DB LAB PROJECT

22F-3068

22F-8772

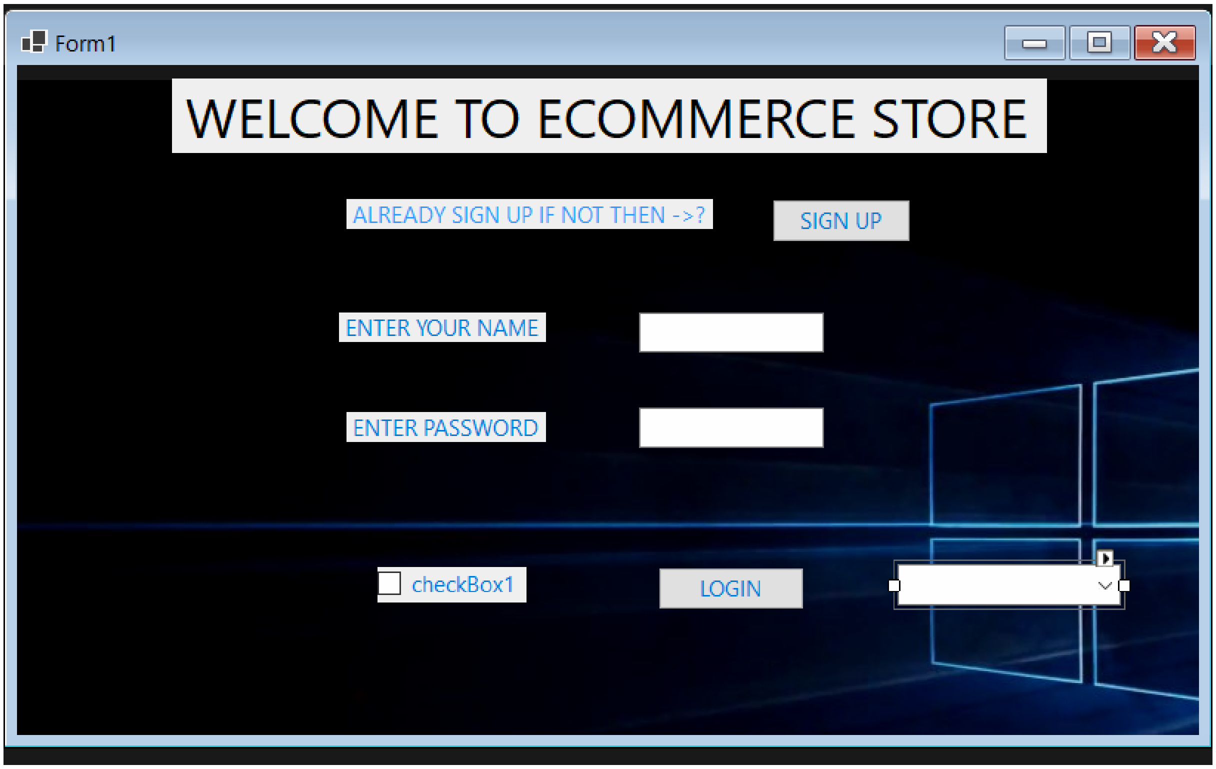
22F-3430

E-COMMERCE

```
using System;
using System.Windows.Forms;
using Oracle.ManagedDataAccess.Client;

namespace WinFormsApp8
{
    public partial class Form1 : Form
    {
        // Oracle database connection string
        private string connectionString = "User Id=MUBASH;Password=789;Data
Source=localhost:1521/xe";

        public Form1()
        {
            InitializeComponent();
        }

        // Event handler for button click
        private void button1_Click(object sender, EventArgs e)
        {
            string username = textBox1.Text.Trim(); // Assuming textBox1 is for username
            string password = textBox2.Text.Trim(); // Assuming textBox2 is for password
            bool isChecked = checkBox1.Checked; // Check if checkbox is checked
            string selectedRole = comboBox1.SelectedItem.ToString(); // Get the selected role from comboBox1

            // Check if the checkbox is checked first
            if (isChecked)
            {
                // Validate username, password, and role from Oracle database
                string roleFromDb = ValidateCredentials(username, password);

                if (!string.IsNullOrEmpty(roleFromDb) && roleFromDb == selectedRole)
                {
                    // If the role from the DB matches the selected role in ComboBox
                    if (selectedRole == "Admin")
                    {
                        // Open Form10 if role is Admin
                        Form10 form10 = new Form10();
                        form10.Show();
                        this.Hide(); // Hide the current form
                    }
                    else if (selectedRole == "Customer")
                    {
                        // Open Form9 if role is Customer
                        Form2 form2 = new Form2();
                        form2.Show();
                        this.Hide(); // Hide the current form
                    }
                }
                else
                {
                    // Show error message if credentials or role doesn't match
                    MessageBox.Show("Invalid username, password, or role!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
            }
            else
            {
                // Show error message if the checkbox is not checked
                MessageBox.Show("Please check the 'Check me' checkbox.", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            }
        }

        // Method to validate the credentials and get the role from the Oracle database
        private string ValidateCredentials(string username, string password)
        {
            try
            {
                // Create a connection to the Oracle database
                using (OracleConnection conn = new OracleConnection(connectionString))
                {
                    conn.Open();
                    // SQL query to get the role based on the username and password
                    string query = "SELECT Role FROM Users WHERE LOWER(Username) = LOWER(:userName) AND Password = :password";

                    using (OracleCommand cmd = new OracleCommand(query, conn))
                    {
                        // Add parameters to prevent SQL injection
                        cmd.Parameters.Add(new OracleParameter("userName", username));
                        cmd.Parameters.Add(new OracleParameter("password", password));

                        // Execute the query and get the role
                        object result = cmd.ExecuteScalar();
                        if (result != null)
                        {
                            return result.ToString(); // Return the role (admin or customer)
                        }
                    }
                }
            }
            catch (OracleException ex)
            {
                // Handle any database errors
                MessageBox.Show($"Database error: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            catch (Exception ex)
            {
                MessageBox.Show($"An error occurred: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }

            return null; // Return null if credentials are not valid
        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            // Optional: Add logic if needed when the username text changes
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            // Optional: Add logic if needed when the password text changes
        }

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            // Optional: Add logic if needed when the checkbox state changes
        }

        private void label1_Click(object sender, EventArgs e)
        {
            // Optional: Add logic if needed when the label is clicked
        }
```

