**User Manual**

### **1. How to Use the Application**

Upon launching the application, the user is presented with the main form, which displays a table of all available products, and a 'View Details' button.

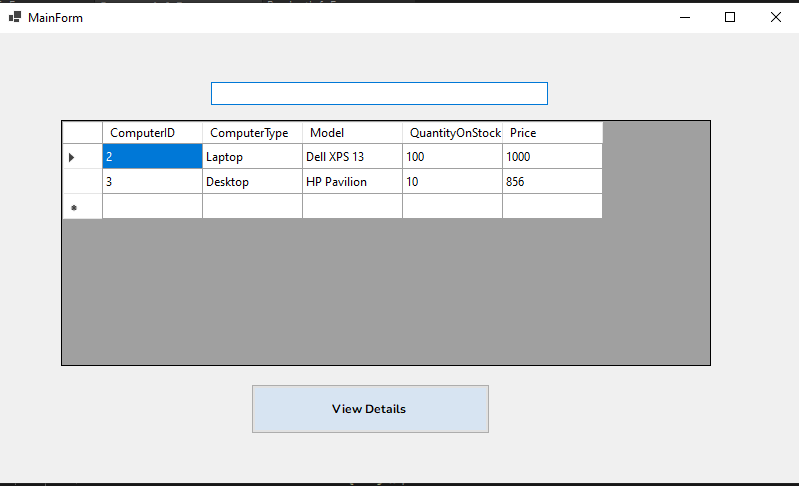
To purchase a product, the user can click on a product in the table to select it, and then click the 'View Details' button. This will open the product details form, displaying more information about the selected product.

If the user wishes to purchase the product, they can click the 'Purchase' button on the product details form. This will open the customer info form, where the user can enter their details and click the 'Process Order' button to complete the purchase.

### **2. Screen Captures and Source Code**

Please replace the placeholders with the actual screenshots and code snippets.

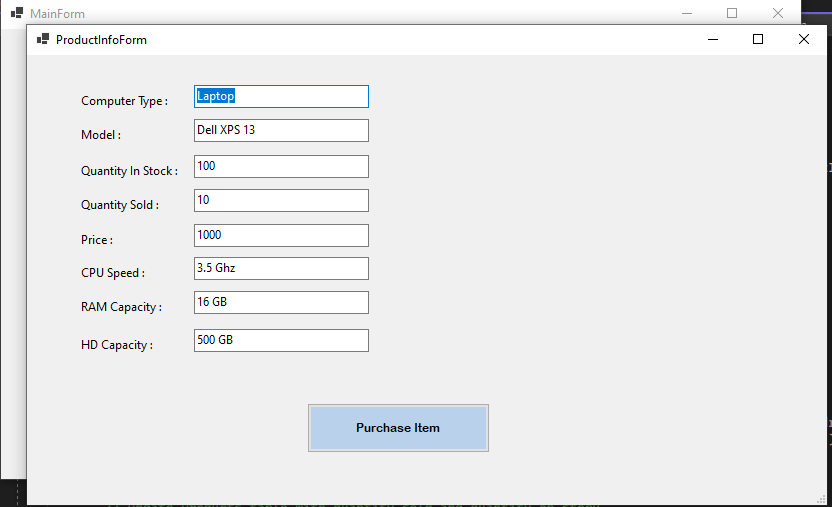
#### **MainForm**



This form displays all available products in a table.

