**JU Mješovita srednja škola „Hasan Kikić“ Školska godina: 2022/2023**

**GRADAČAC**

**MATURSKI RAD**

**Učenik: Ajdin Sušić Mentor: Mirela Avdić**

Sadržaj

[UVOD 3](#_Toc133329969)

[Microsoft Visual Studio 3](#_Toc133329970)

[Uvod u C# 4](#_Toc133329971)

[.NET Platforma 4](#_Toc133329972)

[.NET Framework (okvir) 4](#_Toc133329973)

[TEMPLATE: WindowsFormsApplication) 5](#_Toc133329974)

[Alati za kreiranje aplikacije 6](#_Toc133329975)

[Kontrole PictureBox,Label I IconButton 7](#_Toc133329976)

[Aplikaciija za prodaju Video-igrica 8](#_Toc133329977)

[Kreiranje okvira aplikacije 8](#_Toc133329978)

[Prvi Korak 8](#_Toc133329979)

[Drugi Korak 8](#_Toc133329980)

[Treći Korak 8](#_Toc133329981)

[Dodavanje i manipulisanje kontrolama 8](#_Toc133329982)

[Objasnjenje koda iznad vam se nalazi u donjem tekstu! 10](#_Toc133329983)

[Kreiranje Child Formi 12](#_Toc133329984)

[Spajanje Child Form sa Button Menu-om 13](#_Toc133329985)

[Mogućnost razvlačenja Forme 17](#_Toc133329986)

[Zakljucak 18](#_Toc133329987)

[Izvori 18](#_Toc133329988)

# UVOD

U ovom maturskom radu opisano je kreiranje aplikacije za prodaju video igrica u programskom jeziku C# u okruženju Microsoft Visual Studio. Objasnićemo programski kod za kreiranje aplikacije kao i karakteristike programskog jezika C# i .NET platforme.

C# je jedan od vodećih programskih jezika današnjice. On je objektno orijentisan, što olakšava pisanje programa u skladu sa konceptom objektno orijentisanog programiranja (OOP). OOP je unio novosti u programski svet i rešio velike probleme koji su nastajali. OOP se bazira na upotrebi objekata koji su građeni od varijabli koje se nazivaju atributi, funkcija koje se nazivaju metodama, te identifikatora koji pomaže u razlikovanju istovrsnih objekata. OOP se u pravilu sastoji od tri glavna koncepta: učahurivanje, nasleđivanje i polimorfizam. Pomoću tih koncepata moguće je dizajnirati računarske sisteme koji su iskoristivi u više različitih aplikacija. Učahurivanje je postupak koji onemogućava direktan pristup vrijednostima atributa iz dijelova koda. Kod nasleđivanja se klasa ili objekt baziraju na nekoj već postojećoj klasi ili objektu pri čemu koriste postojeće klase ili atribute. Polimorfizam se može podijeliti u dva koncepta. Prvi koncept omogućava objektu da posjeduje više metoda istog imena koje zatim kroz argumente primaju različite podatke. Drugi koncept se odnosi na nasleđivanje te omogućava spremanje objekta tipa podklase u varijablu deklariranu za tip podatka njene nadklase.

Cilj ovog maturskog rada je kreiranje aplikacije za prodaju video igrica u programskom jeziku C#. Aplikacija je napravljena da bi kupcima omogućila jednostavan i brz pristup video igrama, prikazala informacije o igrama i olakšala proces kupovine. Aplikacija sadrži funkcionalnosti kao što su pretraga igrica po žanru, cijeni ili imenu, pregled detaljnih informacija o igrama, kreiranje korisničkog računa i mogućnost kupovine igrica.

U narednim sekcijama objasniti ćemo proces kreiranja aplikacije i kod koji je korišćen u Microsoft Visual Studiju.

# Microsoft Visual Studio

Microsoft Visual Studio je integrisano razvojno okruženje (IDE) koje je razvila Microsoft kompanija za programere i programerske timove koji žele da stvore visoko kvalitetne aplikacije za različite platforme, uključujući Windows, Android i iOS. Visual Studio nudi mnoge alate i funkcionalnosti koji programerima pomažu da stvore, testiraju i debaguju svoj kod na jednostavan i efikasan način.

Sa integriranim kod editorom, kompajlerom i debagerom, Visual Studio nudi programerima mogućnost da razviju aplikacije brže i efikasnije nego što bi to bilo moguće korišćenjem drugih alata. Visual Studio takođe podržava različite programerske jezike, uključujući C++, C#, Visual Basic, JavaScript, Python i mnoge druge, što znači da programeri mogu da koriste svoj omiljeni programski jezik za razvoj aplikacija.

Uz bogatu biblioteku gotovih koda, aplikacije koje su razvijene u Visual Studiju su skalabilne i fleksibilne, što omogućava programerima da brzo i lako prilagode svoje aplikacije različitim zahtevima. Uz sve ove karakteristike, Visual Studio ostaje jedan od najpopularnijih alata za razvoj softvera među programerima širom sveta.

# Uvod u C#

C# (izgovara se "C sharp") je moderni, objektno-orijentirani programski jezik razvijen od strane Microsofta. Prvi put se pojavio 2000. godine kao ključni jezik platforme Microsoft .NET Framework, a danas se koristi za razvoj različitih aplikacija, uključujući Windows desktop aplikacije, mobilne aplikacije, web aplikacije i igre.

C# se često uspoređuje s programskim jezikom Java, jer dijele slične koncepte poput objektno-orijentiranog programiranja, upravljanja memorijom i sintakse. Međutim, C# se često smatra modernijim i učinkovitijim jezikom, s brojnim naprednim značajkama poput delegata, generičkih tipova, anonimnih tipova i LINQ-a (Language Integrated Query).

S obzirom na svoju popularnost i snažne mogućnosti, C# se često smatra ključnim programskim jezikom za razvoj aplikacija na Microsoft platformi, te stoga ima široku primjenu u industriji softverskog inženjeringa.

# .NET Platforma

.NET platforma je popularna platforma za razvoj softvera koju je razvio Microsoft. Sastoji se od nekoliko alata, programa i biblioteka koji su namijenjeni za izradu različitih vrsta softvera, uključujući web aplikacije, desktop aplikacije, mobilne aplikacije, igre i još mnogo toga.

.NET platforma koristi C# kao svoj primarni programski jezik, ali podržava i druge jezike poput Visual Basic i F#. Jedna od glavnih prednosti .NET platforme je njezina interoperabilnost, koja omogućuje integraciju različitih programskih jezika i biblioteka.

.NET platforma također nudi mnoge druge značajke koje olakšavaju razvoj softvera, poput jednostavnog upravljanja memorijom, integracije s bazama podataka, brze izgradnje aplikacija i lakoće održavanja. Pored toga, .NET platforma pruža razne alate i okvire koji programerima olakšavaju izradu visoko kvalitetnog i pouzdanog softvera.

