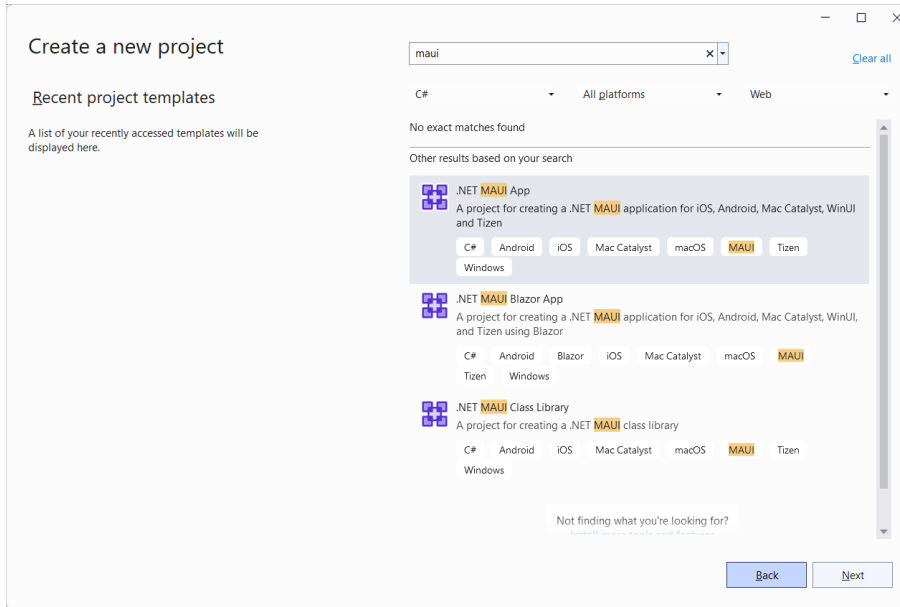
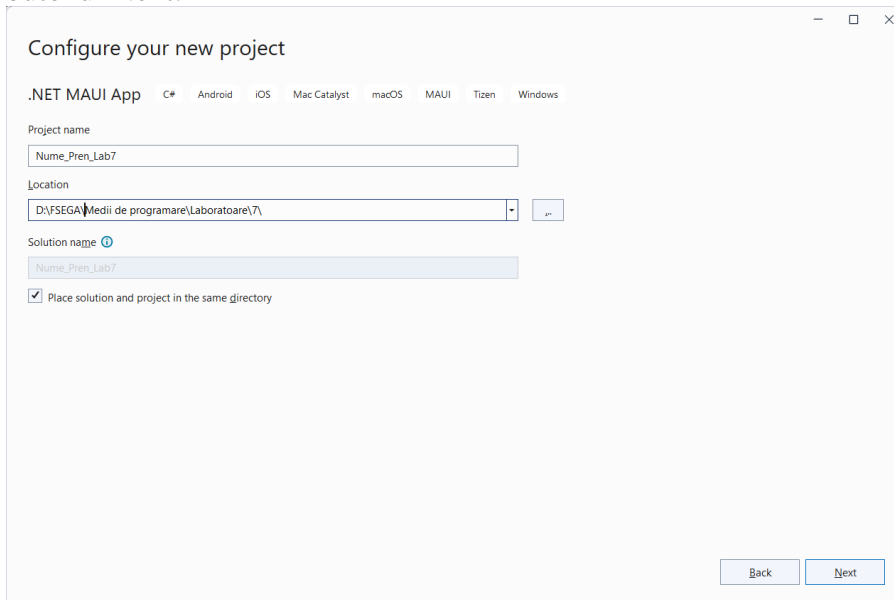


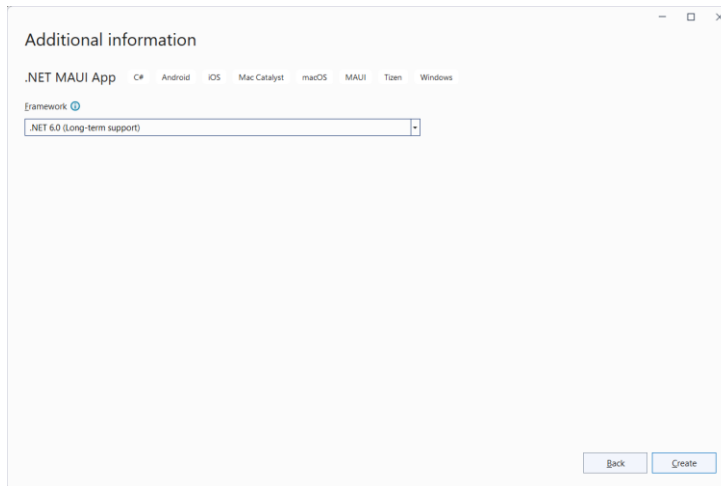
Laborator 7 - Aplicatii Multi-platforma cu .NET MAUI

