RAJALAKSHMI
ENGINEERING
COLLEGE

AN AUTONOMOUS INSTITUTION

CHENNAI

BATCH 2025-2026

🌾SMART FARMING MONITOR

A MINI-PROJECT REPORT

SUBMITTED BY

KAMALI RJ	**240701225**
KANIJA FATHIMA J	**240701226**

in partial fulfillment of the award of the degree of

BACHELOR OF ENGINEERING IN

COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

Certified that this project **“SMART FARMING MONITOR”** is the bonafide work of **“KAMALI RJ , KANIJA FATHIMA J”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. V. JANANEE	**Ms. M . ANITHA**
ASSISTANT PROFESSOR SG	**ASSISTANT PROFESSOR SG**

Dept. of Computer Science and Engg,
Rajalakshmi Engineering College
Chennai

This mini project report is submitted for the viva voce examination to be held on

____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

Agriculture plays a crucial role in our economy , but many farmers still rely on manual methods to manage crop information. The **Smart Farming Monitor** project is developed to overcome this limitation by providing a digital platform for efficient crop management. Built using **Java** for the frontend and **MySQL** for the backend, the system allows users to log in, add, view, and organize crop details easily. It helps maintain accurate farming records, improves data accessibility, and supports better decision-making. This project encourages the use of technology in agriculture, promoting smarter and more organized farming practices.

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal Dr. S.N. MURUGESAN for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide Dr. V. JANANEE ,and Ms. M. ANITHA for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. KAMALI RJ

2. KANIJA FATHIMA J

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The project helps farmers and users to manage and monitor crop information efficiently through a computer-based system. It provides an organized way to store, view, and update farming data such as crop names, seasons, and growth details. The system offers a user-friendly interface that allows users to easily access and manage agricultural information, improving accuracy and saving time in record maintenance.

1.2 SCOPE OF THE WORK

The Smart Farming Monitor aims to digitalize the management of farming activities using a Java-based frontend and a MySQL backend. It helps farmers store essential crop data and retrieve it whenever needed. The system enhances the overall efficiency of farming operations by reducing manual errors and maintaining accurate records. It can be used by individual farmers,

agricultural officers, and organizations for better decision-making and farm management.

1.3 PROBLEM STATEMENT

In traditional farming, most data related to crops and agricultural activities are recorded manually, leading to errors, data loss, and inefficiency. Farmers often struggle to maintain accurate records or retrieve past information when needed. The lack of a digital platform for managing crop data causes difficulties in organizing and analyzing farming information effectively. Hence, there is a need for a simple, reliable system to store and manage farming data digitally.

1.4 AIM AND OBJECTIVES OF THE PROJECT

The main aim of this project is to develop a computerized system that helps in storing,organizing, and retrieving crop-related information efficiently. The objectives include:

- **To design a user-friendly interface using Java for easy access and data entry.**
- **To store and manage farming information securely in a MySQL database.**
- **To allow users to add, view, and edit crop details conveniently.**
- **To promote efficient record management and support better agricultural decision-making.**

CHAPTER 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Component	**Specification**
Processor	Intel Core i5 or higher
Memory Size	8 GB RAM (Minimum)
Hard Disk	1 TB (Minimum)
Monitor	15” LED Display (Minimum)
Keyboard & Mouse	Standard Input Devices

2.2 SOFTWARE SPECIFICATIONS

Software Component	**Specification**
Operating System	Windows 10 / 11
Front-End	Java (Swing / JavaFX)
Back-End	MySQL Database
Programming Language	Java, SQL
IDE Used	Visual Studio Code / NetBeans
Database Connector	JDBC (Java Database Connectivity)

CHAPTER 3

MODULE DESCRIPTION

This application consists of a single main module that allows the user to log in and manage crop details. When the program runs, it opens with a login window where the user must enter valid credentials to access the system. Once logged in, the user can perform several actions such as adding, viewing, updating, and deleting crop information. The description of the module is given below:

---

1. User Login

When the user starts the application, they are prompted to log in using their username and password. After successful login, the user can access the main dashboard where they can manage crop details. The user is allowed to:

- Add new crop details such as crop name, type, and season.
- View all stored crop information in an organized manner.
- Update or delete existing crop records when required.

This module helps users to easily maintain and monitor agricultural data through a simple and efficient interface. The system ensures data security, easy accessibility, and better management of farming information.

