|  |
| --- |
| MainWindow.xaml |
| <Window x:Class="ShoppingMall.MainWindow"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  Title="New Mall" WindowState="Maximized" FontFamily="Times New Roman" FontSize="20">  <Window.Resources>  <ResourceDictionary>  <ResourceDictionary.MergedDictionaries>  <ResourceDictionary Source="/ControlStyle;component/Commonstyle.xaml" />  </ResourceDictionary.MergedDictionaries>  </ResourceDictionary>  </Window.Resources>    <DockPanel LastChildFill="True">  <UniformGrid DockPanel.Dock="Top" Columns="1">  <Label Name="lable1" VerticalContentAlignment="Center" HorizontalContentAlignment="Center" FontFamily="Times New Roman" FontSize="70" Foreground="#FF0966DF"  Content="Welcome to the Shoppingmall" FontWeight="Bold">  <Label.Triggers>  <EventTrigger RoutedEvent="FrameworkElement.Loaded">  <BeginStoryboard>  <Storyboard>  <DoubleAnimation AutoReverse="True" From="2" To="0" Storyboard.TargetName="lable1" Storyboard.TargetProperty="Opacity" RepeatBehavior="Forever">  </DoubleAnimation>  </Storyboard>  </BeginStoryboard>  </EventTrigger>  </Label.Triggers>  </Label>  </UniformGrid>  <UniformGrid DockPanel.Dock="Top" Rows="1" >      </UniformGrid>  <UniformGrid DockPanel.Dock="Top" Margin="2" Height="45">  <TextBlock FontFamily="Sylfaen" FontWeight="Bold" FontStyle="Normal" >  <Button Name="product" Style="{StaticResource ControlBtnStyle}" Click="product\_Click" Height="40">Products</Button>  <Button Name="shop" Style="{StaticResource ControlBtnStyle}" Click="shop\_Click" Height="40">Shops</Button>  <Button Name="floor" Style="{StaticResource ControlBtnStyle}" Height="40" Click="floor\_Click">Floor</Button>  <Button Name="contactusBtn" Style="{StaticResource ControlBtnStyle}" Click="contactusBtn\_Click" Height="40">Contact Us</Button>  <Button Name="adminZoneobjHL" Style="{StaticResource ControlBtnStyle}" Click="adminZoneobjHL\_Click" Height="40">Admin Zone</Button>  </TextBlock>  </UniformGrid>  <DockPanel DockPanel.Dock="Bottom" Name="infodocP" LastChildFill="True">  <UniformGrid DockPanel.Dock="Top">  <UniformGrid.Background>  <ImageBrush ImageSource="/ShoppingMallUI;component/Images/shopping-femmes-hd-fond-d-cran-182156.jpg" />  </UniformGrid.Background>  </UniformGrid>  </DockPanel>  </DockPanel>  </Window> |

|  |
| --- |
| MainWindow.xaml.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  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 ShoppingMall  {  /// <summary>  /// Interaction logic for MainWindow.xaml  /// </summary>  public partial class MainWindow : Window  {  public MainWindow()  {  InitializeComponent();  }  private void product\_Click(object sender, RoutedEventArgs e)  {  ShoppingMall.Products productobj = new ShoppingMall.Products();  infodocP.Children.Clear();  infodocP.Children.Add(productobj);    product.IsEnabled = false;  shop.IsEnabled = true;  floor.IsEnabled = true;  contactusBtn.IsEnabled = true;  adminZoneobjHL.IsEnabled = true;  //var bc = new BrushConverter();  //product.Background = (Brush)bc.ConvertFrom("#FFDFEBF2");  //product.Foreground = (Brush)bc.ConvertFrom("#FF0966DF");  }  private void shop\_Click(object sender, RoutedEventArgs e)  {  ShoppingMall.Shops shopobj = new ShoppingMall.Shops();  infodocP.Children.Clear();  infodocP.Children.Add(shopobj);    product.IsEnabled = true;  shop.IsEnabled = false;  floor.IsEnabled = true;  contactusBtn.IsEnabled = true;  adminZoneobjHL.IsEnabled = true;  }  private void contactusBtn\_Click(object sender, RoutedEventArgs e)  {  ShoppingMall.ContactUs ContactUsobj = new ShoppingMall.ContactUs();  infodocP.Children.Clear();  infodocP.Children.Add(ContactUsobj);  product.IsEnabled = true;  shop.IsEnabled = true;  floor.IsEnabled = true;  contactusBtn.IsEnabled = false;  adminZoneobjHL.IsEnabled = true;  }  private void adminZoneobjHL\_Click(object sender, RoutedEventArgs e)  {  ShoppingMall.AdminZone AdminZoneobj = new ShoppingMall.AdminZone();  infodocP.Children.Clear();  infodocP.Children.Add(AdminZoneobj);  product.IsEnabled = true;  shop.IsEnabled = true;  floor.IsEnabled = true;  contactusBtn.IsEnabled = true;  adminZoneobjHL.IsEnabled = false;  }  private void clearshopfeedbackFields()  {    }  private void floor\_Click(object sender, RoutedEventArgs e)  {  ShoppingMall.Floor Floorobj = new ShoppingMall.Floor();  infodocP.Children.Clear();  infodocP.Children.Add(Floorobj);  product.IsEnabled = true;  shop.IsEnabled = true;  floor.IsEnabled = false;  contactusBtn.IsEnabled = true;  adminZoneobjHL.IsEnabled = true;  }  }  } |

|  |
| --- |
| AdminZone.xmal |
| <UserControl x:Class="ShoppingMall.AdminZone"  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"  DataContext="{Binding RelativeSource={RelativeSource Self}}"  mc:Ignorable="d"  d:DesignHeight="541" d:DesignWidth="851" Style="{DynamicResource UserCntrlStyle}">  <UserControl.Resources>  <ResourceDictionary>  <ResourceDictionary.MergedDictionaries>  <ResourceDictionary Source="/ControlStyle;component/Commonstyle.xaml" />  </ResourceDictionary.MergedDictionaries>  </ResourceDictionary>  </UserControl.Resources>  <DockPanel LastChildFill="True" >  <DockPanel LastChildFill="True" DockPanel.Dock="Top">  <Button DockPanel.Dock="Right" Name="adminlogoutBtn" Style="{StaticResource ControlBtnStyle}" Visibility="Collapsed" Click="adminlogoutBtn\_Click">Logout Admin</Button>  <Label DockPanel.Dock="left" Background="#FF0966DF" Foreground="#FFFCFDFF" VerticalContentAlignment="Center" HorizontalContentAlignment="Center">Admin Zone</Label>  </DockPanel>    <UniformGrid Name="loginUG" Background="White" DockPanel.Dock="Top" Columns="4">  <Label></Label>  <Label Style="{StaticResource LblStyle}" >User Name</Label>  <TextBox Name="adminUserNameTB" Style="{StaticResource commonTBtyle}"></TextBox>  <Label></Label>  <Label></Label>  <Label Style="{StaticResource LblStyle}" >Password</Label>  <PasswordBox Name="adminUserPassPb" ></PasswordBox>  <Label></Label>    <Label></Label>  <Button Name="adminresetBtn" Style="{StaticResource ControlBtnStyle}" Click="adminresetBtn\_Click">Reset</Button>  <Button Name="adminloginBtn" Style="{StaticResource ControlBtnStyle}" Click="adminloginBtn\_Click">Login</Button>  <Label Name="loginErrorlbl" Style="{StaticResource LblStyle}"></Label>  </UniformGrid>  <UniformGrid Name="logoutUG" DockPanel.Dock="Top" Columns="4">  <Label></Label>  <Label></Label>  <Label></Label>      </UniformGrid>  <UniformGrid DockPanel.Dock="Bottom" Name="manageadminUG" Visibility="Collapsed">  <TabControl >    <TabItem Header="Manage Products" Style="{StaticResource TItemStyle}">  <DockPanel LastChildFill="True">  <UniformGrid Name="operationProductAdmitnUG" DockPanel.Dock="Bottom" Rows="1">  <TextBox Name="prdctAdminSearchTB"></TextBox>  <Button Name="searchAdminProductBtn" Style="{StaticResource ControlBtnStyle}" Click="searchAdminProductBtn\_Click">Search</Button>  <Label></Label>  <Button Name="refreshAdminProductBtn" Style="{StaticResource ControlBtnStyle}" Click="refreshAdminProductBtn\_Click">Refresh</Button>  <Button Name="editProductBtn" Style="{StaticResource ControlBtnStyle}" Click="editProductBtn\_Click">Edit</Button>  <Button Name="deleteProduct" Style="{StaticResource ControlBtnStyle}" Click="deleteProduct\_Click">Delete</Button>    </UniformGrid>  <UniformGrid Name="manageUG" DockPanel.Dock="Top" >  <DockPanel LastChildFill="True">    <Grid>  <Image Name="prdctimgPhoto" Stretch="Uniform" Height="142" Margin="14,6,14,53" Width="203"></Image>  <TextBox Name="prdctimagelinkTB" Margin="14,154,100,8" Width="117" Height="39"></TextBox>  <Button Name="prdctbrowseBtn" Style="{StaticResource ControlBtnStyle}" Click="prdctbrowseBtn\_Click" Margin="137,154,6,8" Height="39" Width="88">Browse</Button>  </Grid>  <DockPanel DockPanel.Dock="Right" LastChildFill="True" Margin="6">    <UniformGrid DockPanel.Dock="Top" Columns="2">  <Label >Available in :</Label>  <ComboBox Name="avaiableShopCB" IsEditable="True" Style="{StaticResource comboboxStyle}" ItemsSource="{Binding shopsCollection}" DisplayMemberPath="name" SelectedValuePath="id" MaxWidth="120" ></ComboBox>  <Button Name="deleteAvailShopItemBtn" Style="{StaticResource ControlBtnStyle}" Click="deleteAvailShopItemBtn\_Click">Delete</Button>  <Button Name="addShopToListview" Style="{StaticResource ControlBtnStyle}" Click="addShopToListview\_Click">Add</Button>  </UniformGrid>  <ListView Name="availableShopLView" DockPanel.Dock="Bottom" HorizontalAlignment="Stretch" MaxHeight="120" ScrollViewer.VerticalScrollBarVisibility="Visible">    </ListView>  </DockPanel>  <UniformGrid DockPanel.Dock="Left" Columns="2">  <Label Style="{StaticResource LblStyle}" >Name</Label>  <TextBox Name="nameTB"></TextBox>  <Label Style="{StaticResource LblStyle}" >Brand</Label>  <TextBox Name="BrandTB"></TextBox>  <Label Style="{StaticResource LblStyle}" >Type</Label>  <ComboBox IsEditable="True" Name="ProductypeCB" SelectedIndex="1">  <ComboBoxItem>Electronics</ComboBoxItem>  <ComboBoxItem>Garments</ComboBoxItem>  <ComboBoxItem>Computer</ComboBoxItem>  <ComboBoxItem>Mobile</ComboBoxItem>  <ComboBoxItem>Others</ComboBoxItem>  </ComboBox>  <Label Style="{StaticResource LblStyle}" >Description</Label>  <TextBox Name="productdescriptionTB"></TextBox>  <Button Name="resetBtn" Style="{StaticResource ControlBtnStyle}" Click="resetBtn\_Click">Reset</Button>  <Button Name="submitproductkBtn" Style="{StaticResource ControlBtnStyle}" Click="submitproductkBtn\_Click">Submit</Button>  <Button Name="editProductkBtn" Style="{StaticResource ControlBtnStyle}" Visibility="Collapsed" Click="editProductkBtn\_Click">Update</Button>  </UniformGrid>  </DockPanel>  </UniformGrid>    <UniformGrid DockPanel.Dock="Bottom">  <ListView Name="productsView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding productsCollection}" Loaded="Window\_Loaded">  <ListView.View>  <GridView>  <GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />  <GridViewColumn Width="200" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="250" Header="Brand" DisplayMemberBinding="{Binding brand}" />  <GridViewColumn Width="110" Header="Type" DisplayMemberBinding="{Binding type}" />  <GridViewColumn Width="370" Header="Description" DisplayMemberBinding="{Binding description}" />  <GridViewColumn Width="370" Header="Available In" DisplayMemberBinding="{Binding availableinshop}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>  </DockPanel>  </TabItem>  <TabItem Header="Manage Shops" Style="{StaticResource TItemStyle}">  <DockPanel LastChildFill="True">  <UniformGrid Name="oprtionShopUG" DockPanel.Dock="Bottom" Rows="1">  <TextBox Name="searchAdminShopmangTB"></TextBox>  <Button Name="searchAdminShopmangBtn" Style="{StaticResource ControlBtnStyle}" Click="searchAdminShopmangBtn\_Click">Search</Button>  <Label></Label>  <Button Name="refreshAdminShopBtn" Style="{StaticResource ControlBtnStyle}" Click="refreshAdminShopBtn\_Click">Refresh</Button>  <Button Name="editShopBtn" Style="{StaticResource ControlBtnStyle}" Click="editShopBtn\_Click">Edit</Button>  <Button Name="deleteShop" Click="deleteShop\_Click" Style="{StaticResource ControlBtnStyle}">Delete</Button>  </UniformGrid>  <DockPanel DockPanel.Dock="Top">  <Grid>  <Image Name="shopimgPhoto" Stretch="Uniform" Height="142" Margin="14,6,14,53" Width="203"></Image>  <TextBox Name="shopimagelinkTB" Margin="14,154,100,8" Width="117" Height="39"></TextBox>  <Button Name="shopbrowseBtn" Style="{StaticResource ControlBtnStyle}" Click="shopbrowseBtn\_Click" Margin="137,154,6,8" Height="39" Width="88">Browse</Button>  </Grid>  <DockPanel DockPanel.Dock="Right" LastChildFill="True" Margin="6">  <UniformGrid DockPanel.Dock="Top" Columns="2">  <Label >Available Product :</Label>  <ComboBox Name="avaiableProductCB" IsEditable="True" Style="{StaticResource comboboxStyle}" ItemsSource="{Binding productsCollection}" DisplayMemberPath="name" SelectedValuePath="id" MaxWidth="150"></ComboBox>  <Button Name="deleteAvailProductItemBtn" Style="{StaticResource ControlBtnStyle}" Click="deleteAvailProductItemBtn\_Click">Delete</Button>  <Button Name="addProductToListview" Style="{StaticResource ControlBtnStyle}" Click="addProductToListview\_Click">Add</Button>  </UniformGrid>  <ListView DockPanel.Dock="Bottom" Name="availproductView" HorizontalAlignment="Stretch" MaxHeight="150" ScrollViewer.VerticalScrollBarVisibility="Visible">  </ListView>  </DockPanel>  <UniformGrid Name="manageShopUG" DockPanel.Dock="Top" Columns="2">  <Label Style="{StaticResource LblStyle}" >Name</Label>  <TextBox Name="shopnameTB"></TextBox>  <Label Style="{StaticResource LblStyle}" >Tag</Label>  <TextBox Name="shopTagTB"></TextBox>  <Label Style="{StaticResource LblStyle}" >Type</Label>  <ComboBox IsEditable="True" Name="shopTypeCB" SelectedIndex="1">  <ComboBoxItem>Electronics</ComboBoxItem>  <ComboBoxItem>Garments</ComboBoxItem>  <ComboBoxItem>Computer</ComboBoxItem>  <ComboBoxItem>Mobile</ComboBoxItem>  <ComboBoxItem>Others</ComboBoxItem>  </ComboBox>  <Label Style="{StaticResource LblStyle}" >Available In</Label>  <ComboBox IsEditable="True" Name="availableShopFloorCB" SelectedIndex="1">  <ComboBoxItem>Ground Floor</ComboBoxItem>  <ComboBoxItem>First Floor</ComboBoxItem>  <ComboBoxItem>Second Floor</ComboBoxItem>  </ComboBox>  <Label Style="{StaticResource LblStyle}" >Rate (Out of 10)</Label>  <TextBox Name="shopRateTB"></TextBox>  <Label Style="{StaticResource LblStyle}" >Description</Label>  <TextBox Name="shopDescriptionTB"></TextBox>  <Button Name="resetShopmangBtn" Style="{StaticResource ControlBtnStyle}" Click="resetShopmangBtn\_Click">Reset</Button>  <Button Name="submitkShopmangBtn" Style="{StaticResource ControlBtnStyle}" Click="submitkShopmangBtn\_Click">Submit</Button>  <Button Name="UpdateShopmangBtn" Style="{StaticResource ControlBtnStyle}" Visibility="Collapsed" Click="UpdateShopmangBtn\_Click">Update</Button>  </UniformGrid>  </DockPanel>    <UniformGrid DockPanel.Dock="Bottom">  <ListView Name="shopsView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding shopsCollection}" Loaded="Window\_Loaded">  <ListView.View>  <GridView>  <GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />  <GridViewColumn Width="200" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="250" Header="Tag" DisplayMemberBinding="{Binding tag}" />  <GridViewColumn Width="110" Header="Type" DisplayMemberBinding="{Binding type}" />  <GridViewColumn Width="110" Header="Floor" DisplayMemberBinding="{Binding availableinfloor}" />  <GridViewColumn Width="370" Header="Rate" DisplayMemberBinding="{Binding rating}" />  <GridViewColumn Width="370" Header="Description" DisplayMemberBinding="{Binding description}" />  <GridViewColumn Width="370" Header="Available In" DisplayMemberBinding="{Binding availableProduct}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>  </DockPanel>  </TabItem>  <TabItem Header="Manage FeedBack" Style="{StaticResource TItemStyle}">  <DockPanel LastChildFill="True">  <UniformGrid Rows="1" DockPanel.Dock="Bottom" >  <Label></Label>  <Label></Label>  <Label></Label>  <Label></Label>  <Button Name="refreshFeedbackAdminBtn" Style="{StaticResource ControlBtnStyle}" Click="refreshFeedbackAdminBtn\_Click">Refresh</Button>  <Button Name="deleteContactusBtn" DockPanel.Dock="Bottom" Style="{StaticResource ControlBtnStyle}" Click="deleteContactusBtn\_Click">Delete</Button>  </UniformGrid>    <UniformGrid DockPanel.Dock="Top">  <ListView Name="contactusView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding contactusCollection}" Loaded="Window\_Loaded" >  <ListView.View>  <GridView>  <GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />  <GridViewColumn Width="100" Header="Date" DisplayMemberBinding="{Binding feedDate}" />  <GridViewColumn Width="200" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="250" Header="Address" DisplayMemberBinding="{Binding address}" />  <GridViewColumn Width="150" Header="Contact No." DisplayMemberBinding="{Binding mobileno}"/>  <GridViewColumn Width="200" Header="Email" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="110" Header="Type" DisplayMemberBinding="{Binding type}" />  <GridViewColumn Width="370" Header="Contactus" DisplayMemberBinding="{Binding contactus}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>  </DockPanel>  </TabItem>    <TabItem Header="Manage Others" Style="{StaticResource TItemStyle}">      </TabItem>  </TabControl>  </UniformGrid>  </DockPanel>  </UserControl> |