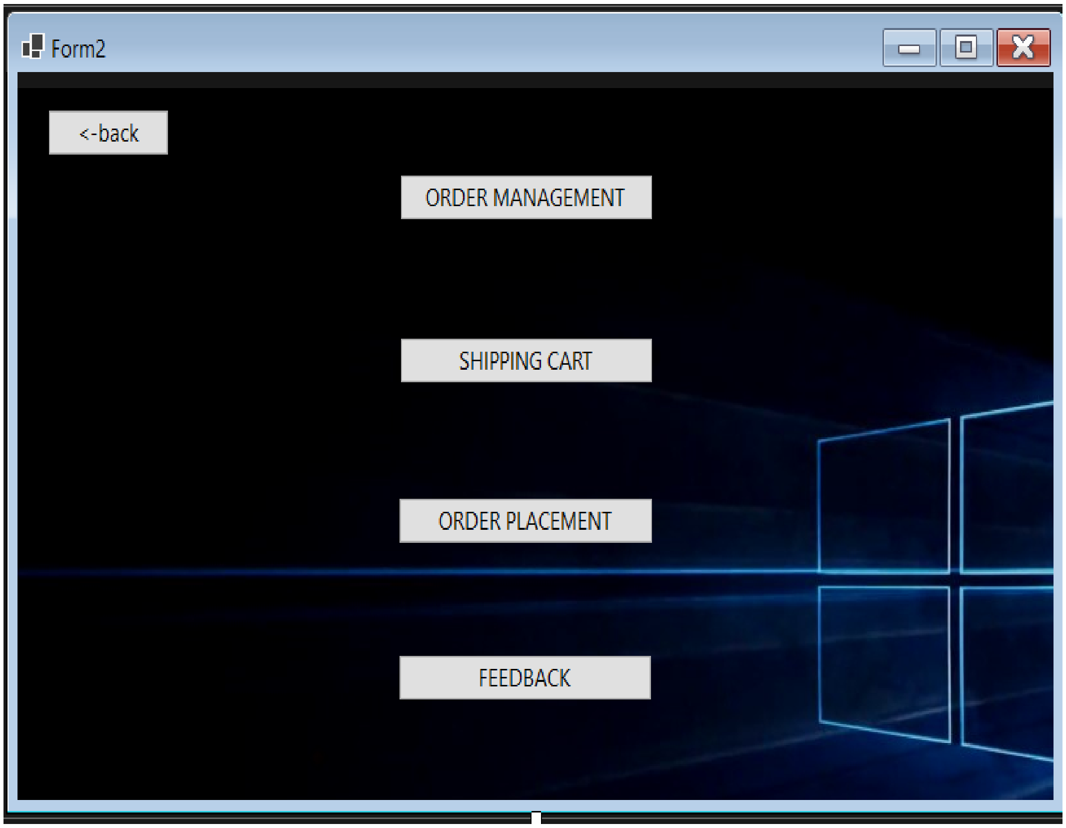
```
private void Form1_Load(object sender, EventArgs e)
{
    // Optional: Add logic if needed when the form loads
}

private void button2_Click(object sender, EventArgs e)
{
    // Create an instance of Form12
    Form12 form12 = new Form12();

    // Show Form12
    form12.Show();

    // Optionally, you can hide the current form (if needed)
    this.Hide();
}

private void label4_Click(object sender, EventArgs e)
{
    // Optional: Add logic if needed when the label is clicked
}

private void label4_Click_1(object sender, EventArgs e)
{
    // Optional: Add logic if needed when the label is clicked
}
```

```
        private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
        {
            // Optional: Add logic if needed when the comboBox selection changes
        }
    }
}
```



```
//FORM 222222
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinFormsApp8
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Show Form1
            Form1 form1 = new Form1();
            form1.Show();

            // Close the current Form2
            this.Close();
        }

        private void button4_Click(object sender, EventArgs e)
        {
            Form5 form5 = new Form5();

            // Show Form3
            form5.Show();

            // Optionally close the current Form2
            this.Close(); // Closes Form2
        }

        private void Form2_Load(object sender, EventArgs e)
        {

        }





        private void button5_Click(object sender, EventArgs e)
        {
            Form6 form6 = new Form6();

            // Show Form3
            form6.Show();

            // Optionally close the current Form2
            this.Close(); // Closes Form2
        }

        private void button6_Click(object sender, EventArgs e)
```

```
        {
            Form7 form7 = new Form7();

            // Show Form3
            form7.Show();

            // Optionally close the current Form2
            this.Close(); // Closes Form2
        }

        private void button7_Click(object sender, EventArgs e)
        {
            Form8 form8 = new Form8();

            // Show Form3
            form8.Show();

            // Optionally close the current Form2
            this.Close(); // Closes Form2
        }
    }
}
```