CHAPTER 4

SAMPLE CODING

□Database Connection (DBConnection.java)

```
package com.smartfarming.util;
```

```
import java.sql.Connection;
import java.sql.DriverManager;


public class DBConnection {
    private static final String URL =
"jdbc:mysql://localhost:3306/smartfarming";
    private static final String USER = "root";
    private static final String PASSWORD = "";


    public static Connection getConnection() {
        Connection con = null;
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con = DriverManager.getConnection(URL, USER,
PASSWORD);
            System.out.println("✅ Database Connected Successfully");
        } catch (Exception e) {
            System.out.println("❌ Database Connection Failed: " + e.getMessage());
        }
        return con;
    }
}
```

2️⃣User Data Access Object (UserDAO.java)

```
package com.smartfarming.dao;


import com.smartfarming.util.DBConnection;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;


public class UserDAO {


    // Check login credentials
    public boolean validateLogin(String username, String password) {

        boolean status = false;

        try (Connection con = DBConnection.getConnection()) {

            String sql = "SELECT * FROM users WHERE username=?
AND password=?";

            PreparedStatement ps = con.prepareStatement(sql);

            ps.setString(1, username);

            ps.setString(2, password);

            ResultSet rs = ps.executeQuery();

            status = rs.next();

        } catch (Exception e) {

            e.printStackTrace();

        }

        return status;

    }



    // Add a new user (used in signup)
    public boolean addUser(String username, String email, String
password) {

        boolean result = false;

        try (Connection con = DBConnection.getConnection()) {

            String sql = "INSERT INTO users(username, email, password)
VALUES (?, ?, ?)";

            PreparedStatement ps = con.prepareStatement(sql);

            ps.setString(1, username);

            ps.setString(2, email);

            ps.setString(3, password);

            result = ps.executeUpdate() > 0;

        } catch (Exception e) {

            e.printStackTrace();

        }

        return result;

    }

}
```

3□Login UI (LoginUI.java)

```
package com.smartfarming.ui;

import com.smartfarming.dao.UserDAO;
import javax.swing.*;
import java.awt.*;

public class LoginUI extends JFrame {
    private JTextField txtUser;
    private JPasswordField txtPass;
    private UserDAO dao = new UserDAO();

    public LoginUI() {
        setTitle("Smart Farming Login");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
```

```
setLayout(new BorderLayout(10,10));



JLabel header = new JLabel("🌾 Smart Farming System", SwingConstants.CENTER);

header.setFont(new Font("Poppins", Font.BOLD, 18));

add(header, BorderLayout.NORTH);



JPanel panel = new JPanel(new GridLayout(3,2,10,10));

panel.setBorder(BorderFactory.createEmptyBorder(20, 50, 20, 50));



panel.add(new JLabel("Username:"));

txtUser = new JTextField();

panel.add(txtUser);



panel.add(new JLabel("Password:"));

txtPass = new JPasswordField();
```

```
    panel.add(txtPass);


    JButton btnLogin = new JButton("Login");

    JButton btnClear = new JButton("Clear");

    panel.add(btnLogin);

    panel.add(btnClear);


    add(panel, BorderLayout.CENTER);


    btnLogin.addActionListener(e -> login());

    btnClear.addActionListener(e -> {

        txtUser.setText("");

        txtPass.setText("");

    });


    setVisible(true);

}
```

```
    private void login() {
        String user = txtUser.getText();
        String pass = new String(txtPass.getPassword());

        if (dao.validateLogin(user, pass)) {
            JOptionPane.showMessageDialog(this, "✅ Login Successful");
            dispose();
            new DashboardUI(); // navigate to dashboard
        } else {
            JOptionPane.showMessageDialog(this, "❌ Invalid Credentials");
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new LoginUI());
    }

}
```

4□Dashboard Page (DashboardUI.java)

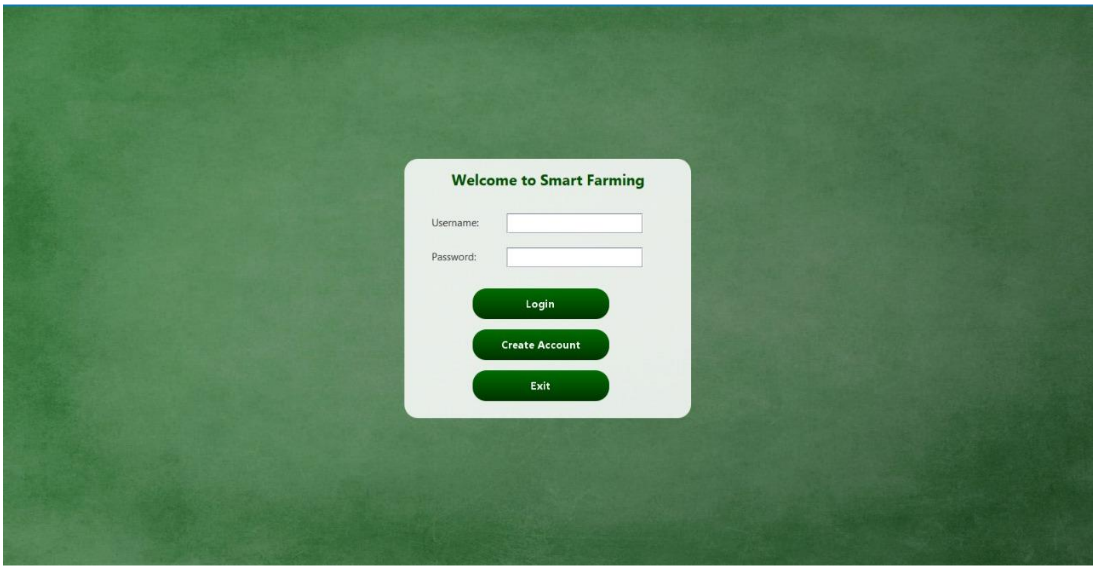
```
package com.smartfarming.ui;


import javax.swing.*;

import java.awt.*;


public class DashboardUI extends JFrame {


    public DashboardUI() {

        setTitle("Smart Farming Dashboard");

        setSize(600, 400);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setLocationRelativeTo(null);
        JLabel lbl = new JLabel("🌱 Welcome to Smart Farming Dashboard", SwingConstants.CENTER);

        lbl.setFont(new Font("Poppins", Font.BOLD, 18));

        add(lbl, BorderLayout.CENTER);



        setVisible(true);

    }

}
```