# .NET Framework (okvir)

.NET Framework je softverska platforma razvijena od strane Microsofta koja pruža okruženje za razvoj, izvođenje i upravljanje aplikacijama. Ova platforma sadrži biblioteke koda, jezgrenu izvršnu okolinu i razvojne alate koji olakšavaju razvoj aplikacija za Windows operativne sisteme.

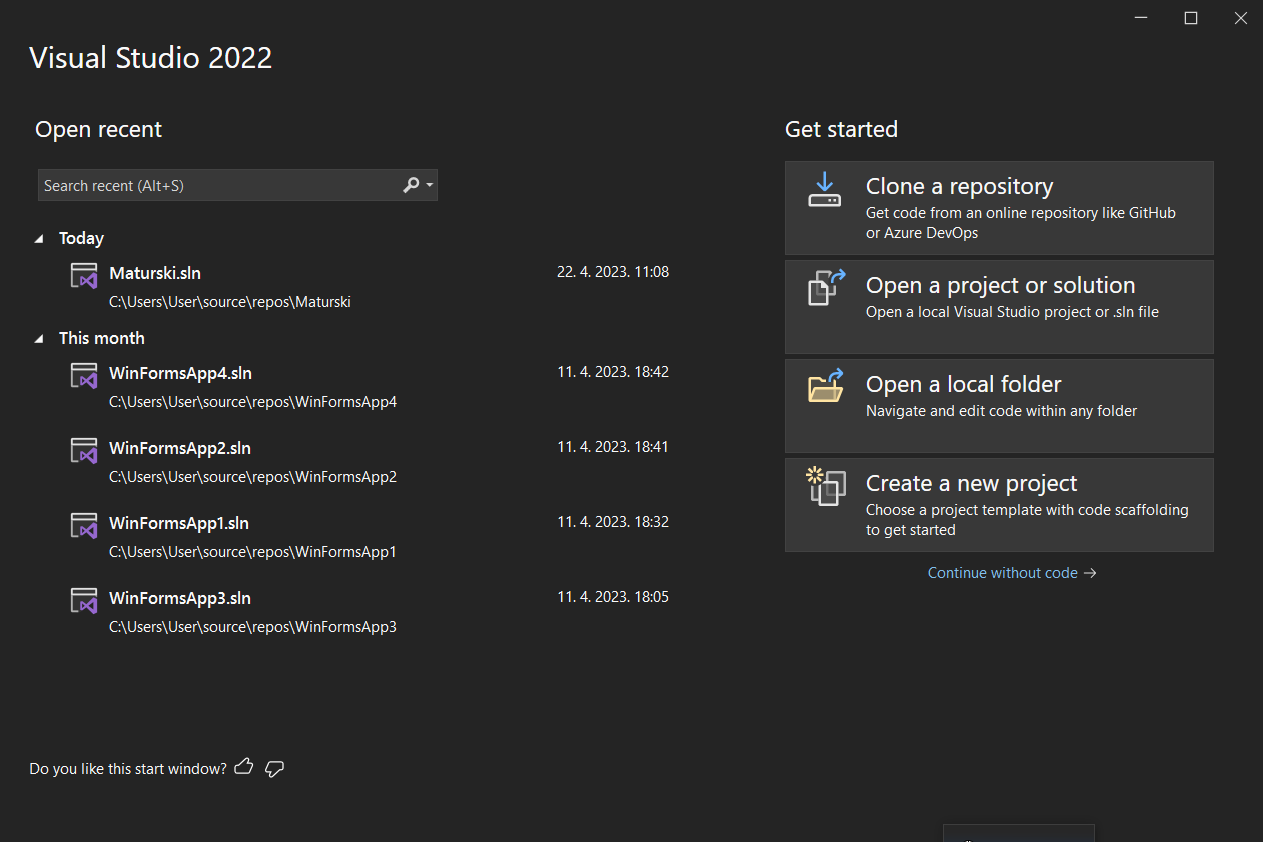
.NET Framework je osmišljen tako da pojednostavi razvoj aplikacija korištenjem višejezične i višesistematske podrške, što omogućava programerima da pišu kod jednom i izvršavaju ga na bilo kojem ciljnom uređaju. Platforma također uključuje mehanizme za upravljanje memorijom, sigurnost i stabilnost aplikacija.

.NET Framework je podržan od strane velikog broja razvojnih alata, uključujući Microsoft Visual Studio, koji pruža sve što je potrebno za razvoj aplikacija, od uređivanja koda do testiranja i objavljivanja aplikacija. Uz to, .NET Framework je omogućio razvoj aplikacija koje se mogu integrirati s drugim Microsoftovim softverskim proizvodima, kao što su Microsoft Office aplikacije, SharePoint i SQL Server.

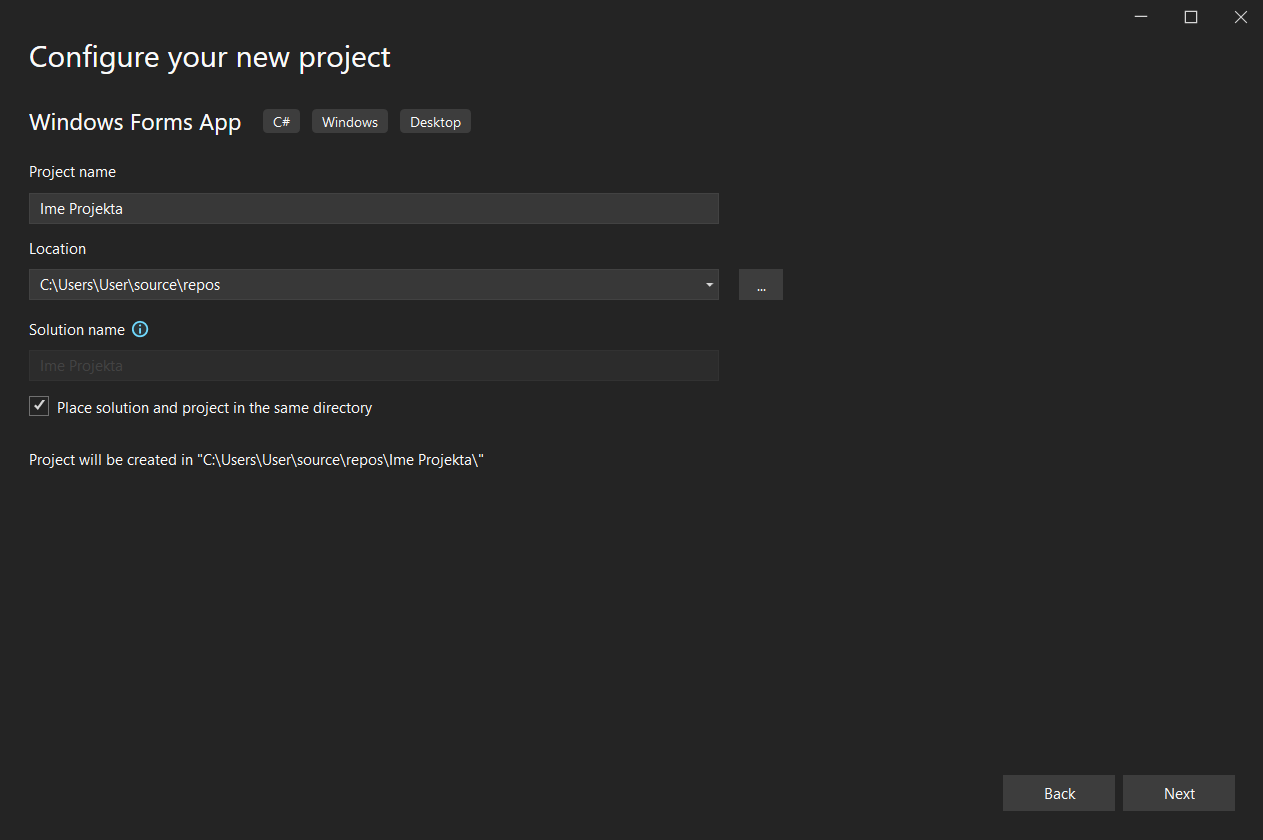
Ukratko, .NET Framework pruža programerima razvojno okruženje koje olakšava razvoj aplikacija za Windows operativne sisteme. Ova platforma nudi mnoštvo alata i funkcionalnosti koje pojednostavljuju proces razvoja, izvršavanja i upravljanja aplikacijama, što omogućava programerima da se fokusiraju na razvoj kvalitetnog softvera koji zadovoljava korisničke potrebe.