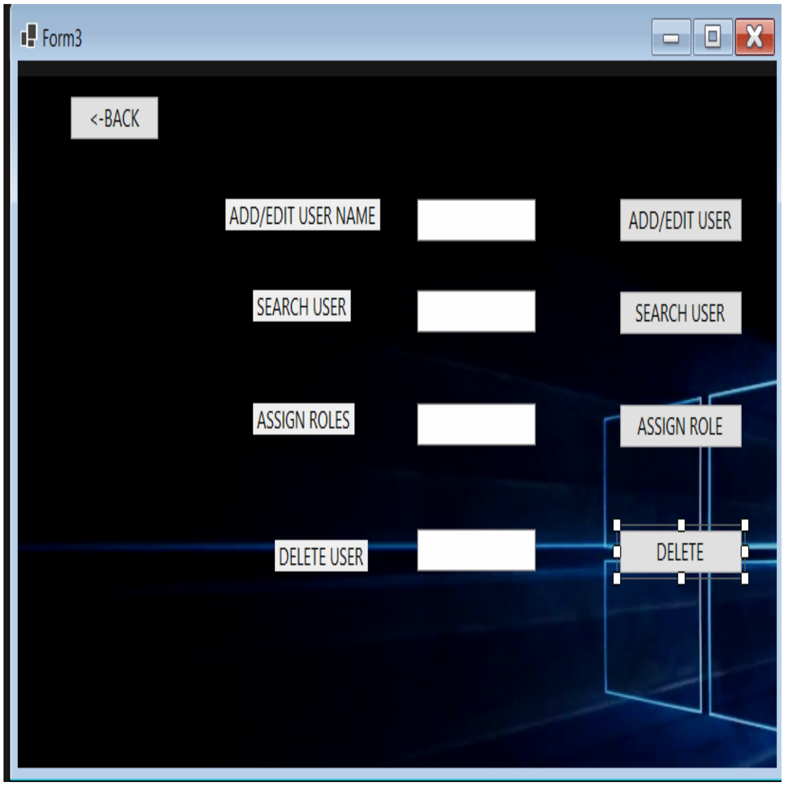


```
//FORM 33333333333333

using System;

using System.Windows.Forms;

using Oracle.ManagedDataAccess.Client;


namespace WinFormsApp8

{

    public partial class Form3 : Form

    {

        // Oracle database connection string

        private string connectionString = "User Id=MUBASH;Password=789;Data Source=localhost:1521/xe";


        public Form3()
```

```
{
    InitializeComponent();
}

// Event handler to add/edit user
private void button1_Click(object sender, EventArgs e)
{
    // Create an instance of Form10
    Form10 form10 = new Form10();

    // Show Form10
    form10.Show();

    // Close the current form
    this.Close();
}



// Method to add or edit user in the database
private bool AddOrEditUser(string username, string password, string role)
{
    try
    {
        using (OracleConnection conn = new OracleConnection(connectionString))
        {
            conn.Open();
            // Check if the user already exists
            string checkQuery = "SELECT COUNT(*) FROM Users WHERE LOWER(Username) = LOWER(:username)";
            using (OracleCommand checkCmd = new OracleCommand(checkQuery, conn))
            {
                checkCmd.Parameters.Add(new OracleParameter("username", username));
                int count = Convert.ToInt32(checkCmd.ExecuteScalar());

                string query;
                if (count == 0)
                {
                    // If user doesn't exist, insert a new user
                    query = "INSERT INTO Users (Username, Password, Role) VALUES (:username, :password, :role)";
                }
                else
                {
                    // If user exists, update the user
                    query = "UPDATE Users SET Password = :password, Role = :role WHERE LOWER(Username) = LOWER(:username)";
                }

                using (OracleCommand cmd = new OracleCommand(query, conn))
                {
                    cmd.Parameters.Add(new OracleParameter("username", username));
                    cmd.Parameters.Add(new OracleParameter("password", password));
                    cmd.Parameters.Add(new OracleParameter("role", role));
```

```
                            int rowsAffected = cmd.ExecuteNonQuery();
                            return rowsAffected > 0; // Return true if the query was successful
                        }
                    }
                }
            }
            catch (OracleException ex)
            {
                MessageBox.Show($"Database error: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
        }

        // Event handler for Button 2 (Edit User)
        private void button2_Click(object sender, EventArgs e)
        {
            string username = textBox1.Text.Trim();
            if (!string.IsNullOrEmpty(username))
            {
                bool isSuccess = EditUsername(username);
                if (isSuccess)
                {
                    MessageBox.Show("Username edited successfully.");
                }
                else
                {
                    MessageBox.Show("Error editing username.");
                }
            }
            else
            {
                MessageBox.Show("Please enter a valid username.");
            }
        }

        // Method to edit the username in the database
        private bool EditUsername(string username)
        {
            try
            {
                using (OracleConnection conn = new OracleConnection(connectionString))
                {
                    conn.Open();

                    string query = "UPDATE Users SET Username = :username WHERE
LOWER(Username) = LOWER(:username)";
                    using (OracleCommand cmd = new OracleCommand(query, conn))
                    {
                        cmd.Parameters.Add(new OracleParameter("username", username));

                        int rowsAffected = cmd.ExecuteNonQuery();
                        return rowsAffected > 0; // Return true if the query was successful
                    }
                }
            }
            catch (OracleException ex)
            {
                MessageBox.Show($"Database error: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
        }

        // Event handler for Button 3 (Search User)
        private void button3_Click(object sender, EventArgs e)
        {
            string username = textBox2.Text.Trim();
            if (!string.IsNullOrEmpty(username))
            {
                SearchUser(username);
            }
            else
            {
                MessageBox.Show("Please enter a username to search.");
            }
        }

        // Method to search for a user in the database
        private void SearchUser(string username)
```