5️⃣MySQL Table

```
CREATE TABLE users (

    user_id INT AUTO_INCREMENT PRIMARY KEY,

    username VARCHAR(100) NOT NULL UNIQUE,

    email VARCHAR(150) NOT NULL UNIQUE,

    password VARCHAR(255) NOT NULL,

    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);
```

CHAPTER 5

SCREEN SHOTS

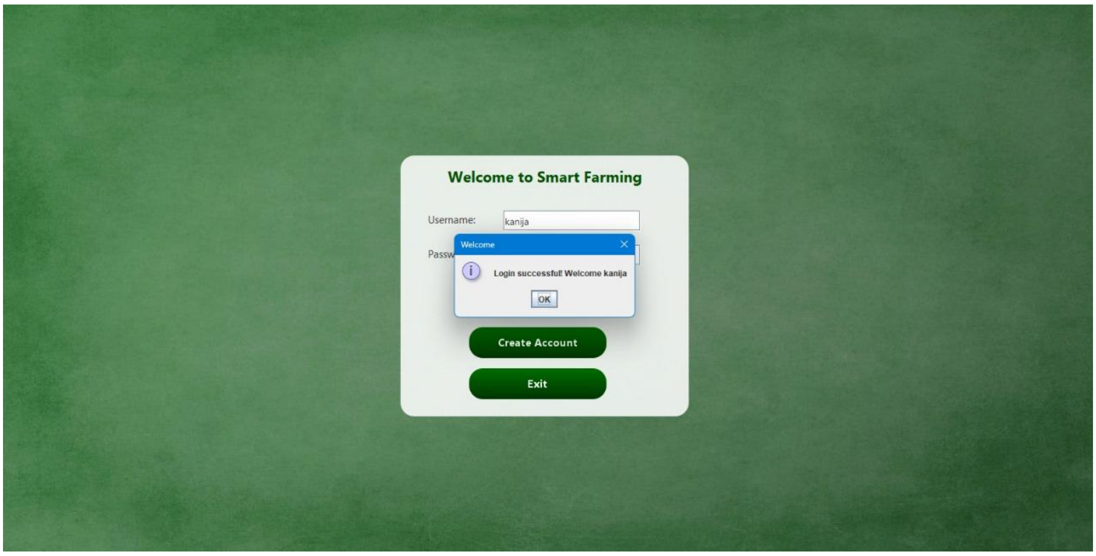


Welcome to Smart Farming

Username:

Password:

Fig 5.1 Introduction Page



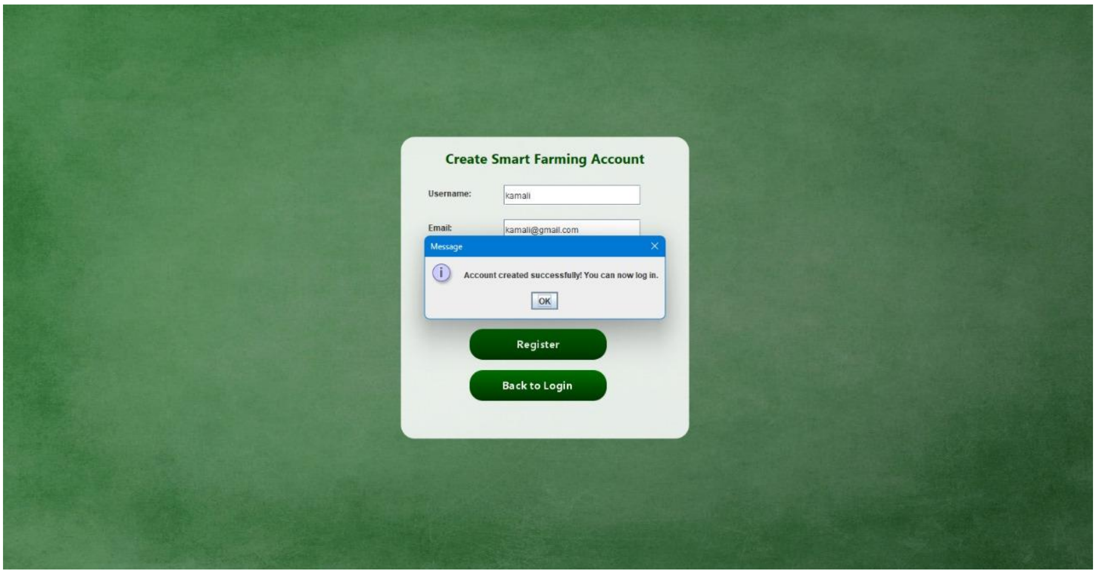
Welcome to Smart Farming

Username: kanija

Passw

Welcome

Login successful! Welcome kanija



Create Smart Farming Account

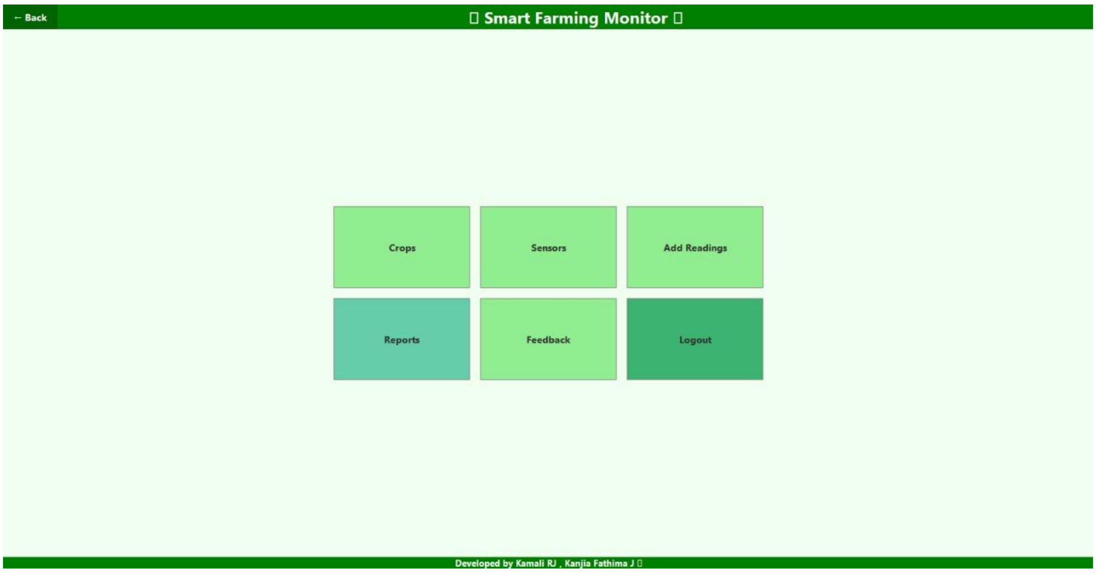
Username: kamali

Email: kamali@gmail.com

Message

Account created successfully! You can now log in.

