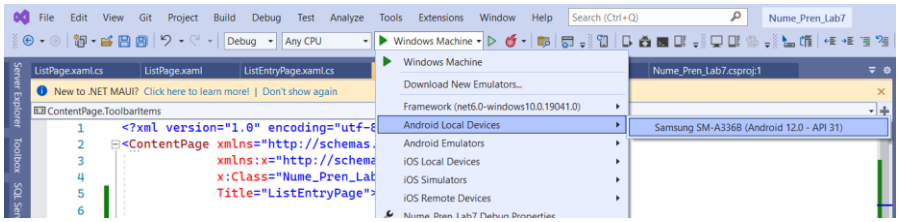
Laborator 8 - Aplicatii Multi-platforma cu .NET MAUI

1. Pentru a rula aplicatia realizata in laboratorul 7 putem sa utilizam dispozitivul fizic (telefon mobil cu SO Android) in locul emulatorului utilizat in lab 7 astfel:

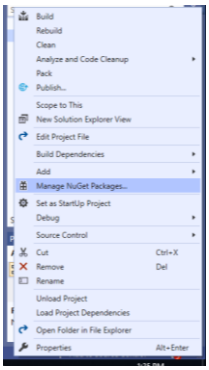
   - Setam telefonul mobil in mod debugging : deschidem optiunea Setari->Despre telefon-> Infomatii software-> si apoi se apasa pe sectiunea “numar versiune” de sapte ori. Vom primi un mesaj de confirmare – “Modul dezvoltator a fost activat”
   - Conectam telefonul mobil cu un cablu USB la laptop (daca suntem intrebati permitem transferul de fisiere)
   - De pe telefon accesam Setari->Optiuni dezvoltator->Remedierea erorilor prin USB (setam pe On/Activ)
   - Daca totul este setat corect vom vedea in Visual Studio numele dispozitivului fizic sub meniul Android Local Devices. Apasam Click pe numele dispozitivului si asteptam pana cand se face deploymentul pe telefonul fizic. Cand deploymentul este realizat, vom putea utiliza/testa aplicatia realizata pe telefon. Aplicatia se va deschide automat dupa ce se termina operatia de deployment sau putem deschide noi aplicatia ca orice alta aplicatie instalata pe telefon.



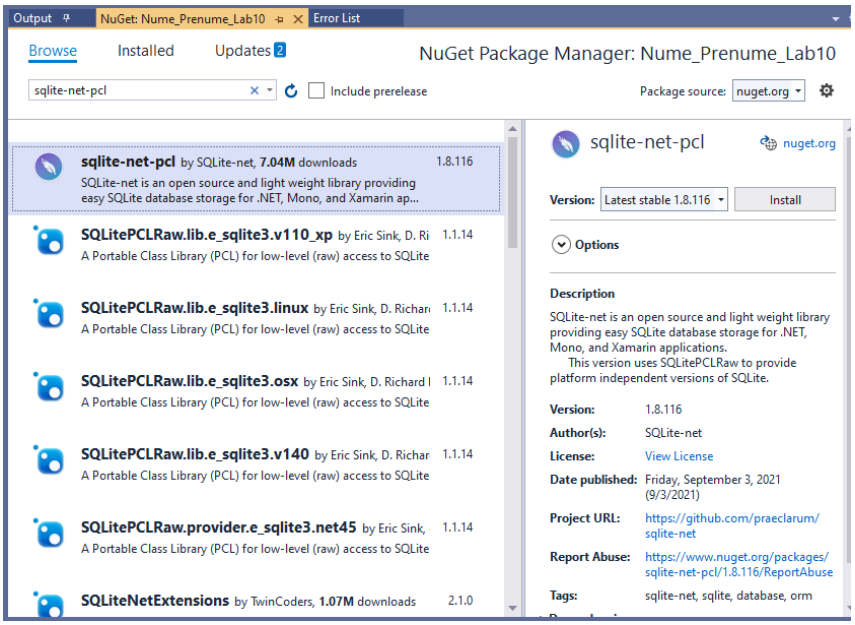
2. In laboratorul curent, vom dezvolta aplicatia realizata in laboratorul 7. Deschidem Visual Studio si apoi alegem optiunea **Open a project or solution** si cautam proiectul **Nume_Pren_Lab7,** creat anterior. Laboratorul curent il vom dezvolta pe un branch nou. Utilizand pasii indicati in Lab 1, pct. 22 vom crea un branch nou (based on master) pe care il vom denumi Laborator8.
3. Adaugam un nou folder in proiectul Nume_Pren_Lab7 - click dreapta pe numele proiectului – Add->New Folder pe care il denumim **Models**
4. Adaugam o clasa in folderul nou creat: **Models** astfel: click-dreapta pe numele folderului Add->Class si ii dam numele **ShopList** ,
5. Modificam fisierul creat marcand clasa ca fiind publica apoi adaugam urmatoarele proprietati:

```
namespace Nume_Pren_Lab7.Models
{
    public class ShopList
    {
        public int ID { get; set; }
        public string Description { get; set; }
        public DateTime Date { get; set; }

    }
}
```