1. In laboratorul curent, vom crea o aplicatie multi-platforma cu .NET MAUI si o vom rula utilizand Emulatorul Android. Deschidem Visual Studio și, în fereastra de pornire, faceți clic pe opțiunea *Create new project* pentru a crea un proiect nou.
2. În fereastra Create new project, selectați template-ul *.NET MAUI App* și faceți clic pe butonul *Next*:

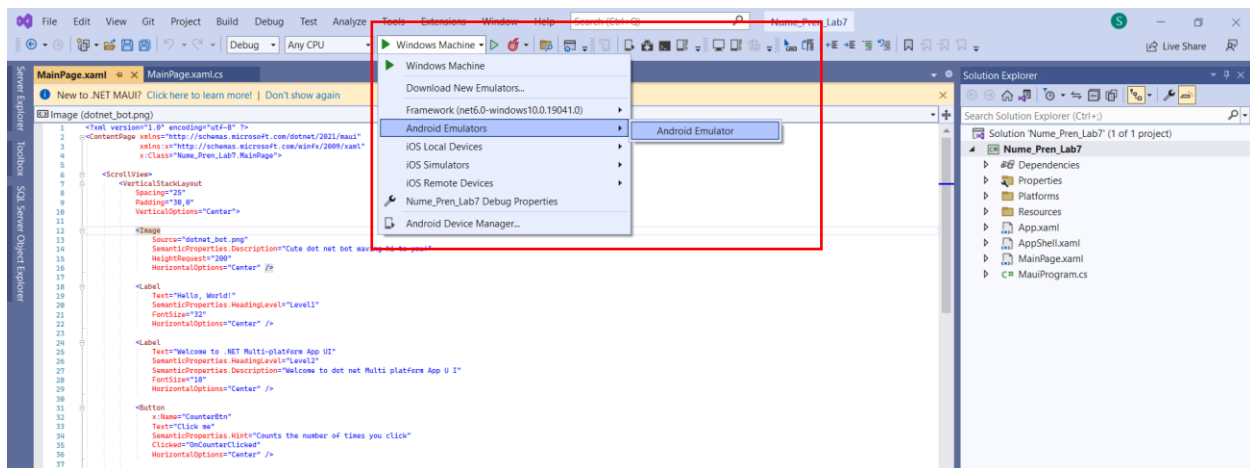


3. Denumiți proiectul **Nume_Pren_Lab7**, alegem o locație potrivită pentru acesta și facem clic pe butonul Next:

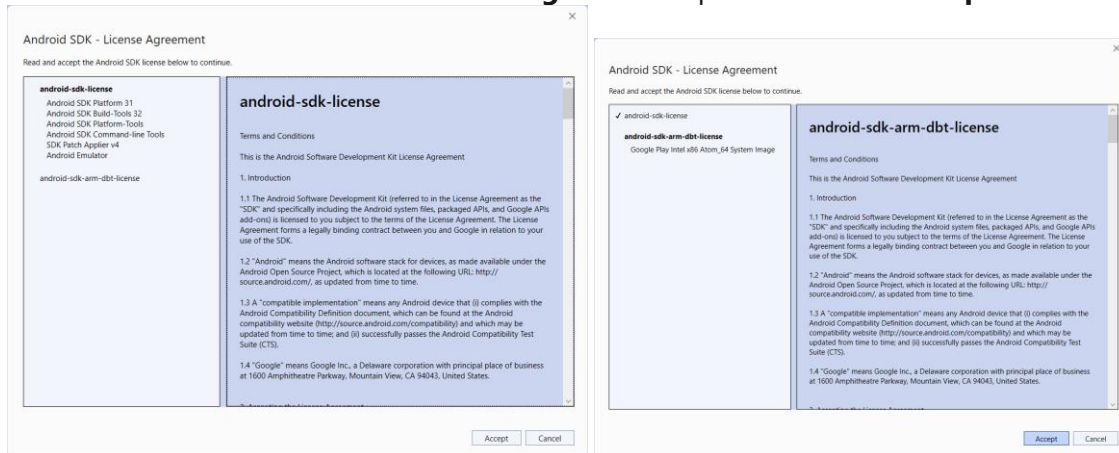




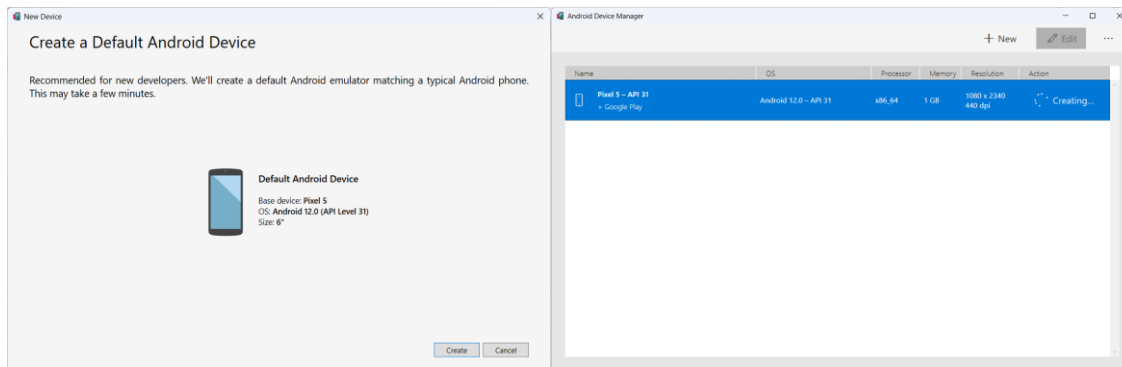
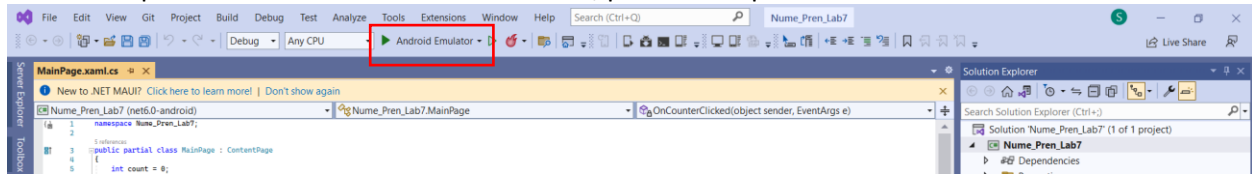
4. Pentru a putea rula aplicatia ca o aplicatie mobile vom utiliza un emulator pentru Android. In toolbar-ul Visual Studio, utilizam lista derulanta **Debug Target** pentru a selecta **Android Emulators** -> **Android Emulator** :



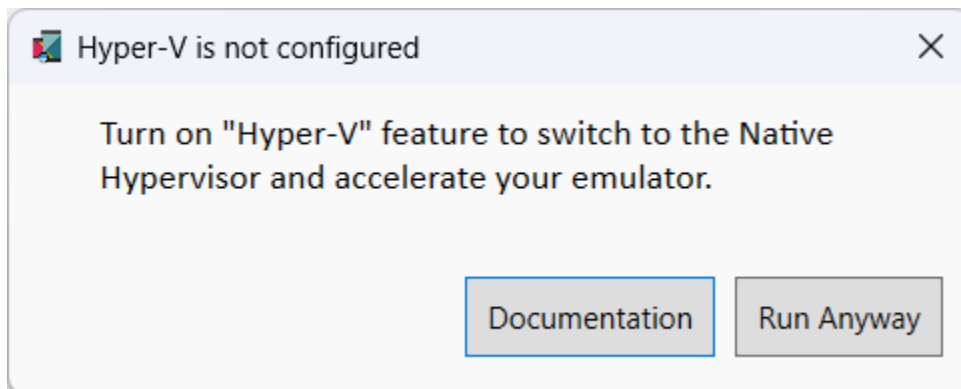
5. In fereastra **Android SDK - License Agreement** apasam butonul **Accept** :



6. Apasam click pe Android Emulator si se va deschide o noua fereastră in care vom apasa butonul Create pentru a crea un emulator de Android, proces care poate dura cateva minute:



Dupa terminarea procesului de configurare a emulatorului daca apasam pe buonul Start din fereastra Android Device Manager este posibil sa primim urmatorul mesaj:



7. In continuare vom utiliza funcțiile de accelerare hardware ale computerului pentru a maximiza performanța emulatorului Android.

Următoarele tehnologii de virtualizare sunt disponibile pentru accelerarea emulatorului Android:

- Hyper-V de la Microsoft și platforma Windows Hypervisor (WHPX) : Hyper-V este o caracteristică de virtualizare a Windows care face posibilă rularea sistemelor computerizate virtualizate pe un computer gazdă fizic.
- Managerul de execuție accelerată hardware de la Intel (HAXM) : HAXM este un motor de virtualizare pentru computere care rulează procesoare Intel.