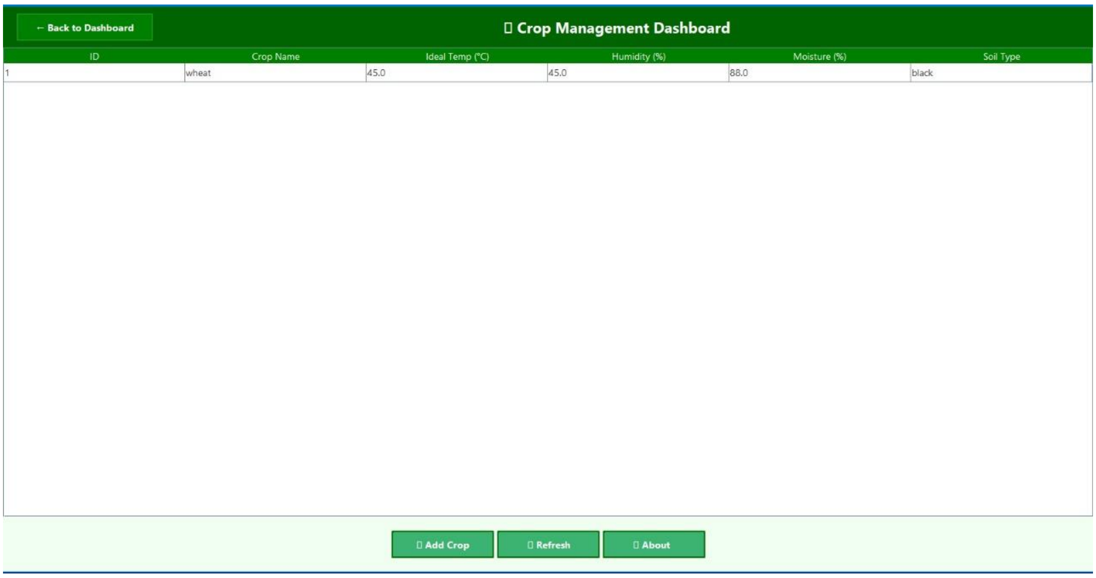
Fig 5.2 Login Page



□ Smart Farming Monitor □

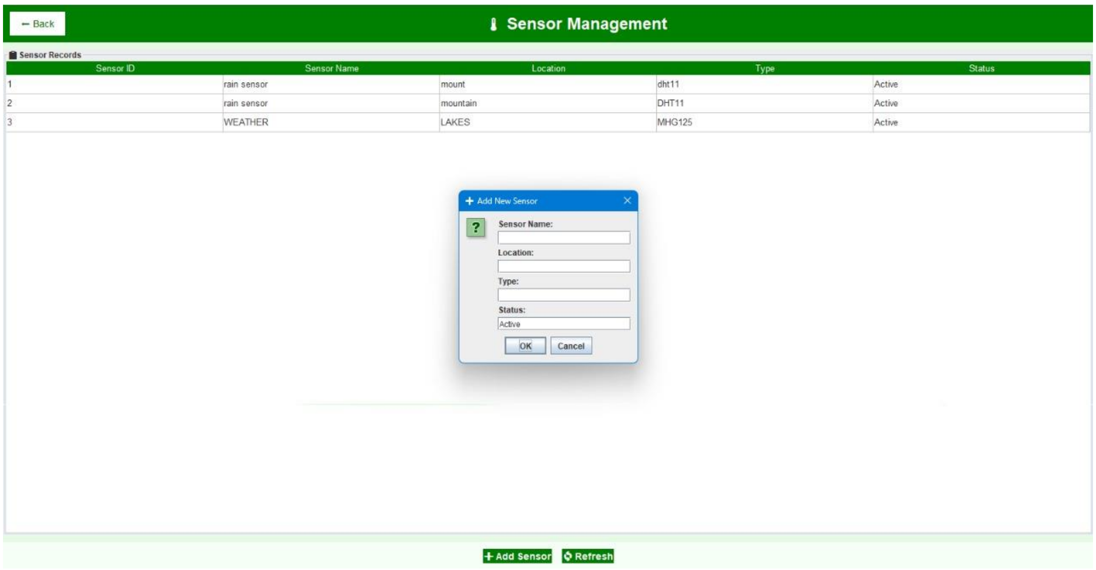
Developed by Kamali RJ , Kanjia Fathima J □

Fig 5.3 Dashboard Page



ID	Crop Name	Ideal Temp (°C)	Humidity (%)	Moisture (%)	Soil Type
1	wheat	45.0	45.0	88.0	black

Fig 5.4 Crop Dashboard Page



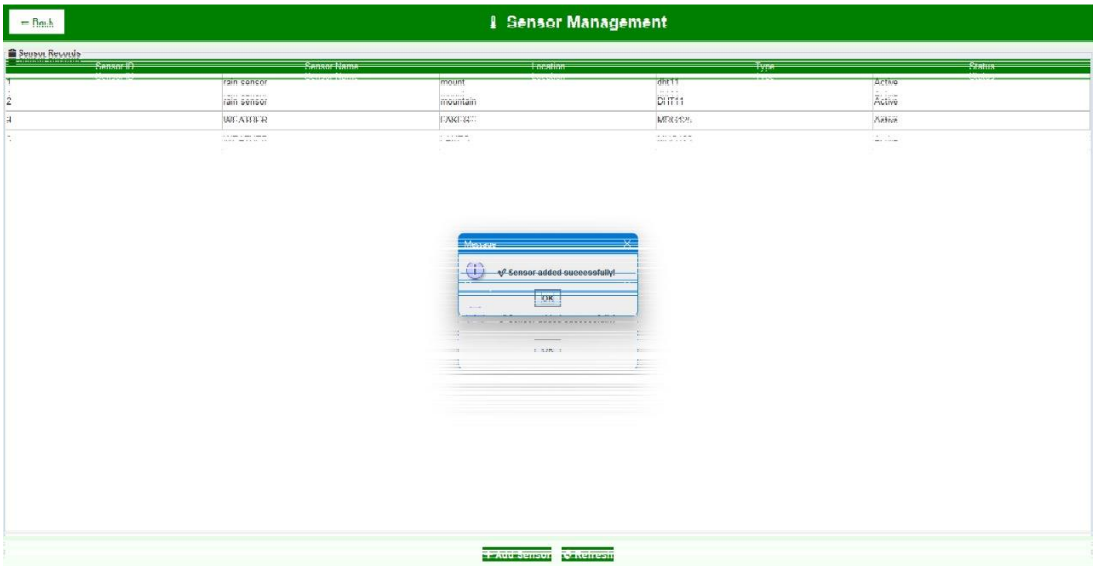
Sensor ID	Sensor Name	Location	Type	Status
1	rain sensor	mount	dht11	Active
2	rain sensor	mountain	DHT11	Active
3	WEATHER	LAKES	MHG125	Active

