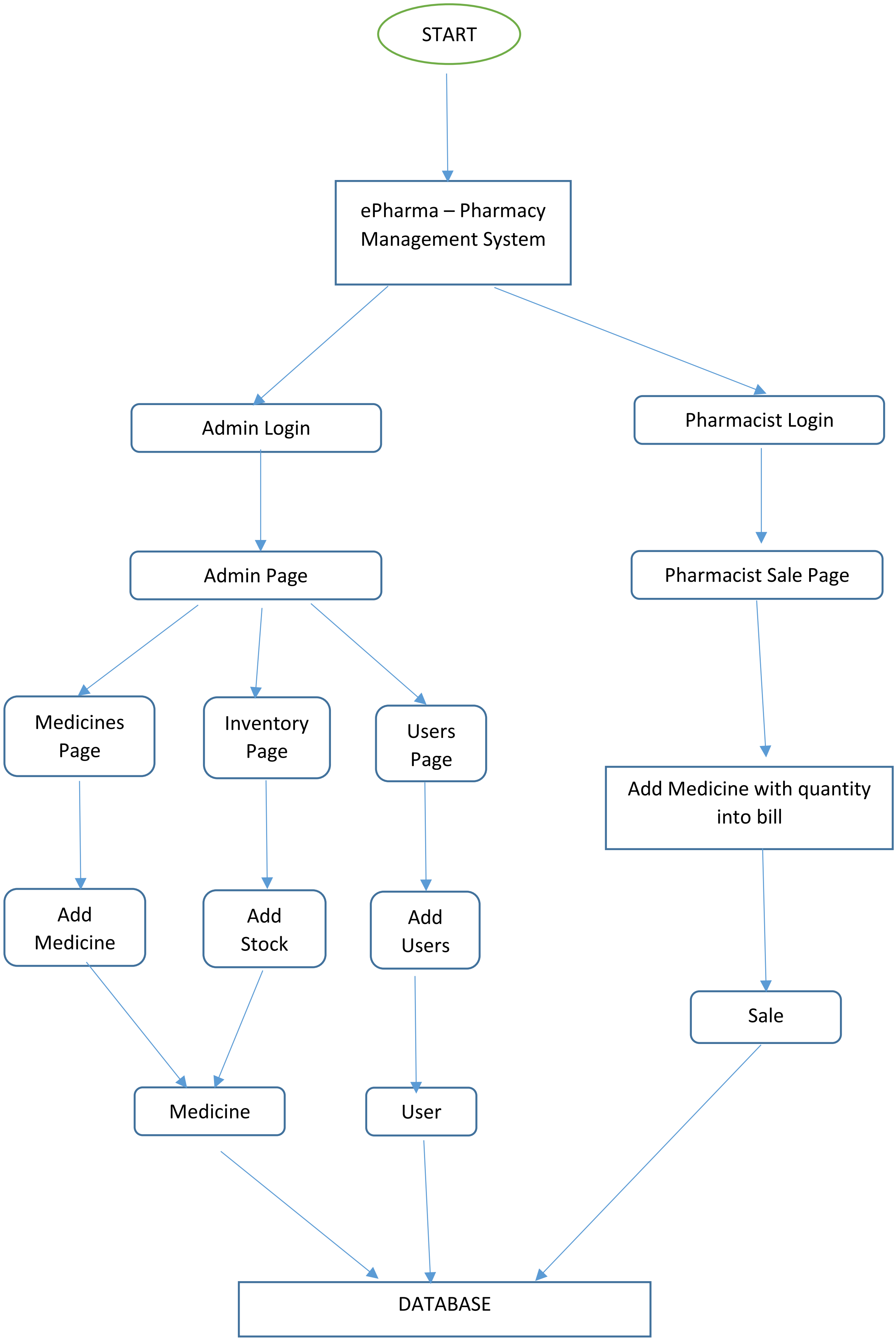
SYSTEM DESIGN

INDRODUCTION TO SYSTEM DESIGN

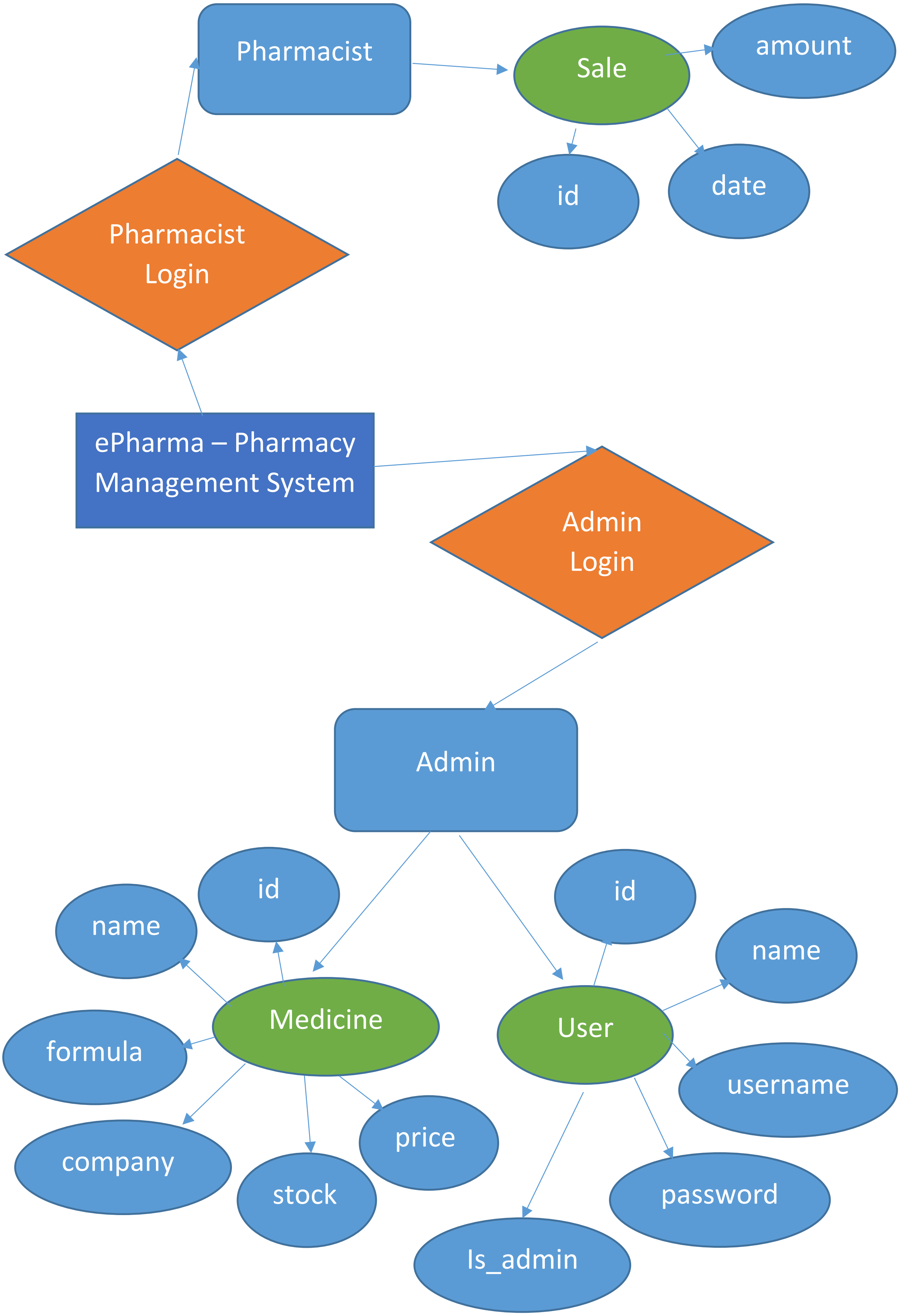
The design phase is the life cycle phase in which the detailed design of the selected system in the study phase is accomplishd . In the design phase, the technical specifications are prepared for the performed of all allocated tasks. It also includes the construction of programs and programs testing. In the design phase, the first step is to determine the output is to produced and in what format.

Systems design is the process of defining elements of a system like modules, architecture, components and their interfaces and data for a system based on the specified requirements. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization

DATA FLOW DIAGRAM


START
ePharma – Pharmacy Management System
Admin Login
Pharmacist Login
Admin Page
Pharmacist Sale Page
Medicines Page
Inventory Page
Users Page
Add Medicine with quantity into bill
Add Medicine
Add Stock
Add Users
Sale
Medicine
User
DATABASE

ER DIAGRAM


Pharmacist
Sale
amount
id
date
Pharmacist Login
ePharma – Pharmacy Management System
Admin Login
Admin
name
id
id
name
Medicine
User
formula
username
company
price
stock
password
Is_admin

DATABASE TABLES

User Table (users)

Column Name	Data Type
id	Int
name	varchar
username	varchar
password	varchar
Is_admin	bool

Medicine Table (medicines)

Column Name	Data Type
id	int
name	varchar
formula	varchar
company	varchar
price	int
stock	int

Sale Table (sales)