Pentru cea mai bună experiență pe Windows, este recomandat să utilizați WHPX pentru a accelera emulatorul Android. Dacă WHPX nu este disponibil pe computer, atunci poate fi utilizat HAXM.

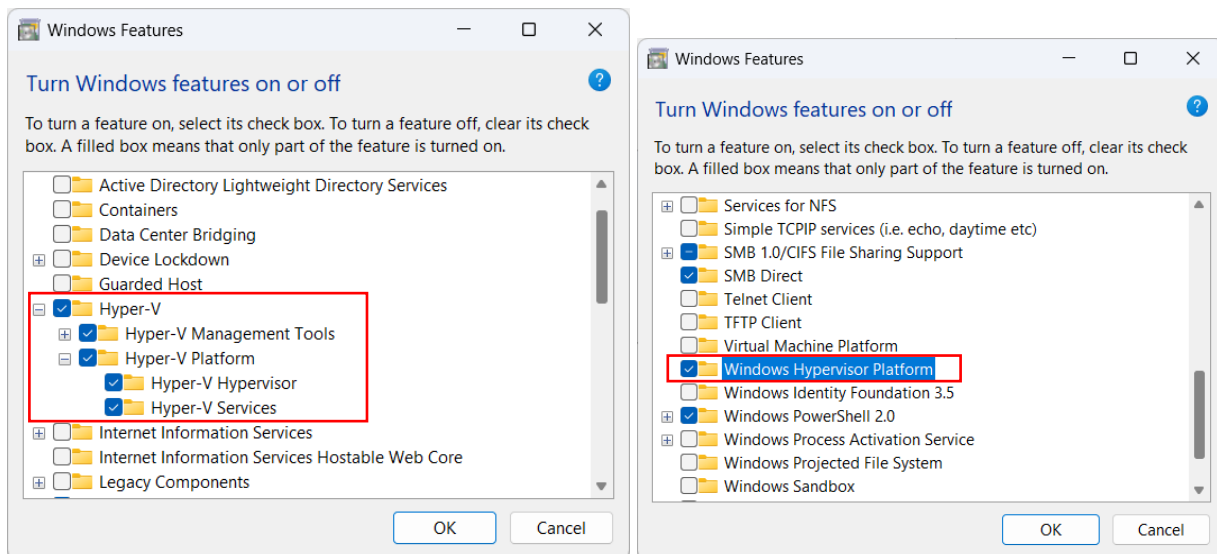
Pentru a verifica dacă hardware-ul și software-ul computerului vostru sunt compatibile cu Hyper-V, deschideți Command Prompt și tastați următoarea comandă:

```
systeminfo
```

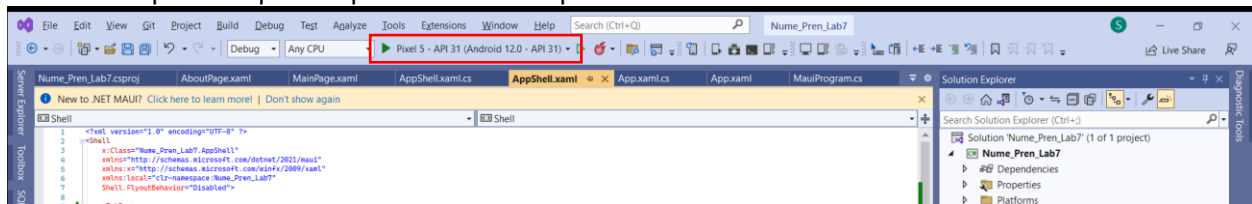
Ar trebui ca toate atributele să aibă valoarea "Yes" similar cu imaginea de mai jos.

```
Hyper-V Requirements:      VM Monitor Mode Extensions: Yes
                           Virtualization Enabled In Firmware: Yes
                           Second Level Address Translation: Yes
                           Data Execution Prevention Available: Yes
```