|  |
| --- |
| AdminZone.xmal.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  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 ShoppingMallData;  using ShoppingMallDb;  using System.Collections.ObjectModel;  namespace ShoppingMall  {  /// <summary>  /// Interaction logic for AdminZone.xaml  /// </summary>  public partial class AdminZone : UserControl  {  public AdminZone()  {  InitializeComponent();  }  private void Window\_Loaded(object sender, RoutedEventArgs e)  {  fetchProductData();  fetchShopData();  fetchFeedBackData();  }  private void adminloginBtn\_Click(object sender, RoutedEventArgs e)  {  if (adminUserNameTB.Text.Equals("1") && adminUserPassPb.Password.Equals("1"))  {  manageadminUG.Visibility = Visibility.Visible;  loginUG.Visibility = Visibility.Collapsed;  adminlogoutBtn.Visibility = Visibility.Visible;  clearSecurityFields();  }  else  {  adminUserPassPb.Password = "";  MessageBox.Show("Please Insert Info Properly");  }    }  private void adminlogoutBtn\_Click(object sender, RoutedEventArgs e)  {  manageadminUG.Visibility = Visibility.Collapsed;  loginUG.Visibility = Visibility.Visible;  adminlogoutBtn.Visibility = Visibility.Collapsed;  clearSecurityFields();  }  private void clearSecurityFields()  {  adminUserNameTB.Text = adminUserPassPb.Password = "";  }  private void adminresetBtn\_Click(object sender, RoutedEventArgs e)  {  clearSecurityFields();  }  private void clearProductFields()  {  nameTB.Text = BrandTB.Text = productdescriptionTB.Text = "";  ProductypeCB.SelectedIndex = 1;  }  private void resetBtn\_Click(object sender, RoutedEventArgs e)  {  clearProductFields();  }  #region Insert Product  private void submitproductkBtn\_Click(object sender, RoutedEventArgs e)  {  string availableinshop = "";  for (int i = 0; i < availableShopLView.Items.Count; i++)  {  availableinshop += availableShopLView.Items[i].ToString() + ", ";  }    if (!nameTB.Text.Equals("") && !BrandTB.Text.Equals("") && !ProductypeCB.Text.Equals("") && !productdescriptionTB.Text.Equals(""))  {  ShoppingMallData.ProductInfo newProduct = new ShoppingMallData.ProductInfo();  newProduct.id = GenerateId();  newProduct.name = nameTB.Text;  newProduct.brand = BrandTB.Text;  newProduct.type = ProductypeCB.Text;  newProduct.description = productdescriptionTB.Text;  newProduct.availableinshop = availableinshop;  //newProduct.image = shopimgPhoto.BitmapImage;  ShoppingMallDb.DbInteraction.DoEnterProduct(newProduct);  clearProductFields();  fetchProductData();  //takepic();  }  else  {  MessageBox.Show("Please Insert Info Properly");  }    }  private string GenerateId()  {  return DateTime.Now.ToOADate().ToString();  }  //private void takepic()  //{  // MySql.Data.MySqlClient.MySqlConnection msqlConnection = new MySql.Data.MySqlClient.MySqlConnection("server=localhost; user id=root;password=technicise;database=shoppingdb;persist security info=false");  // if (msqlConnection.State != System.Data.ConnectionState.Open)  // msqlConnection.Open();  // MySql.Data.MySqlClient.MySqlCommand msqlcommand = new MySql.Data.MySqlClient.MySqlCommand();  // msqlcommand.Connection = msqlConnection;  // msqlcommand.CommandText = "insert into product(image)" + "values(@image)";  // msqlcommand.Parameters.AddWithValue("@image", prdctimgPhoto);  // msqlcommand.ExecuteNonQuery();  // msqlConnection.Close();  // MessageBox.Show("Info Added");  //}  #endregion  #region Get Product  ObservableCollection<ProductInfo> \_productsCollection = new ObservableCollection<ProductInfo>();  public ObservableCollection<ProductInfo> productsCollection  {  get  {  return \_productsCollection;  }  }    private void fetchProductData()  {  List<ProductInfo> products = DbInteraction.GetAllProductList();  \_productsCollection.Clear();  foreach (ProductInfo product in products)  {  \_productsCollection.Add(product);  }  }  #endregion  #region Insert Shop  private void clearShopFields()  {  shopnameTB.Text = shopTagTB.Text = shopRateTB.Text = shopDescriptionTB.Text = "";  shopTypeCB.SelectedIndex = 1;  }  private void resetShopmangBtn\_Click(object sender, RoutedEventArgs e)  {  clearShopFields();  }  private void submitkShopmangBtn\_Click(object sender, RoutedEventArgs e)  {  string availableProduct = "";  for (int i = 0; i < availproductView.Items.Count; i++)  {  availableProduct += availproductView.Items[i].ToString() + ", ";  }    if (!shopnameTB.Text.Equals("") && !shopTagTB.Text.Equals("") && !shopTypeCB.Text.Equals("") && !shopRateTB.Text.Equals("") && !shopDescriptionTB.Text.Equals(""))  {    ShoppingMallData.ShopInfo newShop = new ShoppingMallData.ShopInfo();  newShop.id = GenerateId();  newShop.name = shopnameTB.Text;  newShop.tag = shopTagTB.Text;  newShop.type = shopTypeCB.Text;  newShop.availableinfloor = availableShopFloorCB.Text;  newShop.rating = shopRateTB.Text;  newShop.description = shopDescriptionTB.Text;  newShop.availableProduct = availableProduct;  ShoppingMallDb.DbInteraction.DoEnterShop(newShop);  clearShopFields();  fetchShopData();  }  else  {  MessageBox.Show("Please Insert Info Properly");  }  }  #endregion  #region Get Shop  ObservableCollection<ShopInfo> \_shopsCollection = new ObservableCollection<ShopInfo>();  public ObservableCollection<ShopInfo> shopsCollection  {  get  {  return \_shopsCollection;  }  }  private void fetchShopData()  {  List<ShopInfo> shops = DbInteraction.GetAllShopList();  \_shopsCollection.Clear();  foreach (ShopInfo shop in shops)  {  \_shopsCollection.Add(shop);  }  }  #endregion  #region Get FeedBack  ObservableCollection<ContactusInfo> \_contactusCollection = new ObservableCollection<ContactusInfo>();  public ObservableCollection<ContactusInfo> contactusCollection  {  get  {  return \_contactusCollection;  }  }    private void fetchFeedBackData()  {  List<ContactusInfo> contactuss = DbInteraction.GetAllContactusList();  \_contactusCollection.Clear();  foreach (ContactusInfo contactus in contactuss)  {  \_contactusCollection.Add(contactus);  }  }  #endregion  #region Delete Product  private ProductInfo GetSelectedProductItemforDel()  {  ProductInfo productToDelete = null;  if (productsView.SelectedIndex == -1)  {  MessageBox.Show("Please Select an Item");  }  else  {  ProductInfo i = (ProductInfo)productsView.SelectedItem;  productToDelete = \_productsCollection.Where(item => item.id.Equals(i.id)).First();  }  return productToDelete;  }  private void deleteProduct\_Click(object sender, RoutedEventArgs e)  {  ProductInfo productToDelete = GetSelectedProductItemforDel();  if (productToDelete != null)  {  productsCollection.Remove(productToDelete);  ShoppingMallDb.DbInteraction.DeleteProduct(productToDelete.id);  fetchProductData();  }  }  #endregion  #region Delete Shop  private ShopInfo GetSelectedShopItemforDel()  {  ShopInfo shopToDelete = null;  if (shopsView.SelectedIndex == -1)  MessageBox.Show("Please Select an Item");  else  {  ShopInfo i = (ShopInfo)shopsView.SelectedItem;  shopToDelete = \_shopsCollection.Where(item => item.id.Equals(i.id)).First();  }  return shopToDelete;  }  private void deleteShop\_Click(object sender, RoutedEventArgs e)  {  ShopInfo shopToDelete = GetSelectedShopItemforDel();  if (shopToDelete != null)  {  shopsCollection.Remove(shopToDelete);  ShoppingMallDb.DbInteraction.DeleteShop(shopToDelete.id);  fetchShopData();  }  }  #endregion  #region Delete FeedBack  private ContactusInfo GetSelectedContactusItem()  {  ContactusInfo contactusToDelete = null;  if (contactusView.SelectedIndex == -1)  MessageBox.Show("please select an item");  else  {  ContactusInfo i = (ContactusInfo)contactusView.SelectedItem;  contactusToDelete = \_contactusCollection.Where(item => item.id.Equals(i.id)).First();  }  return contactusToDelete;  }  private void deleteContactusBtn\_Click(object sender, RoutedEventArgs e)  {  ContactusInfo contactusToDelete = GetSelectedContactusItem();  if (contactusToDelete != null)  {  contactusCollection.Remove(contactusToDelete);  ShoppingMallDb.DbInteraction.DeleteContactus(contactusToDelete.id);  }  }  #endregion  #region Edit Product  private ProductInfo GetSelectedProductItemForEdit()  {  ProductInfo productToDelete = null;  if (productsView.SelectedIndex == -1)  {  MessageBox.Show("Please Select an Item");  }  else  {  editProductkBtn.Visibility = Visibility.Visible;  submitproductkBtn.Visibility = Visibility.Collapsed;  operationProductAdmitnUG.IsEnabled = false;  ProductInfo i = (ProductInfo)productsView.SelectedItem;  productToDelete = \_productsCollection.Where(item => item.id.Equals(i.id)).First();  }  return productToDelete;  }  private void editProductBtn\_Click(object sender, RoutedEventArgs e)  {  ProductInfo productToEdit = GetSelectedProductItemForEdit();  if (productToEdit != null)  {  nameTB.Text = productToEdit.name;  BrandTB.Text = productToEdit.brand;  ProductypeCB.Text = productToEdit.type;  productdescriptionTB.Text = productToEdit.description;  }    }  private void editProductkBtn\_Click(object sender, RoutedEventArgs e)  {  if (!nameTB.Text.Equals("") && !BrandTB.Text.Equals("") && !ProductypeCB.Text.Equals("") && !productdescriptionTB.Text.Equals(""))  {  ProductInfo productToEdit = GetSelectedProductItemForEdit();  productToEdit.name = nameTB.Text;  productToEdit.brand = BrandTB.Text;  productToEdit.type = ProductypeCB.Text;  productToEdit.description = productdescriptionTB.Text;  ShoppingMallDb.DbInteraction.EditProduct(productToEdit);  clearProductFields();  editProductkBtn.Visibility = Visibility.Collapsed;  submitproductkBtn.Visibility = Visibility.Visible;  fetchProductData();  operationProductAdmitnUG.IsEnabled = true;  }  else  {  MessageBox.Show("Please Insert Info Properly");  }  }  #endregion  #region Edit Shop    private ShopInfo GetSelectedShopItemforEdit()  {  ShopInfo shopToDelete = null;  if (shopsView.SelectedIndex == -1)  MessageBox.Show("Please Select an Item");  else  {  UpdateShopmangBtn.Visibility = Visibility.Visible;  submitkShopmangBtn.Visibility = Visibility.Collapsed;  oprtionShopUG.IsEnabled = false;  ShopInfo i = (ShopInfo)shopsView.SelectedItem;  shopToDelete = \_shopsCollection.Where(item => item.id.Equals(i.id)).First();  }  return shopToDelete;  }  private void editShopBtn\_Click(object sender, RoutedEventArgs e)  {  ShopInfo shopToEdit = GetSelectedShopItemforEdit();  if (shopToEdit != null)  {  shopnameTB.Text = shopToEdit.name;  shopTagTB.Text = shopToEdit.tag;  shopTypeCB.Text = shopToEdit.type;  availableShopFloorCB.Text = shopToEdit.availableinfloor;  shopRateTB.Text = shopToEdit.rating;  shopDescriptionTB.Text = shopToEdit.description;  }    }  private void UpdateShopmangBtn\_Click(object sender, RoutedEventArgs e)  {  if (!shopnameTB.Text.Equals("") && !shopTagTB.Text.Equals("") && !shopTypeCB.Text.Equals("") && !availableShopFloorCB.Text.Equals("") && !shopRateTB.Text.Equals("") && !shopDescriptionTB.Text.Equals(""))  {  ShopInfo shopToEdit = GetSelectedShopItemforEdit();  shopToEdit.name = shopnameTB.Text;  shopToEdit.tag = shopTagTB.Text;  shopToEdit.type = shopTypeCB.Text;  shopToEdit.availableinfloor = availableShopFloorCB.Text;  shopToEdit.rating = shopRateTB.Text;  shopToEdit.description = shopDescriptionTB.Text;  ShoppingMallDb.DbInteraction.EditShop(shopToEdit);  clearShopFields();  UpdateShopmangBtn.Visibility = Visibility.Collapsed;  submitkShopmangBtn.Visibility = Visibility.Visible;  fetchShopData();  oprtionShopUG.IsEnabled = true;  }  else  {  MessageBox.Show("Please Insert Info Properly");  }  }  #endregion  private void prdctbrowseBtn\_Click(object sender, RoutedEventArgs e)  {  var fd = new Microsoft.Win32.OpenFileDialog();  fd.Filter = "All image formats (\*.jpg; \*.jpeg; \*.bmp; \*.png; \*.gif)|\*.jpg;\*.jpeg;\*.bmp;\*.png;\*.gif";  var ret = fd.ShowDialog();  if (ret.GetValueOrDefault())  {  prdctimagelinkTB.Text = fd.FileName;  try  {  BitmapImage bmp = new BitmapImage(new Uri(fd.FileName, UriKind.Absolute));  prdctimgPhoto.Source = bmp;  }  catch (Exception)  {  MessageBox.Show("Invalid image file.", "Browse", MessageBoxButton.OK, MessageBoxImage.Exclamation);  }  }  }  private void shopbrowseBtn\_Click(object sender, RoutedEventArgs e)  {  var fd = new Microsoft.Win32.OpenFileDialog();  fd.Filter = "All image formats (\*.jpg; \*.jpeg; \*.bmp; \*.png; \*.gif)|\*.jpg;\*.jpeg;\*.bmp;\*.png;\*.gif";  var ret = fd.ShowDialog();  if (ret.GetValueOrDefault())  {  shopimagelinkTB.Text = fd.FileName;  try  {  BitmapImage bmp = new BitmapImage(new Uri(fd.FileName, UriKind.Absolute));  shopimgPhoto.Source = bmp;  }  catch (Exception)  {  MessageBox.Show("Invalid image file.", "Browse", MessageBoxButton.OK, MessageBoxImage.Exclamation);  }  }  }  #region Search Admin Product  private void searchAdminProductBtn\_Click(object sender, RoutedEventArgs e)  {  if (prdctAdminSearchTB.Text == "")  fetchProductData();  else  {  ProductInfo prodctInfo = new ProductInfo();  prodctInfo.name = prdctAdminSearchTB.Text;  List<ProductInfo> products = DbInteraction.searchProductList(prodctInfo);  \_productsCollection.Clear();  foreach (ProductInfo product in products)  {  \_productsCollection.Add(product);  }  }  }  #endregion  private void refreshAdminProductBtn\_Click(object sender, RoutedEventArgs e)  {  fetchProductData();  }  #region Search Admin Shop  private void searchAdminShopmangBtn\_Click(object sender, RoutedEventArgs e)  {  if (searchAdminShopmangTB.Text == "")  fetchShopData();  else  {  ShopInfo shopInfo = new ShopInfo();  shopInfo.name = searchAdminShopmangTB.Text;  List<ShopInfo> shops = DbInteraction.searchShopList(shopInfo);  \_shopsCollection.Clear();  foreach (ShopInfo shop in shops)  {  \_shopsCollection.Add(shop);  }  }  }  #endregion  private void refreshAdminShopBtn\_Click(object sender, RoutedEventArgs e)  {  fetchShopData();  }  private void refreshFeedbackAdminBtn\_Click(object sender, RoutedEventArgs e)  {  fetchFeedBackData();  }  private void deleteAvailShopItemBtn\_Click(object sender, RoutedEventArgs e)  {  availableShopLView.Items.RemoveAt(availableShopLView.Items.IndexOf(availableShopLView.SelectedItem));  }  private void addShopToListview\_Click(object sender, RoutedEventArgs e)  {  availableShopLView.Items.Add(avaiableShopCB.Text);  }  private void addProductToListview\_Click(object sender, RoutedEventArgs e)  {  availproductView.Items.Add(avaiableProductCB.Text);  }  private void deleteAvailProductItemBtn\_Click(object sender, RoutedEventArgs e)  {  availproductView.Items.RemoveAt(availproductView.Items.IndexOf(availproductView.SelectedItem));  }  }  } |

