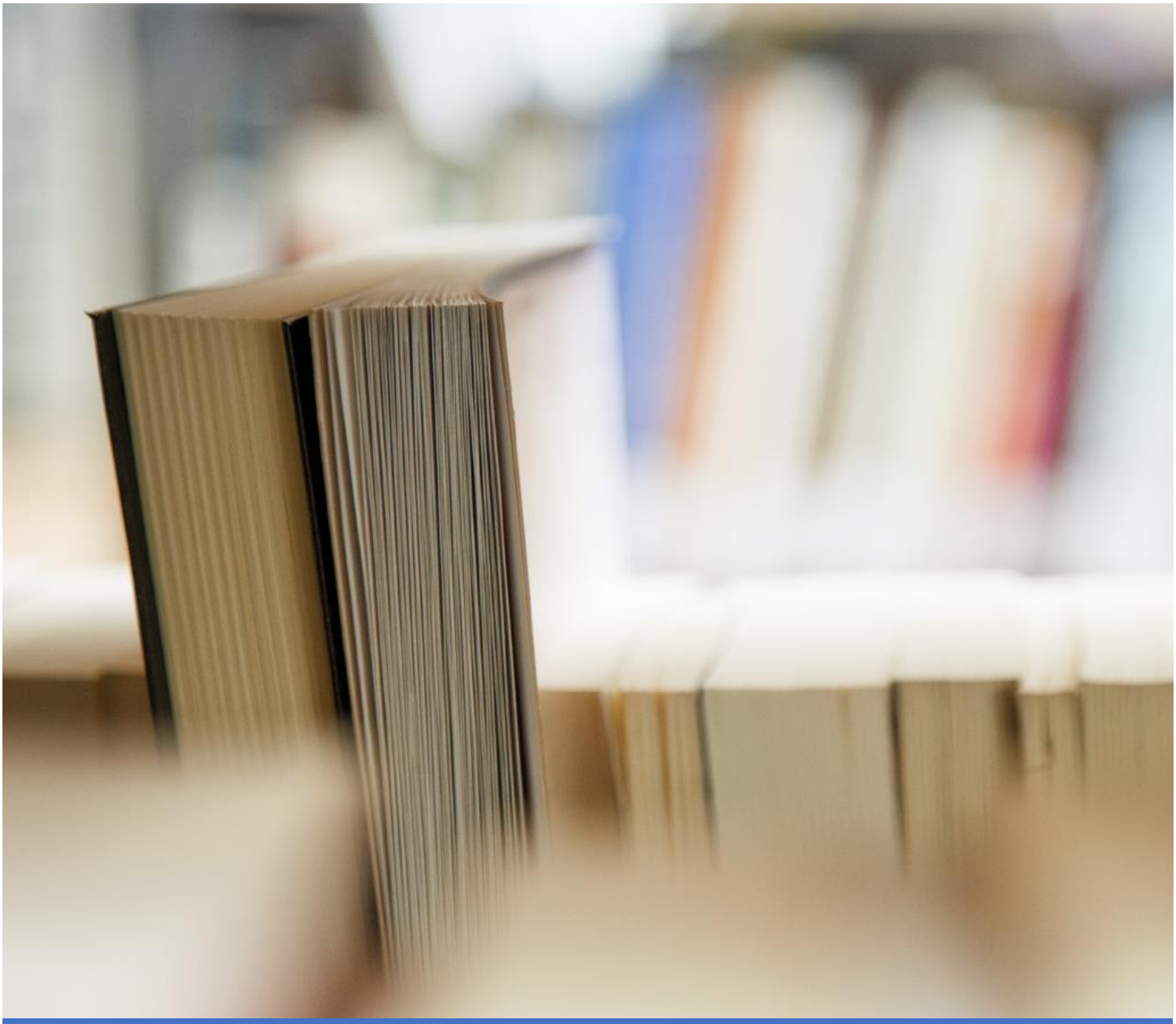


Dynamic Farming

Waldo Ackerman

5/20/19

CMPC 212

Table of Contents

Project Title

- Dynamic Farming

Group Members

- W. Ackerman – 23593644
- M. Werth – 30365198

Introduction and Purpose

The program is intended for farmers to keep better track of their assets, livestock, grain and workers as well as any related product or implement that is required in any of the processes.

You have to log on to the site first then you will be directed to the main page from where you can navigate to the various other pages for each element of the farmers business.

The user will be able to add data, delete data as well as search.

The site is also user friendly and conveniently designed.

Program Description

Functionality

To determine and track stock levels of the various elements of a farming business.

Code

LOGIN PAGE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace Dynamic
{
    public partial class Login : System.Web.UI.Page
    {
        SqlConnection cnn;


        protected void Page_Load(object sender, EventArgs e)
        {
            this.UnobtrusiveValidationMode =
System.Web.UI.UnobtrusiveValidationMode.None;
            string conString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\waldo\Desktop\Dynamic\Dynamic_
Farm.mdf;Integrated Security=True;Connect Timeout=30";
            cnn = new SqlConnection(conString);
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            cnn.Open();
            string sql = "Select Count(1) From Login Where username=@username AND
password=@password";

            SqlCommand command = new SqlCommand(sql, cnn);
            command.Parameters.AddWithValue("@username", txtID.Text.Trim());
            command.Parameters.AddWithValue("@password", txtPW.Text.Trim());
            int count = Convert.ToInt32(command.ExecuteScalar());
            if(count == 1)
            {
                HttpCookie userInfo = new HttpCookie("info");
                userInfo["ID"] = txtID.Text;
                userInfo.Expires = DateTime.Now.AddHours(2);
                Response.Cookies.Add(userInfo);