Column Name	Data Type
id	int
date	datetime
amount	int

CODE

Main Form - Form1.vb

```
Imports MySql.Data.MySqlClient

Public Class Form1

    Public LoggedInUserName As String

    Public DbConnection As MySqlConnection

    Dim PageAddAdmin As AddAdmin
    Dim PageAdminLogin As AdminLogin
    Public PageAdmin As Admin

    Dim PagePharmacistLogin As PharmacistLogin
    Public PagePharmacist As Pharmacist

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        InitDbConnection()
        LoadInitailPage()
    End Sub

    Private Sub InitDbConnection()
        Dim ConnectionString =
"server=127.0.0.1;uid=root;pwd=admin;database=epharma"
        DbConnection = New MySqlConnection With {
            .ConnectionString = ConnectionString
        }

        ' Create tables
        DbHelper.CreateTables()
    End Sub

    Private Sub LoadInitailPage()
        UserModule.IsUsersExists()
        If UserModule.UserExists = True Then
            LoadPharmacistLoginPage()
        Else
            LoadAddAdminPage()
        End If
    End Sub

    Public Sub LoadAddAdminPage()
        PageAddAdmin = New AddAdmin With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        PageAddAdmin.Anchor = CType((((System.Windows.Forms.AnchorStyles.Top Or
System.Windows.Forms.AnchorStyles.Bottom) _
            Or System.Windows.Forms.AnchorStyles.Left) _
            Or System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)

        PanelApp.Controls.Clear()
        PanelApp.Controls.Add(PageAddAdmin)

        PageAddAdmin.Show()
    End Sub

    Public Sub LoadAdminLoginPage()
        PageAdminLogin = New AdminLogin With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        PageAdminLogin.Anchor = CType((((System.Windows.Forms.AnchorStyles.Top Or
System.Windows.Forms.AnchorStyles.Bottom) _
            Or System.Windows.Forms.AnchorStyles.Left) _
            Or System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)

        PanelApp.Controls.Clear()
        PanelApp.Controls.Add(PageAdminLogin)

        PageAdminLogin.Show()
    End Sub

    Public Sub LoadAdminPage()
        PageAdmin = New Admin With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        PageAdmin.Anchor = CType((((System.Windows.Forms.AnchorStyles.Top Or
System.Windows.Forms.AnchorStyles.Bottom) _
            Or System.Windows.Forms.AnchorStyles.Left) _
            Or System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)

        PanelApp.Controls.Clear()
        PanelApp.Controls.Add(PageAdmin)
        PageAdmin.Show()
    End Sub

    Public Sub LoadPharmacistLoginPage()
        PagePharmacistLogin = New PharmacistLogin With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        PagePharmacistLogin.Anchor =
CType((((System.Windows.Forms.AnchorStyles.Top Or
System.Windows.Forms.AnchorStyles.Bottom) _
            Or System.Windows.Forms.AnchorStyles.Left) _
            Or System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)

        PanelApp.Controls.Clear()
        PanelApp.Controls.Add(PagePharmacistLogin)
        PagePharmacistLogin.Show()
    End Sub

    Public Sub LoadPharmacistPage()
        PagePharmacist = New Pharmacist With {
            .TopLevel = False,
            .FormBorderStyle = FormBorderStyle.None
        }

        PagePharmacist.Anchor = CType((((System.Windows.Forms.AnchorStyles.Top Or
System.Windows.Forms.AnchorStyles.Bottom) _
            Or System.Windows.Forms.AnchorStyles.Left) _
            Or System.Windows.Forms.AnchorStyles.Right),
System.Windows.Forms.AnchorStyles)
```

```
        PanelApp.Controls.Clear()
        PanelApp.Controls.Add(PagePharmacist)
        PagePharmacist.Show()
    End Sub
End Class
```

Add Admin Page – AddAdmin.vb

```
Public Class AddAdmin
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim Name = NameInput.Text
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        AdminModule.AddAdmin(Name, Username, Password)
    End Sub
End Class
```

Admin Login Page – AdminLogin.vb

```
Public Class AdminLogin
    Private Sub AdminLogin_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Me.SetStyle(ControlStyles.UserPaint, True)
        Me.SetStyle(ControlStyles.AllPaintingInWmPaint, True)
        Me.SetStyle(ControlStyles.OptimizedDoubleBuffer, True)
    End Sub

    Private Sub LoginButton_Click(sender As Object, e As EventArgs) Handles
LoginButton.Click
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        AdminModule.LoginAdmin(Username, Password)
    End Sub

    Private Sub GoToPharmacistLoginButton_Click(sender As Object, e As EventArgs)
Handles GoToPharmacistLoginButton.Click
        Form1.LoadPharmacistLoginPage()
    End Sub
End Class
```

Admin Page – Admin.vb

```
Public Class Admin
    Public LoggedInAdminName As String

    Dim PageAdminDashboard As AdminDashboard

    Dim PageAdminMedicines As AdminMedicines
    Dim PageAdminAddMedicine As AdminAddMedicine

    Dim PageAdminInventory As AdminInventory
    Dim PageAdminAddInventory As AdminAddInventory
    Dim PageAdminUserManagement As AdminUserManagement
    Dim PageAdminAddUser As AdminAddUserPage


    Private Sub Admin_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        AdminNameLabel.Text = Form1.LoggedInUserName
        LoadAdminMedicinesPage()
    End Sub

    Private Sub LoadAdminDashboardPage()
        PageAdminDashboard = New AdminDashboard
        PanelAdminBody.Controls.Clear()
        PanelAdminBody.Controls.Add(PageAdminDashboard)
    End Sub

    Public Sub LoadAdminMedicinesPage()
        PageAdminMedicines = New AdminMedicines
        PanelAdminBody.Controls.Clear()
        PanelAdminBody.Controls.Add(PageAdminMedicines)
    End Sub

    Public Sub LoadAdminAddMedicinePage()
        PageAdminAddMedicine = New AdminAddMedicine
        PanelAdminBody.Controls.Clear()
        PanelAdminBody.Controls.Add(PageAdminAddMedicine)
    End Sub

    Public Sub LoadAdminInventoryPage()
        PageAdminInventory = New AdminInventory
        PanelAdminBody.Controls.Clear()
        PanelAdminBody.Controls.Add(PageAdminInventory)
    End Sub

    Public Sub LoadAdminAddInventoryPage()
        PageAdminAddInventory = New AdminAddInventory
        PanelAdminBody.Controls.Clear()
        PanelAdminBody.Controls.Add(PageAdminAddInventory)
    End Sub

    Public Sub LoadAdminUserManagementPage()
        PageAdminUserManagement = New AdminUserManagement
        PanelAdminBody.Controls.Clear()
        PanelAdminBody.Controls.Add(PageAdminUserManagement)
    End Sub

    Public Sub LoadAdminAddUserPage()
        PageAdminAddUser = New AdminAddUserPage
        PanelAdminBody.Controls.Clear()
        PanelAdminBody.Controls.Add(PageAdminAddUser)
    End Sub

    Private Sub DashboardLink_Click_1(sender As Object, e As EventArgs)
        LoadAdminDashboardPage()
    End Sub

    Private Sub MedicinesLink_Click(sender As Object, e As EventArgs) Handles
MedicinesLink.Click
        LoadAdminMedicinesPage()
    End Sub

    Private Sub InventoryLink_Click(sender As Object, e As EventArgs) Handles
InventoryLink.Click
        LoadAdminInventoryPage()
    End Sub

    Private Sub UsersLink_Click_1(sender As Object, e As EventArgs) Handles
UsersLink.Click
        LoadAdminUserManagementPage()
    End Sub

    Private Sub LogoutButton_Click(sender As Object, e As EventArgs) Handles
LogoutButton.Click
        Form1.LoggedInUserName = ""
        Form1.LoadPharmacistLoginPage()
    End Sub

    Private Sub Panel1_Paint(sender As Object, e As PaintEventArgs) Handles
Panel1.Paint

    End Sub
End Class
```

Admin Medicines Page – AdminMedicines.vb

```
Public Class AdminMedicines
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.PageAdmin.LoadAdminAddMedicinePage()
    End Sub

    Private Sub AdminMedicines_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        MedicineModule.LoadMedicinesData()
        Dim CurrentItem = 0
        For Each Medicine As MedicineModel In MedicineModule.Medicines
            MedicineDetailsListView.Items.Add(Medicine.Id)

MedicineDetailsListView.Items(CurrentItem).SubItems.Add(Medicine.Name)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Medicine As MedicineModel In MedicineModule.Medicines

MedicineDetailsListView.Items(CurrentItem).SubItems.Add(Medicine.Formula)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Medicine As MedicineModel In MedicineModule.Medicines

MedicineDetailsListView.Items(CurrentItem).SubItems.Add(Medicine.Company)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Medicine As MedicineModel In MedicineModule.Medicines

MedicineDetailsListView.Items(CurrentItem).SubItems.Add(Medicine.Price)
            CurrentItem += 1
        Next
    End Sub
End Class
```

Admin Add Medicine Page – AdminAddMedicine.vb

```
Public Class AdminAddMedicine
    Private Sub GoBackButton_Click(sender As Object, e As EventArgs) Handles
GoBackButton.Click
        Form1.PageAdmin.LoadAdminMedicinesPage()
    End Sub

    Private Sub AddMedicineButton_Click(sender As Object, e As EventArgs) Handles
AddMedicineButton.Click
        Dim Name = NameInput.Text
        Dim Formula = FormulaInput.Text
        Dim Company = CompanyInput.Text
        Dim Price = PriceInput.Value

        If Price <= 0 Then
            MessageBox.Show("Invalid price")
        Else
            MedicineModule.AddNewMedicine(Name, Formula, Company, Price)
        End If
    End Sub
End Class
```

Admin Inventory Page – AdminInventory.vb

```
Public Class AdminInventory
    Private Sub AddStockButton_Click(sender As Object, e As EventArgs) Handles
AddStockButton.Click
        Form1.PageAdmin.LoadAdminAddInventoryPage()
    End Sub

    Private Sub AdminInventory_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        MedicineModule.LoadInventoryData()
        Dim CurrentItem = 0
        For Each Medicine As MedicineStockModel In
MedicineModule.MedicineInventory
            MedicineInventoryList.Items.Add(Medicine.Id)
            MedicineInventoryList.Items(CurrentItem).SubItems.Add(Medicine.Name)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each Medicine As MedicineStockModel In
MedicineModule.MedicineInventory
            MedicineInventoryList.Items(CurrentItem).SubItems.Add(Medicine.Stock)
            CurrentItem += 1
        Next
    End Sub
End Class
```