Sensor Name:

Location:

Type:

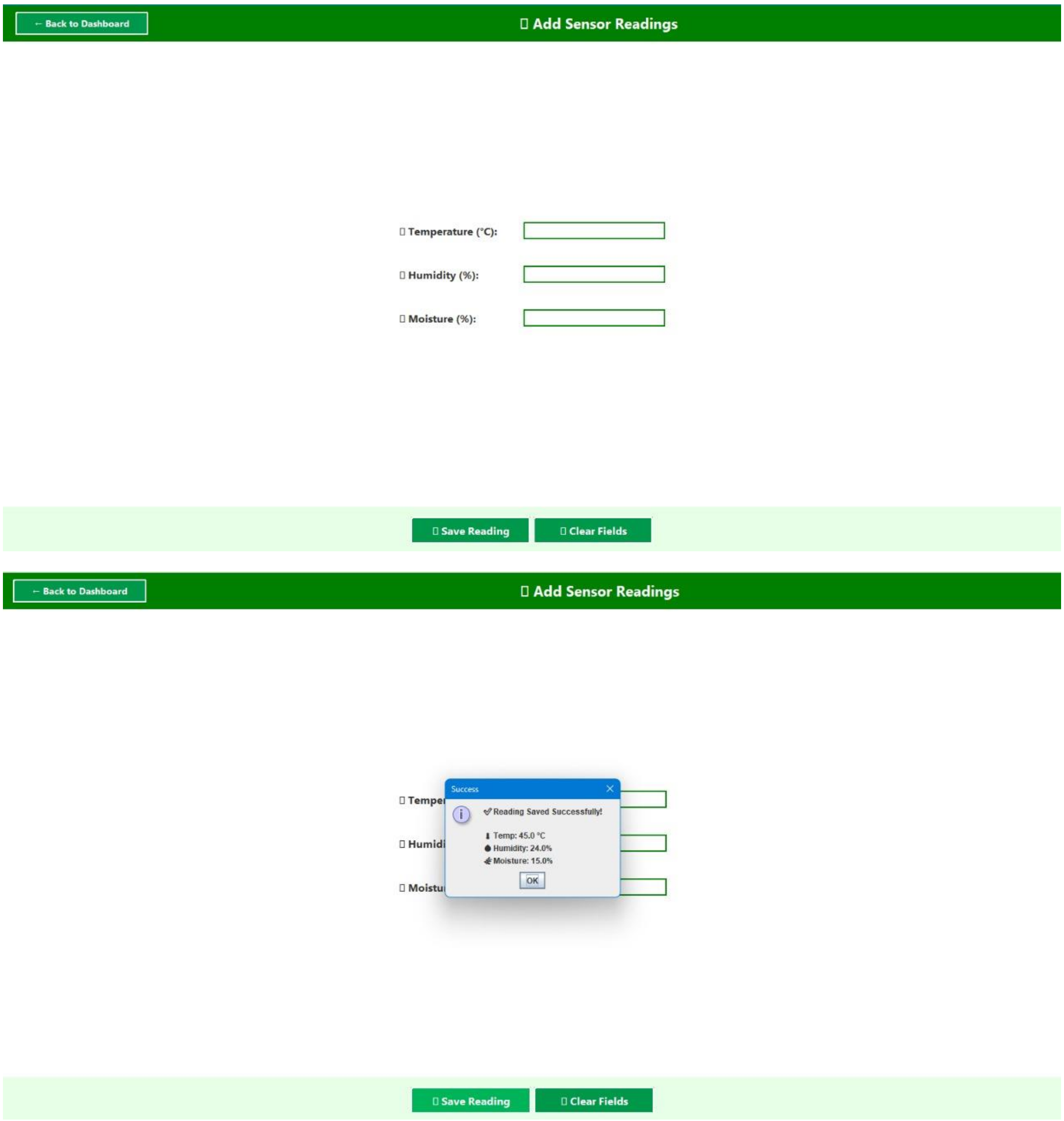
Status: Active



Sensor ID	Sensor Name	Location	Type	Status
1	rain sensor	mount	dht11	Active
2	rain sensor	mountain	DHT11	Active
3	WEATHER	[illegible]	[illegible]	Active

√ Sensor added successfully!