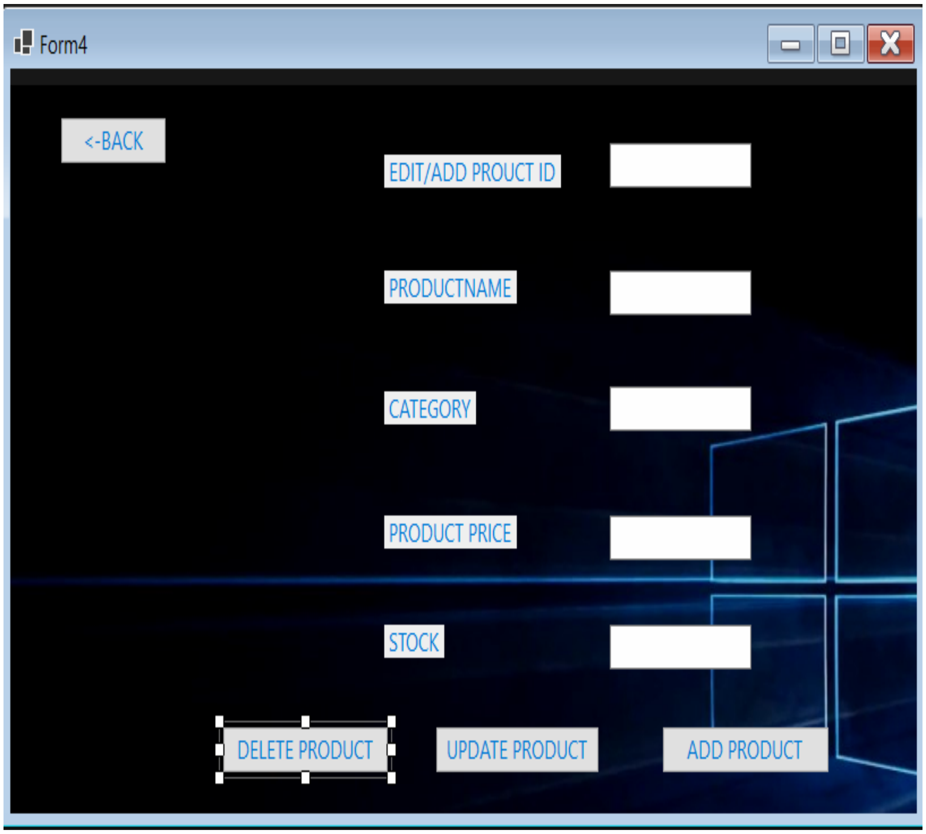
```
        {
            try
            {
                using (OracleConnection conn = new OracleConnection(connectionString))
                {
                    conn.Open();

                    // SQL query to search for a user by username
                    string query = "SELECT * FROM Users WHERE LOWER(Username) =
LOWER(:username)";
                    using (OracleCommand cmd = new OracleCommand(query, conn))
                    {
                        cmd.Parameters.Add(new OracleParameter("username", username));

                        using (OracleDataReader reader = cmd.ExecuteReader())
                        {
                            if (reader.Read())
                            {
                                // Display user details in Textboxes
                                textBox1.Text = reader["Username"].ToString(); // Show username
in textBox1
                                textBox2.Text = reader["Password"].ToString(); // Show password in
textBox2
                                textBox3.Text = reader["Role"].ToString(); // Show role in textBox3
                                MessageBox.Show("User found.");
                            }
                            else
                            {
                                MessageBox.Show("User not found.", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                            }
                        }
                    }
                }
            }
            catch (OracleException ex)
            {
                MessageBox.Show($"Database error: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }

        // Event handler for Button 4 (Assign Role)
        private void button4_Click(object sender, EventArgs e)
        {
            string username = textBox3.Text.Trim();
            string role = textBox2.Text.Trim();
            if (!string.IsNullOrEmpty(username) && !string.IsNullOrEmpty(role))
            {
                bool isSuccess = AssignRole(username, role);
                if (isSuccess)
                {
                    MessageBox.Show("Role assigned successfully.");
                }
                else
                {
                    MessageBox.Show("Error assigning role.");
                }
            }
            else
            {
                MessageBox.Show("Please enter both username and role.");
            }
        }

        // Method to assign a role to a user in the database
        private bool AssignRole(string username, string role)
        {
            try
            {
                using (OracleConnection conn = new OracleConnection(connectionString))
                {
                    conn.Open();

                    string query = "UPDATE Users SET Role = :role WHERE
    LOWER(Username) = LOWER(:username)";
                    using (OracleCommand cmd = new OracleCommand(query, conn))
                    {
                        cmd.Parameters.Add(new OracleParameter("role", role));
                        cmd.Parameters.Add(new OracleParameter("username", username));

                        int rowsAffected = cmd.ExecuteNonQuery();
                        return rowsAffected > 0; // Return true if the query was successful
                    }
                }
            }
            catch (OracleException ex)
            {
                MessageBox.Show($"Database error: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
        }
        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            // Your code for textBox1 changes
        }
        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            // Your code for textBox1 changes
        }
        private void textBox3_TextChanged(object sender, EventArgs e)
        {
            // Your code for textBox1 changes
        }
        // Event handler for Button 5 (Delete User)
        private void button5_Click(object sender, EventArgs e)
        {
            string username = textBox4.Text.Trim();
            if (!string.IsNullOrEmpty(username))
            {
                bool isSuccess = DeleteUser(username);
                if (isSuccess)
                {
                    MessageBox.Show("User deleted successfully.");
                }
                else
                {
                    MessageBox.Show("Error deleting user or user cannot be deleted because they are an Admin.");
                }
            }
            else
            {
                MessageBox.Show("Please enter a username to delete.");
            }
        }

        // Method to delete a user from the database
        private bool DeleteUser(string username)
        {
            try
            {
                using (OracleConnection conn = new OracleConnection(connectionString))
                {
                    conn.Open();
                // Check if the user has the role 'Admin'
                string checkRoleQuery = "SELECT Role FROM Users WHERE
LOWER(Username) = LOWER(:username)";
                using (OracleCommand checkRoleCmd = new
OracleCommand(checkRoleQuery, conn))
                {
                    checkRoleCmd.Parameters.Add(new OracleParameter("username",
username));

                    object roleResult = checkRoleCmd.ExecuteScalar();
                    if (roleResult != null && roleResult.ToString().Equals("Admin",
StringComparison.OrdinalIgnoreCase))
                    {
                        // If the user is an Admin, do not allow deletion
                        MessageBox.Show("User with the role 'Admin' cannot be deleted.",
"Error", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                        return false;
                    }
                }

                // If the user is not an Admin, proceed with deletion
                string query = "DELETE FROM Users WHERE LOWER(Username) =
LOWER(:username)";
                using (OracleCommand cmd = new OracleCommand(query, conn))
                {
                    cmd.Parameters.Add(new OracleParameter("username", username));

                    int rowsAffected = cmd.ExecuteNonQuery();
                    return rowsAffected > 0; // Return true if the query was successful
                    }
                }
            }
            catch (OracleException ex)
            {
                MessageBox.Show($"Database error: {ex.Message}", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);
                return false;
            }
        }


        private void textBox4_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```



```
//FORM 4444444444

using System;

using System.Data;

using System.Windows.Forms;

using Oracle.ManagedDataAccess.Client; // Include this namespace for Oracle operations


namespace WinFormsApp8

{

    public partial class Form4 : Form

    {

        // Connection string for the Oracle database

        private string connectionString = "User Id=MUBASH;Password=789;Data Source=localhost:1521/xe";
        public Form4()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Navigate to another form
            Form10 form10 = new Form10();
            form10.Show();
            this.Close();
        }

        private void button3_Click(object sender, EventArgs e) // ADD PRODUCT
        {
            try
            {
                using (OracleConnection connection = new
OracleConnection(connectionString))
                {
                    connection.Open();


                    string query = "INSERT INTO Products (ProductID, ProductName,
Category, Price, Stock) " +
                                   "VALUES (:ProductID, :ProductName, :Category, :Price, :Stock)";
                    using (OracleCommand command = new OracleCommand(query,
connection))
                    {
                        command.Parameters.Add(":ProductID", OracleDbType.Int32).Value =
int.Parse(textBox4.Text);
                        command.Parameters.Add(":ProductName",
OracleDbType.Varchar2).Value = textBox1.Text;
                        command.Parameters.Add(":Category", OracleDbType.Varchar2).Value
= textBox2.Text;
                        command.Parameters.Add(":Price", OracleDbType.Decimal).Value =
decimal.Parse(textBox3.Text);
                        command.Parameters.Add(":Stock", OracleDbType.Int32).Value =
int.Parse(textBox5.Text);

                        command.ExecuteNonQuery();
                        MessageBox.Show("Product added successfully.");
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error adding product: " + ex.Message);
            }
        }

        private void button4_Click(object sender, EventArgs e) // DELETE PRODUCT
        {
            try
            {
                using (OracleConnection connection = new
OracleConnection(connectionString))
                {
                    connection.Open();


                    string checkQuery = "SELECT COUNT(*) FROM Products WHERE
ProductID = :ProductID";
                    using (OracleCommand checkCommand = new
OracleCommand(checkQuery, connection))
                    {
                        checkCommand.Parameters.Add(":ProductID",
OracleDbType.Int32).Value = int.Parse(textBox4.Text);
                        int count = Convert.ToInt32(checkCommand.ExecuteScalar());


