



Page |  
1

**Title of the Project Report: Inventory Management System**

**Name: Nikhil Ranjan Singh**

**Program Code: BCA**

**Enrolment no: 186737998**

**Regional Centre Code: 32 Ranchi**

**Study Centre: (0534) RKMS School, Jamshedpur -831015**

**Course Code: BCSP-064**

**Mobile Number: 9031350321**

**E-mail ID: nikhilranjansingh16@gmail.com**



Page |  
2

# **PROJECT REPORT**

**ON**

## **INVENTORY MANAGEMENT SYSTEM**

**SUBMITTED BY**

**NIKHIL RANJAN SINGH**

**ENROLLMENT NO.: 186737998**

**UNDER THE GUIDANCE OF**

**SIDHARTH RAHA**

**(IT ANALYST, TCS)**

**INDRA GANDHI NATIONAL OPEN UNIVERSITY**

**REGIONAL CENTRE: RANCHI**

**STUDY CENTRE: RAM KRISHNA MISSION SCHOOL, SIDHGORA,  
JAMSHEDPUR**

**YEAR: 2020-2021**



RC Ranchi <[synopsis32ranchi@gmail.com](mailto:synopsis32ranchi@gmail.com)>

Nov 12, 2021,  
10:32 AM

Page | to Ragini, me  
3

Dear Learner,

Your synopsis has been **approved**. Pl proceed with your final project submission.

----- Forwarded message -----

From: **Raja Ram Dutta** <[rajaramdutta@gmail.com](mailto:rajaramdutta@gmail.com)>

Date: Fri, Nov 12, 2021 at 5:46 AM

Subject: Re: SYNOPSIS FOR APPROVAL (DEC 2021)

To: RC Ranchi <[synopsis32ranchi@gmail.com](mailto:synopsis32ranchi@gmail.com)>

Cc: Dr. Ragini Kumari <[raginikumari@ignou.ac.in](mailto:raginikumari@ignou.ac.in)>

Enrolment No 186737998

Name Nikhil Ranjan Singh

Project Inventory Management System

Status : Approved

--

Raja Ram Dutta

Assistant Professor

Department of Computer Science & Engineering,

B.I.T Mesra , Ranchi

Off Campus Deoghar



SCHOOL OF COMPUTER AND INFORMATION SCIENCES  
IGNOU, MAIDAN GARHI, NEW DELHI – 110 068

II. PROFORMA OF BCA PROJECT PROPOSAL (BCSP-064)  
(Project's Title and Guide's Details)

Enrolment No.: 186737998

Regional Centre Code: 032

Study Centre: 0534P

1. Name of Student: NIKHIL RANJAN SINGH

2. Address of the student: EWS- 18/3, CHHOTA GOVINDPUR, JAMSHEDPUR-831015

.....  
3. (a) E-mail: nikhilranjansingh16@gmail.com      3. (b) Telephone/ Mobile No.: 9031350321

4. Title of the Project: INVENTORY MANAGEMENT SYSTEM

5. Name of Project Guide: SIDHARTH RAHA 5. (b) Designation of Project Guide: IT ANALYST

6. Address of Project Guide: H. NO. 17-01-06 , KARPORI PARK, CHHOTA GOVINDPUR, JAMSHEDPUR- 831015

7. Qualification of the Guide\*:

(Compulsory to Attach bio-data of Guide)

Ph.D.  MTech  BTech  MCA  Any other

\*Note : i. All the above-mentioned Degrees must have been awarded in Computer Science/IT only.

ii. A Guide should not guide more than 8 students of BCA at any point of time.

8. Industrial / Teaching experience of the Guide (in Years) 06 YEARS

9. Software Used for this Project: Visual Studio Dot Net Framework, MS SQL SERVER

Note: 1. Use of Visual Basic and MS-Access as Front End and Back End respectively is forbidden. But you are permitted to use Visual Basic with other Software. Also, you can use MS-Access with other software.

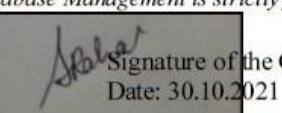
2. Use of C or C++ Programming Language for Project Related to Database Management is strictly forbidden.

Signature of the Student

Date: 30.10.2021

Signature of the Guide

Date: 30.10.2021



Raja Ram Dutta

Assistant Professor  
Dept. of Computer Science & Engg  
Birla Institute of Technology, Meerut  
(Deoghar Campus)

Important:

1. Attach this Proforma along with Guide's Biodata and Project Synopsis in the Project Report.
3. Not more than one student is permitted to work on a project.
4. Complete project as per the comments of Synopsis evaluator, then only submit your

For Office Use Only



Approved



Not approved

.....  
Signature, Designation, Stamp of the  
Project Proposal Evaluator

Date: .....

Suggestions for reformulating the Project:

- 1.
2. Inventory Management System is very common project. Implement with some new and unique functionalities.
- 3.
- 4.

# **Sidharth Raha**

Email [ID: sidharth0991@gmail.com](mailto:sidharth0991@gmail.com)

Mobile no: -09470373722

Page |

5

## Career Objective

To work in a dynamic professional environment with a growing organization and utilize my creativity and innovative thinking for benefit of the organization and myself.

## Summary

- Adapt at all phases of SDLC, which includes Design, Analysis, Coding and Implementation
- Exceptional analytical, troubleshooting organizational, communication, prioritization, and problem solving and leadership skills with ability to create and sustain high work tempo.
- Hands on experience in project scope, technical / functional specifications.
- Experience in .Net development using Microsoft best practices.
- Ability to write business logic meeting the business requirements and making it reusable.

**Total Experience:** 06+ yrs.

## Technology

Below is a list of important frameworks/architecture, tools/technologies and methodologies that I have worked with.

Framework/Architecture/Methodologies	Programming/Scripting Languages	Web Servers/ Markup Languages	Tools and Technologies/Databases/Source Controls
.Net Framework 4.0/3.5, Waterfall	C#, VB, JavaScript, Ajax, jQuery, Bootstrap 3.0/4.0, PowerShell, SSH Client (MobaXterm)	IIS 6.0/7.0, HTML5, CSS3	MS Visual Studio 2017/2015/2013/2010, ASP.Net, SQL Server, Crystal Report, Elite Enterprise 3.7, Informatica, RabbitMQ Message Broker, Restful Web API

## Key Competencies & Skills

My Competency Profile includes the following:

Skill	Proficiency (E0 to E4)
C# 4.0	E3
SQL Server	E3
ASP.NET	E2
jQuery, JavaScript	E2
Elite Enterprise/Met storm	E2
RabbitMQ	E2
Informatica	E2

**EDUCATIONAL DETAILS**

<b>COURSE</b>	<b>COLLEGE/UNIVERSITY</b>	<b>PERCENTAGE /CGPA</b>
B. Tech (Electronics and Communication)	RVS College of Engr and Tech	8.26
Higher Secondary	CISCE	74.00
Secondary	CISCE	77.00

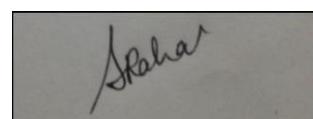
Page | 6

**Work Experience**

<b>Company</b>	<b>Designation</b>	<b>Duration</b>
Tata Consultancy Services LTD.	Assistant System Engineer Trainee	2015-2016
Tata Consultancy Services LTD.	Assistant System Engineer	2016-2017
Tata Consultancy Services LTD.	System Engineer	2017-2019
Tata Consultancy Services LTD.	IT Analyst	2019-Present

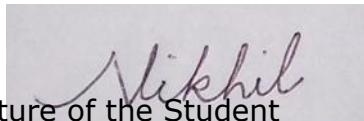
Date: 25<sup>th</sup> April, 2021

Place: Jamshedpur

A rectangular box containing a handwritten signature in black ink, which appears to read "Arshad".

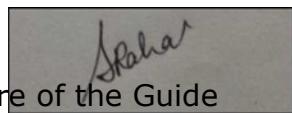
## CERTIFICATE OF ORIGINALITY

Page | 7  
This is to certify that the project report entitled Inventory Management System Submitted to Indira Gandhi National Open University in partial fulfilment of the requirement for the award of the degree of BACHELOR OF COMPUTER APPLICATIONS (BCA), is an original work carried out by Mr. Nikhil Ranjan Singh Enrolment No.: 186737998 under the guidance of Mr. Sidharth Raha. The matter embodied in this project is a genuine work done by the student and has not been submitted whether to this University or to any other University / Institute for the fulfilment of the requirement of any course of study.



Signature of the Student

Nikhil Ranjan Singh  
EWS – 18/3  
Chhota Govindpur  
Jamshedpur – 831015



Signature of the Guide

Sidharth Raha  
H. No. 17-01-06, Karpori Park  
Chhota Govindpur  
Jamshedpur - 831015

# INDEX

Page |  
8

<b>SL. NO.</b>	<b>Topic</b>	<b>Page No.</b>
1.	Introduction	9
2.	Objectives	10
3.	Tools/Environment Used	11
4.	SRS	12
5.	5.1 DFD	17
	5.2 E-R Diagram	18
6.	Database Design	19
7.	UI Design and Code Behind	25
8.	Testing	143
9.	Input and Output Screens	151
10.	Implementation of Security	173
11.	Limitation of Project	173
12.	Future Application	173
13.	Bibliography	174

# 1.0 INTRODUCTION

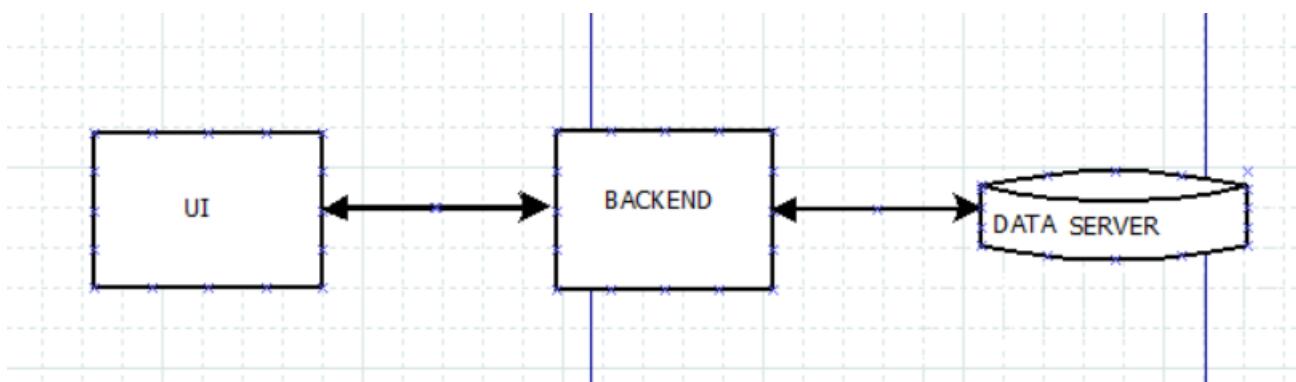
This project entitled on “INVENTORY MANAGEMENT SYSTEM” deals with digitalizing and automatizing the inventory management and control of retail or wholesaler shops which was earlier done manually using paper and pen or spreadsheet.

Inventory changes constantly. Throughout each day, sales, returns, new receipts- even damage and theft- affect inventory level. Inventory management and control are most important jobs in a successful retail or wholesale business.

Inventory management is a technique of controlling, storing and keeping track of inventory items. Inventory management is an essential component of supply chain management, as it regulates all the operations that are involved, from an item enter your store until it has been dispatched.

Inventory management system helps one tracks the amount of product one has on warehouse shelf, in store or sitting with distributors. This enables one to have right number of units in right place at the right time and for the right price. It helps to effectively track and control physical inventory, using which one can tell the exact quantity of each item with location, quantities running low and whether one should replenish them. As a busy business owner, one should be able to do all of this at a glance and this Inventory Management System Software helps achieve this.

This software developed to work as an inventory management System basically follows **Local Client-Server Architecture**. It is a WPF application whose UI is generated using XAML (Extensible Application Markup Language) and backend or code behind is C# and SQL is used to communicate with database.



## 2.0 OBJECTIVES

This project is for small retail or wholesaler shops or small company stores. This helps in managing and controlling their inventory. The main objectives of this project are:

- **REAL TIME INVENTORY UPDATING:** Changes to stock levels and movement of inventory get updated in real time. As soon as a sale is made, delivery arrives, or return is made, system tracks the changes.
- **Track Low Stock:** Owner can view items which are running low in stock to reorder and prevent missing sales.
- **Advanced Reporting:** Help owners learn more about their wider business. Intelligent reporting of sales data, reorder levels and more provides invaluable sight. Such reports help companies to optimize order times and track their most popular products.
- **Searching Item details:** Helps to search items providing its detail such as description, stock amount, selling price and storage location.

### **3.0 TOOLS/ENVIRONMENT USED**

- VISUAL STUDIO 2019
- MICROSOFT SQL SERVER MANAGEMENT STUDIO 18

## 4.0 SRS

### 4.1 PROJECT PROBLEM

- Page |  
12
- Business owner receives items from vendors with bill. He enters the bill in ledger. His staff checks the items and confirms its quality and quantity and tells the owner, if it is correct then stock of each item is updated else returned.
  - When customer comes asks for certain item. Owner/Staff checks its availability and location of storage. Confirms about sale from customer and generate sale receipt. Receipt no. is noted in sale ledger and stock of item is updated.
  - If the customer comes for returning items, then receipt is noted in return ledger. Quality of item is checked, if OK then stock is updated else kept in rejected bin.
  - Business Owner wants real time entry of his inventory data with low stock alert and statistics of his sale.

### 4.2 PRODUCT PERSPECTIVE

This software is standalone software.

### 4.3 PRODUCT FUNCTION

- ALLOWS TO SEARCH FOR ITEM AVAILABILITY AND LOCATION
- BILL ENTRY
- SALE ACCEPTANCE ENTRY
- RETURN SALE ENTRY
- VIEW LOW STOCK ITEM
- VIEW INVENTORY

### 4.4 USER CHARACTERISTIC

This software is designed to be used by one user at a time. Although more than one user id can be created but all valid user access same data and functions. To prevent unauthorized usage provision for creating user-id with specific password is provided to login. After login user access a dashboard from where they can use various functions of software.

### 4.5 SPECIFIC REQUIREMENT

## **4.5.1 USER INTERFACE**

- **LOGIN PAGE** for authenticating user and preventing others to use.
- **WORK PAGE** with menus DASH BOARD

VENDDOR for adding, updating and viewing new items.

PRODUCT for updating detail or adding new vendors.

BILL ENTRY takes to Bill form for entry of new bill.

INVENTORY REPORT

TRASACTION REPORT

## **4.5.2 FUNCTIONAL REQUIREMENT**

<b>Functional Requirement Id</b>	Fr1
<b>Requirement Title</b>	Login Operation
<b>Requirement Description</b>	<ul style="list-style-type: none"> <li>• There is only one user. He is the owner of the Shop.</li> <li>• Registered user enters user id and password</li> <li>• User Click Login button.</li> <li>• System should authenticate and create a valid user session upon successful authentication and redirect to dashboard window.</li> </ul>
<b>Business Rationale</b>	Protecting from unauthorized use
<b>Exception Scenario</b>	If authentication fails, user should be redirected to error page showing “User Id or Password incorrect” message.
<b>Dependencies</b>	Authentication service of corporate LDAP.

<b>Functional Requirement Id</b>	Fr2
<b>Requirement Title</b>	Bill Entry Operation
<b>Requirement Description</b>	<ul style="list-style-type: none"> <li>• Operation under Workspace for Owner</li> <li>• Bill Entry operation through form</li> </ul>

	<ul style="list-style-type: none"> <li>• User selects item name, vendor, update cost price and quantity.</li> <li>• Selects option for action 1. Return or 2. To Store</li> <li>• Press Save Bill Entry</li> <li>• System also updates inventory data upon saving bill.</li> </ul>
<b>Business Rationale</b>	Helps user keep track of incoming items from supplier and update inventory.
<b>Exception Scenario</b>	If item or vendor not found, then add new item or vendor.
<b>Dependencies</b>	FR1 User under Owner type should be logged in.

<b>Functional Requirement Id</b>	Fr3
<b>Requirement Title</b>	Search Operation
<b>Requirement Description</b>	<ul style="list-style-type: none"> <li>• Operation under Workspace for Staff</li> <li>• User searches an item.</li> <li>• As an output item code, item name, item description, item location and available stock appears.</li> </ul>
<b>Business Rationale</b>	Helps user find item in store and its availability.
<b>Exception Scenario</b>	Display “No product found” if no product matches.
<b>Dependencies</b>	FR1 User under Staff type should be logged in.

<b>Functional Requirement Id</b>	Fr4
<b>Requirement Title</b>	Sale Acceptance Entry Operation
<b>Requirement Description</b>	<ul style="list-style-type: none"> <li>• Operation under Workspace for Staff</li> <li>• Sale Acceptance Entry Through form.</li> <li>• User Enters Customer Name, Address</li> </ul>

	<p>and Contact No.</p> <ul style="list-style-type: none"> <li>• User selects Item Code, Item Name, Selling Price and quantity</li> <li>• Click Save Sale Acceptance.</li> <li>• System should update inventory upon saving sale acceptance.</li> </ul>
<b>Business Rationale</b>	Helps user track sale data and update inventory.
<b>Exception Scenario</b>	No exception Scenario.
<b>Dependencies</b>	FR1 User under Staff type should be logged in.

<b>Functional Requirement Id</b>	Fr5
<b>Requirement Title</b>	Return Sale Entry Operation
<b>Requirement Description</b>	<ul style="list-style-type: none"> <li>• Operation under Workspace for Staff</li> <li>• Return Sale Entry Through form.</li> <li>• Select Receipt No.</li> <li>• Select Item Name and Quantity</li> <li>• Select 1. Trash Or 2. Store</li> <li>• Click Save Return Sale.</li> <li>• System should update inventory upon saving return sale.</li> </ul>
<b>Business Rationale</b>	Helps user track return sale data and update inventory.
<b>Exception Scenario</b>	Receipt No., not Found.
<b>Dependencies</b>	FR1 User under Staff type should be logged in.

#### 4.5.3 PERFORMANCE REQUIREMENT

- All pages should load within 3 seconds
- Search result should be displayed within 3 second

#### **4.5.4 HARDWARE REQUIREMENT**

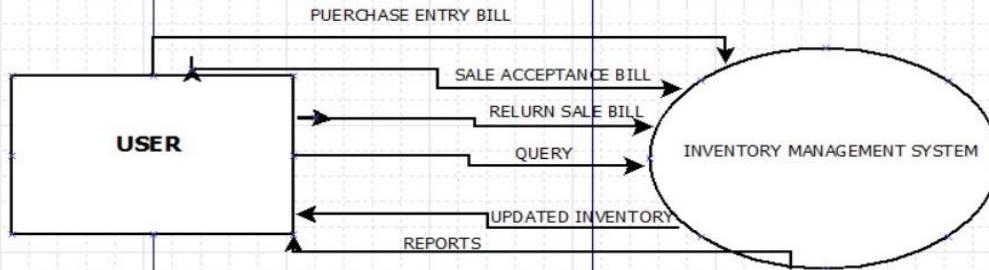
- PROCESSOR: INTEL PENTIUM I5
- HARD DISK: 1TB
- RAM: 4GB
- GRAPHICS CARD

#### **4.5.5 SOFTWARE REQUIREMENT**

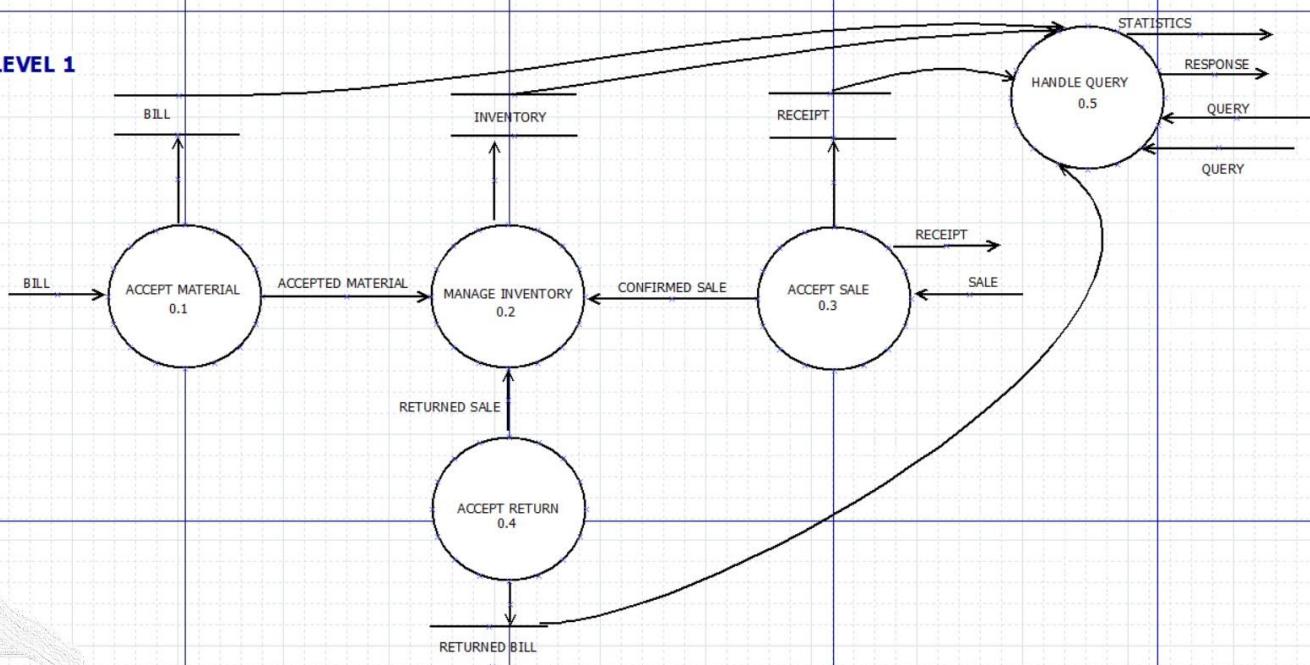
- OPERATING SYSTEM: WINDOWS 8 OR ABOVE
- VISUAL STUDIO DOT NET FRAMEWORK 2019
- C#
- MS SQL SERVER
- MICROSOFT SQL MANAGEMENT

## 5.1 DFD DIAGRAM

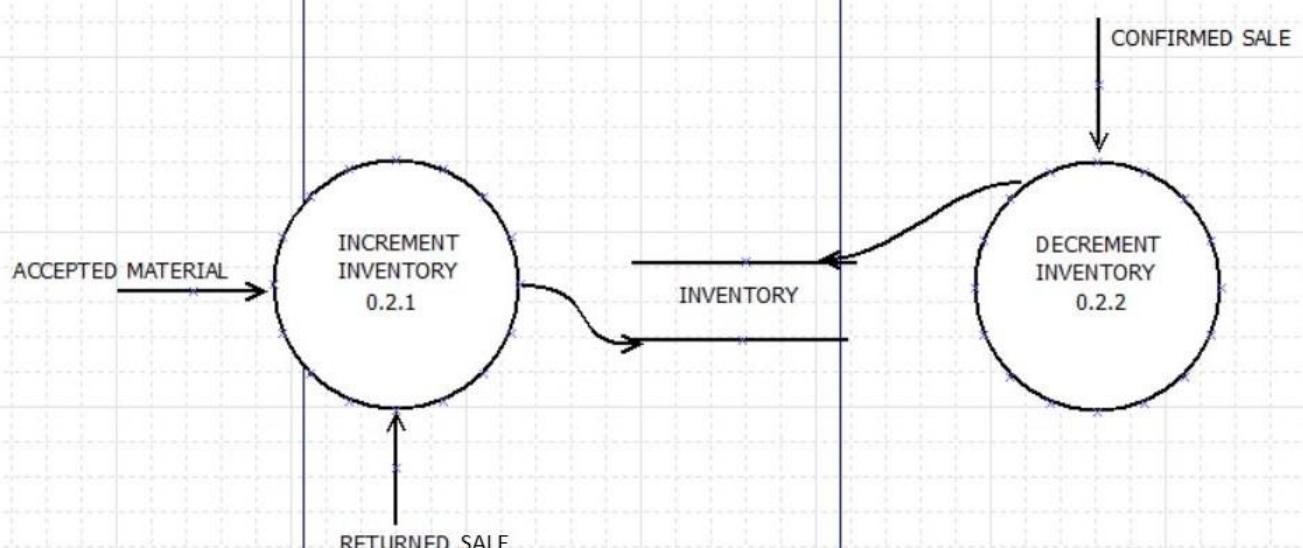
DFD LEVEL 0



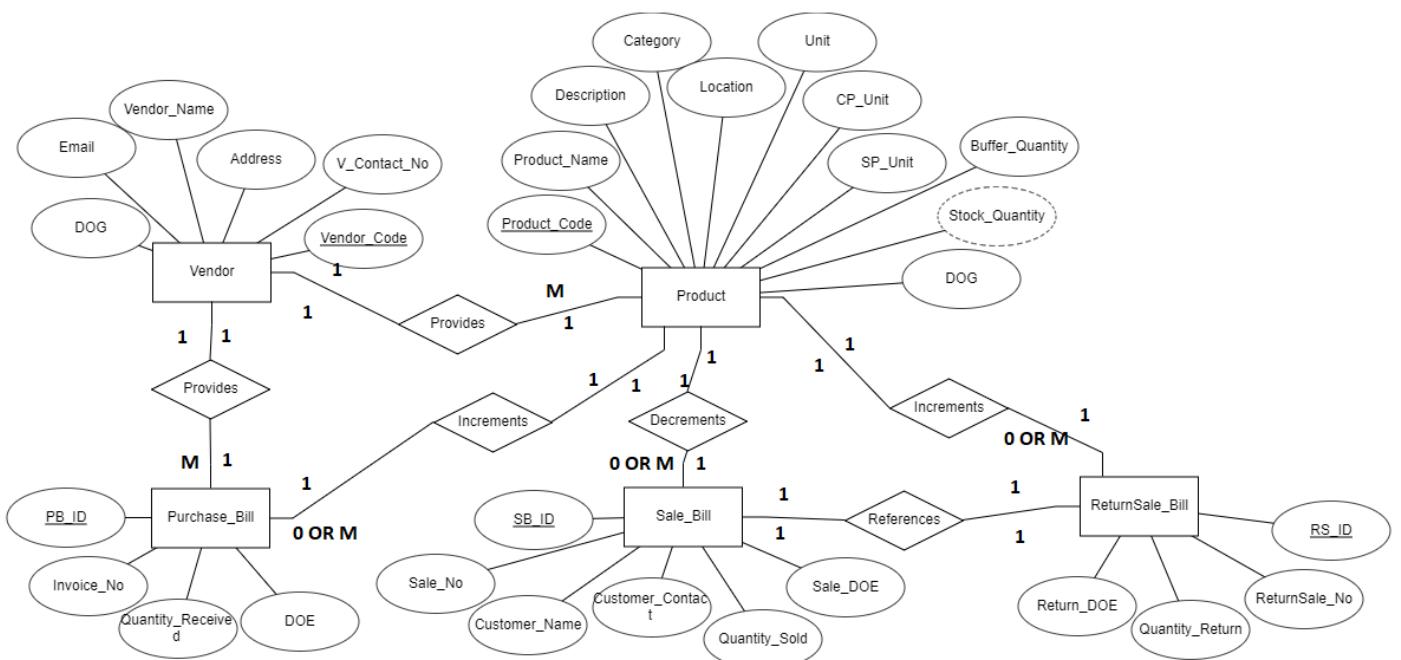
DFD LEVEL 1



DFD LEVEL 2



## 5.2 ER DIAGRAM



## 6.0 DATABASE DESIGN

### tblProduct

FIELD	DATA_TYPE	DESCRIPTION	CONSTRAINT
VENDOR_CODE	VARCHAR (10)		PRIMARY_KEY
VENDOR_NAME	VARCHAR (50)		NOT_NULL
ADDRESS	VARCHAR (100)		NOT_NULL
V_CONTACT_NO	VARCHAR (15)		NOT_NULL
EMAIL_ID	VARCHAR (50)		NOT_NULL
DOG	DATE	DATE OF GENERATION	NOT_NULL

### tblProduct

FIELD	DATA_TYPE	DESCRIPTION	CONSTRAINT
Product_Code	VARCHAR (10)		PRIMARY_KEY
Product_Name	VARCHAR (50)		NOT_NULL
Description	VARCHAR (100)		NOT_NULL
Category	VARCHAR (50)		NOT_NULL
Vendor	VARCHAR (10)	LINK TO tblVendor	FOREIGN_KEY, NOT NULL
Location	VARCHAR (50)		NOT_NULL
Unit	VARCHAR (25)		NOT_NULL
CP_Unit	DOUBLE		NOT_NULL
SP_Unit	DOUBLE		NOT_NULL
Buffer_Quantity	INT	MINIMUM STOCK LEVEL WHICH MUST BE MAINTAINED	NOT_NULL
Stock_Quantity	INT		NOT_NULL
Date	DATE	DATE OF GENERATION	NOT_NULL

## tblPurchaseBill

FIELD	DATA_TYPE	DESCRIPTION	CONSTRAINT
PB_ID	INT		PRIMARY_KEY, AUTO
Invoice_Id	VARCHAR (10)		
Vendor_Code	VARCHAR (10)	LINKS TO tblVendor	FOREIGN_KEY, NOT_NULL
Product_Code	VARCHAR (10)	LINKS TO tblProduct	FOREIGN_KEY, NOT_NULL
QUANTITY_RECIEVED	INT	QUANTITY RECEIVED FROM VENDOR	NOT_NULL
Purchase_DOE	DATE	DATE OF ENTRY	NOT_NULL

## tblSaleBill

FIELD	DATA_TYPE	DESCRIPTION	CONSTRAINT
SB_ID	INT		PRIMARY_KEY, AUTO
Sale_Id	VARCHAR (10)		NOT_NULL
Customer_Name	VARCHAR (50)		NOT_NULL
Customer_Contact	VARCHAR (14)		NOT_NULL
Product_Code	VARCHAR (10)	LINKS TO tblProduct	FOREIGN_KEY, NOT_NULL
Quantity_Sold	INT	QUANTITY SOLD	NOT_NULL
Sale_DOE	DATE	DATE OF ENTRY	NOT_NULL

## tbl\_RSBill

FIELD	DATA_TYPE	DESCRIPTION	CONSTRAINT

RS_ID	INT		PRIMARY_KEY
ReturnSale_Id	VARCHAR (10)		Not_null
Sale_Id	VARCHAR (10)	LINKS TO tblSaleBill	FOREIGN_KEY, NOT_NULL
Product_Code	VARCHAR (10)	LINKS TO tblProduct	FOREIGN_KEY, NOT_NULL
Quantity_Return	INT	QUANTITY RETURNED	NOT_NULL
Return_DOE	DATE	DATE OF ENTRY	NOT_NULL

## tblUser

FIELD	DATA_TYPE	DESCRIPTION	CONSTRAINT
Id	INT		PRIMARY_KEY, AUTO
Username	VARCHAR (50)		NOT_NULL
password	VARCHAR (50)		NOT_NULL

## tblCategory

FIELD	DATA_TYPE	DESCRIPTION	CONSTRAINT
Category_Id	INT		PRIMARY_KEY, AUTO
Category_Name	VARCHAR (50)		NOT_NULL

Page |  
22

```
CREATE TABLE [dbo].[tblUser](  
    [Id] [int] IDENTITY (1,1) NOT NULL,  
    [Username] [VARCHAR] (50) NOT NULL,  
    [password] [VARCHAR] (50) NOT NULL,  
    CONSTRAINT (PK_tbl_User) PRIMARY KEY[Id]  
);
```

```
CREATE TABLE [dbo].[tblCategory](  
    [Id] [int] IDENTITY (1,1) NOT NULL,  
    [Category] [VARCHAR] (50) NOT NULL,  
    CONSTRAINT (PK_tbl_User) PRIMARY KEY[Id]  
);
```

```
CREATE TABLE [dbo].[tblVendor](  
    [VENDOR_CODE] [VARCHAR] (10) NOT NULL,  
    [VENDOR_NAME] [VARCHAR] (50) NOT NULL,  
    [ADDRESS] [VARCHAR] (100) NOT NULL,  
    [V_CONTACT_NO] [VARCHAR] (14) NOT NULL,  
    [EMAIL_ID] [VARCHAR] (50) NOT NULL,  
    [DOG] [DATE] NOT NULL,  
    CONSTRAINT (PK_tbl_Vendor) PRIMARY KEY[VENDOR_CODE]  
);
```

```
CREATE TABLE [dbo].[tblProduct](  
    [Product_Code] [VARCHAR] (10) NOT NULL,  
    [Product_Name] [VARCHAR] (50) NOT NULL,  
    [Description] [VARCHAR] (100) NOT NULL,  
    [Category] [VARCHAR] (50) NOT NULL,
```

[Vendor] [VARCHAR] (10) NOT NULL,  
[Location] [VARCHAR] (50) NOT NULL,  
[Unit] [VARCHAR] (25) NOT NULL,  
[CP\_Unit] [MONEY] NOT NULL,  
[SP\_Unit] [MONEY] (50) NOT NULL,  
[Buffer\_Quantity] [INT] NOT NULL,  
[Stock\_quantity] [INT] NOT NULL,  
[Date] [DATE] NOT NULL,  
CONSTRAINT (PK\_tbl\_Product) PRIMARY KEY[Product\_Code]  
CONSTRAINT (FK\_tbl\_Product) FOREIGN KEY[Vendor]  
REFRENCES tblVendor [VENDOR\_CODE]  
);

CREATE TABLE [dbo].[tblPurchaseBill](  
[PB\_ID] [INT] IDENTITY(100,2) NOT NULL,  
[Invoice\_Id] [VARCHAR] (10) NOT NULL,  
[Vendor\_Code] [VARCHAR] (10) NOT NULL,  
[Product\_Code] [VARCHAR] (10) NOT NULL,  
[Quantity\_Received] [INT] NOT NULL,  
[Purchase\_Date] [DATE] NOT NULL,  
CONSTRAINT (PK\_tbl\_PurchaseBill) PRIMARY KEY[PB\_ID]  
CONSTRAINT (FK\_tbl\_PurchaseBill) FOREIGN KEY[Vendor\_Code]  
REFRENCES tblVendor [dbo].[tblVendor].[VENDOR\_CODE]  
CONSTRAINT (FK\_tbl\_PurchaseBill) FOREIGN KEY [dbo].[tblPurchaseBill].[Product\_Code]  
REFRENCES tblProduct [dbo].[tblProduct].[Product\_Code]  
);

CREATE TABLE [dbo].[tblSaleBill](  
[SB\_ID] [INT] IDENTITY(200,2) NOT NULL,  
[Sale\_Id] [VARCHAR] (10) NOT NULL,

[Customer\_Name] [VARCHAR] (25) NOT NULL,  
[Customer\_Contact] [VARCHAR] (14) NOT NULL,  
[Product\_Code] [VARCHAR] (10) NOT NULL,  
[Quantity\_Sold] [INT] NOT NULL,  
[Purchase\_Date] [DATE] NOT NULL,  
CONSTRAINT (PK\_tbl\_SaleBill) PRIMARY KEY[SB\_ID]  
CONSTRAINT (FK\_tbl\_SaleBill) FOREIGN KEY [dbo].[tblSaleBill].[Product\_Code]  
REFRENCES tblProduct [dbo].[tblProduct].[Product\_Code]  
);

CREATE TABLE [dbo].[tblRSBill](  
[RS\_ID] [INT] IDENTITY(100,2) NOT NULL,  
[ReturnSale\_Id] [VARCHAR] (10) NOT NULL,  
[Sale\_Id] [VARCHAR] (10) NOT NULL,  
[Product\_Code] [VARCHAR] (10) NOT NULL,  
[Quantity\_Return] [INT] NOT NULL,  
[Return\_Date] [DATE] NOT NULL,  
CONSTRAINT (PK\_tbl\_RSBill) PRIMARY KEY[RS\_ID]  
CONSTRAINT (FK\_tbl\_RSBill) FOREIGN KEY [dbo].[tblRSBill].[Sale\_Id]  
REFRENCES tblProduct [dbo].[tblSaleBill].[Sale\_Id]  
CONSTRAINT (FK\_tbl\_RSBill) FOREIGN KEY [dbo].[tblSaleBill].[Product\_Code]  
REFRENCES tblProduct [dbo].[tblProduct].[Product\_Code]  
);

## 7.0 UI DESIGN AND ITS CODE BEHIND

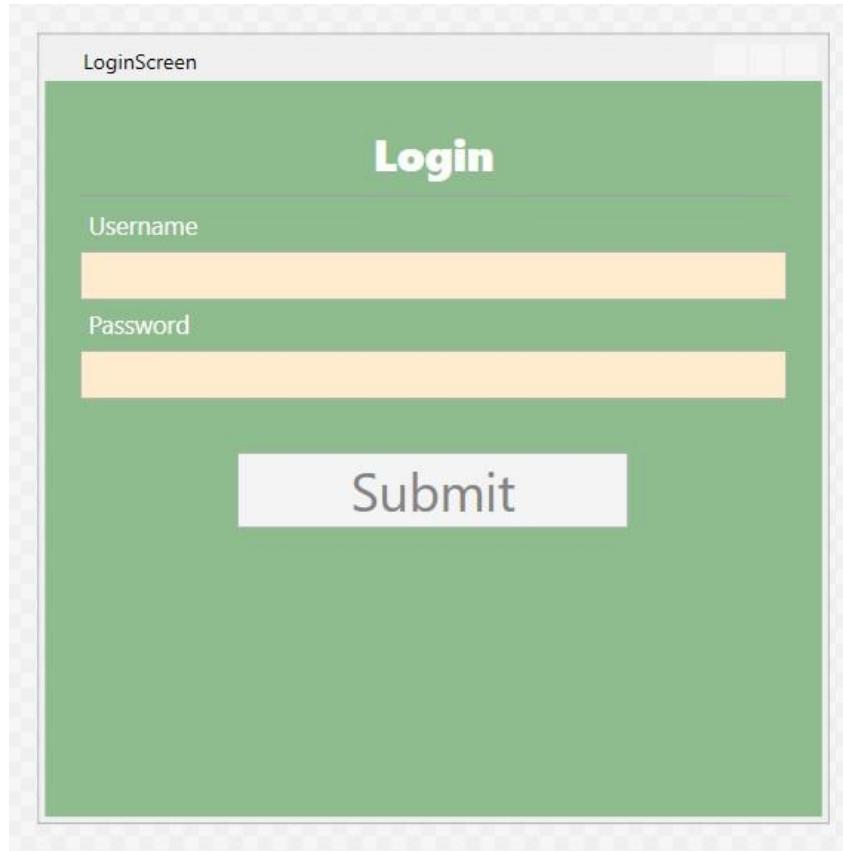
### LoginScreen.xaml

Page | 25

```
<Window x:Class="testWPF.LoginScreen"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    Title="LoginScreen" Height="450" Width="450" FontSize="14"
    Background="DarkSeaGreen">

    <StackPanel Margin="20">
        <Label Content="Login" Foreground="White" FontWeight="Heavy" FontSize="25"
    HorizontalAlignment="Center"></Label>
        <Separator></Separator>
        <Label Content="Username" Foreground="White"></Label>
        <TextBox Name="txtUsername" KeyUp="txtUsername_KeyDown"
    KeyDown="txtUsername_KeyDown" Background="BlanchedAlmond"
    FontSize="20"></TextBox>
        <Label Content="Password" Foreground="White"></Label>
        <PasswordBox Name="txtPassword" KeyUp="txtPassword_KeyDown"
    KeyDown="txtPassword_KeyDown" Background="BlanchedAlmond"
    FontSize="20"></PasswordBox>
        <Button Name="btnSubmit" IsEnabled="False" Click="btnSubmit_Click"
    Background="Aquamarine" Content="Submit" Margin="90 30" FontSize="30"></Button>
    </StackPanel>

</Window>
```



## LoginScreen.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
//using testWPF.Ibal;
using testWPF.Model;

namespace testWPF
```

```
{  
    /// <summary>  
    /// Interaction logic for LoginScreen.xaml  
    /// </summary>  
    public partial class LoginScreen : Window //, ILoginCred  
    {  
        // private ILoginCred LoginCred { get; set; }  
        //public LoginScreen(ILoginCred LoginCred)  
        //{
            // this.LoginCred = LoginCred;  
        //}  
        public LoginScreen()  
        {  
            InitializeComponent();  
  