6. Pentru a lucra cu baza de date locala SQLite.NET este necesara adaugarea pachetului **sqlite-net-pcl. Facem click dreapta pe numele proiectului Nume_Pren_Lab7 si alegem Manage NuGet packages**



7. In tab-ul Browse scriem numele pachetului, il selectam din lista afisata si apoi apasam Install



8. Adaugam directiva using SQLite si adnotarile [PrimaryKey], [AutoIncrement], [MaxLenght] si [Unique]:

```
using SQLite;
namespace Nume_Pren_Lab7.Models
{
    public class ShopList
    {
[PrimaryKey, AutoIncrement]
      public int ID { get; set; }
[MaxLength(250), Unique]
      public string Description { get; set; }
      public DateTime Date { get; set; }
```

```
        }
    }
```

Aceasta clasa defineste un model ShopList care va fi folosit pentru a salva fiecare lista de cumparaturi in aplicatia noastra. Proprietatea ID este marcata ca fiind `PrimaryKey` si `AutoIncrement` pentru a ne asigura ca fiecare lista de cumparaturi din aplicatia noastra va avea un Id unic furnizat de SQLite. Lungimea maxima pentru descrierea listei de cumparaturi va fi de 250 caractere, iar descrierea trebuie sa fie unica.

9. In fereastra Solution Explorer adaugam un nou folder pentru proiectul Nume_Pren_Lab10 - click dreapta pe numele proiectului –Add->New Folder pe care il denumim **Data**
10. Adaugam o clasa in folderul **Data** astfel: click-dreapta pe numele folderului Add->Class si ii dam numele **ShoppingListDatabase**. Aceasta clasa va contine cod pentru crearea, citirea, scrierea si stergerea datelor. Utilizam API-uri SQLite asincrone pentru a pune operatiile pe baza de date in thread-uri de background. Constructorul clasei ShoppingListDatabase ia ca si argument calea catre fisierul de tip baza de date. Aceasta cale este furnizata de clasa App in pasul urmator.

Clasa **ShoppingListDatabase** va avea urmatorul continut:

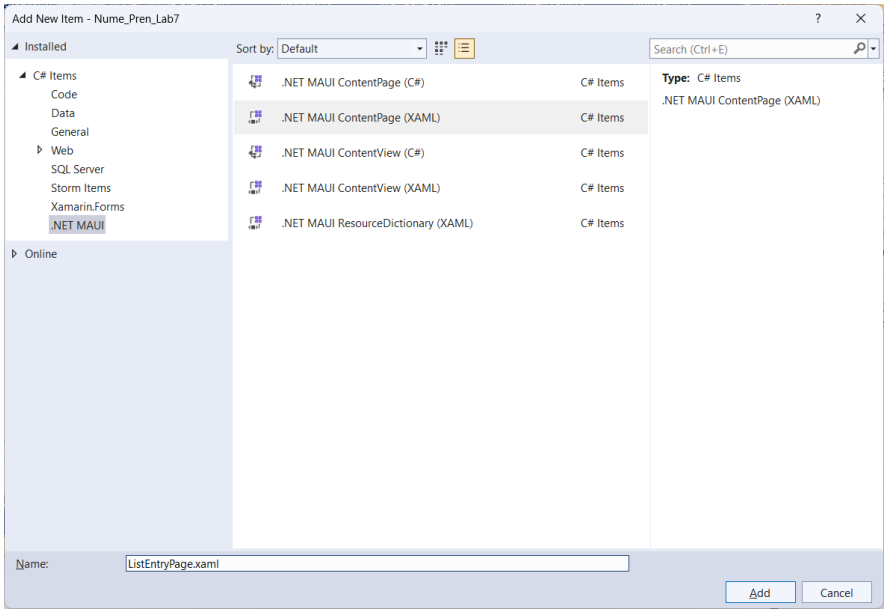
```
using SQLite;
using System.Collections.Generic;
using System.Threading.Tasks;
using Nume_Pren_Lab7.Models;

namespace Nume_Pren_Lab7.Data
{
    public class ShoppingListDatabase
    {
        readonly SQLiteAsyncConnection _database;

        public ShoppingListDatabase(string dbPath)
        {
            _database = new SQLiteAsyncConnection(dbPath);
            _database.CreateTableAsync<ShopList>().Wait();
        }

        public Task<List<ShopList>> GetShopListsAsync()
        {
            return _database.Table<ShopList>().ToListAsync();
        }

        public Task<ShopList> GetShopListAsync(int id)
        {
            return _database.Table<ShopList>()
                            .Where(i => i.ID == id)
                            .FirstOrDefaultAsync();
        }

        public Task<int> SaveShopListAsync(ShopList slist)
        {
            if (slist.ID != 0)
            {
                return _database.UpdateAsync(slist);
```

```
            }
            else
            {
                return _database.InsertAsync(slist);
            }
        }

        public Task<int> DeleteShopListAsync(ShopList slist)
        {
            return _database.DeleteAsync(slist);
        }
}
   }
```

