

README

Matthew Allison

ST10269378

PROG7311 POE PART 2

Agri-Energy Connect

Link to Repo

<https://github.com/Sir-Goose/st10269378-matthew-allison-prog7311-poe-part-2/>

Dependencies

- [.NET 9 SDK](#)

Getting Started

1. Clone the repository

```
git clone https://github.com/Sir-Goose/st10269378-matthew-allison-prog7311-poe-part-2
cd st10269378-matthew-allison-prog7311-poe-part-2
```

2. Restore dependencies

```
dotnet restore
```

3. Build the project

```
dotnet build
```

4. Run the application

```
dotnet run
```

The application will start and be accessible at `https://localhost:<some port>` depending on your machine.

5. Using the App

- Visit the home page. Click the **Login** button to log in as either an Employee or Farmer.
- Employees can manage farmers and view/filter all products.
- Farmers can add products and view their own products.

6. Example Login Credentials

Employees

- **Email:** clive.frankland@example.com
Password: password1
- **Email:** matthew.allison@example.com
Password: password2

Farmers

- **Email:** joshua.shields@example.com
Password: boer123
- **Email:** kashvir.sewpersad@example.com
Password: plaas456
- **Email:** erin.steenveld@example.com
Password: landbou789
- **Email:** luke.carolus@example.com
Password: groente321
- **Email:** rudolf.holzhausen@example.com
Password: vars654

7. Database

- The app uses SQLite (`app.db`) and will automatically create and seed the database on first run.
- Default users are seeded for both employees and farmers.

Project Structure

- `Controllers/` - MVC controllers
- `Models/` - Data models
- `Services/` - Business logic
- `Repository/` - Data access
- `Views/` - Razor views (UI)
- `wwwroot/` - Static files (CSS, JS, images)

Troubleshooting

- Make sure you have the correct .NET SDK version installed.
- If you change the database schema, delete `app.db` to reset.

References

- ardalis (2024). Overview of ASP.NET Core MVC. [online] Microsoft.com. Available at: https://learn.microsoft.com/en-gb/aspnet/core/mvc/overview?view=aspnetcore-9.0&WT.mc_id=dotnet-35129-website.
- BillWagner (2023). Interfaces - define behavior for multiple types. [online] learn.microsoft.com. Available at: <https://learn.microsoft.com/en-us/dotnet/csharp/fundamentals/types/interfaces>.
- bricelam (2020). SQLite Database Provider - EF Core. [online] learn.microsoft.com. Available at: <https://learn.microsoft.com/en-us/ef/core/providers/sqlite/?tabs=dotnet-core-cli>.
- Rick-Anderson (2022). HTTP Cookies in ASP.NET Web API - ASP.NET 4.x. [online] learn.microsoft.com. Available at: <https://learn.microsoft.com/en-us/aspnet/web-api/overview/advanced/http-cookies>.
- Rick-Anderson (2025). Getting Started - EF Core. [online] learn.microsoft.com. Available at: <https://learn.microsoft.com/en-us/ef/core/get-started/overview/first-app?tabs=netcore-cli>.
- StephenWalther (2022). ASP.NET MVC Controller Overview (C#). [online] learn.microsoft.com. Available at: <https://learn.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/controllers-and-routing/aspnet-mvc-controllers-overview-cs>.
- tdykstra (2024). Introduction to Razor Pages in ASP.NET Core. [online] Microsoft.com. Available at: <https://learn.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-9.0&tabs=visual-studio>.
- W3Schools (2019). HTML CSS. [online] W3schools.com. Available at: https://www.w3schools.com/html/html_css.asp.

AI Decleration

AI tools were used to assist with completing the project. Code was not directly copy pasted but rather AI was used to inform on how to do specific things such as only displaying certain html elements on a razor page under certain conditions.

Below are screenshots of chats with AI models. All AI usage was with GPT-4.1 and GPTImageGen

