        }  
  
        private void btnSubmit_Click(object sender, RoutedEventArgs e)  
        {  
            //if (txtUsername.Text != "" && txtPassword.Password != "")  
            //{
                // loginCredential obj = new loginCredential();  
                // obj.Username = txtUsername.Text;  
                // obj.Password = txtPassword.Password.ToString();  
  
                // bool IsAuthenticated=LoginCred.Authenticate(obj);  
            //}  
            SqlConnection sqlCon = new SqlConnection(@"Data  
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect  
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi  
SubnetFailover=False");  
            try  
            {  
                if (sqlCon.State == System.Data.ConnectionState.Closed)  
                    sqlCon.Open();  
  
                string query = "SELECT COUNT(1) FROM [dbo].[tblUser] where  
Username=@Username and password=@Password";  
                SqlCommand sqlcmd = new SqlCommand(query, sqlCon);  
                sqlcmd.CommandType = System.Data.CommandType.Text;  
                sqlcmd.Parameters.AddWithValue("@Username", txtUsername.Text);  
                sqlcmd.Parameters.AddWithValue("@Password", txtPassword.Password);  
                int count = Convert.ToInt32(sqlcmd.ExecuteScalar());  
            }  
        }  
    }  
}
```

```
if (count == 1)
{
    MainWindow dashboard = new MainWindow();
    dashboard.Show();
    this.Close();
}
else
{
    MessageBox.Show("Username Or Password is incorrect;");
    txtUsername.Clear();
    txtPassword.Clear();
    btnSubmit.IsEnabled = false;
}

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
    btnSubmit.IsEnabled = false;
}
finally
{
    sqlCon.Close();
}
}
```

```
private void txtUsername_KeyDown(object sender, KeyEventArgs e)
{
    if(txtUsername.Text!="") && txtPassword.Password != "")
    {
        btnSubmit.IsEnabled = true;
    }
    else
        btnSubmit.IsEnabled = false;
}
```

```
private void txtPassword_KeyDown(object sender, KeyEventArgs e)
{
    if (txtUsername.Text != "" && txtPassword.Password != "")
    {
        btnSubmit.IsEnabled = true;
    }
}
```

```
        else
            btnSubmit.IsEnabled = false;
    }
}
}
```

Page |  
29

## MainWindow.xaml

```
<Window x:Class="testWPF.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    Title="MainWindow" Height="680" Width="800" Background="DarkSeaGreen">

    <Grid>
        <StackPanel Orientation="Vertical" MinWidth="200" HorizontalAlignment="Left">
            <Label Content="Menu Section" HorizontalContentAlignment="Center"
Background="BurlyWood" FontSize="25" MinWidth="200" Width="242"
VerticalAlignment="Top"/>
            <Button x:Name="btnVendor" Content="Vendor" Click="btnVendor_Click"
Padding="12" Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>
            <Button x:Name="btnProduct" Content="Product" Click="btnProduct_Click"
Padding="12" Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>
            <Button x:Name="btnPBE" Content="Purchase Bill Entry" Click="btnPBE_Click"
Padding="12" Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>
            <Button x:Name="btnSAE" Content="Sale Acceptance Entry" Click="btnSAE_Click"
Padding="12" Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>
            <Button x:Name="btnRSE" Content="Return Sale Entry" Click="btnRSE_Click"
Padding="12" Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>
            <Button x:Name="btnInventory" Content="Inventory Report"
Click="btnInventory_Click" Padding="12" Foreground="Beige" Background="RosyBrown"
FontSize="22"></Button>
            <Button x:Name="btnReport" Content="Report" Click="btnReport_Click"
Padding="12" Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>
            <Button x:Name="btnUser" Content="User" Click="btnUser_Click" Padding="12"
Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>
            <Button x:Name="btnCategory" Content="Category" Click="btnCategory_Click"
Padding="12" Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>
```

Page | 30

```
<Button x:Name="btnLogout" Content="Logout" Click="btnLogout_Click"
    Padding="12" Foreground="Beige" Background="RosyBrown" FontSize="22"></Button>

</StackPanel>
<Frame x:Name="Main" Margin="238,0,-0.4,0"
    NavigationUIVisibility="Hidden"></Frame>
</Grid>
</Window>
```



## MainWindow.xaml.cs

```
using System;
Page | using System.Collections.Generic;
31 using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
        }

        private void btnVendor_Click(object sender, RoutedEventArgs e)
        {
            //FOR OPENING NEW VENDOR PAGE IN MAINWINDOW IFRAME
            Main.Content = new Vendor();

            //TO CHANGE BUTTON COLOR ON CLICK
            btnVendor.Background = Brushes.Navy;
            btnProduct.Background = Brushes.RosyBrown;
            btnPBE.Background = Brushes.RosyBrown;
            btnSAE.Background = Brushes.RosyBrown;
            btnRSE.Background = Brushes.RosyBrown;
            btnInventory.Background = Brushes.RosyBrown;
            btnReport.Background = Brushes.RosyBrown;
        }
    }
}
```

```
btnUser.Background = Brushes.RosyBrown;
btnCategory.Background = Brushes.RosyBrown;
btnLogout.Background = Brushes.RosyBrown;

}

private void btnProduct_Click(object sender, RoutedEventArgs e)
{
    //FOR OPENING NEW PRODUCT PAGE IN MAINWINDOW IFRAME
    Main.Content = new Product();

    //TO CHANGE BUTTON COLOR ON CLICK
    btnVendor.Background = Brushes.RosyBrown;
    btnProduct.Background = Brushes.Navy;
    btnPBE.Background = Brushes.RosyBrown;
    btnSAE.Background = Brushes.RosyBrown;
    btnRSE.Background = Brushes.RosyBrown;
    btnInventory.Background = Brushes.RosyBrown;
    btnReport.Background = Brushes.RosyBrown;
    btnUser.Background = Brushes.RosyBrown;
    btnCategory.Background = Brushes.RosyBrown;
    btnLogout.Background = Brushes.RosyBrown;
}

private void btnPBE_Click(object sender, RoutedEventArgs e)
{
    //FOR OPENING NEW PBE PAGE IN MAINWINDOW IFRAME
    Main.Content = new PurchaseBillEntry();

    //TO CHANGE BUTTON COLOR ON CLICK
    btnVendor.Background = Brushes.RosyBrown;
    btnProduct.Background = Brushes.RosyBrown;
    btnPBE.Background = Brushes.Navy;
    btnSAE.Background = Brushes.RosyBrown;
    btnRSE.Background = Brushes.RosyBrown;
    btnInventory.Background = Brushes.RosyBrown;
    btnReport.Background = Brushes.RosyBrown;
    btnUser.Background = Brushes.RosyBrown;
    btnCategory.Background = Brushes.RosyBrown;
    btnLogout.Background = Brushes.RosyBrown;
}

private void btnSAE_Click(object sender, RoutedEventArgs e)
{
```

//FOR OPENING NEW SAE PAGE IN MAINWINDOW IFRAME

Main.Content = new SaleAcceptanceEntry();

//TO CHANGE BUTTON COLOR ON CLICK

btnVendor.Background = Brushes.RosyBrown;  
btnProduct.Background = Brushes.RosyBrown;  
btnPBE.Background = Brushes.RosyBrown;  
btnSAE.Background = Brushes.Navy;  
btnRSE.Background = Brushes.RosyBrown;  
btnInventory.Background = Brushes.RosyBrown;  
btnReport.Background = Brushes.RosyBrown;  
btnUser.Background = Brushes.RosyBrown;  
btnCategory.Background = Brushes.RosyBrown;  
btnLogout.Background = Brushes.RosyBrown;

}

private void btnRSE\_Click(object sender, RoutedEventArgs e)

{

//FOR OPENING NEW RSE PAGE IN MAINWINDOW IFRAME  
Main.Content = new ReturnSaleEntry();

//TO CHANGE BUTTON COLOR ON CLICK

btnVendor.Background = Brushes.RosyBrown;  
btnProduct.Background = Brushes.RosyBrown;  
btnPBE.Background = Brushes.RosyBrown;  
btnSAE.Background = Brushes.RosyBrown;  
btnRSE.Background = Brushes.Navy;  
btnInventory.Background = Brushes.RosyBrown;  
btnReport.Background = Brushes.RosyBrown;  
btnUser.Background = Brushes.RosyBrown;  
btnCategory.Background = Brushes.RosyBrown;  
btnLogout.Background = Brushes.RosyBrown;

}

private void btnInventory\_Click(object sender, RoutedEventArgs e)

{

//FOR OPENING NEW INVENTORY PAGE IN MAINWINDOW IFRAME  
Main.Content = new Inventory();

//TO CHANGE BUTTON COLOR ON CLICK

btnVendor.Background = Brushes.RosyBrown;  
btnProduct.Background = Brushes.RosyBrown;  
btnPBE.Background = Brushes.RosyBrown;  
btnSAE.Background = Brushes.RosyBrown;

```
btnRSE.Background = Brushes.RosyBrown;
btnInventory.Background = Brushes.Navy;
btnReport.Background = Brushes.RosyBrown;
btnUser.Background = Brushes.RosyBrown;
btnCategory.Background = Brushes.RosyBrown;
btnLogout.Background = Brushes.RosyBrown;
}

private void btnReport_Click(object sender, RoutedEventArgs e)
{
    //FOR OPENING NEW REPORT PAGE IN MAINWINDOW IFRAME
    Main.Content = new Report();

    //TO CHANGE BUTTON COLOR ON CLICK
    btnVendor.Background = Brushes.RosyBrown;
    btnProduct.Background = Brushes.RosyBrown;
    btnPBE.Background = Brushes.RosyBrown;
    btnSAE.Background = Brushes.RosyBrown;
    btnRSE.Background = Brushes.RosyBrown;
    btnInventory.Background = Brushes.RosyBrown;
    btnReport.Background = Brushes.Navy;
    btnUser.Background = Brushes.RosyBrown;
    btnCategory.Background = Brushes.RosyBrown;
    btnLogout.Background = Brushes.RosyBrown;
}

private void btnUser_Click(object sender, RoutedEventArgs e)
{
    //FOR OPENING NEW USER PAGE IN MAINWINDOW IFRAME
    Main.Content = new UserList();

    //TO CHANGE BUTTON COLOR ON CLICK
    btnVendor.Background = Brushes.RosyBrown;
    btnProduct.Background = Brushes.RosyBrown;
    btnPBE.Background = Brushes.RosyBrown;
    btnSAE.Background = Brushes.RosyBrown;
    btnRSE.Background = Brushes.RosyBrown;
    btnInventory.Background = Brushes.RosyBrown;
    btnReport.Background = Brushes.RosyBrown;
    btnUser.Background = Brushes.Navy;
    btnCategory.Background = Brushes.RosyBrown;
    btnLogout.Background = Brushes.RosyBrown;
}
```

Page |  
35

```
private void btnCategory_Click(object sender, RoutedEventArgs e)
{
    //FOR OPENING NEW CATEGORY PAGE IN MAINWINDOW IFRAME
    Main.Content = new Category();

    //TO CHANGE BUTTON COLOR ON CLICK
    btnVendor.Background = Brushes.RosyBrown;
    btnProduct.Background = Brushes.RosyBrown;
    btnPBE.Background = Brushes.RosyBrown;
    btnSAE.Background = Brushes.RosyBrown;
    btnRSE.Background = Brushes.RosyBrown;
    btnInventory.Background = Brushes.RosyBrown;
    btnReport.Background = Brushes.RosyBrown;
    btnUser.Background = Brushes.RosyBrown;
    btnCategory.Background = Brushes.Navy;
    btnLogout.Background = Brushes.RosyBrown;
}
```

```
private void btnLogout_Click(object sender, RoutedEventArgs e)
{
    //TO CHANGE BUTTON COLOR ON CLICK
    btnVendor.Background = Brushes.RosyBrown;
    btnProduct.Background = Brushes.RosyBrown;
    btnPBE.Background = Brushes.RosyBrown;
    btnSAE.Background = Brushes.RosyBrown;
    btnRSE.Background = Brushes.RosyBrown;
    btnInventory.Background = Brushes.RosyBrown;
    btnReport.Background = Brushes.RosyBrown;
    btnUser.Background = Brushes.RosyBrown;
    btnCategory.Background = Brushes.RosyBrown;
    btnLogout.Background = Brushes.Navy;

    // TO LOGOUT FROM CURRENT SESSION AND OPEN LOGIN PAGE
    var mv = new LoginScreen();
    mv.Show();
    this.Close();

}
```

## Vendor.xaml

```
<Page x:Class="testWPF.Vendor"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="Vendor" Background="DarkSeaGreen" Foreground="White" FontSize="16">

    <StackPanel Orientation="Vertical">
        <TextBlock Text="VENDOR" HorizontalAlignment="Center" FontSize="30"
        FontWeight="ExtraBold" Margin="0 25 0 0"/>
        <StackPanel Orientation="Horizontal" >
            <Button x:Name="btnVendorAdd" Click="btnVendorAdd_Click" Content="ADD"
            Width="100" Height="30" Margin="25 25 25 10"/>
            <TextBox x:Name="tb_VendorSearch" Width="150" Height="30" Margin="75 25 0
            10"/>
            <Button x:Name="btnVendorEdit" Click="btnVendorEdit_Click" Content="EDIT"
            Width="100" Height="30" Margin="10 25 25 10"/>
        </StackPanel>

        <Separator Background="Black"></Separator>
        <DataGrid x:Name="Vendor_DataGrid" Height="276" Margin="27,0,176,0" />

    </StackPanel>
</Page>
```



## Vendor.xaml.cs

```
Page | 37
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using testWPF.Model;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for Vendor.xaml
    /// </summary>
    public partial class Vendor : Page
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public Vendor()
        {
            InitializeComponent();
            LoadVendorGrid();
        }

        private void LoadVendorGrid()
        {
            try
            {
                if (sqlCon.State == System.Data.ConnectionState.Closed)
                    sqlCon.Open();
            }
        }
    }
}
```

```
string query = "SELECT * FROM tblVendor";
SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
sqlcmd.CommandType = System.Data.CommandType.Text;
DataTable dt = new DataTable();
SqlDataReader sdr = sqlcmd.ExecuteReader();

dt.Load(sdr);
Vendor_Grid.ItemsSource = dt.DefaultView;
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    sqlCon.Close();
}
}

private void btnVendorAdd_Click(object sender, RoutedEventArgs e)
{
    VendorForm ven = new VendorForm();

    ven.Show();
}

private void btnVendorEdit_Click(object sender, RoutedEventArgs e)
{
    try
    {
        VendorForm ven = new VendorForm();
        sqlCon.Open();
        SqlCommand cmd = new SqlCommand("Select VENDOR_CODE, VENDOR_NAME,
ADDRESS, V_CONTACT_NO, EMAIL_ID, DOG FROM tblVendor WHERE VENDOR_CODE ='" +
tb_VendorSearch.Text + "'", sqlCon);
        SqlDataReader da = cmd.ExecuteReader();
        while (da.Read())
        {
            ven.textbox_Vcode.Text = da.GetValue(0).ToString();
            ven.textbox_Vname.Text = da.GetValue(1).ToString();
            ven.textbox_Vaddr.Text = da.GetValue(2).ToString();
            ven.textbox_Vcontact.Text = da.GetValue(3).ToString();
            ven.textbox_Vemail.Text = da.GetValue(4).ToString();
            ven.textbox_Vdate.Text = da.GetValue(5).ToString();
        }
    }
}
```

```
        }

    sqlCon.Close();

    ven.Show();
}

catch (SqlException Ex)
{
    MessageBox.Show(Ex.Message);
}

finally
{
    sqlCon.Close();
}

}
```

## VendorForm.xaml

```
<Window x:Class="testWPF.VendorForm"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    Title="VendorForm" Height="450" Width="800" Background="DarkSeaGreen"
    Foreground="White" FontSize="16">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="20"/>
```

```

<ColumnDefinition Width="Auto"/>
<ColumnDefinition Width="20"/>
</Grid.ColumnDefinitions>
<Grid.RowDefinitions>
    <RowDefinition Height="20"/>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="20"/>
</Grid.RowDefinitions>
<TextBlock Grid.Row="1" Grid.Column="3" Grid.ColumnSpan="3" Text=" VENDOR
FORM " Margin="0 10 0 20" FontWeight="ExtraBlack" FontSize="20"/>
<TextBlock Grid.Row="2" Grid.Column="1" Text="VENDOR CODE" Margin="0 0 10 10"
FontWeight="SemiBold"/>
<TextBlock Grid.Row="3" Grid.Column="1" Text="VENDOR NAME" Margin="0 0 10 10"
FontWeight="SemiBold"/>
<TextBlock Grid.Row="4" Grid.Column="1" Text="ADDRESS" Margin="0 0 10 10"
FontWeight="SemiBold"/>
<TextBlock Grid.Row="5" Grid.Column="1" Text="CONTACT NO." Margin="0 0 10 10"
FontWeight="SemiBold"/>
<TextBlock Grid.Row="6" Grid.Column="1" Text="E-MAIL ID" Margin="0 0 10 10"
FontWeight="SemiBold"/>
<TextBlock Grid.Row="7" Grid.Column="1" Text="DATE" Margin="0 0 10 10"
FontWeight="SemiBold"/>

    <TextBox Grid.Row="2" Grid.Column="2" Width="150" Margin="5 0 5 10"
x:Name="textbox_Vcode"/>
    <TextBox Grid.Row="3" Grid.Column="2" Width="150" Margin="5 0 5 10"
x:Name="textbox_Vname"/>
    <TextBox Grid.Row="4" Grid.Column="2" Width="150" Margin="5 0 5 10"
Height="100" x:Name="textbox_Vaddr"/>

```

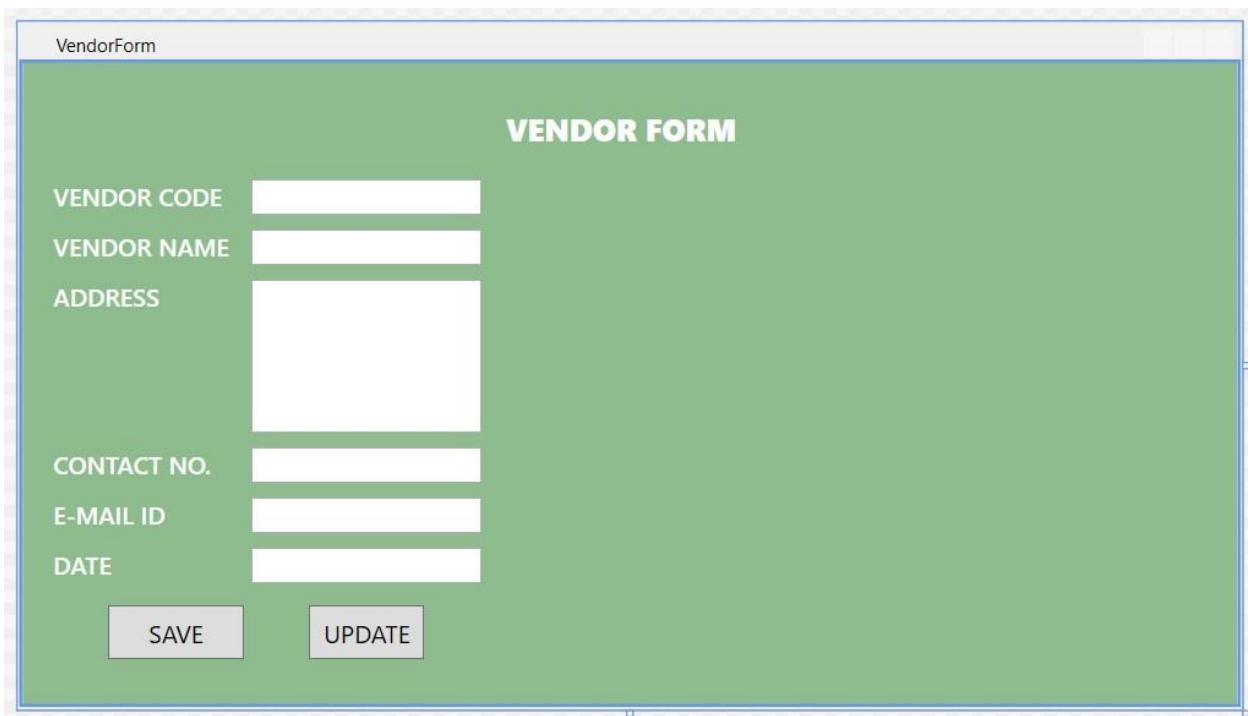
Page | 41

```

<TextBox Grid.Row="5" Grid.Column="2" Width="150" x:Name="textbox_Vcontact"
Margin="5 0 5 10"/>
<TextBox Grid.Row="6" Grid.Column="2" x:Name="textbox_Vemail" Margin="5 0 5
10"/>
<TextBox Grid.Row="7" Grid.Column="2" x:Name="textbox_Vdate" Margin="5 0 5
10"/>

<Button x:Name="btn_VFormSave" Click="btn_VFormSave_Click" Grid.Row="8"
Grid.Column="1" Height="35" Margin="36 5 0 0" Content="SAVE"/>
<Button x:Name="btn_VFormUpdate" Click="btn_VFormUpdate_Click" Grid.Row="8"
Grid.Column="2" Width="75" Height="35" Margin="0 5 0 0" Content="UPDATE"/>
</Grid>
</Window>

```



## VendorForm.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;

```

```
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

Page | namespace testWPF
42 {
    /// <summary>
    /// Interaction logic for VendorForm.xaml
    /// </summary>
    public partial class VendorForm : Window
    {
        public VendorForm()
        {
            InitializeComponent();
        }

        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");

        public void clearData()
        {
            textbox_Vname.Clear();
            textbox_Vaddr.Clear();
            textbox_Vcontact.Clear();
            textbox_Vemail.Clear();
            textbox_Vdate.Clear();
        }

        //CHECK FOR USERNAME AND PASSWORD TEXTBOX TO BE FILLED BEFORE ADDING
        NEW USER
        public bool isValid()
        {
            if (textbox_Vcode.Text == string.Empty)
            {
                MessageBox.Show("Vendor Code is Required.", "Failed");
                return false;
            }
            if (textbox_Vname.Text == string.Empty)
            {
                MessageBox.Show("Vendor Name is Required", "Failed");
            }
        }
    }
}
```

```

        return false;
    }
    if (textbox_Vaddr.Text == string.Empty)
    {
        MessageBox.Show("Vendor Address is Required", "Failed");
        return false;
    }
    if (textbox_Vcontact.Text == string.Empty)
    {
        MessageBox.Show("Vendor Contact Info is Required", "Failed");
        return false;
    }
    if (textbox_Vemail.Text == string.Empty)
    {
        MessageBox.Show("Vendor E-Mail Address is Required", "Failed");
        return false;
    }
    if (textbox_Vdate.Text == string.Empty)
    {
        MessageBox.Show("Date is Required", "Failed");
        return false;
    }
    return true;
}

private void btn_VFormSave_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (isValid())
        {
            SqlCommand cmd = new SqlCommand("INSERT INTO tblVendor VALUES
(@VENDOR_CODE, @VENDOR_NAME, @ADDRESS, @V_CONTACT_NO, @EMAIL_ID,
@DOG)", sqlCon);
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@VENDOR_CODE", textbox_Vcode.Text);
            cmd.Parameters.AddWithValue("@VENDOR_NAME", textbox_Vname.Text);
            cmd.Parameters.AddWithValue("@ADDRESS", textbox_Vaddr.Text);
            cmd.Parameters.AddWithValue("@V_CONTACT_NO", textbox_Vcontact.Text);
            cmd.Parameters.AddWithValue("@EMAIL_ID", textbox_Vemail.Text);

            cmd.Parameters.AddWithValue("@DOG", Convert.ToDateTime(textbox_Vdate.Text));
            sqlCon.Open();
            cmd.ExecuteNonQuery();
        }
    }
}

```

```
        sqlCon.Close();
        MessageBox.Show("New Vendor Successfully Added");
        clearData();
    }
}
catch (SqlException ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    sqlCon.Close();
}
}

private void btn_VFormUpdate_Click(object sender, RoutedEventArgs e)
{
    sqlCon.Open();
    SqlCommand cmd = new SqlCommand("UPDATE tblVendor set VENDOR_NAME='" +
    textbox_Vname.Text + "', ADDRESS ='" + textbox_Vaddr.Text + "', V_CONTACT_NO =
    "'"+textbox_Vcontact.Text+"', EMAIL_ID = '" + textbox_Vemail.Text + "', DOG =
    "'"+textbox_Vdate.Text+"' WHERE VENDOR_CODE='" + textbox_Vcode.Text + "'", sqlCon);
    try
    {
        cmd.ExecuteNonQuery();
        MessageBox.Show("Record has been updated successfully");
    }
    catch (SqlException ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
        clearData();
    }
}
}
```

## Product.xaml

```
Page | 45
<Page x:Class="testWPF.Product"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:testWPF"
      mc:Ignorable="d"
      d:DesignHeight="750" d:DesignWidth="800"
      Title="Product" Background="DarkSeaGreen" Foreground="White" FontSize="16">
    <StackPanel Orientation="Vertical">
        <TextBlock Text="PRODUCT" HorizontalAlignment="Center" FontSize="30"
        FontWeight="ExtraBold" Margin="0 25 0 0"/>
        <StackPanel Orientation="Horizontal" >
            <Button x:Name="btnProductAdd" Click="btnProductAdd_Click" Content="ADD"
            Width="100" Height="30" Margin="25 25 25 10"/>
            <TextBox x:Name="tb_ProductSearch" Width="150" Height="30" Margin="75 25 0
            10"/>
            <Button x:Name="btnProductEdit" Click="btnProductEdit_Click" Content="EDIT"
            Width="100" Height="30" Margin="10 25 25 10"/>
        </StackPanel>
        <Separator Background="Black"></Separator>
        <DataGrid x:Name="dg_Pview" Height="597" Margin="28,0" />
    </StackPanel>
</Page>
```



## Product.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Data;
46 using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for Product.xaml
    /// </summary>
    public partial class Product : Page
    {
        public Product()
        {
            InitializeComponent();
            LoadProductGrid();
        }

        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");

        private void LoadProductGrid()
        {
            try
            {
                if (sqlCon.State == System.Data.ConnectionState.Closed)
                    sqlCon.Open();
            }
        }
    }
}
```

```
        string query = "SELECT * FROM tblProduct";
        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
        sqlcmd.CommandType = System.Data.CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataReader sdr = sqlcmd.ExecuteReader();
        dt.Load(sdr);
        dg_Pview.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}

private void btnProductAdd_Click(object sender, RoutedEventArgs e)
{
    ProductForm pdt = new ProductForm();

    pdt.Show();
}

private void btnProductEdit_Click(object sender, RoutedEventArgs e)
{
    try
    {
        ProductForm pdt = new ProductForm();
        sqlCon.Open();
        SqlCommand cmd = new SqlCommand("Select Product_Code, Product_Name,
Description, Category, Vendor, Location, Unit, CP_Unit, SP_Unit, Buffer_Quantity,
Stock_Quantity, Date FROM tblProduct WHERE Product_Code ='" + tb_ProductSearch.Text
+ "'", sqlCon);
        SqlDataReader da = cmd.ExecuteReader();
        while (da.Read())
        {
            pdt.textbox_Pcode.Text = da.GetValue(0).ToString();
            pdt.textbox_Pname.Text = da.GetValue(1).ToString();
            pdt.textbox_Pdesc.Text = da.GetValue(2).ToString();
            pdt.cb_Pcategory.SelectedItem = da.GetValue(3).ToString();
            pdt.cb_Pvendor.SelectedItem= da.GetValue(4).ToString();
        }
    }
}
```

```
Page | 48
      pdt.textbox_Plocation.Text = da.GetValue(5).ToString();
      pdt.textbox_Punit.Text = da.GetValue(6).ToString();
      pdt.textbox_Pcp.Text = da.GetValue(7).ToString();
      pdt.textbox_Psp.Text = da.GetValue(8).ToString();
      pdt.textbox_Pbuffer.Text = da.GetValue(9).ToString();
      pdt.textbox_Pstock.Text = da.GetValue(10).ToString();
      pdt.textbox_Pdog.Text = da.GetValue(5).ToString();
    }

    sqlCon.Close();

    pdt.Show();
}
catch (SqlException Ex)
{
  MessageBox.Show(Ex.Message);
}

finally
{
  sqlCon.Close();
}

}
}
```



<TextBlock Grid.Row="1" Grid.Column="3" Grid.ColumnSpan="3" Text=" PRODUCT FORM " Margin="0 10 0 20" FontWeight="ExtraBlack" FontSize="20"/>  
 <TextBlock Grid.Row="2" Grid.Column="1" Text="PRODUCT CODE" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="3" Grid.Column="1" Text="PRODUCT NAME" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="4" Grid.Column="1" Text="DESCRIPTION" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="5" Grid.Column="1" Text="CATEGORY" Margin="0 0 10 10" FontWeight="SemiBold"/>  
  
 <TextBlock Grid.Row="6" Grid.Column="1" Text="VENDOR" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="7" Grid.Column="1" Text="LOCATION" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="8" Grid.Column="1" Text="UNIT" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="9" Grid.Column="1" Text="CP/UNIT" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="10" Grid.Column="1" Text="SP/UNIT" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="11" Grid.Column="1" Text="BUFFER QUANTITY" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="12" Grid.Column="1" Text="STOCK QUANTITY" Margin="0 0 10 10" FontWeight="SemiBold"/>  
 <TextBlock Grid.Row="13" Grid.Column="1" Text="DATE" Margin="0 0 10 10" FontWeight="SemiBold"/>  
  
 <TextBox Grid.Row="2" Grid.Column="2" Width="150" Margin="5 0 5 10" x:Name="textbox\_Pcode"/>  
 <TextBox Grid.Row="3" Grid.Column="2" Width="150" Margin="5 0 5 10" x:Name="textbox\_Pname"/>  
 <TextBox Grid.Row="4" Grid.Column="2" Width="150" Margin="5 0 5 10" Height="100" x:Name="textbox\_Pdesc"/>  
 <ComboBox Grid.Row="5" Grid.Column="2" Width="150" x:Name="cb\_Pcategory" Margin="5 0 5 10">  
  
 </ComboBox>  
 <ComboBox Grid.Row="6" Grid.Column="2" x:Name="cb\_Pvendor" Margin="5 0 5 10" SelectionChanged="cb\_Pvendor\_SelectionChanged">  
  
 </ComboBox>  
 <TextBox Grid.Row="7" Grid.Column="2" x:Name="textbox\_Plocation" Margin="5 0 5 10"/>

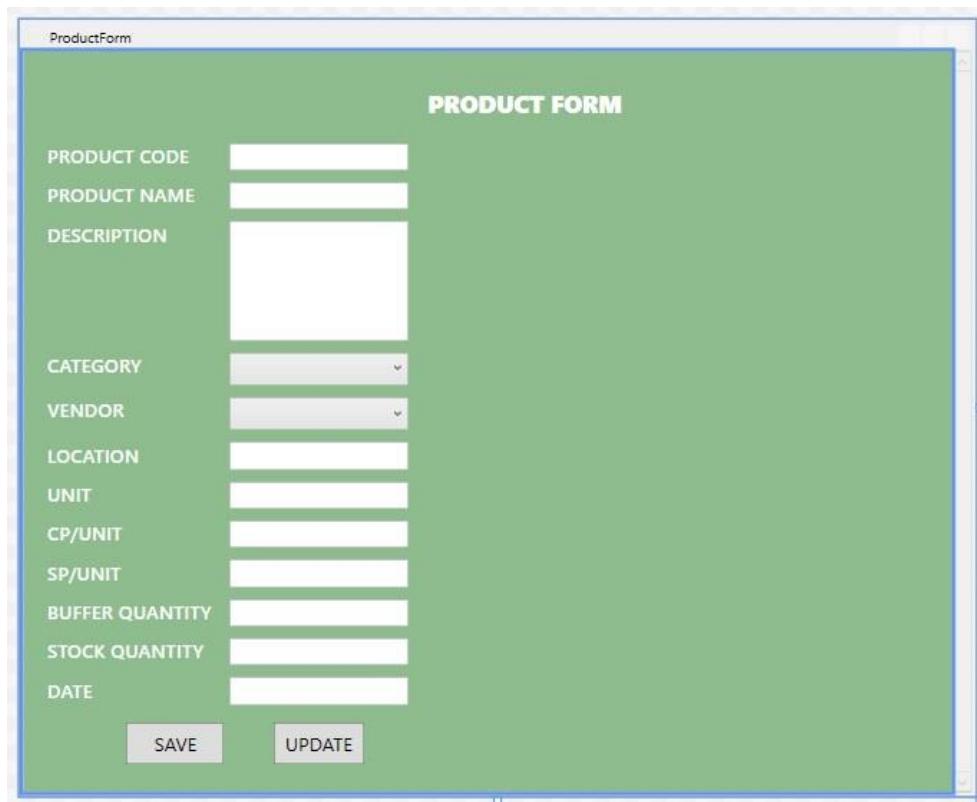
Page | 51

```

        <TextBox Grid.Row="8" Grid.Column="2" x:Name="textbox_Punit" Margin="5 0 5
10"/>
        <TextBox Grid.Row="9" Grid.Column="2" x:Name="textbox_Pcp" Margin="5 0 5
10"/>
        <TextBox Grid.Row="10" Grid.Column="2" x:Name="textbox_Psp" Margin="5 0 5
10"/>
        <TextBox Grid.Row="11" Grid.Column="2" x:Name="textbox_Pbuffer" Margin="5 0 5
10"/>
        <TextBox Grid.Row="12" Grid.Column="2" x:Name="textbox_Pstock" Margin="5 0 5
10"/>
        <TextBox Grid.Row="13" Grid.Column="2" x:Name="textbox_Pdog" Margin="5 0 5
10"/>
        <Button x:Name="btn_PFormSave" Click="btn_PFormSave_Click" Grid.Row="14"
Grid.Column="1" Height="35" Margin="66 5 0 0" Content="SAVE"/>
        <Button x:Name="btn_pFormUpdate" Click="btn_pFormUpdate_Click"
Grid.Row="14" Grid.Column="2" Width="75" Height="35" Margin="0 5 0 0"
Content="UPDATE"/>

    </Grid>
</ScrollViewer>
</Window>

```



## ProductForm.xaml.cs

```
using System;
Page | using System.Collections.Generic;
52 using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using System.Data.SqlClient;
using System.Data;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for ProductForm.xaml
    /// </summary>
    public partial class ProductForm : Window
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");

        public ProductForm()
        {
            InitializeComponent();
            loadComboBoxVendor();
        }

        //Function to Load ComboBox in ProductForm
        private void loadComboBoxVendor()
        {
            try
            {
                if (sqlCon.State == System.Data.ConnectionState.Closed)
```

```
sqlCon.Open();

string query = "SELECT vendor_code, vendor_name FROM tblVendor";
SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
sqlcmd.CommandType = System.Data.CommandType.Text;
DataTable dt = new DataTable();
SqlDataAdapter adapter = new SqlDataAdapter(sqlcmd);

adapter.Fill(dt);
cb_Pvendor.ItemsSource = dt.DefaultView;
cb_Pvendor.DisplayMemberPath = "vendor_name";
cb_Pvendor.SelectedValuePath = "vendor_code";
sqlcmd.Dispose();

string query1 = "SELECT Category_id, Category_Name FROM tblCategory";
SqlCommand sqlcmd1 = new SqlCommand(query1, sqlCon);
sqlcmd1.CommandType = System.Data.CommandType.Text;
DataTable dt1 = new DataTable();
SqlDataAdapter adapter1 = new SqlDataAdapter(sqlcmd1);

adapter1.Fill(dt1);
cb_Pcategory.ItemsSource = dt1.DefaultView;
cb_Pcategory.DisplayMemberPath = "Category_Name";
cb_Pcategory.SelectedValuePath = "Category_id";
sqlcmd1.Dispose();
sqlCon.Close();

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    sqlCon.Close();
}
}

public void clearData()
{
    textbox_Pcode.Clear();
    textbox_Pname.Clear();
    textbox_Pdesc.Clear();
```

```
cb_Pcategory.SelectedItem = null;
cb_Pvendor.SelectedItem = null;
textbox_Plocation.Clear();
textbox_Punit.Clear();
textbox_Pcp.Clear();
textbox_Psp.Clear();
textbox_Pbuffer.Clear();
textbox_Pstock.Clear();
textbox_Pdog.Clear();
```

```
}
```

```
public bool isValid()
{
    if(textbox_Pcode.Text == string.Empty)
    {
        MessageBox.Show("Product Code is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return false;
    }
    if (textbox_Pname.Text == string.Empty)
    {
        MessageBox.Show("Product Name is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return false;
    }
    if (cb_Pcategory.SelectedItem == null)
    {
        MessageBox.Show("Product Category is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
        return false;
    }
    if (cb_Pvendor.SelectedItem == null)
    {
        MessageBox.Show("Product Vendor is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
        return false;
    }
    if (textbox_Plocation.Text == string.Empty)
```

Page | 55

```
        MessageBox.Show("Product Storage Location is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
    return false;
}
if (textbox_Punit.Text == string.Empty)
{
    MessageBox.Show("Product Measuring Unit is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
    return false;
}
if (textbox_Pcp.Text == string.Empty)
{
    MessageBox.Show("Cost Price per Unit is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
    return false;
}
if (textbox_Psp.Text == string.Empty)
{
    MessageBox.Show("Selling Price per Unit is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
    return false;
}
if (textbox_Pbuffer.Text == string.Empty)
{
    MessageBox.Show("Product Buffer Quantity Unit is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
    return false;
}
if (textbox_Pstock.Text == string.Empty)
{
    MessageBox.Show("Product Stock Quantity Unit is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
    return false;
}
if (textbox_Pdog.Text == string.Empty)
{
    MessageBox.Show("Product Date Of Generation is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
    return false;
}
return true;
}

private void btn_PFormSave_Click(object sender, RoutedEventArgs e)
```

```
Page | 56
{
    try
    {
        if (isValid())
        {
            SqlCommand cmd = new SqlCommand("INSERT INTO tblProduct VALUES (@Product_Code, @Product_Name, @Description, @Category, @Vendor, @Location, @Unit, @CP_Unit, @SP_Unit, @Buffer_Quantity, @Stock_Quantity, @Date)", sqlCon);
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@Product_Code", textbox_Pcode.Text);
            cmd.Parameters.AddWithValue("@Product_Name", textbox_Pname.Text);
            cmd.Parameters.AddWithValue("@Description", textbox_Pdesc.Text);
            cmd.Parameters.AddWithValue("@Category",
cb_Pcategory.SelectedValue.ToString());
            cmd.Parameters.AddWithValue("@Vendor",
cb_Pvendor.SelectedValue.ToString());
            cmd.Parameters.AddWithValue("@Location", textbox_Plocation.Text);
            cmd.Parameters.AddWithValue("@Unit", textbox_Punit.Text);
            cmd.Parameters.AddWithValue("@CP_Unit", textbox_Pcp.Text);
            cmd.Parameters.AddWithValue("@SP_Unit", textbox_Psp.Text);
            cmd.Parameters.AddWithValue("@Buffer_Quantity", textbox_Pbuffer.Text);
            cmd.Parameters.AddWithValue("@Stock_Quantity", textbox_Pstock.Text);
            cmd.Parameters.AddWithValue("@Date", (textbox_Pdog.Text));
            sqlCon.Open();
            cmd.ExecuteNonQuery();
            sqlCon.Close();
            MessageBox.Show("New Product Successfully Added");
            clearData();
        }
    }
    catch (SqlException ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}

private void btn_pFormUpdate_Click(object sender, RoutedEventArgs e)
{
    sqlCon.Open();
```

```
Page | 57
SqlCommand cmd = new SqlCommand("UPDATE tblProduct set Product_Name='" +
textbox_Pname.Text + "', Description ='" + textbox_Pdesc.Text + "', Category =
""+cb_Pcategory.SelectedValue+"", Vendor = ""+cb_Pvendor.SelectedValue+"", Location =
"+ textbox_Plocation + "", Unit = ""+textbox_Punit+"", Vendor = ""+textbox_Pcp+"", Vendor =
""+textbox_Psp+"", Vendor = ""+textbox_Pbuffer+"", Vendor = ""+textbox_Pstock+"", Vendor =
""+textbox_Pdog+""
 WHERE Product_Code = '" + textbox_Pcode.Text + "'", sqlCon);
try
{
    cmd.ExecuteNonQuery();
    MessageBox.Show("Record has been updated successfully");
}

catch (SqlException ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    sqlCon.Close();
    clearData();