                Response.Redirect("Main.aspx");
            }

            cnn.Close();
        }
```

```
    }
}
```

MAIN PAGE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Dynamic
{
    public partial class Main : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            HttpCookie userInfo = Request.Cookies["info"];
            lblWelcome.Text = "Welcome: " + userInfo["ID"];
            Session["date"] = DateTime.Today.DayOfWeek.ToString() + "," +
DateTime.Today.ToShortDateString();

            this.Calendar1.SelectedDate = DateTime.Today;

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            Response.Redirect("Login.aspx");
        }

        protected void Calendar1_SelectionChanged(object sender, EventArgs e)
        {
            if (this.Calendar1.SelectedDate.DayOfYear < DateTime.Today.DayOfYear)
            {
                this.lblDate.Text = "Select the future date";
            }


        }
    }
}
```

IMPLEMENTS PAGE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace Dynamic
{
    public partial class Implement : System.Web.UI.Page
    {
```

```
        SqlConnection cnnn;
        protected void Page_Load(object sender, EventArgs e)
        {
            HttpCookie userInfo = Request.Cookies["info"];
            lblImp.Text = "Welcome: " + userInfo["ID"];
            lblDate.Text = "The date is: " + Session["date"];

            try
            {

                string connection = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\waldo\Desktop\Dynamic\Dynamic_
Farm.mdf;Integrated Security=True;Connect Timeout=30";
                cnnn = new SqlConnection(connection);

                cnnn.Open();
                SqlCommand comm;
                string sqlh = "";
                SqlDataAdapter adap = new SqlDataAdapter();

                sqlh = @"Select * From Implements";
                comm = new SqlCommand(sqlh, cnnn);
                DataSet ds = new DataSet();

                adap.SelectCommand = comm;
                adap.Fill(ds);

                gvImp.DataSource = ds;
                gvImp.DataBind();

                cnnn.Close();
            }
            catch(Exception error)
            {
                txtSearchImp.Text = "Unable to connect";
            }

        }

        protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Login.aspx");
        }

        protected void TextBox2_TextChanged(object sender, EventArgs e)
        {
        }

        protected void btnSearchImp_Click(object sender, EventArgs e)
        {
            try
            {

                cnnn.Open();
                SqlCommand comm;
                string sqlk = "";
                SqlDataAdapter adap = new SqlDataAdapter();
```

```
                sqlk = @"Select * FROM Implements WHERE Implement LIKE '%" +
txtSearchImp.Text + "%'" +
                    "OR Description  LIKE '%" + txtSearchImp.Text + "%'" +
                    "OR CostToPurchase LIKE '%" + txtSearchImp.Text + "%'" +
                    "OR SellengPrice LIKE '%" + txtSearchImp.Text + "%'" +
                    "OR DateOfPurchase  LIKE '%" + txtSearchImp.Text + "%'" +
                    "OR Condition LIKE '%" + txtSearchImp.Text + "%'" +
                    "OR Repair LIKE '%" + txtSearchImp.Text + "%'";



                comm = new SqlCommand(sqlk, cnnn);
                DataSet ds = new DataSet();

                adap.SelectCommand = comm;
                adap.Fill(ds);

                gvImp.DataSource = ds;
                gvImp.DataBind();

                cnnn.Close();
            }

            catch(Exception error)
            {
                txtSearchImp.Text = "Unable to connect";
            }

}

        protected void Button2_Click(object sender, EventArgs e)
        {

            try
            {
                cnnn.Open();
                SqlCommand command;
                SqlDataAdapter adap = new SqlDataAdapter();
                string dela = "DELETE FROM Implements WHERE Implement LIKE '%" +
txtDELimp.Text + "%'";

                command = new SqlCommand(dela, cnnn);
                command.ExecuteNonQuery();
                cnnn.Close();

                cnnn.Open();
                SqlCommand comm;
                string sqll = "";
                SqlDataAdapter adapt = new SqlDataAdapter();

                sqll = @"Select * From Implements";
                comm = new SqlCommand(sqll, cnnn);
                DataSet ds = new DataSet();

                adapt.SelectCommand = comm;
                adapt.Fill(ds);

                gvImp.DataSource = ds;
                gvImp.DataBind();
                cnnn.Close();
            }
```

```
    catch (Exception error)
    {
        txtDELimp.Text = "Invalid, unable to complete action";
    }

}

protected void btnRepair_Click(object sender, EventArgs e)
{
    try
    {

        cnnn.Open();
        SqlCommand comm;
        string sql = "";
        SqlDataAdapter adap = new SqlDataAdapter();

        sql = @"Select * From Implements WHERE Repair = 'Y'";
        comm = new SqlCommand(sql, cnnn);
        DataSet ds = new DataSet();

        adap.SelectCommand = comm;
        adap.Fill(ds);

        gvImp.DataSource = ds;
        gvImp.DataBind();

        cnnn.Close();
    }
    catch (Exception error)
    {
        txtSearchImp.Text = "Unable to connect";
    }
}

protected void btnCon_Click(object sender, EventArgs e)
{
    try
    {

        cnnn.Open();
        SqlCommand comm;
        string sql = "";
        SqlDataAdapter adap = new SqlDataAdapter();

        sql = @"Select * From Implements WHERE Condition = 'Good'";
        comm = new SqlCommand(sql, cnnn);
        DataSet ds = new DataSet();

        adap.SelectCommand = comm;
        adap.Fill(ds);

        gvImp.DataSource = ds;
        gvImp.DataBind();

        cnnn.Close();
    }
    catch (Exception error)
    {
        txtSearchImp.Text = "Unable to connect";
    }
}
```

