Nagyszerű! Akkor lépésről lépésre megmutatom, **hogyan készíthetsz grafikus felületű (Windows Forms) desktop alkalmazást** Visual Studio 2022-ben C# nyelven.

**1. Új projekt létrehozása**

1. **Indítsd el a Visual Studio 2022-t**.
2. Kattints a **"Create a new project"** gombra.
3. A keresőbe írd be: Windows Forms App.
4. Válaszd ki:  
   **"Windows Forms App (.NET Framework)"** vagy  
   **"Windows Forms App (.NET)"** *(ajánlott, ha .NET 6/7/8-at használsz)*.
5. Kattints a **Next** gombra.
6. Add meg:
   * A projekt nevét
   * A mentési helyet
7. Kattints a **Create** gombra.

**2. Grafikus felület (UI) szerkesztése**

1. Megnyílik egy Form1.cs [Design] nevű lap.
2. Itt látsz egy üres ablakot (formot) – ez lesz az alkalmazás főablaka.

**Eszköztár megnyitása (Toolbox)**

1. A bal oldalon kattints a **"Toolbox"** fülre.
   * Ha nem látod: View > Toolbox vagy nyomd meg: **Ctrl + Alt + X**.
2. Húzd rá az ablakra az elemeket, pl.:
   * **Button** (gomb)
   * **Label** (szöveg)
   * **TextBox** (szövegbevitel)

**3. Elemek beállítása**

1. Kattints egy elemre (pl. Button1), majd a jobb oldali **Properties** ablakban:
   * Állítsd be a **Text** tulajdonságot pl.: OK gomb
   * Állítsd be a **Name** tulajdonságot pl.: btnOk

**4. Kód írása**

1. Duplán kattints a gombra (btnOk) – ez automatikusan létrehozza az eseménykezelőt.
2. Írd be a kívánt kódot pl.:

private void btnOk\_Click(object sender, EventArgs e)

{

   MessageBox.Show("Szia világ!");

}

**5. Alkalmazás futtatása**

1. Fent kattints a **zöld „Start” gombra** vagy nyomd meg az **F5** billentyűt.
2. Megjelenik az alkalmazás ablak, és a gomb működni fog.

Szeretnél képeket is ezekhez a lépésekhez, vagy inkább videós útmutatót? Tudok linkelni pontos, magyar nyelvű példákat.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**📁 1. CSV Fájl előkészítése**

A projekt mappájába (azonos helyen, ahol a .csproj fájl van) hozz létre egy fájlt:

**office\_costs\_2024.csv**

**Tartalma (példa):**

id;officename;datum;kategoria;osszeg;megjegyzes

1;Budapest HQ;2024-01-15;Maintenance;2500;Légkondi karbantartás

2;Debrecen Office;2024-03-10;Insurance;1800;

**🖼️ 2. Felület (UI) kialakítása**

A Form1.cs [Design] nézetben:

**Hozzáadandó elemek:**

| **Típus** | **Név** | **Megjegyzés** |
| --- | --- | --- |
| DataGridView | dataGridView1 | Táblázatos adatmegjelenítés |
| TextBox | txtOfficeName | Iroda neve |
| DateTimePicker | dtpDatum | Dátum |
| ComboBox | cmbKategoria | Költség típus |
| TextBox | txtOsszeg | Összeg |
| TextBox | txtMegjegyzes | Megjegyzés |
| Button | btnHozzaad | "Hozzáadás" gomb |

👉 **Ne felejts el címkéket (Label) is tenni a mezők mellé.**

**💡 3. Kód megírása**

Nyisd meg a Form1.cs-t, és írd be a következő kódot:

**A tetején:**

csharp

MásolásSzerkesztés

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Windows.Forms;

**Osztályon belül:**

public partial class Form1 : Form

{

string csvPath = "office\_costs\_2024.csv";

List<OfficeCost> costList = new List<OfficeCost>();

public Form1()

{

InitializeComponent();

cmbKategoria.Items.AddRange(new string[] { "Maintenance", "Repairs", "Insurance", "Utilities", "Other" });

dtpDatum.Format = DateTimePickerFormat.Custom;

dtpDatum.CustomFormat = "yyyy-MM-dd";

