Project Proposal

**Digital Farm Management System**

**OBJECT ORIENTED PROGRAMMING**

|  |  |
| --- | --- |
| **INSTRUCTOR** | MS. Bushra Mehmood |
| **COURSE CODE** | CMC 112 |
| **SEMESTER** | 2nd |
| **CREDIT HOURS** | 3 (Theory) + 1 (Lab) |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Group Members/Student Name** | **Registration Number** | **Project Title** |
| Muhammad Noman | 42124 |  |
| Muhammad Illyas | 42105 | Digital Farm Management System |
| Hamza Siddique | 39912 |  |
|  |  |  |

# **Project Summary**:

The Digital Farm Management System (DFMS) is a C++ project that applies Object-Oriented Programming concepts such as encapsulation, inheritance, and polymorphism to simulate the daily tasks of managing a farm. Users can care for animals, plant and water crops, assign tasks to workers, and buy or sell products in a dynamic market environment. Each simulated day influences the health and productivity of the farm, encouraging users to balance resources and plan effectively.

To make the system more user-friendly and visually engaging, a Graphical User Interface (GUI) is integrated into the project. This GUI enhances navigation and provides visual feedback, helping beginner programmers better understand the link between programming concepts and real-world applications through interactive elements.

By combining practical simulation, strategic decision-making, and visual interaction, DFMS offers an engaging and educational experience. It serves as a valuable tool for teaching programming fundamentals, making abstract OOP principles easier to grasp through a fun and realistic farming scenario.

# Introduction:

Digital Farm Management System is built using C++ and using Object Oriented Programming concepts. It’s a simulation of a digital farm where user can manage animals, grow crops, assign tasks to workers for better efficiency, check stock level and finally to make this look like it’s based on real farm, this project also has market feature where user can buy and sell stocks accordingly. The main goal of this code is to apply OOP features like encapsulation, inheritance, polymorphism, and finally abstraction to create a real-life farm simulation.

This project is mainly divided into three main sections: **Animal Management**, **Crop Management**, and **Worker Management**. Users can add animals like cow or chicken, check their health to see if they aren’t at the risk of dying, collect products like milk and eggs, which the user can sell too in the market. Users can plant crops like wheat and corn, which can help in feeding animals or sell them based on their need.

To make this look more realistic, simulation of new day is added. This feature is important as it helps crops grow with time and also animals need to be fed each day. Workers are also added which can help in different task regarding animals or crops, like feeding animals or watering crops, so the crops don’t die. So basically, everything is directly linked with each other.

# Problem Statement:

Running a farm isn’t easy, it takes a lot of work to feed animals, grow crops, manage workers, and make sure everything is running smoothly. In real life, all these things are connected, and if one part fails, the whole system can suffer.

It can be tough to understand how concepts like encapsulation or inheritance actually work in real scenarios. That’s where this project comes in. The **Digital Farm Management System** is a fun and interactive way to bring those OOP concepts to life. It lets users manage a virtual farm where they can take care of animals, grow crops, assign tasks to workers, and even buy or sell items in a market. Everything is linked just like in a real farm, making it a great hand on the learning experience.

# Proposed Solution:

To solve the problem of understanding and applying Object-Oriented Programming in a meaningful way, this project introduces a **Digital Farm Management System** a complete farm simulation built using C++ and OOP concepts.

The system allows users to:

* **Manage animals** like cows and chickens by feeding them, checking their health, and collecting products like milk and eggs.
* **Grow crops** such as wheat and corn that can be used to feed animals or sold in the market.
* **Assign tasks to workers**, who help by feeding animals or watering crops, improving overall efficiency.
* **Simulate daily progress**, where a new day affects animal health, crop growth, and overall farm productivity.
* **Buy and sell items** through a market system, making the experience more realistic and engaging.

# Simulation:

The DFMS simulates a real farm environment by allowing users to manage animals, crops, and workers through daily activities. Each "day" in the simulation represents a cycle where:

* Time in the farm passes in daily cycles (each day is simulated).
* Animals must be fed daily to maintain their health and productivity.
* If animals are not fed, their health decreases and they risk dying.
* Healthy animals produce products like milk and eggs that can be collected.
* Crops need to be watered regularly to grow properly.
* Crops take several days to grow and must be harvested when ready.
* Harvested crops can be used to feed animals or sold in the market.
* Workers assist in tasks like feeding animals and watering crops to keep the farm running smoothly.
* Users manage the balance between animals, crops, and resources to keep the farm successful.
* The market allows users to buy and sell farm products like milk, eggs, wheat, and corn.
* Prices and available stock in the market change daily, simulating real market fluctuations.
* Selling products helps earn money, and buying resources supports farm operations.
* The simulation models real farm management challenges in an interactive way.

# Conclusion:

This Digital Farm Management System project successfully simulates real-life farm activities using object-oriented programming in C++. It allows users to manage animals, grow and water crops, assign tasks to workers, and participate in a dynamic market for buying and selling products. By linking all these parts together, the system creates an engaging and realistic farm experience. This project not only helps users understand farm management but also demonstrates important programming concepts like encapsulation, inheritance, polymorphism, and abstraction in a practical way.