Admin Add Inventory Page – AdminAddInventory.vb

```
Public Class AdminAddInventory
    Dim MedicineOptionsselectedIndex As Integer
    Private Sub GoBackInventoryButton_Click(sender As Object, e As EventArgs)
        Form1.PageAdmin.LoadAdminInventoryPage()
    End Sub
    Private Sub AdminAddInventory_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        MedicineModule.LoadMedicinesData()
        MedicineOptions.Items.Clear()

        For Each Medicine As MedicineStockModel In
MedicineModule.MedicineInventory
            MedicineOptions.Items.Add(Medicine.Name)
        Next
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        If (MedicineOptions.SelectedIndex < 0) Then
            MessageBox.Show("No medicine selected")
        Else
            MedicineModule.LoadMedicinesData()

            Dim SelectedMedicine =
MedicineModule.Medicines.ElementAt(MedicineOptions.SelectedIndex)
            Dim Stock = StockInput.Value

            MedicineModule.AddStock(SelectedMedicine.Id, Stock)

        End If
    End Sub

    Private Sub GoBackInventoryButton_Click_1(sender As Object, e As EventArgs)
Handles GoBackInventoryButton.Click
        Form1.PageAdmin.LoadAdminInventoryPage()
    End Sub
End Class
```

Admin Users Page – AdminUserManagement.vb

```
Public Class AdminUserManagement
    Dim SelectedIndex As Integer
    Private Sub AdminUserManagement_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
        UserModule.LoadUserData()
        Dim CurrentItem = 0
        For Each User As UserModel In UserModule.Users
            UsersListView.Items.Add(User.Id)
            UsersListView.Items(CurrentItem).SubItems.Add(User.Name)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each User As UserModel In UserModule.Users
            UsersListView.Items(CurrentItem).SubItems.Add(User.Username)
            CurrentItem += 1
        Next

        CurrentItem = 0
        For Each User As UserModel In UserModule.Users
            If User.IsAdmin Then
                UsersListView.Items(CurrentItem).SubItems.Add("Admin")
            Else
                UsersListView.Items(CurrentItem).SubItems.Add("Pharmacist")
            End If
            CurrentItem += 1
        Next
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
GoToAddUserPageButton.Click
        Form1.PageAdmin.LoadAdminAddUserPage()
    End Sub
End Class
```

Admin Add User Page – AdminAddUserPage.vb

```
Public Class AdminAddUserPage
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Form1.PageAdmin.LoadAdminUserManagementPage()
    End Sub

    Private Sub Label1_Click(sender As Object, e As EventArgs) Handles
Label1.Click

    End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
Button2.Click
        Dim Name = NameInput.Text
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text
        Dim IsAdmin As Boolean

        If (IsAdminInput.Text = "Admin") Then
            IsAdmin = True
        Else
            IsAdmin = False
        End If

        UserModule.AddUser(Name, Username, Password, IsAdmin)
    End Sub
End Class
```

Pharmacist Login Page – PharmacistLogin.vb

```
Public Class PharmacistLogin
    Private Sub GoToAdminLoginButton_Click(sender As Object, e As EventArgs)
Handles GoToAdminLoginButton.Click
        Form1.LoadAdminLoginPage()
    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
        Dim Username = UsernameInput.Text
        Dim Password = PasswordInput.Text

        PharmacistModule.PharmacistLogin(Username, Password)
    End Sub
End Class
```

Pharmacist Page – Pharmacist.vb

```
Public Class Pharmacist
```

```
    Dim PagePharmacistSale As PharmacistSales
    Private Sub Pharmacist_Load(sender As Object, e As EventArgs) Handles 
MyBase.Load
        PharmacistNameLabel.Text = Form1.LoggedInUserName
        LoadPharmacistSalePage()
    End Sub

    Private Sub LoadPharmacistSalePage()
        PagePharmacistSale = New PharmacistSales
        PanelBody.Controls.Clear()
        PanelBody.Controls.Add(PagePharmacistSale)
    End Sub

    Private Sub LogoutButton_Click(sender As Object, e As EventArgs) Handles 
LogoutButton.Click
        Form1.LoggedInUserName = ""
        Form1.LoadPharmacistLoginPage()
    End Sub
End Class
```

Pharmacist Sale Page – PharmacistSales.vb

```
Imports System.Drawing.Printing

Public Class PharmacistSales

    Dim ItemsInBill As New List(Of BillItem)
    Dim Total As Integer

    Dim WithEvents PD As New PrintDocument
    Dim PPD As New PrintPreviewDialog
    Dim PaperHeight = 150

    Private Sub PharmacistSales_Load(sender As Object, e As EventArgs) Handles 
MyBase.Load
        LoadMedicineDataIntoComboBox()
    End Sub


    Private Sub AddToBillButton_Click(sender As Object, e As EventArgs) Handles 
AddToBillButton.Click
        If (QuantityInput.Value <= 0) Then
            MessageBox.Show("Invalid quantity")
        ElseIf (MedicineSelector.SelectedIndex < 0) Then
            MessageBox.Show("Please select a medicine")
        Else
            Dim NewItem = New BillItem With {
                .Id = 
MedicineModule.Medicines.ElementAt(MedicineSelector.SelectedIndex).Id,
                .Name = 
MedicineModule.Medicines.ElementAt(MedicineSelector.SelectedIndex).Name,
                .Price = 
MedicineModule.Medicines.ElementAt(MedicineSelector.SelectedIndex).Price,
                .Quantity = QuantityInput.Value,
                .Amount = 
MedicineModule.Medicines.ElementAt(MedicineSelector.SelectedIndex).Price * 
QuantityInput.Value
            }

            Dim AlreadyExists = False
            Dim Index As Integer
            For Each Item As BillItem In ItemsInBill
                If (Item.Id = NewItem.Id) Then
```

```
                    AlreadyExists = True
                    Index = ItemsInBill.IndexOf(Item)
                End If
            Next

            If (AlreadyExists) Then

                Dim Quantity = ItemsInBill.ElementAt(Index).Quantity +
QuantityInput.Value
                MedicineModule.GetStockById(NewItem.Id)
                If (MedicineModule.MedicineStock < Quantity) Then
                    MessageBox.Show("Only " +
MedicineModule.MedicineStock.ToString() + " available")
                Else
                    ItemsInBill.ElementAt(Index).Quantity = Quantity
                    ItemsInBill.ElementAt(Index).Amount =
ItemsInBill.ElementAt(Index).Amount + QuantityInput.Value *
ItemsInBill.ElementAt(Index).Price
                End If
            Else
                MedicineModule.GetStockById(NewItem.Id)
                If (MedicineModule.MedicineStock < NewItem.Quantity) Then
                    MessageBox.Show("Only " +
MedicineModule.MedicineStock.ToString() + " available")
                Else
                    ItemsInBill.Add(NewItem)
                End If
            End If
            RenderListInGridView()
            ResetForm()
        End If
    End Sub

    Private Sub CheckoutBill()

        If (ItemsInBill.Count < 1) Then
            MessageBox.Show("Empty Bill")
        Else
            For Each Item As BillItem In ItemsInBill
                MedicineModule.AdjustInventory(Item.Id, Item.Quantity)
            Next

            SalesModule.CheckoutSale(Total)
            PrintBill()
            ResetForm()
            BillView.Rows.Clear()
            Total = 0
            ItemsInBill.Clear()
            TotalAmountLabel.Text = 0
        End If
    End Sub

    Private Sub LoadMedicineDataIntoComboBox()
        MedicineModule.LoadMedicinesData()

        For Each Medicine As MedicineModel In MedicineModule.Medicines
            MedicineSelector.Items.Add(Medicine.Name)
        Next
    End Sub

    Private Sub RenderListInGridView()
        BillView.Rows.Clear()
        Total = 0
        For Each Item As BillItem In ItemsInBill
            BillView.Rows.Add(Item.Name, Item.Price, Item.Quantity, Item.Amount)

            Total = Total + Item.Amount
        Next

        TotalAmountLabel.Text = Total
    End Sub

    Private Sub ResetForm()
        MedicineSelector.SelectedIndex = -1
        QuantityInput.Value = 1
    End Sub

    Private Sub CheckoutButton_Click(sender As Object, e As EventArgs) Handles
CheckoutButton.Click
        CheckoutBill()
    End Sub

    Private Sub ResetBillButtton_Click(sender As Object, e As EventArgs) Handles
ResetBillButtton.Click
        ResetForm()
        BillView.Rows.Clear()
        Total = 0
        ItemsInBill.Clear()
        TotalAmountLabel.Text = 0
    End Sub

    Private Sub PD_BeginPrint(sender As Object, e As PrintEventArgs) Handles
PD.BeginPrint
        Dim PageSetup As New PageSettings With {
        .PaperSize = New PaperSize("Custom", 250, PaperHeight)
        }

        PD.DefaultPageSettings = PageSetup
    End Sub

    Private Sub PD_PrintPage(sender As Object, e As PrintPageEventArgs) Handles
PD.PrintPage
        Dim FNormal As New Font("Calibri", 10, FontStyle.Regular)
        Dim FSmall As New Font("Calibri", 8, FontStyle.Regular)
        Dim FBold As New Font("Calibri", 10, FontStyle.Bold)
        Dim FBoldBig As New Font("Calibri", 12, FontStyle.Bold)

        Dim LeftMargin = PD.DefaultPageSettings.Margins.Left
        Dim CenterMargin = PD.DefaultPageSettings.PaperSize.Width / 2
        Dim RightMargin = PD.DefaultPageSettings.PaperSize.Width

        Dim RightAlign = New StringFormat With {
            .Alignment = StringAlignment.Far
        }
        Dim CenterAlign = New StringFormat With {
            .Alignment = StringAlignment.Center
        }

        Dim LeftAlign = New StringFormat With {
            .Alignment = StringAlignment.Near
        }

        Dim LineBraek = "--------------------------------------------------------
--------"

        ' Writing string
        e.Graphics.DrawString("ePharma", FBoldBig, Brushes.Black, CenterMargin,
10, CenterAlign)
        e.Graphics.DrawString("Invoice", FNormal, Brushes.Black, CenterMargin,
25, CenterAlign)
        e.Graphics.DrawString(Date.Now().ToString("dd-MM-yy | HH:mm"), FSmall,
Brushes.Black, 10, 40, LeftAlign)
        e.Graphics.DrawString(LineBraek, FSmall, Brushes.Black, 10, 50,
LeftAlign)
        e.Graphics.DrawString("#", FSmall, Brushes.Black, 20, 57, CenterAlign)
        e.Graphics.DrawString("Item", FSmall, Brushes.Black, 35, 57, LeftAlign)
        e.Graphics.DrawString("Price", FSmall, Brushes.Black, RightMargin - 100,
57, CenterAlign)
        e.Graphics.DrawString("Qty", FSmall, Brushes.Black, RightMargin - 70, 57,
CenterAlign)
        e.Graphics.DrawString("Amount", FSmall, Brushes.Black, RightMargin - 35,
57, CenterAlign)
        e.Graphics.DrawString(LineBraek, FSmall, Brushes.Black, 10, 64,
LeftAlign)

        Dim BaseLine = 63
        Dim ItemCount = 1
        For Each Item As BillItem In ItemsInBill
            e.Graphics.DrawString(ItemCount, FNormal, Brushes.Black, 20, BaseLine
+ 12, CenterAlign)
            e.Graphics.DrawString(Item.Name, FNormal, Brushes.Black, 35, BaseLine
+ 12, LeftAlign)
            e.Graphics.DrawString(Item.Price, FNormal, Brushes.Black, RightMargin
- 100, BaseLine + 12, CenterAlign)
            e.Graphics.DrawString(Item.Quantity, FNormal, Brushes.Black,
RightMargin - 70, BaseLine + 12, CenterAlign)
            e.Graphics.DrawString(Item.Quantity, FNormal, Brushes.Black,
RightMargin - 20, BaseLine + 12, RightAlign)

            BaseLine += 12
            ItemCount += 1
        Next

        BaseLine += 11
        e.Graphics.DrawString(LineBraek, FSmall, Brushes.Black, 10, BaseLine,
LeftAlign)

        e.Graphics.DrawString("Total (Rs.) :", FBold, Brushes.Black, RightMargin
- 50, BaseLine + 10, RightAlign)
        e.Graphics.DrawString(Total, FBold, Brushes.Black, RightMargin - 20,
BaseLine + 10, RightAlign)

        e.Graphics.DrawString("Thank you, have a nice day", FSmall,
Brushes.Black, CenterMargin, BaseLine + 30, CenterAlign)

    End Sub

    Private Sub CalculatePaperHeight()
        Dim Height = 63
        For Each Item As BillItem In ItemsInBill
            Height += 12
        Next

        Height += 70

        PaperHeight = Height
    End Sub
    Private Sub PrintBill()
        CalculatePaperHeight()
        PPD.Document = PD
        PPD.ShowDialog()
    End Sub
End Class

Class BillItem
    Public Id As Integer
    Public Name As String
    Public Price As Integer
    Public Quantity As Integer
    Public Amount As Integer
End Class
```