```
    }
}
```

LIVESTOCK PAGE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace Dynamic
{
    public partial class LiveStock : System.Web.UI.Page
    {
        SqlConnection cnns;
        protected void Page_Load(object sender, EventArgs e)
        {
            HttpCookie userInfo = Request.Cookies["info"];
            lblLive.Text = "Welcome: " + userInfo["ID"] + " The date is: " +
Session["date"];


            try
            {


            string connection = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\waldo\Desktop\Dynamic\Dynamic_
Farm.mdf;Integrated Security=True;Connect Timeout=30";
            cnns = new SqlConnection(connection);

            cnns.Open();
            SqlCommand comm;
            string sqlo = "";
            SqlDataAdapter adapt = new SqlDataAdapter();

            sqlo = @"Select * From LiveStock";
            comm = new SqlCommand(sqlo, cnns);
            DataSet ds = new DataSet();

            adapt.SelectCommand = comm;
            adapt.Fill(ds);

            gvLive.DataSource = ds;
            gvLive.DataBind();
            cnns.Close();
            }
            catch (Exception error)
            {
                txtSearchLive.Text = "Unable to connect";
            }

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Login.aspx");
        }
```

```
        protected void btnSearchLive_Click(object sender, EventArgs e)
        {
            try
            {
                cnns.Open();
                SqlCommand comm;
                string sqlz = "";
                SqlDataAdapter adap = new SqlDataAdapter();

                sqlz = @"Select * FROM LiveStock WHERE LiveStockType LIKE '%" +
txtSearchLive.Text + "%'" +
                    "OR LiveStockAmount  LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR LiveStockLocation LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR TypeOfFeed LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR FeedOnHand_pKg  LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR ArrivingWeight_pKg LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR SellingWeight_pKg LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR BoughtFor LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR SellengFor LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR ArrivingDate LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR SellingDate LIKE '%" + txtSearchLive.Text + "%'" +
                    "OR AmountSold LIKE '%" + txtSearchLive.Text + "%'";



                comm = new SqlCommand(sqlz, cnns);
                DataSet ds = new DataSet();

                adap.SelectCommand = comm;
                adap.Fill(ds);

                gvLive.DataSource = ds;
                gvLive.DataBind();
                cnns.Close();
            }
            catch (Exception error)
            {
                txtSearchLive.Text = "Unable to connect";
            }

        }

        protected void btnDELLive_Click(object sender, EventArgs e)
        {
            try
            {
                cnns.Open();
                SqlCommand command;
                SqlDataAdapter adap = new SqlDataAdapter();
                string delm = "DELETE FROM LiveStock WHERE LiveStockType LIKE '%" +
txtDELLive.Text + "%'";

                command = new SqlCommand(delm, cnns);
                command.ExecuteNonQuery();
                cnns.Close();

                cnns.Open();
                SqlCommand comm;
                string sqlq = "";
                SqlDataAdapter adapt = new SqlDataAdapter();
```

```
                sqlq = @"Select * From LiveStock";
                comm = new SqlCommand(sqlq, cnns);
                DataSet ds = new DataSet();

                adapt.SelectCommand = comm;
                adapt.Fill(ds);

                gvLive.DataSource = ds;
                gvLive.DataBind();
                cnns.Close();
            }
            catch (Exception error)
            {
                txtDELLive.Text = "Invalid, unable to complete action";
            }

        }

        protected void btnSortFeed_Click(object sender, EventArgs e)
        {
            cnns.Open();
            SqlCommand comm;
            string sqlx = "";
            SqlDataAdapter adapt = new SqlDataAdapter();

            sqlx = @"Select * From LiveStock WHERE FeedOnHand_pKg < 2300";
            comm = new SqlCommand(sqlx, cnns);
            DataSet ds = new DataSet();

            adapt.SelectCommand = comm;
            adapt.Fill(ds);

            gvLive.DataSource = ds;
            gvLive.DataBind();
            cnns.Close();
        }

        protected void btnSell_Click(object sender, EventArgs e)
        {
            cnns.Open();
            SqlCommand comm;
            string sqlx = "";
            SqlDataAdapter adapt = new SqlDataAdapter();

            sqlx = @"Select * From LiveStock WHERE BoughtFor < 1000";
            comm = new SqlCommand(sqlx, cnns);
            DataSet ds = new DataSet();

            adapt.SelectCommand = comm;
            adapt.Fill(ds);

            gvLive.DataSource = ds;
            gvLive.DataBind();
            cnns.Close();
        }
    }
}
```