# TEMPLATE: WindowsFormsApplication)

Program u kojem ćemo raditi naziva se Microsoft Visual Studio 2022.Prvi korak nakon paljenja programa je opcija Create a new project, na početnoj strain programa.

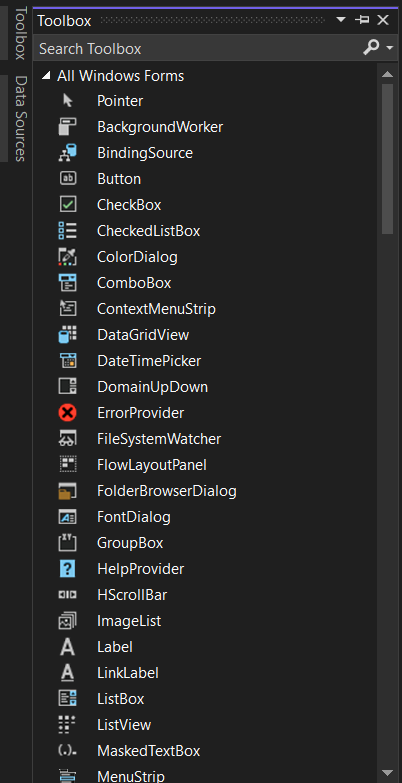


Nakon toga, bit će nam prikazan prozor gdje ćemo odabrati koju uslugu celimo i koja razvojna okruženja želimo da aplikacija koristi. Mi ćemo odabrati "Windows Forms App" pritiskanjem na tu opciju, a naravno potrebno je dati i ime našem projektu.



# Alati za kreiranje aplikacije

Da bismo uradili bilo šta u našem projektu, potrebno je da koristimo alate za kreiranje aplikacija. Sve alate možemo pronaći u lijevom meniju pod nazivom "Toolbox".



# Kontrole PictureBox,Label I IconButton

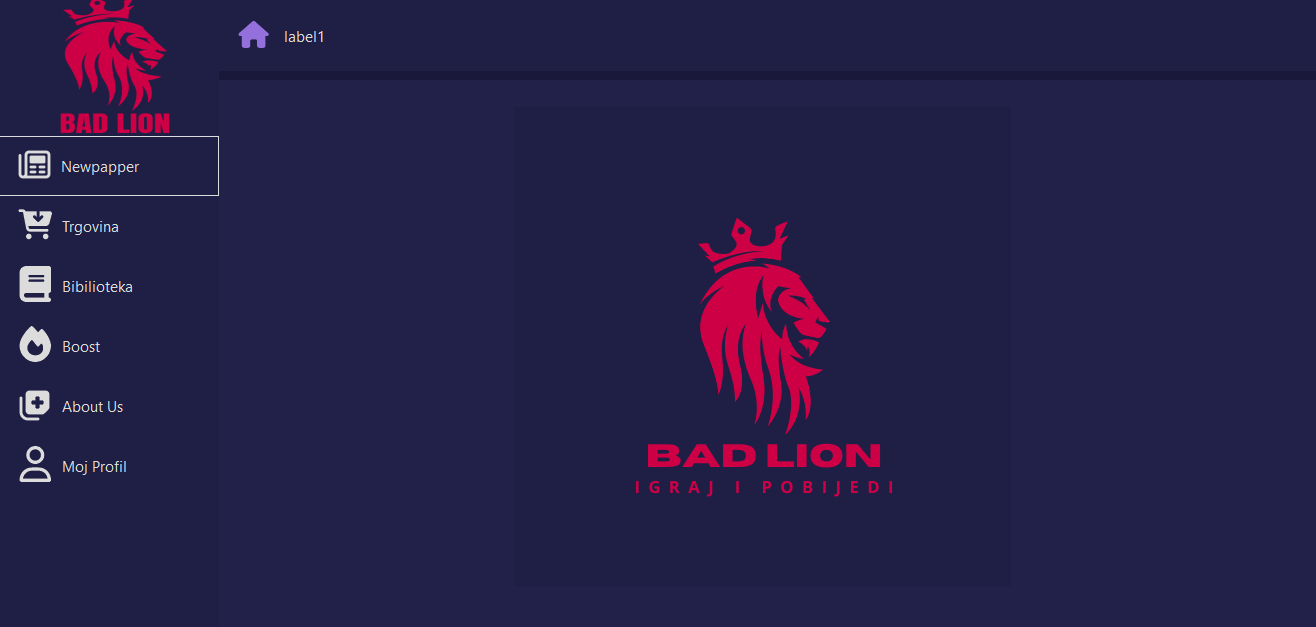
Kontrola Label: Ova kontrola omogućava prikaz teksta na formi, ali se obično vezuje za neku drugu kontrolu i na taj način opisuje njeno značenje. Labela pod imenom (name) "lblprikaz" je kreirana za prikaz otvorene Child forme, odnosno njenog imena (Name).

PictureBox: Ova kontrola nam omogućava da prikažemo sliku sa našeg računara u našoj Windows Form aplikaciji. Također, izvršenje ove radnje radi se u opciji "Choose image" u meniju pod nazivom "Properties".