DbHelper.vb

```
Imports MySql.Data.MySqlClient

Module DbHelper
    Sub CreateTables()
        Dim UsersTableSchema = "create table if not exists users (id int not null
auto_increment primary key, name varchar(25) not null, username varchar(25) not
null, password varchar(20) not null, is_admin bool not null )"
        Dim MedicineTableSchema = "create table if not exists medicines (id int
not null auto_increment primary key, name varchar(50) not null, formula
varchar(25) not null, company varchar(25), price int, stock int)"
        Dim SalesTableSchema = "create table if not exists sales(id int not null
auto_increment primary key,date datetime default now(), amount int not null)"
        Try
            Form1.DbConnection.Open()
            Dim cmd = New MySqlCommand(UsersTableSchema, Form1.DbConnection)
            Dim Cmd2 = New MySqlCommand(MedicineTableSchema, Form1.DbConnection)
            Dim Cmd3 = New MySqlCommand(SalesTableSchema, Form1.DbConnection)
            cmd.ExecuteNonQuery()
            Cmd2.ExecuteNonQuery()
            Cmd3.ExecuteNonQuery()
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
            Form1.DbConnection.Dispose()
        End Try
    End Sub
End Module
```

AdminModule.vb

```
Imports MySql.Data.MySqlClient

Module AdminModule
    Sub AddAdmin(Name As String, Username As String, Password As String)
        Dim Sql = "insert into users (name, username, password, is_admin)
values(@Name, @Username, @Password, true)"
        Try
            Form1.DbConnection.Open()
            Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)

            Cmd.Parameters.AddWithValue("@Name", Name)
            Cmd.Parameters.AddWithValue("@Username", Username)
            Cmd.Parameters.AddWithValue("@Password", Password)
            Dim Res = Cmd.ExecuteScalar()

            ' Navigate to admin page
            Form1.LoadAdminPage()

        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Sub LoginAdmin(Username As String, password As String)
        Dim Sql = "select name, username, password, is_admin from users where
username=@Username"
        Try
            Form1.DbConnection.Open()
            Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
            Cmd.Parameters.AddWithValue("@Username", Username)

            Dim Reader = Cmd.ExecuteReader

            If Reader.Read() = True Then
                If Reader("is_admin") = True Then
                    If Reader("password") = password Then
                        Dim Name = Reader("name")
                        Form1.LoggedInUserName = Name

                        Form1.DbConnection.Close()
                        Form1.LoadAdminPage()
                    Else
                        MessageBox.Show("Invalid password")
                    End If
                Else
                    MessageBox.Show("Invalid username")
                End If
            Else
                MessageBox.Show("Invalid username")
            End If



        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub
End Module
```

PharmacistModule.vb

```
Imports MySql.Data.MySqlClient

Module PharmacistModule
    Public Sub PharmacistLogin(Username As String, Password As String)
        Dim Sql = "select name, username, password, is_admin from users where
username=@Username and is_admin=false"
        Try
            Form1.DbConnection.Open()
            Dim Cmd = New MySqlCommand(Sql, Form1.DbConnection)
            Cmd.Parameters.AddWithValue("@Username", Username)

            Dim Reader = Cmd.ExecuteReader

            If Reader.Read() = True Then
                If Reader("password") = Password Then
                    Form1.LoggedInUserName = Reader("name")

                    Form1.DbConnection.Close()

                    Form1.LoadPharmacistPage()
                Else
                    MessageBox.Show("Invalid password")
                End If
            Else
                MessageBox.Show("Invalid username")
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub
End Module
```

UserModule.vb

```
Imports MySql.Data.MySqlClient

Module UserModule
    Public UserExists = False
    Public Users As New List(Of UserModel)

    Sub IsUsersExists()
        Dim Cmd = New MySqlCommand("select * from users", Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            If Reader.HasRows() = True Then
                UserExists = True
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)

        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Sub LoadUserData()
        Users.Clear()
        Dim Cmd = New MySqlCommand("select * from users", Form1.DbConnection)

        Try
            Form1.DbConnection.Open()
            Dim Reader = Cmd.ExecuteReader()

            If Reader.HasRows Then
```

```
                While Reader.Read()
                    Users.Add(New UserModel With {
                        .Id = Reader("id"),
                        .Name = Reader("name"),
                        .Username = Reader("username"),
                        .IsAdmin = Reader("is_admin")
                    })
                End While
            End If
        Catch ex As Exception
            MessageBox.Show(ex.Message)

        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub

    Sub AddUser(Name As String, Username As String, Password As String, IsAdmin
As Boolean)
        Dim Sql1 = "select name, username, password, is_admin from users where
username=@Username"
        Dim Sql2 = "insert into users (name, username, password, is_admin) values
(@Name, @Username, @Password, @IsAdmin)"



        Try
            Form1.DbConnection.Open()

            ' Check if the user already exists
            Dim CheckCmd = New MySqlCommand(Sql1, Form1.DbConnection)
            CheckCmd.Parameters.AddWithValue("@Username", Username)
            Dim Reader = CheckCmd.ExecuteReader

            If Reader.Read() = True Then
                MessageBox.Show("User laready exists")
                Return
            End If
            Form1.DbConnection.Close()

            ' Adding new user
            Form1.DbConnection.Open()
            Dim Cmd = New MySqlCommand(Sql2, Form1.DbConnection)
            Cmd.Parameters.AddWithValue("@Name", Name)
            Cmd.Parameters.AddWithValue("@Username", Username)
            Cmd.Parameters.AddWithValue("@Password", Password)
            Cmd.Parameters.AddWithValue("@IsAdmin", IsAdmin)
            Cmd.ExecuteScalar()

            Form1.DbConnection.Close()

            ' Navigate to user management page
            Form1.PageAdmin.LoadAdminUserManagementPage()

        Catch ex As Exception
            MessageBox.Show(ex.Message)

        Finally
            Form1.DbConnection.Dispose()
        End Try
    End Sub

End Module
```

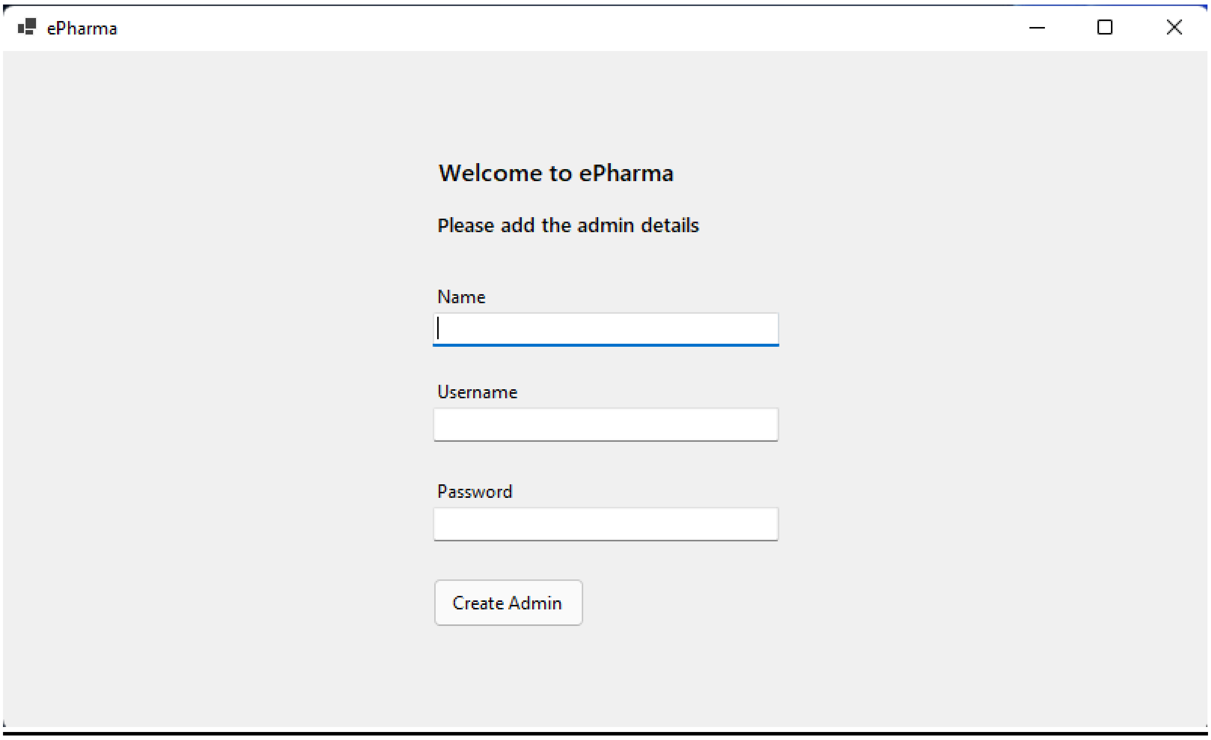
```
Public Class UserModel
    Public Id As Integer
    Public Name As String
    Public Username As String
    Public IsAdmin As Boolean
End Class
```