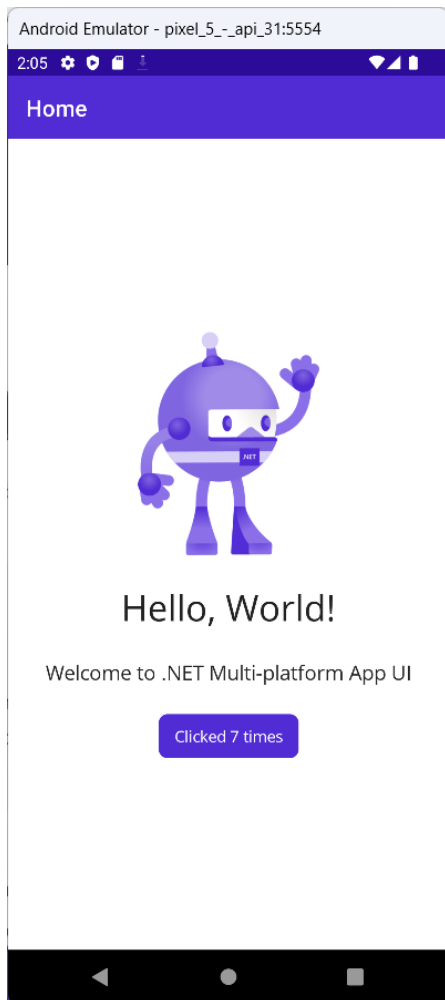
Pentru a activa Hyper-V scriem în search box-ul de la Windows **"windows features"** și din rezultatele afișate selectăm **Turn Windows features on or off**. În fereastra Windows Features bifăm opțiunea Hyper-V și Windows Hypervisor Platform apoi vom da restart la computer.



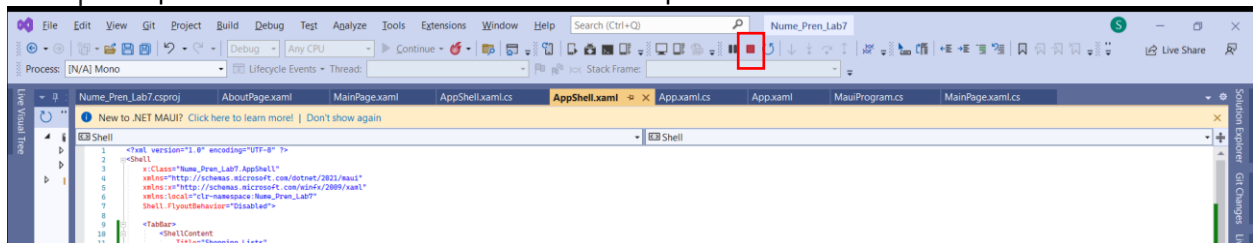
8. Rulam aplicația apăsând pe buonul unde apare numele emulatorului creat mai sus :



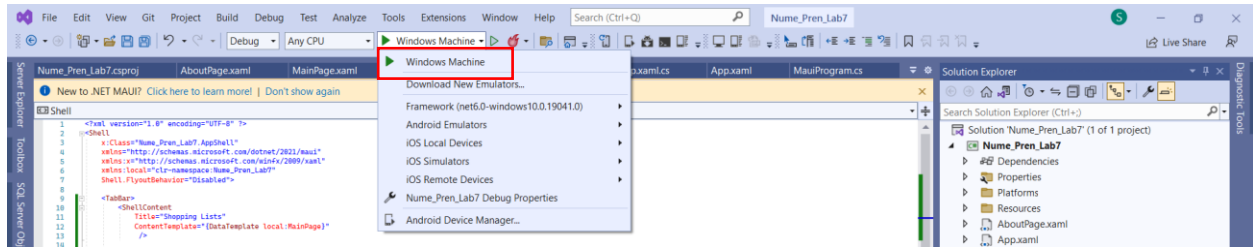
Asteptam pana se initializeaza emulatorul si se face deployment-ul aplicatiei pe emulator, iar apoi aplicatia se va deschide similar cu imaginea de mai jos. Daca facem click pe button se va afisa de cate ori am facut click.



