**Agri-Energy Connect Project Readme File**

**Overview**

The Agri Energy Connect project is a website created to link the farming industry with providers of eco energy technology, in South Africa. It aims to be easy to use for a range of users regardless of their technical knowledge. The platform enables real time sharing of data and collaboration all while prioritizing the protection and confidentiality of user information.

**Architecture and Technologies**

The architecture implemented is Clean Architecture integrating the Model View Controller (MVC) design pattern, in the web application layer. This approach simplifies maintenance, testing and the ability to scale the application.

**Key Technologies and Patterns**

1. **.NET 8**: The project is built using .NET 8, leveraging its latest features and improvements.
2. **Microsoft Identity**: For authentication and authorization, Microsoft's Identity framework has been used to handle user roles and access control securely.
3. **SQL Server on Azure**: The application uses SQL Server hosted in Azure for data storage, providing robust and scalable database solutions.
4. **CQRS with MediatR**: To promote coupling between components I implemented the Command Query Responsibility Segregation (CQRS) pattern using MediatR. This separation of read and write operations enhances performance and scalability.
5. **Transactional Outbox Pattern**: To ensure the successful delivery of events, Transactional Outbox Pattern was implemented. This pattern helps in reliably sending messages to other services or components.
6. **Quartz.NET**: Quartz.NET manages background jobs efficiently by executing queued events to ensure task processing.
7. **Pipeline Behaviours in MediatR**: MediatR's pipeline behaviour feature was used to implement the Decorator pattern. This allowed us to handle side events like logging and validation, in an efficient manner.

**Getting Started**

**Development Environment Setup**

To set up the development environment and run the application, follow these steps:

1. **Download and Install Visual Studio**: Visit the [Visual Studio website](https://visualstudio.microsoft.com/downloads/) to download and install Visual Studio.
2. **Clone the Repository**: Clone the Agri-Energy Connect repository to your local machine using Git. [Tashiee/AgriEnergyConnect: PROG7311 PART 2 (github.com)](https://github.com/Tashiee/AgriEnergyConnect/tree/master)
3. **Open Solution in Visual Studio**: Open the solution file (AgriEnergyConnect.sln) in Visual Studio.
4. **Build Solution**: Build the solution to ensure all dependencies are resolved.
5. **Run the Application**: Press F5 or click on the **Start** button to run the application.
6. **Login**: Once the application starts, you should see the welcome page in your default web browser. Log in as either an employee or a farmer to access the respective functionalities.

**Setting up the Database**

**Option 1: Using Existing Database**

If you prefer to use the existing database configuration without making any changes, follow these steps:

1. Open the solution in Visual Studio.
2. Build the solution to ensure all dependencies are resolved.
3. Press F5 or click on the **Start** button to run the application.
4. Once the application starts, you should see the welcome page in your default web browser.

**Option 2: Configuring a New Database**

If you want to connect to a different database, follow these steps:

1. **Configure Connection String**: Open the appsettings.json file in the project's root directory. Locate the ConnectionStrings section and update the DefaultConnection string with your SQL Server connection details. Replace the placeholder values with your SQL Server instance name, database name, username, and password.
2. **Run Migrations**: Open a command prompt or terminal in the project's root directory and run the following commands to apply the database migrations:

dotnet ef database update --context ApplicationDbContext

dotnet ef database update --context AppDataContext

This will create the necessary tables in the database and apply the authentication and data migrations.

1. **Verify Database Creation**: Once the migrations are applied successfully, verify that the agriDb database has been created in your SQL Server instance.

**System Functionality and User Roles**

Upon logging into the platform users will see a page where they can choose to log in as either a farmer or an administrator. Farmers can manage their products while administrators have access to features, for handling farmer profiles and product listings.

* Farmers:
* Farmers can set up profiles add new products and view or edit their existing products.
* They can explore a marketplace to find compare and buy eco energy solutions designed for agricultural use.
* Administrators:
* Administrators can oversee farmer profiles and can add new farmer to the database.
* filter a list of products from any farmer based on criteria like date range and product. And they can filter the product by clicking a filter sign that will be on the headings
* handle tasks like user management and content moderation.

**Contributing**

I welcome contributions from the community. Feel free to open issues, suggest enhancements, or submit pull requests.

**License**

This project is licensed under the MIT License - see the [LICENSE](LICENSE) file for details.