IconButton: IconButton je modifikovani Button koji dobijamo uključenjem biblioteke pod nazivom FontAwesome, naravno prije toga je moramo instalirati u naš program, a ona nam je ponuđena u samom Visual Studio programu. Ova kontrola nam, ustvari, u našu Windows formu dodaje button u koji možemo ubaciti ikonu po našoj želji, promijeniti boju ikone, font i podesiti lokaciju naredbom "Dock".



# Aplikaciija za prodaju Video-igrica



Na slici iznad je prikazana početna strana (Home) aplikacije. Kao što možemo vidjeti, sastoji se od glavne forme (Main Form) i sporednih formi koje se aktiviraju klikom na jedno od dugmadi lijevog IconButton menija.

# Kreiranje okvira aplikacije

Okvir aplikacije smo kreirali dodavanjem elemenata iz Toolbox-a na sljedeći način.

## Prvi Korak

Prvo smo podesili dužinu i širinu naše forme u željenim mjerama. Nakon toga, dodali smo Panel elemente koje smo obojali, tako što smo podesili Fore Color u opciji Properties.

## Drugi Korak

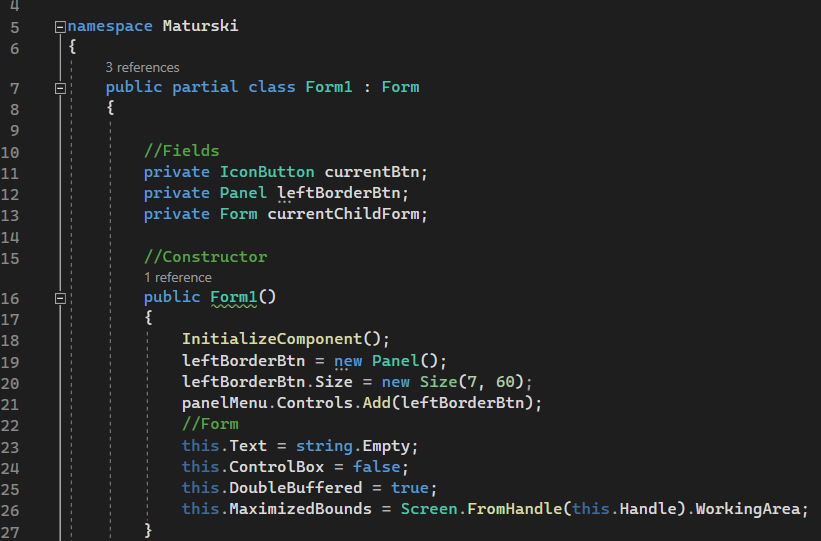
Nakon prvog koraka, prelazimo na dodavanje IconButton elemenata. U našem slučaju, dodali smo 6 IconButtona. Nakon toga, u opciji Properties podesili smo boju (ForeColor), ikonu (Icon), boju teksta (TextColor) i lokaciju teksta (Dock).

## Treći Korak

Dodat ćemo Labelu koja će klikom na neki od IconButtona promijeniti boju i tekst.

# Dodavanje i manipulisanje kontrolama

Pošto smo kontrole već dodali, sada ćemo prijeći na dio gdje klikom na IconButton promijenimo boju teksta i spustimo panel koji se nalazi sa lijeve strane i predstavlja selektivni panel. Kod za to se nalazi na sljedećim slikama.

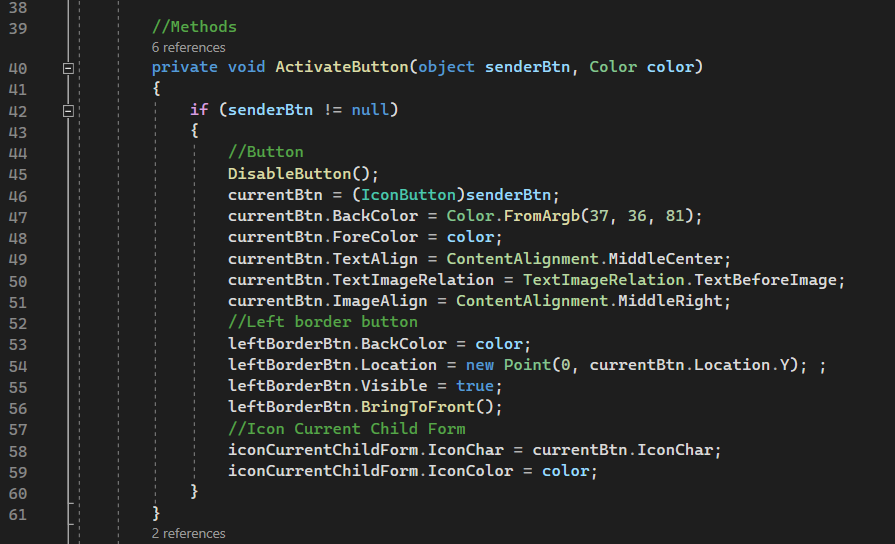


U dijelu Fields definišemo IconButton, LeftBorderButton (Selektivni panel) i trenutnu formu. Nakon toga, podesavamo radnje koje će naš selektivni panel obavljati na IconButton meniju. Ovaj dio se nalazi u dijelu konstruktora (Constructor).

Sto se tiče dijela //Form svaku liniju ćemo posebno objasniti!

1. **this.Text = string.Empty;** - Postavlja vrijednost **Text** svojstva trenutnog objekta na prazan string. Ovo će učiniti da se naslovna traka prozora (ukoliko je prozor u pitanju) prikaže bez teksta.
2. **this.ControlBox = false;** - Postavlja vrijednost **ControlBox** svojstva trenutnog objekta na **false**. Ovo će ukloniti gumb za kontrolu prozora (tipično gumb za zatvaranje, maksimiziranje, itd.) iz naslovne trake.
3. **this.DoubleBuffered = true;** - Postavlja vrijednost **DoubleBuffered** svojstva trenutnog objekta na **true**. Ovo će omogućiti dvosmjerno bojanje površine objekta, čime se smanjuje mogućnost vidljivih vizualnih artefakta tijekom crtanja.
4. **this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;** - Postavlja vrijednost **MaximizedBounds** svojstva trenutnog objekta na radnu površinu (working area) ekrana na kojem se prozor trenutno nalazi. Ovo će osigurati da se prozor maksimizira samo na područje koje je vidljivo, bez prekrivanja sustavnih elemenata poput trake zadataka ili izbornika.

Ovaj dio koda se odnosi na funkciju **ActivateButton** koja se poziva kada korisnik klikne na neki **IconButton** u bočnoj traci aplikacije. Funkcija mijenja vizualni izgled tog gumba i označava ga kao trenutno odabranog.



## Objasnjenje koda iznad vam se nalazi u donjem tekstu!

**if (senderBtn != null)** - provjera da li je proslijeđeni gumb zaista postavljen, kako bi se izbjegle potencijalne greške u programu.