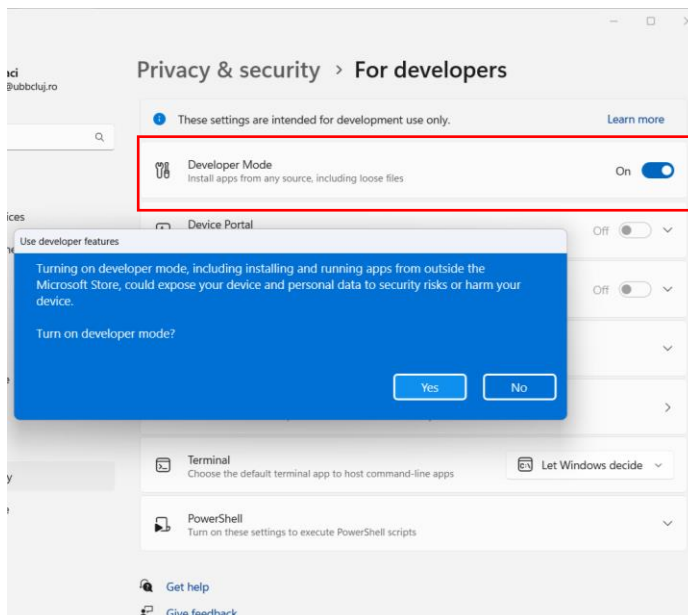
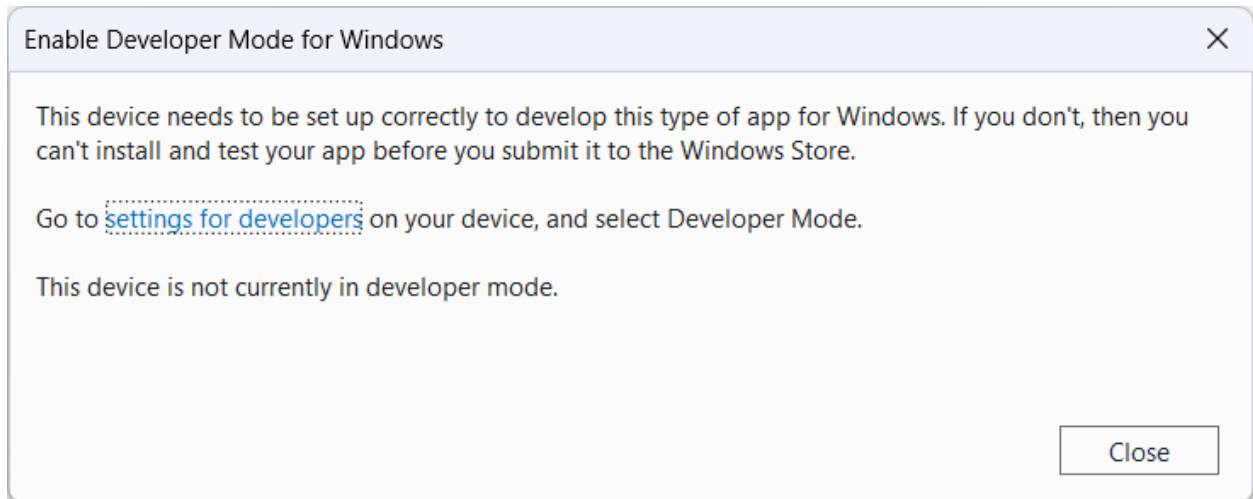
9. Oprim aplicatia din executie de la butonul rosu Stop din Visual Studio



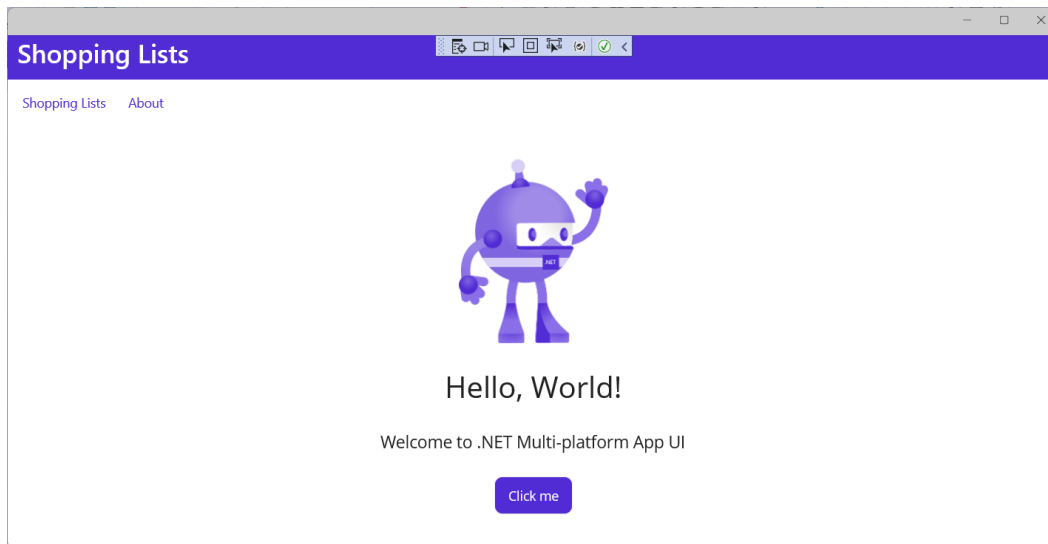
10. Pentru ca .NET MAUI creaza aplicatii multi-platforma putem rula aceeaasi aplicatie pe platforma Windows. Selectam din lista Windows Machine