| using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Data.OleDb; using System.Drawing; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms;  namespace ComputerStore {  public partial class MainForm : Form  {  private string databaseFileName = "ComputerStore.accdb";  private string connectionString;   private OleDbConnection connection;  private OleDbDataAdapter dataAdapter;  private DataTable dataTable;   public MainForm()  {  InitializeComponent();   this.Load += MainForm\_Load;   string appFolderPath = Path.GetDirectoryName(Application.ExecutablePath);  string databaseFilePath = Path.Combine(appFolderPath, databaseFileName);   connectionString = $"Provider=Microsoft.ACE.OLEDB.12.0;Data Source={databaseFilePath};";   connection = new OleDbConnection(connectionString);  dataTable = new DataTable();  }   private void MainForm\_Load(object sender, EventArgs e)  {  LoadProducts();  }   private void LoadProducts()  {  try  {  connection.Open();  string query = "SELECT ComputerID, ComputerType, Model, QuantityOnStock, Price FROM Products";  dataAdapter = new OleDbDataAdapter(query, connection);  dataTable.Clear();  dataAdapter.Fill(dataTable);  dataGridViewProducts.DataSource = dataTable;  }  catch (Exception ex)  {  MessageBox.Show("Error loading products: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  connection.Close();  }  }   private void btnViewDetails\_Click(object sender, EventArgs e)  {  if (dataGridViewProducts.SelectedRows.Count > 0)  {  int selectedProductID = Convert.ToInt32(dataGridViewProducts.SelectedRows[0].Cells["ComputerID"].Value);  ProductInfoForm productInfoForm = new ProductInfoForm(selectedProductID);  productInfoForm.ShowDialog();  }  }  } } |
| --- |

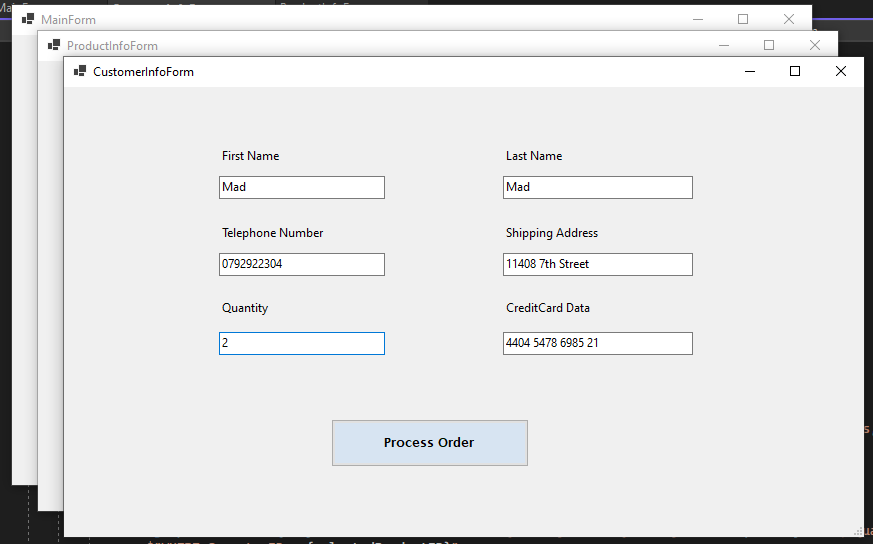
#### **ProductInfoForm**



This form displays detailed information about the selected product.

| using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Data.OleDb; using System.Drawing; using System.Linq; using System.Text; using System.Threading.Tasks; using System.IO; using System.Windows.Forms;  namespace ComputerStore {  public partial class ProductInfoForm : Form  {  private int selectedProductID;  private string connectionString;  private OleDbConnection connection;  private DataTable dataTable;   public ProductInfoForm(int productID)  {  InitializeComponent();  this.Load += ProductInfoForm\_Load;   selectedProductID = productID;   string appFolderPath = Path.GetDirectoryName(Application.ExecutablePath);  string databaseFilePath = Path.Combine(appFolderPath, "ComputerStore.accdb");  connectionString = $"Provider=Microsoft.ACE.OLEDB.12.0;Data Source={databaseFilePath};";   connection = new OleDbConnection(connectionString);  dataTable = new DataTable();   this.Load += ProductInfoForm\_Load;  }   private void ProductInfoForm\_Load(object sender, EventArgs e)  {  LoadProductInfo();  }   private void LoadProductInfo()  {  try  {  connection.Open();  string query = "SELECT \* FROM Products WHERE ComputerID = " + selectedProductID;  OleDbCommand command = new OleDbCommand(query, connection);  OleDbDataReader reader = command.ExecuteReader();   if (reader.Read())  {  ComputerType.Text = reader["ComputerType"].ToString();  Model.Text = reader["Model"].ToString();  QuantityOnStock.Text = reader["QuantityOnStock"].ToString();  Price.Text = reader["Price"].ToString();  QuantitySold.Text = reader["QuantitySold"].ToString();  CPUSpeed.Text = reader["CPUSpeed"].ToString();  RAMCapacity.Text = reader["RAMCapacity"].ToString();  HDCapacity.Text = reader["HDCapacity"].ToString();   *// Load the picture if available*  */\* if (!(reader["Image"] is DBNull))  {  byte[] imageData = (byte[])reader["Image"];  MemoryStream ms = new MemoryStream(imageData);  pictureBoxProduct.Image = Image.FromStream(ms);  }  else  {  pictureBoxProduct.Image = null;  } \*/*  }   reader.Close();  }  catch (Exception ex)  {  MessageBox.Show("Error loading product info: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  connection.Close();  }  }   private void btnPurchase\_Click(object sender, EventArgs e)  {  CustomerInfoForm customerInfoForm = new CustomerInfoForm(selectedProductID);  customerInfoForm.ShowDialog();  }  } } |
| --- |