GRAIN PAGE

```
using System;
using System.Collections.Generic;
```

```
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace Dynamic
{
    public partial class GrainPage : System.Web.UI.Page
    {
        SqlConnection cnn;
        protected void Page_Load(object sender, EventArgs e)
        {
            HttpCookie userInfo = Request.Cookies["info"];
            lblGrain.Text = "Welcome: " + userInfo["ID"] + " The date is: " +
Session["date"];

            try
            {

                string connection = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\waldo\Desktop\Dynamic\Dynamic_
Farm.mdf;Integrated Security=True;Connect Timeout=30";
                cnn = new SqlConnection(connection);

                cnn.Open();
                SqlCommand comm;
                string sqlx = "";
                SqlDataAdapter adapt = new SqlDataAdapter();

                sqlx = @"Select * From Grain";
                comm = new SqlCommand(sqlx, cnn);
                DataSet ds = new DataSet();

                adapt.SelectCommand = comm;
                adapt.Fill(ds);

                gvGrains.DataSource = ds;
                gvGrains.DataBind();
                cnn.Close();
                }
            catch (Exception error)
            {
                txtsearchGrain.Text = "Unable to connect";
            }

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Login.aspx");
        }


        protected void btnSearchGrain_Click(object sender, EventArgs e)
        {
            try
            {

            cnn.Open();
            SqlCommand comm;
```

```
            string sqlk = "";
            SqlDataAdapter adap = new SqlDataAdapter();

            sqlk = @"Select * FROM Grain WHERE GrainType LIKE '%" +
txtsearchGrain.Text + "%'" +
                "OR GrainOnHand_pKg  LIKE '%" + txtsearchGrain.Text + "%'" +
                "OR CostPrice_pTon LIKE '%" + txtsearchGrain.Text + "%'" +
                "OR SellingPrice_pTon LIKE '%" + txtsearchGrain.Text + "%'" +
                "OR GrainLocation  LIKE '%" + txtsearchGrain.Text + "%'" +
                "OR TypeFertilizer LIKE '%" + txtsearchGrain.Text + "%'" +
                "OR FertilizerOnHand_pKg LIKE '%" + txtsearchGrain.Text + "%'" +
                "OR CostFertilizer_pKg LIKE '%" + txtsearchGrain.Text + "%'" +
                "OR FarmPlanted LIKE '%" + txtsearchGrain.Text + "%'";



            comm = new SqlCommand(sqlk, cnn);
            DataSet ds = new DataSet();

            adap.SelectCommand = comm;
            adap.Fill(ds);

            gvGrains.DataSource = ds;
            gvGrains.DataBind();
            cnn.Close();
            }
            catch (Exception error)
            {
                txtsearchGrain.Text = "Unable to connect";
            }
        }

        protected void btnDELGrain_Click(object sender, EventArgs e)
        {
            try
            {
                cnn.Open();
                SqlCommand command;
                SqlDataAdapter adap = new SqlDataAdapter();
                string delg = "DELETE FROM Grain WHERE GrainType LIKE '%" +
txtDELGrain.Text + "%'";

                command = new SqlCommand(delg, cnn);
                command.ExecuteNonQuery();
                cnn.Close();

                cnn.Open();
                SqlCommand comm;
                string sqll = "";
                SqlDataAdapter adapt = new SqlDataAdapter();

                sqll = @"Select * From Grain";
                comm = new SqlCommand(sqll, cnn);
                DataSet ds = new DataSet();

                adapt.SelectCommand = comm;
                adapt.Fill(ds);

                gvGrains.DataSource = ds;
                gvGrains.DataBind();
                cnn.Close();
            }
```

```
            catch (Exception error)
            {
                txtDELGrain.Text = "Invalid, unable to complete action";
            }
        }

        protected void btnSortOnHand_Click(object sender, EventArgs e)
        {
            cnn.Open();
            SqlCommand comm;
            string sqlx = "";
            SqlDataAdapter adapt = new SqlDataAdapter();

            sqlx = @"Select * From Grain WHERE GrainOnHand_pKg < 400";
            comm = new SqlCommand(sqlx, cnn);
            DataSet ds = new DataSet();

            adapt.SelectCommand = comm;
            adapt.Fill(ds);

            gvGrains.DataSource = ds;
            gvGrains.DataBind();
            cnn.Close();
        }

        protected void btnFertAvail_Click(object sender, EventArgs e)
        {
            cnn.Open();
            SqlCommand comm;
            string sqlx = "";
            SqlDataAdapter adapt = new SqlDataAdapter();

            sqlx = @"Select * From Grain WHERE FertilizerOnHand_pKg < 900";
            comm = new SqlCommand(sqlx, cnn);
            DataSet ds = new DataSet();

            adapt.SelectCommand = comm;
            adapt.Fill(ds);

            gvGrains.DataSource = ds;
            gvGrains.DataBind();
            cnn.Close();
        }
    }
}
```

WORKERS PAGE

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace Dynamic
{
    public partial class Workers : System.Web.UI.Page
    {
        SqlConnection cnn;
```

```
        DataSet ds;
        SqlDataAdapter adapter;

        protected void Page_Load(object sender, EventArgs e)
        {
            HttpCookie userInfo = Request.Cookies["info"];
            lblWork.Text = "Welcome: " + userInfo["ID"] + " The date is: " +
Session["date"];

            try
            {

            string connString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\waldo\Desktop\Dynamic\Dynamic_
Farm.mdf;Integrated Security=True;Connect Timeout=30";
            cnn = new SqlConnection(connString);

            cnn.Open();
            SqlCommand comm;
            string sql = "";
            SqlDataAdapter adap = new SqlDataAdapter();

            sql = @"Select WorkerName, JobDescription, AessetsAssigned, Availability
From Workers";
            comm = new SqlCommand(sql, cnn);
            DataSet ds = new DataSet();

            adap.SelectCommand = comm;
            adap.Fill(ds);

            gvWorkers.DataSource = ds;
            gvWorkers.DataBind();

            cnn.Close();
            }
            catch (Exception error)
            {
                txtSearchWork.Text = "Unable to connect";
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Login.aspx");
        }

        protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        protected void TextBox2_TextChanged(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {


        }
```

```
        protected void Button3_Click(object sender, EventArgs e)
        {

        }

        protected void btnSortAlpha_Click(object sender, EventArgs e)
        {
            try
            {

            cnn.Open();
            SqlCommand comm;
            string sql = "";
            SqlDataAdapter adap = new SqlDataAdapter();

            sql = @"Select WorkerName, JobDescription, AessetsAssigned, Availability
From Workers ORDER BY WorkerName DESC";
            comm = new SqlCommand(sql, cnn);
            DataSet ds = new DataSet();

            adap.SelectCommand = comm;
            adap.Fill(ds);

            gvWorkers.DataSource = ds;
            gvWorkers.DataBind();

            cnn.Close();
            }
            catch (Exception error)
            {
                txtSearchWork.Text = "Unable to connect";
            }

        }

        protected void btnSortAvail_Click(object sender, EventArgs e)
        {
            try
            {

            cnn.Open();
            SqlCommand comm;
            string sql = "";
            SqlDataAdapter adap = new SqlDataAdapter();

            sql = @"Select WorkerName, JobDescription, AessetsAssigned, Availability
From Workers WHERE Availability = 'N'";
            comm = new SqlCommand(sql, cnn);
            DataSet ds = new DataSet();

            adap.SelectCommand = comm;
            adap.Fill(ds);

            gvWorkers.DataSource = ds;
            gvWorkers.DataBind();

            cnn.Close();
            }
            catch (Exception error)
            {
                txtSearchWork.Text = "Unable to connect";
            }
```

