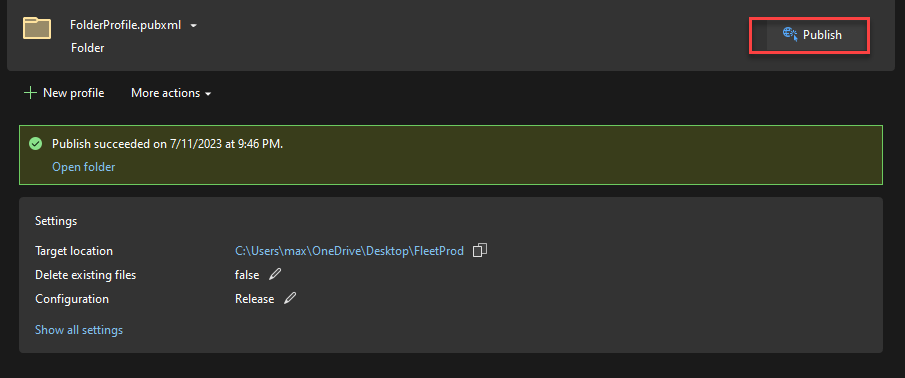
To publish the project from the visual studio you would need to right click the “CPT-275 Capstone” project in the Microsoft Visual Studio and click on Publish.



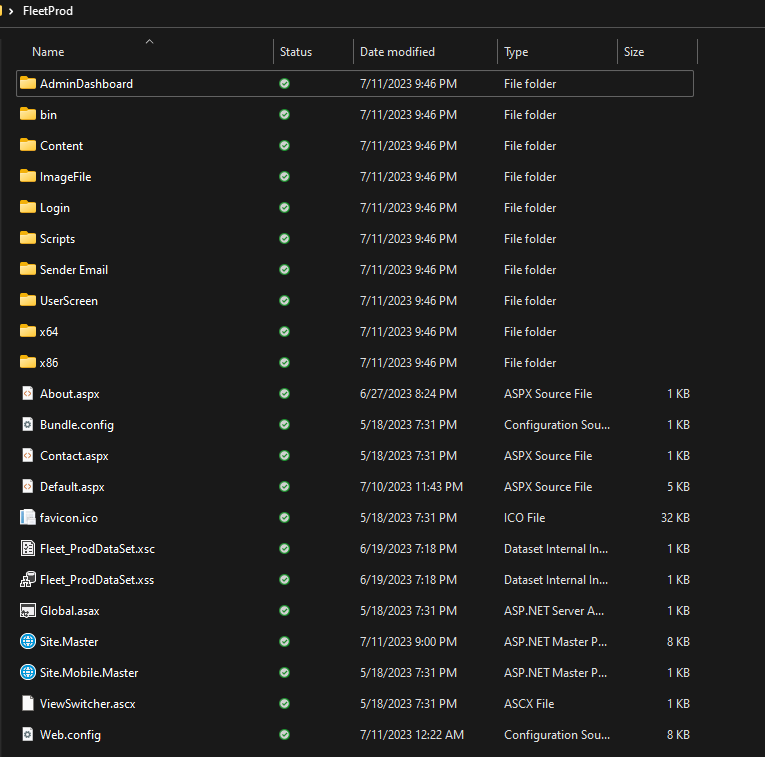
Once you clicked on the Publish button, you will select to publish to the Folder and point the location of where the folder will be located on your computer.





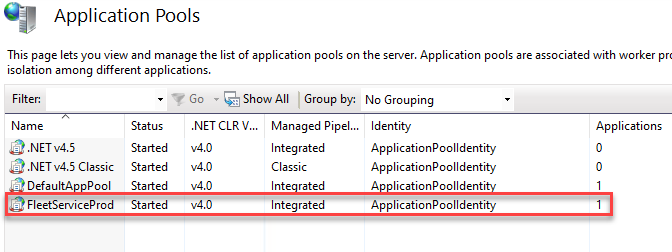


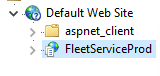
Once the publication is completed, the application can be found in the destination folder.

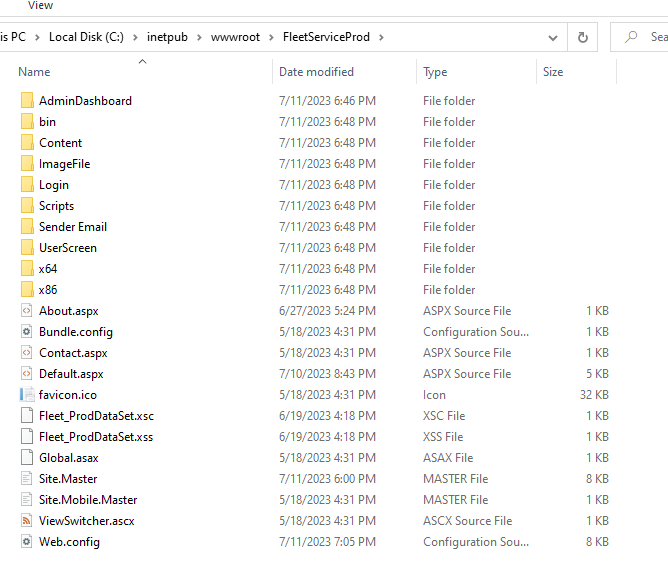


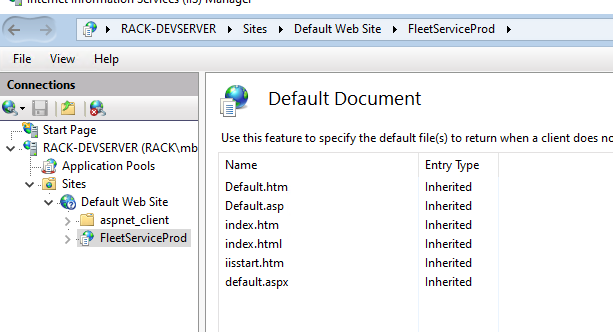
### Promoting to the server

When the application is published on the local computer it needs to be copied to the IIS on the server and applied to the Default Web Site, which should be associated to the AppPool of the application to run. The application can use the application identity and v 4.0 of .Net









# Setting up an Event Viewer Log

The application uses the event viewer log to store all the errors and for easy troubleshooting if any issues occur.

Execute the following script in the PowerShell on the server where the application is hosted.

New-EventLog -source SCCFleetServices -LogName SCCFleetServiceEventLog

*For any questions associated with this Installation guide,*

*please contact us at via maksbotukh@gmail.com*