}
}
```

## PurchaseBillEntry.xaml

```
<Page x:Class="testWPF.PurchaseBillEntry"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="PurchaseBillEntry" Background="DarkSeaGreen" Foreground="White"
    FontSize="16">

    <StackPanel Orientation="Vertical">
        <TextBlock Text="PURCHASE BILL ENTRY" HorizontalAlignment="Center" FontSize="30"
        FontWeight="ExtraBold" Margin="0 25 0 0"/>

        <StackPanel Orientation="Horizontal" >
            <Button x:Name="btnPBEAdd" Click="btnPBEAdd_Click" Content="ADD"
            Width="100" Height="30" Margin="25 25 25 10"/>
            <TextBox x:Name="tb_PBEsearch" Width="200" Height="30" Margin="50 25 0 10"/>
            <Button x:Name="btnPBView" Click="btnPBView_Click" Content="VIEW"
            Height="30" Width="100" Margin="10 25 25 10"/>
        </StackPanel>

        <Separator Background="Black"></Separator>
        <DataGrid Height="200" x:Name="dg_PBView" Margin="30,0,145,0" />
    </StackPanel>
</Page>
```



## PurchaseBillEntry.xaml.cs

```
Page | 59
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for PurchaseBillEntry.xaml
    /// </summary>
    public partial class PurchaseBillEntry : Page
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public PurchaseBillEntry()
        {
            InitializeComponent();
        }

        //TO OPEN PBEForm WINDOW
        private void btnPBEAdd_Click(object sender, RoutedEventArgs e)
        {
            PBEForm pbef = new PBEForm();
            pbef.Show();
        }
    }
}
```

```
//TO LOAD DATAGRID WITH TABLE DATA
private void btnPBEView_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        string query = "SELECT Invoice_Id 'Invoice No', PB_ID, Vendor_Code,
Product_Code, Quantity_Received, Purchase_DOE FROM tblPurchaseBill WHERE
Invoice_Id=" + tb_PBEsearch.Text + "";
        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
        sqlcmd.CommandType = System.Data.CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataReader sdr = sqlcmd.ExecuteReader();

        dt.Load(sdr);
        dg_PBEview.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}
}
```

## PBEForm.xaml

```
Page | 61
<Window x:Class="testWPF.PBEForm"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    Title="PBEForm" Height="500" Width="1100" Background="DarkSeaGreen"
    Foreground="White" FontSize="18">
    <Window.Resources>
        <Style TargetType="TextBox">
            <Setter Property="BorderBrush" Value="Black"/>
            <Setter Property="BorderThickness" Value="2"/>
        </Style>
        <Style TargetType="Label">
            <Setter Property="BorderBrush" Value="Black"/>
            <Setter Property="BorderThickness" Value="2"/>
            <Setter Property="Background" Value="White"/>
            <Setter Property="Width" Value="200"/>
            <Setter Property="FontWeight" Value="SemiBold"/>
        </Style>
    </Window.Resources>
    <ScrollViewer HorizontalScrollBarVisibility="Visible" VerticalScrollBarVisibility="Visible">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width ="20"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="20"/>
            </Grid.ColumnDefinitions>
            <Grid.RowDefinitions>
                <RowDefinition Height="20"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
            </Grid.RowDefinitions>
        </Grid>
    </ScrollViewer>

```

```
<RowDefinition Height="Auto"/>
<RowDefinition Height="20"/>
</Grid.RowDefinitions>
<TextBlock Grid.Row="1" Grid.Column="3" Grid.ColumnSpan="3" Text="PURCHASE
BILL ENTRY" Margin="10" FontSize="20" FontWeight="ExtraBlack">

    <TextBlock Grid.Row="2" Grid.Column="1" Text="INVOICE NO. " Margin="0 5 0 5"
FontWeight="SemiBold"/>
    <TextBox Grid.Row="2" Grid.Column="2" x:Name="tb_PBEinvoice" Margin="0 5 0
5"/>

    <TextBlock Grid.Row="3" Grid.Column="1" Text="VENDOR" Margin="0 5 0 5"
FontWeight="SemiBold"/>
    <ComboBox Grid.Row="3" Grid.Column="2" x:Name="cb_PBEvendor" Margin="0 5 0
5" SelectionChanged="cb_PBEvendor_SelectionChanged">
        </ComboBox>

    <TextBlock Grid.Row="4" Grid.Column="1" Text="DATE" Width="150" Margin="0 5 0
5" FontWeight="SemiBold"/>
    <TextBox Grid.Row="4" Grid.Column="2" x:Name="tb_PBEdate" Width="200"
Margin="0 5 0 10"/>

    <Label Grid.Row="5" Grid.Column="1" Content="Product Code"/>
    <Label Grid.Row="5" Grid.Column="2" Content="Product Name"/>
    <Label Grid.Row="5" Grid.Column="3" Content="CP/Unit"/>
    <Label Grid.Row="5" Grid.Column="4" Content="Quantity"/>
    <Label Grid.Row="5" Grid.Column="5" Content="Purchase Price"/>

    <ComboBox Grid.Row="6" Grid.Column="1" x:Name="cb_PBEpc1"
SelectionChanged="cb_PBEpc1_SelectionChanged" >
        </ComboBox>
```

Page | 63

```
<ComboBox Grid.Row="7" Grid.Column="1" x:Name="cb_PBEpc2"
SelectionChanged="cb_PBEpc2_SelectionChanged">

</ComboBox>
<ComboBox Grid.Row="8" Grid.Column="1" x:Name="cb_PBEpc3"
SelectionChanged="cb_PBEpc3_SelectionChanged">

</ComboBox>
<ComboBox Grid.Row="9" Grid.Column="1" x:Name="cb_PBEpc4"
SelectionChanged="cb_PBEpc4_SelectionChanged">

</ComboBox>
<ComboBox Grid.Row="10" Grid.Column="1" x:Name="cb_PBEpc5"
SelectionChanged="cb_PBEpc5_SelectionChanged">

</ComboBox>

<TextBox Grid.Row="6" Grid.Column="2" x:Name="tb_PBEpn1" />
<TextBox Grid.Row="7" Grid.Column="2" x:Name="tb_PBEpn2" />
<TextBox Grid.Row="8" Grid.Column="2" x:Name="tb_PBEpn3" />
<TextBox Grid.Row="9" Grid.Column="2" x:Name="tb_PBEpn4" />
<TextBox Grid.Row="10" Grid.Column="2" x:Name="tb_PBEpn5" />

<TextBox Grid.Row="6" Grid.Column="3" x:Name="tb_PBEcp1" />
<TextBox Grid.Row="7" Grid.Column="3" x:Name="tb_PBEcp2" />
<TextBox Grid.Row="8" Grid.Column="3" x:Name="tb_PBEcp3" />
<TextBox Grid.Row="9" Grid.Column="3" x:Name="tb_PBEcp4" />
<TextBox Grid.Row="10" Grid.Column="3" x:Name="tb_PBEcp5" />

<TextBox Grid.Row="6" Grid.Column="4" x:Name="tb_PBEq1"
LostFocus="tb_PBEq1_LostFocus"/>
<TextBox Grid.Row="7" Grid.Column="4" x:Name="tb_PBEq2"
LostFocus="tb_PBEq2_LostFocus" />
<TextBox Grid.Row="8" Grid.Column="4" x:Name="tb_PBEq3"
LostFocus="tb_PBEq3_LostFocus" />
<TextBox Grid.Row="9" Grid.Column="4" x:Name="tb_PBEq4"
LostFocus="tb_PBEq4_LostFocus" />
<TextBox Grid.Row="10" Grid.Column="4" x:Name="tb_PBEq5"
LostFocus="tb_PBEq5_LostFocus" />

<TextBox Grid.Row="6" Grid.Column="5" x:Name="tb_PBEp1" />
<TextBox Grid.Row="7" Grid.Column="5" x:Name="tb_PBEp2" />
<TextBox Grid.Row="8" Grid.Column="5" x:Name="tb_PBEp3" />
<TextBox Grid.Row="9" Grid.Column="5" x:Name="tb_PBEp4" />
```

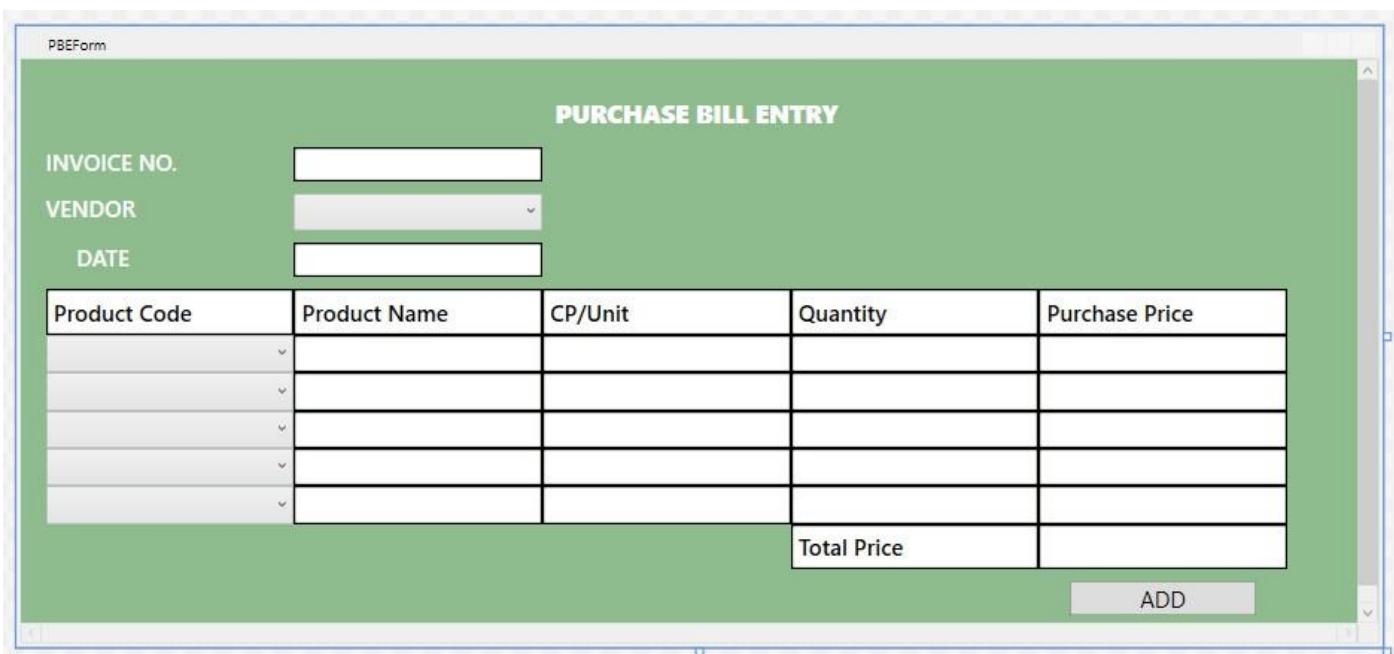
```

<TextBox Grid.Row="10" Grid.Column="5" x:Name="tb_PBEp5" />

<Label Grid.Row="11" Grid.Column="4" Content="Total Price"/>
<TextBox Grid.Row="11" Grid.Column="5" x:Name="tb_PBEtp"/>

Page | 64
      <Button x:Name="btn_AddPB" Click="btn_AddPB_Click" Grid.Row="12"
      Grid.Column="5" Content="ADD" Width="150" Margin="0 10 0 0"/>
    </Grid>
  </ScrollViewer>
</Window>

```



## PBEForm.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;

```

```
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

Page | 65
namespace testWPF
{
    /// <summary>
    /// Interaction logic for PBEForm.xaml
    /// </summary>
    public partial class PBEForm : Window
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        bool LoadVendor = true;
        public PBEForm()
        {
            InitializeComponent();
            if (LoadVendor)
                loadComboBoxVendor();
        }
        //FUNCTION TO LOAD VENDOR COMBOBOX
        private void loadComboBoxVendor()
        {

            if (sqlCon.State == System.Data.ConnectionState.Closed)
                sqlCon.Open();

            string query = "SELECT vendor_code, vendor_name FROM tblVendor";
            SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
            sqlcmd.CommandType = System.Data.CommandType.Text;
            DataTable dt = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(sqlcmd);

            adapter.Fill(dt);
            cb_PBEvendor.ItemsSource = dt.DefaultView;
            cb_PBEvendor.DisplayMemberPath = "vendor_name";
            cb_PBEvendor.SelectedValuePath = "vendor_code";
            sqlcmd.Dispose();

            sqlCon.Close();
        }
    }
}
```

}

//FUNCTION TO CLEAR TEXTBOXES AND UNSELECT COMBOBOX

public void clearData()

{

LoadVendor = false;

tb\_PBEinvoice.Clear();

cb\_PBEvendor.SelectedItem = null;

tb\_PBEdate.Clear();

cb\_PBEpc1.SelectedItem = null;

cb\_PBEpc2.SelectedItem = null;

cb\_PBEpc3.SelectedItem = null;

cb\_PBEpc4.SelectedItem = null;

cb\_PBEpc5.SelectedItem = null;

tb\_PBEpn1.Clear();

tb\_PBEpn2.Clear();

tb\_PBEpn3.Clear();

tb\_PBEpn4.Clear();

tb\_PBEpn5.Clear();

tb\_PBEcp1.Clear();

tb\_PBEcp2.Clear();

tb\_PBEcp3.Clear();

tb\_PBEcp4.Clear();

tb\_PBEcp5.Clear();

tb\_PBEq1.Clear();

tb\_PBEq2.Clear();

tb\_PBEq3.Clear();

tb\_PBEq4.Clear();

tb\_PBEq5.Clear();

tb\_PBEp1.Clear();

tb\_PBEp2.Clear();

tb\_PBEp3.Clear();

tb\_PBEp4.Clear();

tb\_PBEp5.Clear();

tb\_PBEtp.Clear();

}

Page | 67

```
//FUNCTION TO CHECK BEFORE ADDING DATA TO DATABASE
public bool isValid()
{
    if (tb_PBEinvoice.Text == string.Empty)
    {
        MessageBox.Show("Invoice No. is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return false;
    }
    if (cb_PBEvendor.SelectedItem == null)
    {
        MessageBox.Show("Vendor is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return false;
    }
    if (tb_PBEdate.Text == string.Empty)
    {
        MessageBox.Show("Date is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return false;
    }

    return true;
}

//ADD CLICK EVENT FUNCTION TO ADD DATA TO tblPurchaseBill TABLE
private void btn_AddPB_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        if (isValid())
        {
            bool k = false;
            if (tb_PBEq1.Text != null && tb_PBEq1.Text != "")
            {

                SqlCommand cmd = new SqlCommand("INSERT INTO tblPurchaseBill VALUES
(@Invoice_Id, @Vendor_Code, @Product_Code, @Quantity_Received, @Purchase_DOE)",
sqlCon);
                cmd.CommandType = CommandType.Text;
```

```
cmd.Parameters.AddWithValue("@Invoice_Id", tb_PBEinvoice.Text);
cmd.Parameters.AddWithValue("@Vendor_Code",
cb_PBEvendor.SelectedValue);
cmd.Parameters.AddWithValue("@Product_Code",
cb_PBEpc1.SelectedValue);
cmd.Parameters.AddWithValue("@Quantity_Received",
Convert.ToInt32(tb_PBEq1.Text));
cmd.Parameters.AddWithValue("@Purchase_DOE", tb_PBEdate.Text);

cmd.ExecuteNonQuery();

int Stock_Quantity
=Convert.ToInt32(((System.Data.DataRowView)(cb_PBEpc1.SelectedItem))["Stock_Quantity
"].ToString());
Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_PBEq1.Text);

SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" +
cb_PBEpc1.SelectedValue+";", sqlCon);
cmd1.CommandType = CommandType.Text;

cmd1.ExecuteNonQuery();

// SqlCommand cmd2 = new SqlCommand("INSERT INTO tblPurchaseBilltotal
VALUES (@Invoice_Id, @total_bill)", sqlCon);
// cmd2.CommandType = CommandType.Text;
// cmd2.Parameters.AddWithValue("@Invoice_Id", tb_PBEinvoice.Text);
// cmd2.Parameters.AddWithValue("@total_bill", tb_PBEtp.Text);

// cmd2.ExecuteNonQuery();

k = true;
}

if (tb_PBEq2.Text != null && tb_PBEq2.Text != "")
{
    SqlCommand cmd = new SqlCommand("INSERT INTO tblPurchaseBill VALUES
(@Invoice_Id, @Vendor_Code, @Product_Code, @Quantity_Received, @Purchase_DOE)",
sqlCon);
```

```
cmd.CommandType = CommandType.Text;
cmd.Parameters.AddWithValue("@Invoice_Id", tb_PBEinvoice.Text);
cmd.Parameters.AddWithValue("@Vendor_Code",
cb_PBEvendor.SelectedValue);
cmd.Parameters.AddWithValue("@Product_Code",
cb_PBEpc2.SelectedValue);
cmd.Parameters.AddWithValue("@Quantity_Received", tb_PBEq2.Text);
cmd.Parameters.AddWithValue("@Purchase_DOE", System.DateTime.Today);

cmd.ExecuteNonQuery();

int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_PBEpc2.SelectedItem))["Stock_Quantity"]
].ToString());
Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_PBEq2.Text);

SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" +
cb_PBEpc2.SelectedValue + ";", sqlCon);
cmd1.CommandType = CommandType.Text;

cmd1.ExecuteNonQuery();

k = true;
}
if (tb_PBEq3.Text != null && tb_PBEq3.Text != "")
{
    SqlCommand cmd = new SqlCommand("INSERT INTO tblPurchaseBill VALUES
(@Invoice_Id, @Vendor_Code, @Product_Code, @Quantity_Received, @Purchase_DOE",
sqlCon);
cmd.CommandType = CommandType.Text;
cmd.Parameters.AddWithValue("@Invoice_Id", tb_PBEinvoice.Text);
cmd.Parameters.AddWithValue("@Vendor_Code",
cb_PBEvendor.SelectedValue);
cmd.Parameters.AddWithValue("@Product_Code",
cb_PBEpc3.SelectedValue);
cmd.Parameters.AddWithValue("@Quantity_Received", tb_PBEq3.Text);
cmd.Parameters.AddWithValue("@Purchase_DOE", System.DateTime.Today);

cmd.ExecuteNonQuery();
```

```

    int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_PBEpc3.SelectedItem))["Stock_Quantity"]
].ToString());
    Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_PBEq3.Text);

Page | 70
    SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" +
cb_PBEpc3.SelectedValue + ";", sqlCon);
    cmd1.CommandType = CommandType.Text;

    cmd1.ExecuteNonQuery();

    k = true;
}
if (tb_PBEq4.Text != null && tb_PBEq4.Text != "")
{
    SqlCommand cmd = new SqlCommand("INSERT INTO tblPurchaseBill VALUES
(@Invoice_Id, @Vendor_Code, @Product_Code, @Quantity_Received, @Purchase_DOE",
sqlCon);
    cmd.CommandType = CommandType.Text;
    cmd.Parameters.AddWithValue("@Invoice_Id", tb_PBEinvoice.Text);
    cmd.Parameters.AddWithValue("@Vendor_Code",
cb_PBEvendor.SelectedValue);
    cmd.Parameters.AddWithValue("@Product_Code",
cb_PBEpc4.SelectedValue);
    cmd.Parameters.AddWithValue("@Quantity_Received", tb_PBEq4.Text);
    cmd.Parameters.AddWithValue("@Purchase_DOE", System.DateTime.Today);

    cmd.ExecuteNonQuery();

    int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_PBEpc4.SelectedItem))["Stock_Quantity"]
].ToString());
    Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_PBEq4.Text);

    SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" +
cb_PBEpc4.SelectedValue + ";", sqlCon);
    cmd1.CommandType = CommandType.Text;

    cmd1.ExecuteNonQuery();

```

```

        k = true;
    }
    if (tb_PBEq5.Text != null && tb_PBEq5.Text != "")
    {
        SqlCommand cmd = new SqlCommand("INSERT INTO tblPurchaseBill VALUES
(@Invoice_Id, @Vendor_Code, @Product_Code, @Quantity_Received, @Purchase_DOE",
sqlCon);
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@Invoice_Id", tb_PBEinvoice.Text);
        cmd.Parameters.AddWithValue("@Vendor_Code",
cb_PBEvendor.SelectedValue);
        cmd.Parameters.AddWithValue("@Product_Code",
cb_PBEpc5.SelectedValue);
        cmd.Parameters.AddWithValue("@Quantity_Received", tb_PBEq5.Text);
        cmd.Parameters.AddWithValue("@Purchase_DOE", System.DateTime.Today);

        cmd.ExecuteNonQuery();

        int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_PBEpc5.SelectedItem))["Stock_Quantity"]
].ToString());
        Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_PBEq5.Text);

        SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" +
cb_PBEpc5.SelectedValue + ";", sqlCon);
        cmd1.CommandType = CommandType.Text;

        k = true;
    }

    if (k == true)
    {
        MessageBox.Show("Purchase Bill Entry Successfully");
    }
    else
    {
        MessageBox.Show("Some Field Missing. Entry Unsuccessful.");
    }

}

```

```
    }
    catch (SqlException ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}

private void cb_PBEvendor_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{
    loadComboBoxProduct();
}

//FUNCTION TO LOAD PRODUCT COMBOBOX
private void loadComboBoxProduct()
{
    if (sqlCon.State == System.Data.ConnectionState.Closed)
        sqlCon.Open();
    string query = "select [Product_Name],[Product_Code],[CP_Unit],[Stock_Quantity]
from [IMDB].[dbo].[tblProduct] where [Vendor]='" +
cb_PBEvendor.SelectedValue.ToString() + "'";
    SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
    sqlcmd.CommandType = System.Data.CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter adapter = new SqlDataAdapter(sqlcmd);

    adapter.Fill(dt);
    cb_PBEpc1.ItemsSource = dt.DefaultView;
    cb_PBEpc1.DisplayMemberPath = "Product_Code";
    cb_PBEpc1.SelectedValuePath = "Product_Code";
    cb_PBEpc2.ItemsSource = dt.DefaultView;
    cb_PBEpc2.DisplayMemberPath = "Product_Code";
    cb_PBEpc2.SelectedValuePath = "Product_Code";
    cb_PBEpc3.ItemsSource = dt.DefaultView;
    cb_PBEpc3.DisplayMemberPath = "Product_Code";
    cb_PBEpc3.SelectedValuePath = "Product_Code";
    cb_PBEpc4.ItemsSource = dt.DefaultView;
    cb_PBEpc4.DisplayMemberPath = "Product_Code";
    cb_PBEpc4.SelectedValuePath = "Product_Code";
```

```
cb_PBEpc5.ItemsSource = dt.DefaultView;
cb_PBEpc5.DisplayMemberPath = "Product_Code";
cb_PBEpc5.SelectedValuePath = "Product_Code";

sqlcmd.Dispose();

sqlCon.Close();
}

private void cb_PBEpc1_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {

        tb_PBEcp1.Text =
((System.Data.DataRowView)(cb_PBEpc1.SelectedItem))["CP_Unit"].ToString();
        tb_PBEpn1.Text =
((System.Data.DataRowView)(cb_PBEpc1.SelectedItem))["Product_Name"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }

    finally
    {

        sqlCon.Close();
    }
}

private void cb_PBEpc2_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {

        tb_PBEcp2.Text =
((System.Data.DataRowView)(cb_PBEpc2.SelectedItem))["CP_Unit"].ToString();
    }
}
```

```
tb_PBEpn2.Text =
((System.Data.DataRowView)(cb_PBEpc2.SelectedItem))["Product_Name"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }

    finally
    {

        //sqlCon.Close();
    }

}

private void cb_PBEpc3_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {

        tb_PBEcp3.Text =
((System.Data.DataRowView)(cb_PBEpc3.SelectedItem))["CP_Unit"].ToString();
        tb_PBEpn3.Text =
((System.Data.DataRowView)(cb_PBEpc3.SelectedItem))["Product_Name"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }

    finally
    {

        //sqlCon.Close();
    }

}
```

```
private void cb_PBEpc4_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {
        tb_PBEcp4.Text =
((System.Data.DataRowView)(cb_PBEpc4.SelectedItem))["CP_Unit"].ToString();
        tb_PBEpn4.Text =
((System.Data.DataRowView)(cb_PBEpc4.SelectedItem))["Product_Name"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }

    finally
    {

        //sqlCon.Close();
    }
}

private void cb_PBEpc5_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {
        tb_PBEcp5.Text =
((System.Data.DataRowView)(cb_PBEpc5.SelectedItem))["CP_Unit"].ToString();
        tb_PBEpn5.Text =
((System.Data.DataRowView)(cb_PBEpc5.SelectedItem))["Product_Name"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }
}
```

```
finally
{
    //sqlCon.Close();
}

}
private void tb_PBEq1_LostFocus(object sender, RoutedEventArgs e)
{
    double Cost_price = Convert.ToDouble(tb_PBEcp1.Text);
    int Quatity = Convert.ToInt32(tb_PBEq1.Text);
    tb_PBEp1.Text = (Quatity * Cost_price).ToString();
    if (tb_PBEp1.Text != null)
        tb_PBEtp.Text = tb_PBEp1.Text;
}

private void tb_PBEq2_LostFocus(object sender, RoutedEventArgs e)
{
    double Cost_price = Convert.ToDouble(tb_PBEcp2.Text);
    int Quatity = Convert.ToInt32(tb_PBEq2.Text);
    tb_PBEp2.Text = (Quatity * Cost_price).ToString();
    if (tb_PBEp2.Text != null && tb_PBEp1.Text != null)
    {
        double total1 = Convert.ToDouble(tb_PBEp1.Text);
        int total2 = (int)Convert.ToDouble(tb_PBEp2.Text);

        tb_PBEtp.Text = (total1 + total2).ToString();
    }
}

private void tb_PBEq3_LostFocus(object sender, RoutedEventArgs e)
{
    double Cost_price = Convert.ToDouble(tb_PBEcp3.Text);
    int Quatity = Convert.ToInt32(tb_PBEq3.Text);
    tb_PBEp3.Text = (Quatity * Cost_price).ToString();
    if (tb_PBEp2.Text != null && tb_PBEp1.Text != null && tb_PBEp3.Text != null)
    {
        double total1 = Convert.ToDouble(tb_PBEp1.Text);
        double total2 = Convert.ToDouble(tb_PBEp2.Text);
        double total3 = Convert.ToDouble(tb_PBEp3.Text);

        tb_PBEtp.Text = (total1 + total2 + total3).ToString();
    }
}
```

```
//tb_PBEtp.Text = (Convert.ToDouble(tb_PBEp1.Text) +
Convert.ToDouble(tb_PBEp2.Text) + Convert.ToDouble(tb_PBEp3.Text)).ToString();
}

private void tb_PBEq4_LostFocus(object sender, RoutedEventArgs e)
{
    double Cost_price = Convert.ToDouble(tb_PBEcp4.Text);
    int Quatity = Convert.ToInt32(tb_PBEq4.Text);
    tb_PBEp4.Text = (Quatity * Cost_price).ToString();
    if (tb_PBEp2.Text != null && tb_PBEp1.Text != null && tb_PBEp3.Text != null &&
tb_PBEp4.Text != null)
        tb_PBEtp.Text = (Convert.ToDouble(tb_PBEp1.Text) +
Convert.ToDouble(tb_PBEp2.Text) + Convert.ToDouble(tb_PBEp3.Text) +
Convert.ToDouble(tb_PBEp4.Text)).ToString();

}

private void tb_PBEq5_LostFocus(object sender, RoutedEventArgs e)
{
    double Cost_price = Convert.ToDouble(tb_PBEcp5.Text);
    int Quatity = Convert.ToInt32(tb_PBEq5.Text);
    tb_PBEp5.Text = (Quatity * Cost_price).ToString();
    if (tb_PBEp2.Text != null && tb_PBEp1.Text != null && tb_PBEp3.Text != null &&
tb_PBEp4.Text != null && tb_PBEp5.Text != null)
        tb_PBEtp.Text = (Convert.ToDouble(tb_PBEp1.Text) +
Convert.ToDouble(tb_PBEp2.Text) +
+ Convert.ToDouble(tb_PBEp3.Text) + Convert.ToDouble(tb_PBEp4.Text) +
Convert.ToDouble(tb_PBEp5.Text)).ToString();

}
}
```

## SALEACCEPTANCEENTRY.xaml

```
Page | 78
<Page x:Class="testWPF.SaleAcceptanceEntry"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:testWPF"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="SaleAcceptanceEntry" Background="DarkSeaGreen" Foreground="White"
      FontSize="16">
    <StackPanel Orientation="Vertical">
        <TextBlock Text="SALE ACCEPTANCE ENTRY" HorizontalAlignment="Center"
        FontSize="30" FontWeight="ExtraBold" Margin="0 25 0 0"/>
        <StackPanel Orientation="Horizontal" >
            <Button x:Name="btnSAEAdd" Click="btnSAEAdd_Click" Content="ADD"
            Width="100" Height="30" Margin="25 25 25 10"/>
            <TextBox x:Name="tb_SAEsearch" Width="200" Height="30" Margin="50 25 0 10"/>
            <Button x:Name="btnSAEView" Click="btnSAEView_Click" Content="VIEW"
            Height="30" Width="100" Margin="25 25 25 10"/>
        </StackPanel>
        <Separator Background="Black"></Separator>
        <DataGrid x:Name="dg_SAEview" HorizontalAlignment="Left" Height="221"
        Margin="30,0,0,0" Width="734"/>
    </StackPanel>
</Page>
```



## SaleAcceptanceENTRY.xaml.cs

```
Page | 79
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for SaleAcceptanceEntry.xaml
    /// </summary>
    public partial class SaleAcceptanceEntry : Page
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public SaleAcceptanceEntry()
        {
            InitializeComponent();
        }

        //TO OPEN SAEForm WINDOW ON CLICKING ADD BUTTON
        private void btnSAEAdd_Click(object sender, RoutedEventArgs e)
        {
            SAEform sbef = new SAEform();
            sbef.Show();
        }
    }
}
```

Page | 80

```
//TO LOAD DATAGRID WITH DATA TABLE
private void btnSAEView_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        string query = "SELECT Sale_Id 'Sale No', SB_ID, Customer_Name,
Customer_Contact, Product_Code, Quantity_Sold, Sale_DOE FROM tblSaleBill WHERE
Sale_Id=" + tb_SAEsearch.Text + "";
        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
        sqlcmd.CommandType = System.Data.CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataReader sdr = sqlcmd.ExecuteReader();

        dt.Load(sdr);
        dg_SAEview.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}
```

## SAEform.xaml

```
Page | 81
<Window x:Class="testWPF.SAEform"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    Title="SAEform" Height="550" Width="1100" Background="DarkSeaGreen"
    Foreground="White" FontSize="18">
    <Window.Resources>
        <Style TargetType="TextBox">
            <Setter Property="BorderBrush" Value="Black"/>
            <Setter Property="BorderThickness" Value="2"/>
            <Setter Property="Width" Value="200"/>
        </Style>
        <Style TargetType="Label">
            <Setter Property="BorderBrush" Value="Black"/>
            <Setter Property="BorderThickness" Value="2"/>
            <Setter Property="Background" Value="White"/>
            <Setter Property="Width" Value="200"/>
            <Setter Property="FontWeight" Value="SemiBold"/>
        </Style>
    </Window.Resources>
    <ScrollViewer>
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width ="20"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="*"/>
                <ColumnDefinition Width="20"/>
            </Grid.ColumnDefinitions>
            <Grid.RowDefinitions>
                <RowDefinition Height="20"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="Auto"/>
            </Grid.RowDefinitions>
```

```
<RowDefinition Height="Auto"/>
<RowDefinition Height="20"/>
</Grid.RowDefinitions>
<TextBlock Grid.Row="1" Grid.Column="3" Grid.ColumnSpan="3" Text="SALE  
ACCEPTANCE ENTRY" Margin="10" FontSize="20" FontWeight="ExtraBlack"/>

<TextBlock Grid.Row="2" Grid.Column="1" Text="SALE BILL NO. " Margin="0 5 0 5"  
FontWeight="SemiBold"/>
<TextBox Grid.Row="2" Grid.Column="2" x:Name="tb_SAEbill" Margin="0 5 0 5"/>

<TextBlock Grid.Row="3" Grid.Column="1" Grid.ColumnSpan="1" Text="CUSTOMER  
NAME" Margin="0 5 0 5" FontWeight="SemiBold"/>
<TextBox Grid.Row="3" Grid.Column="2" x:Name="tb_SAEcname" Margin="0 5 0  
5"/>

<TextBlock Grid.Row="4" Grid.Column="1" Grid.ColumnSpan="1" Text="Contact"  
Margin="0 5 0 5" FontWeight="SemiBold"/>
<TextBox Grid.Row="4" Grid.Column="2" x:Name="tb_SAEcno" Margin="0 5 0 5"/>

<TextBlock Grid.Row="5" Grid.Column="1" Text="DATE" Width="200" Margin="0 5 0  
5" FontWeight="SemiBold"/>
<TextBox Grid.Row="5" Grid.Column="2" x:Name="tb_SAEdate" Margin="0 5 0 10"/>

<Label Grid.Row="6" Grid.Column="1" Content="Product Name"/>
<Label Grid.Row="6" Grid.Column="2" Content="Product Code"/>
<Label Grid.Row="6" Grid.Column="3" Content="SP/Unit"/>
<Label Grid.Row="6" Grid.Column="4" Content="Quantity"/>
<Label Grid.Row="6" Grid.Column="5" Content="Sale Price"/>

<ComboBox Grid.Row="7" Grid.Column="1" x:Name="cb_SAEpn1"  
SelectionChanged="cb_SAEpn1_SelectionChanged">
```

```
</ComboBox>
<ComboBox Grid.Row="8" Grid.Column="1" x:Name="cb_SAEpn2"
SelectionChanged="cb_SAEpn2_SelectionChanged">

</ComboBox>
<ComboBox Grid.Row="9" Grid.Column="1" x:Name="cb_SAEpn3"
SelectionChanged="cb_SAEpn3_SelectionChanged">

</ComboBox>
<ComboBox Grid.Row="10" Grid.Column="1" x:Name="cb_SAEpn4"
SelectionChanged="cb_SAEpn4_SelectionChanged">

</ComboBox>
<ComboBox Grid.Row="11" Grid.Column="1" x:Name="cb_SAEpn5"
SelectionChanged="cb_SAEpn5_SelectionChanged">

</ComboBox>

<TextBox Grid.Row="7" Grid.Column="2" x:Name="tb_SAEpc1" />
<TextBox Grid.Row="8" Grid.Column="2" x:Name="tb_SAEpc2" />
<TextBox Grid.Row="9" Grid.Column="2" x:Name="tb_SAEpc3" />
<TextBox Grid.Row="10" Grid.Column="2" x:Name="tb_SAEpc4" />
<TextBox Grid.Row="11" Grid.Column="2" x:Name="tb_SAEpc5" />

<TextBox Grid.Row="7" Grid.Column="3" x:Name="tb_SAEsp1" />
<TextBox Grid.Row="8" Grid.Column="3" x:Name="tb_SAEsp2" />
<TextBox Grid.Row="9" Grid.Column="3" x:Name="tb_SAEsp3" />
<TextBox Grid.Row="10" Grid.Column="3" x:Name="tb_SAEsp4" />
<TextBox Grid.Row="11" Grid.Column="3" x:Name="tb_SAEsp5" />

<TextBox Grid.Row="7" Grid.Column="4" x:Name="tb_SAEq1"
LostFocus="tb_SAEq1_LostFocus" />
<TextBox Grid.Row="8" Grid.Column="4" x:Name="tb_SAEq2"
LostFocus="tb_SAEq2_LostFocus" />
<TextBox Grid.Row="9" Grid.Column="4" x:Name="tb_SAEq3"
LostFocus="tb_SAEq3_LostFocus" />
<TextBox Grid.Row="10" Grid.Column="4" x:Name="tb_SAEq4"
LostFocus="tb_SAEq4_LostFocus" />
<TextBox Grid.Row="11" Grid.Column="4" x:Name="tb_SAEq5"
LostFocus="tb_SAEq5_LostFocus" />

<TextBox Grid.Row="7" Grid.Column="5" x:Name="tb_SAEp1" />
<TextBox Grid.Row="8" Grid.Column="5" x:Name="tb_SAEp2" />
<TextBox Grid.Row="9" Grid.Column="5" x:Name="tb_SAEp3" />
```

Page | 84

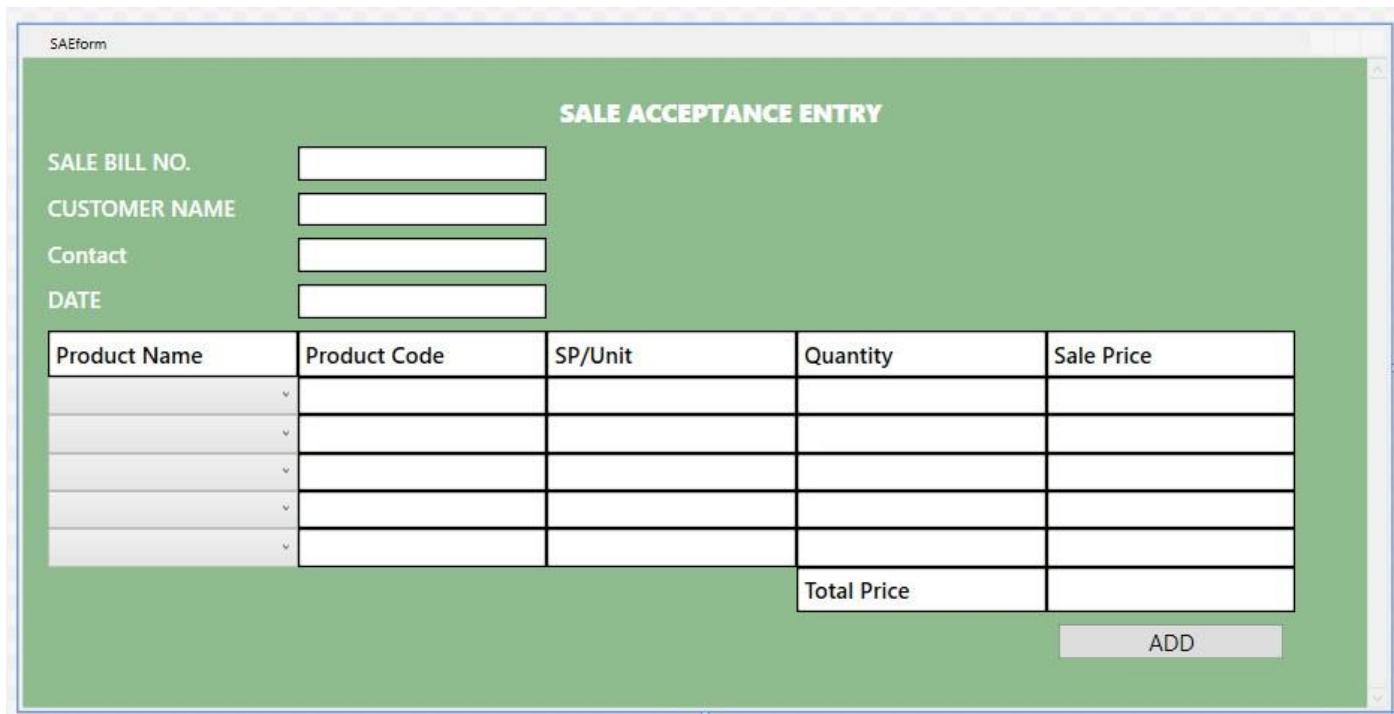
```

<TextBox Grid.Row="10" Grid.Column="5" x:Name="tb_SAEp4" />
<TextBox Grid.Row="11" Grid.Column="5" x:Name="tb_SAEp5" />

<Label Grid.Row="12" Grid.Column="4" Content="Total Price"/>
<TextBox Grid.Row="12" Grid.Column="5" x:Name="tb_SAEtp"/>

<Button x:Name="btn_AddSB" Click="btn_AddSB_Click" Grid.Row="13"
Grid.Column="5" Content="ADD" Margin="10"/>
</Grid>
</ScrollViewer>
</Window>

```



## SAEform.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;

```

```
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

Page | 85 namespace testWPF
{
    /// <summary>
    /// Interaction logic for SAEform.xaml
    /// </summary>

    public partial class SAEform : Window
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public SAEform()
        {
            InitializeComponent();
            loadComboBoxProduct();
        }