```
        }

        protected void btnSearchW_Click(object sender, EventArgs e)
        {
            try
            {

                cnn.Open();
                SqlCommand comm;
                string sqlt = "";
                SqlDataAdapter adap = new SqlDataAdapter();

                sqlt = @"Select WorkerName, JobDescription, AessetsAssigned,
Availability FROM Workers WHERE WorkerName LIKE '%" + txtSearchWork.Text + "%'" + "OR
JobDescription  LIKE '%" + txtSearchWork.Text + "%'" + "OR AessetsAssigned LIKE '%" +
txtSearchWork.Text + "%'" + "OR Availability LIKE '%" + txtSearchWork.Text + "%'";
                comm = new SqlCommand(sqlt, cnn);
                DataSet ds = new DataSet();

                adap.SelectCommand = comm;
                adap.Fill(ds);

                gvWorkers.DataSource = ds;
                gvWorkers.DataBind();
                cnn.Close();
            }
            catch (Exception error)
            {
                txtSearchWork.Text = "Unable to connect";
            }
        }

        protected void btnDelWork_Click(object sender, EventArgs e)
        {
            try
            {
                cnn.Open();
                SqlCommand command;
                SqlDataAdapter adap = new SqlDataAdapter();
                string del = "DELETE FROM Workers WHERE WorkerName LIKE '%" +
txtDELWork.Text + "%'";

                command = new SqlCommand(del, cnn);
                command.ExecuteNonQuery();
                cnn.Close();

                cnn.Open();
                SqlCommand comm;
                string sqls = "";
                SqlDataAdapter adapt = new SqlDataAdapter();

                sqls = @"Select WorkerName, JobDescription, AessetsAssigned,
Availability From Workers";
                comm = new SqlCommand(sqls, cnn);
                DataSet ds = new DataSet();

                adapt.SelectCommand = comm;
                adapt.Fill(ds);

                gvWorkers.DataSource = ds;
                gvWorkers.DataBind();
                cnn.Close();
```

```
            }
            catch (Exception error)
            {
                txtDELWork.Text = "";
            }
        }

        protected void LinkButton5_Click(object sender, EventArgs e)
        {

        }
    }
}
```

DATABASES

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Data;

namespace Dynamic
{
    public partial class DataBase : System.Web.UI.Page
    {
        SqlConnection cns;

        protected void Page_Load(object sender, EventArgs e)
        {
            HttpCookie userInfo = Request.Cookies["info"];
            lblData.Text = "Welcome: " + userInfo["ID"] + " The date is: " +
Session["date"];


            Panel1.Visible = false;
            Panel3.Visible = false;
            Panel4.Visible = false;
            Panel5.Visible = false;
            Panel6.Visible = false;

            string connString = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\waldo\Desktop\Dynamic\Dynamic_
Farm.mdf;Integrated Security=True;Connect Timeout=30";
            cns = new SqlConnection(connString);


        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("Login.aspx");
        }

        protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs
e)
        {

        }
```

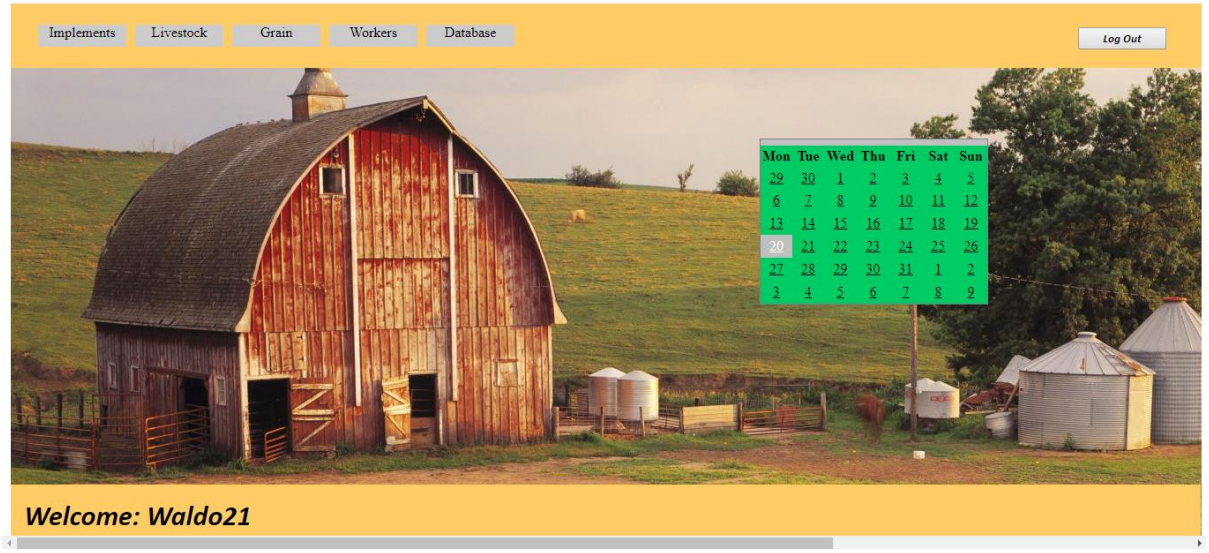
```
        protected void RadioButtonList1_SelectedIndexChanged1(object sender, EventArgs
e)
        {

        }

        protected void Button3_Click(object sender, EventArgs e)
        {
            if (RadioButtonList1.SelectedIndex == 0)
            {
                Panel1.Visible = true;
            }
            if (RadioButtonList1.SelectedIndex == 1)
            {
                Panel3.Visible = true;
            }
            if (RadioButtonList1.SelectedIndex == 2)
            {
                Panel4.Visible = true;
            }
            if (RadioButtonList1.SelectedIndex == 3)
            {
                Panel5.Visible = true;
            }
            if (RadioButtonList1.SelectedIndex == 4)
            {
                Panel6.Visible = true;
            }

        }

        protected void btnLogin_Click(object sender, EventArgs e)
        {
            int ID;
            string userName, password;

            try
            {
                ID = Convert.ToInt32(TextBox49.Text);
                userName = TextBox50.Text;
                password = TextBox51.Text;

                cns.Open();

                string sqlw = @"INSERT INTO Login VALUES('" + ID + "','" + userName +
"','" + password + "')";

                SqlCommand command = new SqlCommand(sqlw, cns);

                command.ExecuteNonQuery();
                lblDisplay.Text = "Data Succesfully Added!!";

                cns.Close();

            }
            catch(Exception error)
            {
                TextBox49.Text = "Error";
            }
        }
```