                        if (count > 0)
                        {
                            string deleteQuery = "DELETE FROM Products WHERE ProductID =
:ProductID";
                            using (OracleCommand deleteCommand = new
OracleCommand(deleteQuery, connection))
                            {
                                deleteCommand.Parameters.Add(":ProductID",
OracleDbType.Int32).Value = int.Parse(textBox4.Text);
                                deleteCommand.ExecuteNonQuery();
                                MessageBox.Show("Product deleted successfully.");
                            }
                        }
                        else
                        {
                            MessageBox.Show("Product not found.");
                        }
                    }
```

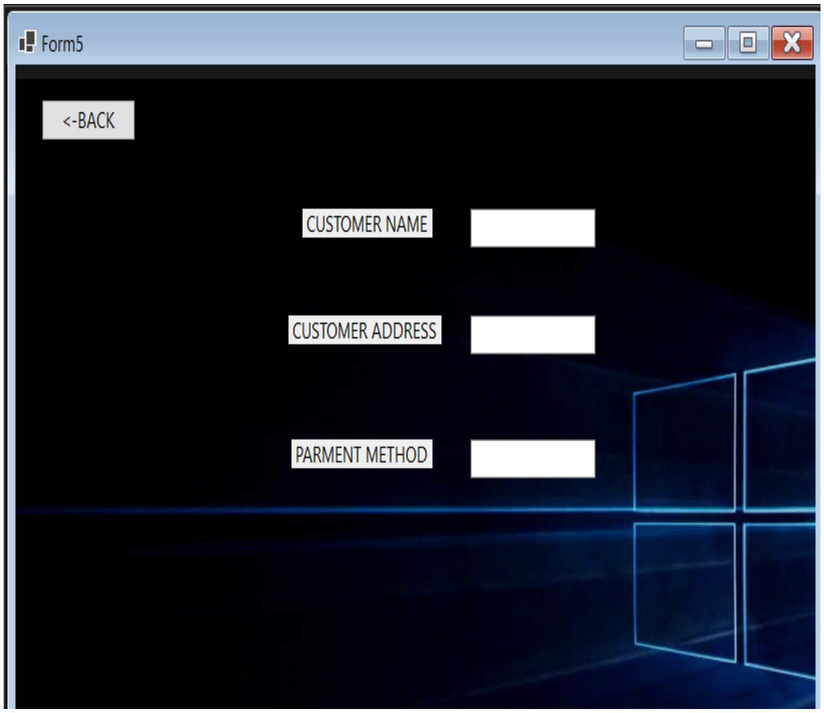
```
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error deleting product: " + ex.Message);
            }
        }

        private void button2_Click(object sender, EventArgs e) // UPDATE PRODUCT
        {
            try
            {
                using (OracleConnection connection = new
OracleConnection(connectionString))
                {
                    connection.Open();

                    // Query to check if the product exists with matching ProductID,
ProductName, Category, and Stock
                    string checkQuery = "SELECT COUNT(*) FROM Products " +
                                        "WHERE ProductID = :ProductID AND ProductName =
:ProductName " +
                                        "AND Category = :Category AND Stock = :Stock";
                    using (OracleCommand checkCommand = new
OracleCommand(checkQuery, connection))
                    {
                        // Add parameters for validation
                        checkCommand.Parameters.Add(":ProductID",
OracleDbType.Int32).Value = int.Parse(textBox4.Text);
```

```
                        checkCommand.Parameters.Add(":ProductName", OracleDbType.Varchar2).Value = textBox1.Text;
                        checkCommand.Parameters.Add(":Category", OracleDbType.Varchar2).Value = textBox2.Text;
                        checkCommand.Parameters.Add(":Stock", OracleDbType.Int32).Value = int.Parse(textBox5.Text);

                        // Execute the query to check if the record exists
                        int count = Convert.ToInt32(checkCommand.ExecuteScalar());

                        if (count > 0)
                        {
                            // Update query to modify only the Price
                            string updateQuery = "UPDATE Products SET Price = :Price " +
                                                 "WHERE ProductID = :ProductID AND ProductName = :ProductName " +
                                                 "AND Category = :Category AND Stock = :Stock";
                            using (OracleCommand updateCommand = new OracleCommand(updateQuery, connection))
                            {
                                // Add parameters for the update query
                                updateCommand.Parameters.Add(":Price", OracleDbType.Decimal).Value = decimal.Parse(textBox3.Text);
                                updateCommand.Parameters.Add(":ProductID", OracleDbType.Int32).Value = int.Parse(textBox4.Text);
                                updateCommand.Parameters.Add(":ProductName", OracleDbType.Varchar2).Value = textBox1.Text;
                                updateCommand.Parameters.Add(":Category", OracleDbType.Varchar2).Value = textBox2.Text;
                                updateCommand.Parameters.Add(":Stock",
OracleDbType.Int32).Value = int.Parse(textBox5.Text);


                                // Execute the update command
                                updateCommand.ExecuteNonQuery();
                                MessageBox.Show("Product price updated successfully.");
                            }
                        }
                        else
                        {
                            MessageBox.Show("Product not found with the specified details.");
                        }
                    }
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("Error updating product: " + ex.Message);
            }
        }


        private void textBox4_TextChanged(object sender, EventArgs e) { } // Product ID
        private void textBox1_TextChanged(object sender, EventArgs e) { } // Product
Name
        private void textBox2_TextChanged(object sender, EventArgs e) { } // Product
Category
        private void textBox3_TextChanged(object sender, EventArgs e) { } // Product Price
        private void textBox5_TextChanged(object sender, EventArgs e) { } // Product Stock

        private void label5_Click(object sender, EventArgs e) { }

    }

}
```



```
//FORM 55555555555\

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;
```

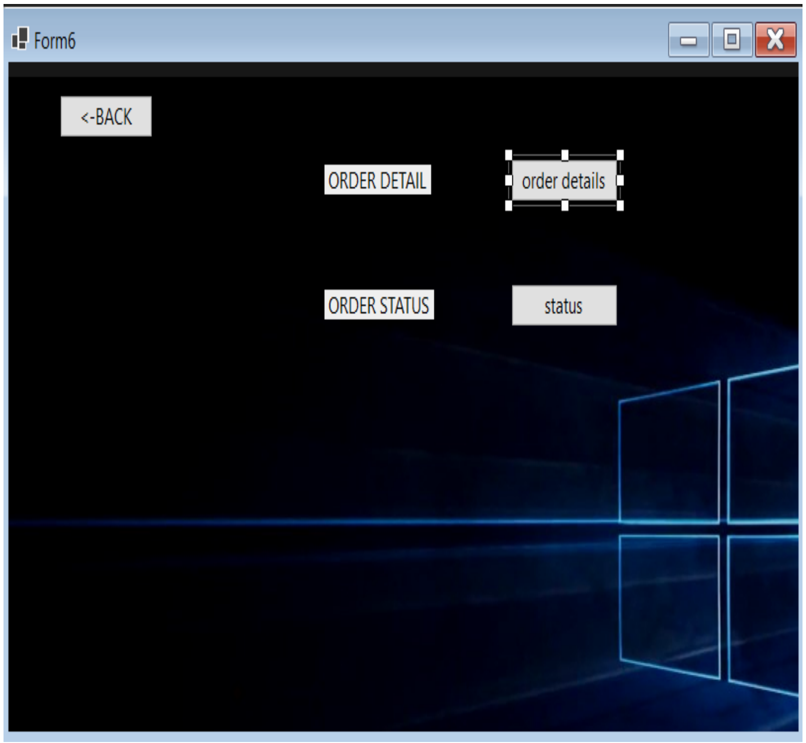
```
}