        //FUNCTION TO LOAD PRODUCT COMBOBOX
        private void loadComboBoxProduct()
        {
            if (sqlCon.State == System.Data.ConnectionState.Closed)
                sqlCon.Open();
            string query = "SELECT Product_Code, Product_Name, SP_Unit, Stock_Quantity
FROM tblProduct";
            SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
            sqlcmd.CommandType = System.Data.CommandType.Text;
            DataTable dt = new DataTable();
            SqlDataAdapter adapter = new SqlDataAdapter(sqlcmd);

            adapter.Fill(dt);
            cb_SAEpn1.ItemsSource = dt.DefaultView;
            cb_SAEpn1.DisplayMemberPath = "Product_Name";
            cb_SAEpn1.SelectedValuePath = "Product_Code";
            cb_SAEpn2.ItemsSource = dt.DefaultView;
            cb_SAEpn2.DisplayMemberPath = "Product_Name";
            cb_SAEpn2.SelectedValuePath = "Product_Code";
            cb_SAEpn3.ItemsSource = dt.DefaultView;
```

Page |  
86

```
cb_SAEpn3.DisplayMemberPath = "Product_Name";
cb_SAEpn3.SelectedValuePath = "Product_Code";
cb_SAEpn4.ItemsSource = dt.DefaultView;
cb_SAEpn4.DisplayMemberPath = "Product_Name";
cb_SAEpn4.SelectedValuePath = "Product_Code";
cb_SAEpn5.ItemsSource = dt.DefaultView;
cb_SAEpn5.DisplayMemberPath = "Product_Name";
cb_SAEpn5.SelectedValuePath = "Product_Code";
sqlcmd.Dispose();

sqlCon.Close();
```

```
}
```

```
//FUNCTION TO CLEAR TEXTBOXES AND UNSELECT COMBOBOX
```

```
public void clearData()
{
    tb_SAEbill.Clear();
    tb_SAEcname.Clear();
    tb_SAEcno.Clear();
    tb_SAEdate.Clear();

    cb_SAEpn1.SelectedItem = null;
    cb_SAEpn2.SelectedItem = null;
    cb_SAEpn3.SelectedItem = null;
    cb_SAEpn4.SelectedItem = null;
    cb_SAEpn5.SelectedItem = null;

    tb_SAEpc1.Clear();
    tb_SAEpc2.Clear();
    tb_SAEpc3.Clear();
    tb_SAEpc4.Clear();
    tb_SAEpc5.Clear();

    tb_SAEsp1.Clear();
    tb_SAEsp2.Clear();
    tb_SAEsp3.Clear();
    tb_SAEsp4.Clear();
    tb_SAEsp5.Clear();

    tb_SAEq1.Clear();
    tb_SAEq2.Clear();
```

```
tb_SAEq3.Clear();
tb_SAEq4.Clear();
tb_SAEq5.Clear();
```

```
tb_SAEp1.Clear();
tb_SAEp2.Clear();
tb_SAEp3.Clear();
tb_SAEp4.Clear();
tb_SAEp5.Clear();
```

```
tb_SAEtp.Clear();
```

```
}
```

```
//FUNCTION TO CHECK ANY DATA MISSING BEFORE ADDING DATA TO DATABASE
```

```
public bool isValid()
{
    if (tb_SAEbill.Text == string.Empty)
    {
        MessageBox.Show("Bill No. is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return false;
    }
    if (tb_SAEcname.Text == string.Empty)
    {
        MessageBox.Show("Customer Name is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return false;
    }
    if (tb_SAEcno.Text == string.Empty)
    {
        MessageBox.Show("Customer Contact No. is required", "Failed",
MessageBoxButton.OK, MessageBoxIcon.Error);
        return false;
    }
    if (tb_SAEdate.Text == string.Empty)
    {
        MessageBox.Show("Date is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
        return false;
    }

    return true;
}
```

```
}

private void btn_AddSB_Click(object sender, RoutedEventArgs e)
{
    try
    {
        sqlCon.Open();

        if (isValid())
        {
            bool k = false;
            if (tb_SAEq1.Text != null && tb_SAEq1.Text != "")
            {
                SqlCommand cmd = new SqlCommand("INSERT INTO tblSaleBill VALUES (@Sale_Id, @Customer_Name, @Customer_Contact, @Product_Code ,@Quantity_Sold, @Sale_DOE)", sqlCon);
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@Sale_Id", tb_SAEbill.Text);
                cmd.Parameters.AddWithValue("@Customer_Name", tb_SAEcname.Text);
                cmd.Parameters.AddWithValue("@Customer_Contact", tb_SAEcno.Text);
                cmd.Parameters.AddWithValue("@Product_Code", tb_SAEpc1.Text);
                cmd.Parameters.AddWithValue("@Quantity_Sold",
Convert.ToInt32(tb_SAEq1.Text));
                cmd.Parameters.AddWithValue("@Sale_DOE",
Convert.ToDateTime(tb_SAEdate.Text));

                cmd.ExecuteNonQuery();

                int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn1.SelectedItem))["Stock_Quantity"]
.ToString());
                Stock_Quantity = Stock_Quantity - Convert.ToInt32(tb_SAEq1.Text);

                SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" +
cb_SAEpn1.SelectedValue + ";", sqlCon);
                cmd1.CommandType = CommandType.Text;

                cmd1.ExecuteNonQuery();

                k = true;
            }
        }
    }
}
```

```
Page | 89
      if(tb_SAEq2.Text != null && tb_SAEq2.Text != "") {
        SqlCommand cmd = new SqlCommand("INSERT INTO tblSaleBill VALUES
(@Sale_Id, @Customer_Name, @Customer_Contact, @Product_Code ,@Quantity_Sold,
@Sale_DOE)", sqlCon);
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@Sale_Id", tb_SAEbill.Text);
        cmd.Parameters.AddWithValue("@Customer_Name", tb_SAEcname.Text);
        cmd.Parameters.AddWithValue("@Customer_Contact", tb_SAEcno.Text);
        cmd.Parameters.AddWithValue("@Product_Code", tb_SAEpc2.Text);
        cmd.Parameters.AddWithValue("@Quantity_Sold",
Convert.ToInt32(tb_SAEq2.Text));
        cmd.Parameters.AddWithValue("@Sale_DOE",
Convert.ToDateTime(tb_SAEdate.Text));

        cmd.ExecuteNonQuery();

        int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn2.SelectedItem))["Stock_Quantity"]
].ToString());
        Stock_Quantity = Stock_Quantity - Convert.ToInt32(tb_SAEq2.Text);

        SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" +
cb_SAEpn2.SelectedValue + ";", sqlCon);
        cmd1.CommandType = CommandType.Text;

        cmd1.ExecuteNonQuery();

        k = true;
      }

      if (tb_SAEq3.Text != null && tb_SAEq3.Text != "") {
        SqlCommand cmd = new SqlCommand("INSERT INTO tblSaleBill VALUES
(@Sale_Id, @Customer_Name, @Customer_Contact, @Product_Code ,@Quantity_Sold,
@Sale_DOE)", sqlCon);
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@Sale_Id", tb_SAEbill.Text);
        cmd.Parameters.AddWithValue("@Customer_Name", tb_SAEcname.Text);
        cmd.Parameters.AddWithValue("@Customer_Contact", tb_SAEcno.Text);
```

```
cmd.Parameters.AddWithValue("@Product_Code", tb_SAEpc3.Text);
cmd.Parameters.AddWithValue("@Quantity_Sold",
Convert.ToInt32(tb_SAEq3.Text));
cmd.Parameters.AddWithValue("@Sale_DOE",
Convert.ToDateTime(tb_SAEdate.Text));
cmd.ExecuteNonQuery();

int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn3.SelectedItem))["Stock_Quantity"]
].ToString());
Stock_Quantity = Stock_Quantity - Convert.ToInt32(tb_SAEq3.Text);

SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" +
cb_SAEpn3.SelectedValue + ";", sqlCon);
cmd1.CommandType = CommandType.Text;

cmd1.ExecuteNonQuery();

k = true;
}

if (tb_SAEq4.Text != null && tb_SAEq4.Text != "")
{
    SqlCommand cmd = new SqlCommand("INSERT INTO tblSaleBill VALUES
(@Sale_Id, @Customer_Name, @Customer_Contact, @Product_Code ,@Quantity_Sold,
@Sale_DOE)", sqlCon);
    cmd.CommandType = CommandType.Text;
    cmd.Parameters.AddWithValue("@Sale_Id", tb_SAEbill.Text);
    cmd.Parameters.AddWithValue("@Customer_Name", tb_SAEcname.Text);
    cmd.Parameters.AddWithValue("@Customer_Contact", tb_SAEcno.Text);
    cmd.Parameters.AddWithValue("@Product_Code", tb_SAEpc4.Text);
    cmd.Parameters.AddWithValue("@Quantity_Sold",
Convert.ToInt32(tb_SAEq4.Text));
    cmd.Parameters.AddWithValue("@Sale_DOE",
Convert.ToDateTime(tb_SAEdate.Text));

cmd.ExecuteNonQuery();
```

```

    int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn4.SelectedItem))["Stock_Quantity"]
].ToString());
    Stock_Quantity = Stock_Quantity - Convert.ToInt32(tb_SAEq4.Text);

Page | 91
        SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code='" +
cb_SAEpn4.SelectedValue + "';", sqlCon);
    cmd1.CommandType = CommandType.Text;

        cmd1.ExecuteNonQuery();

        k = true;
    }

    if (tb_SAEq5.Text != null && tb_SAEq5.Text != "")
    {
        SqlCommand cmd = new SqlCommand("INSERT INTO tblSaleBill VALUES
(@Sale_Id, @Customer_Name, @Customer_Contact, @Product_Code ,@Quantity_Sold,
@Sale_DOE)", sqlCon);
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@Sale_Id", tb_SAEbill.Text);
        cmd.Parameters.AddWithValue("@Customer_Name", tb_SAEcname.Text);
        cmd.Parameters.AddWithValue("@Customer_Contact", tb_SAEcno.Text);
        cmd.Parameters.AddWithValue("@Product_Code", tb_SAEpc5.Text);
        cmd.Parameters.AddWithValue("@Quantity_Sold",
Convert.ToInt32(tb_SAEq5.Text));
        cmd.Parameters.AddWithValue("@Sale_DOE",
Convert.ToDateTime(tb_SAEdate.Text));

        cmd.ExecuteNonQuery();

        int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn5.SelectedItem))["Stock_Quantity"]
].ToString());
        Stock_Quantity = Stock_Quantity - Convert.ToInt32(tb_SAEq5.Text);

        SqlCommand cmd1 = new SqlCommand("UPDATE tblproduct SET
Stock_Quantity = " + Stock_Quantity + " WHERE product_code='" +
cb_SAEpn5.SelectedValue + "';", sqlCon);
        cmd1.CommandType = CommandType.Text;

```

```
        cmd1.ExecuteNonQuery();

        k = true;
    }

    if ( k == true)
    {
        MessageBox.Show("Sale Bill Entry is Successful.");
    }
    else
    {
        MessageBox.Show("Some Field is Missing. Entry Unsuccessful.");
    }

}

}

catch (SqlException ex)
{
    MessageBox.Show(ex.Message);
}

}

private void cb_SAEpn1_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {

        tb_SAEpc1.Text =
((System.Data.DataRowView)(cb_SAEpn1.SelectedItem))["Product_Code"].ToString();
        tb_SAEsp1.Text =
((System.Data.DataRowView)(cb_SAEpn1.SelectedItem))["SP_Unit"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }
}
```

```
        }

    private void cb_SAEpn2_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {

        tb_SAEpc2.Text =
((System.Data.DataRowView)(cb_SAEpn2.SelectedItem))["Product_Code"].ToString();
        tb_SAEsp2.Text =
((System.Data.DataRowView)(cb_SAEpn2.SelectedItem))["SP_Unit"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }
}

private void cb_SAEpn3_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {

        tb_SAEpc3.Text =
((System.Data.DataRowView)(cb_SAEpn3.SelectedItem))["Product_Code"].ToString();
        tb_SAEsp3.Text =
((System.Data.DataRowView)(cb_SAEpn3.SelectedItem))["SP_Unit"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }
}

private void cb_SAEpn4_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
```

```
{  
  
    tb_SAEpc4.Text =  
    ((System.Data.DataRowView)(cb_SAEpn4.SelectedItem))["Product_Code"].ToString();  
    tb_SAEsp4.Text =  
    ((System.Data.DataRowView)(cb_SAEpn4.SelectedItem))["SP_Unit"].ToString();  
  
}  
  
}  
  
}  
  
  


```
private void cb_SAEpn5_SelectionChanged(object sender, SelectionChangedEventArgs e)  
{  
    try  
  
        tb_SAEpc5.Text =  
        ((System.Data.DataRowView)(cb_SAEpn5.SelectedItem))["Product_Code"].ToString();  
        tb_SAEsp5.Text =  
        ((System.Data.DataRowView)(cb_SAEpn5.SelectedItem))["SP_Unit"].ToString();  
  
    }  
    catch (SqlException Ex)  
    {  
        MessageBox.Show(Ex.Message);  
    }  
  
}  
  
}  
  
  


```
private void tb_SAEq1_LostFocus(object sender, RoutedEventArgs e)  
{  
    int Stock_Quantity =  
    Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn1.SelectedItem))["Stock_Quantity"]  
    ].ToString());  
    if (Stock_Quantity == 0)
```


```


```

```
Page | 95
{
    MessageBox.Show("Stock Not Available. Please Buy Another Product.");
}
if (Stock_Quantity > 0)
{
    double Sale_price = Convert.ToDouble(tb_SAEsp1.Text);
    int Quantity = Convert.ToInt32(tb_SAEq1.Text);
    if (Quantity <= Stock_Quantity)
    {
        tb_SAEp1.Text = (Quantity * Sale_price).ToString();
        if (tb_SAEp1.Text != null)
            tb_SAEtp.Text = tb_SAEp1.Text;
    }
    else
    {
        MessageBox.Show("Available Quantity: " + Stock_Quantity + " Please Enter Less than that.");
    }
}

private void tb_SAEq2_LostFocus(object sender, RoutedEventArgs e)
{
    int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn2.SelectedItem))["Stock_Quantity"]
].ToString());
    if (Stock_Quantity == 0)
    {
        MessageBox.Show("Stock Not Available. Please Buy Another Product.");
    }
    if (Stock_Quantity > 0)
    {
        double Sale_price = Convert.ToDouble(tb_SAEsp2.Text);
        int Quantity = Convert.ToInt32(tb_SAEq1.Text);
        if (Quantity <= Stock_Quantity)
        {
            tb_SAEp2.Text = (Quantity * Sale_price).ToString();
            if (tb_SAEp2.Text != null && tb_SAEp1.Text != null)
            {
                double total1 = Convert.ToDouble(tb_SAEp1.Text);
                int total2 = (int)Convert.ToDouble(tb_SAEp2.Text);

                tb_SAEtp.Text = (total1 + total2).ToString();
            }
        }
    }
}
```

```
        }
    }
else
{
    MessageBox.Show("Available Quantity: " + Stock_Quantity + " Please Enter Less
Page | than that.");
}

}

private void tb_SAEq3_LostFocus(object sender, RoutedEventArgs e)
{
    int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn3.SelectedItem))["Stock_Quantity"]
].ToString());
    if (Stock_Quantity == 0)
    {
        MessageBox.Show("Stock Not Available. Please Buy Another Product.");
    }
    if (Stock_Quantity > 0)
    {
        double Sale_price = Convert.ToDouble(tb_SAEsp3.Text);
        int Quantity = Convert.ToInt32(tb_SAEq3.Text);
        if (Quantity <= Stock_Quantity)
        {
            tb_SAEp3.Text = (Quantity * Sale_price).ToString();
            if (tb_SAEp2.Text != null && tb_SAEp1.Text != null && tb_SAEp3.Text != null)
            {
                double total1 = Convert.ToDouble(tb_SAEp1.Text);
                double total2 = Convert.ToDouble(tb_SAEp2.Text);
                double total3 = Convert.ToDouble(tb_SAEp3.Text);

                tb_SAEtp.Text = (total1 + total2 + total3).ToString();
            }
        }
        else
        {
            MessageBox.Show("Available Quantity: " + Stock_Quantity + " Please Enter Less
than that.");
        }
    }
}
```

```
}
```

```
}
```

```
Page | 97
      private void tb_SAEq4_LostFocus(object sender, RoutedEventArgs e)
      {
          int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn4.SelectedItem))["Stock_Quantity"]
].ToString());
          if (Stock_Quantity == 0)
          {
              MessageBox.Show("Stock Not Available. Please Buy Another Product.");
          }
          if (Stock_Quantity > 0)
          {
              double Sale_price = Convert.ToDouble(tb_SAEsp4.Text);
              int Quantity = Convert.ToInt32(tb_SAEq4.Text);
              if (Quantity <= Stock_Quantity)
              {
                  tb_SAEp4.Text = (Quantity * Sale_price).ToString();
                  if (tb_SAEp2.Text != null && tb_SAEp1.Text != null && tb_SAEp3.Text != null &&
tb_SAEp4.Text != null)
                      tb_SAEtp.Text = (Convert.ToDouble(tb_SAEp1.Text) +
Convert.ToDouble(tb_SAEp2.Text) + Convert.ToDouble(tb_SAEp3.Text) +
Convert.ToDouble(tb_SAEp4.Text)).ToString();
                  }
                  else
                  {
                      MessageBox.Show("Available Quantity: " + Stock_Quantity + " Please Enter Less
than that.");
                  }
              }
          }
      }

      private void tb_SAEq5_LostFocus(object sender, RoutedEventArgs e)
      {
          int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_SAEpn5.SelectedItem))["Stock_Quantity"]
].ToString());
```

```
Page | 98
if (Stock_Quantity == 0)
{
    MessageBox.Show("Stock Not Available. Please Buy Another Product.");
}
if (Stock_Quantity > 0)
{
    double Sale_price = Convert.ToDouble(tb_SAEsp5.Text);
    int Quantity = Convert.ToInt32(tb_SAEq5.Text);
    if(Quantity <= Stock_Quantity)
    {
        tb_SAEp5.Text = (Quantity * Sale_price).ToString();
        if (tb_SAEp1.Text != null && tb_SAEp2.Text != null && tb_SAEp3.Text != null &&
tb_SAEp4.Text != null && tb_SAEp5.Text != null)
            tb_SAEtp.Text = (Convert.ToDouble(tb_SAEp1.Text) +
Convert.ToDouble(tb_SAEp2.Text)+ Convert.ToDouble(tb_SAEp3.Text) +
Convert.ToDouble(tb_SAEp4.Text) + Convert.ToDouble(tb_SAEp5.Text)).ToString();
        }
        else
        {
            MessageBox.Show("Available Quantity: " + Stock_Quantity + " Please Enter Less
than that.");
        }
    }
}
}
```

## ReturnSaleEntry.xaml

```
Page | 99
<Page x:Class="testWPF.ReturnSaleEntry"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="ReturnSaleEntry" Background="DarkSeaGreen" Foreground="White"
    FontSize="16">
    <StackPanel Orientation="Vertical">
        <TextBlock Text="RETURN SALE ENTRY" HorizontalAlignment="Center" FontSize="30"
        FontWeight="ExtraBold" Margin="0 25 0 0"/>
        <StackPanel Orientation="Horizontal" >
            <Button x:Name="btnRSEAdd" Click="btnRSEAdd_Click" Content="ADD"
            Width="100" Height="30" Margin="25 25 25 10"/>
            <TextBox x:Name="tb_RSEsearch" Width="200" Height="30" Margin="50 25 0 10"/>
            <Button x:Name="btnRSEView" Click="btnRSEView_Click" Content="VIEW"
            Height="30" Width="100" Margin="25 25 25 10"/>
        </StackPanel>
        <Separator Background="Black"></Separator>
        <DataGrid Height="242" x:Name="dg_RSEview" Margin="28,0,29,0"/>
    </StackPanel>
</Page>
```



## ReturnSaleEntry.xaml.cs

```
Page | 100
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
-using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for ReturnSaleEntry.xaml
    /// </summary>
    public partial class ReturnSaleEntry : Page
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public ReturnSaleEntry()
        {
            InitializeComponent();
        }

    /// <summary>
    /// OPEN RSEform WINDOW ON CLICKING ADD BUTTON
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void btnRSEAdd_Click(object sender, RoutedEventArgs e)
    {
```

```
RSEform rbef = new RSEform();
rbef.Show();
```

Page | 101 }

```
/// <summary>
/// /LOAD DATAGRID WITH DATA TABLE AS PER ReturnSale No
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void btnRSEView_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        string query = "SELECT ReturnSale_Id 'Return Sale No', RS_ID, Sale_Id 'Sale No',
Product_Code, Quantity_Return, Return_DOE FROM tbIRSBill WHERE ReturnSale_Id='"
+ tb_RSEsearch.Text + "'";
        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
        sqlcmd.CommandType = System.Data.CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataReader sdr = sqlcmd.ExecuteReader();

        dt.Load(sdr);
        dg_RSEview.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}
```

## RSEform.xaml

```
Page | 102
<Window x:Class="testWPF.RSEform"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    Title="RSEform" Height="550" Width="1100" Background="DarkSeaGreen"
    Foreground="White" FontSize="18">
    <Window.Resources>
        <Style TargetType="TextBox">
            <Setter Property="BorderBrush" Value="Black"/>
            <Setter Property="BorderThickness" Value="2"/>
            <Setter Property="Width" Value="200"/>
        </Style>
        <Style TargetType="Label">
            <Setter Property="BorderBrush" Value="Black"/>
            <Setter Property="BorderThickness" Value="2"/>
            <Setter Property="Background" Value="White"/>
            <Setter Property="Width" Value="200"/>
            <Setter Property="FontWeight" Value="SemiBold"/>
        </Style>
    </Window.Resources>
    <ScrollViewer HorizontalScrollBarVisibility="Visible" VerticalScrollBarVisibility="Visible">
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width ="20"/>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="20"/>
            </Grid.ColumnDefinitions>
            <Grid.RowDefinitions>
```

```
<RowDefinition Height="20"/>
<RowDefinition Height="Auto"/>
<RowDefinition Height="20"/>
</Grid.RowDefinitions>
<TextBlock Grid.Row="1" Grid.Column="3" Grid.ColumnSpan="3" Text="RETURN
SALE ENTRY" Margin="10" FontSize="20" FontWeight="ExtraBlack"/>

<TextBlock Grid.Row="2" Grid.Column="1" Text="RETURN SALE NO." Margin="0 5 0
5" FontWeight="SemiBold"/>
<TextBox Grid.Row="2" Grid.Column="2" x:Name="tb_RSEno" Margin="0 5 0 5"/>

<TextBlock Grid.Row="3" Grid.Column="1" Text="SALE NO." Margin="0 5 0 5"
FontWeight="SemiBold"/>
<TextBox Grid.Row="3" Grid.Column="2" x:Name="tb_RSEsaleno" Margin="0 5 0 5"
LostFocus="tb_RSEsaleno_LostFocus"/>
<Button Grid.Row="3" Grid.Column="3" Height="40" Width="175"
Name="btn_SearchSale" Content="Search" Click="btn_SearchSale_Click" />

<TextBlock Grid.Row="4" Grid.Column="1" Text="DATE" Margin="0 5 0 5"
FontWeight="SemiBold"/>
<TextBox Grid.Row="4" Grid.Column="2" x:Name="tb_RSEdate" Margin="0 5 0 10"/>

<Label Grid.Row="5" Grid.Column="1" Content="PRODUCT NAME"
FontWeight="SemiBold"/>
<Label Grid.Row="5" Grid.Column="2" Content="SP/UNIT"/>
<Label Grid.Row="5" Grid.Column="3" Content="QUANTITY SOLD"
FontWeight="SemiBold"/>
```

```
<Label Grid.Row="5" Grid.Column="4" Content="QUANTITY RETURNED"  
FontWeight="SemiBold"/>  
<Label Grid.Row="5" Grid.Column="5" Content="PRICE RETURNED"  
FontWeight="SemiBold"/>
```

Page |  
104

```
<ComboBox Grid.Row="5" Grid.Column="1" x:Name="cb_RSEpn1"  
SelectionChanged="cb_RSEpn1_SelectionChanged" Height="31" Margin="0,36,4,0,0"  
Grid.RowSpan="3" VerticalAlignment="Top" />  
<ComboBox Grid.Row="7" Grid.Column="1" x:Name="cb_RSEpn2"  
SelectionChanged="cb_RSEpn2_SelectionChanged" />  
<ComboBox Grid.Row="8" Grid.Column="1" x:Name="cb_RSEpn3"  
SelectionChanged="cb_RSEpn3_SelectionChanged" />  
<ComboBox Grid.Row="9" Grid.Column="1" x:Name="cb_RSEpn4"  
SelectionChanged="cb_RSEpn4_SelectionChanged" />  
<ComboBox Grid.Row="10" Grid.Column="1" x:Name="cb_RSEpn5"  
SelectionChanged="cb_RSEpn5_SelectionChanged" />  
  
<TextBox Grid.Row="6" Grid.Column="2" x:Name="tb_RSEsp1" />  
<TextBox Grid.Row="7" Grid.Column="2" x:Name="tb_RSEsp2" />  
<TextBox Grid.Row="8" Grid.Column="2" x:Name="tb_RSEsp3" />  
<TextBox Grid.Row="9" Grid.Column="2" x:Name="tb_RSEsp4" />  
<TextBox Grid.Row="10" Grid.Column="2" x:Name="tb_RSEsp5" />  
  
<TextBox Grid.Row="6" Grid.Column="3" x:Name="tb_RSEqs1" />  
<TextBox Grid.Row="7" Grid.Column="3" x:Name="tb_RSEqs2" />  
<TextBox Grid.Row="8" Grid.Column="3" x:Name="tb_RSEqs3" />  
<TextBox Grid.Row="9" Grid.Column="3" x:Name="tb_RSEqs4" />  
<TextBox Grid.Row="10" Grid.Column="3" x:Name="tb_RSEqs5" />  
  
<TextBox Grid.Row="6" Grid.Column="4" x:Name="tb_RSEq1"  
LostFocus="tb_RSEq1_LostFocus" />  
<TextBox Grid.Row="7" Grid.Column="4" x:Name="tb_RSEq2"  
LostFocus="tb_RSEq2_LostFocus" />  
<TextBox Grid.Row="8" Grid.Column="4" x:Name="tb_RSEq3"  
LostFocus="tb_RSEq3_LostFocus" />  
<TextBox Grid.Row="9" Grid.Column="4" x:Name="tb_RSEq4"  
LostFocus="tb_RSEq4_LostFocus" />  
<TextBox Grid.Row="10" Grid.Column="4" x:Name="tb_RSEq5"  
LostFocus="tb_RSEq5_LostFocus" />  
  
<TextBox Grid.Row="6" Grid.Column="5" x:Name="tb_RSEp1" />
```

Page | 105

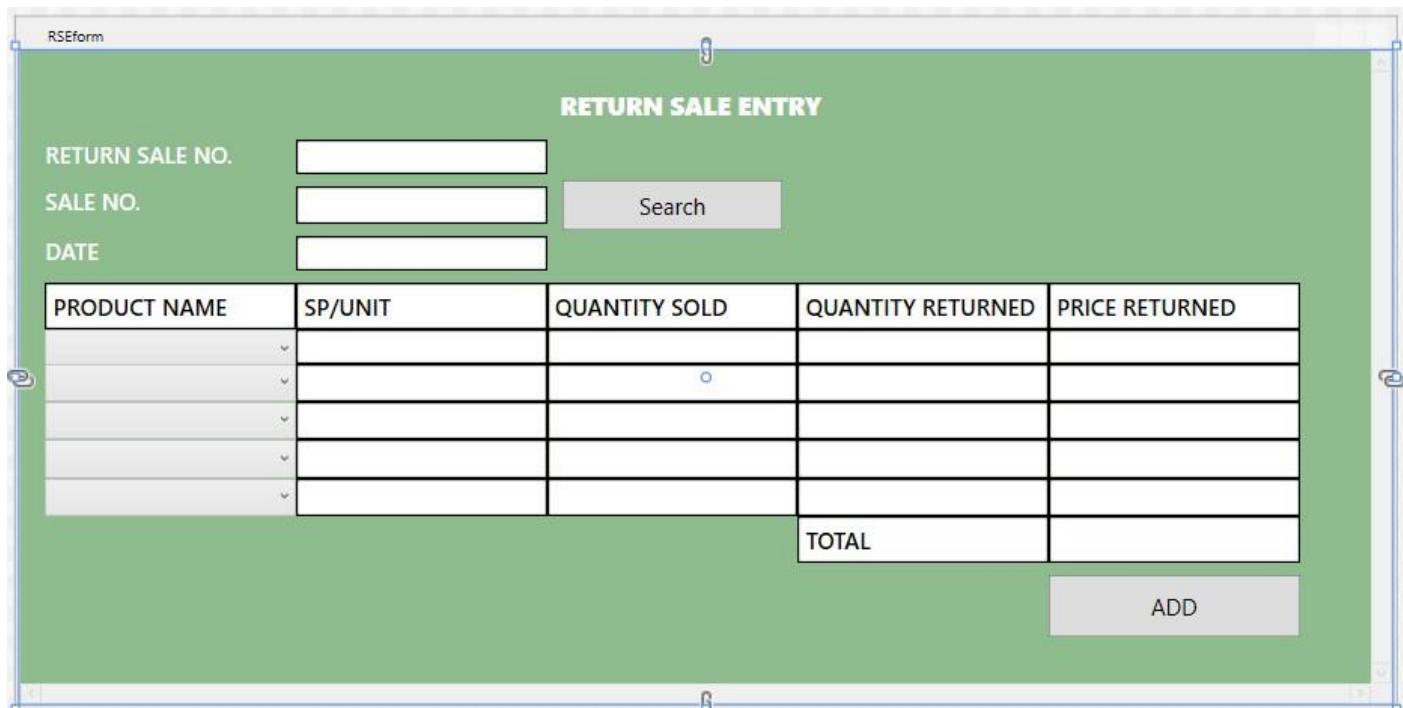
```

<TextBox Grid.Row="7" Grid.Column="5" x:Name="tb_RSEp2" />
<TextBox Grid.Row="8" Grid.Column="5" x:Name="tb_RSEp3" />
<TextBox Grid.Row="9" Grid.Column="5" x:Name="tb_RSEp4" />
<TextBox Grid.Row="10" Grid.Column="5" x:Name="tb_RSEp5" />

/>
<Label Grid.Row="11" Grid.Column="4" Content="TOTAL" FontWeight="SemiBold" />
<TextBox Grid.Row="11" Grid.Column="5" x:Name="tb_RSEtp" />

<Button Grid.Row="12" Grid.Column="5" Height="50" Width="200"
Name="btn_AddRSB" Click="btn_AddRSB_Click" Content="ADD" Margin="0 10 0 0"/>

</Grid>
</ScrollViewer>
</Window>
```



## RSEform.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
```

```
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for RSEform.xaml
    /// </summary>
    public partial class RSEform : Window
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public RSEform()
        {
            InitializeComponent();
        }

        /// <summary>
        ///FUNCTION CHECK ANY MISSING DATA AND SHOW MESSAGE OF MISSING FIELD
        /// </summary>
        /// <returns></returns>
        public bool isValid()
        {
            if (tb_RSEno.Text == string.Empty)
            {
                MessageBox.Show("Return Bill No. is required", "Failed", MessageBoxButton.OK,
MessageBoxImage.Error);
                return false;
            }
            if (tb_RSEsaleno.Text == string.Empty)
            {
                MessageBox.Show("Sale No. is required", "Failed", MessageBoxButton.OK,
MessageBoxImage.Error);
                return false;
            }
            if (tb_RSDate.Text == string.Empty)
```

```

        MessageBox.Show("Date is required", "Failed", MessageBoxButtons.OK,
MessageBoxImage.Error);
    return false;
}

return true;
}

/// <summary>
/// SEARCH SALE AS PER ENTERED Sale No
/// </summary>

public void searchSale()
{
    if (sqlCon.State == System.Data.ConnectionState.Closed)
        sqlCon.Open();
    string query = "SELECT SB_ID, Product_Code, Quantity_Sold FROM tblSaleBill WHERE
Sale_Id = tb_RSEsaleno.Text";
    SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
    sqlcmd.CommandType = System.Data.CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter adapter = new SqlDataAdapter(sqlcmd);

}

private void tb_RSEsaleno_LostFocus(object sender, RoutedEventArgs e)
{
    searchSale();
}

private void btn_SearchSale_Click(object sender, RoutedEventArgs e)
{
    if (sqlCon.State == System.Data.ConnectionState.Closed)
        sqlCon.Open();
    string query = "SELECT [SB_ID] ,[Sale_Id] ,[dbo].[tblSaleBill].[Product_Code]
,[Product_Name] ,[Quantity_Sold] ,[SP_Unit] ,[Stock_Quantity] FROM[dbo].[tblSaleBill]
inner join[dbo].[tblProduct]      on[dbo].[tblSaleBill].Product_Code
=[dbo].[tblProduct].Product_Code where[dbo].[tblSaleBill].Sale_Id = '" + tb_RSEsaleno.Text
+ "'";
    SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
    sqlcmd.CommandType = System.Data.CommandType.Text;
    DataTable dt = new DataTable();
}

```

```
SqlDataAdapter adapter = new SqlDataAdapter(sqlcmd);
adapter.Fill(dt);
cb_RSEpn1.ItemsSource = dt.DefaultView;
cb_RSEpn1.DisplayMemberPath = "Product_Name";
cb_RSEpn1.SelectedValuePath = "SB_ID";
cb_RSEpn2.ItemsSource = dt.DefaultView;
cb_RSEpn2.DisplayMemberPath = "Product_Name";
cb_RSEpn2.SelectedValuePath = "SB_ID";
cb_RSEpn3.ItemsSource = dt.DefaultView;
cb_RSEpn3.DisplayMemberPath = "Product_Name";
cb_RSEpn3.SelectedValuePath = "SB_ID";
cb_RSEpn4.ItemsSource = dt.DefaultView;
cb_RSEpn4.DisplayMemberPath = "Product_Name";
cb_RSEpn4.SelectedValuePath = "SB_ID";
cb_RSEpn5.ItemsSource = dt.DefaultView;
cb_RSEpn5.DisplayMemberPath = "Product_Name";
cb_RSEpn5.SelectedValuePath = "SB_ID";
```

}

```
private void cb_RSEpn1_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
    {

        tb_RSEsp1.Text =
((System.Data.DataRowView)(cb_RSEpn1.SelectedItem))["SP_Unit"].ToString();
        tb_RSEqs1.Text =
((System.Data.DataRowView)(cb_RSEpn1.SelectedItem))["Quantity_Sold"].ToString();

    }
    catch (SqlException Ex)
    {
        MessageBox.Show(Ex.Message);
    }
}

private void cb_RSEpn2_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
```

```
{  
  
    tb_RSEsp2.Text =  
    ((System.Data.DataRowView)(cb_RSEpn2.SelectedItem))["SP_Unit"].ToString();  
    tb_RSEqs2.Text =  
    ((System.Data.DataRowView)(cb_RSEpn2.SelectedItem))["Quantity_Sold"].ToString();  
  
}  
catch (SqlException Ex)  
{  
    MessageBox.Show(Ex.Message);  
}  
  
}  
  
private void cb_RSEpn3_SelectionChanged(object sender, SelectionChangedEventArgs  
e)  
{  
  
    try  
{  
  
        tb_RSEsp3.Text =  
        ((System.Data.DataRowView)(cb_RSEpn3.SelectedItem))["SP_Unit"].ToString();  
        tb_RSEqs3.Text =  
        ((System.Data.DataRowView)(cb_RSEpn3.SelectedItem))["Quantity_Sold"].ToString();  
  
    }  
catch (SqlException Ex)  
{  
    MessageBox.Show(Ex.Message);  
}  
  
}  
  
private void cb_RSEpn4_SelectionChanged(object sender, SelectionChangedEventArgs  
e)  
{  
  
    try  
{  
  
        tb_RSEsp4.Text =  
        ((System.Data.DataRowView)(cb_RSEpn4.SelectedItem))["SP_Unit"].ToString();  
    }  
}
```

Page | 110

```
        tb_RSEqs4.Text =
((System.Data.DataRowView)(cb_RSEpn4.SelectedItem))["Quantity_Sold"].ToString();

    }
    catch (SqlException Ex)
{
    MessageBox.Show(Ex.Message);
}

}

private void cb_RSEpn5_SelectionChanged(object sender, SelectionChangedEventArgs e)
{
    try
{

    tb_RSEsp5.Text =
((System.Data.DataRowView)(cb_RSEpn5.SelectedItem))["SP_Unit"].ToString();
    tb_RSEqs5.Text =
((System.Data.DataRowView)(cb_RSEpn5.SelectedItem))["Quantity_Sold"].ToString();

}
catch (SqlException Ex)
{
    MessageBox.Show(Ex.Message);
}

}

private void tb_RSEq1_LostFocus(object sender, RoutedEventArgs e)
{
    double Return_price = Convert.ToDouble(tb_RSEsp1.Text);
    int Quantity = Convert.ToInt32(tb_RSEq1.Text);
    if (Quantity <= (Convert.ToInt32(tb_RSEqs1.Text)))
    {
        tb_RSEp1.Text = (Quantity * Return_price).ToString();
        if (tb_RSEp1.Text != null)
            tb_RSEtp.Text = tb_RSEp1.Text;
    }
    else
    {
        MessageBox.Show("Entered Return Quantity is More than Sold Quantity.");
    }
}
```

```
        }

    }

private void tb_RSEq2_LostFocus(object sender, RoutedEventArgs e)
{
    double Return_price = Convert.ToDouble(tb_RSEsp2.Text);
    int Quantity = Convert.ToInt32(tb_RSEq2.Text);
    if (Quantity <= (Convert.ToInt32(tb_RSEqs2.Text)))
    {
        tb_RSEp2.Text = (Quantity * Return_price).ToString();
        if (tb_RSEp2.Text != null && tb_RSEp1.Text != null)
        {
            double total1 = Convert.ToDouble(tb_RSEp1.Text);
            int total2 = (int)Convert.ToDouble(tb_RSEp2.Text);

            tb_RSEtp.Text = (total1 + total2).ToString();
        }
    }
    else
    {
        MessageBox.Show("Entered Return Quantity is More than Sold Quantity.");
    }
}

private void tb_RSEq3_LostFocus(object sender, RoutedEventArgs e)
{
    double Return_price = Convert.ToDouble(tb_RSEsp3.Text);
    int Quantity = Convert.ToInt32(tb_RSEq3.Text);
    if (Quantity <= (Convert.ToInt32(tb_RSEqs3.Text)))
    {
        tb_RSEp3.Text = (Quantity * Return_price).ToString();
        if (tb_RSEp2.Text != null && tb_RSEp1.Text != null && tb_RSEp3.Text != null)
        {
            double total1 = Convert.ToDouble(tb_RSEp1.Text);
            double total2 = Convert.ToDouble(tb_RSEp2.Text);
            double total3 = Convert.ToDouble(tb_RSEp3.Text);

            tb_RSEtp.Text = (total1 + total2 + total3).ToString();
        }
    }
    else
```

```
{  
    MessageBox.Show("Entered Return Quantity is More than Sold Quantity.");  
}  
  
}  
  
private void tb_RSEq4_LostFocus(object sender, RoutedEventArgs e)  
{  
    double Return_price = Convert.ToDouble(tb_RSEsp4.Text);  
    int Quantity = Convert.ToInt32(tb_RSEq4.Text);  
    if (Quantity <= (Convert.ToInt32(tb_RSEqs4.Text)))  
    {  
        tb_RSEp4.Text = (Quantity * Return_price).ToString();  
        if (tb_RSEp2.Text != null && tb_RSEp1.Text != null && tb_RSEp3.Text != null &&  
        tb_RSEp4.Text != null)  
            tb_RSEtp.Text = (Convert.ToDouble(tb_RSEp1.Text) +  
Convert.ToDouble(tb_RSEp2.Text) + Convert.ToDouble(tb_RSEp3.Text) +  
Convert.ToDouble(tb_RSEp4.Text)).ToString();  
    }  
    else  
    {  
        MessageBox.Show("Entered Return Quantity is More than Sold Quantity.");  
    }  
  
}  
  
private void tb_RSEq5_LostFocus(object sender, RoutedEventArgs e)  
{  
    double Return_price = Convert.ToDouble(tb_RSEsp5.Text);  
    int Quantity = Convert.ToInt32(tb_RSEq5.Text);  
    if (Quantity <= (Convert.ToInt32(tb_RSEqs5.Text)))  
    {  
        tb_RSEp5.Text = (Quantity * Return_price).ToString();  
        if (tb_RSEp1.Text != null && tb_RSEp2.Text != null && tb_RSEp3.Text != null &&  
        tb_RSEp4.Text != null && tb_RSEp5.Text != null)  
            tb_RSEtp.Text = (Convert.ToDouble(tb_RSEp1.Text) +  
Convert.ToDouble(tb_RSEp2.Text)  
                + Convert.ToDouble(tb_RSEp3.Text) + Convert.ToDouble(tb_RSEp4.Text) +  
Convert.ToDouble(tb_RSEp5.Text)).ToString();  
    }  
    else  
    {
```

```
        MessageBox.Show("Entered Return Quantity is More than Sold Quantity.");
    }

}

Page | 113 /// <summary>
/// ADDING DATA OF RETURN SALE BILL TO tbIRSBill
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>

private void btn_AddRSB_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        if (isValid())
        {
            bool k = false;
            if (tb_RSEq1.Text != null && tb_RSEq1.Text != "")
            {
                SqlCommand cmd = new SqlCommand("INSERT INTO tbIRSBill VALUES (@ReturnSale_Id, @Sale_Id, @Product_Code ,@Quantity_Return, @Return_DOE",
                    sqlCon);
                cmd.CommandType = CommandType.Text;
                cmd.Parameters.AddWithValue("@ReturnSale_Id", tb_RSEno.Text);
                cmd.Parameters.AddWithValue("@Sale_Id", tb_RSEsaleno.Text);
                cmd.Parameters.AddWithValue("@Product_Code",
                    ((System.Data.DataRowView)(cb_RSEpn1.SelectedItem))["Product_Code"].ToString());
                cmd.Parameters.AddWithValue("@Quantity_Return",
                    Convert.ToInt32(tb_RSEq1.Text));
                cmd.Parameters.AddWithValue("@Return_DOE",
                    Convert.ToDateTime(tb_RSEdate.Text));

                cmd.ExecuteNonQuery();

                int Stock_Quantity =
                    Convert.ToInt32(((System.Data.DataRowView)(cb_RSEpn1.SelectedItem))["Stock_Quantity"]
                        .ToString());
                Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_RSEq1.Text);
            }
        }
    }
}
```

```

        string product_code=
((System.Data.DataRowView)(cb_RSEpn1.SelectedItem))["Product_Code"].ToString();

        SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code='" + product_code + "'",
sqlCon);
cmd1.CommandType = CommandType.Text;

        cmd1.ExecuteNonQuery();

        k = true;

    }

    if (tb_RSEq2.Text != null && tb_RSEq2.Text != "")
{
    SqlCommand cmd = new SqlCommand("INSERT INTO tbIRSBill VALUES
(@ReturnSale_Id, @Sale_Id, @Product_Code ,@Quantity_Return, @Return_DOE",
sqlCon);
cmd.CommandType = CommandType.Text;
cmd.Parameters.AddWithValue("@ReturnSale_Id", tb_RSEno.Text);
cmd.Parameters.AddWithValue("@Sale_Id", tb_RSEsaleno.Text);

    cmd.Parameters.AddWithValue("@Product_Code",
((System.Data.DataRowView)(cb_RSEpn2.SelectedItem))["Product_Code"].ToString());
    cmd.Parameters.AddWithValue("@Quantity_Return",
Convert.ToInt32(tb_RSEq2.Text));
    cmd.Parameters.AddWithValue("@Return_DOE",
Convert.ToDateTime(tb_RSEdate.Text));

    cmd.ExecuteNonQuery();

    int Stock_Quantity =
Convert.ToInt32((System.Data.DataRowView)(cb_RSEpn2.SelectedItem)["Stock_Quantity"]
.ToString());
    Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_RSEq2.Text);

    SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code='" +
cb_RSEpn2.SelectedValue + "'", sqlCon);
cmd1.CommandType = CommandType.Text;