11. In fereastra **Solution Explorer** deschidem fisierul **App.xaml.cs** din proiectul **Nume_Pren_Lab7** si adaugam urmatorul continut:

```
using System;
using Nume_Pren_Lab7.Data;
using System.IO;

namespace Nume_Pren_Lab7

    public partial class App : Application
    {
        static ShoppingListDatabase database;

        public static ShoppingListDatabase Database
        {
            get
            {
                if (database == null)
                {
                    database = new
ShoppingListDatabase(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.
LocalApplicationData), "ShoppingList.db3"));
                }
                return database;
            }
        }

        public App()
        {
            InitializeComponent();

           MainPage = new AppShell();
           }
```

12. In proiectul Nume_Pren_Lab7 adaugam o pagina de tip ContentPage. In fereastra Solution Explorer facem click dreapta pe numele proiectului Nume_Pren_Lab7, selectam Add->New Item si din categoria .NETMAUI alegem ContentPage pe care o denumim **ListEntryPage.xaml**



13. Vom utiliza un ListView care utilizeaza Databinding pentru a afisa toate listele de cumparaturi din aplicatie, selectarea uneia dintre aceste liste va duce la ListPage care permite modificarea acesteia. Vom folosi ToolbarItem pentru a crea o noua nota la apasarea acestuia. Din fereastra **Solution Explorer** deschidem fisierul ListEntryPage.xaml si modificam continutul acestuia astfel:

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Nume_Pren_Lab7.ListEntryPage"
             Title="ListEntryPage">
<ContentPage.ToolbarItems>
        <ToolbarItem Text="Add Shopping List"
                     Clicked="OnShopListAddedClicked" />
    </ContentPage.ToolbarItems>
    <ListView x:Name="listView"
              Margin="20"
              ItemSelected="OnListViewItemSelected">
        <ListView.ItemTemplate>
            <DataTemplate>
                <TextCell Text="{Binding Description}"
                          Detail="{Binding Date}" />
            </DataTemplate>
        </ListView.ItemTemplate>
    </ListView>
      </ContentPage>
```

14. Din fereastra **Solution Explorer** deschidem fisierul ListEntryPage.xaml.cs si adaugam la continutul acestuia codul de mai jos:

```
public partial class ListEntryPage : ContentPage
{
    public ListEntryPage()
    {
            InitializeComponent();
    }

protected override async void OnAppearing()
        {
            base.OnAppearing();

            listView.ItemsSource = await App.Database.GetShopListsAsync();
        }

async void OnShopListAddedClicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new ListPage
            {
                BindingContext = new ShopList()
            });
        }

async void OnListViewItemSelected(object sender, SelectedItemChangedEventArgs e)
        {
            if (e.SelectedItem != null)
            {
                await Navigation.PushAsync(new ListPage
                {
                    BindingContext = e.SelectedItem as ShopList
                });
            }