SalesModule.vb

```
Imports MySql.Data.MySqlClient

Module SalesModule
    Public Sub CheckoutSale(total As Integer)
        Dim Cmd = New MySqlCommand("insert into sales (amount) values (@Total)",
Form1.DbConnection)
        Cmd.Parameters.AddWithValue("@Total", total)

        Try
            Form1.DbConnection.Open()
            Cmd.ExecuteNonQuery()
            Form1.DbConnection.Close()

        Catch ex As Exception
            MessageBox.Show(ex.Message)
        Finally
            Form1.DbConnection.Close()
        End Try
    End Sub
End Module
```

UI (SCREENSHOTS)

Add Admin Page



Welcome to ePharma

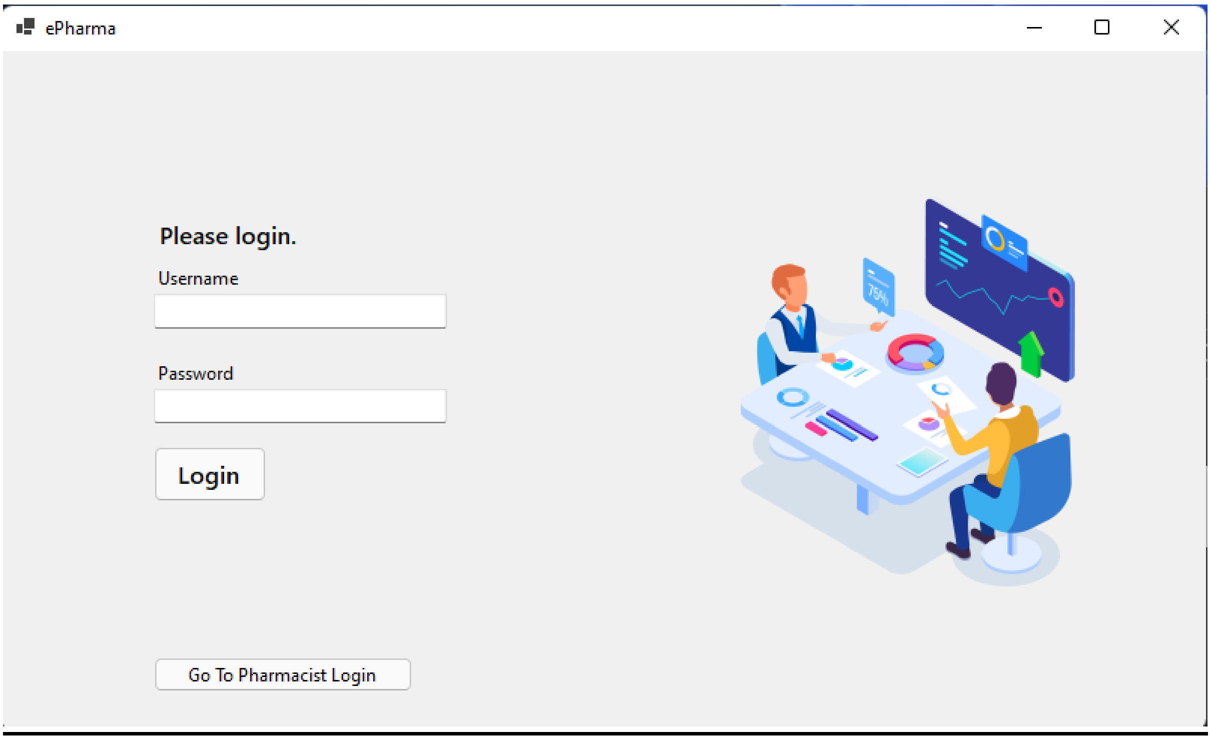
Please add the admin details

Name

Username

Password

Admin Login Page

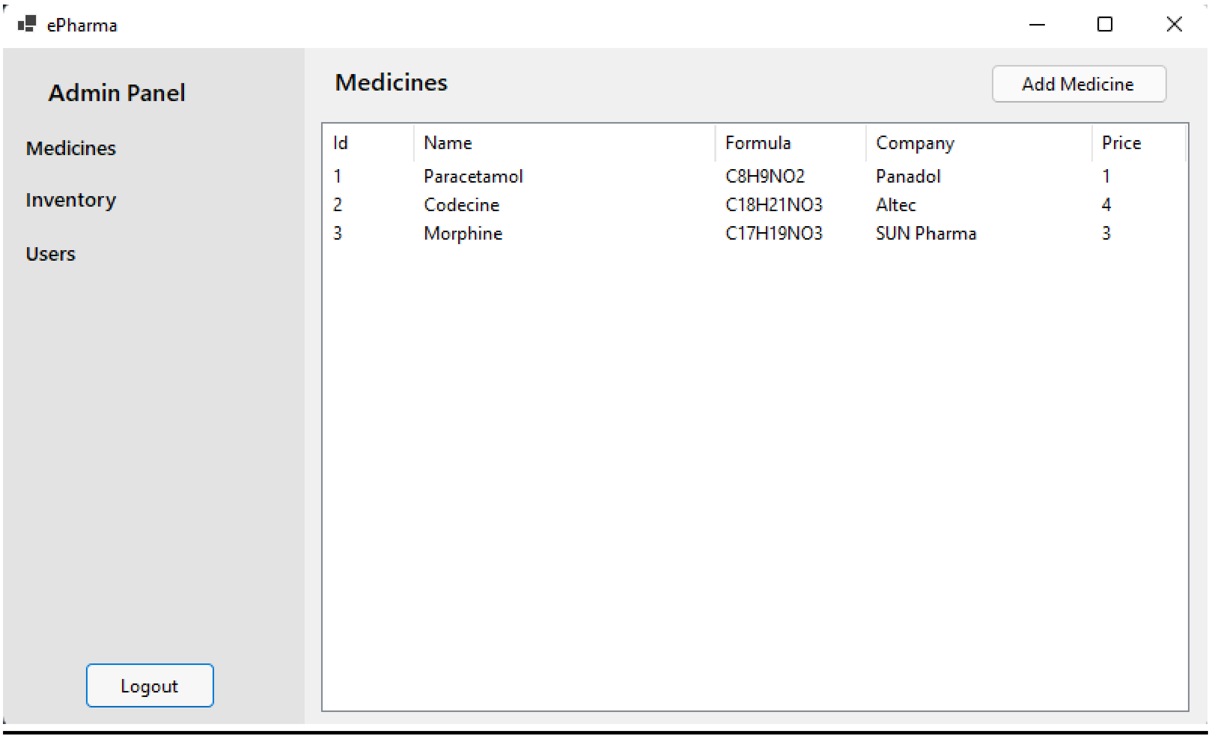


Please login.

Username

Password

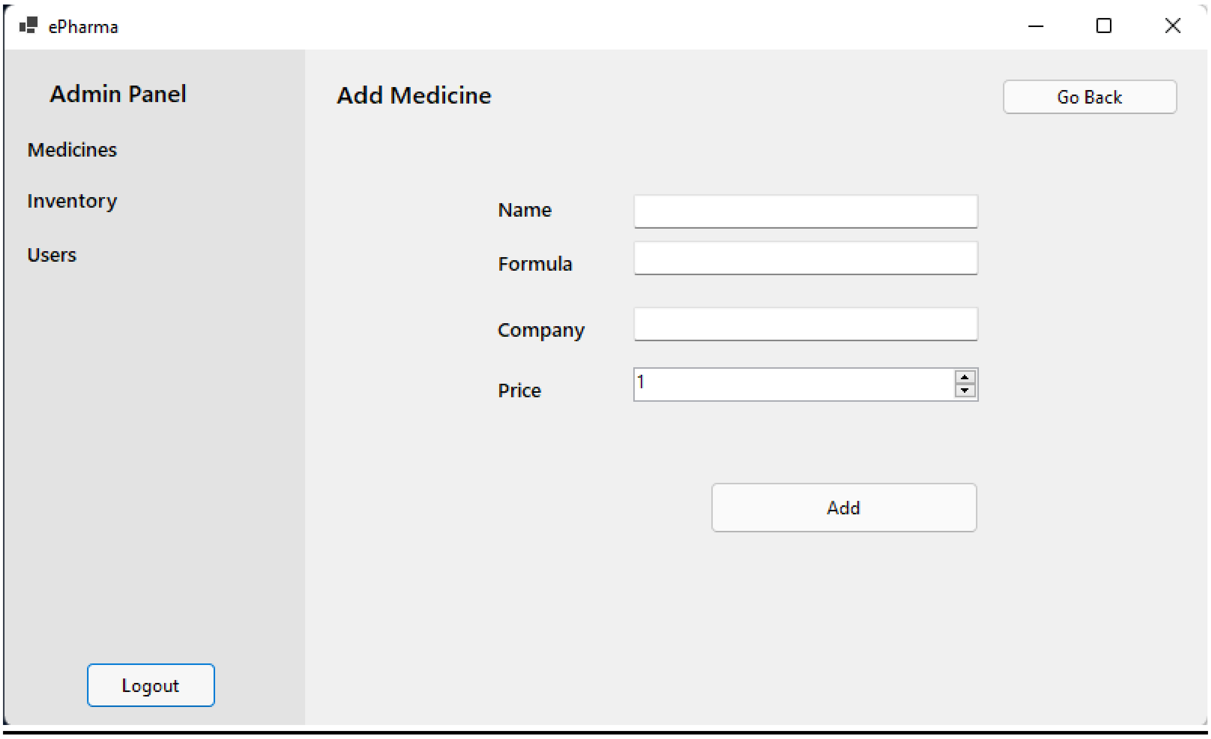
Admin Medicines Page



Medicines

Id	Name	Formula	Company	Price
1	Paracetamol	C8H9NO2	Panadol	1
2	Codecine	C18H21NO3	Altec	4
3	Morphine	C17H19NO3	SUN Pharma	3