```

Page | 115

```

        cmd1.ExecuteNonQuery();

        k = true;
    }

    if (tb_RSEq3.Text != null && tb_RSEq3.Text != "")
    {
        SqlCommand cmd = new SqlCommand("INSERT INTO tblRSBill VALUES
(@ReturnSale_Id, @Sale_Id, @Product_Code ,@Quantity_Return, @Return_DOE",
sqlCon);
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@ReturnSale_Id", tb_RSEno.Text);
        cmd.Parameters.AddWithValue("@Sale_Id", tb_RSEsaleno.Text);

        cmd.Parameters.AddWithValue("@Product_Code",
((System.Data.DataRowView)(cb_RSEpn3.SelectedItem))["Product_Code"].ToString());
        cmd.Parameters.AddWithValue("@Quantity_Return",
Convert.ToInt32(tb_RSEq3.Text));
        cmd.Parameters.AddWithValue("@Return_DOE",
Convert.ToDateTime(tb_RSEdate.Text));

        cmd.ExecuteNonQuery();

        int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_RSEpn3.SelectedItem))["Stock_Quantity"]
].ToString());
        Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_RSEq3.Text);

        SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code=" + cb_RSEpn3.SelectedValue + ";", sqlCon);
        cmd1.CommandType = CommandType.Text;

        cmd1.ExecuteNonQuery();

        k = true;
    }

    if (tb_RSEq4.Text != null && tb_RSEq4.Text != "")
    {

```

Page | 116

```
        SqlCommand cmd = new SqlCommand("INSERT INTO tblRSBill VALUES  
(@ReturnSale_Id, @Sale_Id, @Product_Code ,@Quantity_Return, @Return_DOE)",  
sqlCon);  
        cmd.CommandType = CommandType.Text;  
        cmd.Parameters.AddWithValue("@ReturnSale_Id", tb_RSEno.Text);  
        cmd.Parameters.AddWithValue("@Sale_Id", tb_RSEsaleno.Text);  
  
        cmd.Parameters.AddWithValue("@Product_Code",  
((System.Data.DataRowView)(cb_RSEpn4.SelectedItem))["Product_Code"].ToString());  
        cmd.Parameters.AddWithValue("@Quantity_Return",  
Convert.ToInt32(tb_RSEq4.Text));  
        cmd.Parameters.AddWithValue("@Return_DOE",  
Convert.ToDateTime(tb_RSEdate.Text));  
  
        cmd.ExecuteNonQuery();  
  
        int Stock_Quantity =  
Convert.ToInt32(((System.Data.DataRowView)(cb_RSEpn4.SelectedItem))["Stock_Quantity"]  
].ToString());  
        Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_RSEq4.Text);  
  
        SqlCommand cmd1 = new SqlCommand("update tblproduct set  
Stock_Quantity = " + Stock_Quantity + " where product_code=" +  
cb_RSEpn4.SelectedValue + ";", sqlCon);  
        cmd1.CommandType = CommandType.Text;  
  
        cmd1.ExecuteNonQuery();  
  
        k = true;  
    }  
  
    if (tb_RSEq5.Text != null && tb_RSEq5.Text != "")  
    {  
        SqlCommand cmd = new SqlCommand("INSERT INTO tblRSBill VALUES  
(@ReturnSale_Id, @Sale_Id, @Product_Code ,@Quantity_Return, @Return_DOE)",  
sqlCon);  
        cmd.CommandType = CommandType.Text;  
        cmd.Parameters.AddWithValue("@ReturnSale_Id", tb_RSEno.Text);  
        cmd.Parameters.AddWithValue("@Sale_Id", tb_RSEsaleno.Text);  
  
        cmd.Parameters.AddWithValue("@Product_Code",  
((System.Data.DataRowView)(cb_RSEpn5.SelectedItem))["Product_Code"].ToString());
```

```
        cmd.Parameters.AddWithValue("@Quantity_Return",
Convert.ToInt32(tb_RSEq5.Text));
        cmd.Parameters.AddWithValue("@Return_DOE",
Convert.ToDateTime(tb_RSEdate.Text));

Page | 117
      cmd.ExecuteNonQuery();

      int Stock_Quantity =
Convert.ToInt32(((System.Data.DataRowView)(cb_RSEpn5.SelectedItem))["Stock_Quantity"]
].ToString());
      Stock_Quantity = Stock_Quantity + Convert.ToInt32(tb_RSEq5.Text);

      SqlCommand cmd1 = new SqlCommand("update tblproduct set
Stock_Quantity = " + Stock_Quantity + " where product_code='"
+ cb_RSEpn5.SelectedValue + "';", sqlCon);
      cmd1.CommandType = CommandType.Text;

      cmd1.ExecuteNonQuery();

      k = true;

    }

    if (k == true)
    {
      MessageBox.Show("Return Bill Entry Successful.");
    }
    else
    {
      MessageBox.Show("Some Field is Missing. Entry Unsuccessful.");
    }

  }

}

catch (SqlException ex)
{
  MessageBox.Show(ex.Message);

}
```

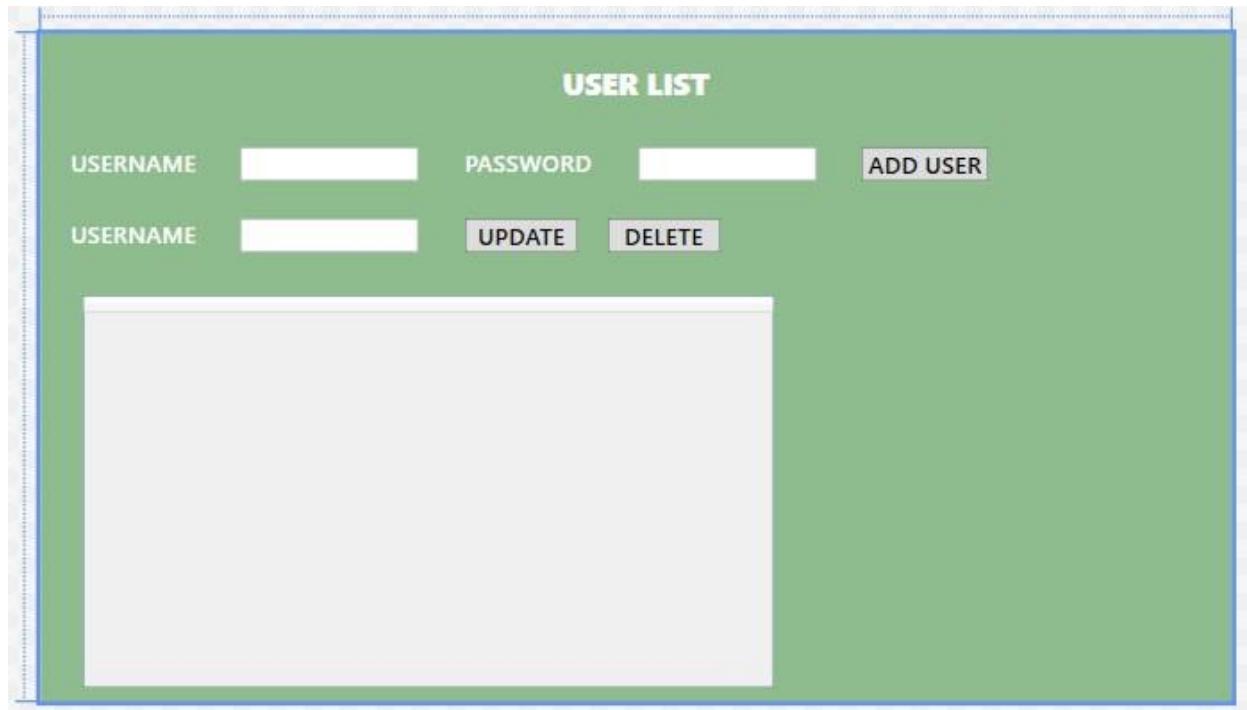
## UserList.xaml

```
<Page x:Class="testWPF.UserList"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:testWPF"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"
    Title="UserList" Background="DarkSeaGreen" Foreground="White" FontSize="16">

    <Grid>
        <StackPanel Orientation="Vertical">
            <TextBlock Text="USER LIST" Margin="350 20 330 20" FontSize="20"
FontWeight="ExtraBold"/>
            <StackPanel Orientation="Horizontal">
                <TextBlock Text="USERNAME" VerticalAlignment="Top" Margin="20 10 10 10"
FontWeight="SemiBold"/>
                <TextBox Name="txtNewUserName" Height="23" VerticalAlignment="Top"
Width="120" Margin="20 10 10 10"/>
                <TextBlock Text="PASSWORD" VerticalAlignment="Top" Margin="20 10 10 10"
FontWeight="SemiBold"/>
                <TextBox Name="PwdUser" Height="23" VerticalAlignment="Top" Width="120"
Margin="20 10 10 10"/>
                <Button Content="ADD USER" VerticalAlignment="Top" Width="85" Height="23"
Name="btnUserSubmit" Click="btnUserSubmit_Click" Margin="20 10 10 10"
FontWeight="SemiBold"/>
            </StackPanel>
            <Separator HorizontalAlignment="Center" Background="Black"/>
            <StackPanel Orientation="Horizontal">
                <TextBlock Text="USERNAME" VerticalAlignment="Top" Margin="20 10 10 10"
FontWeight="SemiBold"/>
                <TextBox x:Name="textbox_UsrSearch" VerticalAlignment="Top" Width="120"
Height="23" Margin="20 10 10 10"/>
            </StackPanel>
        </StackPanel>
    </Grid>

```

```
Page | 119
      <Button x:Name="btn_UsrUpdate" Click="btn_UsrUpdate_Click"
Content="UPDATE" VerticalAlignment="Top" Width="75" Height="23" Margin="20 10 10
10" FontWeight="SemiBold"/>
      <Button x:Name="btn_UsrDel" Click="btn_UsrDel_Click" Content="DELETE"
VerticalAlignment="Top" Width="75" Height="23" Margin="10" FontWeight="SemiBold"/>
    </StackPanel>
</StackPanel>
<Separator HorizontalAlignment="Center" Background="Black"/>
<DataGrid HorizontalAlignment="Left" Height="264" Margin="28,176,0,0"
VerticalAlignment="Top" Width="465" Name="DataGrid_User"/>
</Grid>
</Page>
```



## UserList.xaml.cs

```
Page | 120
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for UserList.xaml
    /// </summary>
    public partial class UserList : Page
    {
        public UserList()
        {
            InitializeComponent();
            LoadUserGrid();
        }

        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");

        public void clearData()
        {
            txtNewUserName.Clear();
            PwdUser.Clear();
```

```
        textBox_UsrSearch.Clear();
    }

    //FOR SHOWING TABLE(tblUser) DATA ON DATAGRID(DataGrid_User) ON
    PAGE(UserList)
Page | 121
    private void LoadUserGrid()
    {
        try
        {
            if (sqlCon.State == System.Data.ConnectionState.Closed)
                sqlCon.Open();

            string query = "SELECT * FROM tblUser";
            SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
            sqlcmd.CommandType = System.Data.CommandType.Text;
            DataTable dt = new DataTable();
            SqlDataReader sdr = sqlcmd.ExecuteReader();
            dt.Load(sdr);
            DataGrid_User.ItemsSource = dt.DefaultView;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        finally
        {
            sqlCon.Close();
        }
    }

    //CHECK FOR USERNAME AND PASSWORD TEXTBOX TO BE FILLED BEFORE ADDING
    NEW USER
    public bool isValid()
    {
        if(txtNewUserName.Text == string.Empty)
        {
            MessageBox.Show("UserName is Required", "Failed");
            return false;
        }
        if (PwdUser.Text == string.Empty)
        {
            MessageBox.Show("Password is Required", "Failed");
            return false;
        }
    }
```

```
        return true;
    }

//FUNCTIONALITY FOR BUTTON(ADD USER)
private void btnUserSubmit_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (isValid())
        {
            if (sqlCon.State == System.Data.ConnectionState.Closed)
                sqlCon.Open();
            string query = "SELECT count(1) FROM tblUser where Username=@NewUserName";
            SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
            sqlcmd.CommandType = System.Data.CommandType.Text;
            sqlcmd.Parameters.AddWithValue("@NewUserName", txtNewUserName.Text);

            int count = (int)sqlcmd.ExecuteScalar();
            if (count == 0)
            {
                string query1 = "INSERT INTO tblUser ( Username, password )
VALUES(@NewUserName,@password); ";
                SqlCommand sqlcmd1 = new SqlCommand(query1, sqlCon);
                sqlcmd1.CommandType = System.Data.CommandType.Text;
                sqlcmd1.Parameters.AddWithValue("@NewUserName",
txtNewUserName.Text);
                sqlcmd1.Parameters.AddWithValue("@Password", PwdUser.Text);
                DataTable dt = new DataTable();
                sqlcmd1.ExecuteNonQuery();
                LoadUserGrid();
                MessageBox.Show("New Username is added.");
            }
            else
            {
                MessageBox.Show("Username is already there.");
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
```

```
        {
            sqlCon.Close();
        }
    }
```

Page | 123

```
/// <summary>
/// FUNCTIONALITY FOR BUTTON(UPDATE)
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void btn_UsrUpdate_Click(object sender, RoutedEventArgs e)
{
    sqlCon.Open();

    SqlCommand cmd = new SqlCommand("UPDATE tblUser set
Username='"+txtNewUserName.Text+"', password ='"+PwdUser.Text+"' WHERE
Username='"+textbox_UsrSearch.Text+"'", sqlCon );
    try
    {
        cmd.ExecuteNonQuery();
        MessageBox.Show("Record has been updated successfully");
    }
    catch(SqlException ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
        clearData();
        LoadUserGrid();
    }
}

///FUNCTIONALITY FOR BUTTON(DELETE)
private void btn_UsrDel_Click(object sender, RoutedEventArgs e)
{
    sqlCon.Open();
    SqlCommand cmd = new SqlCommand("DELETE FROM tblUser WHERE Username='"
+ textbox_UsrSearch.Text + "'", sqlCon);
    try
    {
        cmd.ExecuteNonQuery();
```

```

    MessageBox.Show("Record has been deleted successfully");
    clearData();
    LoadUserGrid();
}
catch (SqlException ex)
{
    MessageBox.Show("Not Deleted"+ex.Message);
}
finally
{
    sqlCon.Close();
}
}
}

}

```

## Category.xaml

```

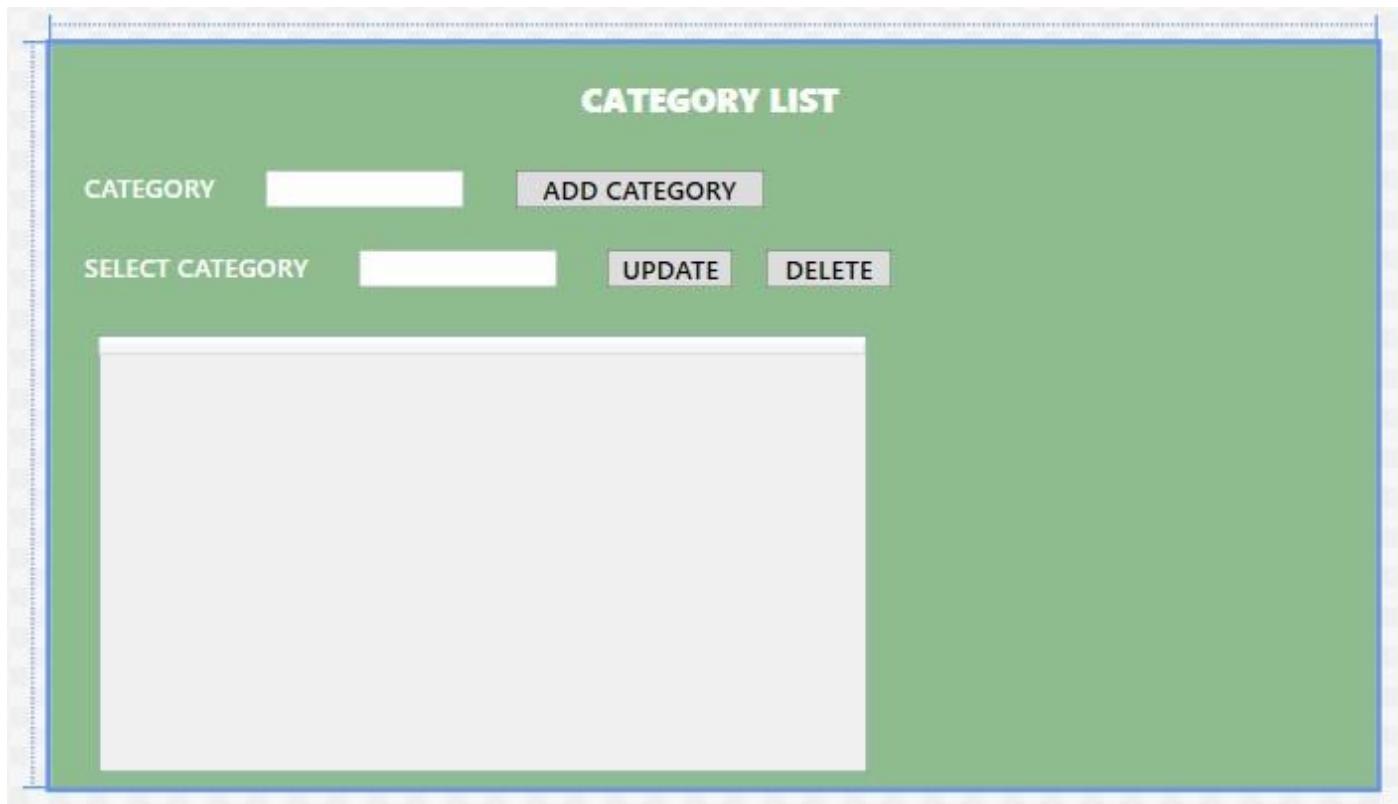
<Page x:Class="testWPF.Category"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:testWPF"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="Category" Background="DarkSeaGreen" Foreground="White" FontSize="16">
<Grid>
    <StackPanel Orientation="Vertical">
        <TextBlock Text="CATEGORY LIST" Margin="350 20 330 20" FontSize="20"
FontWeight="ExtraBold"/>
        <StackPanel Orientation="Horizontal">
            <TextBlock Text="CATEGORY" VerticalAlignment="Top" Margin="20 10 10 10"
FontWeight="SemiBold"/>
            <TextBox Name="tb_Category" Height="23" VerticalAlignment="Top"
Width="120" Margin="20 10 10 10"/>
            <Button Content="ADD CATEGORY" VerticalAlignment="Top" Width="150"
Height="23" Name="btn_AddCat" Click="btn_AddCat_Click" Margin="20 10 10 10"
FontWeight="SemiBold"/>
        </StackPanel>
    
```

```

<Separator HorizontalAlignment="Center" Background="Black"/>
<StackPanel Orientation="Horizontal">
    <TextBlock Text="SELECT CATEGORY" VerticalAlignment="Top" Margin="20 10 10
10" FontWeight="SemiBold"/>
    <TextBox x:Name="textbox_CSelect" VerticalAlignment="Top" Width="120"
Height="23" Margin="20 10 10 10"/>
    <Button x:Name="btn_CUp" Click="btn_CUp_Click" Content="UPDATE"
VerticalAlignment="Top" Width="75" Height="23" Margin="20 10 10 10"
FontWeight="SemiBold"/>
    <Button x:Name="btn_CDelete" Click="btn_CDelete_Click" Content="DELETE"
VerticalAlignment="Top" Width="75" Height="23" Margin="10" FontWeight="SemiBold"/>
</StackPanel>
</StackPanel>
<Separator HorizontalAlignment="Center" Background="Black"/>
<DataGrid HorizontalAlignment="Left" Height="264" Margin="28,176,0,0"
VerticalAlignment="Top" Width="465" Name="DataGrid_Category"/>
</Grid>

</Page>

```



## Category.xaml.cs

```
Page | 126
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for Category.xaml
    /// </summary>
    public partial class Category : Page
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public Category()
        {
            InitializeComponent();
            LoadCategoryGrid();
            clearData();
        }

        public void clearData()
        {
            tb_Category.Clear();
            textbox_CSelect.Clear();
        }
    }
}
```

```
}

private void LoadCategoryGrid()
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        string query = "SELECT * FROM tblCategory";
        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
        sqlcmd.CommandType = System.Data.CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataReader sdr = sqlcmd.ExecuteReader();
        dt.Load(sdr);
        DataGrid_Category.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}

//CHECK FOR CATEGORY TEXTBOX TO BE FILLED BEFORE ADDING NEW CATEGORY
public bool isValid()
{
    if (tb_Category.Text == string.Empty)
    {
        MessageBox.Show("Category Name is Required", "Failed");
        return false;
    }
    return true;
}

private void btn_AddCat_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (isValid())
        {
            if (sqlCon.State == System.Data.ConnectionState.Closed)
```

```
    sqlCon.Open();
    string query = "SELECT count(1) FROM tblCategory where Category_Name=@CategoryName;";
    SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
    sqlcmd.CommandType = System.Data.CommandType.Text;
    sqlcmd.Parameters.AddWithValue("@CategoryName", tb_Category.Text);

    int count = (int)sqlcmd.ExecuteScalar();
    if (count == 0)
    {
        string query1 = "INSERT INTO tblCategory ( Category_Name ) VALUES (@CategoryName); ";
        SqlCommand sqlcmd1 = new SqlCommand(query1, sqlCon);
        sqlcmd1.CommandType = System.Data.CommandType.Text;
        sqlcmd1.Parameters.AddWithValue("@CategoryName", tb_Category.Text);

        DataTable dt = new DataTable();
        sqlcmd1.ExecuteNonQuery();
        LoadCategoryGrid();
        MessageBox.Show("New Category is added.");
    }

    else
    {
        MessageBox.Show("Category Name is already there.");
    }
}

catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    sqlCon.Close();
}

}

private void btn_CDelete_Click(object sender, RoutedEventArgs e)
{
    sqlCon.Open();
```

```
Page | 129
      SqlCommand cmd = new SqlCommand("UPDATE tblCategory set Category_Name ='" + tb_Category.Text + "' WHERE Category_Name ='" + textbox_CSelect.Text + "'", sqlCon);
      try
      {
          cmd.ExecuteNonQuery();
          MessageBox.Show("Record has been updated successfully");
      }
      catch (SqlException ex)
      {
          MessageBox.Show(ex.Message);
      }
      finally
      {
          sqlCon.Close();
          clearData();
      }
  }

private void btn_CUp_Click(object sender, RoutedEventArgs e)
{
    sqlCon.Open();
    SqlCommand cmd = new SqlCommand("DELETE FROM tblCategory WHERE Category_Name ='" + textbox_CSelect.Text + "'", sqlCon);
    try
    {
        cmd.ExecuteNonQuery();
        MessageBox.Show("Record has been deleted successfully");
        clearData();
        LoadCategoryGrid();
    }
    catch (SqlException ex)
    {
        MessageBox.Show("Not Deleted" + ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}
}
```

## Inventory.xaml

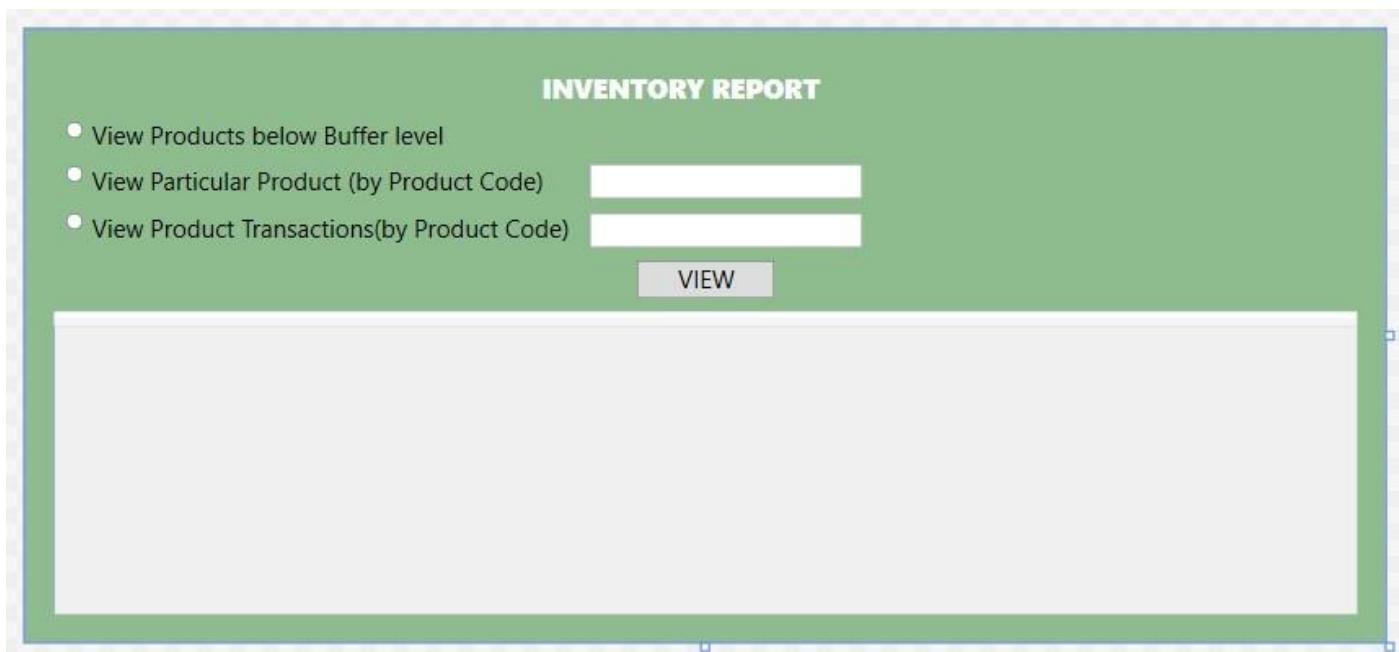
```
Page | 130
<Page x:Class="testWPF.Inventory"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:testWPF"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="1000"
      Title="Inventory" Background="DarkSeaGreen" Foreground="White" FontSize="18">
<Grid>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="20"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="20"/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="20"/>
        <RowDefinition Height="409.6"/>
        <RowDefinition Height="20"/>
    </Grid.RowDefinitions>
    <StackPanel Grid.Row="1" Grid.Column="1">
        <StackPanel Margin="350 10 380 10">
            <TextBlock Text="INVENTORY REPORT" Margin="10 0 0 0" FontSize="20"
FontWeight="Black"/>
        </StackPanel>
        <StackPanel Margin="10 0 0 10">
            <RadioButton Content="View Products below Buffer level" x:Name="rb_1"
GroupName="Iview" Checked="rb_1_Checked" />
            </StackPanel>
            <StackPanel Orientation="Horizontal" Margin="10 0 0 10">
                <RadioButton Content="View Particular Product (by Product Code)"
x:Name="rb_2" GroupName="Iview" Checked="rb_2_Checked"/>
                <TextBox x:Name="tb1_Psearch" Width="200" Margin="34 0 0 0"/>
            </StackPanel>
            <StackPanel Orientation="Horizontal" Margin="10 0 0 10">
```

Page | 131

```

<RadioButton Content="View Product Transactions(by Product Code)"
x:Name="rb_3" GroupName="Iview" Checked="rb_3_Checked" />
    <TextBox x:Name="tb2_Psearch" Width="200" Margin="15 0 0 0"/>
</StackPanel>
<Button x:Name="btn_View" Click="btn_View_Click" Content="VIEW" Width="100"
Margin="0 0 0 10"/>
    <DataGrid x:Name="dg_Iview" HorizontalAlignment="Left" Height="300"
Width="1200" />
</StackPanel>
</Grid>

</Page>
```



## Inventory.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
```

```
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

Page | 132
namespace testWPF
{
    /// <summary>
    /// Interaction logic for Inventory.xaml
    /// </summary>

    public partial class Inventory : Page
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public Inventory()
        {
            InitializeComponent();
        }

        private void btn_View_Click(object sender, RoutedEventArgs e)
        {
            if (rb_1.IsChecked == true)
            {
                try
                {
                    if (sqlCon.State == System.Data.ConnectionState.Closed)
                        sqlCon.Open();

                    string query = "SELECT [Product_Code],[Product_Name],
[IMDB].[dbo].[tblVendor].[VENDOR_NAME], [Buffer_Quantity],[Stock_Quantity] FROM
[IMDB].[dbo].[tblProduct] INNER JOIN[IMDB].[dbo].[tblVendor]
ON[IMDB].[dbo].[tblProduct].Vendor = [IMDB].[dbo].[tblVendor].VENDOR_CODE
WHERE[IMDB].[dbo].[tblProduct].Stock_Quantity <
[IMDB].[dbo].[tblProduct].Buffer_Quantity";
                    SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
                    sqlcmd.CommandType = System.Data.CommandType.Text;
                    DataTable dt = new DataTable();
                    SqlDataReader sdr = sqlcmd.ExecuteReader();

                    dt.Load(sdr);
                }
            }
        }
    }
}
```

```
        dg_Iview.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}

if (rb_2.IsChecked == true)
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        string query = "SELECT [Product_Code],[Product_Name], [Description],
[Category], [IMDB].[dbo].[tblVendor].[VENDOR_NAME], [Location], [SP_Unit],
[Stock_Quantity] FROM [IMDB].[dbo].[tblProduct] INNER JOIN[IMDB].[dbo].[tblVendor]
ON[IMDB].[dbo].[tblProduct].Vendor = [IMDB].[dbo].[tblVendor].VENDOR_CODE
WHERE[IMDB].[dbo].[tblProduct].Product_Code = '"+tb1_Psearch.Text+"'";
        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
        sqlcmd.CommandType = System.Data.CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataReader sdr = sqlcmd.ExecuteReader();

        dt.Load(sdr);
        dg_Iview.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}
```

```
if (rb_3.IsChecked == true)
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        string query = "SELECT 'Purchase' AS TYPE, [Invoice_Id] 'Number',
[Product_Code], [Quantity_Received] 'Quantity', [Purchase_DOE] 'Date' FROM
[IMDB].[dbo].[tblPurchaseBill] WHERE [Product_Code] = '" + tb2_Psearch.Text + "' UNION
SELECT 'Sale' AS TYPE, [Sale_Id] 'Number', [Product_Code], [Quantity_Sold] 'Quantity',
[Sale_DOE] 'Date' FROM [IMDB].[dbo].[tblSaleBill] WHERE [Product_Code] = '" +
tb2_Psearch.Text + "' UNION SELECT 'Return' AS TYPE, [ReturnSale_Id] 'Number',
[Product_Code], [Quantity_Return] 'Quantity', [Return_DOE] 'Date' FROM
[IMDB].[dbo].[tblRSBill] WHERE [Product_Code] = '" + tb2_Psearch.Text + "' ORDER BY Date;
    ";

        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
        sqlcmd.CommandType = System.Data.CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataReader sdr = sqlcmd.ExecuteReader();

        dt.Load(sdr);
        dg_lview.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}

private void rb_1_Checked(object sender, RoutedEventArgs e)
{
```

```
rb_1.Background = Brushes.Yellow;
rb_1.Foreground = Brushes.Navy;
rb_2.Background = Brushes.White;
rb_2.Foreground = Brushes.Black;
rb_3.Background = Brushes.White;
rb_3.Foreground = Brushes.Black;
tb1_Psearch.Clear();
tb2_Psearch.Clear();
}

private void rb_2_Checked(object sender, RoutedEventArgs e)
{
    rb_1.Background = Brushes.White;
    rb_1.Foreground = Brushes.Black;
    rb_2.Background = Brushes.Yellow;
    rb_2.Foreground = Brushes.Navy;
    rb_3.Background = Brushes.White;
    rb_3.Foreground = Brushes.Black;
    tb2_Psearch.Clear();
}

private void rb_3_Checked(object sender, RoutedEventArgs e)
{
    rb_1.Background = Brushes.White;
    rb_1.Foreground = Brushes.Black;
    rb_2.Background = Brushes.White;
    rb_2.Foreground = Brushes.Black;
    rb_3.Background = Brushes.Yellow;
    rb_3.Foreground = Brushes.Navy;
    tb1_Psearch.Clear();
}

}
```

## Report.xaml

```
Page | 136
<Page x:Class="testWPF.Report"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:local="clr-namespace:testWPF"
      mc:Ignorable="d"
      d:DesignHeight="450" d:DesignWidth="800"
      Title="Report" Background="DarkSeaGreen" Foreground="White" FontSize="18">
<Grid>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="20"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="20"/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="20"/>
        <RowDefinition Height="409.6"/>
        <RowDefinition Height="20"/>
    </Grid.RowDefinitions>
    <StackPanel Grid.Row="1" Grid.Column="1">
        <StackPanel Margin="250 10 250 10">
            <TextBlock Text="TRANSACTION REPORT" Margin="10 0 0 0" FontSize="20"
FontWeight="Black"/>
        </StackPanel>
        <StackPanel Orientation="Vertical" Margin="10 0 0 10">
            <RadioButton Content="View Purchase Transcation" x:Name="rb_r1"
GroupName="lview" Checked="rb_r1_Checked" />
            <StackPanel Orientation="Horizontal">
                <TextBlock Text="START(YYYY-MM-DD)" Width="180" Margin="0 0 10 0"/>
                <TextBox x:Name="tb_Sdate1" Width="150" Margin="0 0 10 0" />
                <TextBlock Text="END(YYYY-MM-DD)" Margin="10 0 10 0"/>
                <TextBox x:Name="tb_Edate1" Width="150"/>
            </StackPanel>
        </StackPanel>
    </StackPanel>

    <StackPanel Orientation="Vertical" Margin="10 0 0 10">
        <RadioButton Content="View Sale Transaction" x:Name="rb_r2"
GroupName="lview" Checked="rb_r2_Checked"/>
        <StackPanel Orientation="Horizontal">
```

```
<TextBlock Text="START(YYYY-MM-DD)" Width="180" Margin="0 0 10 0"/>
<TextBox x:Name="tb_Sdate2" Width="150" Margin="0 0 10 0" />
<TextBlock Text="END(YYYY-MM-DD)" Margin="10 0 10 0"/>
<TextBox x:Name="tb_Edate2" Width="150"/>
</StackPanel>
</StackPanel>

<StackPanel Orientation="Vertical" Margin="10 0 0 10">
    <RadioButton Content="Return Sale Transactions" x:Name="rb_r3"
    GroupName="lview" Checked="rb_r3_Checked" />
    <StackPanel Orientation="Horizontal">
        <TextBlock Text="START(YYYY-MM-DD)" Width="180" Margin="0 0 10 0"/>
        <TextBox x:Name="tb_Sdate3" Width="150" Margin="0 0 10 0" />
        <TextBlock Text="END(YYYY-MM-DD)" Margin="10 0 10 0"/>
        <TextBox x:Name="tb_Edate3" Width="150"/>
    </StackPanel>
</StackPanel>
<Button x:Name="btn_Rview" Click="btn_Rview_Click" Content="VIEW"
Width="100" Margin="0 0 0 10"/>
</StackPanel>
<DataGrid x:Name="dg_Rview" HorizontalAlignment="Left" Height="450"
Width="1000" />
</Grid>

</Page>
```

The screenshot shows a Windows application window titled "TRANSACTION REPORT". The window has a green header bar. Below it, there are three sections for filtering data:

- View Purchase Transaction:** Includes "START(YYYY-MM-DD)" and "END(YYYY-MM-DD)" date input fields.
- View Sale Transaction:** Includes "START(YYYY-MM-DD)" and "END(YYYY-MM-DD)" date input fields.
- Return Sale Transactions:** Includes "START(YYYY-MM-DD)" and "END(YYYY-MM-DD)" date input fields.

A "VIEW" button is located below the filters. A large, empty DataGrid control occupies the bottom half of the window, representing where the filtered transaction data would be displayed.

## Report.xaml.cs

```
Page | 138
      using System;
      using System.Collections.Generic;
      using System.Data;
      using System.Data.SqlClient;
      using System.Linq;
      using System.Text;
      using System.Threading.Tasks;
      using System.Windows;
      using System.Windows.Controls;
      using System.Windows.Data;
      using System.Windows.Documents;
      using System.Windows.Input;
      using System.Windows.Media;
      using System.Windows.Media.Imaging;
      using System.Windows.Navigation;
      using System.Windows.Shapes;

namespace testWPF
{
    /// <summary>
    /// Interaction logic for Report.xaml
    /// </summary>
    public partial class Report : Page
    {
        SqlConnection sqlCon = new SqlConnection(@"Data
Source=(localdb)\ProjectsV13;Initial Catalog=IMDB;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Multi
SubnetFailover=False");
        public Report()
        {
            InitializeComponent();
        }

        private void rb_r1_Checked(object sender, RoutedEventArgs e)
        {
            rb_r1.Background = Brushes.Yellow;
            rb_r1.Foreground = Brushes.Navy;
            rb_r2.Background = Brushes.White;
            rb_r2.Foreground = Brushes.Black;
```