using System.Windows.Forms;


namespace WinFormsApp8
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();
        }


        private void button1_Click(object sender, EventArgs e)
        {
            // Create a new instance of Form2
            Form2 form2 = new Form2();


            // Show Form2
            form2.Show();


            // Close the current form (e.g., Form3)
            this.Close();
        }


        private void textBox1_TextChanged(object sender, EventArgs e)
        {


        }
    }
}
```



```
//FORM 6666666
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WinFormsApp8
{
    public partial class Form6 : Form
    {
        // Simulating a shared order data structure.
        private static List<Order> Orders = new List<Order>();
```

```
public Form6()
{
    InitializeComponent();
}

// Simulated method to add orders (called from Form7 or elsewhere).
public static void AddOrder(Order newOrder)
{
    Orders.Add(newOrder);
}

private void button1_Click(object sender, EventArgs e)
{
    // Navigate to Form2
    Form2 form2 = new Form2();
    form2.Show();
    this.Close();
}







private void button2_Click(object sender, EventArgs e)//ORSER DETAIL
```

```
        {

        }

        private void button3_Click(object sender, EventArgs e)//ORDER STATUS
        {

        }




    }

    // Simulated Order class to store order details.
    public class Order
    {
        public int OrderID { get; set; }
        public string ProductName { get; set; }
        public int Quantity { get; set; }
        public decimal Total { get; set; }
        public string Status { get; set; } = "Processing"; // Default status.
    }
}
```



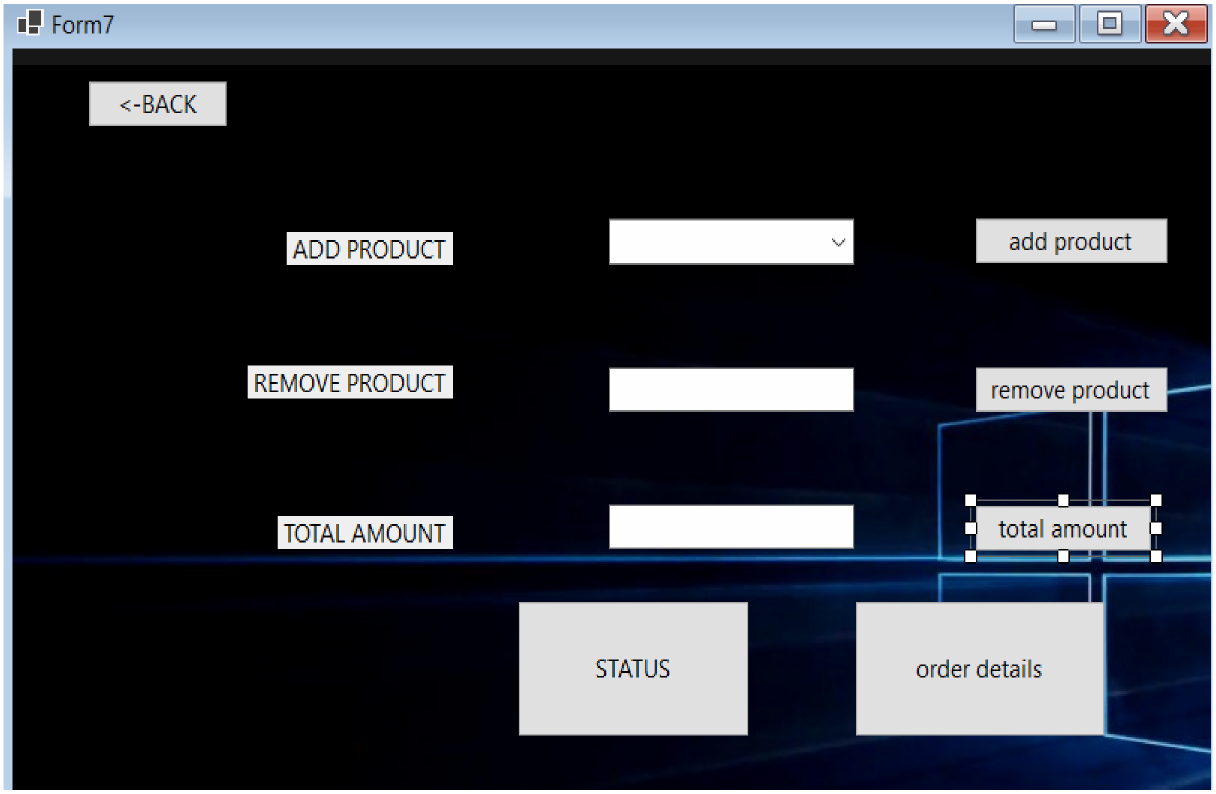
```
using System;

using System.Collections.Generic;

using System.Data;

using System.Windows.Forms;


namespace WinFormsApp8

{

   public partial class Form7 : Form

   {

      // Dictionary to hold product details (ProductName -> (Price, Stock))

      private Dictionary<string, (decimal Price, int Stock)> products = new
Dictionary<string, (decimal Price, int Stock)>

      {

         { "Wireless Headphones", (59.99m, 100) },

         { "Smartphone", (699.99m, 50) },
    { "Gaming Laptop", (1199.99m, 20) },
    { "Office Chair", (149.99m, 30) },
    { "Running Shoes", (89.99m, 75) },
    { "Coffee Maker", (49.99m, 40) },
    { "Fitness Tracker", (129.99m, 60) }
};

// List to hold selected products
private List<string> cart = new List<string>();

public Form7()
{
    InitializeComponent();
}

private void Form7_Load(object sender, EventArgs e)
{
    // Populate the ComboBox with product names
    comboBox1.Items.AddRange(products.Keys.ToArray());
}

private void button2_Click(object sender, EventArgs e) // ADD PRODUCT
{
    // Add selected product to the cart
    string selectedProduct = comboBox1.SelectedItem?.ToString();
    if (!string.IsNullOrEmpty(selectedProduct))
    {
```