Admin Add Medicine Page



Add Medicine

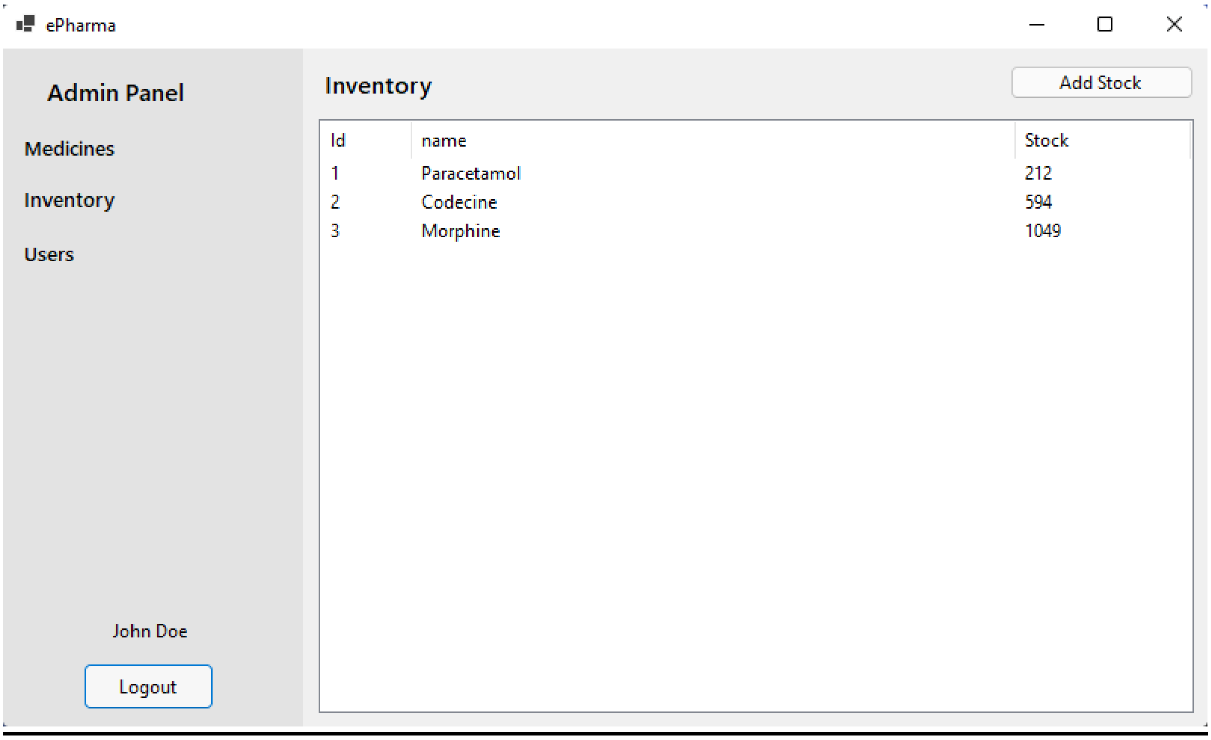
Name

Formula

Company

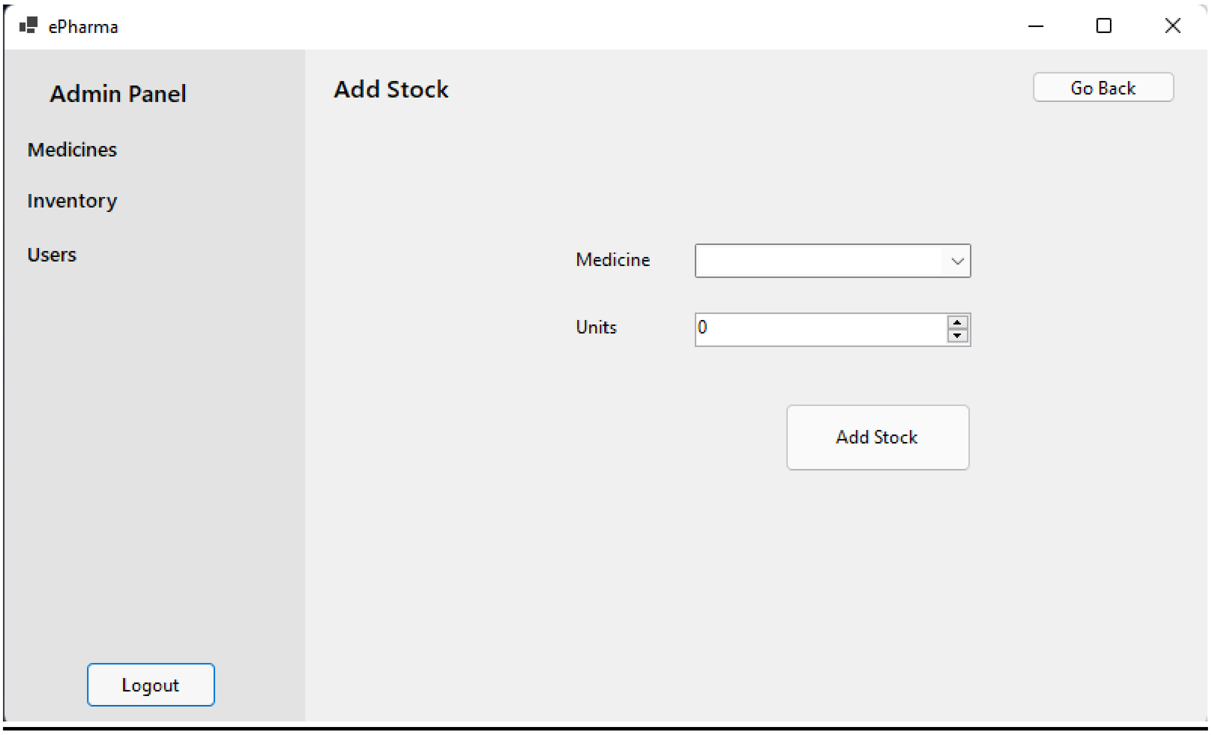
Price 1

Admin Inventory Page



Id	name	Stock
1	Paracetamol	212
2	Codecine	594
3	Morphine	1049

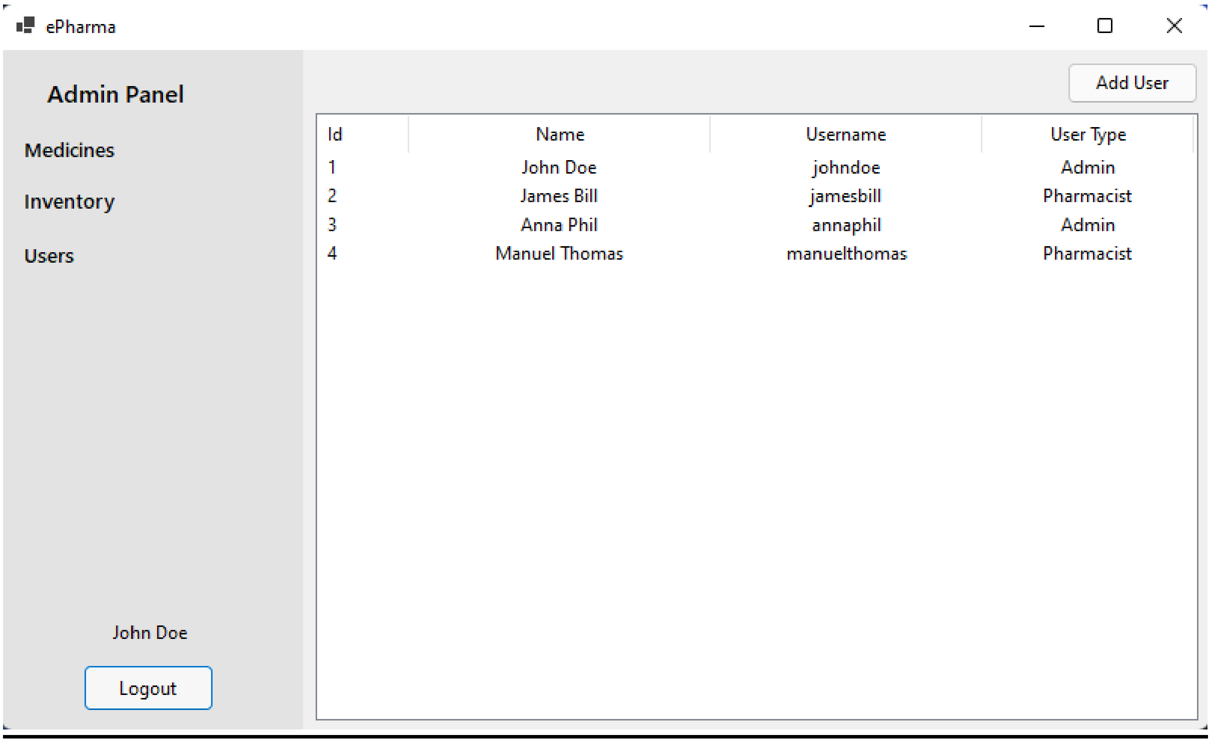