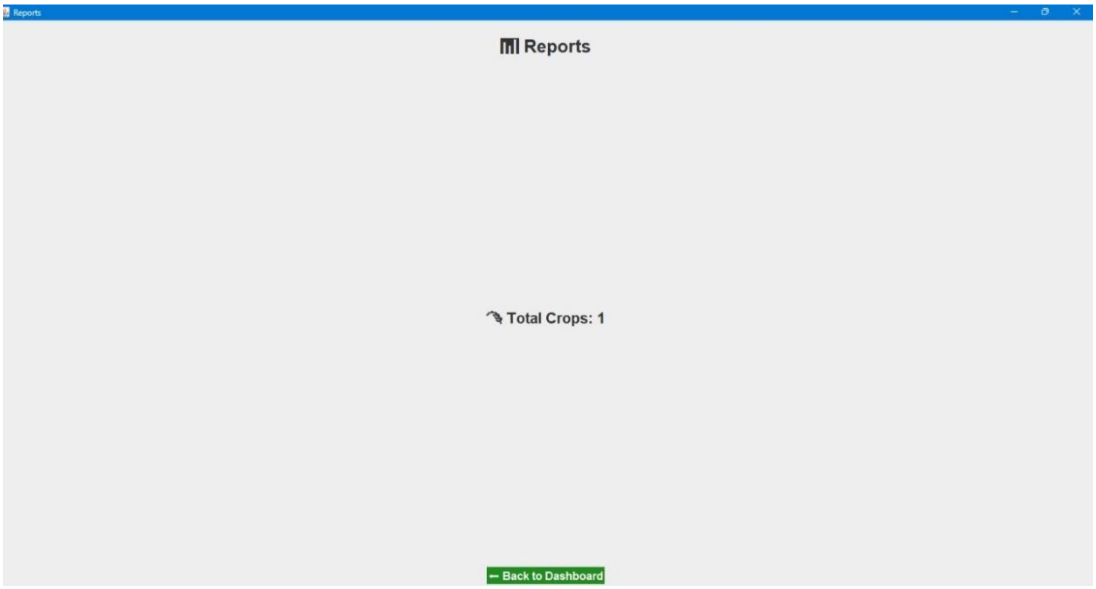
Fig 5.5 Sensor Page



✔ Reading Saved Successfully!

