# Overview

This project is a Blazor WebAssembly application designed to manage information about farmers and their products. It includes functionalities for different user roles (Farmers and Employees) and integrates various modern development practices and tools.

# Development Environment Setup

## Prerequisites

.NET SDK 8.0 or higher: Download and install from [here](https://dotnet.microsoft.com/en-us/download).

Visual Studio 2022: Ensure you have the Blazor WebAssembly template installed.

Service Based Database: Local Database.

## Steps

1. Clone the Repository:  
   <https://github.com/ST10134012/ST10134012-AgriConnect>

Opening from VCLearn

1. Download the project and unzip
2. If there is a 0x80010135 file path too long error, shorten the name of the zipped project e.g. proj or agri.

# Building and Running the Prototype

## Build the Project

1. Open the solution in Visual Studio.
2. Restore NuGet packages by right-clicking on the solution and selecting "Restore NuGet Packages".
3. Build the solution by pressing Ctrl+Shift+B.

## Setup Database

1. In the solution open AgriConnect\AgriConnect\AgriConnect.Infrastructure\Migrations\AgriConnect.mdf.
2. In Server Explorer, under Data Connections right-click on AgriConnect.mdf and select Properties.
3. Copy the Connection String.
4. Open apssettings.json in AgriConnect.WebApp and update the two connection strings.

## Install Dependencies

1. Go to Tools then NuGet Package Manager and open the Package Manager Console
2. Enter the following commands: ‘***dotnet restore***’ then ‘***Add-Migration InitialCreate***’ then ‘***Update-Database -Context ApplicationDbContext***’ then ‘***Update-Database -Context AuthenticationDbContext***’.
3. Rebuild the project.

## Run the Project

1. Set the AgriConnect.WebApp project as the startup project.
2. Run the application by pressing F5 or clicking the "Run" button.

# System Functionalities

## Farmers

* Register and Login: Farmers can register a profile and login
* Add Products: Farmers can add new products with details such as name, category, and production date.
* View Products: Farmers can view a list of their own products.

## Employees

* Register and Login: Employees can register a profile and login
* Add Farmer Profiles: Employees can add new farmer profiles with essential details.
* View and Filter Products: Employees can view and filter a comprehensive list of products from any farmer based on criteria like date range and product type.

## User Roles

**Farmer**

1. Can add products to their profile.
2. Can view their own product listings.

**Employee**

1. Can add new farmer profiles.
2. Can view all products from specific farmers.
3. Can use filters for product searching.

# Technologies Used

Blazor WebAssembly: For building interactive client-side web UIs with .NET.

Entity Framework Core: For database interactions and migrations.

Service Based Database: For relational data storage.

Dependency Injection: To manage dependencies throughout the application.

Caching: To improve performance by reducing database calls.

MediatR: For handling commands, queries, and events.

Repositories: To abstract data access logic.

Exception Handling: To manage errors gracefully.

# Additional Features

Responsive Design: Ensures the application is accessible on desktops, tablets, and smartphones.

Data Validation: Ensures the accuracy and consistency of information entered.

Error Handling: Prevents system crashes and data corruption.