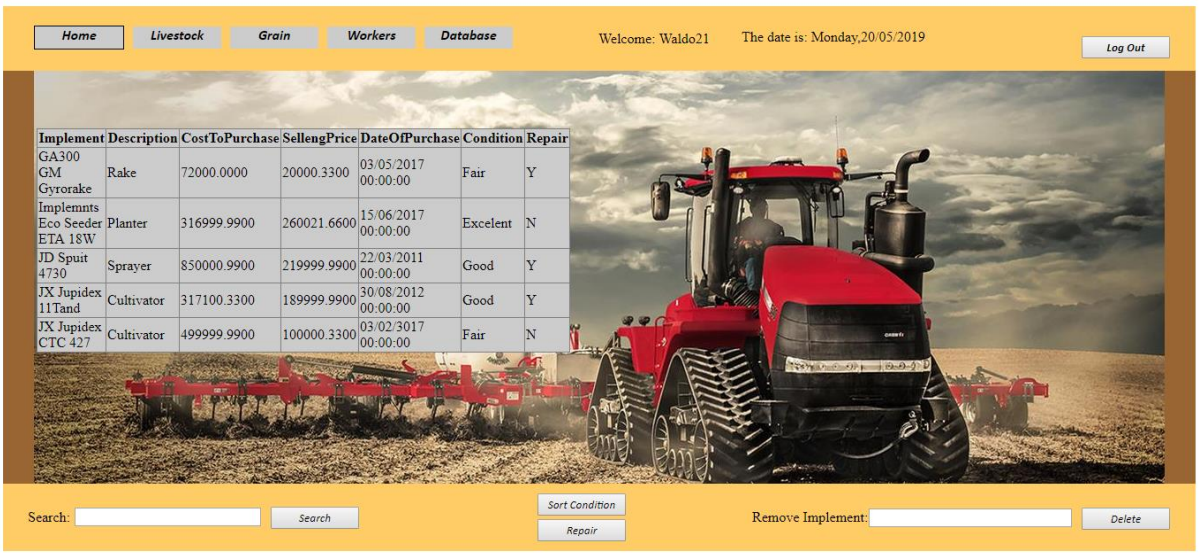
```
        protected void Button6_Click(object sender, EventArgs e)
        {
            string Name, jobDesc, asset, avail, hoursWorked, hourRate, wage;

            try
            {
                Name = TextBox42.Text;
                jobDesc = TextBox43.Text;
                asset = TextBox44.Text;
                avail = TextBox45.Text;
                hoursWorked = TextBox46.Text;
                hourRate = TextBox47.Text;
                wage = TextBox48.Text;

                cns.Open();

                string sqlw = @"INSERT INTO Workers VALUES('" + Name + "','" + jobDesc
+ "','" + asset +"','" + avail +"','" + hoursWorked +"','" + hourRate +"','" + wage +
"')";

                SqlCommand command = new SqlCommand(sqlw, cns);

                command.ExecuteNonQuery();
                lblDisplay.Text = "Data Succesfully Added!!";

                cns.Close();

            }
            catch (Exception error)
            {
                TextBox49.Text = "Error";
            }
        }

        protected void btnGrain_Click(object sender, EventArgs e)
        {
            string type, grainOnHand, costPrice, sellingPrice, location, fertilizer,
fertOnHand, fertCost, farmPlanted;

            try
            {
                type = TextBox56.Text;
                grainOnHand = TextBox57.Text;
                costPrice = TextBox58.Text;
                sellingPrice = TextBox59.Text;
                location = TextBox60.Text;
                fertilizer = TextBox61.Text;
                fertOnHand = TextBox62.Text;
                fertCost = TextBox63.Text;
                farmPlanted = TextBox64.Text;

                cns.Open();

                string sqlss = @"INSERT INTO Grain VALUES('" + type + "','" +
grainOnHand + "','" + costPrice + "','" + sellingPrice + "','" + location + "','" +
fertilizer + "','" + fertOnHand + "','" + fertCost + "','" + farmPlanted + "')";

                SqlCommand command = new SqlCommand(sqlss, cns);

                command.ExecuteNonQuery();
                lblDisplay.Text = "Data Succesfully Added!!";
```

```
                cns.Close();

            }
            catch (Exception error)
            {
                TextBox49.Text = "Error";
            }
        }

        protected void btnLive_Click(object sender, EventArgs e)
        {
            string type, amount, location, feedType, feedOnHand, arrivingWeight,
sellingWeight, costPrice, sellingPrice, arriveDate, sellDate, amountSold;

            try
            {
                type = TextBox9.Text;
                amount = TextBox10.Text;
                location = TextBox11.Text;
                feedType = TextBox12.Text;
                feedOnHand = TextBox13.Text;
                arrivingWeight = TextBox14.Text;
                sellingWeight = TextBox29.Text;
                costPrice = TextBox30.Text;
                sellingPrice = TextBox31.Text;
                arriveDate = TextBox32.Text;
                sellDate = TextBox33.Text;
                amountSold = TextBox34.Text;

                cns.Open();

                string sqlw = @"INSERT INTO LiveStock VALUES('" + type + "','" +
amount + "','" + location + "','" + feedType + "','" + feedOnHand + "','" +
arrivingWeight + "','" + sellingWeight + "','" + costPrice + "','" + sellingPrice +
"','" + arriveDate + "','" + sellDate + "','" + amountSold + "')";

                SqlCommand command = new SqlCommand(sqlw, cns);