```
rb_r3.Background = Brushes.White;
rb_r3.Foreground = Brushes.Black;
tb_Sdate2.Clear();
tb_Edate2.Clear();
tb_Sdate3.Clear();
tb_Edate3.Clear();
}

private void rb_r2_Checked(object sender, RoutedEventArgs e)
{
    rb_r1.Background = Brushes.White;
    rb_r1.Foreground = Brushes.Black;
    rb_r2.Background = Brushes.Yellow;
    rb_r2.Foreground = Brushes.Navy;
    rb_r3.Background = Brushes.White;
    rb_r3.Foreground = Brushes.Black;
    tb_Sdate1.Clear();
    tb_Edate1.Clear();
    tb_Sdate3.Clear();
    tb_Edate3.Clear();

}

private void rb_r3_Checked(object sender, RoutedEventArgs e)
{
    rb_r1.Background = Brushes.White;
    rb_r1.Foreground = Brushes.Black;
    rb_r2.Background = Brushes.White;
    rb_r2.Foreground = Brushes.Black;
    rb_r3.Background = Brushes.Yellow;
    rb_r3.Foreground = Brushes.Navy;
    tb_Sdate1.Clear();
    tb_Edate1.Clear();
    tb_Sdate2.Clear();
    tb_Edate2.Clear();

}

private void btn_Rview_Click(object sender, RoutedEventArgs e)
{
    if (rb_r1.IsChecked == true)
    {
        try
```

```

    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        string query = "SELECT [PB_ID], [Invoice_Id] 'Invoice No', [Vendor_Code],
[IMDB].[dbo].[tblProduct].[Product_Name] , [IMDB].[dbo].[tblProduct].[Product_Code],
[Quantity_Received], [Purchase_DOE] FROM [IMDB].[dbo].[tblPurchaseBill] INNER JOIN
[IMDB].[dbo].[tblProduct] ON
[IMDB].[dbo].[tblPurchaseBill].[Product_Code]=[IMDB].[dbo].[tblProduct].[Product_Code]
WHERE [Purchase_DOE] BETWEEN '" + tb_Sdate1.Text + "' AND '" + tb_Edate1.Text + "' ";
        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
        sqlcmd.CommandType = System.Data.CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataReader sdr = sqlcmd.ExecuteReader();

        dt.Load(sdr);
        dg_Rview.ItemsSource = dt.DefaultView;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        sqlCon.Close();
    }
}

if (rb_r2.IsChecked == true)
{
    try
    {
        if (sqlCon.State == System.Data.ConnectionState.Closed)
            sqlCon.Open();

        string query = "SELECT [SB_ID], [Sale_Id] 'Sale No', [Customer_Name],
[Customer_Contact], [IMDB].[dbo].[tblProduct].[Product_Name] ,
[IMDB].[dbo].[tblProduct].[Product_Code], [Quantity_Sold], [Sale_DOE] FROM
[IMDB].[dbo].[tblSaleBill] INNER JOIN [IMDB].[dbo].[tblProduct] ON
[IMDB].[dbo].[tblSaleBill].[Product_Code]=[IMDB].[dbo].[tblProduct].[Product_Code]
WHERE [Sale_DOE] BETWEEN '" + tb_Sdate2.Text + "' AND '" + tb_Edate2.Text + "' ";
        SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
    }
}

```

```
sqlcmd.CommandType = System.Data.CommandType.Text;
DataTable dt = new DataTable();
SqlDataReader sdr = sqlcmd.ExecuteReader();
```

```
dt.Load(sdr);
dg_Rview.ItemsSource = dt.DefaultView;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
finally
{
    sqlCon.Close();
}
}
```

```
if (rb_r3.IsChecked == true)
{
```

```
try
{
    if (sqlCon.State == System.Data.ConnectionState.Closed)
        sqlCon.Open();
```

```
    string query = "SELECT [RS_ID], [ReturnSale_Id] 'Return Sale No',[Sale_Id] 'Sale
No', [IMDB].[dbo].[tblProduct].[Product_Name] ,
[IMDB].[dbo].[tblProduct].[Product_Code], [Quantity_Return], [Return_DOE] FROM
[IMDB].[dbo].[tblRSBill] INNER JOIN [IMDB].[dbo].[tblProduct] ON
[IMDB].[dbo].[tblRSBill].[Product_Code]=[IMDB].[dbo].[tblProduct].[Product_Code] WHERE
[Return_DOE] BETWEEN '" + tb_Sdate3.Text + "' AND '" + tb_Edate3.Text+ "' ";
```

```
    SqlCommand sqlcmd = new SqlCommand(query, sqlCon);
    sqlcmd.CommandType = System.Data.CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataReader sdr = sqlcmd.ExecuteReader();
```

```
    dt.Load(sdr);
    dg_Rview.ItemsSource = dt.DefaultView;
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
```