**DisableButton()** - poziva funkciju **DisableButton** koja mijenja izgled svih gumba u bočnoj traci na default izgled, osim trenutno odabranog.

**currentBtn = (IconButton)senderBtn;** - trenutno kliknuti gumb se sprema u varijablu **currentBtn** kako bi se njegova svojstva mogla promijeniti.

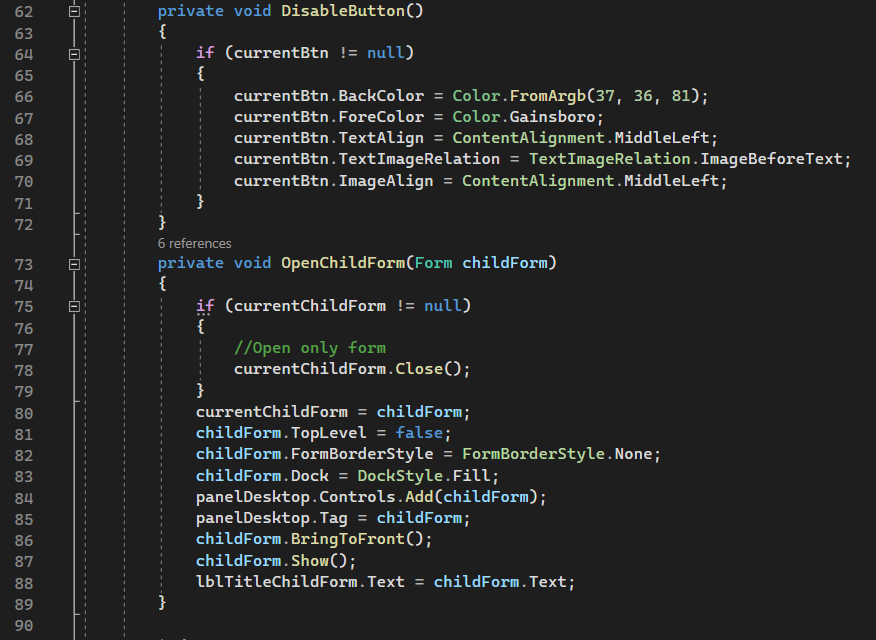
**currentBtn.BackColor**, **currentBtn.ForeColor**, **currentBtn.TextAlign**, **currentBtn.TextImageRelation** i **currentBtn.ImageAlign** - mijenjaju se svojstva trenutno kliknutog gumba kako bi se postigao željeni vizualni efekt.

**leftBorderBtn.BackColor**, **leftBorderBtn.Location** i **leftBorderBtn.Visible** - mijenjaju se svojstva "lijeve granice" bočne trake kako bi se istaknuo trenutno odabrani gumb.

**leftBorderBtn.BringToFront()** - pomjera se granica lijevo od trenutno odabranog gumba ispred ostalih kontrola u bočnoj traci.

**iconCurrentChildForm.IconChar** i **iconCurrentChildForm.IconColor** - mijenjaju se ikona i boja ikone prikazane na vrhu glavnog prozora, kako bi se prikazala ikona trenutno odabranog gumba.

Ovim kodom se postiže vizualno isticanje trenutno odabranog gumba u bočnoj traci, što korisniku olakšava navigaciju kroz aplikaciju.



Ovaj dio koda se sastoji od dvije metode, **DisableButton()** i **OpenChildForm(Form childForm)**.

**Metoda** **DisableButton()** mijenja vizualni izgled trenutno odabranog gumba u bočnoj traci. Ako je trenutno odabrani gumb postavljen (što se provjerava u uvjetu **if (currentBtn != null)**), tada se mijenjaju sljedeća svojstva tog gumba:

**currentBtn.BackColor** - mijenja se pozadinska boja gumba u tamno plavu boju.

**currentBtn.ForeColor** - mijenja se boja teksta gumba u svjetlo sivu boju.

**currentBtn.TextAlign** - postavlja se poravnavanje teksta na lijevu stranu gumba.

**currentBtn.TextImageRelation** - postavlja se raspored teksta i ikone na gumbu da bude tekst prije ikone.

**currentBtn.ImageAlign** - postavlja se poravnavanje ikone na lijevu stranu gumba.

Metoda **OpenChildForm(Form childForm)** otvara novi prozor (child form) u glavnom prozoru aplikacije. Metoda prvo zatvara bilo koji trenutno otvoreni child form (ako postoji) pozivom funkcije **Close()** na tom formu. Zatim se postavlja novi child form, koji se dobiva kao parametar metode **childForm**.

Nakon toga, postavlja se nekoliko svojstava na tom child formu:

**childForm.TopLevel = false** - označava da se radi o child formi i da se ne prikazuje kao samostalan prozor.

**childForm.FormBorderStyle = FormBorderStyle.None** - uklanja okvir oko child forme.

**childForm.Dock = DockStyle.Fill** - popunjava se cijeli dostupan prostor na panelu **panelDesktop** (koji se koristi za prikazivanje child formi).

**panelDesktop.Controls.Add(childForm)** - dodaje se child forma kao kontrola na panel **panelDesktop**.

**panelDesktop.Tag = childForm** - postavlja se oznaka child forme na panel **panelDesktop**.

**childForm.BringToFront()** - postavlja se child forma na vrh ostalih kontrola na panelu **panelDesktop**.

**childForm.Show()** - prikazuje se child forma.

**lblTitleChildForm.Text = childForm.Text** - postavlja se tekst naslova child forme u **lblTitleChildForm**, koji prikazuje naslov trenutno otvorene child forme.

Ovaj kod omogućuje otvaranje i zatvaranje child formi u glavnom prozoru aplikacije, što omogućuje korisniku navigaciju kroz različite dijelove aplikacije.

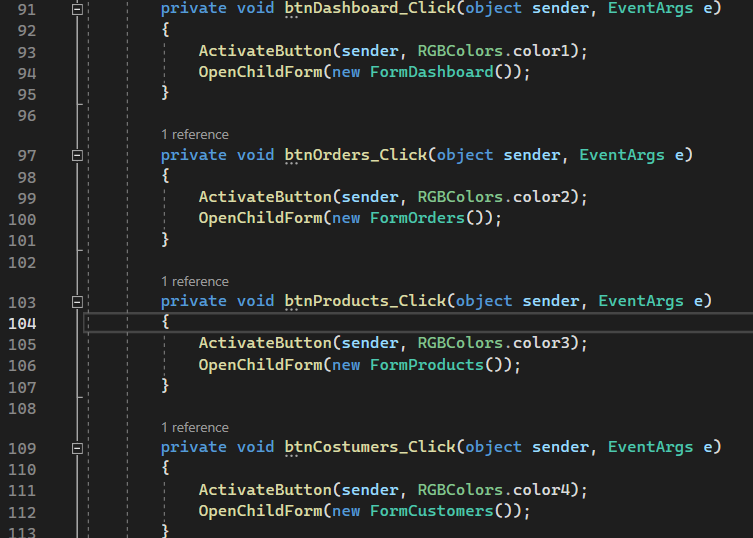
# Kreiranje Child Formi

U Visual Studio-u možete kreirati Child forme na sličan način kao što kreirate obične forme. Evo koraka:

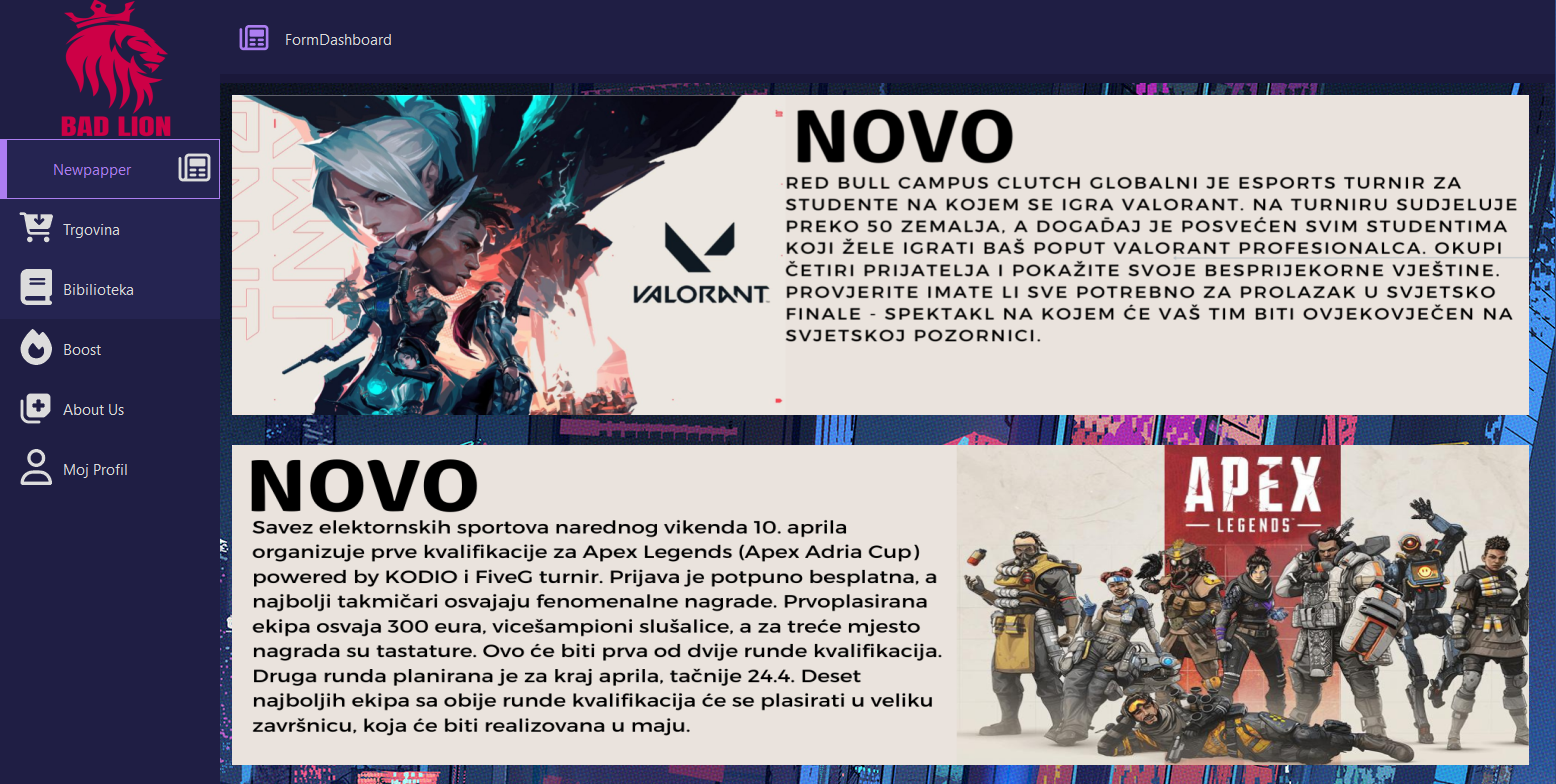
Kliknite na Solution Explorer-u desnim klikom miša i odaberite "Add" -> "New Item".

1. Odaberite "Windows Forms" pod "Installed" -> "Visual C#" na lijevoj strani.
2. Odaberite "Windows Form" na desnoj strani i nazovite formu prema svojim željama.
3. Kliknite "Add" i nova forma će se dodati vašem projektu.
4. U glavnoj formi, dodajte kontejner u koji ćete staviti child forme. To može biti Panel, GroupBox, TabControl ili neki drugi kontejner.
5. Kada ste dodali kontejner, možete kreirati instance Child formi koje želite otvoriti. To možete učiniti korištenjem "new" ključne riječi i naziva forme.
6. Konfigurirajte Child formu prema željama, postavite je na "TopLevel = false", "FormBorderStyle = None" i "Dock = DockStyle.Fill". Ovo će omogućiti da se forma prilagodi veličini kontejnera.
7. Dodajte Child formu u kontejner koristeći "panel.Controls.Add(childForm)" metodu.
8. Nakon dodavanja, prikažite formu koristeći "childForm.Show()" metodu.

