**FarmLink**

In Bangladesh, farmers work tirelessly to grow crops, yet when it comes to selling them, they are often trapped by powerful **market syndicates**. These groups control supply chains, raise prices unfairly, and leave farmers earning only a fraction of what their produce is worth. Meanwhile, consumers and small shop owners pay inflated prices without realizing the struggle behind the market.

**FarmLink** is a C# .NET-based platform designed to break this cycle. It directly connects **farmers** with **everyday buyers** and **local shops**, allowing farmers to list their products and sell without relying on exploitative middlemen. Buyers can easily browse and purchase fresh goods, ensuring farmers get a fair price while consumers enjoy more affordable options.

However, farmers often face challenges when selling in bulk - they may not have the time, skills, or resources to negotiate large deals. This is why **agents** are introduced into the system. Approved by an admin, agents help farmers sell large quantities of products to big buyers or shop owners and, in return, earn a fair **commission**. This ensures that even bulk goods move quickly and farmers focus on what they do best -farming.

The **admin** oversees agent approvals and major transactions, maintaining the platform’s trust and transparency.  
Through FarmLink, we aim to **empower farmers**, **reduce market manipulation**, and **build a fairer, more direct agricultural market** for Bangladesh.

**Case Study**

In the FarmLink system, multiple types of users interact with the platform: **farmers**, **buyers** (which include both regular customers and shop owners), **agents**, and **an admin**. A **farmer** can register on the platform, log in, and **list multiple products** for sale (e.g., tomatoes, potatoes, rice). Each **product** contains details like name, category, quantity, price per unit, and availability. Buyers can browse and **place orders** for one or more products, creating a relationship between buyers and products through an **order table**. An order stores buyer ID, order date, delivery status, and references to the products purchased.

Sometimes, a **farmer may assign an agent** to help sell their products. When this happens, the **agent gets linked to the product**, and if that product is sold through the agent, a **commission percentage** is recorded. Every **agent** must be approved by the **admin**, who verifies their information. The admin is also responsible for approving **high-value orders** and monitoring the platform.

In summary, the key entities are **Farmer**, **Buyer**, **Product**, **Order**, **Agent**, and **Admin**, with relationships such as:

* A **Farmer has many Products**
* A **Buyer places many Orders**, and each Order has one or more **Products**
* An **Agent may help sell multiple Products**, and **Products may or may not involve an Agent**
* The **Admin approves Agents** and can approve **high-value Orders**

This structure ensures transparency, scalability, and direct engagement between farmers and the market.