Admin Add Stock Page



Medicine

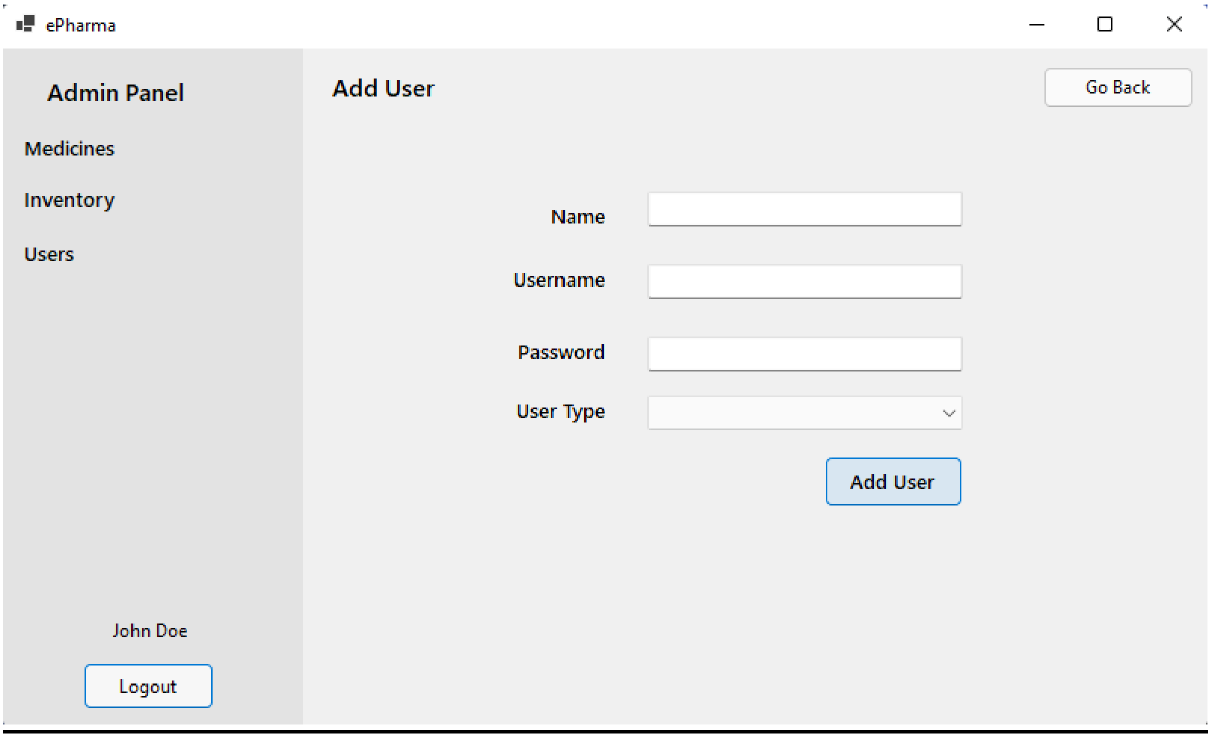
Units 0

Admin User Management Page

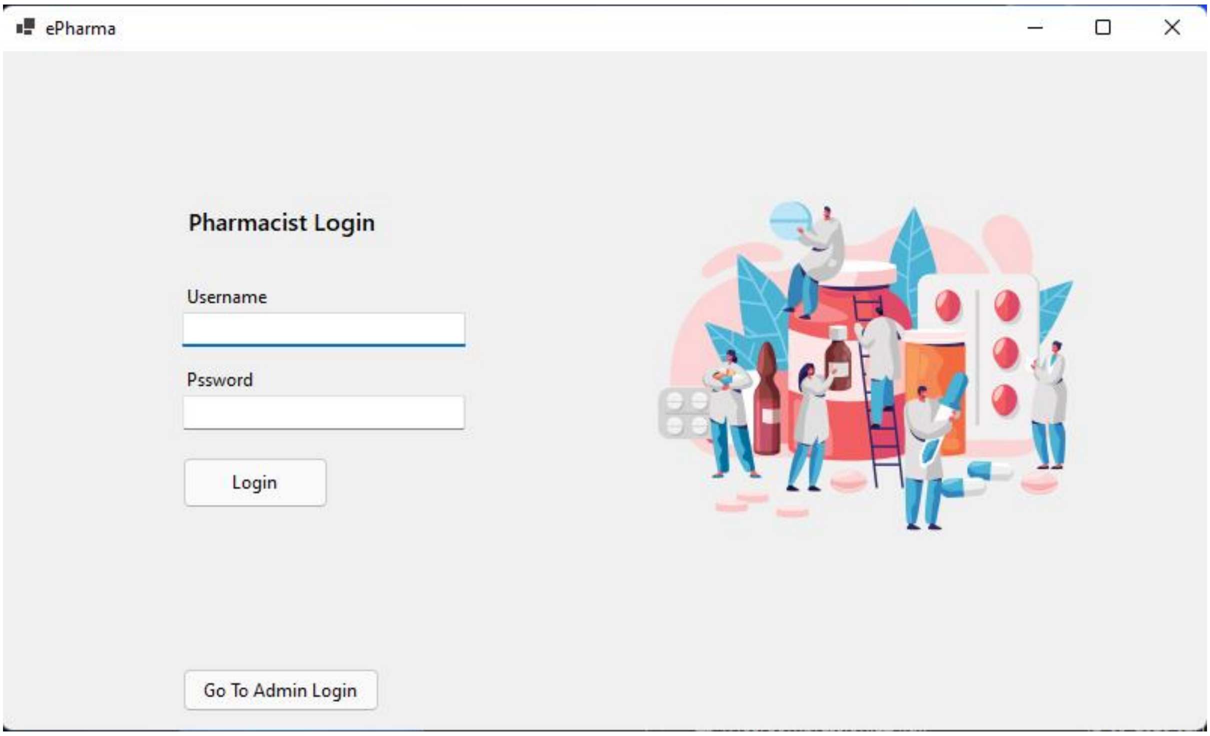


Id	Name	Username	User Type
1	John Doe	johndoe	Admin
2	James Bill	jamesbill	Pharmacist
3	Anna Phil	annaphil	Admin
4	Manuel Thomas	manuelthomas	Pharmacist