#### **CustomerInfoForm**



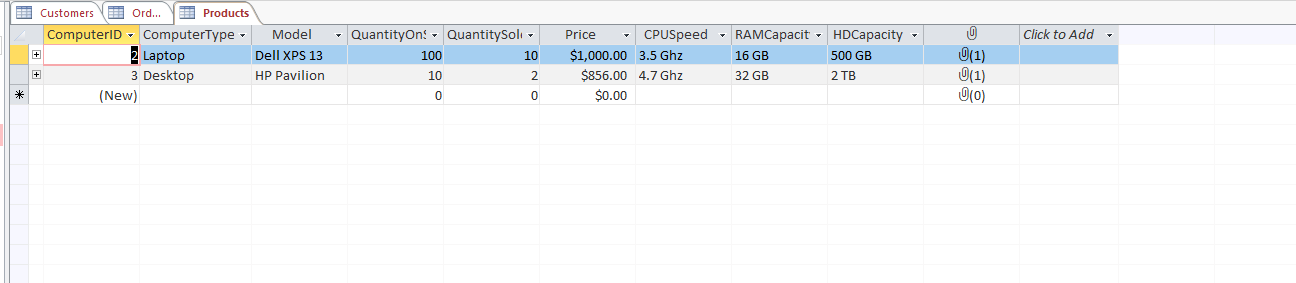
This form allows the user to enter their details and process their order.

| using System; using System.Collections.Generic; using System.ComponentModel; using System.Data; using System.Data.OleDb; using System.Drawing; using System.Linq; using System.Text; using System.Threading.Tasks; using System.Windows.Forms;  namespace ComputerStore {  public partial class CustomerInfoForm : Form  {  private int selectedProductID;  private string connectionString;  private OleDbConnection connection;   public CustomerInfoForm(int productID)  {  InitializeComponent();  selectedProductID = productID;   string appFolderPath = Path.GetDirectoryName(Application.ExecutablePath);  string databaseFilePath = Path.Combine(appFolderPath, "ComputerStore.accdb");  connectionString = $"Provider=Microsoft.ACE.OLEDB.12.0;Data Source={databaseFilePath};";   connection = new OleDbConnection(connectionString);  }   private void btnProcessOrder\_Click(object sender, EventArgs e)  {  try  {  connection.Open();   *// Get price and quantity sold from Products table*  string query = "SELECT Price, QuantitySold FROM Products WHERE ComputerID = " + selectedProductID;  OleDbCommand command = new OleDbCommand(query, connection);  OleDbDataReader reader = command.ExecuteReader();  reader.Read();  decimal price = Convert.ToDecimal(reader["Price"]);  int quantitySold = Convert.ToInt32(reader["QuantitySold"]);  reader.Close();   string firstName = FirstName.Text;  string lastName = LastName.Text;  string telephoneNumber = TelephoneNumber.Text;  string address = Address.Text;  string creditCardData = CreditCardData.Text;   *// Insert customer information into Customers table*  string insertCustomerQuery = $"INSERT INTO Customers (FirstName, LastName, TelephoneNumber, Address, CreditCardData) " +  $"VALUES ('{firstName}', '{lastName}', '{telephoneNumber}', '{address}', '{creditCardData}')";  OleDbCommand insertCustomerCommand = new OleDbCommand(insertCustomerQuery, connection);  insertCustomerCommand.ExecuteNonQuery();   *// Update Products table with quantity sold and quantity on stock*  string updateProductsQuery = $"UPDATE Products SET QuantitySold = QuantitySold + {quantitySold}, QuantityOnStock = QuantityOnStock - {quantitySold} " +  $"WHERE ComputerID = {selectedProductID}";  OleDbCommand updateProductsCommand = new OleDbCommand(updateProductsQuery, connection);  updateProductsCommand.ExecuteNonQuery();   *// Insert order information into Orders table*  decimal totalCost = quantitySold \* price;  string currentTime = DateTime.Now.ToString("MM/dd/yyyy hh:mm:ss");  OleDbCommand getMaxCustomerIdCommand = new OleDbCommand("SELECT MAX(CustomerID) FROM Customers", connection);  int customerId = (int)getMaxCustomerIdCommand.ExecuteScalar();   string insertOrderQuery = $"INSERT INTO Orders (CustomerID, ComputerID, OrderDate, TotalCost) " +  $"VALUES ({customerId}, {selectedProductID}, #{currentTime}#, {totalCost})";  OleDbCommand insertOrderCommand = new OleDbCommand(insertOrderQuery, connection);  insertOrderCommand.ExecuteNonQuery();   MessageBox.Show("Order processed successfully!", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);  this.Close();  }  catch (Exception ex)  {  MessageBox.Show("Error processing order: " + ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  finally  {  connection.Close();  }  }  } } |
| --- |