🌡 Temp: 45.0 °C
💧 Humidity: 24.0%
🌱 Moisture: 15.0%

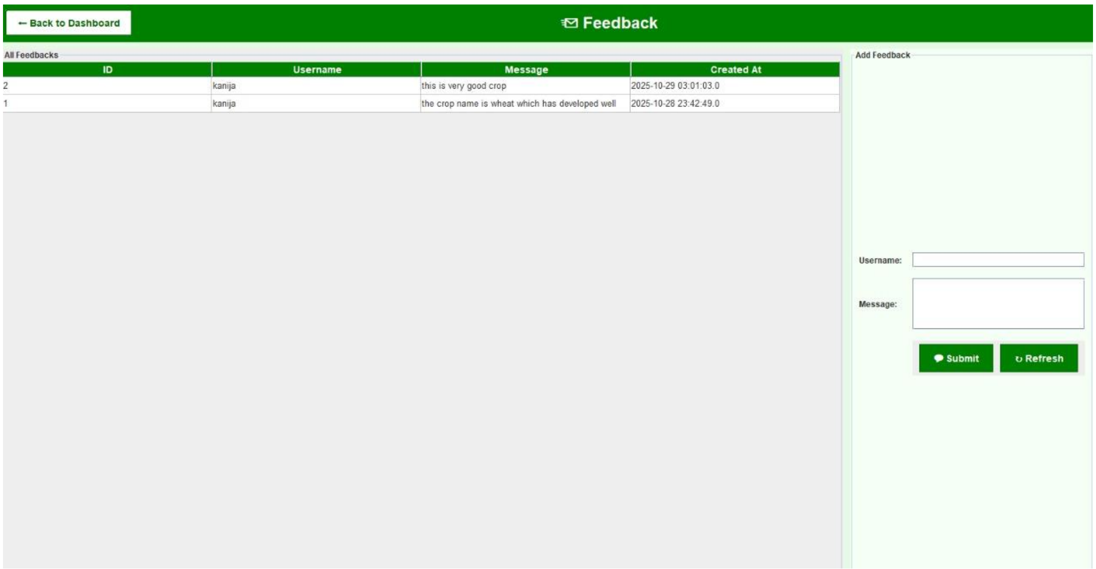
Fig 5.6 Sensor Readings Page



Reports

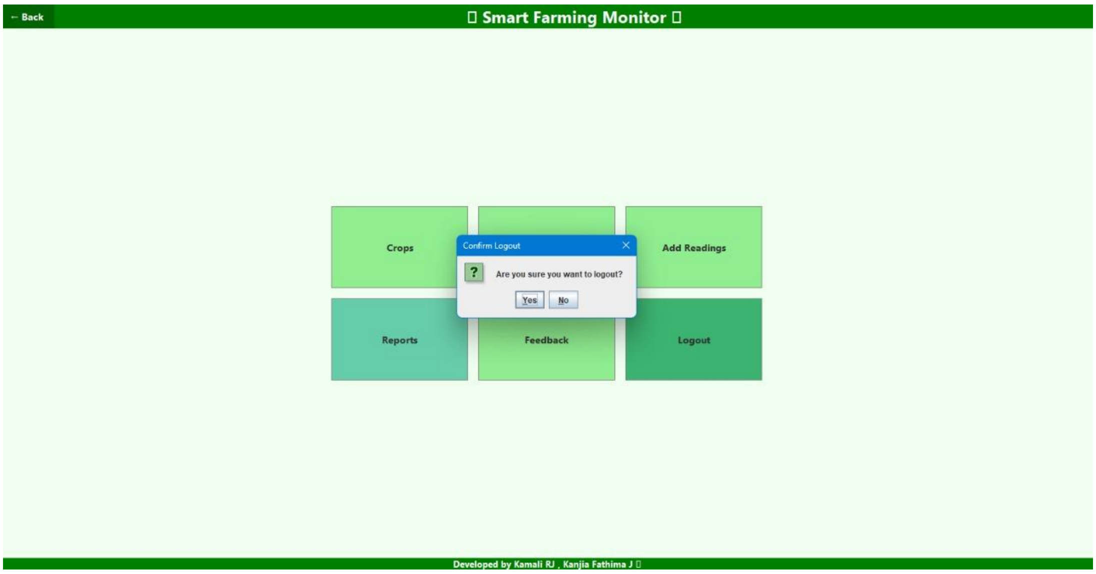
Total Crops: 1

Fig 5.7 Report Page



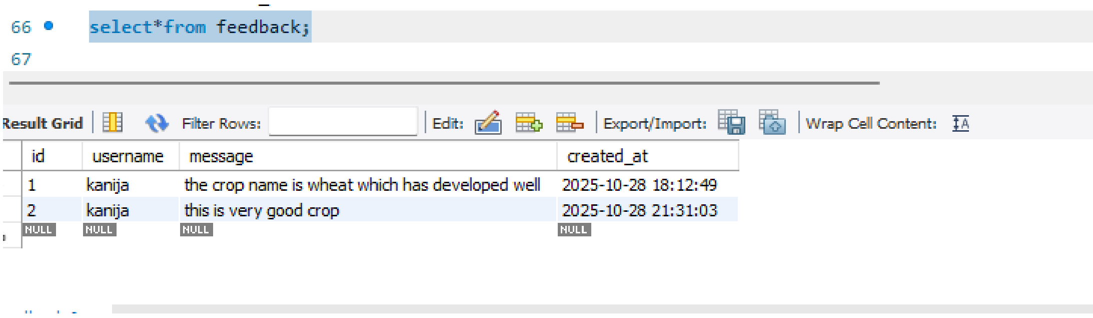
ID	Username	Message	Created At
2	kanija	this is very good crop	2025-10-29 03:01:03.0
1	kanija	the crop name is wheat which has developed well	2025-10-28 23:42:49.0

Fig 5.8 Feedback Page



Are you sure you want to logout?

Fig 5.9 Logout Page



```
select*from feedback;
```

id	username	message	created_at
1	kanija	the crop name is wheat which has developed well	2025-10-28 18:12:49
2	kanija	this is very good crop	2025-10-28 21:31:03
NULL	NULL	NULL	NULL

Fig 5.10 Database Table

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

Through this project, users can efficiently manage and monitor crop information in a digital format. The system simplifies storing, viewing, and updating crop details, making data handling easier and more reliable for farmers. It provides a user-friendly interface and secure database connection for effective record management.

In the future, this system can be enhanced by adding features such as weather data integration, crop growth prediction, and report generation to make farming decisions even smarter and more efficient.

1. https://www.w3schools.com/java/

2.https://www.mysqltutorial.org/

3.https://www.geeksforgeeks.org/java-database-connectivity-jdbc/