|  |
| --- |
| App.xmal |
| <Application x:Class="ShoppingMall.App"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  StartupUri="MainWindow.xaml">  <Application.Resources>    </Application.Resources>  </Application> |

|  |
| --- |
| App.xmal.cs |
| using System;  using System.Collections.Generic;  using System.Configuration;  using System.Data;  using System.Linq;  using System.Windows;  namespace ShoppingMall  {  /// <summary>  /// Interaction logic for App.xaml  /// </summary>  public partial class App : Application  {  }  } |

|  |
| --- |
| ContactUs.xmal |
| <UserControl x:Class="ShoppingMall.ContactUs"  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:sys="clr-namespace:System;assembly=mscorlib"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  DataContext="{Binding RelativeSource={RelativeSource Self}}"  mc:Ignorable="d"  d:DesignHeight="453" d:DesignWidth="785" Style="{DynamicResource UserCntrlStyle}">  <UserControl.Resources>  <ResourceDictionary>  <ResourceDictionary.MergedDictionaries>  <ResourceDictionary Source="/ControlStyle;component/Commonstyle.xaml"/>  </ResourceDictionary.MergedDictionaries>  </ResourceDictionary>  </UserControl.Resources>  <DockPanel LastChildFill="True" >  <Label DockPanel.Dock="Top" Background="#FF0966DF" Foreground="#FFFCFDFF" VerticalContentAlignment="Center" HorizontalContentAlignment="Center">Contract Us</Label>  <UniformGrid Name="upinfo" Background="White" DockPanel.Dock="Top" Columns="4">  <Label></Label>  <Label Style="{StaticResource LblStyle}" >Today</Label>  <DatePicker Name="feedDateDp" IsEnabled="False" VerticalContentAlignment="Center" HorizontalContentAlignment="Center" SelectedDate="{x:Static sys:DateTime.Now}" ></DatePicker>  <Label></Label>  <Label></Label>  <Label Style="{StaticResource LblStyle}" >Name</Label>  <TextBox Name="nameTB"></TextBox>  <Label></Label>  <Label></Label>  <Label Style="{StaticResource LblStyle}" >Address</Label>  <TextBox Name="addressTB"></TextBox>  <Label></Label>  <Label Style="{StaticResource LblStyle}" >Enter your</Label>  <Label Style="{StaticResource LblStyle}" >Mobile</Label>  <TextBox Name="mobilenoTB"></TextBox>  <Label></Label>  <Label Style="{StaticResource LblStyle}" >Feed Back</Label>  <Label Style="{StaticResource LblStyle}" >Email</Label>  <TextBox Name="emailTB"></TextBox>  <Label></Label>  <Label></Label>  <Label Style="{StaticResource LblStyle}" >Type</Label>  <ComboBox IsEditable="True" Name="typeCB" SelectedIndex="1">  <ComboBoxItem>Guest</ComboBoxItem>  <ComboBoxItem>Customer</ComboBoxItem>  <ComboBoxItem>Others</ComboBoxItem>  </ComboBox>  <Label></Label>  <Label></Label>  <Label Style="{StaticResource LblStyle}" >Complain/Feedback</Label>  <TextBox Name="contactusTB"></TextBox>  <Label></Label>  <Label></Label>  <Button Name="resetFdbckBtn" Style="{StaticResource ControlBtnStyle}" Click="resetFdbckBtn\_Click">Reset</Button>  <Button Name="submitFdBckBtn" Style="{StaticResource ControlBtnStyle}" Click="submitFdBckBtn\_Click">Submit</Button>  <Label></Label>      </UniformGrid>    <UniformGrid DockPanel.Dock="Bottom">  <ListView Name="contactusView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding contactusCollection}" Loaded="Window\_Loaded" >  <ListView.View>  <GridView>  <!--<GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />-->  <GridViewColumn Width="100" Header="Date" DisplayMemberBinding="{Binding feedDate}" />  <GridViewColumn Width="200" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="250" Header="Address" DisplayMemberBinding="{Binding address}" />  <GridViewColumn Width="150" Header="Contact No." DisplayMemberBinding="{Binding mobileno}"/>  <GridViewColumn Width="200" Header="Email" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="110" Header="Type" DisplayMemberBinding="{Binding type}" />  <GridViewColumn Width="370" Header="Feedback" DisplayMemberBinding="{Binding feedback}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>  </DockPanel>  </UserControl> |

|  |
| --- |
| ContuctUs.xmal.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  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 ShoppingMallData;  using ShoppingMallDb;  using System.Collections.ObjectModel;  namespace ShoppingMall  {  /// <summary>  /// Interaction logic for ContactUs.xaml  /// </summary>  public partial class ContactUs : UserControl  {  public ContactUs()  {  InitializeComponent();  }  #region insert FeedBack  private void submitFdBckBtn\_Click(object sender, RoutedEventArgs e)  {  if (!nameTB.Text.Equals("") && !addressTB.Text.Equals("") && !mobilenoTB.Text.Equals("") && !emailTB.Text.Equals("") && !typeCB.Text.Equals("") && !contactusTB.Text.Equals(""))  {    ShoppingMallData.ContactusInfo newContactus = new ShoppingMallData.ContactusInfo();  newContactus.id = GenerateId();  newContactus.feedDate = feedDateDp.SelectedDate.Value;  newContactus.name = nameTB.Text;  newContactus.address = addressTB.Text;  newContactus.mobileno = mobilenoTB.Text;  newContactus.email = emailTB.Text;  newContactus.type = typeCB.Text;  newContactus.feedback = contactusTB.Text;  ShoppingMallDb.DbInteraction.DoEnterContactus(newContactus);  clearFeedBackFields();  fetchFeedBackData();  }  else  {  MessageBox.Show("Please Insert Info Properly");  }  }  private string GenerateId()  {  return DateTime.Now.ToOADate().ToString();  }  private void resetFdbckBtn\_Click(object sender, RoutedEventArgs e)  {  clearFeedBackFields();  }  private void clearFeedBackFields()  {  nameTB.Text = addressTB.Text = mobilenoTB.Text = emailTB.Text = typeCB.Text= contactusTB.Text = "";  }  #endregion  ObservableCollection<ContactusInfo> \_contactusCollection = new ObservableCollection<ContactusInfo>();  public ObservableCollection<ContactusInfo> contactusCollection  {  get  {  return \_contactusCollection;  }  }  private void Window\_Loaded(object sender, RoutedEventArgs e)  {  fetchFeedBackData();  }  private void fetchFeedBackData()  {  List<ContactusInfo> contactuss = DbInteraction.GetAllContactusList();  \_contactusCollection.Clear();  foreach (ContactusInfo contactus in contactuss)  {  \_contactusCollection.Add(contactus);  }  }  }  } |

|  |
| --- |
| Feedback.xmal |
| <UserControl x:Class="ShoppingMall.Feedback"  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"  mc:Ignorable="d"  d:DesignHeight="300" d:DesignWidth="300" Style="{DynamicResource UserCntrlStyle}">  <UserControl.Resources>  <ResourceDictionary>  <ResourceDictionary.MergedDictionaries>  <ResourceDictionary Source="/ControlStyle;component/Commonstyle.xaml"/>  </ResourceDictionary.MergedDictionaries>  </ResourceDictionary>  </UserControl.Resources>  <Grid>  <Grid.RowDefinitions>  <RowDefinition Height="Auto"></RowDefinition>  <RowDefinition Height="\*"></RowDefinition>  <RowDefinition Height="Auto"></RowDefinition>  </Grid.RowDefinitions>  <Grid.ColumnDefinitions>  <ColumnDefinition Width="\*"></ColumnDefinition>  <ColumnDefinition Width="Auto"></ColumnDefinition>    </Grid.ColumnDefinitions>  <TextBlock Text="Name:" Grid.Row="0" Margin="5"></TextBlock>  <TextBlock Text="Date:" Grid.Row="0" Grid.Column="1" Margin="5"></TextBlock>  <TextBlock Text="Feedback:" Grid.Row="1" Margin="5" ScrollViewer.VerticalScrollBarVisibility="Visible"></TextBlock>  <TextBlock Text="Rating:" Grid.Row="2" Margin="5"></TextBlock>  </Grid>  </UserControl> |