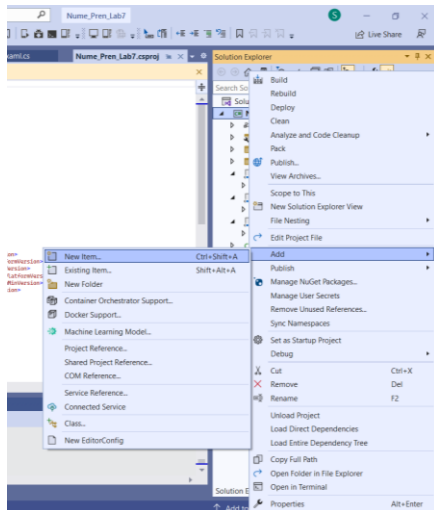
Se va deschide o ferestra similara cu cea de mai jos. Apasam click pe Settings for developers pentru a activa Developer Mode



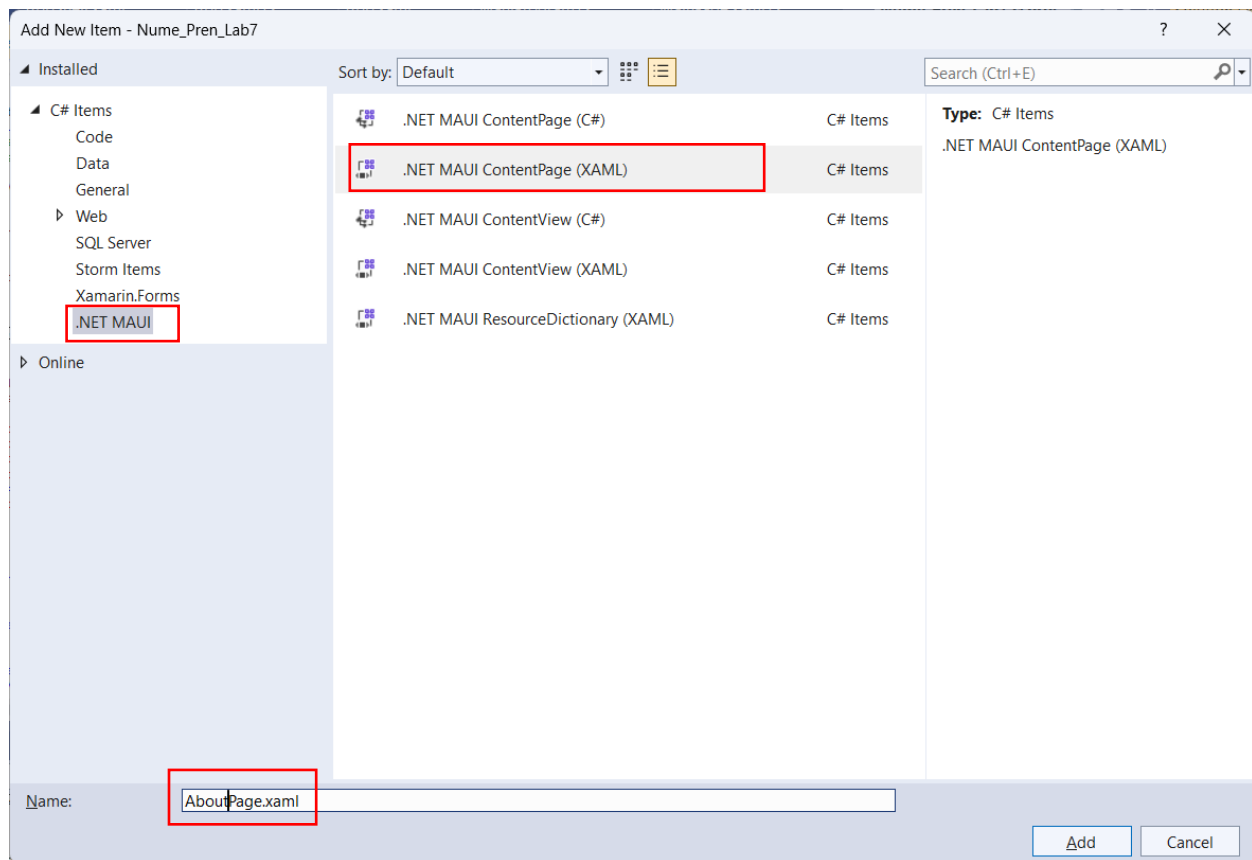
11. Rulam aplicatia si observam ca Layout-ul aplicatiei difera in functie de platforma pe care ruleaza:



12. Vom modifica aplicatia, adăugand o noua pagină la proiect. Această pagină este o pagină „About”, care reprezintă informații despre această aplicație, cum ar fi autorul, versiunea și un link pentru mai multe informații. In fereastra Solution Explorer apasam click dreapta pe numele solutiei ->Add->New Item



13. In noua fereastra care se deschide, selectam în partea stangă din lista de template-uri .NET MAUI, apoi selectam template-ul de pagina .NET MAUI ContentPage (XAML). Denumim pagina AboutPage.xaml și apoi apasam butonul Add.



