**CHAPTER FOUR**

**SYSTEM IMPLEMENTATION**

**4.1 Chapter Overview**

This chapter presents the implementation details of the simulated ammonia monitoring system developed for freshwater aquaculture. It builds upon the architectural design outlined in Chapter Three and demonstrates how each module—sensor emulation, microcontroller logic, communication protocol, and dashboard interface—was realized using virtual components and software routines.

**4.2 Features and Choice of Implementation Technology**

The system was implemented using **Python** for GUI development and **C++/Arduino-style pseudocode** to simulate ESP32 firmware logic. Python’s Tkinter library was selected for its lightweight interface capabilities, while Modbus register emulation was handled using structured data arrays. The choice of these technologies was driven by their accessibility, flexibility, and suitability for simulation-based testing.

**4.3 System Testing Strategies**

**4.3.1 Unit Testing**

Each module was tested independently:

* **Sensor Emulation**: Verified register output consistency across simulated NH₄⁺ ranges.
* **ESP32 Logic**: Validated Modbus polling routines and threshold evaluation.
* **Dashboard GUI**: Confirmed correct parsing and display of incoming packets.

**4.3.2 Integration Testing**

The full system was tested as a pipeline:

* Synthetic sensor data was injected into the converter logic.
* ESP32 simulation parsed and transmitted data via TCP/IP.
* Dashboard received and visualized readings in real time.

**4.4 Target Computer System Requirements**

| **Hardware Requirements** | **Specification** |
| --- | --- |
| **Processor** | **Intel Core i3 or higher** |
| **RAM** | **Minimum 4 GB** |
| **Storage** | **500 MB free disk space** |
| **Display** | **1366 × 768 resolution or higher** |

| **Software Requirements** | **Specification** |
| --- | --- |
| Operating System | Windows 10 / Linux / macOS |
| Programming Environment | Python 3.8+, Arduino IDE (for simulation) |
| Libraries | Tkinter, PyModbus, Socket, Matplotlib |

**4.5 Results**

The system successfully simulated ammonia readings across safe and hazardous thresholds. Alerts were triggered when NH₄⁺ levels exceeded 5 mg/L, and the dashboard updated dynamically. Sample outputs included:

* Real-time graphs of NH₄⁺ concentration
* Alert pop-ups for threshold breaches
* Logged data entries with timestamps

**4.6 Discussion**

The simulation demonstrated reliable data flow from virtual sensor to dashboard. Modbus emulation proved effective for representing RS-485 communication, and the GUI responded accurately to changing input values. While the system is virtual, its architecture mirrors deployable field systems, making it a valid prototype for future hardware integration.