|  |
| --- |
| Feedback.xmal.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  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 ShoppingMall  {  /// <summary>  /// Interaction logic for Feedback.xaml  /// </summary>  public partial class Feedback : UserControl  {  public Feedback()  {  InitializeComponent();  }  }  } |

|  |
| --- |
| Floor.xmal |
| <UserControl x:Class="ShoppingMall.Floor"  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:sys="clr-namespace:System;assembly=mscorlib"  DataContext="{Binding RelativeSource={RelativeSource Self}}"  xmlns:my="clr-namespace:ShoppingMall"  mc:Ignorable="d" Style="{DynamicResource UserCntrlStyle}">  <UserControl.Resources>  <ResourceDictionary>  <ResourceDictionary.MergedDictionaries>  <ResourceDictionary Source="/ControlStyle;component/Commonstyle.xaml" />  </ResourceDictionary.MergedDictionaries>  </ResourceDictionary>  </UserControl.Resources>    <DockPanel>  <Label DockPanel.Dock="Top" Background="#FF0966DF" Foreground="#FFFCFDFF" VerticalContentAlignment="Center" HorizontalContentAlignment="Center">Floor</Label>  <UniformGrid>  <TabControl >  <TabItem Header="Ground Floor" Style="{StaticResource TItemStyle}">  <DockPanel LastChildFill="True">    <DockPanel DockPanel.Dock="Left" LastChildFill="True">  <UniformGrid DockPanel.Dock="Top">  <DockPanel LastChildFill="True">  <Button Name="refreshgroundShopBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" Click="refreshgroundShopBtn\_Click">Refresh</Button>  <Button Name="gogroundShopBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" >Go</Button>  <ComboBox DockPanel.Dock="Left" Name="groundShopNameSrchCB" Style="{StaticResource comboboxStyle}" ItemsSource="{Binding shopsCollection}" DisplayMemberPath="name" SelectedValuePath="id"></ComboBox>  </DockPanel>  </UniformGrid>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ScrollViewer Width="250">  <ListView HorizontalAlignment="Stretch" Background="#FF0966DF" Foreground="#FFFCFDFF" Name="shopDetailsList" ItemsSource="{Binding shopsCollection}" Loaded="Window\_Loaded" SelectionChanged="ListView\_GroundFloorSelectionChanged">  <ListView.View>  <GridView>  <GridViewColumn Width="250" Header="Shop List" DisplayMemberBinding="{Binding name}" />  </GridView>  </ListView.View>  </ListView>  </ScrollViewer>  </UniformGrid>  </DockPanel>  <UniformGrid Name="grndflrupinfo" Background="White">  <DockPanel LastChildFill="True">  <Label DockPanel.Dock="Top" Background="#FF0966DF" Foreground="#FFFCFDFF" VerticalContentAlignment="Center" HorizontalContentAlignment="Center">Ground Floor</Label>  <DockPanel DockPanel.Dock="Right" >  <Image DockPanel.Dock="top" Source="/ShoppingMallUI;component/Images/Shopping-Black-Friday1.jpg" Width="234" Height="138" />    </DockPanel>  <DockPanel LastChildFill="True">  <TextBox Name="grndflrshopNameTb" DockPanel.Dock="Top" HorizontalContentAlignment="Center" IsEnabled="False">Shop Name</TextBox>  <TextBlock Name="grndflrshopdetailsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" ScrollViewer.VerticalScrollBarVisibility="Visible">Details</TextBlock>  <Label DockPanel.Dock="Top" >Available Products :</Label>  <TextBlock Name="availableGroundProductsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" MouseDown="availableProducts\_MouseDown">Available Products</TextBlock>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ListView Name="grndsfloorhopFeedbackView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding shopFeedbackCollection}" Loaded="Window\_Loaded" >  <ListView.View>  <GridView>  <!--<GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />-->  <GridViewColumn Width="100" Header="Date" DisplayMemberBinding="{Binding feedDate}" />  <GridViewColumn Width="150" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="150" Header="Email" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="70" Header="Rating" DisplayMemberBinding="{Binding rate}" />  <GridViewColumn Width="370" Header="Feedback" DisplayMemberBinding="{Binding feedback}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>  </DockPanel>  </DockPanel>  </UniformGrid>  </DockPanel>  </TabItem>  <TabItem Header="First Floor" Style="{StaticResource TItemStyle}">  <DockPanel LastChildFill="True">  <DockPanel DockPanel.Dock="Left" LastChildFill="True">  <UniformGrid DockPanel.Dock="Top">  <DockPanel LastChildFill="True">  <Button Name="refreshfirstflrShopBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" Click="refreshfirstflrShopBtn\_Click">Refresh</Button>  <Button Name="gofirstflrShopBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" >Go</Button>  <ComboBox DockPanel.Dock="Left" Name="firstflrNameSrchCB" Style="{StaticResource comboboxStyle}" ItemsSource="{Binding shopsCollection}" DisplayMemberPath="name" SelectedValuePath="id"></ComboBox>  </DockPanel>  </UniformGrid>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ScrollViewer Width="250">  <ListView HorizontalAlignment="Stretch" Background="#FF0966DF" Foreground="#FFFCFDFF" Name="firstflrDetailsList" ItemsSource="{Binding frstfloorshopsCollection}" SelectionChanged="ListView\_FirstFloorSelectionChanged">  <ListView.View>  <GridView>  <GridViewColumn Width="250" Header="Shop List" DisplayMemberBinding="{Binding name}" />  </GridView>  </ListView.View>  </ListView>  </ScrollViewer>  </UniformGrid>  </DockPanel>  <UniformGrid Name="firstflrupinfo" Background="White">  <DockPanel LastChildFill="True">  <Label DockPanel.Dock="Top" Background="#FF0966DF" Foreground="#FFFCFDFF" VerticalContentAlignment="Center" HorizontalContentAlignment="Center">First Floor</Label>  <DockPanel DockPanel.Dock="Right" >  <Image DockPanel.Dock="top" Source="/ShoppingMallUI;component/Images/Shopping-Black-Friday1.jpg" Width="234" Height="138" />  </DockPanel>  <DockPanel LastChildFill="True">  <TextBox Name="firstflrshopNameTb" DockPanel.Dock="Top" HorizontalContentAlignment="Center" IsEnabled="False">Shop Name</TextBox>  <TextBlock Name="firstflrshopdetailsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" ScrollViewer.VerticalScrollBarVisibility="Visible">Details</TextBlock>  <Label DockPanel.Dock="Top" >Available Products :</Label>  <TextBlock Name="availableFirstProductsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" MouseDown="availableProducts\_MouseDown">Available Products</TextBlock>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ListView Name="firstfloorshopFeedbackView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding shopFeedbackCollection}" Loaded="Window\_Loaded" >  <ListView.View>  <GridView>  <!--<GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />-->  <GridViewColumn Width="100" Header="Date" DisplayMemberBinding="{Binding feedDate}" />  <GridViewColumn Width="150" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="150" Header="Email" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="70" Header="Rating" DisplayMemberBinding="{Binding rate}" />  <GridViewColumn Width="370" Header="Feedback" DisplayMemberBinding="{Binding feedback}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>  </DockPanel>  </DockPanel>  </UniformGrid>  </DockPanel>  </TabItem>  <TabItem Header="Second Floor" Style="{StaticResource TItemStyle}">  <DockPanel LastChildFill="True">  <DockPanel DockPanel.Dock="Left" LastChildFill="True">  <UniformGrid DockPanel.Dock="Top">  <DockPanel LastChildFill="True">  <Button Name="refreshsecondflrShopBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" Click="refreshsecondflrShopBtn\_Click">Refresh</Button>  <Button Name="gosecondflrShopBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" >Go</Button>  <ComboBox DockPanel.Dock="Left" Name="secondflrNameSrchCB" Style="{StaticResource comboboxStyle}" ItemsSource="{Binding shopsCollection}" DisplayMemberPath="name" SelectedValuePath="id"></ComboBox>  </DockPanel>  </UniformGrid>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ScrollViewer Width="250">  <ListView HorizontalAlignment="Stretch" Background="#FF0966DF" Foreground="#FFFCFDFF" Name="secondflrDetailsList" ItemsSource="{Binding secondfloorshopsCollection}" SelectionChanged="ListView\_SecondFloorSelectionChanged">  <ListView.View>  <GridView >  <GridViewColumn Width="250" Header="Shop List" DisplayMemberBinding="{Binding name}" />  </GridView>  </ListView.View>  </ListView>  </ScrollViewer>  </UniformGrid>  </DockPanel>  <UniformGrid Name="secondflrupinfo" Background="White">  <DockPanel LastChildFill="True">  <Label DockPanel.Dock="Top" Background="#FF0966DF" Foreground="#FFFCFDFF" VerticalContentAlignment="Center" HorizontalContentAlignment="Center">Second Floor</Label>  <DockPanel DockPanel.Dock="Right" >  <Image DockPanel.Dock="top" Source="/ShoppingMallUI;component/Images/Shopping-Black-Friday1.jpg" Width="234" Height="138" />  </DockPanel>  <DockPanel LastChildFill="True">  <TextBox Name="secondflrshopNameTb" DockPanel.Dock="Top" HorizontalContentAlignment="Center" IsEnabled="False">Shop Name</TextBox>  <TextBlock Name="secondflrshopdetailsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" ScrollViewer.VerticalScrollBarVisibility="Visible">Details</TextBlock>  <Label DockPanel.Dock="Top" >Available Products :</Label>  <TextBlock Name="availableSecondProductsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" MouseDown="availableProducts\_MouseDown">Available Products</TextBlock>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ListView Name="secondfloorshopFeedbackView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding shopFeedbackCollection}" Loaded="Window\_Loaded" >  <ListView.View>  <GridView>  <!--<GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />-->  <GridViewColumn Width="100" Header="Date" DisplayMemberBinding="{Binding feedDate}" />  <GridViewColumn Width="150" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="150" Header="Email" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="70" Header="Rating" DisplayMemberBinding="{Binding rate}" />  <GridViewColumn Width="370" Header="Feedback" DisplayMemberBinding="{Binding feedback}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>  </DockPanel>  </DockPanel>  </UniformGrid>  </DockPanel>  </TabItem>    </TabControl>  </UniformGrid>    </DockPanel>  </UserControl> |

|  |
| --- |
| Floor.xmal.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  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 System.Collections.ObjectModel;  using ShoppingMallData;  using ShoppingMallDb;  namespace ShoppingMall  {  /// <summary>  /// Interaction logic for Floor.xaml  /// </summary>  public partial class Floor : UserControl  {  public Floor()  {  InitializeComponent();  }  private void Window\_Loaded(object sender, RoutedEventArgs e)  {  GetSelectedGroundFloorItem();  //GetSelectedFirstFloorItem();  //GetSelectedSecondFloorItem();  }  #region Get Groundfloor Shop  ObservableCollection<ShopInfo> \_shopsCollection = new ObservableCollection<ShopInfo>();  public ObservableCollection<ShopInfo> shopsCollection  {  get  {  return \_shopsCollection;  }  }  ObservableCollection<FeedbackInfo> \_shopFeedbackCollection = new ObservableCollection<FeedbackInfo>();  public ObservableCollection<FeedbackInfo> shopFeedbackCollection  {  get  {  return \_shopFeedbackCollection;  }  }  private void GetSelectedGroundFloorItem()  {  ShopInfo shopInfo = new ShopInfo();  shopInfo.name = "Ground Floor";  List<ShopInfo> shops = DbInteraction.getGroundfloorShopList(shopInfo);  \_shopsCollection.Clear();  foreach (ShopInfo shop in shops)  {  \_shopsCollection.Add(shop);  }  }  #endregion  #region Get First Floor Shop  ObservableCollection<ShopInfo> \_frstfloorshopsCollection = new ObservableCollection<ShopInfo>();  public ObservableCollection<ShopInfo> frstfloorshopsCollection  {  get  {  return \_frstfloorshopsCollection;  }  }  private void GetSelectedFirstFloorItem()  {  ShopInfo shopInfo = new ShopInfo();  shopInfo.name = "First Floor";  List<ShopInfo> shops = DbInteraction.getFirstFloorShopList(shopInfo);  \_frstfloorshopsCollection.Clear();  foreach (ShopInfo shop in shops)  {  \_frstfloorshopsCollection.Add(shop);  }  }  #endregion  #region Get Second Floor Shop  ObservableCollection<ShopInfo> \_secondfloorshopsCollection = new ObservableCollection<ShopInfo>();  public ObservableCollection<ShopInfo> secondfloorshopsCollection  {  get  {  return \_secondfloorshopsCollection;  }  }  private void GetSelectedSecondFloorItem()  {  ShopInfo shopInfo = new ShopInfo();  shopInfo.name = "Second Floor";  List<ShopInfo> shops = DbInteraction.getSecondFloorShopList(shopInfo);  \_secondfloorshopsCollection.Clear();  foreach (ShopInfo shop in shops)  {  \_secondfloorshopsCollection.Add(shop);  }  }  #endregion  private void refreshgroundShopBtn\_Click(object sender, RoutedEventArgs e)  {  GetSelectedGroundFloorItem();  }  private void refreshfirstflrShopBtn\_Click(object sender, RoutedEventArgs e)  {  GetSelectedFirstFloorItem();  }  private void refreshsecondflrShopBtn\_Click(object sender, RoutedEventArgs e)  {  GetSelectedSecondFloorItem();  }  private void ListView\_GroundFloorSelectionChanged(object sender, SelectionChangedEventArgs e)  {  if (shopDetailsList.SelectedIndex != -1)  {  ShopInfo shopInfoObj = \_shopsCollection.ElementAt(shopDetailsList.SelectedIndex);  List<ShopInfo> shops = DbInteraction.GetSelectedShopList(shopInfoObj);  grndflrshopNameTb.Text = shopInfoObj.name;  grndflrshopdetailsTBlock.Text = shopInfoObj.description;  availableGroundProductsTBlock.Text = shopInfoObj.availableProduct;  GetSelectedshopfeedbackGrndflrItem();  }  else  MessageBox.Show("SelectedIndex equals -1");  }  private void ListView\_FirstFloorSelectionChanged(object sender, SelectionChangedEventArgs e)  {  if (firstflrDetailsList.SelectedIndex != -1)  {  ShopInfo shopInfoObj = \_shopsCollection.ElementAt(firstflrDetailsList.SelectedIndex);  List<ShopInfo> shops = DbInteraction.GetSelectedShopList(shopInfoObj);  firstflrshopNameTb.Text = shopInfoObj.name;  firstflrshopdetailsTBlock.Text = shopInfoObj.description;  availableFirstProductsTBlock.Text = shopInfoObj.availableProduct;  GetSelectedshopfeedbackfrstflrItem();  }  else  MessageBox.Show("SelectedIndex equals -1");  }  private void ListView\_SecondFloorSelectionChanged(object sender, SelectionChangedEventArgs e)  {  if (secondflrDetailsList.SelectedIndex != -1)  {  ShopInfo shopInfoObj = \_shopsCollection.ElementAt(secondflrDetailsList.SelectedIndex);  List<ShopInfo> shops = DbInteraction.GetSelectedShopList(shopInfoObj);  secondflrshopNameTb.Text = shopInfoObj.name;  secondflrshopdetailsTBlock.Text = shopInfoObj.description;  availableSecondProductsTBlock.Text = shopInfoObj.availableProduct;  GetSelectedshopfeedbackSecondflrItem();  }  else  MessageBox.Show("SelectedIndex equals -1");  }  private void GetSelectedshopfeedbackGrndflrItem()  {  FeedbackInfo shopInfo = new FeedbackInfo();  shopInfo.name = grndflrshopNameTb.Text;  List<FeedbackInfo> shops = DbInteraction.getshopFeedbackList(shopInfo);  \_shopFeedbackCollection.Clear();  foreach (FeedbackInfo shop in shops)  {  \_shopFeedbackCollection.Add(shop);  }  }  private void GetSelectedshopfeedbackfrstflrItem()  {  FeedbackInfo shopInfo = new FeedbackInfo();  shopInfo.name = firstflrshopNameTb.Text;  List<FeedbackInfo> shops = DbInteraction.getshopFeedbackList(shopInfo);  \_shopFeedbackCollection.Clear();  foreach (FeedbackInfo shop in shops)  {  \_shopFeedbackCollection.Add(shop);  }  }  private void GetSelectedshopfeedbackSecondflrItem()  {  FeedbackInfo shopInfo = new FeedbackInfo();  shopInfo.name = secondflrshopNameTb.Text;  List<FeedbackInfo> shops = DbInteraction.getshopFeedbackList(shopInfo);  \_shopFeedbackCollection.Clear();  foreach (FeedbackInfo shop in shops)  {  \_shopFeedbackCollection.Add(shop);  }  }  private void availableProducts\_MouseDown(object sender, RoutedEventArgs e)  {  MessageBox.Show("Go To Product");  }  }  } |