LoadCsv();

}

void LoadCsv()

{

if (!File.Exists(csvPath)) return;

var lines = File.ReadAllLines(csvPath, Encoding.UTF8).Skip(1); // Skip header

costList.Clear();

foreach (var line in lines)

{

var parts = line.Split(';');

costList.Add(new OfficeCost

{

Id = int.Parse(parts[0]),

OfficeName = parts[1],

Datum = DateTime.Parse(parts[2]),

Kategoria = parts[3],

Osszeg = decimal.Parse(parts[4]),

Megjegyzes = parts.Length > 5 ? parts[5] : ""

});

}

RefreshGrid();

}

void RefreshGrid()

{

dataGridView1.DataSource = null;

dataGridView1.DataSource = costList;

}

private void btnHozzaad\_Click(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(txtOfficeName.Text) || cmbKategoria.SelectedItem == null || !decimal.TryParse(txtOsszeg.Text, out decimal osszeg))

{

MessageBox.Show("Kérlek töltsd ki az összes kötelező mezőt.");

return;

}

int newId = costList.Any() ? costList.Max(x => x.Id) + 1 : 1;

OfficeCost newCost = new OfficeCost

{

Id = newId,

OfficeName = txtOfficeName.Text.Trim(),

Datum = dtpDatum.Value,

Kategoria = cmbKategoria.SelectedItem.ToString(),

Osszeg = osszeg,

Megjegyzes = txtMegjegyzes.Text.Trim()

};

costList.Add(newCost);

SaveToCsv();

RefreshGrid();

ClearForm();

}

void SaveToCsv()

{

var lines = new List<string> { "id;officename;datum;kategoria;osszeg;megjegyzes" };

foreach (var c in costList)

{

lines.Add($"{c.Id};{c.OfficeName};{c.Datum:yyyy-MM-dd};{c.Kategoria};{c.Osszeg};{c.Megjegyzes}");

}

File.WriteAllLines(csvPath, lines, Encoding.UTF8);

}

void ClearForm()

{

txtOfficeName.Clear();

txtOsszeg.Clear();

txtMegjegyzes.Clear();

cmbKategoria.SelectedIndex = -1;

dtpDatum.Value = DateTime.Today;

}

}

**Adatmodell (az osztály végén vagy külön fájlban):**

public class OfficeCost

{

public int Id { get; set; }

public string OfficeName { get; set; }

public DateTime Datum { get; set; }

public string Kategoria { get; set; }

public decimal Osszeg { get; set; }

public string Megjegyzes { get; set; }

}

**🧪 Tesztelés**

1. Futtasd az alkalmazást **F5**-tel vagy a **Start** gombbal.
2. Ellenőrizd, hogy a táblázat betölti-e a CSV fájl adatait.
3. Tölts ki egy új költségadatot, kattints a **Hozzáadás** gombra.
4. Ellenőrizd, hogy megjelenik-e az új adat:
   * a táblázatban
   * a CSV fájlban

**🎯 Kész!**

Ezzel az alkalmazás:

✅ Beolvassa a CSV fájlt  
✅ Megjeleníti az adatokat táblázatban  
✅ Lehetővé teszi új adatok hozzáadását  
✅ Ment minden adatot vissza a CSV fájlba  
✅ Automatikusan generál id-t minden új sorhoz

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Cél: Felhasználói felület létrehozása az OfficeSphere költségkezelő alkalmazásához

Ez a felület tartalmazza:

* **Táblázatot** az adatok megjelenítéséhez
* **Űrlapot** új költség felviteléhez

## ✅ Lépésenkénti útmutató (Design nézetben)

### 🔧 1. Nyisd meg a grafikus felületet

* Nyisd meg a Form1.cs fájlt.
* Kattints a Form1.cs [Design] fülre vagy jobb klikk a fájlon → View Designer.