```
                cart.Add(selectedProduct);
                MessageBox.Show($"{selectedProduct} has been added to the cart.");
            }
            else
            {
                MessageBox.Show("Please select a product to add.");
            }
        }

        private void button3_Click(object sender, EventArgs e) // REMOVE PRODUCT
        {
            // Remove product from the cart based on TextBox1 input
            string productToDelete = textBox1.Text.Trim();
            if (!string.IsNullOrEmpty(productToDelete) && cart.Contains(productToDelete))
            {
                cart.Remove(productToDelete);
                MessageBox.Show($"{productToDelete} has been removed from the cart.");
            }
            else
            {
                MessageBox.Show("Product not found in the cart or input is invalid.");
            }
        }

        private void button4_Click(object sender, EventArgs e) // CALCULATE TOTAL AMOUNT
        {
```

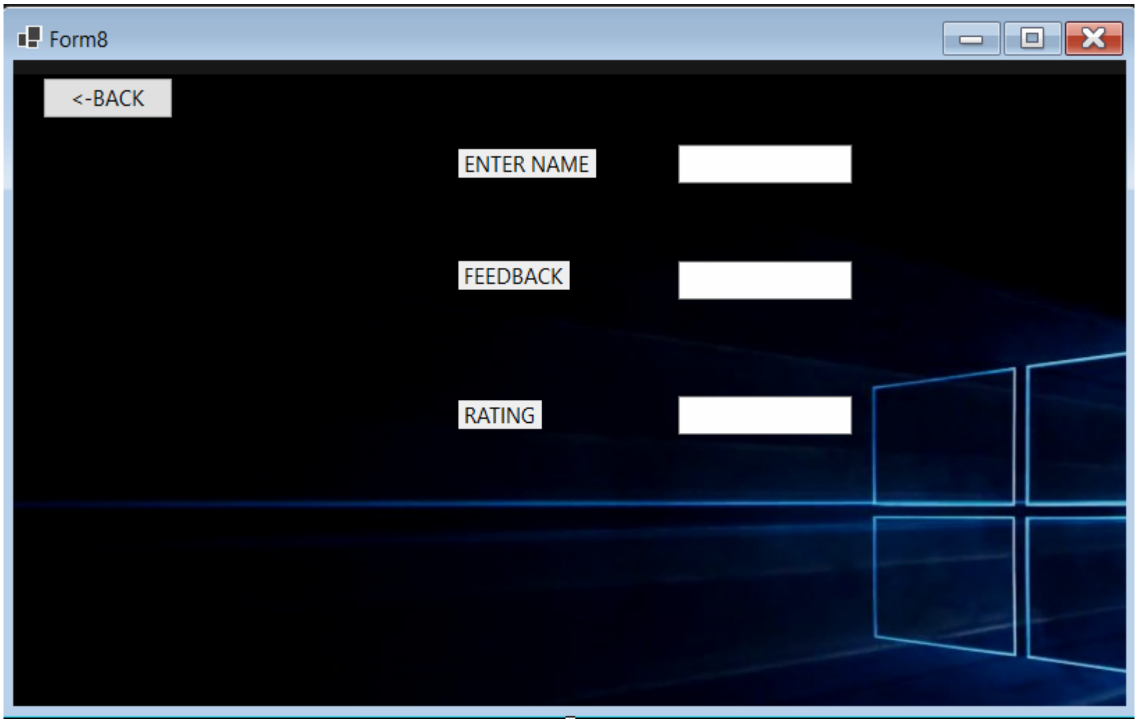
```
    // Calculate the total price of products in the cart
    decimal totalPrice = 0;
    foreach (var product in cart)
    {
        totalPrice += products[product].Price;
    }

    MessageBox.Show($"Total price of selected products: {totalPrice:C}");
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    // Add logic for when the selected item in comboBox1 changes (if needed).
}

private void button1_Click(object sender, EventArgs e)
{
    // Logic for navigating to Form2.
    Form2 form2 = new Form2();
    form2.Show();
    this.Close();
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    // Logic for TextBox1 text change (if needed).
}
```

```
private void textBox2_TextChanged(object sender, EventArgs e)
{
    // Logic for TextBox2 text change (if needed).
}
private void button6_Click(object sender, EventArgs e)
{
    MessageBox.Show("PENDING");
}

private void button5_Click(object sender, EventArgs e)
{
    if (cart.Count == 0)
    {
        MessageBox.Show("No products have been added to the cart.");
        return;
    }

    // Initialize a string to hold order details
    string orderDetails = "Order Details:\n\n";

    decimal totalPrice = 0;

    // Loop through the cart to show each product, its price, and stock status
    foreach (var product in cart)
    {
        decimal price = products[product].Price;
        int stock = products[product].Stock;
```

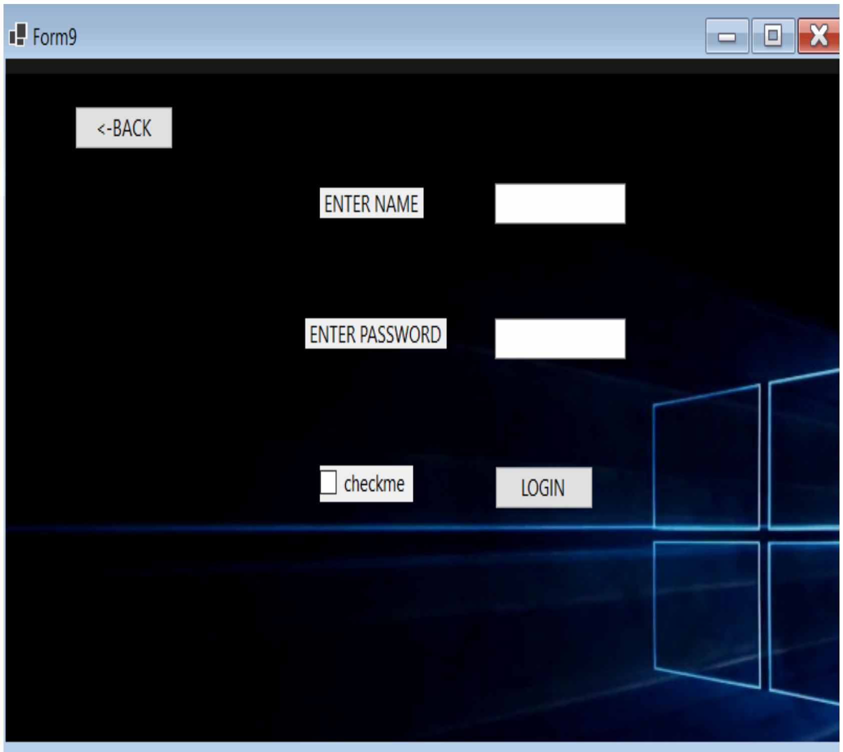
```
                // Add product details to the string
                orderDetails += $"{product} - Price: {price:C}, Stock: {stock}\n";