|  |
| --- |
| Products.xmal |
| <UserControl x:Class="ShoppingMall.Products"  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:sys="clr-namespace:System;assembly=mscorlib"  DataContext="{Binding RelativeSource={RelativeSource Self}}"  xmlns:toolkit="http://schemas.microsoft.com/wpf/2008/toolkit"  xmlns:my="clr-namespace:ShoppingMall"  mc:Ignorable="d"  d:DesignHeight="586" d:DesignWidth="923" FontFamily="Times New Roman" FontSize="20" Foreground="#FF318931" Loaded="UserControl\_Loaded" >  <UserControl.Resources>  <ResourceDictionary>  <ResourceDictionary.MergedDictionaries>  <ResourceDictionary Source="/ControlStyle;component/Commonstyle.xaml" />  </ResourceDictionary.MergedDictionaries>  </ResourceDictionary>  </UserControl.Resources>  <DockPanel LastChildFill="True" >  <Label DockPanel.Dock="Top" Background="#FF0966DF" Foreground="#FFFCFDFF" VerticalContentAlignment="Center" HorizontalContentAlignment="Center">Products</Label>  <DockPanel DockPanel.Dock="Left" LastChildFill="True">  <UniformGrid DockPanel.Dock="Top">  <DockPanel LastChildFill="True">  <Button Name="refreshProductBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" Click="refreshProductBtn\_Click">Refresh</Button>  <Button Name="goProductBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" Click="goProductBtn\_Click">Go</Button>  <ComboBox DockPanel.Dock="Left" Name="productNameSrchCB" Style="{StaticResource comboboxStyle}" ItemsSource="{Binding productsCollection}" DisplayMemberPath="name" SelectedValuePath="id"></ComboBox>  </DockPanel>  </UniformGrid>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ScrollViewer Width="250">  <ListView Name="productDetailsList" ItemsSource="{Binding productsCollection}" HorizontalAlignment="Stretch" Background="#FF0966DF" Foreground="#FFFCFDFF" SelectionChanged="productDetailsList\_SelectionChanged">  <ListView.View>  <GridView>  <GridViewColumn Width="250" Header="Product Item" DisplayMemberBinding="{Binding name}" />  </GridView>  </ListView.View>  </ListView>  </ScrollViewer>  </UniformGrid>  </DockPanel>  <UniformGrid Name="upinfo" Background="White">  <DockPanel LastChildFill="True">  <DockPanel DockPanel.Dock="Right" >  <Image DockPanel.Dock="top" Source="/ShoppingMallUI;component/Images/Shopping-Black-Friday1.jpg" Width="234" Height="138" />  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <DatePicker Name="feedDateDp" Visibility="Collapsed" SelectedDate="{x:Static sys:DateTime.Now}" ></DatePicker>  <Label Style="{StaticResource LblStyle}" VerticalContentAlignment="Bottom">Your Name</Label>  <TextBox Name="nameTb"></TextBox>  <Label Style="{StaticResource LblStyle}" VerticalContentAlignment="Bottom">Email</Label>  <TextBox Name="mailTb"></TextBox>  <Label Style="{StaticResource LblStyle}" VerticalContentAlignment="Bottom">Product Rating (Out of 10)</Label>  <TextBox Name="ratingTb"></TextBox>  <Label Style="{StaticResource LblStyle}" VerticalContentAlignment="Bottom">Share your experience with all</Label>  <TextBox Name="feedbackTb"></TextBox>  <Label></Label>    <UniformGrid Rows="1">  <Button Name="clearProductfeedbackBtn" Style="{StaticResource ControlBtnStyle}" Click="clearProductfeedbackBtn\_Click">Reset</Button>  <Button Name="prdfdbckbtn" Style="{StaticResource ControlBtnStyle}" Click="prdfdbckbtn\_Click">Submit</Button>  </UniformGrid>  </UniformGrid>  </DockPanel>  <DockPanel LastChildFill="True">  <TextBox Name="productNameTb" DockPanel.Dock="Top" HorizontalContentAlignment="Center" IsEnabled="False">Product Name</TextBox>  <TextBlock Name="productDetailsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}">Contactus</TextBlock>  <Label DockPanel.Dock="Top" >Available In Following Shops : </Label>  <TextBlock Name="availableInShopTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" MouseDown="availableShops\_MouseDown">Available In Shop</TextBlock>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ListView Name="productFeedbackView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding productFeedbackCollection}" Loaded="Window\_Loaded" >  <ListView.View>  <GridView>  <!--<GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />-->  <GridViewColumn Width="100" Header="Date" DisplayMemberBinding="{Binding feedDate}" />  <GridViewColumn Width="150" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="150" Header="Email" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="70" Header="Rating" DisplayMemberBinding="{Binding rate}" />  <GridViewColumn Width="370" Header="Feedback" DisplayMemberBinding="{Binding feedback}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>  </DockPanel>  </DockPanel>  </UniformGrid>  </DockPanel>  </UserControl> |

|  |
| --- |
| Products.xmal.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  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 Microsoft.Windows.Controls;  using ShoppingMallData;  using System.Collections.ObjectModel;  using ShoppingMallDb;  namespace ShoppingMall  {  /// <summary>  /// Interaction logic for Products.xaml  /// </summary>  public partial class Products : UserControl  {  ObservableCollection<ProductInfo> \_productsCollection = new ObservableCollection<ProductInfo>();  public ObservableCollection<ProductInfo> productsCollection  {  get  {  return \_productsCollection;  }  }  public Products()  {  InitializeComponent();  }  //private void babiesproduct\_MouseEnter(object sender, MouseEventArgs e)  //{  // ShoppingMall.BabiesProduct bobj = new BabiesProduct();  // upinfo.Children.Clear();  // upinfo.Children.Add(bobj);  //}  private void Slider\_ValueChanged(  object sender,  RoutedPropertyChangedEventArgs<double> e)  {  var slider = sender as Slider;  var tick = slider.Ticks  .Where(xx => Math.Abs(e.NewValue - xx) < slider.LargeChange);  if (tick.Any())  {  var newValue = tick.First();  if (e.NewValue != newValue)  {  //DispatcherInvoke(() => slider.Value = newValue);  }  }  }  private void fetchProductData()  {  List<ProductInfo> products = DbInteraction.GetAllProductList();  \_productsCollection.Clear();  foreach (ProductInfo product in products)  {  \_productsCollection.Add(product);  }  }  private void UserControl\_Loaded(object sender, RoutedEventArgs e)  {  fetchProductData();  }  private void productDetailsList\_SelectionChanged(object sender, SelectionChangedEventArgs e)  {  if (productDetailsList.SelectedIndex != -1)  {  ProductInfo productInfoObj = \_productsCollection.ElementAt(productDetailsList.SelectedIndex);  List<ProductInfo> products = DbInteraction.GetSelectedProductList(productInfoObj);  productNameTb.Text = productInfoObj.name;  productDetailsTBlock.Text = productInfoObj.description;  availableInShopTBlock.Text = productInfoObj.availableinshop;  GetSelectedProductfeedbackItem();  }  else  MessageBox.Show("SelectedIndex equals -1");  }  private void prdfdbckbtn\_Click(object sender, RoutedEventArgs e)  {  if (!nameTb.Text.Equals("") && !mailTb.Text.Equals("") && !ratingTb.Text.Equals("") && !feedbackTb.Text.Equals(""))  {    ShoppingMallData.FeedbackInfo newFeedback = new ShoppingMallData.FeedbackInfo();  newFeedback.id = GenerateId();  newFeedback.item = productNameTb.Text;  newFeedback.feedDate = feedDateDp.SelectedDate.Value;  newFeedback.name = nameTb.Text;  newFeedback.email = mailTb.Text;  newFeedback.rate = ratingTb.Text;  newFeedback.feedback = feedbackTb.Text;  ShoppingMallDb.DbInteraction.DoEnterFeedback(newFeedback);    //fetchFeedBackData();  clearProductfeedbackFields();  GetSelectedProductfeedbackItem();  }  else  {  MessageBox.Show("Please Insert Info Properly");  }  }  private string GenerateId()  {  return DateTime.Now.ToOADate().ToString();  }  private void clearProductfeedbackFields()  {  nameTb.Text = mailTb.Text = ratingTb.Text = feedbackTb.Text = "";  }  private void clearProductfeedbackBtn\_Click(object sender, RoutedEventArgs e)  {  clearProductfeedbackFields();  }  private void goProductBtn\_Click(object sender, RoutedEventArgs e)  {  if (productNameSrchCB.Text == "")  fetchProductData();  else  {  ProductInfo prodctInfo = new ProductInfo();  prodctInfo.name = productNameSrchCB.Text;  List<ProductInfo> products = DbInteraction.searchProductList(prodctInfo);  \_productsCollection.Clear();  foreach (ProductInfo product in products)  {  \_productsCollection.Add(product);  }  }  }  private void refreshProductBtn\_Click(object sender, RoutedEventArgs e)  {  fetchProductData();  }  private void Window\_Loaded(object sender, RoutedEventArgs e)  {  //fetchshopFeedbackData();  }  #region Display only selected product Feedback  ObservableCollection<FeedbackInfo> \_productFeedbackCollection = new ObservableCollection<FeedbackInfo>();  public ObservableCollection<FeedbackInfo> productFeedbackCollection  {  get  {  return \_productFeedbackCollection;  }  }  private void GetSelectedProductfeedbackItem()  {  FeedbackInfo productInfo = new FeedbackInfo();  productInfo.name = productNameTb.Text;  List<FeedbackInfo> products = DbInteraction.getproductFeedbackList(productInfo);  \_productFeedbackCollection.Clear();  foreach (FeedbackInfo product in products)  {  \_productFeedbackCollection.Add(product);  }  }  #endregion  private void availableShops\_MouseDown(object sender, RoutedEventArgs e)  {  MessageBox.Show("Go To shop");  }  }  } |