                command.ExecuteNonQuery();
                lblDisplay.Text = "Data Succesfully Added!!";

                cns.Close();

            }
            catch (Exception error)
            {
                TextBox49.Text = "Error";
            }

        }

        protected void btnImp_Click(object sender, EventArgs e)
        {
            string Imp, desr, costPrice, sellingPrice, purchDate, condition, repair;

            try
            {
                Imp = TextBox1.Text;
                desr = TextBox2.Text;
                costPrice = TextBox3.Text;
                sellingPrice = TextBox4.Text;
                purchDate = TextBox5.Text;
```

```
                condition = TextBox6.Text;
                repair = TextBox7.Text;

                cns.Open();

                string sqlw = @"INSERT INTO Implements VALUES('" + Imp + "','" + desr
+ "','" + costPrice + "','" + sellingPrice + "','" + purchDate + "','" + condition +
"','" + repair + "')";

                SqlCommand command = new SqlCommand(sqlw, cns);

                command.ExecuteNonQuery();
                lblDisplay.Text = "Data Succesfully Added!!";

                cns.Close();

            }
            catch (Exception error)
            {
                TextBox49.Text = "Error";
            }

        }

        protected void TextBox32_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

Screenshots





Home | Implements | Grain | Workers | Database

Welcome: Waldo21 The date is: Monday,20/05/2019

Log Out

LiveStockType	LiveStockAmount	LiveStockLocation	TypeOfFeed	FeedOnHand_pKg	ArrivingWeight_pKg	SellingWeight_pKg	BoughtFor	SellengFor	ArrivingDate	SellingDate	AmountSold
Bosmare Cow 211C	426.0000	Feedingstaion-A1	TypeOfFeed Dry sublement V14603	11002	355	560	3521.8800	9069.3300	01/02/2019 00:00:00	04/07/2019 00:00:00	0
Chester White Pig 311P	98.0000	FeedingStaion-SC	TypeOfFeed Molasses Meal N-FF0079	21045	65	95	984.2200	1120.9900	02/01/2019 00:00:00	29/05/2019 00:00:00	0
Dorper Sheep 121S	265.0000	BlockK-GrassField	TEFF	892	36	51	1123.6900	6032.2200	15/01/2019 00:00:00	11/07/2019 00:00:00	0
Landrace Pig 321P	36.0000	FeedingStaion-SP	TypeOfFeed Molvite V7266	25410	88	91	752.3300	801.2200	01/03/2019 00:00:00	03/05/2019 00:00:00	1233.21
Large white Pig 331P	22.0000	FeedingSation-S1	Korn Kandy V1551	2310	85	96	3650.2200	4300.2200	02/02/2019 00:00:00	31/05/2019 00:00:00	0
Merino Sheep 111S	636.0000	BlockA-GrassField	TEFF	850	50	72	955.3300	2111.0000	03/05/2019 00:00:00	02/08/2019 00:00:00	0
Nguni Cow 221C	111.0000	BlockF-GrassField	TEFF	1120	360	630	6251.6600	1125.3300	31/01/2019 00:00:00	18/09/2019 00:00:00	0

Search: Search

Sort Feed Availability

Sort Price

Remove records: Delete

Home | Implements | Livestock | Workers | Database

Welcome: Waldo21 The date is: Monday,20/05/2019

Log Out

GrainType	GrainOnHand_pKg	CostPrice_pTon	SellingPrice_pTon	GrainLocation	TypeFertilizer	FertilizerOnHand_pKg	CostFertilizer_pKg	FarmPlanted
CornV2-2586	400	489.33	2510.33	StoreA4	UREA46%N	3600	883.56	Block-B
Hybrid Maize 3M-01	952	650.99	1100.66	Store-KL2	CUAN	4500	600.25	BETA
Monceren Corn	490	980.24	1200.33	SiloRoom-3	NPK4|2|3	5000	392.66	Uitzhoek
Soya B-k4314	380	366.87	2360.22	Store -A2	DAP8|15|36	1235	930.88	Hermanus
Sunflow BK_sl908	982	755.36	4366.22	SiloRoom-3	NPK9|13|22	3302	380.22	Paradys

Search: Search

Grain Availability

Fertilize Availabilty

Remove by type: Delete

Home | Implements | Livestock | Grain | Database

Welcome: Waldo21 The date is: Monday,20/05/2019

Log Out