        }
}
```

15. In proiectul Nume_Pren_Lab7 adaugam o pagina de tip ContentPage. In fereastra Solution Explorer facem click dreapta pe numele proiectului Nume_Pren_Lab7, selectam Add->New Item si din categoria .NET MAUI alegem ContentPage (XAML) pe care o denumim **ListPage.xaml**
16. Vom utiliza un Editor pentru a introduce textul si doua butoane Save si Delete. Cele doua butoane sunt dispuse orizontal intr-un grid cu doua coloane.

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="Nume_Pren_Lab7.ListPage"
             Title="ListPage">
<StackLayout Margin="20">
        <Editor Placeholder="Enter the description of the shopping list"
                Text="{Binding Description}"
                HeightRequest="100" />
        <Grid>
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*" />
                <ColumnDefinition Width="*" />
```

```
            </Grid.ColumnDefinitions>
            <Button Text="Save"
                    Clicked="OnSaveButtonClicked" />
            <Button Grid.Column="1"
                    Text="Delete"
                    Clicked="OnDeleteButtonClicked"/>
        </Grid>
    </StackLayout>
</ContentPage>
```

17. Din fereastra **Solution Explorer** deschidem fisierul ListPage.xaml.cs si adaugam la continutul acestuia codul de mai jos. Cand este apasat butonul Save, handler-ul de eveniment `OnSaveButtonClicked` este executat, lista de cumparaturi este salvata in baza de date si aplicatia navigheaza la pagina precedenta. La executarea handlerului de eveniment `OnDeleteButtonClicked` , lista de cumparaturi este stearsa din baza de date si aplicatia navigheaza la pagina precedenta .

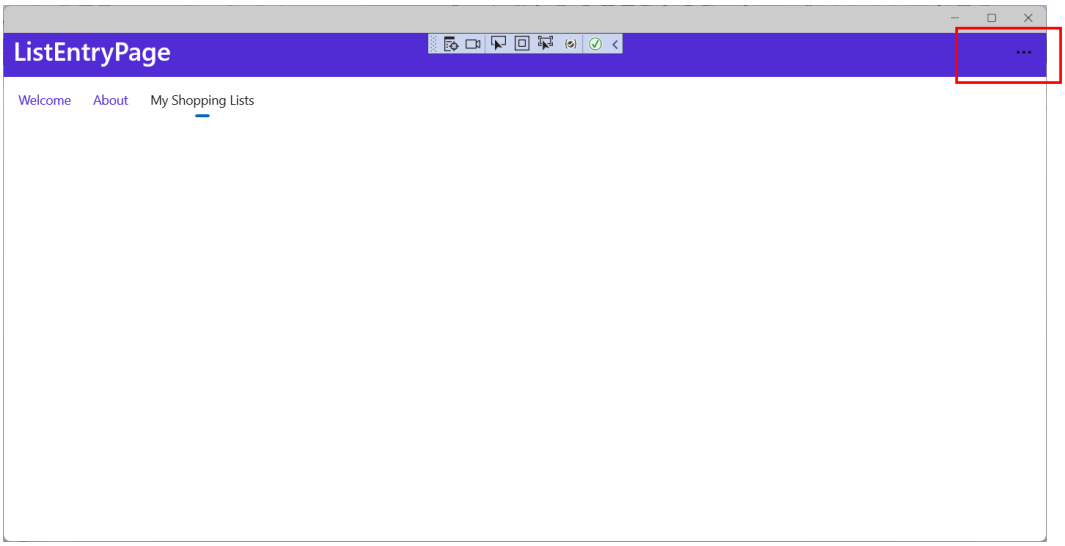
```
using Nume_Pren_Lab7.Models;
…
public partial class ListPage : ContentPage
{
       public ListPage()
       {
              InitializeComponent();
       }
async void OnSaveButtonClicked(object sender, EventArgs e)
{
    var slist = (ShopList)BindingContext;
    slist.Date = DateTime.UtcNow;
    await App.Database.SaveShopListAsync(slist);
    await Navigation.PopAsync();
}

async void OnDeleteButtonClicked(object sender, EventArgs e)
{
    var slist = (ShopList)BindingContext;
    await App.Database.DeleteShopListAsync(slist);
    await Navigation.PopAsync();
}
}
```