|  |
| --- |
| Shops.xmal |
| <UserControl x:Class="ShoppingMall.Shops"  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:sys="clr-namespace:System;assembly=mscorlib"  xmlns:my="clr-namespace:ShoppingMall"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  DataContext="{Binding RelativeSource={RelativeSource Self}}"  mc:Ignorable="d"  d:DesignHeight="548" d:DesignWidth="882" FontFamily="Times New Roman" FontSize="20" Foreground="#FF318931" Loaded="UserControl\_Loaded">  <UserControl.Resources>  <ResourceDictionary>  <ResourceDictionary.MergedDictionaries>  <ResourceDictionary Source="/ControlStyle;component/Commonstyle.xaml" />  </ResourceDictionary.MergedDictionaries>  </ResourceDictionary>  </UserControl.Resources>  <DockPanel LastChildFill="True">  <Label DockPanel.Dock="Top" Background="#FF0966DF" Foreground="#FFFCFDFF" VerticalContentAlignment="Center" HorizontalContentAlignment="Center">Shops</Label>  <DockPanel DockPanel.Dock="Left" LastChildFill="True">  <UniformGrid DockPanel.Dock="Top">  <DockPanel LastChildFill="True">  <Button Name="refreshShopBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" Click="refreshShopBtn\_Click">Refresh</Button>  <Button Name="goShopBtn" DockPanel.Dock="Right" Style="{StaticResource ControlBtnStyle}" Click="goShopBtn\_Click">Go</Button>  <ComboBox DockPanel.Dock="Left" Name="shopNameSrchCB" Style="{StaticResource comboboxStyle}" ItemsSource="{Binding shopsCollection}" DisplayMemberPath="name" SelectedValuePath="id"></ComboBox>  </DockPanel>  </UniformGrid>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ScrollViewer Width="250">  <ListView HorizontalAlignment="Stretch" Background="#FF0966DF" Foreground="#FFFCFDFF" Name="shopDetailsList" ItemsSource="{Binding shopsCollection}" Loaded="ListView\_Loaded" SelectionChanged="ListView\_SelectionChanged">  <ListView.View>  <GridView>  <GridViewColumn Width="250" Header="Shop List" DisplayMemberBinding="{Binding name}" />  </GridView>  </ListView.View>  </ListView>  </ScrollViewer>  </UniformGrid>  </DockPanel>    <UniformGrid Name="upinfo" Background="White">    <DockPanel LastChildFill="True">    <DockPanel DockPanel.Dock="Right" >  <Image DockPanel.Dock="top" Source="/ShoppingMallUI;component/Images/Shopping-Black-Friday1.jpg" Width="234" Height="138" />  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <DatePicker Name="feedDateDp" Visibility="Collapsed" SelectedDate="{x:Static sys:DateTime.Now}" ></DatePicker>  <Label Style="{StaticResource LblStyle}" VerticalContentAlignment="Bottom">Your Name</Label>  <TextBox Name="nameTb"></TextBox>  <Label Style="{StaticResource LblStyle}" VerticalContentAlignment="Bottom">Email</Label>  <TextBox Name="mailTb"></TextBox>  <Label Style="{StaticResource LblStyle}" VerticalContentAlignment="Bottom">Shop Rating (Out of 10)</Label>  <TextBox Name="ratingTb"></TextBox>  <Label Style="{StaticResource LblStyle}" VerticalContentAlignment="Bottom">Share your experience with all</Label>  <TextBox Name="feedbackTb"></TextBox>  <Label></Label>  <UniformGrid Rows="1">  <Button Name="resetshopFeedback" Style="{StaticResource ControlBtnStyle}" Click="resetshopFeedback\_Click">Reset</Button>  <Button Name="submitshopBtn" Style="{StaticResource ControlBtnStyle}" Click="submitshopBtn\_Click">Submit</Button>  </UniformGrid>  </UniformGrid>  </DockPanel>  <DockPanel LastChildFill="True">  <TextBox Name="shopNameTb" DockPanel.Dock="Top" HorizontalContentAlignment="Center" IsEnabled="False">Shop Name</TextBox>  <TextBlock Name="shopdetailsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" ScrollViewer.VerticalScrollBarVisibility="Visible">Details</TextBlock>  <Label DockPanel.Dock="Top" >Available Products :</Label>  <TextBlock Name="availableProductsTBlock" DockPanel.Dock="Top" Style="{StaticResource TextBlockStyle}" MouseDown="availableProducts\_MouseDown">Available Products</TextBlock>  <UniformGrid DockPanel.Dock="Bottom" Columns="1">  <ListView Name="shopFeedbackView" HorizontalAlignment="Stretch" ScrollViewer.VerticalScrollBarVisibility="Visible" ItemsSource="{Binding shopFeedbackCollection}" Loaded="Window\_Loaded" >  <ListView.View>  <GridView>  <!--<GridViewColumn Width="100" Header="Id" DisplayMemberBinding="{Binding id}" />-->  <GridViewColumn Width="100" Header="Date" DisplayMemberBinding="{Binding feedDate}" />  <GridViewColumn Width="150" Header="Name" DisplayMemberBinding="{Binding name}" />  <GridViewColumn Width="150" Header="Email" DisplayMemberBinding="{Binding email}" />  <GridViewColumn Width="70" Header="Rating" DisplayMemberBinding="{Binding rate}" />  <GridViewColumn Width="370" Header="Feedback" DisplayMemberBinding="{Binding feedback}" />  </GridView>  </ListView.View>  </ListView>  </UniformGrid>    </DockPanel>  </DockPanel>  </UniformGrid>  </DockPanel>  </UserControl> |

|  |
| --- |
| Shops.xmal.cs |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  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 System.Collections.ObjectModel;  using ShoppingMallData;  using ShoppingMallDb;  namespace ShoppingMall  {  /// <summary>  /// Interaction logic for Shops.xaml  /// </summary>  public partial class Shops : UserControl  {  ObservableCollection<ShopInfo> \_shopsCollection = new ObservableCollection<ShopInfo>();  public ObservableCollection<ShopInfo> shopsCollection  {  get  {  return \_shopsCollection;  }  }  public Shops()  {  InitializeComponent();  }  private void ListView\_Loaded(object sender, RoutedEventArgs e)  {    }  private void fetchShopData()  {  List<ShopInfo> shops = DbInteraction.GetAllShopList();  \_shopsCollection.Clear();    foreach (ShopInfo shop in shops)  {  \_shopsCollection.Add(shop);  }  }  private void ListView\_SelectionChanged(object sender, SelectionChangedEventArgs e)  {  if (shopDetailsList.SelectedIndex != -1)  {  ShopInfo shopInfoObj = \_shopsCollection.ElementAt(shopDetailsList.SelectedIndex);  List<ShopInfo> shops = DbInteraction.GetSelectedShopList(shopInfoObj);  shopNameTb.Text = shopInfoObj.name;  shopdetailsTBlock.Text = shopInfoObj.description;  availableProductsTBlock.Text = shopInfoObj.availableProduct;  GetSelectedshopfeedbackItem();  }  else  MessageBox.Show("SelectedIndex equals -1");  }  private void submitshopBtn\_Click(object sender, RoutedEventArgs e)  {    if (!nameTb.Text.Equals("") && !mailTb.Text.Equals("") && !ratingTb.Text.Equals("") && !feedbackTb.Text.Equals(""))  {  ShoppingMallData.FeedbackInfo newFeedback = new ShoppingMallData.FeedbackInfo();  newFeedback.id = GenerateId();  newFeedback.item = shopNameTb.Text;  newFeedback.feedDate = feedDateDp.SelectedDate.Value;  newFeedback.name = nameTb.Text;  newFeedback.email = mailTb.Text;  newFeedback.rate = ratingTb.Text;  newFeedback.feedback = feedbackTb.Text;  ShoppingMallDb.DbInteraction.DoEnterFeedback(newFeedback);  clearshopfeedbackFields();  //fetchFeedBackData();  GetSelectedshopfeedbackItem();  }  else  {  MessageBox.Show("Please Insert Info Properly");  }  }  private string GenerateId()  {  return DateTime.Now.ToOADate().ToString();  }  private void clearshopfeedbackFields()  {  nameTb.Text = mailTb.Text = ratingTb.Text = feedbackTb.Text = "";  }  private void resetshopFeedback\_Click(object sender, RoutedEventArgs e)  {  clearshopfeedbackFields();  }  private void refreshShopBtn\_Click(object sender, RoutedEventArgs e)  {  fetchShopData();  }  private void goShopBtn\_Click(object sender, RoutedEventArgs e)  {  //shopDetailsList.UnselectAll();  if (shopNameSrchCB.Text == "")  fetchShopData();  else  {  ShopInfo shopInfo = new ShopInfo();  shopInfo.name = shopNameSrchCB.Text;  List<ShopInfo> shops = DbInteraction.searchShopList(shopInfo);  \_shopsCollection.Clear();  foreach (ShopInfo shop in shops)  {  \_shopsCollection.Add(shop);  }  }  }    private void UserControl\_Loaded(object sender, RoutedEventArgs e)  {  fetchShopData();  }  private void Window\_Loaded(object sender, RoutedEventArgs e)  {    }  #region Display only selected Shop Feedback  ObservableCollection<FeedbackInfo> \_shopFeedbackCollection = new ObservableCollection<FeedbackInfo>();  public ObservableCollection<FeedbackInfo> shopFeedbackCollection  {  get  {  return \_shopFeedbackCollection;  }  }  private void GetSelectedshopfeedbackItem()  {  FeedbackInfo shopInfo = new FeedbackInfo();  shopInfo.name = shopNameTb.Text;  List<FeedbackInfo> shops = DbInteraction.getshopFeedbackList(shopInfo);  \_shopFeedbackCollection.Clear();  foreach (FeedbackInfo shop in shops)  {  \_shopFeedbackCollection.Add(shop);  }  }  #endregion  private void availableProducts\_MouseDown(object sender, RoutedEventArgs e)  {  MessageBox.Show("Go To Product");  }    }  } |

|  |
| --- |
| AssemblyInfo.cs |
| using System.Reflection;  using System.Resources;  using System.Runtime.CompilerServices;  using System.Runtime.InteropServices;  using System.Windows;  // General Information about an assembly is controlled through the following  // set of attributes. Change these attribute values to modify the information  // associated with an assembly.  [assembly: AssemblyTitle("ShoppingMall")]  [assembly: AssemblyDescription("")]  [assembly: AssemblyConfiguration("")]  [assembly: AssemblyCompany("")]  [assembly: AssemblyProduct("ShoppingMall")]  [assembly: AssemblyCopyright("Copyright © 2013")]  [assembly: AssemblyTrademark("")]  [assembly: AssemblyCulture("")]  // Setting ComVisible to false makes the types in this assembly not visible  // to COM components. If you need to access a type in this assembly from  // COM, set the ComVisible attribute to true on that type.  [assembly: ComVisible(false)]  //In order to begin building localizable applications, set  //<UICulture>CultureYouAreCodingWith</UICulture> in your .csproj file  //inside a <PropertyGroup>. For example, if you are using US english  //in your source files, set the <UICulture> to en-US. Then uncomment  //the NeutralResourceLanguage attribute below. Update the "en-US" in  //the line below to match the UICulture setting in the project file.  //[assembly: NeutralResourcesLanguage("en-US", UltimateResourceFallbackLocation.Satellite)]  [assembly: ThemeInfo(  ResourceDictionaryLocation.None, //where theme specific resource dictionaries are located  //(used if a resource is not found in the page,  // or application resource dictionaries)  ResourceDictionaryLocation.SourceAssembly //where the generic resource dictionary is located  //(used if a resource is not found in the page,  // app, or any theme specific resource dictionaries)  )]  // Version information for an assembly consists of the following four values:  //  // Major Version  // Minor Version  // Build Number  // Revision  //  // You can specify all the values or you can default the Build and Revision Numbers  // by using the '\*' as shown below:  // [assembly: AssemblyVersion("1.0.\*")]  [assembly: AssemblyVersion("1.0.0.0")]  [assembly: AssemblyFileVersion("1.0.0.0")] |

|  |
| --- |
| Resources.Designer.cs |
| //------------------------------------------------------------------------------  // <auto-generated>  // This code was generated by a tool.  // Runtime Version:4.0.30319.1  //  // Changes to this file may cause incorrect behavior and will be lost if  // the code is regenerated.  // </auto-generated>  //------------------------------------------------------------------------------  namespace ShoppingMall.Properties  {  /// <summary>  /// A strongly-typed resource class, for looking up localized strings, etc.  /// </summary>  // This class was auto-generated by the StronglyTypedResourceBuilder  // class via a tool like ResGen or Visual Studio.  // To add or remove a member, edit your .ResX file then rerun ResGen  // with the /str option, or rebuild your VS project.  [global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Tools.StronglyTypedResourceBuilder", "4.0.0.0")]  [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]  [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]  internal class Resources  {  private static global::System.Resources.ResourceManager resourceMan;  private static global::System.Globalization.CultureInfo resourceCulture;  [global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.Performance", "CA1811:AvoidUncalledPrivateCode")]  internal Resources()  {  }  /// <summary>  /// Returns the cached ResourceManager instance used by this class.  /// </summary>  [global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]  internal static global::System.Resources.ResourceManager ResourceManager  {  get  {  if ((resourceMan == null))  {  global::System.Resources.ResourceManager temp = new global::System.Resources.ResourceManager("ShoppingMall.Properties.Resources", typeof(Resources).Assembly);  resourceMan = temp;  }  return resourceMan;  }  }  /// <summary>  /// Overrides the current thread's CurrentUICulture property for all  /// resource lookups using this strongly typed resource class.  /// </summary>  [global::System.ComponentModel.EditorBrowsableAttribute(global::System.ComponentModel.EditorBrowsableState.Advanced)]  internal static global::System.Globalization.CultureInfo Culture  {  get  {  return resourceCulture;  }  set  {  resourceCulture = value;  }  }  }  } |

|  |
| --- |
| Setting.cs |
| namespace ShoppingMall.Properties {      // This class allows you to handle specific events on the settings class:  // The SettingChanging event is raised before a setting's value is changed.  // The PropertyChanged event is raised after a setting's value is changed.  // The SettingsLoaded event is raised after the setting values are loaded.  // The SettingsSaving event is raised before the setting values are saved.  internal sealed partial class Settings {    public Settings() {  // // To add event handlers for saving and changing settings, uncomment the lines below:  //  // this.SettingChanging += this.SettingChangingEventHandler;  //  // this.SettingsSaving += this.SettingsSavingEventHandler;  //  }    private void SettingChangingEventHandler(object sender, System.Configuration.SettingChangingEventArgs e) {  // Add code to handle the SettingChangingEvent event here.  }    private void SettingsSavingEventHandler(object sender, System.ComponentModel.CancelEventArgs e) {  // Add code to handle the SettingsSaving event here.  }  }  } |

|  |
| --- |
| Setting.Designer.cs |
| //------------------------------------------------------------------------------  // <auto-generated>  // This code was generated by a tool.  // Runtime Version:4.0.30319.1  //  // Changes to this file may cause incorrect behavior and will be lost if  // the code is regenerated.  // </auto-generated>  //------------------------------------------------------------------------------  namespace ShoppingMall.Properties  {  [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]  [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator", "10.0.0.0")]  internal sealed partial class Settings : global::System.Configuration.ApplicationSettingsBase  {  private static Settings defaultInstance = ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new Settings())));  public static Settings Default  {  get  {  return defaultInstance;  }  }  }  } |