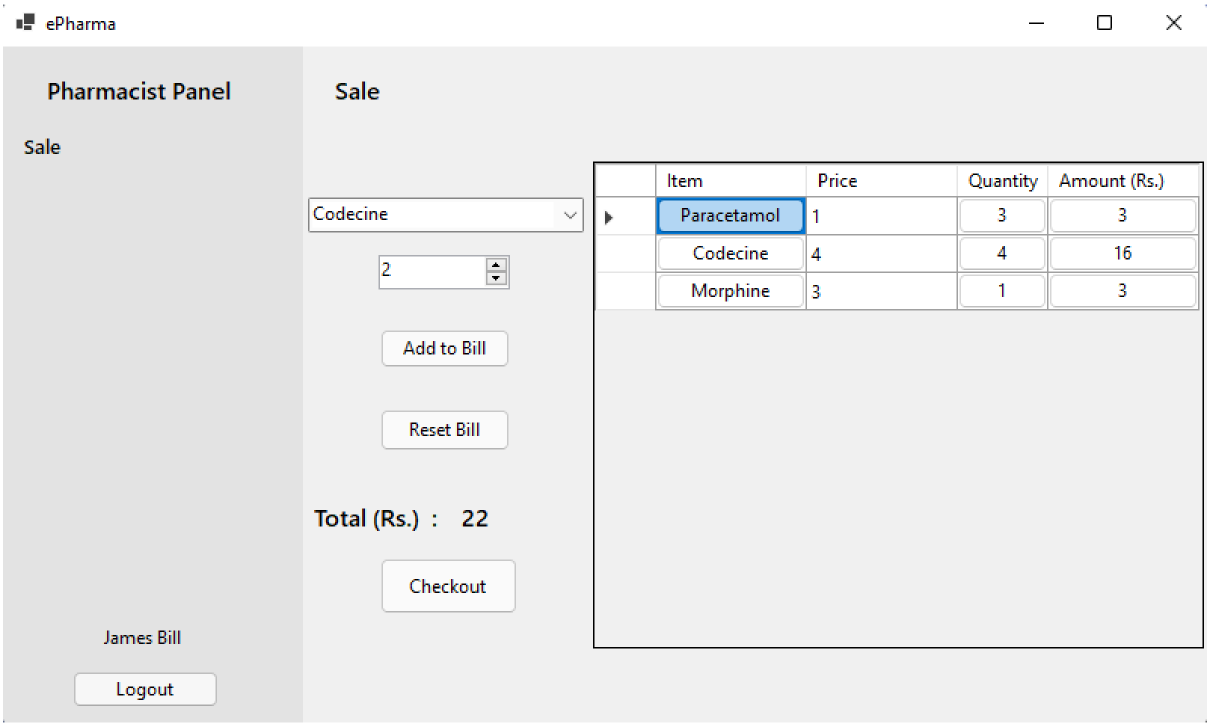
Admin Add User Page



Pharmacist Login Page



Pharmacist Sale Page



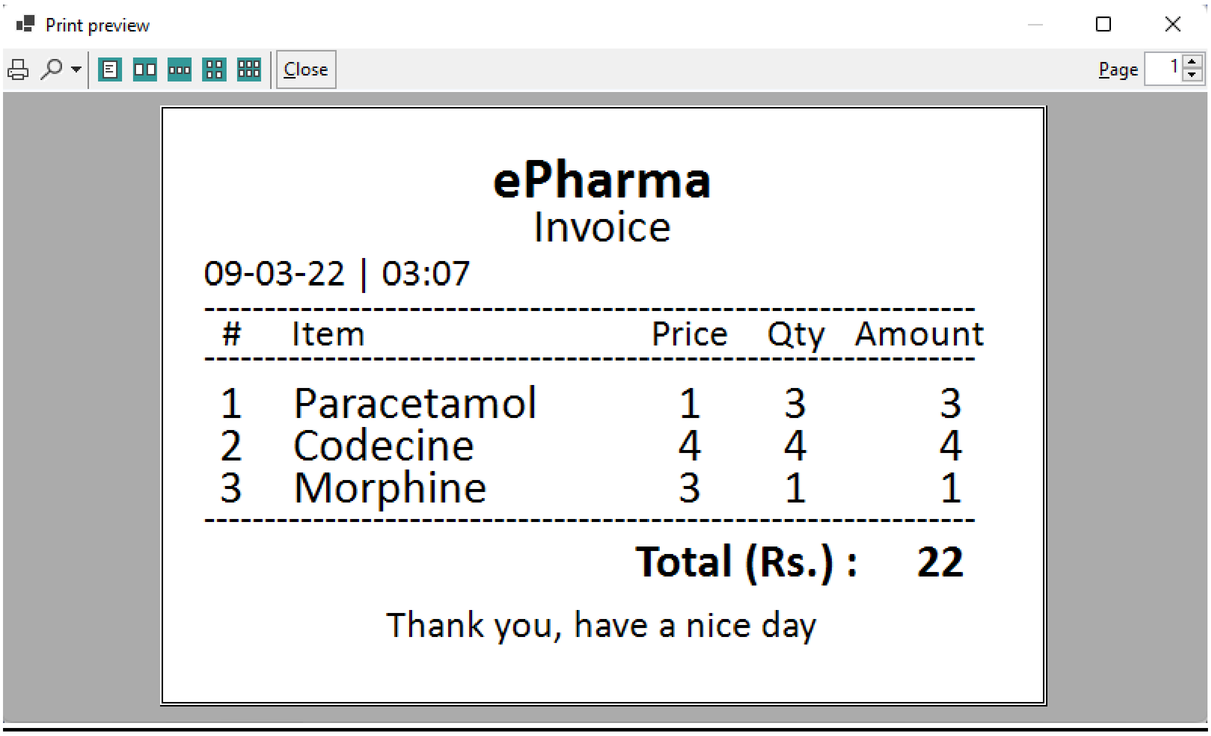
Codecine

2

Item	Price	Quantity	Amount (Rs.)
Paracetamol	1	3	3
Codecine	4	4	16
Morphine	3	1	3

Total (Rs.) : 22

Generated Bill Preview



ePharma
Invoice

09-03-22 | 03:07

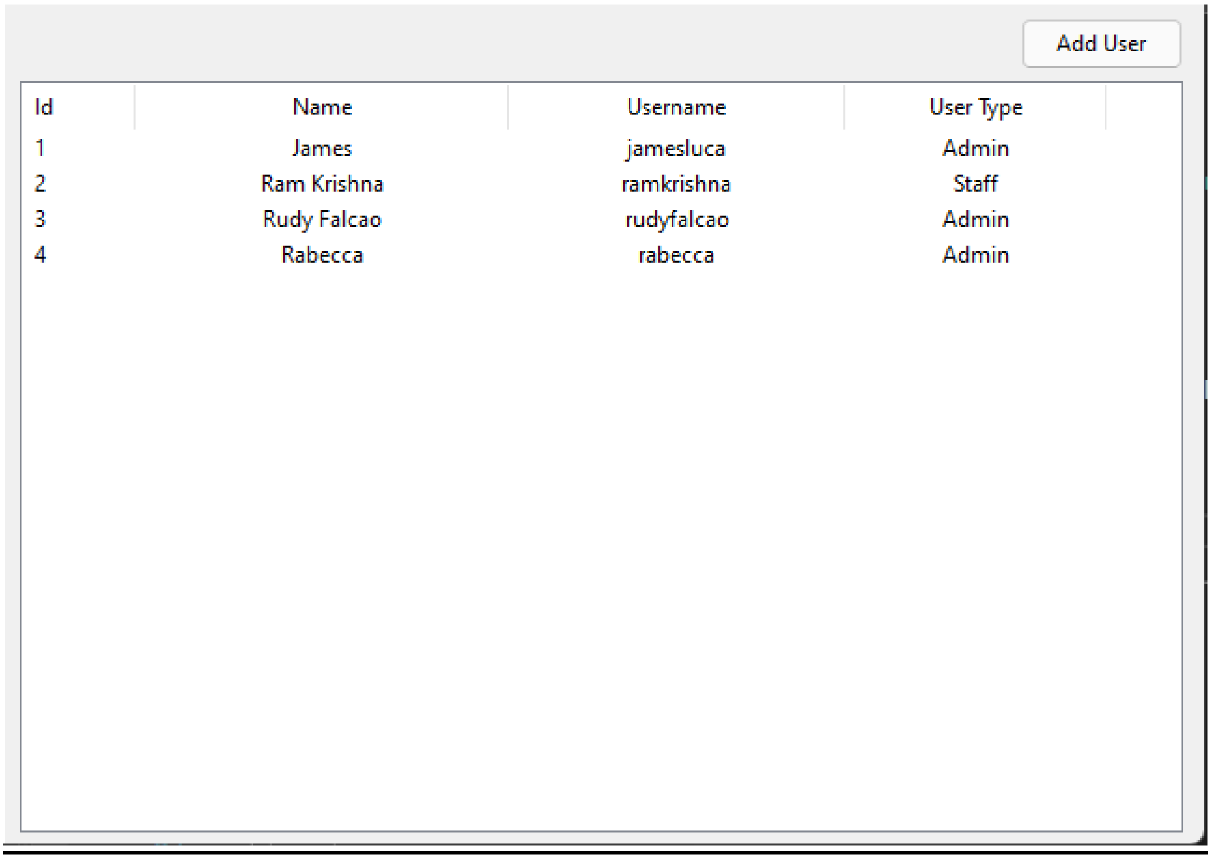
#	Item	Price	Qty	Amount
1	Paracetamol	1	3	3
2	Codecine	4	4	4
3	Morphine	3	1	1

Total (Rs.) : 22

Thank you, have a nice day

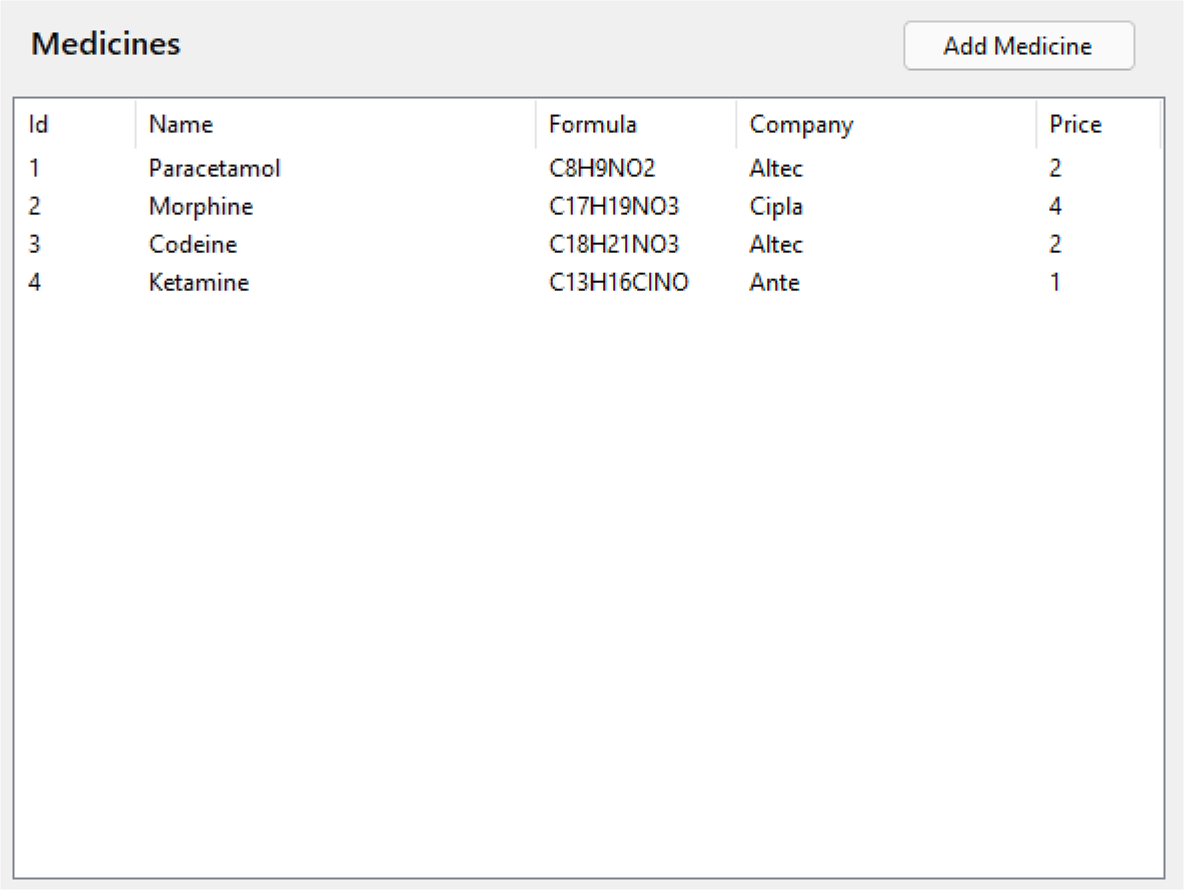
REPORTS

User Report



Id	Name	Username	User Type
1	James	jamesluca	Admin
2	Ram Krishna	ramkrishna	Staff
3	Rudy Falcao	rudyfalcao	Admin
4	Rabecca	rabecca	Admin

Medicine Report



Id	Name	Formula	Company	Price
1	Paracetamol	C8H9NO2	Altec	2
2	Morphine	C17H19NO3	Cipla	4
3	Codeine	C18H21NO3	Altec	2
4	Ketamine	C13H16ClNO	Ante	1