### **3. Database Tables**

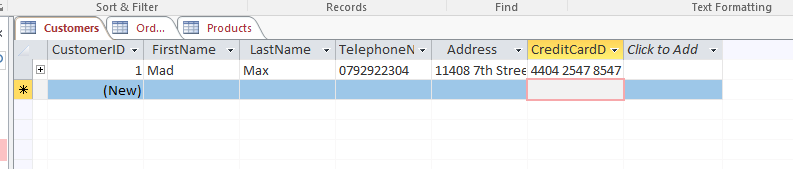
#### **Products Table**

This table stores information about all available products. Each product has an ID, type, model, quantity on stock, price, and other details.



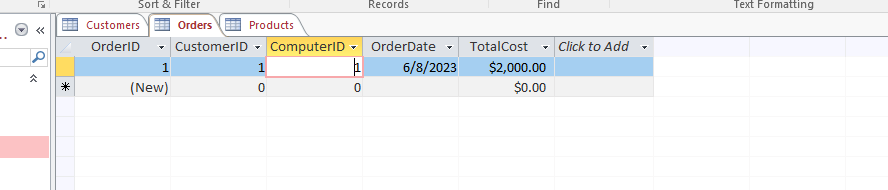
#### **Customers Table**

This table stores information about customers who have made a purchase. Each customer has an ID, first name, last name, telephone number, address, and credit card data.



#### **Orders Table**

This table stores information about all completed orders. Each order has an ID, customer ID, computer ID, order date, and total cost.



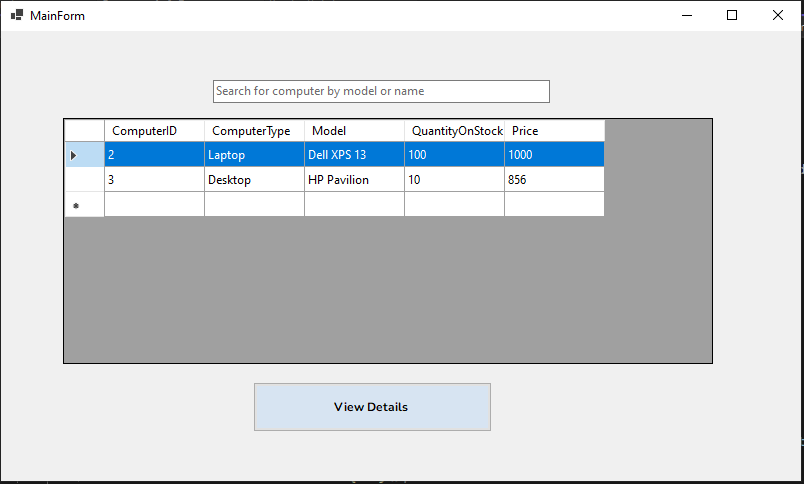
### **4. Main code and process screenshots**

| namespace ComputerStore {  internal static class Program  {  */// <summary>*  */// The main entry point for the application.*  */// </summary>*  [STAThread]  static void Main()  {  *// To customize application configuration such as set high DPI settings or default font,*  *// see https://aka.ms/applicationconfiguration.*  ApplicationConfiguration.Initialize();  Application.Run(new MainForm());  }  } } |
| --- |