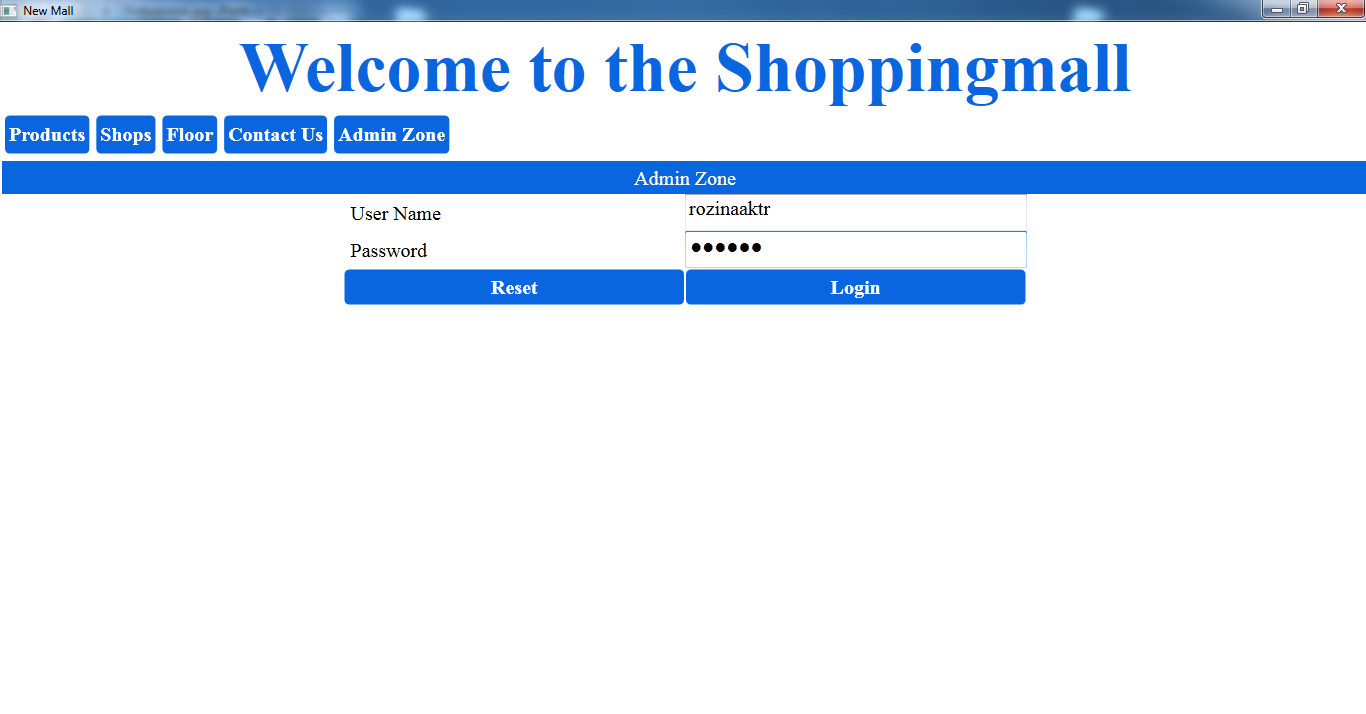
## PROCEDURAL DESIGN / OBJECT ORIENTED DESIGN

## USER INTERFACE DESIGN

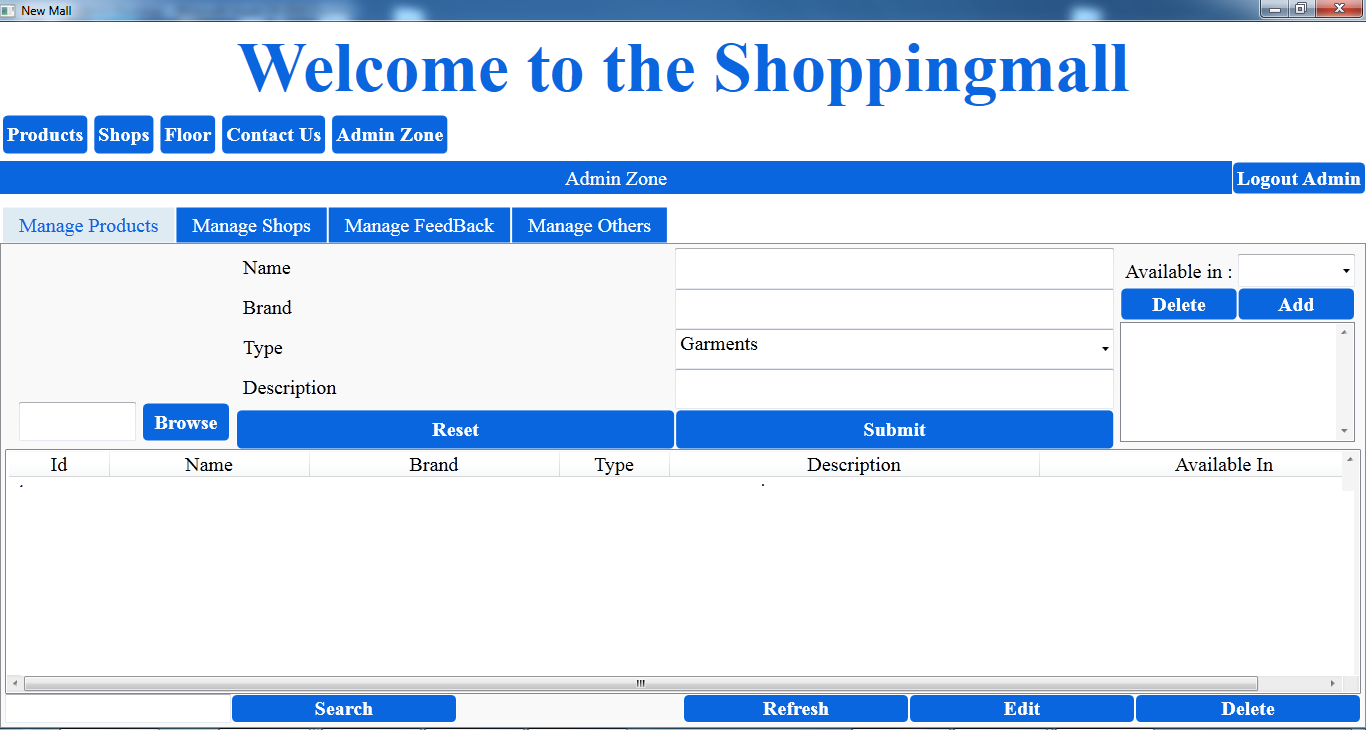
#### For Windows(C#.net)