14. Se va deschide un fisier AboutPage.xaml care contine markup XAML prin care este descrisa interfața de utilizare a paginii. Modificam fisierul AboutPage.xaml 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.AboutPage"
    Title="AboutPage">

    <VerticalStackLayout Spacing="10" Margin="10">
        <HorizontalStackLayout Spacing="10">
            <Image Source="shop_list.png"
                SemanticProperties.Description="My Shopping List"
                HeightRequest="64" />
            <Label FontSize="22" FontAttributes="Bold" Text="My Shopping
Lists" VerticalOptions="End" />
            <Label FontSize="22" Text="v1.0" VerticalOptions="End" />
        </HorizontalStackLayout>

        <Label Text="This app will help you to manage your shopping lists" />
    </VerticalStackLayout>
</ContentPage>
```

15. Descarcam fisierul shop_list.png de pe Moodle si il vom copia in locatia unde am salvat proiectul Nume_Pren_Lab7->Resources->Images

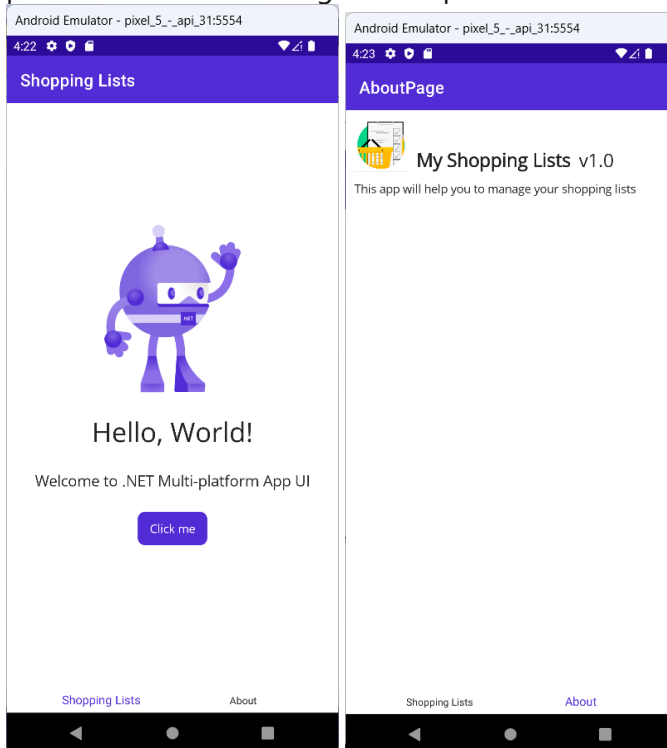
16. In fereastra Solution Explorer, faceți dublu clic pe fișierul AppShell.xaml pentru a deschide editorul XAML. Clasa AppShell definește ierarhia vizuală a unei aplicații. Actualizăm markup-ul XAML pentru a adăuga un control TabBar astfel:

```
<?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">

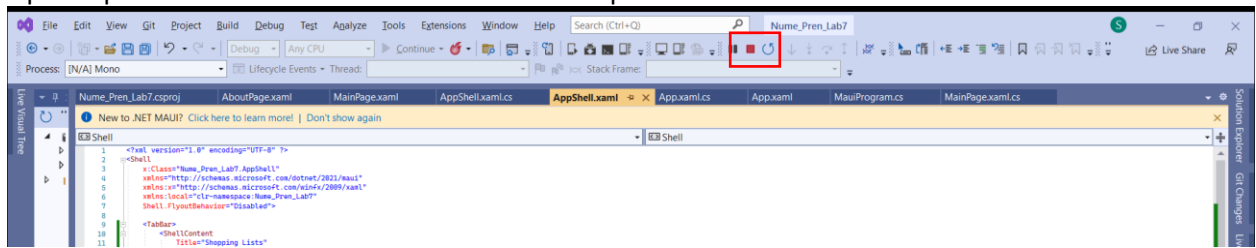
  <TabBar>
    <ShellContent
      Title="Shopping Lists"
      ContentTemplate="{DataTemplate local:MainPage}"
    />

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

17. Salvăm modificările în cele două fișiere și rulăm apoi aplicația și observăm faptul că am adăugat un TabBar în partea de jos a paginii și putem naviga spre noua pagină AboutPage pe care tocmai am adăugat-o în aplicația noastră.



18. Oprim aplicatia din executie de la butonul rosu Stop din Visual Studio



Sarcina laborator :

Personalizati pagina MainPage.xaml astfel incat fiecare student sa utilizeze o imagine si un text relevant pentru el.