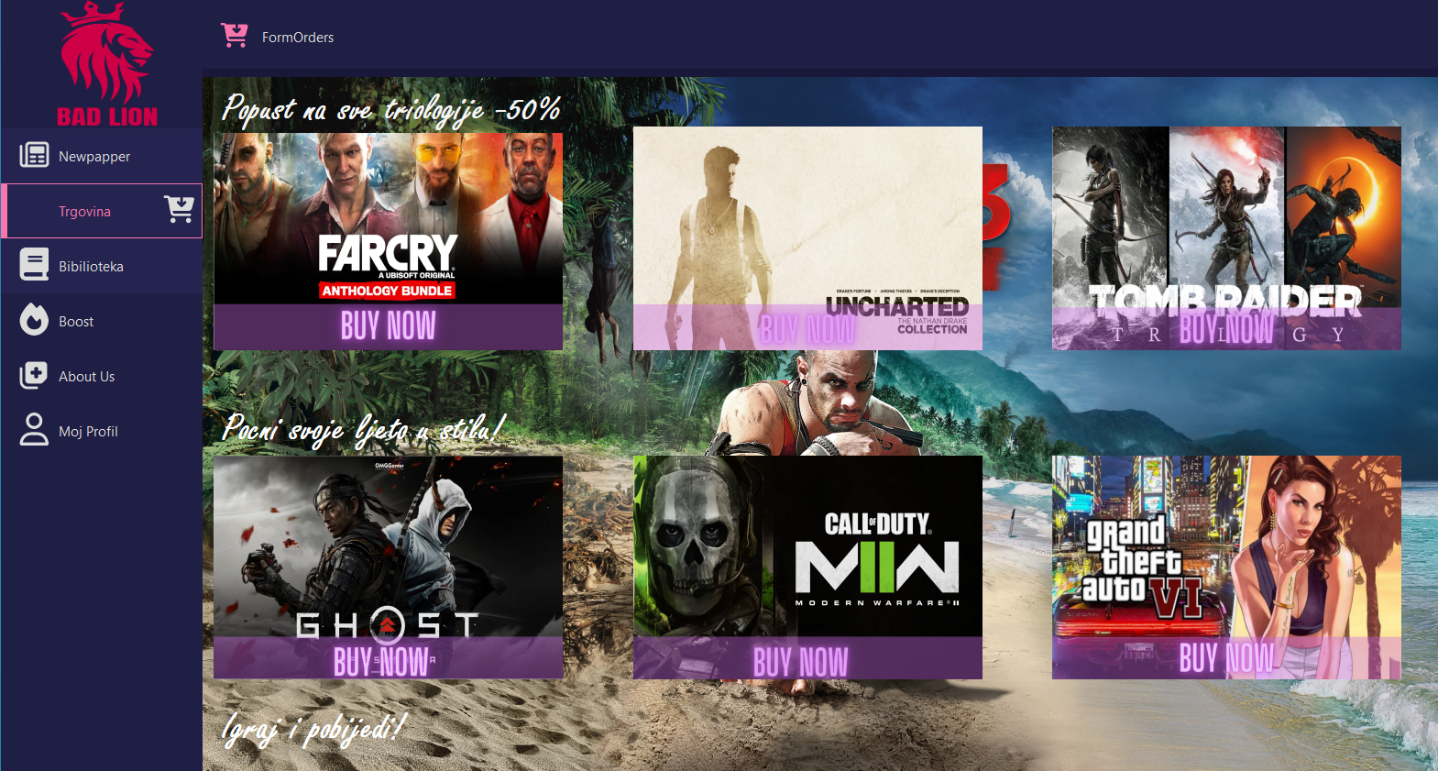
# Spajanje Child Form sa Button Menu-om

Child Forme ćemo spojiti sa IconButton menijem sledećim kodom.  


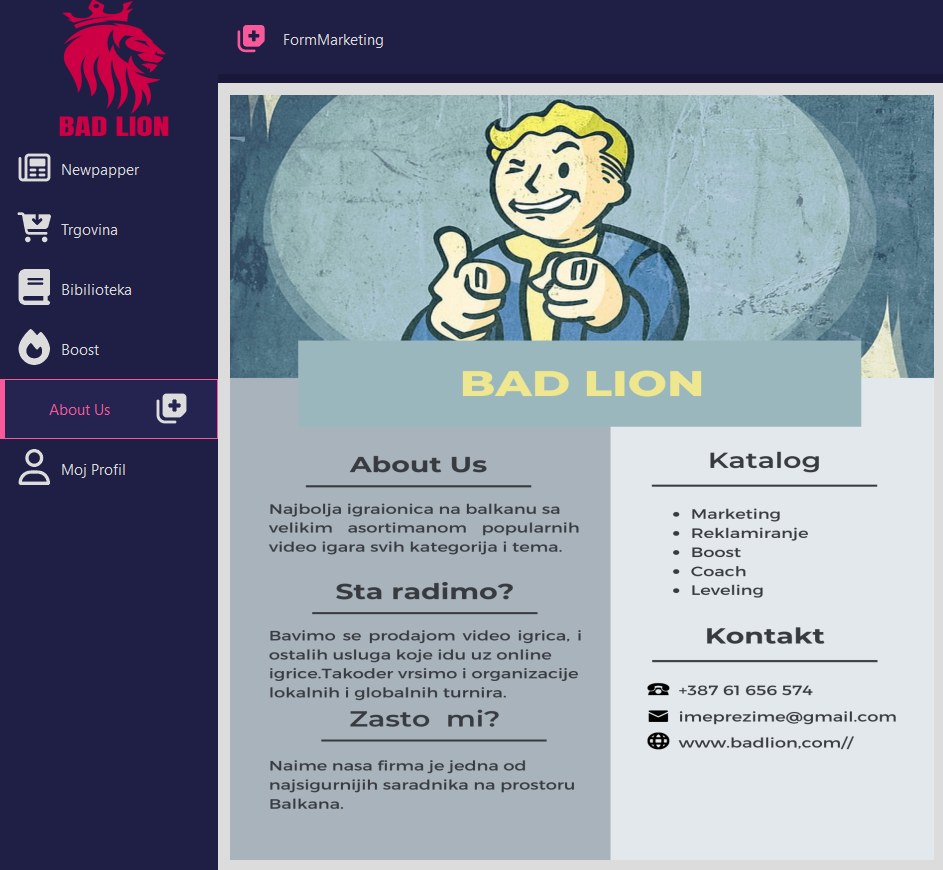
Kao sto mozemo vidjeti u kodu, aktivacijom OconButtona na slici otvorit ce nam se Child Forma a to ce izgledati ovako.Sljdecce slike prikazat ce sve Child forme pojedinacno.