```
        }
    finally
    {
        sqlCon.Close();
    }
}

}
```

## 8.0 TESTING

### 1. LOGIN

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_LOGIN_001	Verify Login	Enter Valid Username and Valid Password	Need a Valid Username and Valid Password	1. Enter Valid Username 2. Enter Password 3. Press Submit Button	<Valid User Name> <Valid Password>	Successful Login and MainWindow Window Opens	Same as Expected	Pass
TG_LOGIN_002		Enter Valid Username and Invalid Password		1. Enter Username 2. Enter Invalid Password 3. Press Submit Button	<Valid Username> <Invalid Password>	Message Box Showing Message User Name or Password is Incorrect appears	Same as Expected	Pass
TG_LOGIN_003		Enter Invalid Username and Valid Password		1. Enter Invalid Username 2. Enter Valid Password 3. Press Submit Button	<Invalid Username> <Valid Password>	Message Box Showing Message User Name or Password is Incorrect appears	Same as Expected	Pass
TG_LOGIN_004		Enter Invalid Username and Invalid Password		1. Enter Invalid Username 2. Enter Invalid Password 3. Press Submit Button	<Invalid Username> <Invalid Password>	Message Box Showing Message User Name or Password is Incorrect appears	Same as Expected	Pass
TG_LOGIN_005		Empty Username Textbox Empty Password Textbox		1. Press Submit Button		Nothing Happens	Same as Expected	Pass

### 2.0 ADDING VENDORS

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_VENDOR_001	Verify addition of vendor data	Enter Fields in Vendor Form and Submitting it through SAVE button	Successful Logged in Select Vendor from Dashboard	Fill various Fields with Valid value and Press Save Button	<Valid Value of Various Fields>	Message Box Showing Message 'New Vendor Successfully Added' and same is loaded in	Same as Expected	Pass

			Select ADD from Vendor Page			Vendor Page's Data Grid		
TG_VENDOR_002			Successful Logged in Select Vendor from Dashboard Select ADD from Vendor Page	Fill Various Fields with Valid Value and leave one or more fields empty	<Valid Value of Various Fields> <Leave Some Field Empty>	Message Box Showing Message 'Field_Name is Required'+'Failed' Form is Successfully Added only upon Filling all fields	Same as Expected	Pass

### 3.0 ADDING PRODUCTS

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_PRODUCT_001	Verify addition of products data	Enter Fields in Product Form and Submitting it through SAVE button	Successful Logged in Select Product from Dashboard Select ADD from Product Page	Fill various Fields with Valid value and Press Save Button	<Valid Value of Various Fields>	Message Box Showing Message 'New Product Successfully Added' and same is loaded in Product Page's Data Grid	Same as Expected	Pass
TG_PRODUCT_002			Successful Logged in Select Product from Dashboard Select ADD from Product Page	Fill Various Fields with Valid Value and leave one or more fields empty	<Valid Value of Various Fields> <Leave Some Field Empty>	Message Box Showing Message 'Field_Name Required' 'Failed' Form is Successfully Added only upon Filling all fields	Same as Expected	Pass

### 4.0 ADDING PURCHASE BILL

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_PBE_001	Verify addition of purchase bill data	Enter Fields in Purchase Bill Entry Form and Submitting it through ADD button	Successful Logged in Select Purchase Bill Entry from Dashboard Select ADD from Purchase Bill Entry Page	On Purchase Bill Entry Form fill Invoice No, select Vendor, fill Date, select Product, Enter Quantity and Press Add Button	<Valid Invoice No> <Valid Vendor> <Valid Date> <Valid Product> <Valid Quantity>	Message Box Showing Message 'Purchase Bill Entry Successful' appears and Product Stock Quantity is Updated	Same As Expected	Pass
TG_PBE_002	Verify addition of	Enter Fields in Purchase	Successful Logged in	On Purchase Bill Entry	<Empty Invoice No>	Message Box Showing	Same As Expected	Pass

	purchase bill data	Bill Entry Form and Submitting it through ADD button	Select Purchase Bill Entry from Dashboard Select ADD from Purchase Bill Entry Page	Form Invoice No is Kept empty, select Vendor, fill Date, select Product, Enter Quantity and Press Add Button	<Valid Vendor> <Valid Date> <Valid Product> <Valid Quantity>	Message 'Invoice No. is Required' and Product Stock Quantity is not Updated		
TG_PBE_003	Verify addition of purchase bill data	Enter Fields in Purchase Bill Entry Form and Submitting it through ADD button	Successful Logged in Select Purchase Bill Entry from Dashboard Select ADD from Purchase Bill Entry Page	On Purchase Bill Entry Form fill Invoice No, Vendor is not selected, fill Date, select Product, Enter Quantity and Press Add Button	<Valid Invoice No> <Vendor not Selected> <Valid Date> <Valid Product> <Valid Quantity>	Message Box Showing Message 'Vendor is Required' and Product Stock Quantity is not Updated	Same As Expected	Pass
TG_PBE_004	Verify addition of purchase bill data	Enter Fields in Purchase Bill Entry Form and Submitting it through ADD button	Successful Logged in Select Purchase Bill Entry from Dashboard Select ADD from Purchase Bill Entry Page	On Purchase Bill Entry Form fill Invoice No, select Vendor, Date is kept Empty, select Product, Enter Quantity and Press Add Button	<Empty Invoice No> <Valid Vendor> <Empty Date> <Valid Product> <Valid Quantity>	Message Box Showing Message 'Date is Required' and Product Stock Quantity is not Updated	Same As Expected	Pass
TG_PBE_005	Verify addition of purchase bill data	Enter Fields in Purchase Bill Entry Form and Submitting it through ADD button	Successful Logged in Select Purchase Bill Entry from Dashboard Select ADD from Purchase Bill Entry Page	On Purchase Bill Entry Form fill Invoice No, Select Vendor, fill Date, Product is not selected or Enter Quantity is not filled or both and Press Add Button	<Valid Invoice No> <Valid Vendor> <Valid Date> <Product not> <Valid Quantity>/<Product Selected> <Empty Quantity>/ Both empty	Message Box Showing Message 'Some Field is Missing. Entry Unsuccessful' and Product Stock Quantity is not Updated	Same As Expected	Pass

## 5.0 ADDING SALE BILL

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_SAE_001	Verify addition of sale bill data	Enter Fields in Sale Acceptance Entry Form and Submitting it	Successful Logged in Select Sale Acceptance Entry Form from Dashboard	On Sale Acceptance Entry Form Fill Sale Bill No, fill Customer Name, fill Customer Name, fill Contact	<Valid Sale Bill No> <Valid Customer Name> <Valid Contact>	Message Box Showing Message 'Sale Bill Entry Successful' appears and	Same As Expected	Pass

		through ADD button	Select ADD from Sale Acceptance Entry Page	Contact, fill Date, select Product, Enter Quantity and Press Add Button	<Valid Date> <Valid Product> <Valid Quantity>	Product Stock Quantity is Updated			
Page   146	TG_SAE_002	Verify addition of sale bill data	Enter Fields in Sale Acceptance Entry Form and Submitting it through ADD button	Successful Logged in Select Sale Acceptance Entry from Dashboard Select ADD from Sale Acceptance Entry Page	On Sale Acceptance Entry Form Empty Sale Bill No, fill Costumer Name, fill Contact, fill Date, select Product, Enter Quantity and Press Add Button	<Empty Sale No> <Valid Customer Name> <Valid Contact> <Valid Date> <Valid Product> <Valid Quantity>	Message Box Showing Message 'Sale Bill No. is Required' and Product Stock Quantity is not Updated	Same As Expected	Pass
	TG_SAE_003	Verify addition of sale bill data	Enter Fields in Sale Acceptance Entry Form and Submitting it through ADD button	Successful Logged in Select Sale Acceptance Entry from Dashboard Select ADD from Sale Acceptance Entry Page	On Sale Acceptance Entry Form Fill Sale Bill No, Empty Costumer Name, fill Contact, fill Date, select Product, Enter Quantity and Press Add Button	<Valid Sale No> <Empty Customer Name> <Valid Contact> <Valid Date> <Valid Product> <Valid Quantity>	Message Box Showing Message 'Customer Name is Required' and Product Stock Quantity is not Updated	Same As Expected	Pass
	TG_SAE_004	Verify addition of sale bill data	Enter Fields in Sale Acceptance Entry Form and Submitting it through ADD button	Successful Logged in Select Sale Acceptance Entry from Dashboard Select ADD from Sale Acceptance Entry Page	On Sale Acceptance Entry Form Fill Sale Bill No, fill Costumer Name, fill Contact, Empty Contact, fill Date, select Product, Enter Quantity and Press Add Button	<Valid Sale No> <Valid Customer Name> <Empty Contact> <Valid Date> <Valid Product> <Valid Quantity>	Message Box Showing Message 'Contact is Required' and Product Stock Quantity is not Updated	Same As Expected	Pass
	TG_SAE_004	Verify addition of sale bill data	Enter Fields in Sale Acceptance Entry Form and Submitting it through ADD button	Successful Logged in Select Sale Acceptance Entry from Dashboard Select ADD from Sale Acceptance Entry Page	On Sale Acceptance Entry Form Fill Sale Bill No, fill Costumer Name, fill Contact, Empty Date, select Product, Enter Quantity and Press Add Button	<Valid Sale No> <Valid Customer Name> <Valid Contact> <Empty Date> <Valid Product> <Valid Quantity>	Message Box Showing Message 'Date is Required' and Product Stock Quantity is not Updated	Same As Expected	Pass
	TG_SAE_004	Verify addition of sale bill data	Enter Fields in Sale Acceptance	Successful Logged in	On Sale Acceptance Entry Form	<Valid Sale No>	Message Box Showing Message	Same As Expected	Pass

		Entry Form and Submitting it through ADD button	Select Sale Acceptance Entry from Dashboard Select ADD from Sale Acceptance Entry Page	Fill Sale Bill No, fill Costumer Name, fill Contact, Empty Date, Product not selected, Enter Quantity and Press Add Button	<Valid Customer Name> < Valid Contact> <Valid Date> <Product not Selected> <Valid Quantity>	'Some Field is Missing. Entry Unsuccessful' and Product Stock Quantity is not Updated		
TG_SAE_004	Verify addition of sale bill data	Enter Fields in Sale Acceptance Entry Form and Submitting it through ADD button	Successful Logged in Select Sale Acceptance Entry from Dashboard Select ADD from Sale Acceptance Entry Page	On Sale Acceptance Entry Form Fill Sale Bill No, fill Costumer Name, fill Contact, fill Date, select Product, Empty Quantity and Press Add Button	<Valid Sale No> <Valid Customer Name> < Valid Contact> <Empty Date> <Valid Product> <Empty Quantity Sold>	Message Box Showing Message 'Some Field is Missing. Entry Unsuccessful' and Product Stock Quantity is not Updated	Same As Expected	Pass
TG_SAE_005	Verify addition of sale bill data	Enter Fields in Sale Acceptance Entry Form and Submitting it through ADD button	Successful Logged in Select Sale Acceptance Entry from Dashboard Select ADD from Sale Acceptance Entry Page	On Sale Acceptance Entry Form Fill Sale Bill No, fill Costumer Name, fill Contact, fill Date, select Product, Enter Invalid Quantity and Press Add Button	<Valid Sale Bill No> <Valid Customer Name> <Valid Contact> <Valid Date> <Valid Product> (Quantity Sold> Stock Quantity)	Message Box Showing Message 'Stock Not Available. Please Buy Another Product' appears and Product Stock Quantity is not Updated	Same As Expected	Pass

## 6.0 ADDING RETURN SALE BILL

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_RSE_001	Verify addition of return sale bill data	Enter Fields in Return Sale Entry Form and Submitting it through ADD button	Successful Logged in Select Return Sale Entry from Dashboard Select ADD from Return Sale Entry Page	On Return Sale Entry Form Fill Return Sale No, fill Sale No and Press Search Button, fill Date, select Product, Enter Quantity Returned and Press Add Button	<Valid Return Sale No> <Valid Sale No> <Valid Date> <Valid Product> <Valid Quantity Returned>	Message Box Showing Message 'Return Sale Bill Entry Successful' appears and Product Stock Quantity is Updated	Same As Expected	Pass
TG_RSE_001	Verify addition of return sale bill data	Enter Fields in Return Sale Entry Form and Submitting it	Successful Logged in Select Return Sale	On Return Sale Entry Form Empty Return Sale No, fill Sale	<Valid Return Sale No> <Valid Sale No>	Message Box Showing Message 'Return Sale No is	Same As Expected	Pass

		through ADD button	Entry from Dashboard Select ADD from Return Sale Entry Page	No and Press Search Button, fill Date, select Product, Enter Quantity Returned and Press Add Button	<Valid Date> <Valid Product> <Valid Quantity Returned>	Required' appears and Product Stock Quantity is not Updated		
TG_RSE_001	Verify addition of return sale bill data	Enter Fields in Return Sale Entry Form and Submitting it through ADD button	Successful Logged in Select Return Sale Entry from Dashboard Select ADD from Return Sale Entry Page	On Return Sale Entry Form fill Return Sale No, either Empty Sale No or Search Button not pressed or both, fill Date, select Product, Enter Quantity Returned and Press Add Button	<Valid Return Sale No> <Invalid Sale No> <Valid Date> <Valid Product> <Valid Quantity Returned>	Message Box Showing Message 'Sale No is Required' appears, Product Combo box not loaded and Product Stock Quantity is not Updated	Same As Expected	Pass
TG_RSE_004	Verify addition of return sale bill data	Enter Fields in Return Sale Entry Form and Submitting it through ADD button	Successful Logged in Select Return Sale Entry from Dashboard Select ADD from Return Sale Entry Page	On Return Sale Entry Form fill Return Sale No, fill Sale No and Press Search Button, Empty Date, select Product, Enter Quantity Returned and Press Add Button	<Valid Return Sale No> <Valid Sale No> <Empty Date> <Valid Product> <Valid Quantity Returned>	Message Box Showing Message 'Date is Required' appears and Product Stock Quantity is not Updated	Same As Expected	Pass
TG_RSE_005	Verify addition of return sale bill data	Enter Fields in Return Sale Entry Form and Submitting it through ADD button	Successful Logged in Select Return Sale Entry from Dashboard Select ADD from Return Sale Entry Page	On Return Sale Entry Form fill Return Sale No, fill Sale No and Press Search Button, fill Date, either Product not selected or Quantity Returned not entered or both and Press Add Button	<Valid Return Sale No> <Valid Sale No> <Valid Date> <Invalid Product> <Valid Quantity Returned> or <Valid Product> <Invalid Quantity Returned> or both Invalid	Message Box Showing Message 'Some Field is Missing. Return Sale Bill Entry Unsuccessful' appears and Product Stock Quantity is not Updated	Same As Expected	Pass
TG_RSE_006	Verify addition of return sale bill data	Enter Fields in Return Sale Entry Form and Submitting it	Successful Logged in Select Return Sale Entry from Dashboard	On Return Sale Entry Form fill Return Sale No, fill Sale No and	<Valid Return Sale No> <Valid Sale No> <Valid Date>	Message Box Showing Message 'Entered Quantity is more than	Same As Expected	Pass

		through ADD button	Select ADD from Return Sale Entry Page	Press Search Button, fill Date, fill Product, fill Quantity Returned and Press Add Button	<Valid Product> (Quantity Returned > Quantity Sold)	Sold Quantity.' appears and Product Stock Quantity is not Updated		
--	--	--------------------	--	---	---	---	--	--

Page |  
149

## 7.0 VIEWING INVENTORY REPORT

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_IR_001	View Report in Data Grid as per selection of Radio Button 1	All other text box which are not concerned should be cleared and change in colour of radio button and text selected	Successful Logged in Select Inventory Report from Dashboard	Select 1 <sup>st</sup> Radio Button and Press View		Change in colour of radio button, clear other text boxes, presentation of data in view	Same As Expected	Pass
TG_IR_002	View Report in Data Grid as per selection of Radio Button 1	All other text box which are not concerned should be cleared and change in colour of radio button and text selected	Successful Logged in Select Inventory Report from Dashboard	Select 2nd Radio Button, Enter Valid Product Code in Text Box1 and Press View	<Valid Product Code>	Change in colour of radio button, clear other text boxes, presentation of data in view	Same As Expected	Pass
TG_IR_003	View Report in Data Grid as per selection of Radio Button 1	All other text box which are not concerned should be cleared and change in colour of radio button and text selected	Successful Logged in Select Inventory Report from Dashboard	Select 3rd Radio Button, Enter Valid Product Code in Text Box2 and Press View	<Valid Product Code>	Change in colour of radio button, clear other text boxes, presentation of data in view	Same As Expected	Pass

## 8.0 VIEWING REPORT

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_TR_001	View Report in Data Grid as per selection of Radio Button 1	All other text box which are not concerned should be cleared and change in colour of	Successful Logged in Select Inventory Report from Dashboard	Select 1 <sup>st</sup> Radio Button, End Valid Start Date & End Date and Press View	<Valid Start Date> <Valid End Date>	Change in colour of radio button, clear other text boxes, presentation of data in view	Same As Expected	Pass

		radio button and text selected						
TG_TR_002	View Report in Data Grid as per selection of Radio Button 2	All other text box which are not concerned should be cleared and change in colour of radio button and text selected	Successful Logged in Select Inventory Report from Dashboard Select 2nd Radio Button and Press View	Select 2nd Radio Button, End Valid Start Date & End Date and Press View	<Valid Start Date> <Valid End Date>	Change in colour of radio button, clear other text boxes, presentation of data in view	Same As Expected	Pass
TG_TR_003	View Report in Data Grid as per selection of Radio Button 3	All other text box which are not concerned should be cleared and change in colour of radio button and text selected	Successful Logged in Select Inventory Report from Dashboard	Select 3rd Radio Button, End Valid Start Date & End Date and Press View	<Valid Start Date> <Valid End Date>	Change in colour of radio button, clear other text boxes, presentation of data in view	Same As Expected	Pass

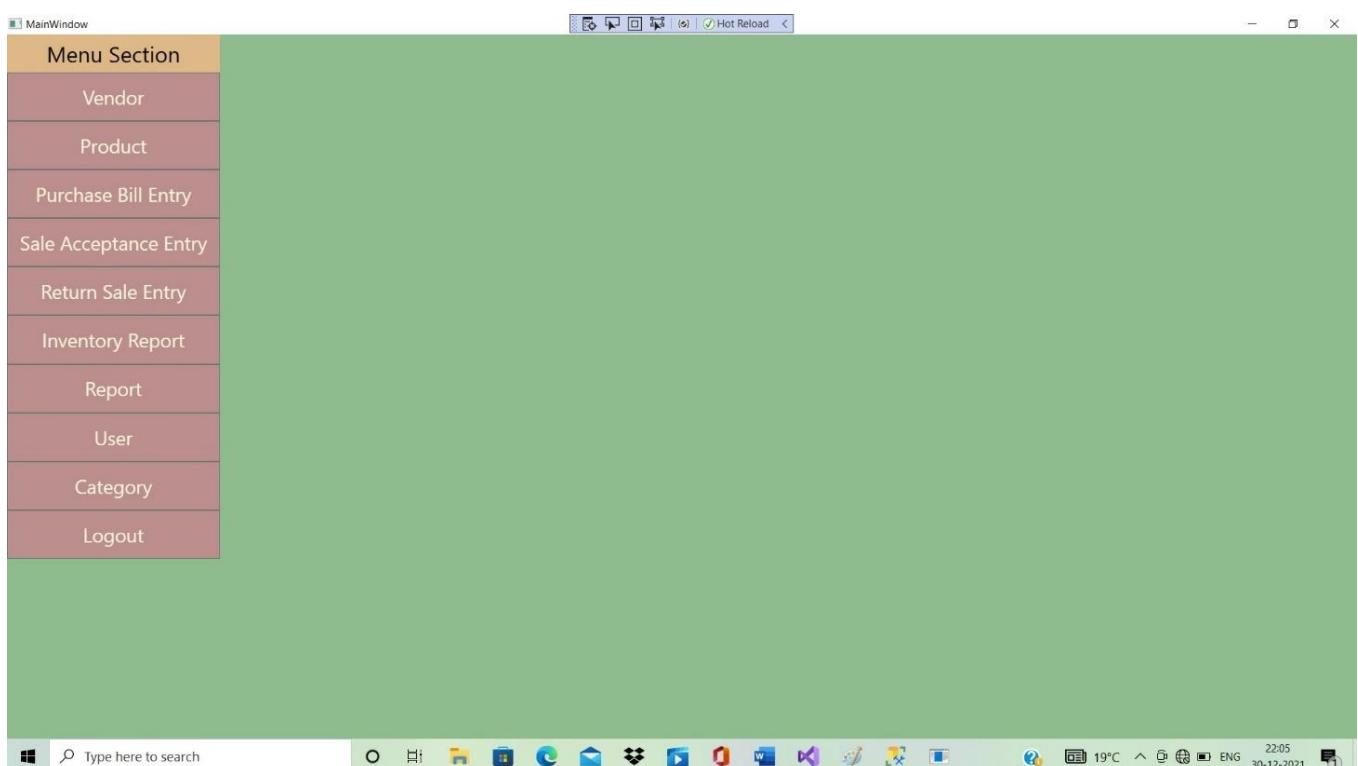
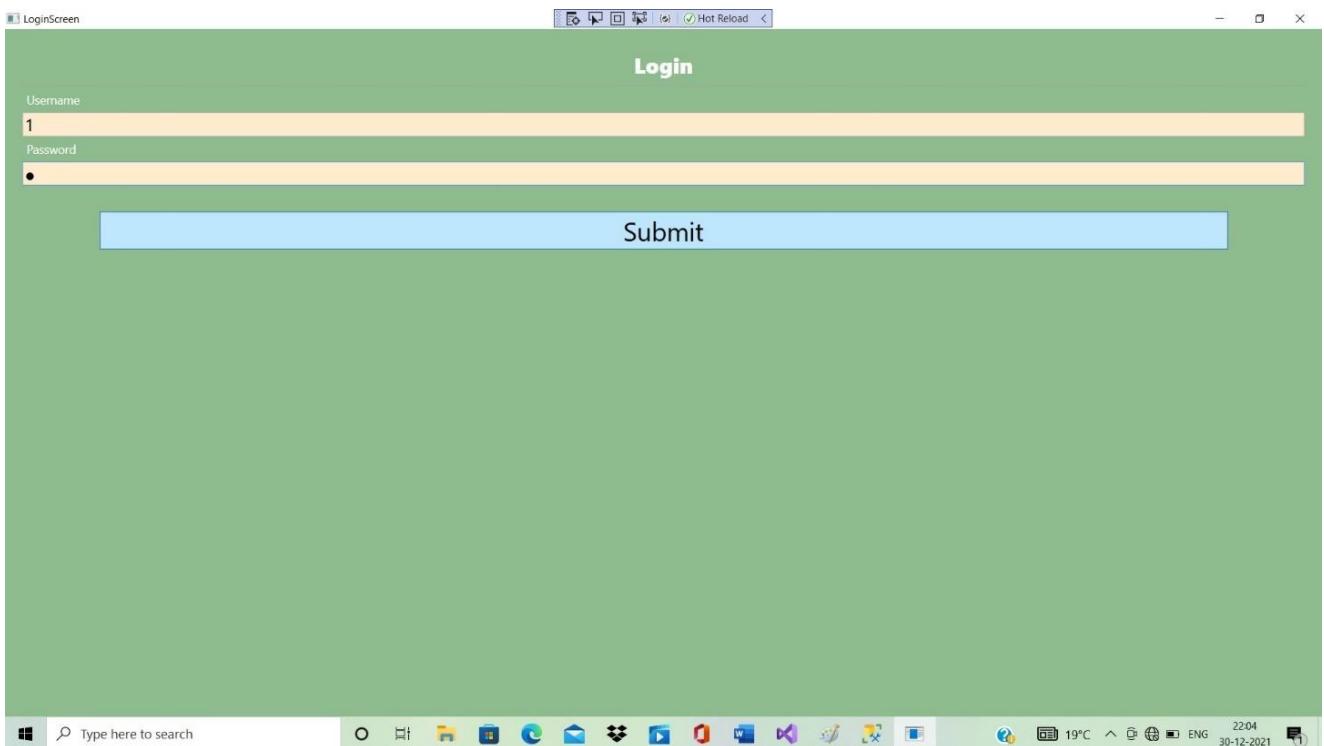
## 9.0 Logout

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS
TG_LO_001	Verify Logout Operation	Click Logout Button on Menu Section of MainWindow Window	Successful Logged in	Successful Login, Click Logout Button		Should close Main Window and take to login Page	Same as Expected	Pass

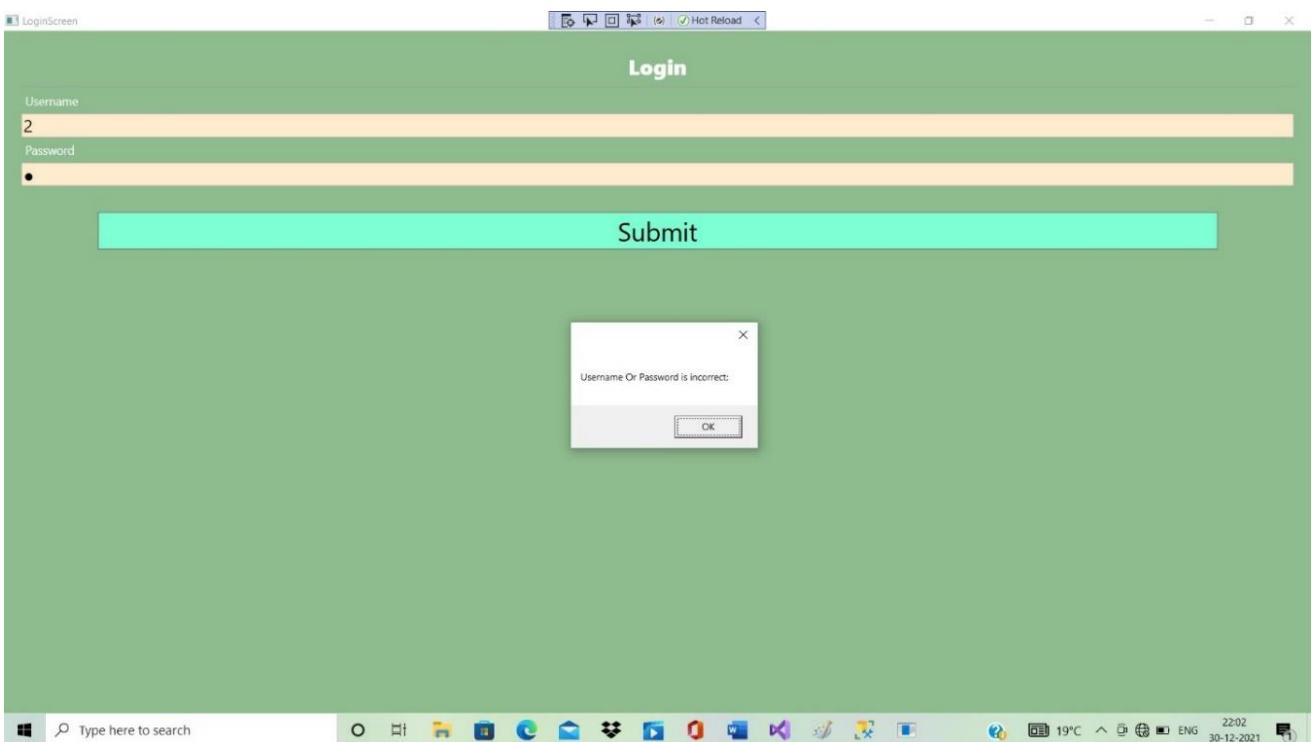
## 9.0 INPUT AND OUTPUT SCREEN

### Login Page

- Page |  
151
- Fill Username and Password.
  - Press Submit Button.
  - If Valid Username and Password, it opens the application to MainWindow Window for doing business.
  - If Invalid Username or Invalid Password, it shows a message box displaying Invalid Username or Password

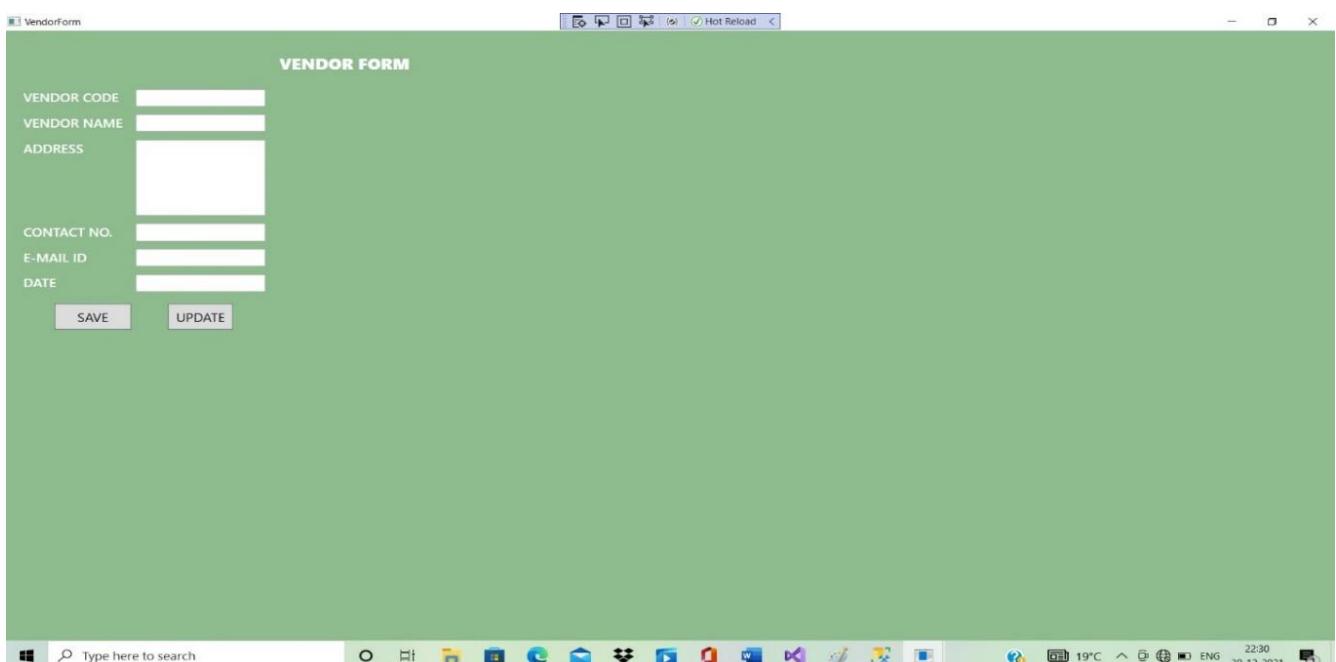


Unsuccessful Login Produces following:

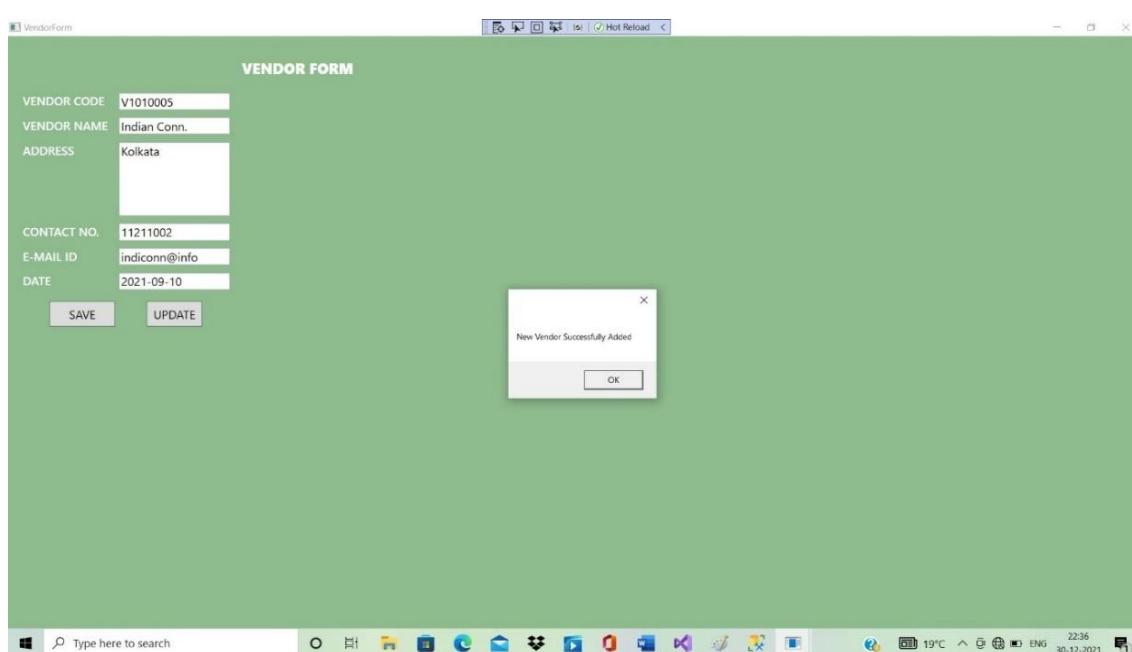
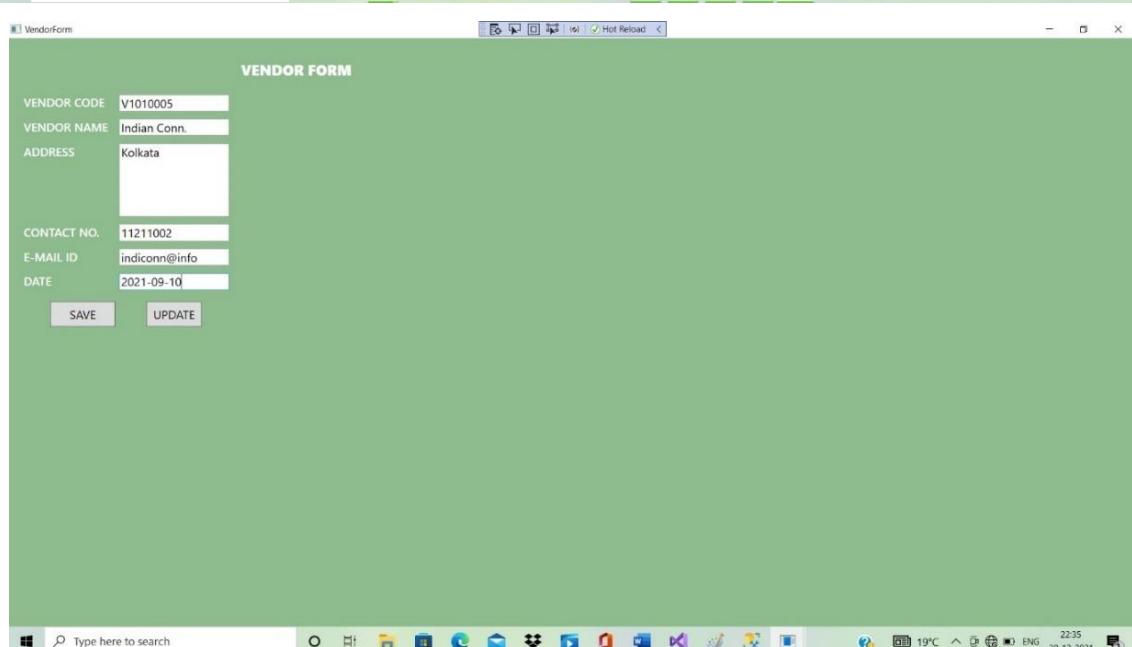
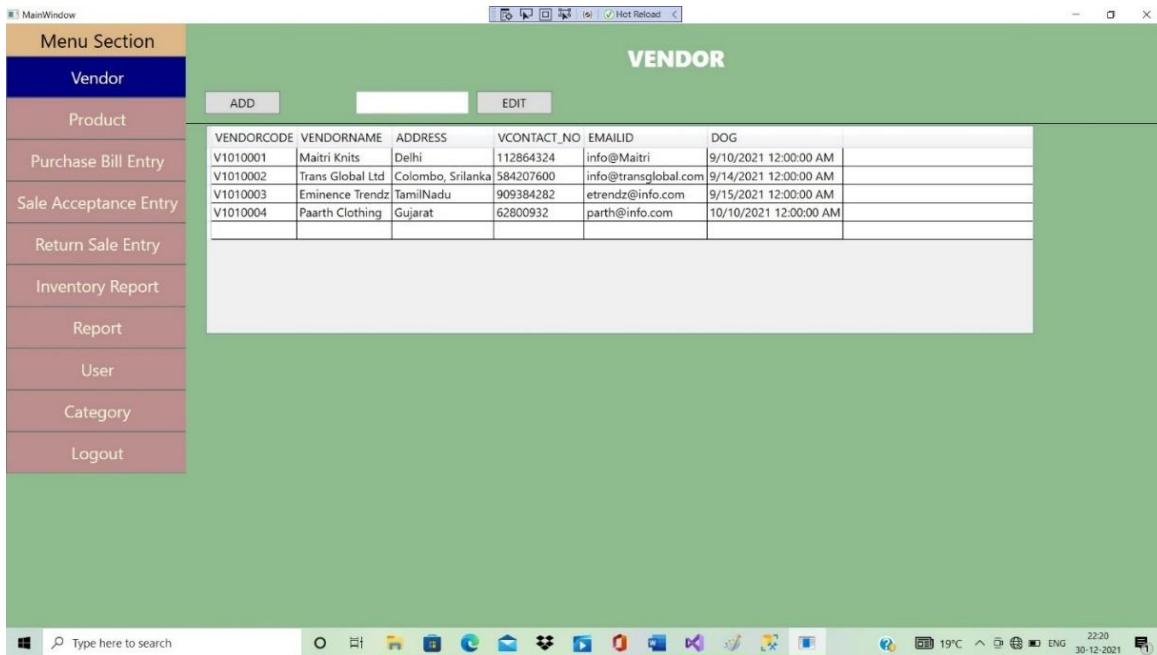


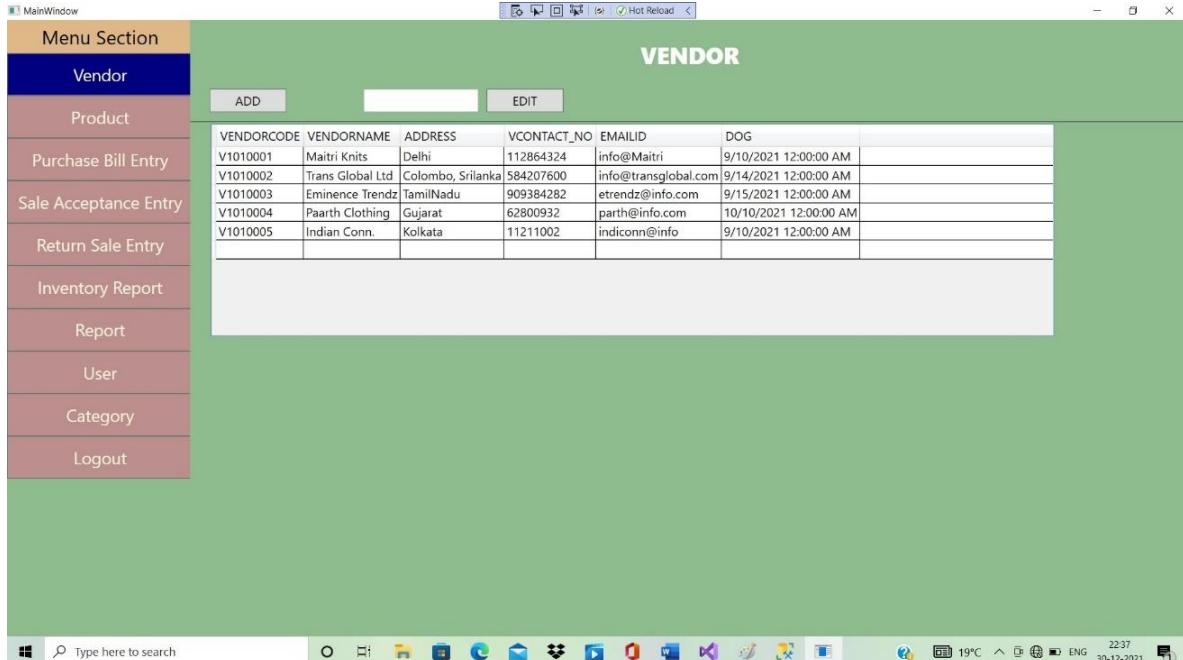
## ADDING, EDITING AND VIEWING VENDOR

- After Successful Login, Select Vendor Button
- Clicking Vendor Button will open Vendor Page in Iframe



- Clicking ADD Button in Vendor Page will open a Vendor Form.
- Fill the fields in Vendor Form(Window) and Click SAVE to store in database.
- Saved data can be seen in Data Grid of Vendor Page.

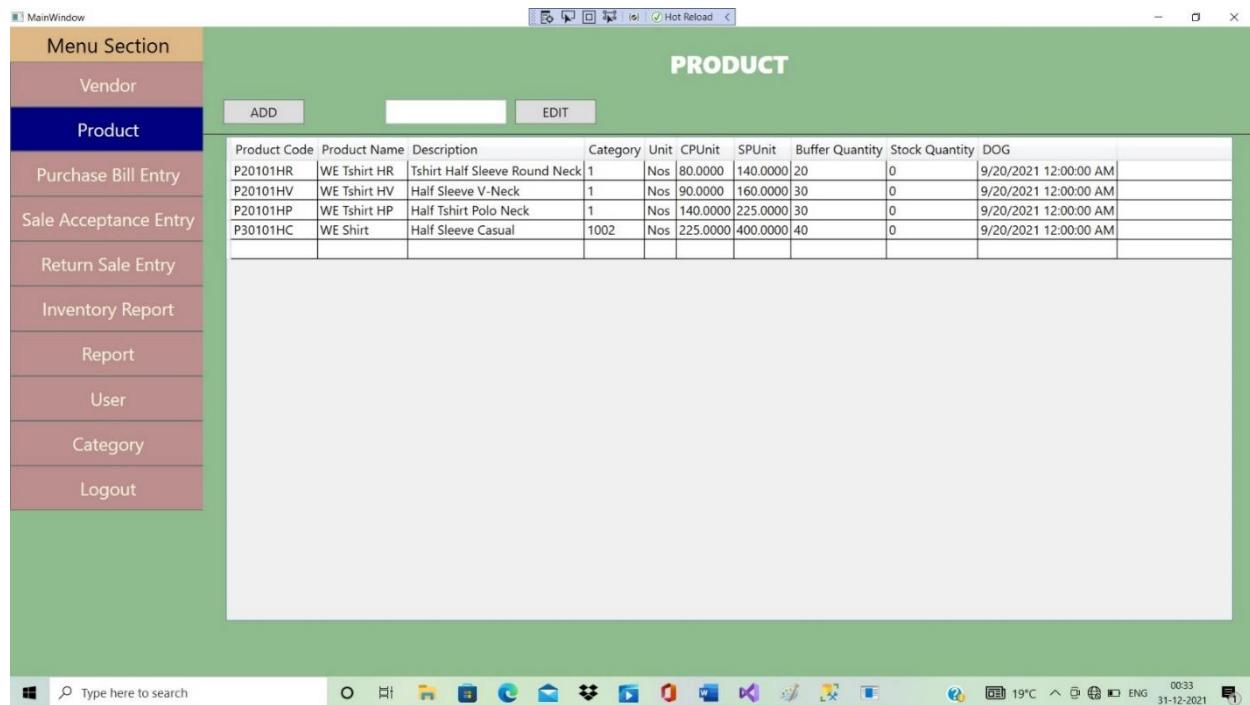