                // Add the price to the total
                totalPrice += price;
            }

            // Add the total price to the details string
            orderDetails += $"\nTotal Price: {totalPrice:C}";

            // Show the order details in a MessageBox
            MessageBox.Show(orderDetails, "Order Details");
        }

    }
}
```



```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;


namespace WinFormsApp8

{

    public partial class Form8 : Form

    {
        public Form8()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Create a new instance of Form2
            Form2 form2 = new Form2();

            // Show Form2
            form2.Show();

            // Close the current form (e.g., Form3)
            this.Close();
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }
    }
}
```

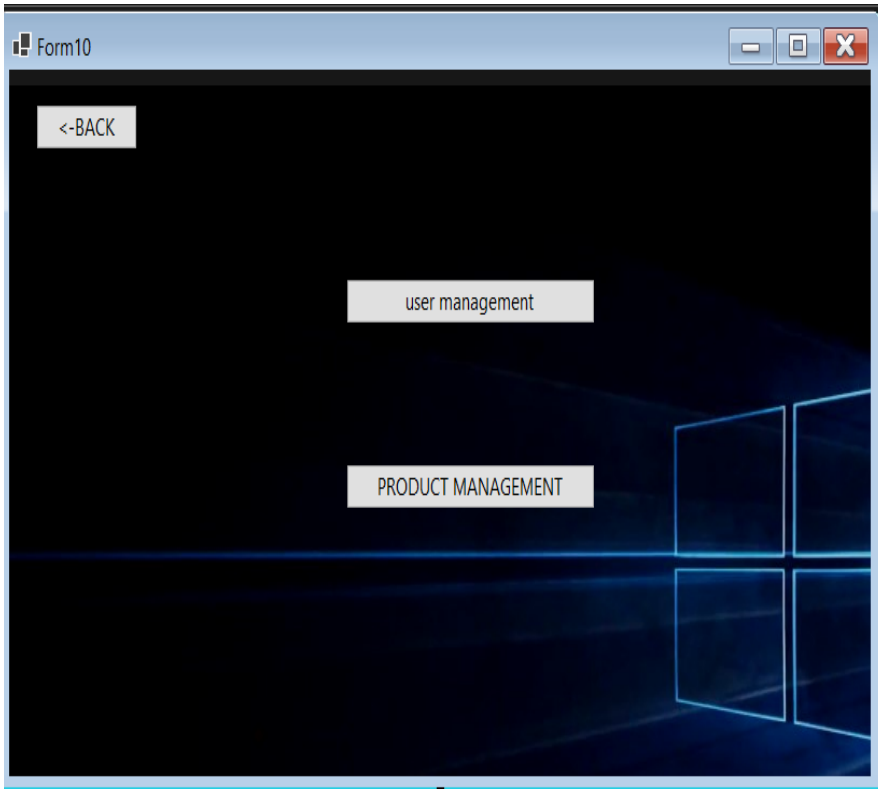


```
using System;
using System.Data;
using System.Windows.Forms;
using Oracle.ManagedDataAccess.Client;

namespace WinFormsApp8
{
    public partial class Form9 : Form
    {
        public Form9()
        {
            InitializeComponent();
        }

        // Oracle database connection string
        private string connectionString = "User Id=MUBASH;Password=789;Data Source=localhost:1521/xe";

        private void button1_Click(object sender, EventArgs e)
        {
            // Get the values entered in the textboxes and checkbox
            string userName = textBox1.Text;
            string password = textBox2.Text;
            bool isChecked = checkBox1.Checked;

            // Check if the "Check me" checkbox is checked
            if (isChecked)
            {
                // Validate the credentials against the USERS table
                if (ValidateCredentials(userName, password))
                {
                    // If valid and role is Admin, open Form10
                    Form10 form10 = new Form10();
                    form10.Show();
                    this.Hide(); // Hide Form9
                }
                else
                {
                    MessageBox.Show("Invalid credentials or role is not Admin.");
                }
            }
            else
            {
                MessageBox.Show("Please check the 'Check me' box.");
            }
        }

        // Method to validate the credentials from the USERS table
        private bool ValidateCredentials(string userName, string password)
        {
            try
            {
                using (OracleConnection conn = new OracleConnection(connectionString))
                {
                    conn.Open();
                    string query = "SELECT Role FROM Users WHERE Username =
:userName AND Password = :password";

                    using (OracleCommand cmd = new OracleCommand(query, conn))
                    {
                        cmd.Parameters.Add(new OracleParameter("userName", userName));
                        cmd.Parameters.Add(new OracleParameter("password", password));

                        // Execute the query and get the role
                        object result = cmd.ExecuteScalar();
                        if (result != null && result.ToString() == "Admin")
                        {
                            return true; // Role is Admin
                        }
                    }
                }
            }
            catch (OracleException ex)
            {
                MessageBox.Show($"Database error: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            catch (Exception ex)
            {
                MessageBox.Show($"An error occurred: {ex.Message}", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }

            return false; // Invalid credentials or role is not Admin
        }

        // Optional: Event handler for textBox1 TextChanged (if you need it)
        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            // Your code for textBox1 changes
        }

        // Optional: Event handler for textBox2 TextChanged (if you need it)
        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            // Your code for textBox2 changes
        }
        // Optional: Event handler for checkBox1 CheckedChanged (if you need it)
        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            // You can add any logic if needed for checkbox state changes
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Form12 form12 = new Form12();
            form12.Show(); // Opens Form12 as a separate window
        }
    }
}
```



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WinFormsApp8
{
    public partial class Form10 : Form
    {
        public Form10()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // Create an instance of Form3
            Form3 form3 = new Form3();

            // Show Form3
            form3.Show();

            // Optional: Hide or Close Form1 if needed
        }

            // this.Hide(); // To hide Form1

            // this.Close(); // To close Form1

        }


        private void button2_Click(object sender, EventArgs e)

        {


            // Create an instance of Form3

            Form4 form4 = new Form4();


            // Show Form3

            form4.Show();


            // Optional: Hide or Close Form1 if needed

            // this.Hide(); // To hide Form1

            // this.Close(); // To close Form1




        }


        private void button3_Click(object sender, EventArgs e)

        {

            Form9 form9 = new Form9();

            form9.Show(); // Opens Form9

        }


    }
```

```
}
```



Form10
<-BACK
user management
PRODUCT MANAGEMENT