### 📐 2. Húzd be az elemeket az Eszköztárból (Toolbox)

Ha nem látod a **Toolbox**-ot:

* Menüből: View → Toolbox
* Vagy: Ctrl+Alt+X

#### Helyezd el az alábbi vezérlőelemeket:

| **Vezérlő** | **Név** | **Leírás** |
| --- | --- | --- |
| **DataGridView** | dataGridView1 | Adatok megjelenítése |
| **Label** | - | Címke: "Iroda neve:" |
| **TextBox** | txtOfficeName | Iroda neve |
| **Label** | - | Címke: "Dátum:" |
| **DateTimePicker** | dtpDatum | Költség dátuma |
| **Label** | - | Címke: "Költség típusa:" |
| **ComboBox** | cmbKategoria | Költség kategória |
| **Label** | - | Címke: "Összeg (€):" |
| **TextBox** | txtOsszeg | Költség összege |
| **Label** | - | Címke: "Megjegyzés:" |
| **TextBox** | txtMegjegyzes | Opcionális megjegyzés |
| **Button** | btnHozzaad | Felirat: "Hozzáadás" gomb |

### 📏 3. Elemsorrend és elhelyezés

A javasolt sorrend a formon:

#### Felső rész:

* **DataGridView** (foglalja el a bal oldal nagy részét – pl. 600×300 méret)

#### Alsó rész vagy jobb oldal:

* Címkék + mezők egymás alatt:

css

MásolásSzerkesztés

[ Iroda neve: ][ txtOfficeName ]

[ Dátum: ][ dtpDatum ]

[ Költség típusa: ][ cmbKategoria ]

[ Összeg (€): ][ txtOsszeg ]

[ Megjegyzés: ][ txtMegjegyzes ]

[ ][ btnHozzaad ]

Tipp: MultiLine beállítás txtMegjegyzes-nél = true + állítsd magasabbra.

## 🔗 4. Beállítások (Properties panelben)

### dataGridView1

* Name: dataGridView1
* ReadOnly: true
* AllowUserToAddRows: false
* AutoSizeColumnsMode: Fill

### cmbKategoria

* Name: cmbKategoria
* Kézzel is hozzáadhatod az elemeket a Items property-ben:
  + Maintenance
  + Repairs
  + Insurance
  + Utilities
  + Other

VAGY: kódon keresztül, lásd lejjebb.

## 🔘 5. Gombhoz esemény (Hozzáadás gomb)

### Kattints duplán a ****Hozzáadás**** gombra

Ez létrehozza automatikusan a btnHozzaad\_Click eseményt a kódban.

Ezután a következő kódot írd be a Form1.cs-be (ahogy az előző válaszban is szerepelt):

csharp

MásolásSzerkesztés

private void btnHozzaad\_Click(object sender, EventArgs e)

{

// ... lásd korábbi válaszban a teljes kódot

}

## ⏱️ 6. Űrlap betöltésekor (Form Load esemény)

### Kattints az üres felületre a Form1-en → ****Properties → Events (villám ikon) → Load****

→ duplakatt, létrejön: Form1\_Load

csharp

MásolásSzerkesztés

private void Form1\_Load(object sender, EventArgs e)

{

cmbKategoria.Items.AddRange(new string[] { "Maintenance", "Repairs", "Insurance", "Utilities", "Other" });

dtpDatum.Format = DateTimePickerFormat.Custom;

dtpDatum.CustomFormat = "yyyy-MM-dd";

LoadCsv();

}

Ne felejtsd el ezt a részt sem hozzáadni a Form1() konstruktorba, ha nem a Load eseményből hívod:

csharp

MásolásSzerkesztés

public Form1()

{

InitializeComponent();

// cmbKategoria és dtpDatum inicializálása

}

## 🧪 Tesztelés

1. Futtasd az alkalmazást (**F5**)
2. Ellenőrizd:
   * Betölti-e a CSV tartalmát?
   * Jól jelennek-e meg az adatok?
   * Hozzáad-e új adatot a táblázathoz és a fájlhoz?

## 🖼️ Végső elrendezési javaslat

### [dataGridView1] a bal oldalon

### [Form mezők] jobb oldalon egy oszlopban

* Esztétikailag: rendezett távolságok, egyvonalban címkék + mezők
* Mezők szélessége: kb. 200 px

Ha szeretnéd, küldhetek képet is az elrendezésről, vagy elküldhetem a teljes .Designer.cs fájlt a grafikus elemekkel együtt.