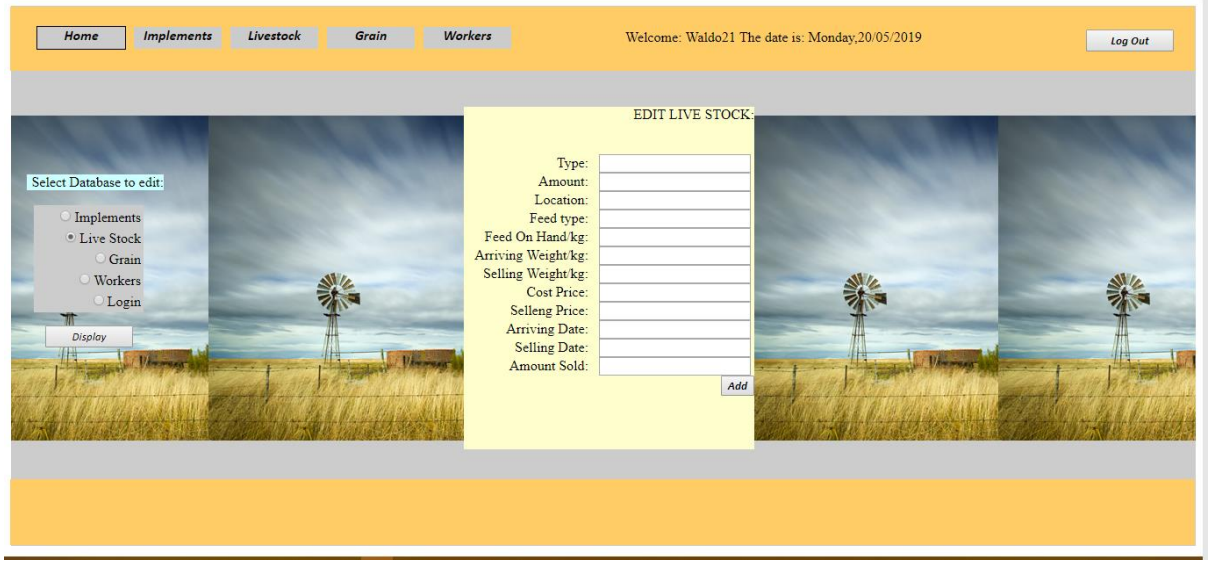
WorkerName	JobDescription	AessetsAssigned	Availability
Jan	522	22	22
Piet	652	62	62

Search: Search

Sort Alphabetically

Sort by Availability

Remove by name: Delete


Home
Implements
Livestock
Grain
Workers
Welcome: Waldo21 The date is: Monday,20/05/2019
Log Out
Select Database to edit:
Implements
Live Stock
Grain
Workers
Login
Display
EDIT LIVE STOCK:
Type:
Amount:
Location:
Feed type:
Feed On Hand/kg:
Arriving Weight/kg:
Selling Weight/kg:
Cost Price:
Selleng Price:
Arriving Date:
Selling Date:
Amount Sold:
Add

Databases

	UserID	UserName	Password
▷	1	Farmer223	12345
	2	Farmer332	54321
	3	GuessUser	15937
	4	Worker025	95137
	5	Worker114	42685
	7	Waldo21	1123
	9	Mark18	123456
⚙	*NULL*	*NULL*	*NULL*

	Implement	Description	CostToPurchase	SellengPrice	DateOfPurchase	Condition	Repair
▷	A300 GM Gyrorake	Rake	72000.0000	20000.3300	03/05/2017	Fair	Y
	Implemnts Eco Seeder ETA 18W	Planter	316999.9900	260021.6600	15/06/2017	Excelent	N
	JD Spuit 4730	Sprayer	850000.9900	219999.9900	22/03/2011	Good	Y
	JX Jupidex 11Tand	Cultivator	317100.3300	189999.9900	30/08/2012	Good	Y
	JX Jupidex CTC 427	Cultivator	499999.9900	100000.3300	03/02/3017	Fair	N
⚙	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

	LiveStockType	LiveStockAmo...	LiveStockLoca...	TypeOfFeed	FeedOnHand_...	ArrivingWeigh...	SellingWeight_...	BoughtFor	SellengFor	ArrivingDate	SellingDate	AmountSold
▷	Bosmare Cow 211C	426.0000	Feedingstaion-...	TypeOfFeedDry...	11002	355	560	3521.8800	9069.3300	01/02/2019 00:0...	04/07/2019	0
	Chester White Pig 311P	98.0000	FeedingStaion-...	TypeOfFeedMo...	21045	65	95	984.2200	1120.9900	02/01/2019 00:0...	29/05/2019	0
	Dorper Sheep 121S	265.0000	BlockK-GrassFi...	TEFF	892	36	51	1123.6900	6032.2200	15/01/2019 00:0...	11/07/2019	0
	Landrace Pig 321P	36.0000	FeedingStaion-...	TypeOfFeedMo...	25410	88	91	752.3300	801.2200	01/03/2019 00:0...	03/05/2019	1233.21
	Large white Pig 331P	22.0000	FeedingSation-SI	Korn Kandy V15...	2310	85	96	3650.2200	4300.2200	02/02/2019 00:0...	31/05/2019	0
	Merino Sheep 111S	636.0000	BlockA-GrassFi...	TEFF	850	50	72	955.3300	2111.0000	03/05/2019 00:0...	02/08/2019	0
	Nguni Cow 221C	111.0000	BlockF-GrassFie...	TEFF	1120	360	630	6251.6600	1125.3300	31/01/2019 00:0...	18/09/2019	0
⚙	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

	GrainType	GrainOnHand_...	CostPrice_pTon	SellingPrice_p...	GrainLocation	TypeFertilizer	FertilizerOnHa...	CostFertilizer_...	FarmPlanted
▷	CornV2-2586	400	489.33	2510.33	StoreA4	UREA46%N	3600	883.56	Block-B
	Hybrid Maize 3M-01	952	650.99	1100.66	Store-KL2	CUAN	4500	600.25	BETA
	Monceren Corn	490	980.24	1200.33	SiloRoom-3	NPK4|2|3	5000	392.66	Uitzhoek
	Soya B-k43I4	380	366.87	2360.22	Store -A2	DAP8|15|36	1235	930.88	Hermanus
	Sunflow BK_sI908	982	755.36	4366.22	SiloRoom-3	NPK9|13|22	3302	380.22	Paradys
⚙	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

	WorkerName	JobDescription	AessetsAssigned	Availability	HoursWorked	Rate_Hour	Weekly_Mont...
▷	Jan	522	22	22	22	22.0000	22.0000
	Piet	652	62	62	62	644.0000	6541.0000
⚙	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

Group Experience

Problems

Mainly the coding had a few problems as well as our laptops having technical errors.

Frustrations

Out laptops experiencing technical errors and unable to connect with databases.

Solutions

Created new database in sql.

Advantages

Having two team members and two laptops to work with.

Disadvantages

Our visual studio was faulty.

The program/website is only for farmers.