### 

### 

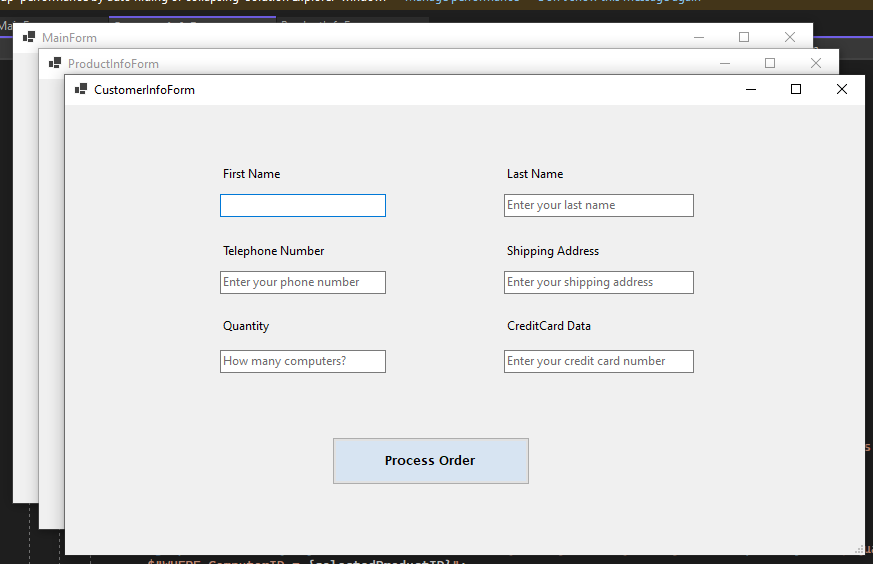
*Initial Load*



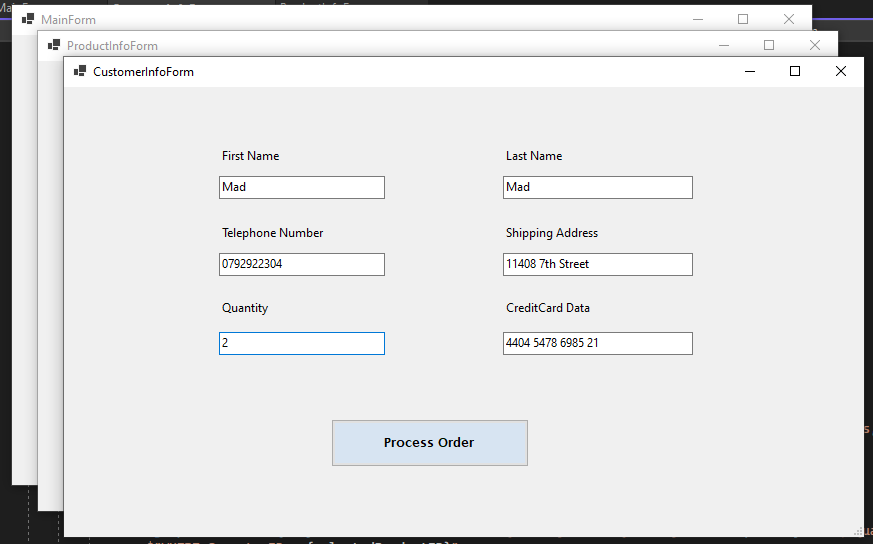
*Product Selected*

### 

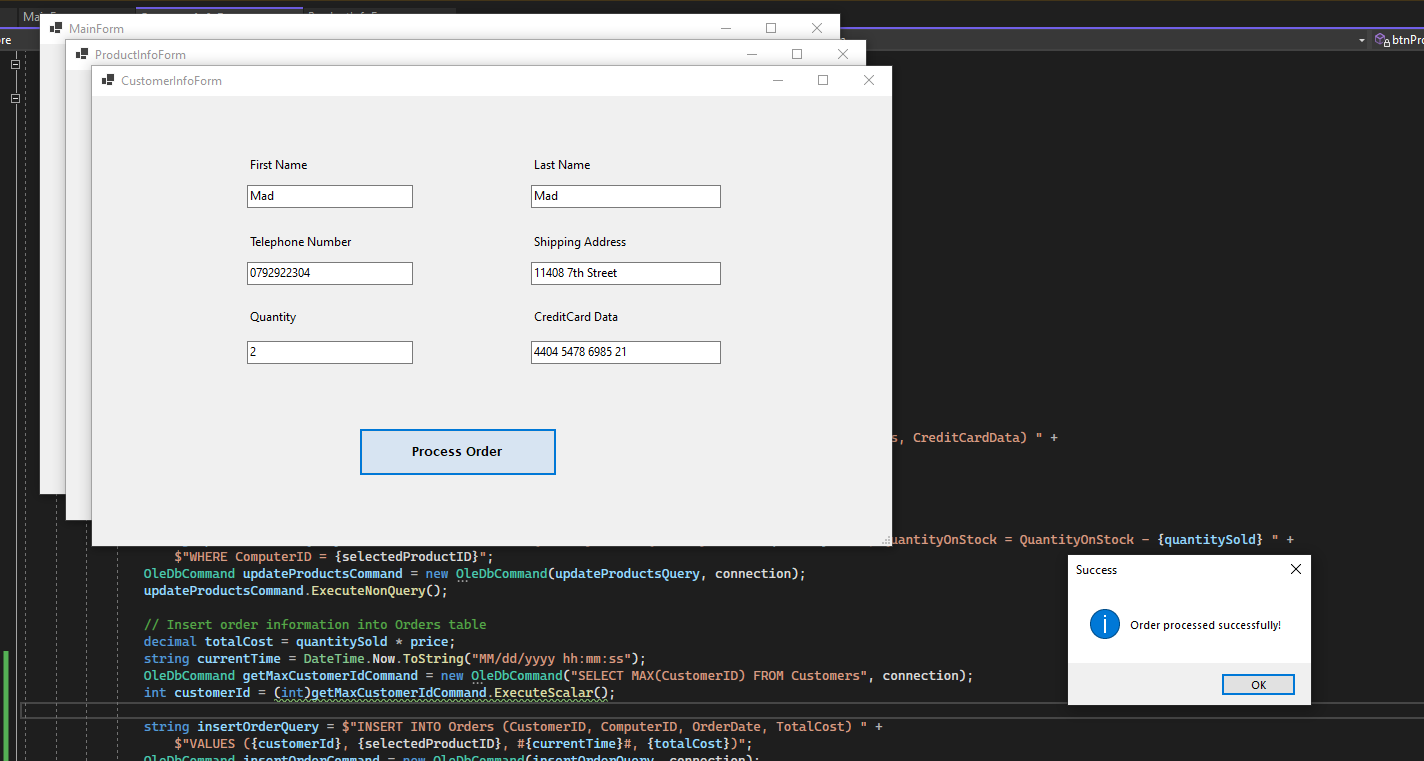
*Product Info*

**

*Contact Info Form*

**

*Form filled*

**

*Order processed successfully!*

### **Application Report**

This application allows users to view and purchase computer products. It was developed in C#, using Windows Forms for the user interface and an Access database for storing data.

All product data is loaded from the database upon launch and displayed in a table on the main form. The user can then select a product and click the 'View Details' button to view more detailed information about the product, including an image if one is available.

If the user decides to purchase the product, they can click the 'Purchase' button on the product details form. This will open the customer info form, where the user can enter their details and process their order. The order data, including the customer's details and the total cost of the order, is then stored in the database.

The application includes error handling to ensure that any issues are reported to the user in a clear and understandable manner. This includes any problems with loading the product data, processing an order, or accessing the database.