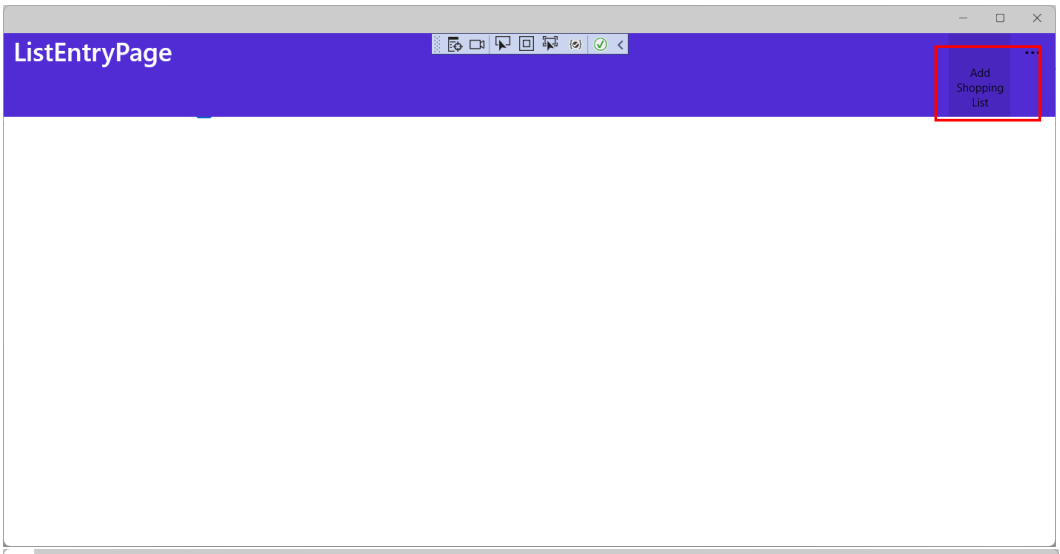
18. Deschidem fisierul AppShell.xaml si modificam codul astfel incat in TabBar sa apara noua pagina create ListEntryPage

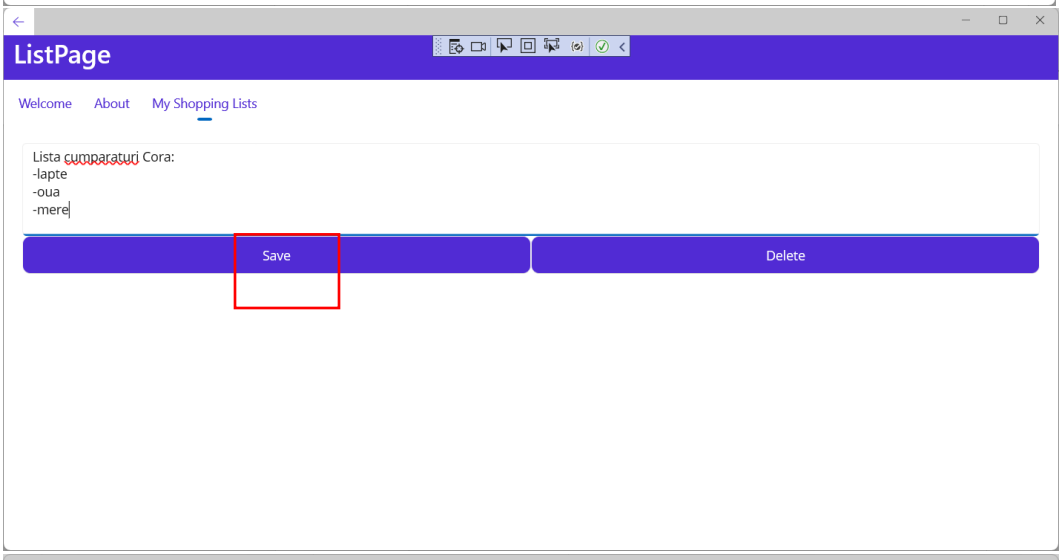
```
<?xml version="1.0" encoding="UTF-8" ?>
<Shell
    x:Class="Nume_Pren_Lab7.AppShell"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
```