- To edit Vendor details, type Vendor Code of Vendor in Text Box and click Edit Button.
- Clicking Edit Button, will open a Vendor Form (Window) filled with previous data.
- Make required changes and click Update Button

### ADDING, EDITING AND VIEWING PRODUCTS

- After successful login, one can add new products detail by clicking Product Button from Dashboard (MainWindow).
- Clicking Product Button will open a Product Page in its Iframe.



- To add new products one can, click on Add Button.
- Clicking Add Button Will Open a new ProductForm Window. One can fill various field and click on Save Button to save new product details in database. A message box will appear displaying message "New Product Added Successfully".
- It is loaded in Data Grid of Product Page.

ProductForm

PRODUCT FORM

PRODUCT CODE	<input type="text"/>
PRODUCT NAME	<input type="text"/>
DESCRIPTION	<input type="text"/>
CATEGORY	<input type="text"/>
VENDOR	<input type="text"/>
LOCATION	<input type="text"/>
UNIT	<input type="text"/>
CP/UNIT	<input type="text"/>
SP/UNIT	<input type="text"/>
BUFFER QUANTITY	<input type="text"/>
STOCK QUANTITY	<input type="text"/>
DATE	<input type="text"/>

Type here to search

Hot Reload

ProductForm

PRODUCT FORM

PRODUCT CODE	P20102HR
PRODUCT NAME	NXT Tshirt HR
DESCRIPTION	f Sleeve Round Neck
CATEGORY	<input type="text"/>
VENDOR	<input type="text"/>
LOCATION	<input type="text"/>
UNIT	<input type="text"/>
CP/UNIT	<input type="text"/>
SP/UNIT	<input type="text"/>
BUFFER QUANTITY	<input type="text"/>
STOCK QUANTITY	<input type="text"/>
DATE	<input type="text"/>

Type here to search

Hot Reload

ProductForm

PRODUCT FORM

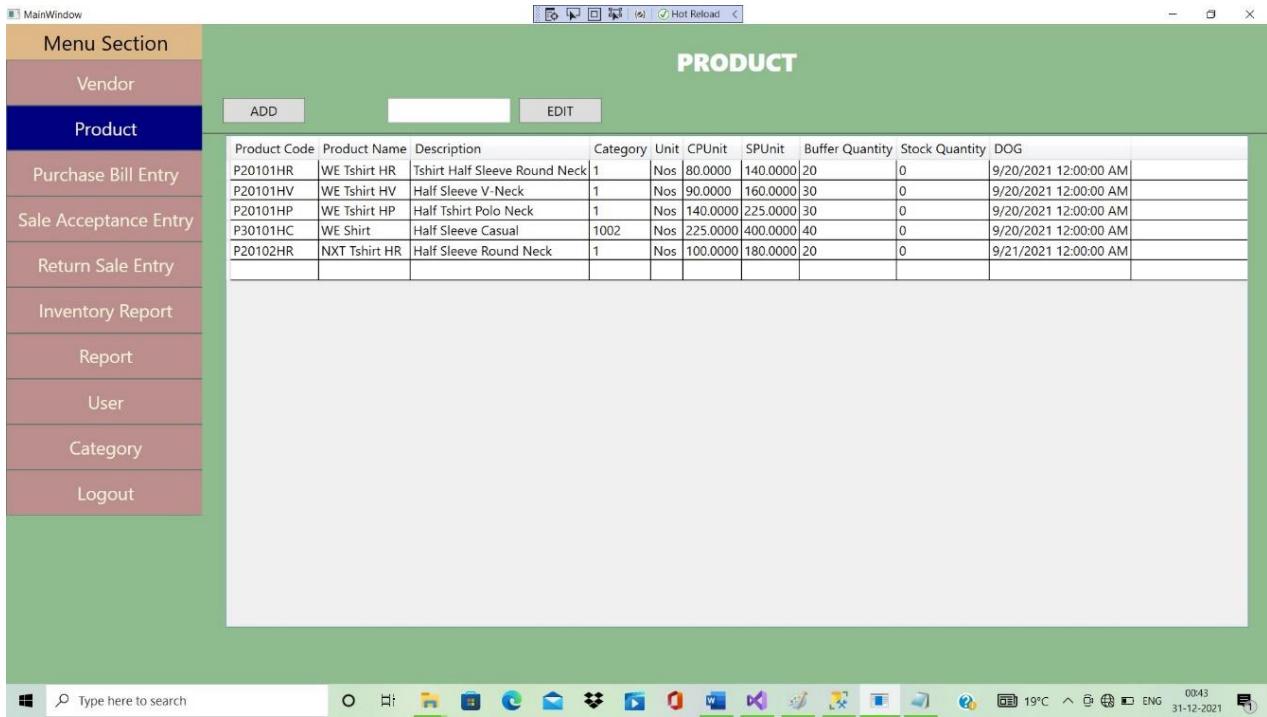
PRODUCT CODE	P20102HR
PRODUCT NAME	NXT Tshirt HR
DESCRIPTION	f Sleeve Round Neck
CATEGORY	<input type="text"/>
VENDOR	<input type="text"/>
LOCATION	<input type="text"/>
UNIT	<input type="text"/>
CP/UNIT	<input type="text"/>
SP/UNIT	<input type="text"/>
BUFFER QUANTITY	<input type="text"/>
STOCK QUANTITY	<input type="text"/>
DATE	<input type="text"/>

New Product Successfully Added

OK

Type here to search

Hot Reload



If one wants to edit details, one will click Edit button on Product Page.

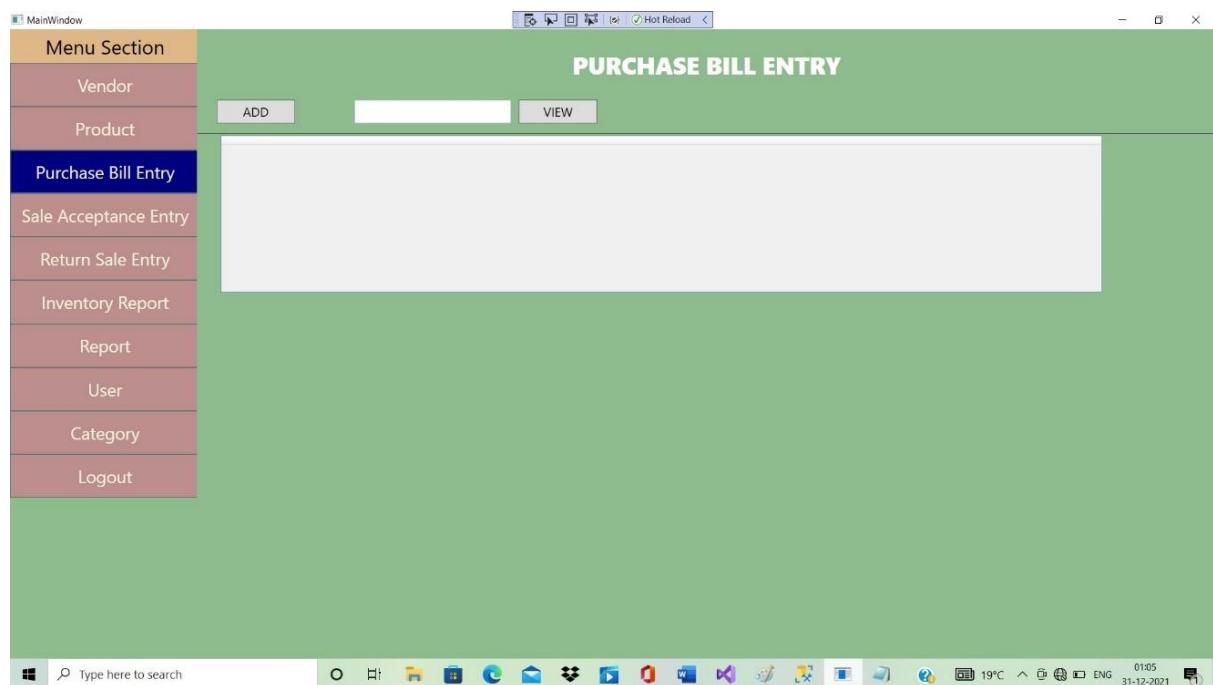
Clicking Edit Button, opens a filled ProductForm.

Make required changes and click Update Button.

A message Box appear displaying "Product Updated Successfully.".

## Adding and Viewing Purchase Bill Entry

- After Successful Login, one can add Purchase Bill Details in his database.
- First from MainWindow Dashboard, He need to click Purchase Bill Entry Button.
- Clicking it opens a PurchaseBillEntry Page in its IFrame. On this Page one can view Add Button, A Text Box and a View Button linked to it.



- Clicking on Add Button, will open a Window named PBEform.
- Fill the fields and click the Add Button.
- A message box displaying message “Purchase Bill Successfully Added” will appear confirming the addition of Purchase Bill Details to database.
- On PurchaseBillEntry Page one can type Invoice No. in Text box and click View Button to see details of particular invoice.

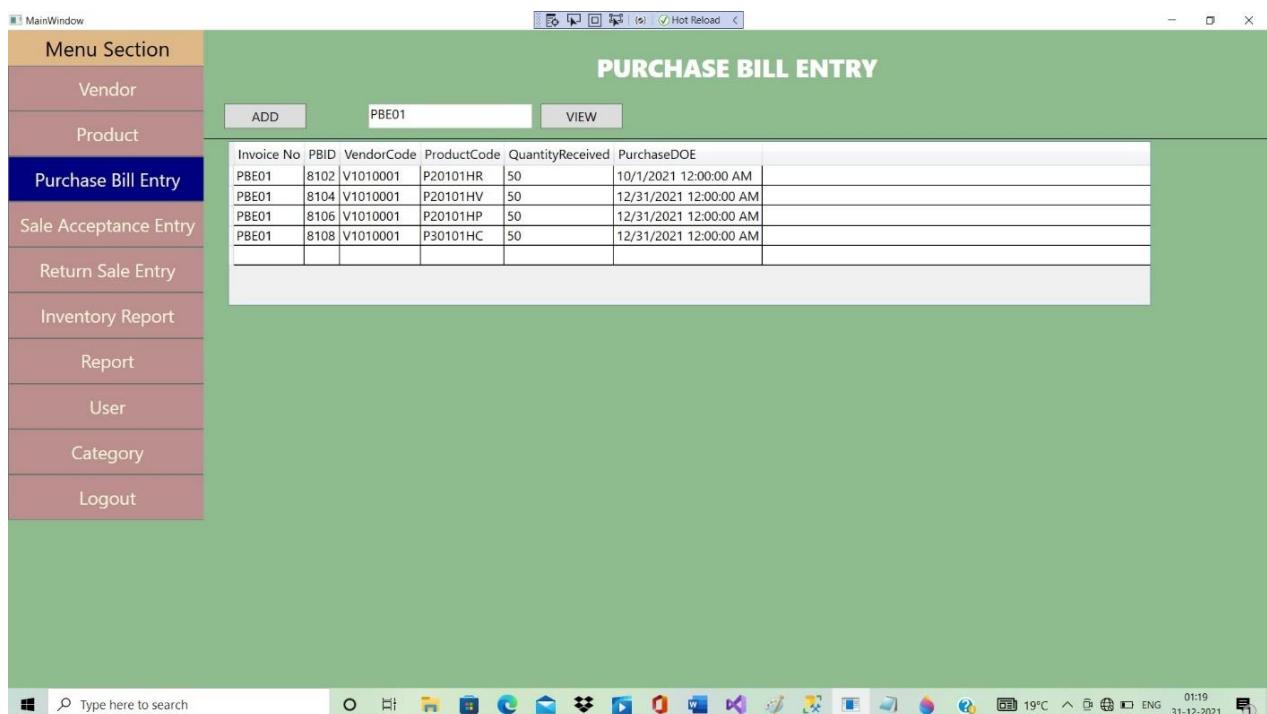
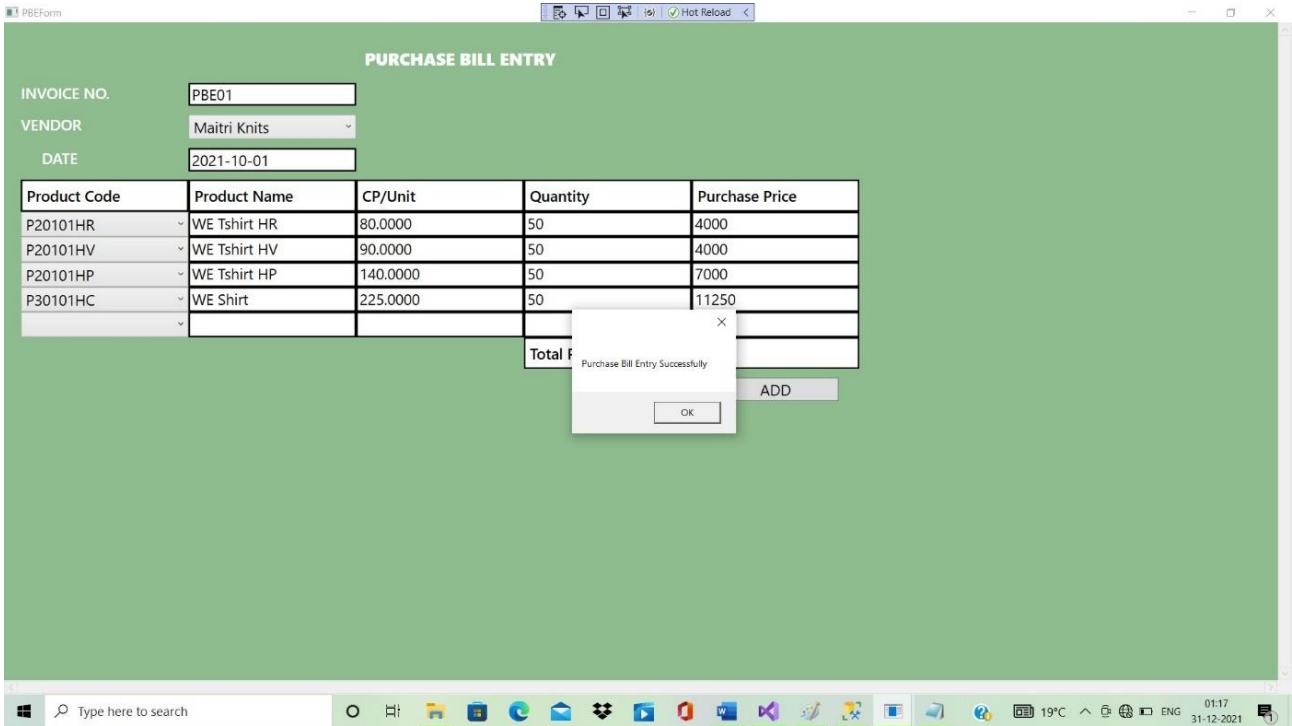
Page | 157

The screenshot shows a Windows application window titled "PBEForm". The main title bar has icons for minimize, maximize, and close, along with a "Hot Reload" button. Below the title bar is a toolbar with standard Windows buttons. The main area is titled "PURCHASE BILL ENTRY". It contains three input fields: "INVOICE NO.", "VENDOR", and "DATE", each with a dropdown arrow. Below these is a table with columns: "Product Code", "Product Name", "CP/Unit", "Quantity", and "Purchase Price". There are 10 rows in the table, each with a dropdown arrow in the first column. At the bottom of the table is a row labeled "Total Price". A grey "ADD" button is located at the bottom right of the table area. The status bar at the bottom shows the date and time as 31-12-2021 01:14.

The screenshot shows the same "PBEForm" window with populated data. The "INVOICE NO." field contains "PBE01", the "VENDOR" field contains "Maitri Knits", and the "DATE" field contains "2021-10-01". The table below contains the following data:

Product Code	Product Name	CP/Unit	Quantity	Purchase Price
P20101HR	WE Tshirt HR	80.0000	50	4000
P20101HV	WE Tshirt HV	90.0000	50	4000
P20101HP	WE Tshirt HP	140.0000	50	7000
P30101HC	WE Shirt	225.0000	50	11250
			Total Price	26250

A grey "ADD" button is located at the bottom right of the table area. The status bar at the bottom shows the date and time as 31-12-2021 01:16.



## Stock Quantity After Purchase:

Page | 159

MainWindow

Menu Section

Vendor

**Product**

Purchase Bill Entry

Sale Acceptance Entry

Return Sale Entry

Inventory Report

Report

User

Category

Logout

PRODUCT

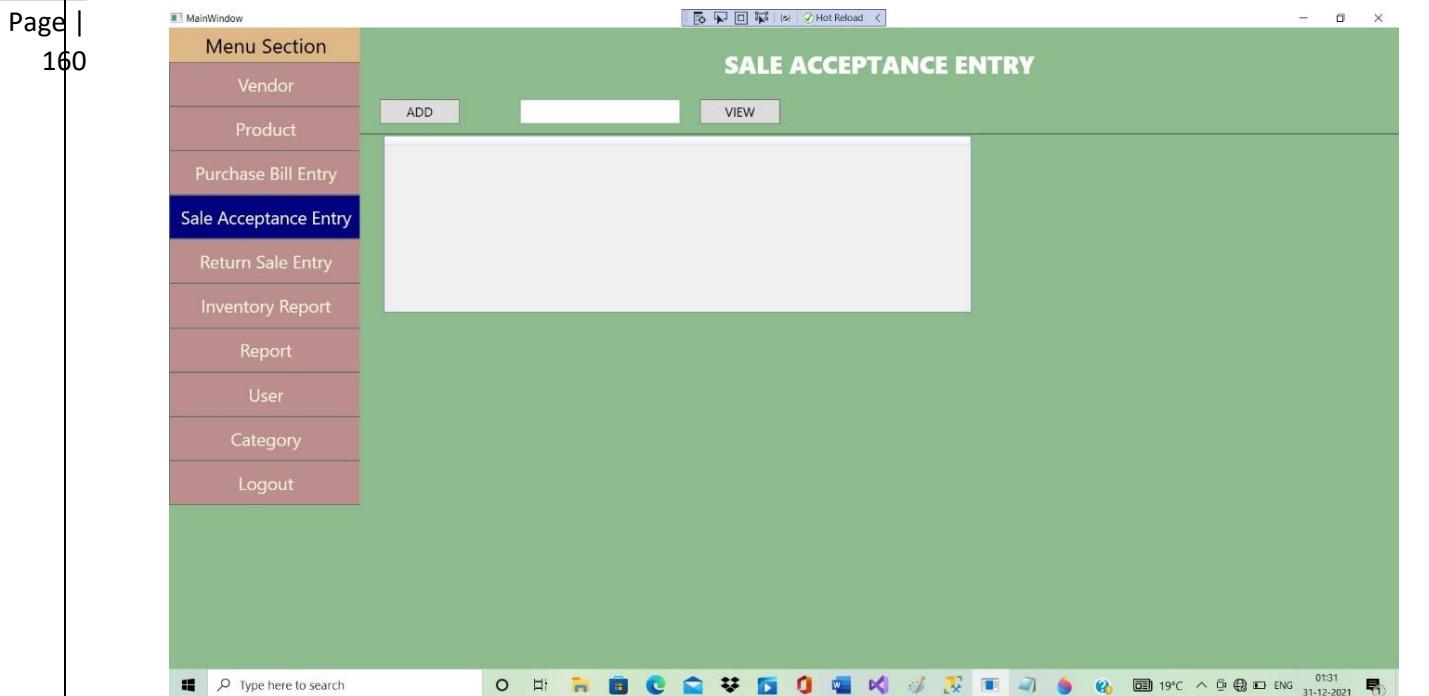
Product Code	Product Name	Description	Category	Unit	CPUnit	SPUnit	Buffer Quantity	Stock Quantity	DOG
P20101HR	WE Tshirt HR	Tshirt Half Sleeve Round Neck	1	Nos	80.0000	140.0000	20	50	9/20/2021 12:00:00 AM
P20101HV	WE Tshirt HV	Half Sleeve V-Neck	1	Nos	90.0000	160.0000	30	50	9/20/2021 12:00:00 AM
P20101HP	WE Tshirt HP	Half Tshirt Polo Neck	1	Nos	140.0000	225.0000	30	50	9/20/2021 12:00:00 AM
P30101HC	WE Shirt	Half Sleeve Casual	1002	Nos	225.0000	400.0000	40	50	9/20/2021 12:00:00 AM
P20102HR	NXT Tshirt HR	Half Sleeve Round Neck	1	Nos	100.0000	180.0000	20	0	9/21/2021 12:00:00 AM

Type here to search

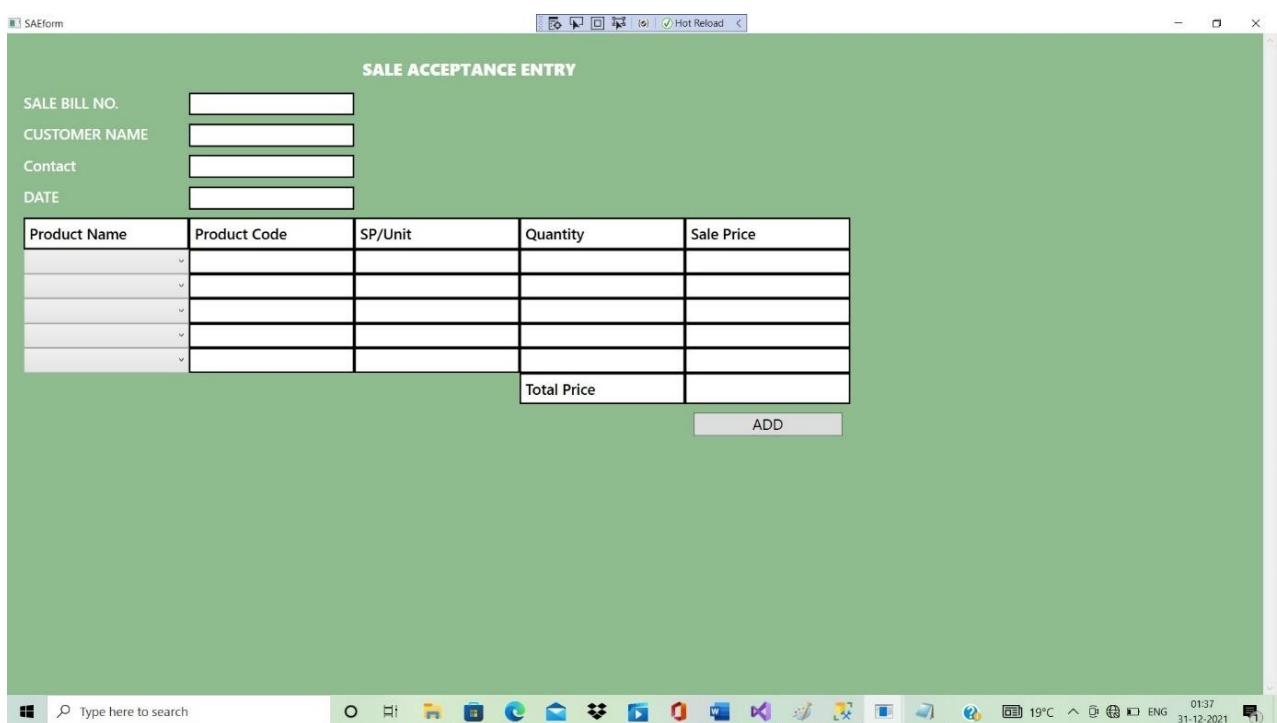
Windows Start Taskbar Icons Date/Time

## SALE ACCEPTANCE ENTRY

- After Successful Login, one can add Sale Bill Details in his database.
- First from MainWindow Dashboard, He need to click Sale Acceptance Entry Button.
- Clicking it opens a SaleAcceptanceEntry Page in its Iframe. On this Page one can view Add Button, A Text Box and a View Button linked to it.



- Clicking on Add Button, will open a Window named SAEform.
- Fill the fields and click the Add Button.
- A message box displaying message “Sale Bill Successfully Added” will appears confirmed the addition of Purchase Bill Details to database.
- On SaleAcceptanceEntry Page one can type Sale No. in Text box and click View Button to see details of particular sale.



SAEform

### SALE ACCEPTANCE ENTRY

SALE BILL NO.	SBE01
CUSTOMER NAME	Sidharth Raha
Contact	9031214031
DATE	2021-10-02

Product Name	Product Code	SP/Unit	Quantity	Sale Price
WE Tshirt HR	P20101HR	140.0000	2	280
WE Tshirt HV	P20101HV	160.0000	2	320
WE Tshirt HP	P20101HP	225.0000	2	450
WE Shirt	P30101HC	400.0000	2	800
			Total Price	1850

**ADD**

SAEform

### SALE ACCEPTANCE ENTRY

SALE BILL NO.	SBE01
CUSTOMER NAME	Sidharth Raha
Contact	9031214031
DATE	2021-10-02

Product Name	Product Code	SP/Unit	Quantity	Sale Price
WE Tshirt HR	P20101HR	140.0000	2	280
WE Tshirt HV	P20101HV	160.0000	2	320
WE Tshirt HP	P20101HP	225.0000	2	450
WE Shirt	P30101HC	400.0000	2	
			Total Price	

**OK** **ADD**

Sale Bill Entry is Successful.

MainWindow

Menu Section

- Vendor
- Product
- Purchase Bill Entry
- Sale Acceptance Entry**
- Return Sale Entry
- Inventory Report
- Report
- User
- Category
- Logout

### SALE ACCEPTANCE ENTRY

**ADD** SBE01 **VIEW**

Sale No	SBID	CustomerName	CustomerContact	ProductCode	QuantitySold	SaleDOE
SBE01	10102	Sidharth Raha	9031214031	P20101HR	2	10/2/2021 12:00:00 A
SBE01	10104	Sidharth Raha	9031214031	P20101HV	2	10/2/2021 12:00:00 A
SBE01	10106	Sidharth Raha	9031214031	P20101HP	2	10/2/2021 12:00:00 A
SBE01	10108	Sidharth Raha	9031214031	P30101HC	2	10/2/2021 12:00:00 A

< >

SAEform

Type here to search

19°C ENG 01:53 31-12-2021

## Stock Quantity After Sale:

Page | 162

MainWindow

Menu Section

Vendor

**Product**

Purchase Bill Entry

Sale Acceptance Entry

Return Sale Entry

Inventory Report

Report

User

Category

Logout

ADD

EDIT

PRODUCT

Product Code	Product Name	Description	Category	Unit	CPUnit	SPUnit	Buffer Quantity	Stock Quantity	DOG
P20101HR	WE Tshirt HR	Tshirt Half Sleeve Round Neck	1	Nos	80.0000	140.0000	20	48	9/20/2021 12:00:00 AM
P20101HV	WE Tshirt HV	Half Sleeve V-Neck	1	Nos	90.0000	160.0000	30	48	9/20/2021 12:00:00 AM
P20101HP	WE Tshirt HP	Half Tshirt Polo Neck	1	Nos	140.0000	225.0000	30	48	9/20/2021 12:00:00 AM
P30101HC	WE Shirt	Half Sleeve Casual	1002	Nos	225.0000	400.0000	40	48	9/20/2021 12:00:00 AM
P20102HR	NXT Tshirt HR	Half Sleeve Round Neck	1	Nos	100.0000	180.0000	20	0	9/21/2021 12:00:00 AM

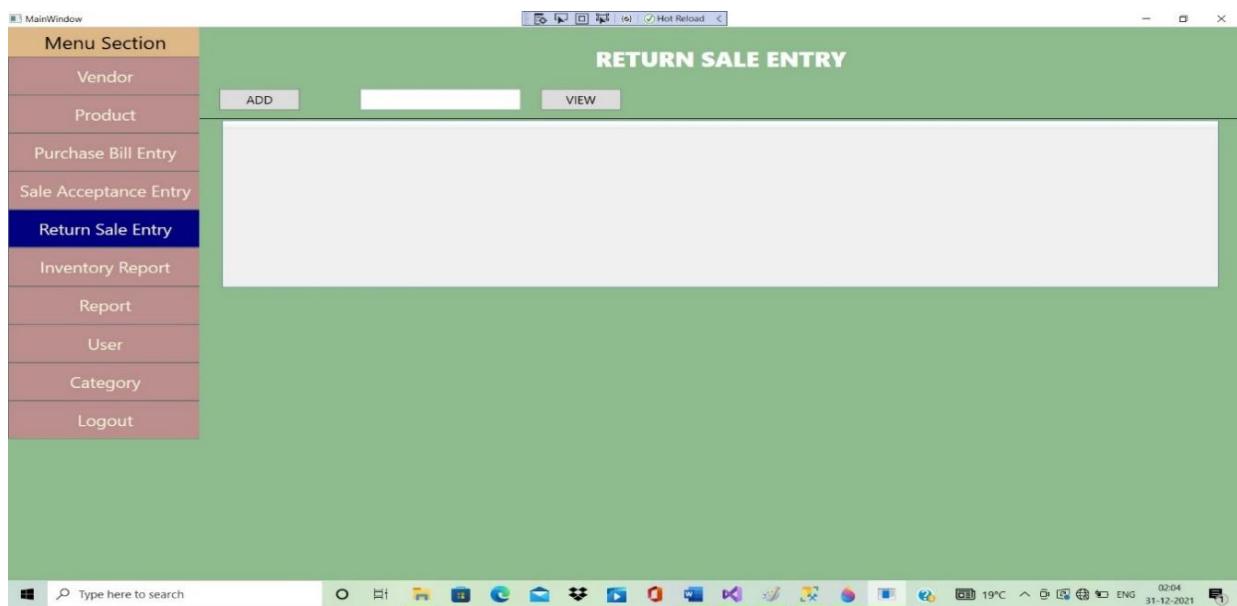
Type here to search

19°C 02:11 31-12-2021

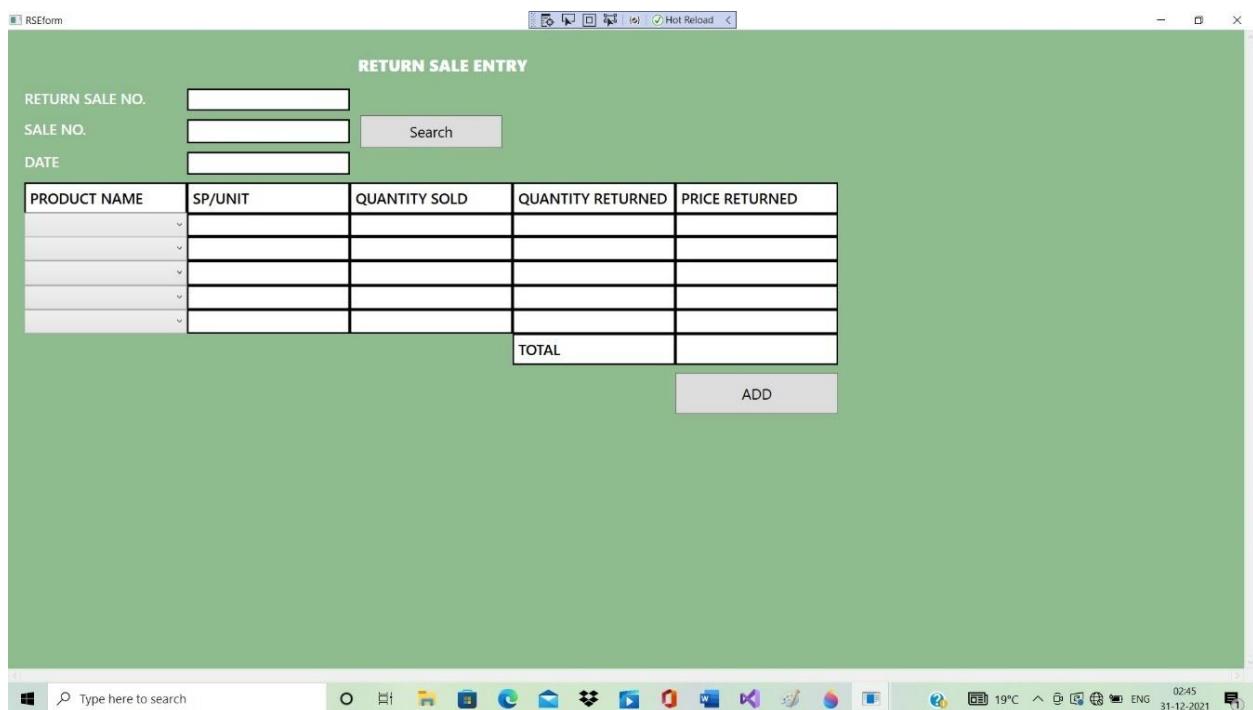
## Return Sale Entry

- After Successful Login, one can add Return Sale Bill Details in his database.  
• First from MainWindow Dashboard, He need to click Return Sale Entry Button.  
• Clicking it opens a ReturnSaleEntry Page in its Iframe. On this Page one can view Add Button, A Text Box and a View Button linked to it.

Page |  
163



- Clicking on Add Button, will open a Window named RSEform.
- Fill the fields and click the Add Button.
- A message box displaying message “Return Sale Bill Successfully Added” will appears confirmed the addition of Purchase Bill Details to database.
- On ReturnSaleEntry Page one can type Return Sale No. in Text box and click View Button to see details of particular return.



Page |  
164

RSEform

**RETURN SALE ENTRY**

RETURN SALE NO.	RSE01			
SALE NO.	SBE01	<input type="button" value="Search"/>		
DATE	2021-10-05			
PRODUCT NAME	SP/UNIT	QUANTITY SOLD	QUANTITY RETURNED	PRICE RETURNED
WE Shirt	400.0000	2	2	800
WE Tshirt HR	140.0000	2	1	140
		TOTAL		940
<input type="button" value="ADD"/>				

Type here to search

RSEform

RETURN SALE ENTRY

RETURN SALE NO.	RSE01
SALE NO.	SBE01
DATE	2021-10-05

Search

PRODUCT NAME	SP/UNIT	QUANTITY SOLD	QUANTITY RETURNED	PRICE RETURNED
WE Shirt	400.0000	2	2	800
WE Tshirt HR	140.0000	2	1	140

X

TOTAL

Return Bill Entry Successful.

OK ADD

Type here to search

0 11: 19°C ENG 31-12-2021 02:51

The screenshot shows a Windows application window titled "RETURN SALE ENTRY". The window has a green header bar with standard window controls (minimize, maximize, close) and a toolbar with icons for file operations like Open, Save, Print, and Hot Reload.

The main area contains a table with the following columns: Return Sale No, RSID, Sale No, ProductCode, QuantityReturn, and ReturnDOE. There are two rows of data:

Return Sale No	RSID	Sale No	ProductCode	QuantityReturn	ReturnDOE
RSE01	8102	SBE01	P30101HC	2	10/3/2021 12:00:00 AM
RSE01	8104	SBE01	P20101HR	1	10/3/2021 12:00:00 AM

On the left side, there is a vertical navigation menu bar with the following items:

- Main Window
- Menu Section
- Vendor
- Product
- Purchase Bill Entry
- Sale Acceptance Entry
- Return Sale Entry** (This item is highlighted in blue)
- Inventory Report
- Report
- User
- Category
- Logout

The status bar at the bottom displays system information: 19°C, ENG, 21.12.2021, and 02:21.

## Stock Quantity After Return Sale:

Page | 165

Product Code	Product Name	Description	Category	Unit	CPUnit	SPUnit	Buffer Quantity	Stock Quantity	DOG
P20101HR	WE Tshirt HR	Tshirt Half Sleeve Round Neck	1	Nos	80.0000	140.0000	20	49	9/20/2021 12:00:00 AM
P20101HV	WE Tshirt HV	Half Sleeve V-Neck	1	Nos	90.0000	160.0000	30	48	9/20/2021 12:00:00 AM
P20101HP	WE Tshirt HP	Half Tshirt Polo Neck	1	Nos	140.0000	225.0000	30	48	9/20/2021 12:00:00 AM
P30101HC	WE Shirt	Half Sleeve Casual	1002	Nos	225.0000	400.0000	40	50	9/20/2021 12:00:00 AM
P20102HR	NXT Tshirt HR	Half Sleeve Round Neck	1	Nos	100.0000	180.0000	20	0	9/21/2021 12:00:00 AM

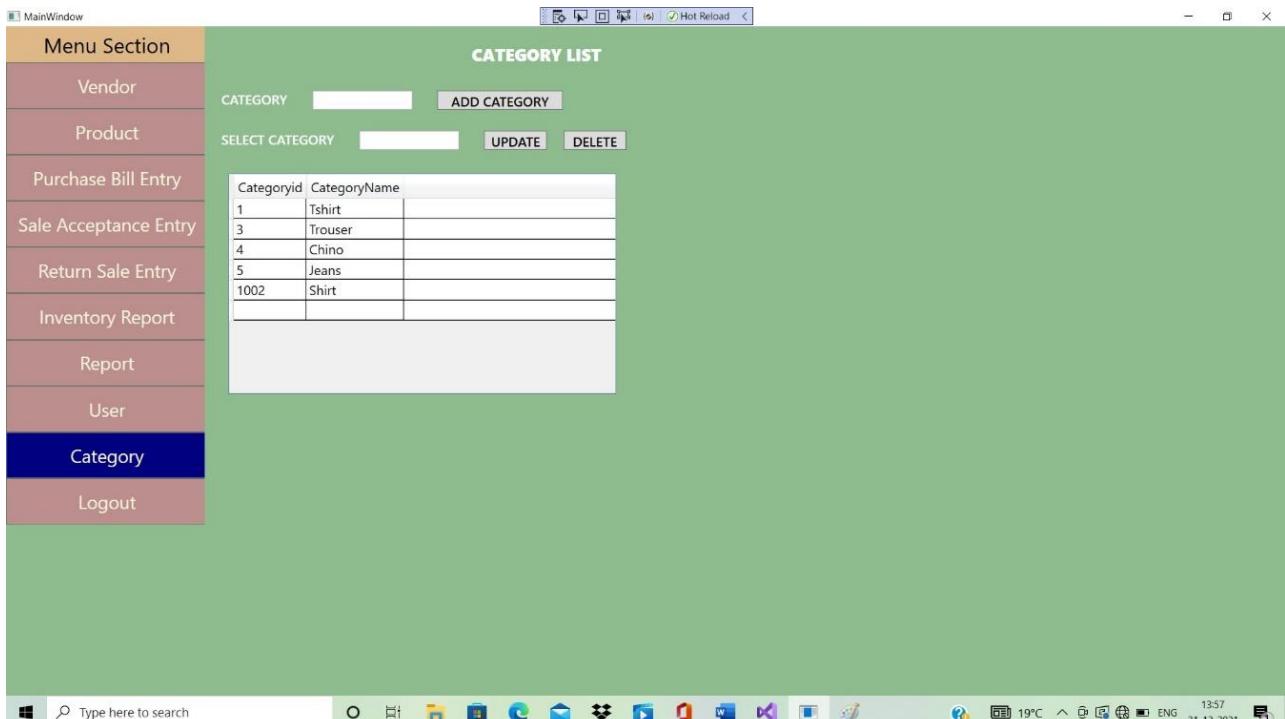
## Adding, Viewing and Editing User

- After successful login, in MainWindow Window Menu Section, Click User Button.
- Clicking User will open a UserList Page in MainWindow iFrame. This Page consist of "Username" Textbox, "Password" Textbox and Add Button. It also consists of "Username" Textbox, Update and Delete Button. A Message Box with message appears "New User is added Successfully.".
- Fill "Username" Textbox and "Password" Textbox and click Add Button. A New Username and Password is created, which can be seen in DataGrid of page.
- Type Username in 2<sup>nd</sup> Textbox and click Delete Button to delete Username and Password. A Message Box with message appears "Record has been deleted Successfully.".
- Type Username whose details you want to edit in 2<sup>nd</sup> row and type new Username and New Password in First Row Textbox and Click Update to Edit username and password. A Message Box with message appears "Record has been updated Successfully.".

Id	Username	password
5	123	123
6	sid	nlk
3002	1	1

## Adding, Viewing and Editing Category

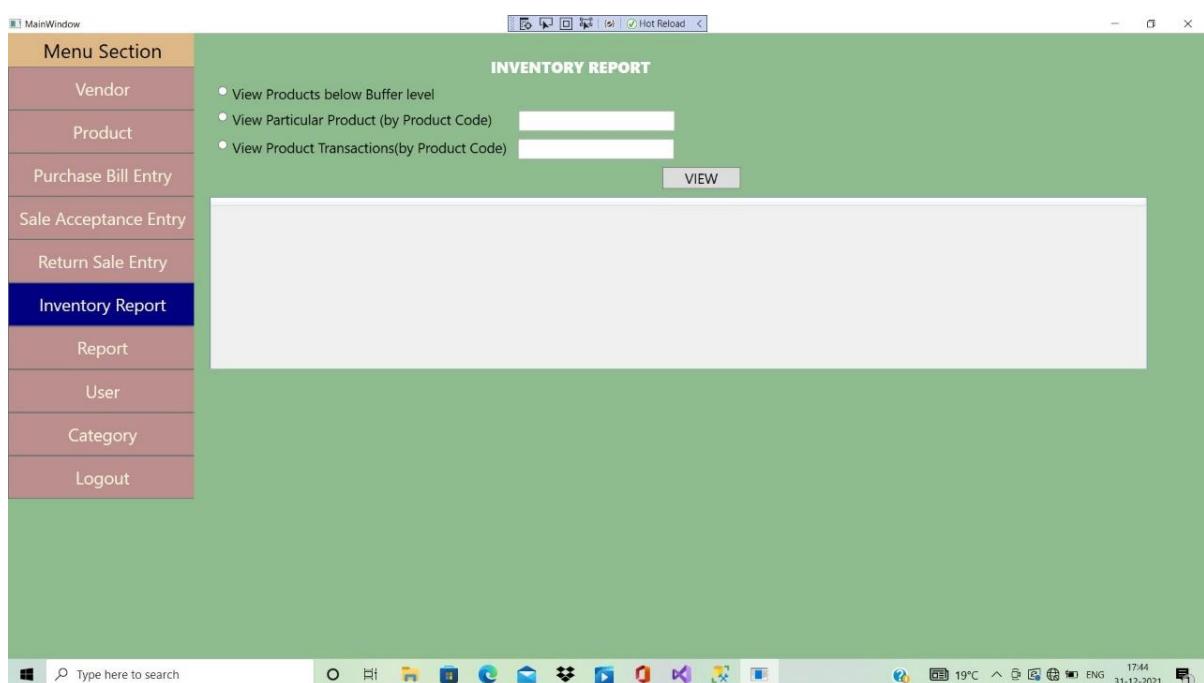
- After successful login, in MainWindow Window Menu Section, Click Category Button.
- Clicking User will open a Category Page in MainWindow iFrame. This Page consist of “Category” Textbox, and Add Button. It also consists of “Select Category” Textbox, Update and Delete Button.
- Fill “Category” Textbox and click Add Button. A New Category is created, which can be seen in DataGrid of page. A Message Box with message appears “New Category is added Successfully.”.
- Type Category Name which you want to edit in “Select Category” Textbox and Type new Updated Category Name in 1<sup>st</sup> Category Textbox and Click Update Button. A Message Box with message appears “Record has been updated Successfully.”.
- Type Category Name in “Select Category” Textbox and Click Delete Button to delete category. A Message Box with message appears “Record has been deleted Successfully.”.



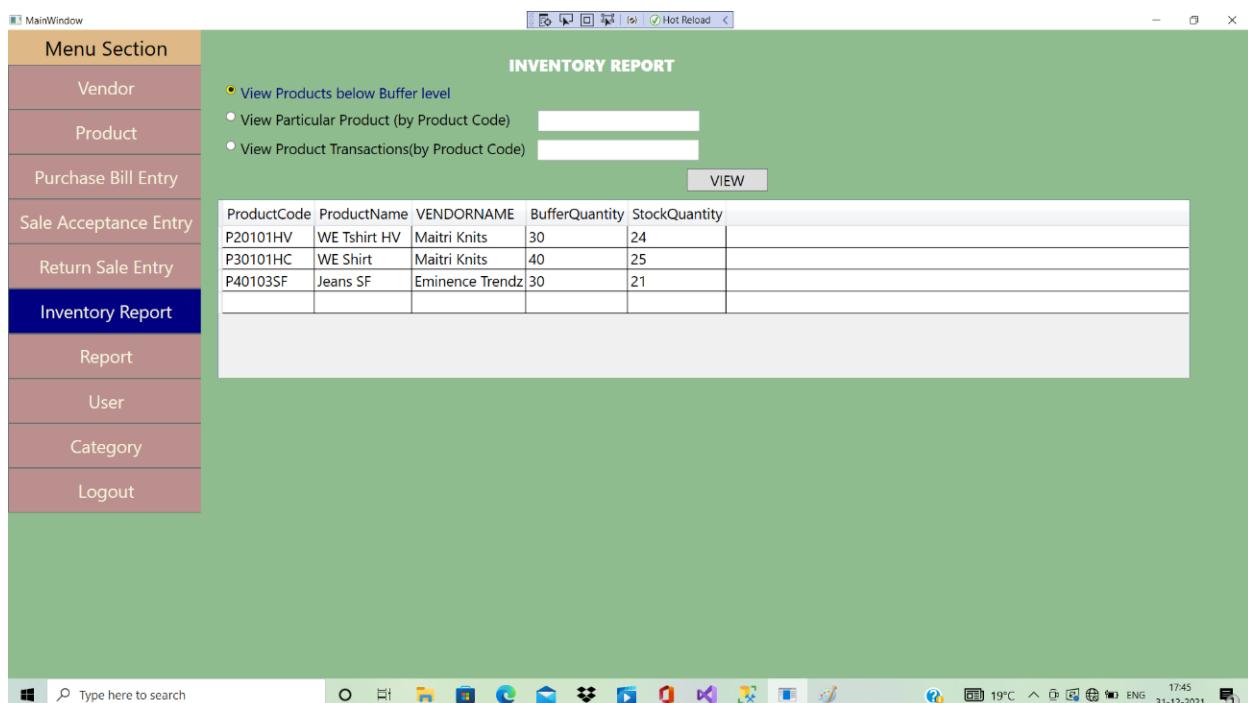
## VIEWING INVENTORY REPORT

- After successful Login, click Inventory report Button in Menu Section of MainWindow Window.
- Clicking it, will open an Inventory Page where there will be three Radio Button (1. To view Products which are less than Buffer Stock 2. View details of any product-by-Product Code 3. View Transaction For any Product-by-Product Code), a View Button a Data Grid to display Data

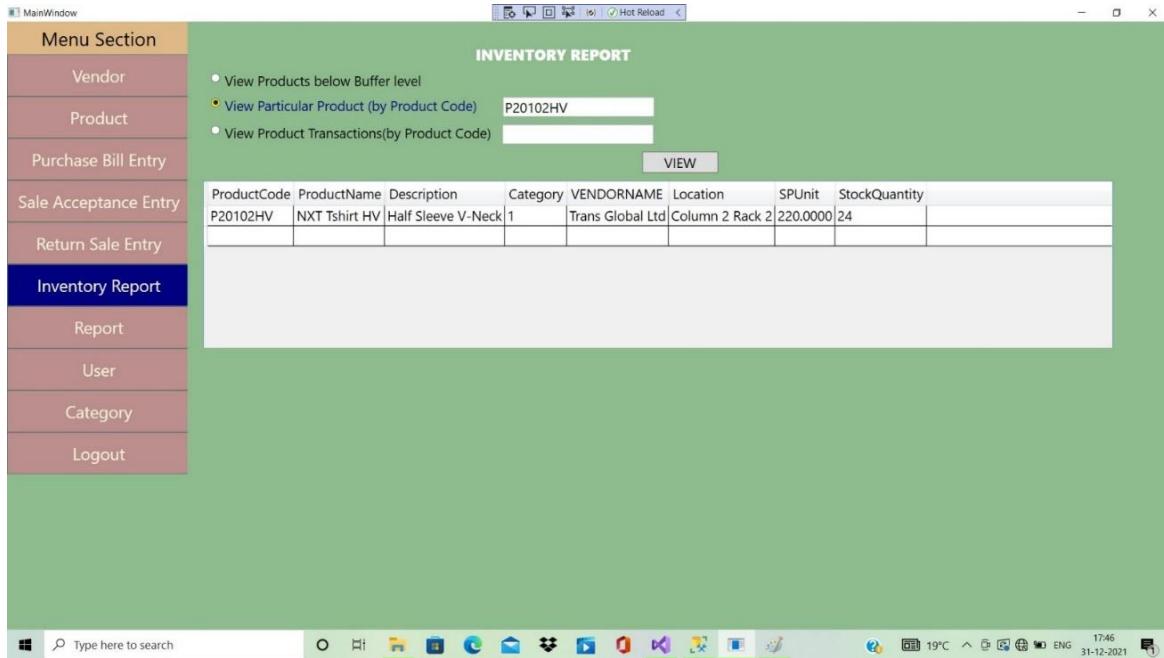
Page |  
167



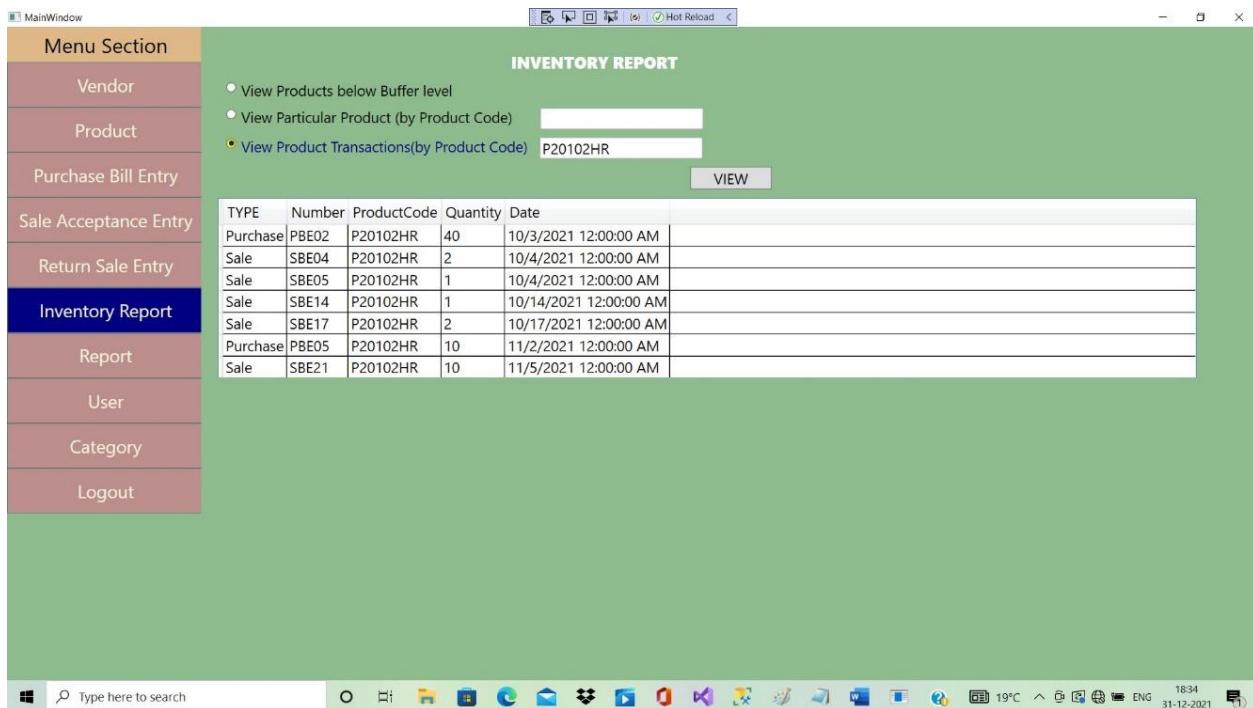
Click 1<sup>st</sup> Radio Button to view Product below buffer level. Radio Button and text color changes and data is loaded in Data Grid.



Click 2<sup>nd</sup> Radio Button to view details of particular product searching by Product Code Entered in Textbox beside it,

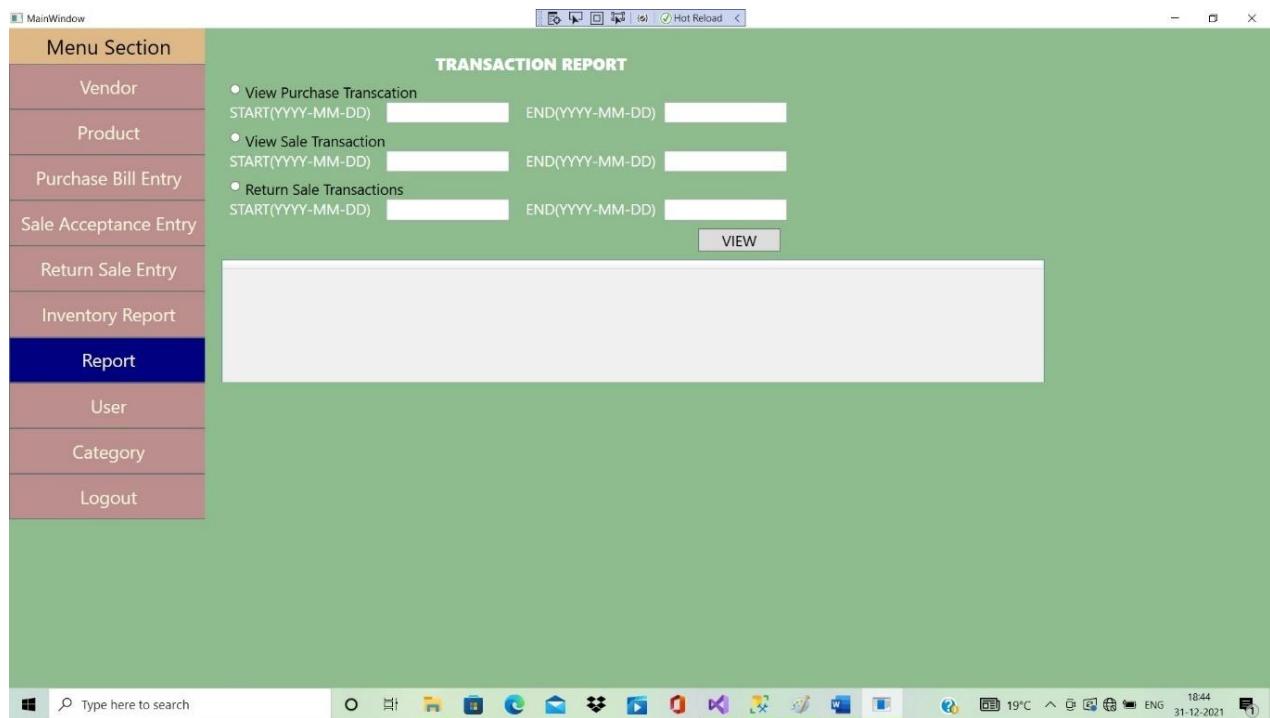


- Click 3rd Radio Button to view Transactions related to any particular Product by Product Code entered in Textbox beside it.

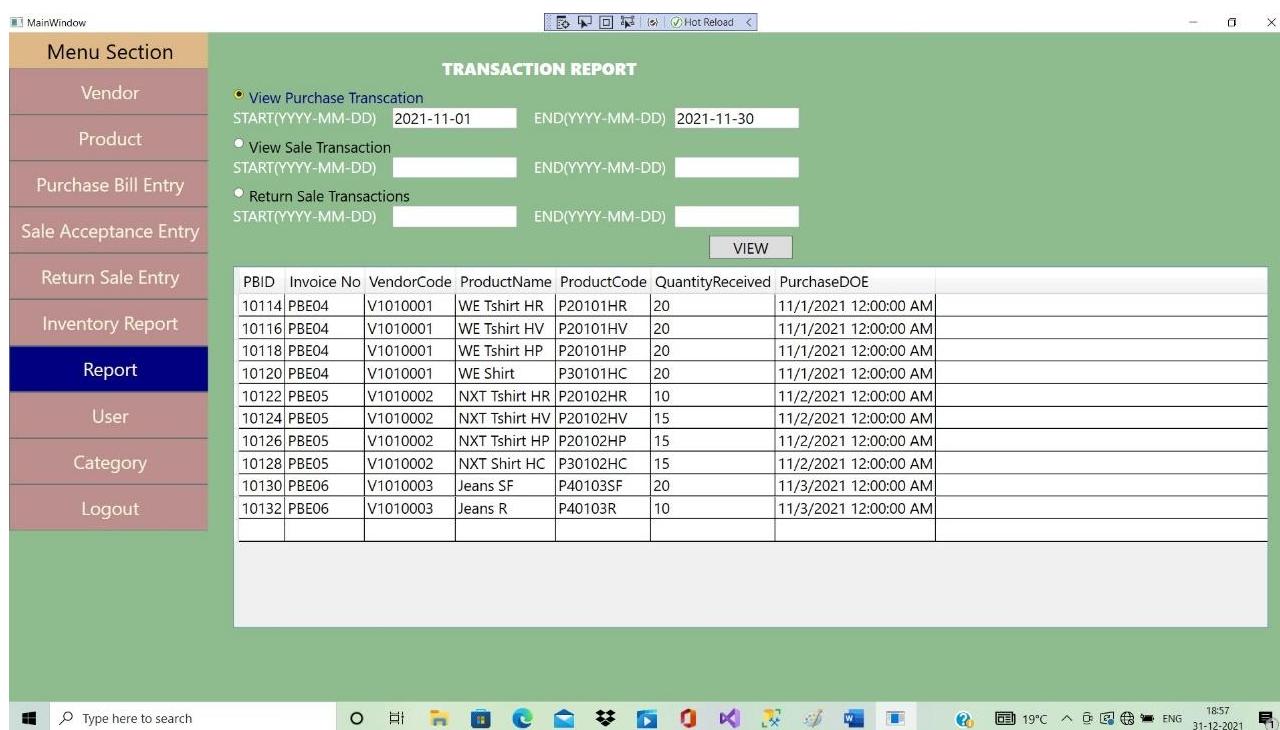


## Viewing Report

- After Successful Login, Report Button in Menu Section of MainWindow Window.
- Clicking it, will open an Inventory Page where there will be three Radio Button (1. To view Purchase Transactions 2. To view Sale Transactions 3. To view Return Sale Transaction), pair of Textbox set



Click 1<sup>st</sup> Radio Button and Enter Start and End Text Box. Radio Button and its associated text color changes. And then press View Button. Purchase Transactions between entered dates appear in DataGrid.

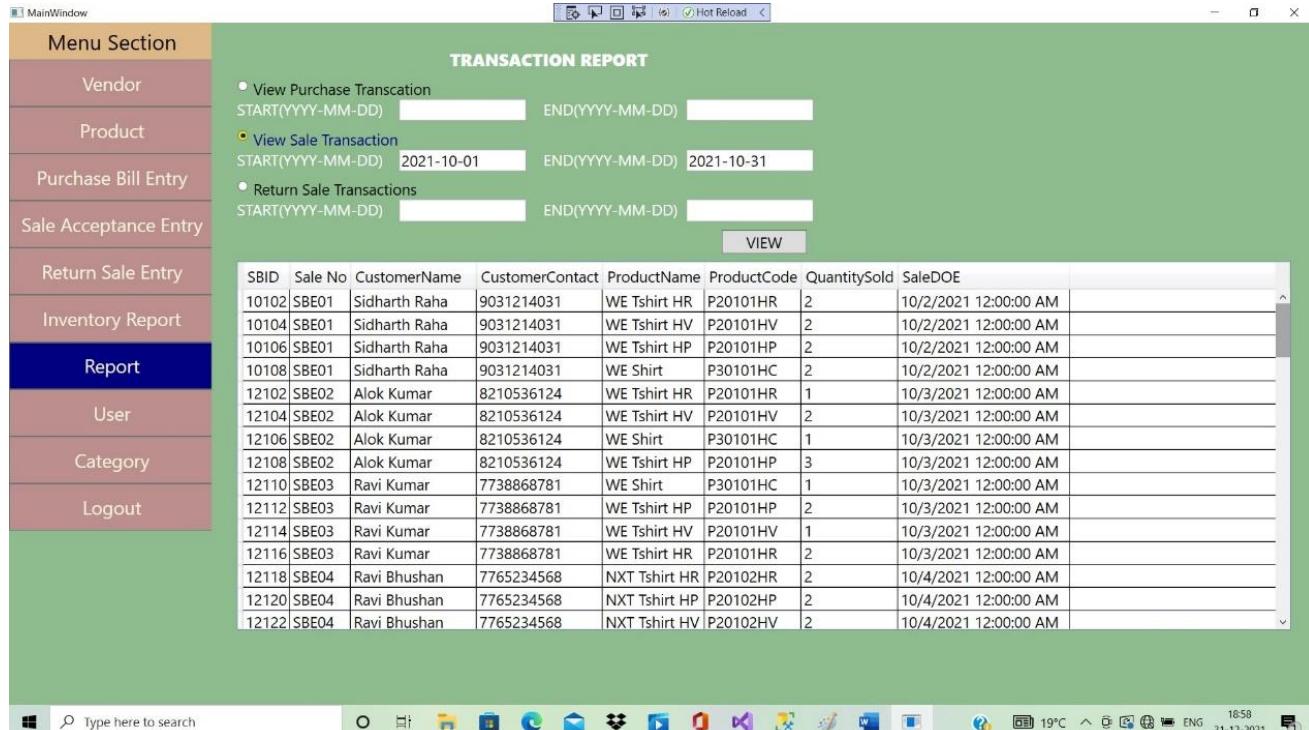


Page | 170

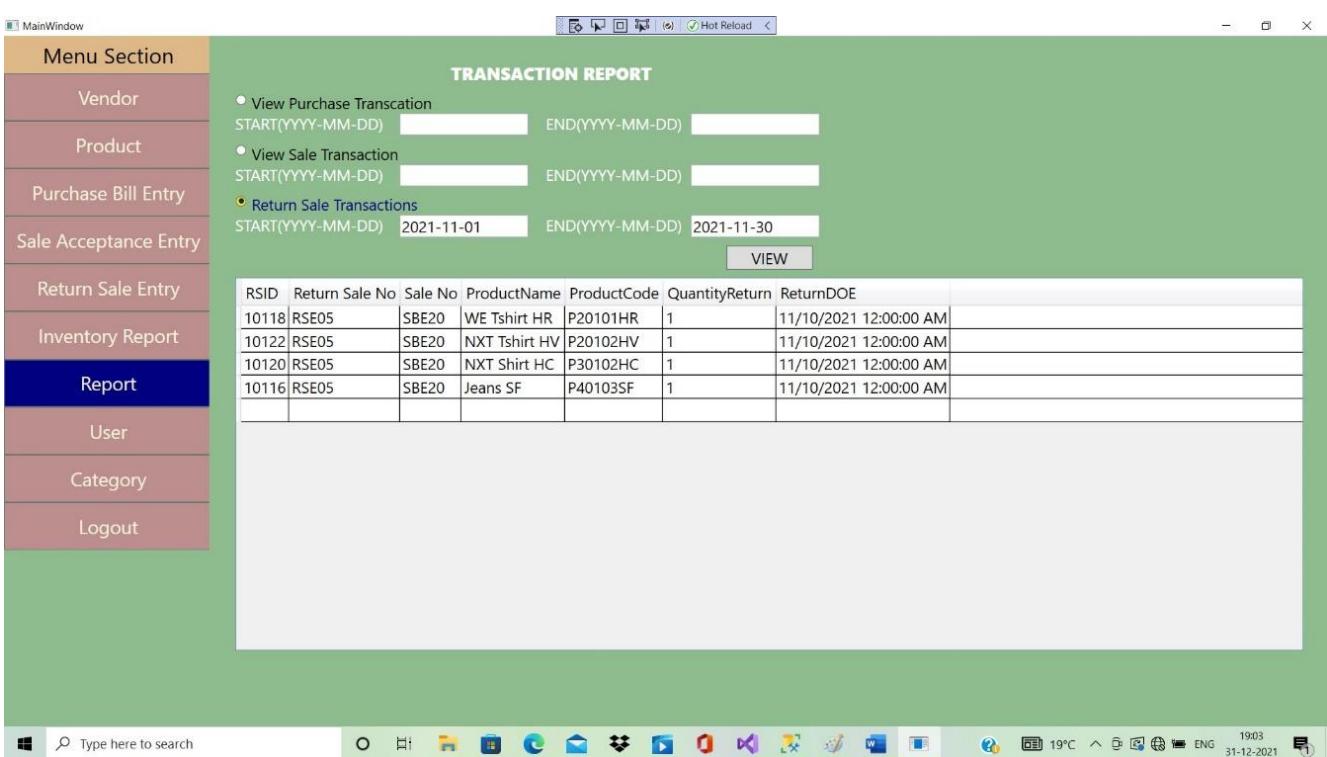
PBID	Invoice No	VendorCode	ProductName	ProductCode	QuantityReceived	PurchaseDOE
8102	PBE01	V1010001	WE Tshirt HR	P20101HR	50	10/1/2021 12:00:00 AM
8104	PBE01	V1010001	WE Tshirt HV	P20101HV	50	10/1/2021 12:00:00 AM
8106	PBE01	V1010001	WE Tshirt HP	P20101HP	50	10/1/2021 12:00:00 AM
8108	PBE01	V1010001	WE Shirt	P30101HC	50	10/1/2021 12:00:00 AM
10102	PBE02	V1010002	NXT Tshirt HR	P20102HR	40	10/3/2021 12:00:00 AM
10104	PBE02	V1010002	NXT Tshirt HV	P20102HV	30	10/3/2021 12:00:00 AM
10106	PBE02	V1010002	NXT Tshirt HP	P20102HP	30	10/3/2021 12:00:00 AM
10108	PBE02	V1010002	NXT Shirt HC	P30102HC	30	10/3/2021 12:00:00 AM
10110	PBE03	V1010003	Jeans SF	P40103SF	50	10/5/2021 12:00:00 AM
10112	PBE03	V1010003	Jeans R	P40103R	50	10/5/2021 12:00:00 AM

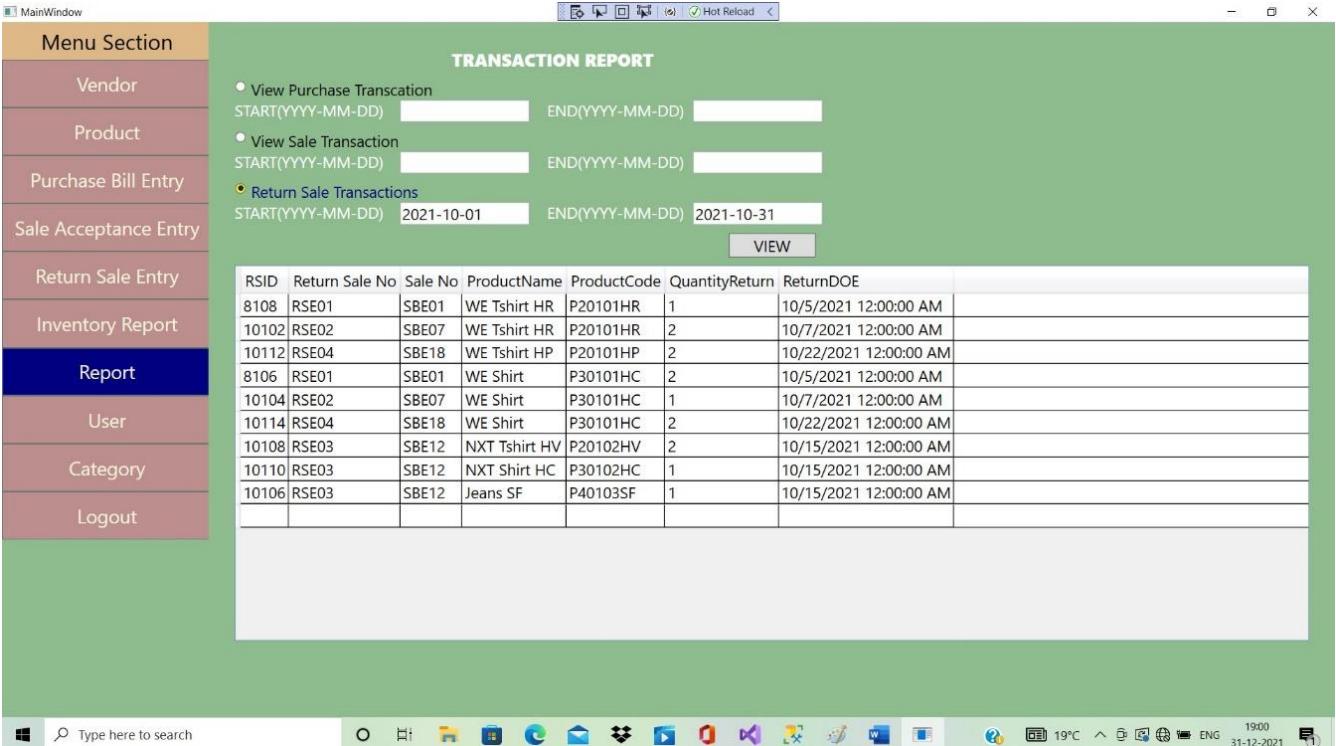
Click 2nd Radio Button and Enter Start and End Text Box. Radio Button and its associated text color changes. And then press View Button. Sale Transactions between entered dates appear in DataGrid.

SBID	Sale No	CustomerName	CustomerContact	ProductName	ProductCode	QuantitySold	SaleDOE
12262	SBE20	Sanjeet Kumar	7875565378	Jeans SF	P40103SF	4	11/4/2021 12:00:00 AM
12264	SBE20	Sanjeet Kumar	7875565378	WE Tshirt HR	P20101HR	4	11/4/2021 12:00:00 AM
12266	SBE20	Sanjeet Kumar	7875565378	NXT Shirt HC	P30102HC	5	11/4/2021 12:00:00 AM
12268	SBE20	Sanjeet Kumar	7875565378	NXT Tshirt HV	P20102HV	5	11/4/2021 12:00:00 AM
12270	SBE21	Abdul	9678533263	Jeans R	P40103R	10	11/5/2021 12:00:00 AM
12272	SBE21	Abdul	9678533263	Jeans SF	P40103SF	10	11/5/2021 12:00:00 AM
12274	SBE21	Abdul	9678533263	NXT Tshirt HR	P20102HR	10	11/5/2021 12:00:00 AM
12276	SBE21	Abdul	9678533263	WE Tshirt HV	P20101HV	10	11/5/2021 12:00:00 AM
12278	SBE21	Abdul	9678533263	WE Shirt	P30101HC	10	11/5/2021 12:00:00 AM
12280	SBE22	Sandeep Kumar	6207567856	WE Tshirt HR	P20101HR	4	11/6/2021 12:00:00 AM
12282	SBE22	Sandeep Kumar	6207567856	WE Tshirt HP	P20101HP	2	11/6/2021 12:00:00 AM
12284	SBE22	Sandeep Kumar	6207567856	WE Tshirt HV	P20101HV	3	11/6/2021 12:00:00 AM
12286	SBE22	Sandeep Kumar	6207567856	Jeans SF	P40103SF	2	11/6/2021 12:00:00 AM
12288	SBE22	Sandeep Kumar	6207567856	Jeans R	P40103R	2	11/6/2021 12:00:00 AM
12290	SBE23	Sanjeev Kumar	6783435521	Jeans R	P40103R	2	11/11/2021 12:00:00 AM



Click 3rd Radio Button and Enter Start and End Text Box. Radio Button and its associated text color changes. And then press View Button. Purchase Transactions between entered dates appear in DataGrid.





## LOGOUT

- After successful Login, and doing business Click on the Logout Button,
- On Clicking it will take user to Login Screen terminating all other Screens.

## **10.0 IMPLEMENTATION OF SECURITY**

Security is provided in this software through providing access to various Functionality of software only after successful Login. Only Persons with valid Username and Password would be able to login and use the functionality of the software. The Password cannot be seen while typing.

## **11.0 LIMITAION OF SOFTWARE**

- Reports or Bills cannot be generated in pdf or Print Format.
- Pictorial View of Data could be Seen.
- Password is not Encrypted.

## **12.0 FUTURE APPLICATION OF THE PROJECT**

- This project can be enhanced to develop a real-life product for shops, restaurants, malls, industries etc.
- Functionality to generate pdf and print format for reports and bills.
- Functionality to integrate e-mail system.
- Functionality to generate proforma invoice.
- More Security Features.
- More analytics and pictorial view of Data.

## 13.0 BIBLIOGRAPHY

Making this Project was a great learning experience. I got to know practicality of how we use programming to develop a software and got to experience whole phase of SDLC (Software Development Life Cycle). I would like to thanks my Project Guide Sidharth Raha, for being my source of information. Other than that, I referenced many websites and YouTube channel to get various information. Here is a list:

- <https://www.c-sharpcorner.com>
- <https://stackoverflow.com>
- <https://docs.microsoft.com>wpf>
- <https://www.tutorialspoint.com>wpf>
- YouTube: tutorialEU
- YouTube: AngelSix
- YouTube: DotNetSkoool
- YouTube: IAmTimCorrey