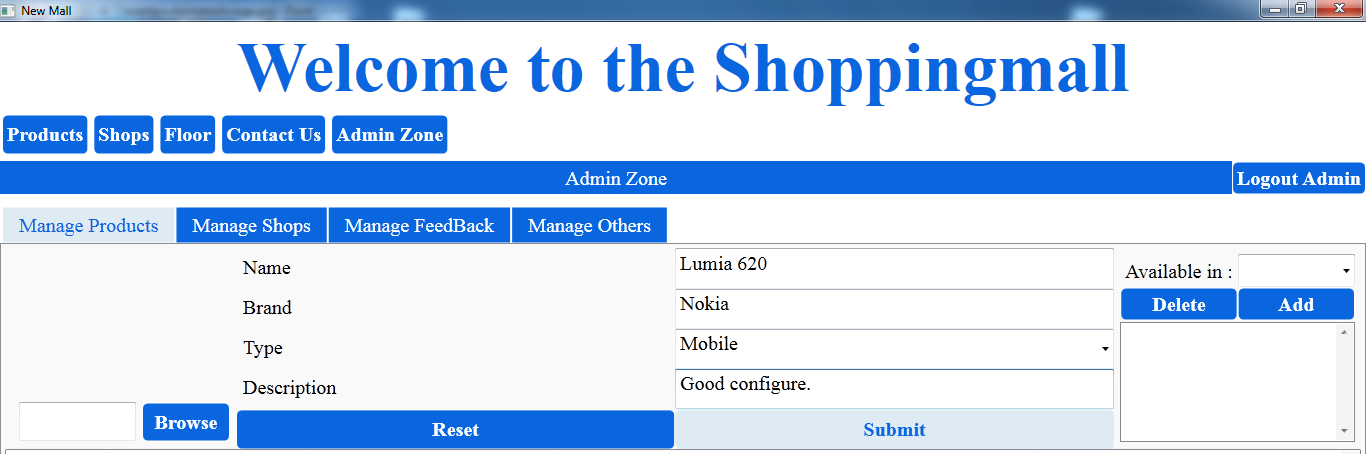
Main window



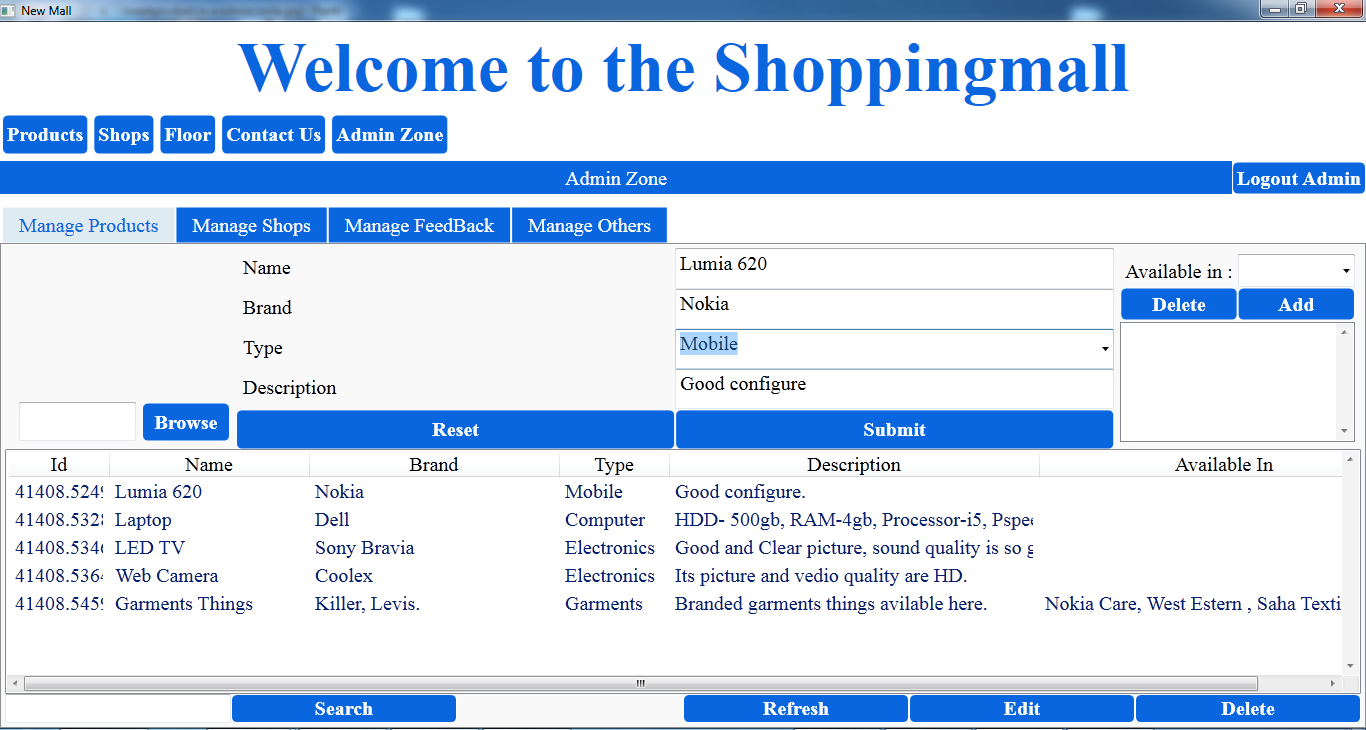
Login window



After login



Add Products



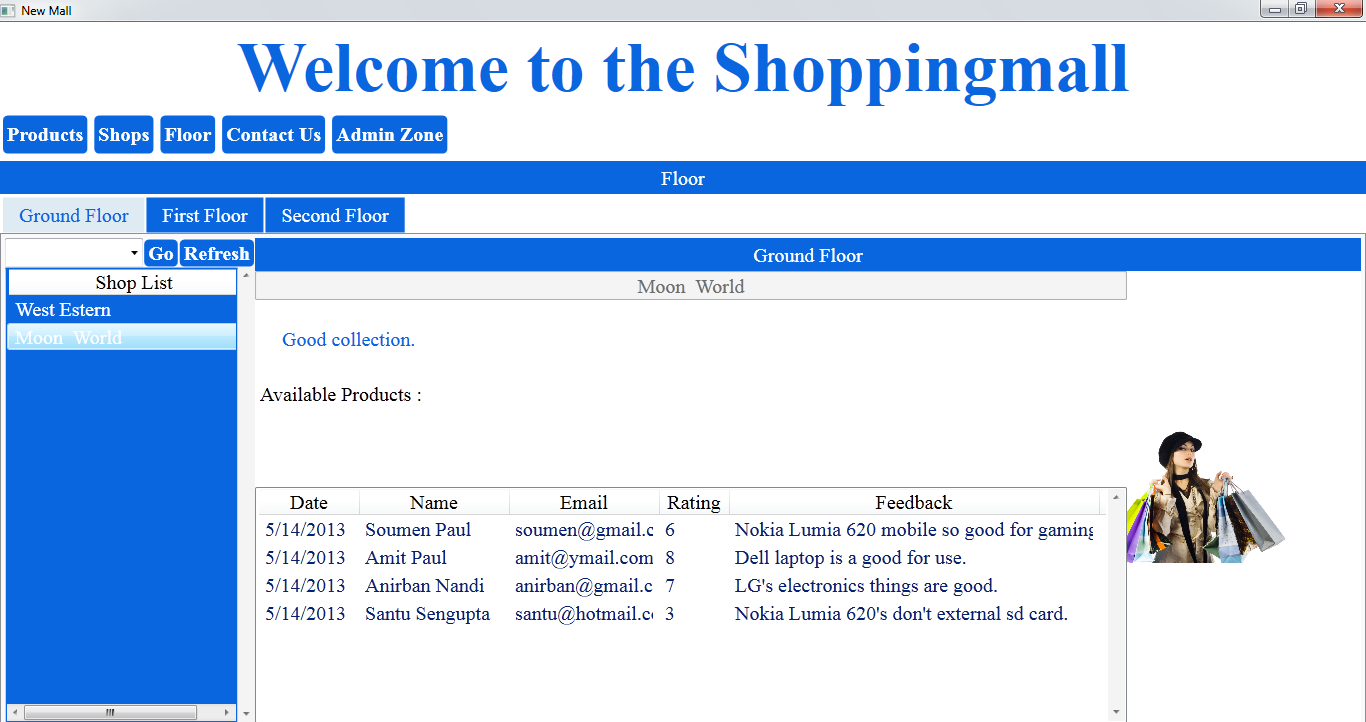
View Products Details



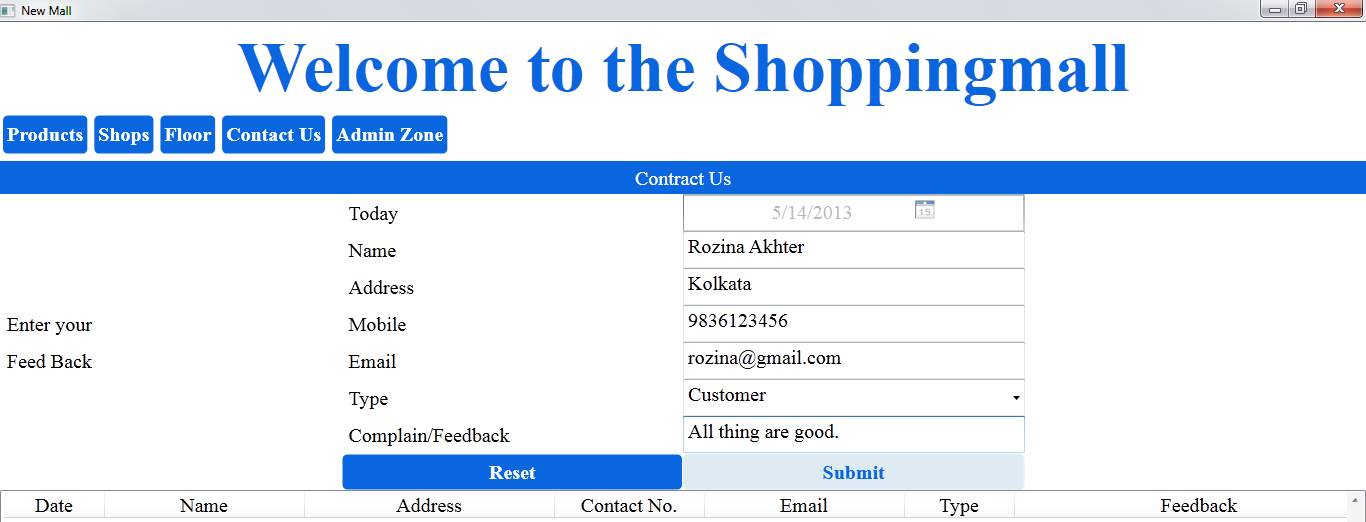
Add Shops



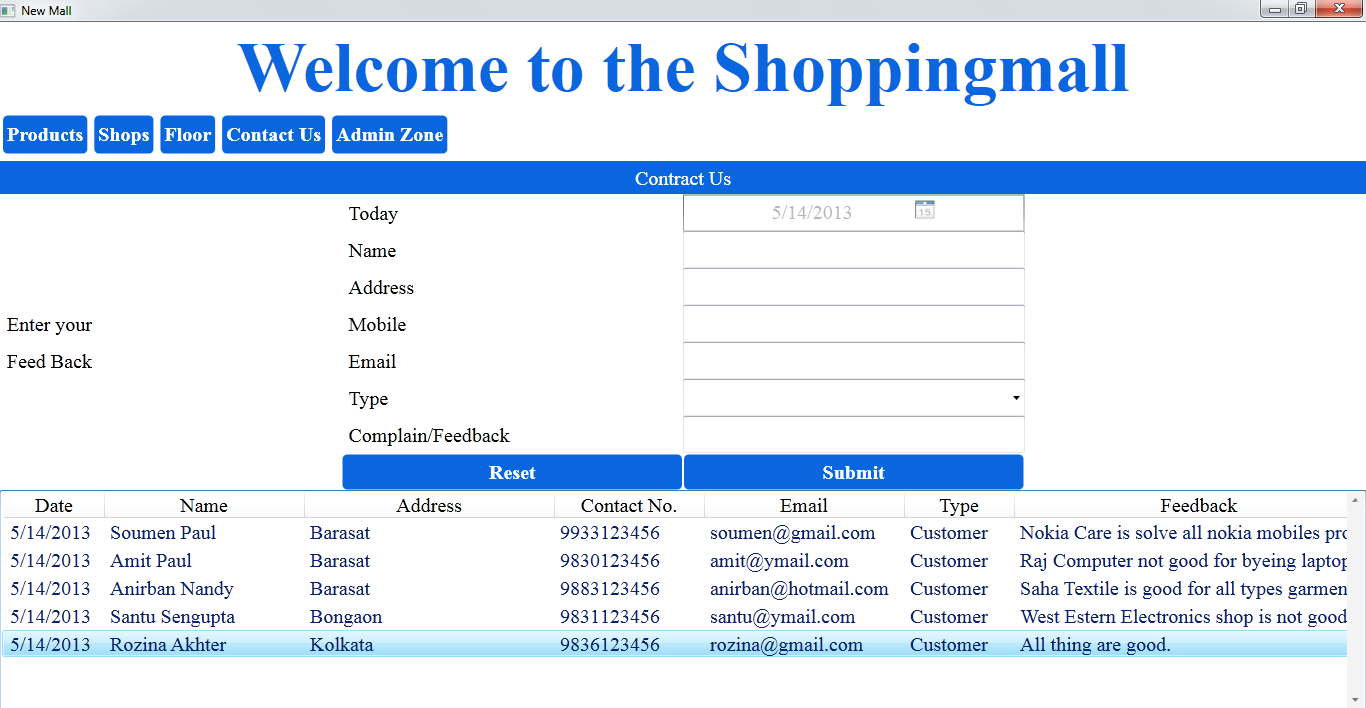
View shops details



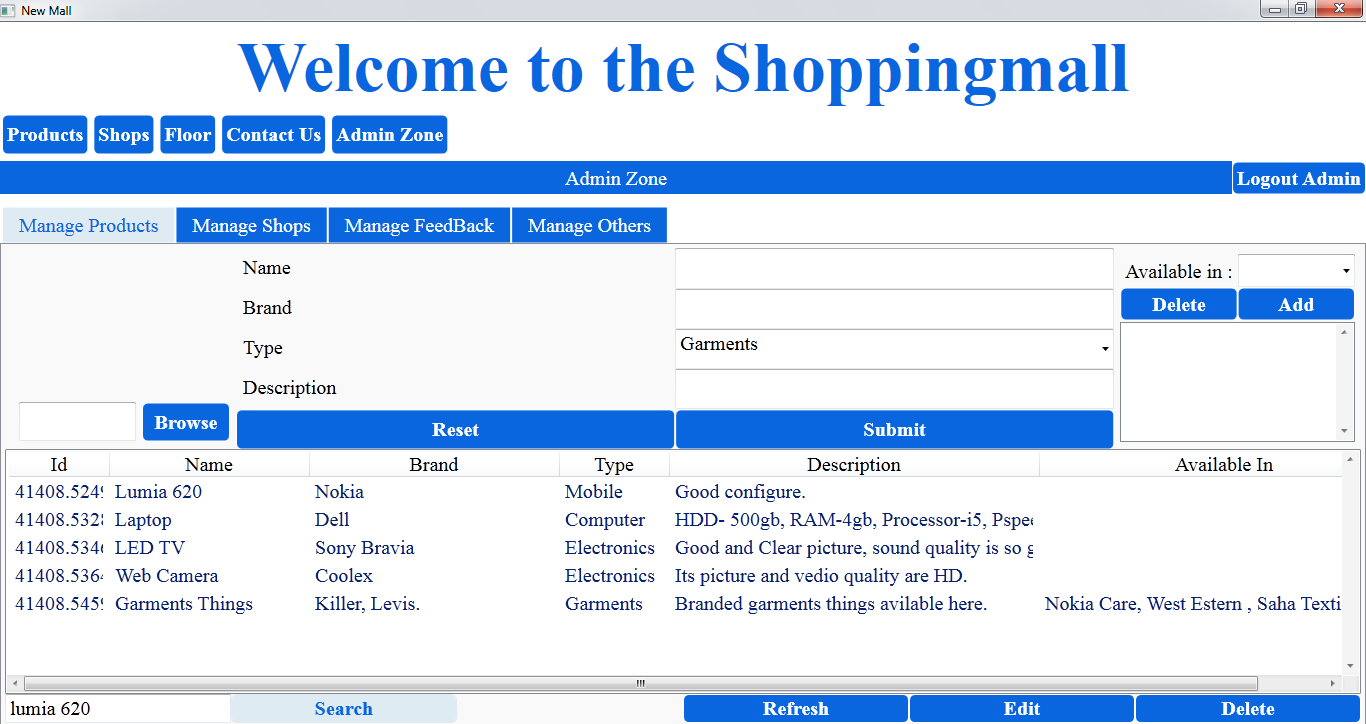
Floor Details View



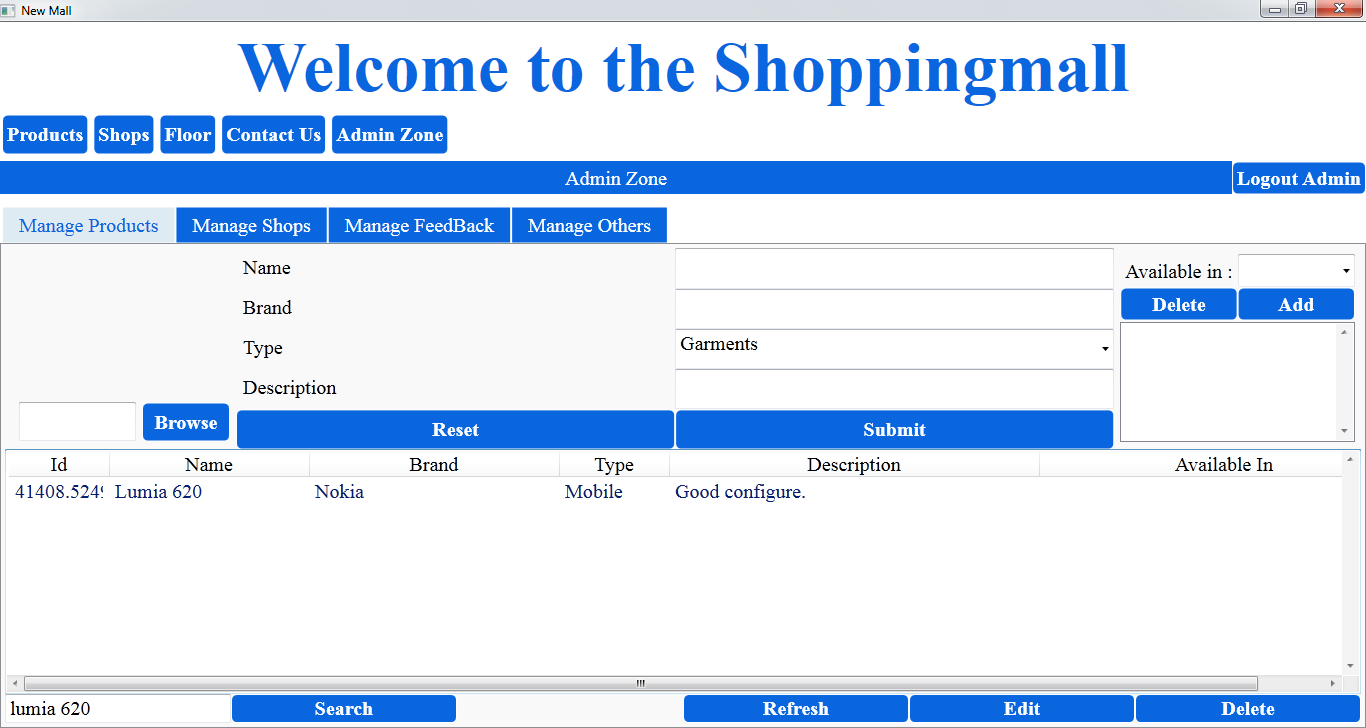
Add contuct us



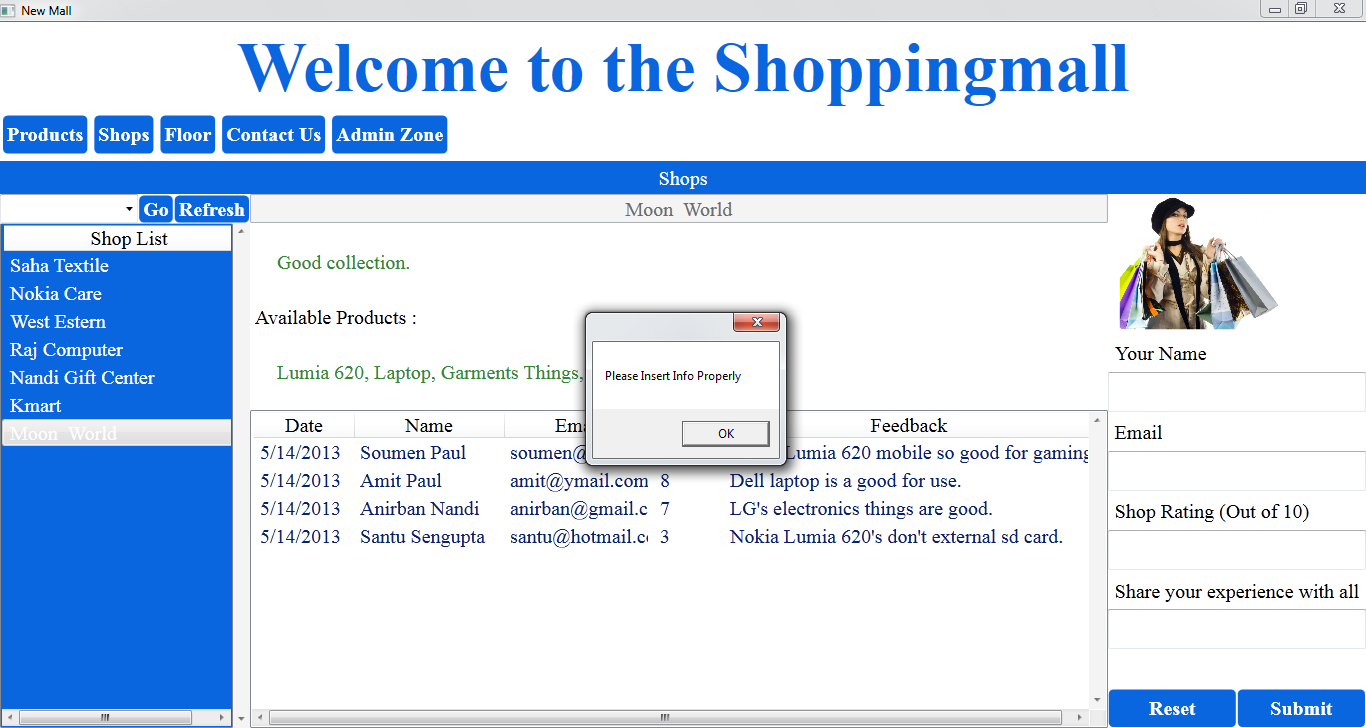
Contuct us details view



Insert search



View search value



without value error massage