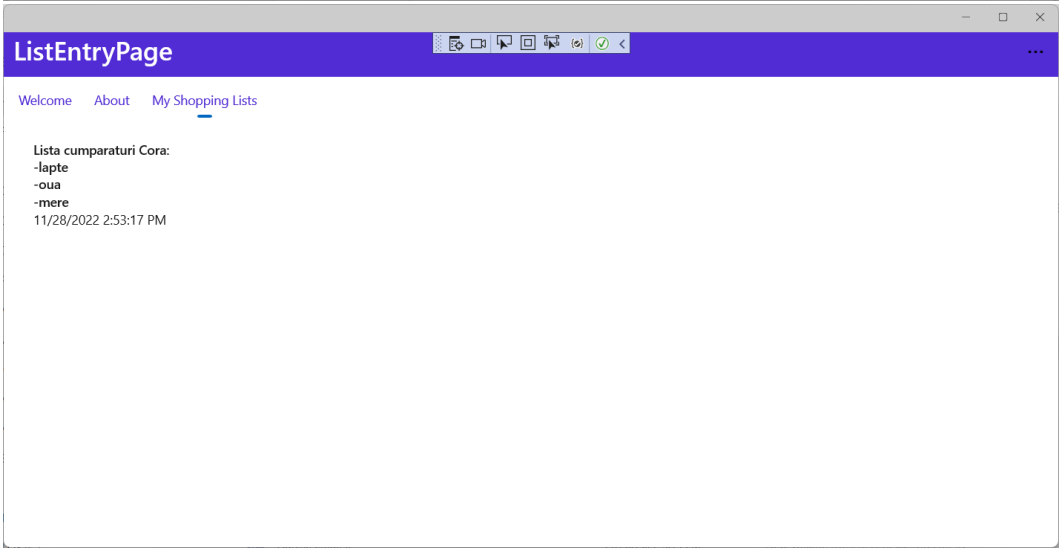
```
        xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
        xmlns:local="clr-namespace:Nume_Pren_Lab7"
        Shell.FlyoutBehavior="Disabled">

        <TabBar>
            <ShellContent
                Title="Welcome"
                ContentTemplate="{DataTemplate local:MainPage}"
                 />

            <ShellContent
                Title="About"
                ContentTemplate="{DataTemplate local:AboutPage}"
                 />

            <ShellContent
                Title="My Shopping Lists"
                ContentTemplate="{DataTemplate local:ListEntryPage}"
                 />
        </TabBar>

    </Shell>
    19.
```

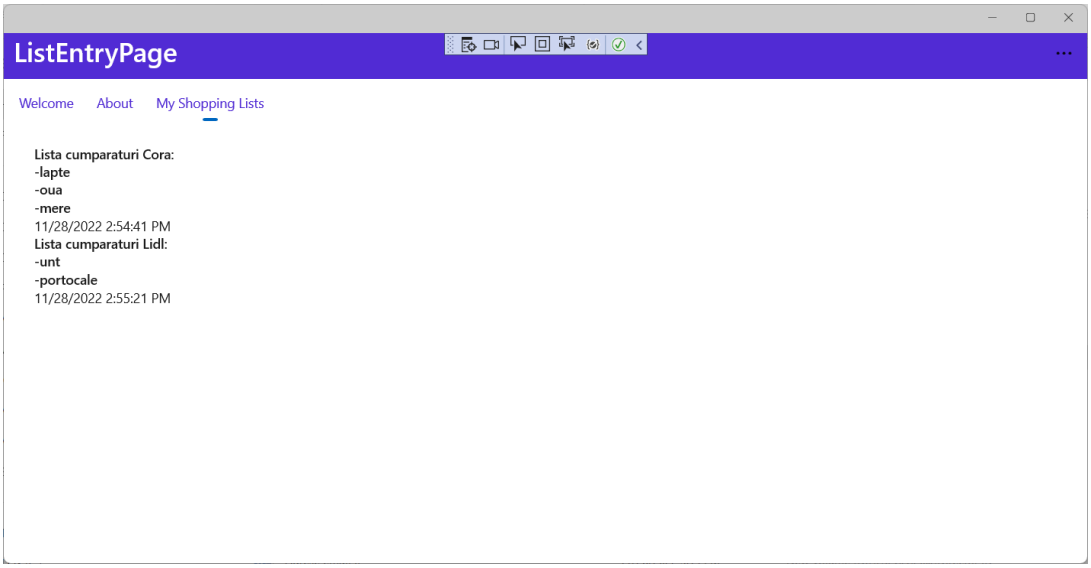
20. Din meniul Build alegem optiunea Build Solution si daca nu avem erori putem face deployment in emulator (Windows Machine) apasand click pe butonul verde iar apoi testam functionalitatile create




ListEntryPage
Add Shopping List


ListPage
Welcome
About
My Shopping Lists
Lista cumparaturi Cora:
-lapte
-oua
-mere
Save
Delete


ListEntryPage
Welcome
About
My Shopping Lists
Lista cumparaturi Cora:
-lapte
-oua
-mere
11/28/2022 2:53:17 PM


ListEntryPage
Welcome
About
My Shopping Lists
Lista cumparaturi Cora:
-lapte
-oua
-mere
11/28/2022 2:54:41 PM
Lista cumparaturi Lidl:
-unt
-portocale
11/28/2022 2:55:21 PM