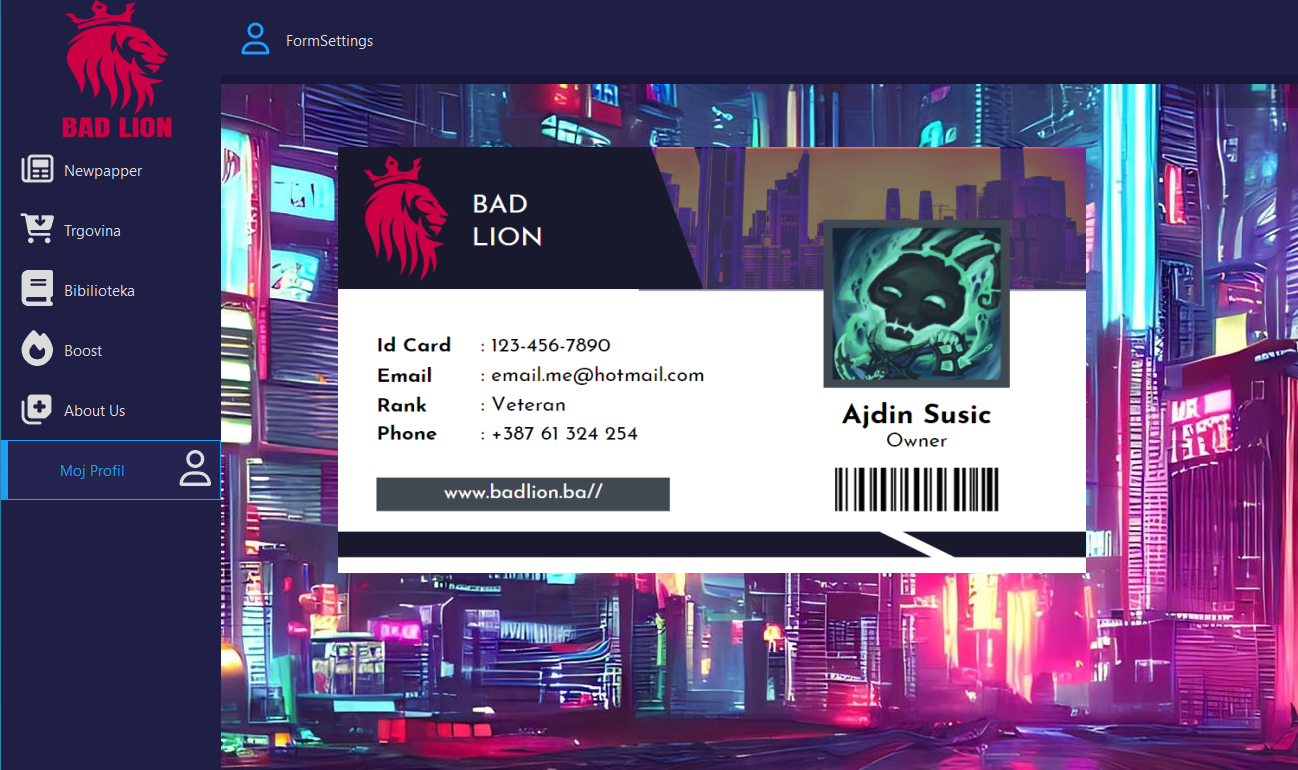


Klikom na IconButton "Newspaper" otvorila nam se Child forma. Također, IconButton sa lijeve strane se promijenio u ljubičastu boju. Na ovoj Child formi se nalaze novosti vezane za igre.



Child Forma pod nazivom “Trgovina” predstavlja dio aplikacije koji služi za kupovinu video igrica. Kreirali smo je dodavanjem PictureBox-ova i labela koji odvajaju grupe određenih igrica. Za pozadinsku sliku odabrali smo naslovnu sliku iz igre “Far Cry 3”, odnosno sliku smo odabrali prema kontrastu boja sa naše Child Forme. Također, sliku smo ubacili koristeći opciju BackgroundImage u dijelu Properties.

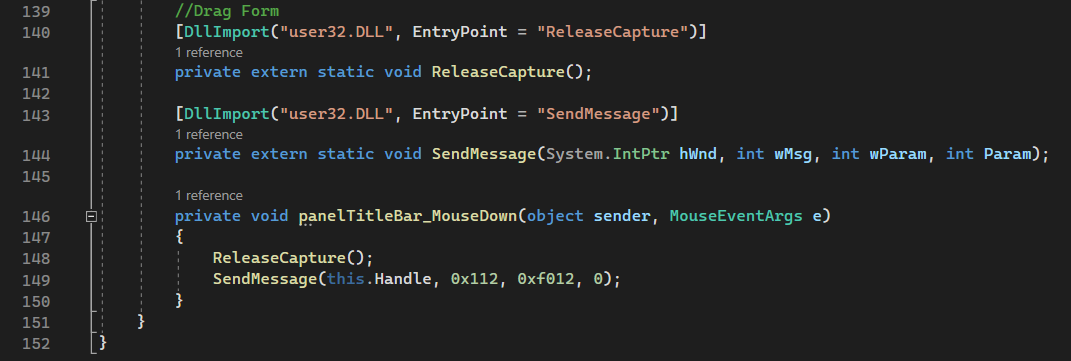
"Child Forma biblioteka predstavlja PDF formate knjiga u kojima je ispričana priča od neke igrice. Ubacivši ovaj dio, obratili smo pažnju na korisnike koji više vole doživjeti neku priču kroz čitanje nego kroz igranje. Naravno, ovo smo napravili ubacivanjem PictureBox-a koji je sadržavao sliku. Za BackColor smo odabrali bijelu boju jer nam najbolje odgovara uz elemente Child Forme.  


Dio pod nazivom "About Us" predstavlja odsječak u kojem je ukratko predstavljena aplikacija, odnosno sve što nam ta aplikacija nudi, kontakti i preporuke.  


Moj profil, odnosno naša najbitnija Child Forma, predstavlja dio u kojem smo se registrovali na početku korištenja aplikacije. Nakon registracije, aplikacija će napraviti vašu ID karticu. Na kartici se nalaze osnovni podaci i vaš rang na aplikaciji. Od rangova moguće je izdvojiti: member, donator, owner, admin i veteran. Rangovi se stječu po zasluži ili doprinosu samoj aplikaciji.

# Mogućnost razvlačenja Forme

Naravno, ništa od ovoga ne bi moglo biti realizovano da naša aplikacija nema opciju Form resize. Za ovaj dio koristit ćemo sljedeći kod:

  
Ovaj kod sadrži definicije dvije metode koje su uvezane iz Windows API-ja korištenjem DllImport atributa. Te metode su "ReleaseCapture" i "SendMessage".

"ReleaseCapture" funkcija oslobađa miša iz kontrole na koju je bio vezan, što omogućuje da se korisnik može slobodno kretati mišem po zaslonu. To se obično koristi za povlačenje prozora bez korištenja okvira prozora.

"SendMessage" funkcija omogućuje slanje poruka sustavu. Ovdje se koristi za slanje poruke da se aktivira akcija "WM\_SYSCOMMAND" za prebacivanje prozora, koja se definira kao 0x112. Parametri 0xf012 i 0 su wParam i lParam poruka i označavaju da se prozor treba premjestiti.

Te dvije funkcije su pozvane u događaju "MouseDown" panela "panelTitleBar". Kada se korisnik klikne i drži miša na tom panelu, poziva se "ReleaseCapture" funkcija koja oslobađa miša iz panela. Nakon toga, poziva se "SendMessage" funkcija koja šalje poruku koja prebacuje prozor na novu lokaciju. Kao što se može vidjeti, ova funkcija omogućuje korisniku da povlači prozor bez korištenja okvira prozora, što je česta praksa u modernim aplikacijama.

# Zakljucak

Forma se sastoji od bočnog panela s gumbima za navigaciju i glavnog panela za prikazivanje dječjih formi koje se prikazuju nakon odabira odgovarajućeg gumba u bočnom panelu.

Kod sadrži nekoliko metoda koje se koriste za aktiviranje gumba u bočnom panelu, otvaranje dječjih formi u glavnom panelu, resetiranje bočnog panela i postavljanje naslova za dječje forme.

Osim toga, kod sadrži i funkcionalnosti za pomicanje forme putem miša. Naime, koristi se unaprijed definirana funkcija "ReleaseCapture()" koja omogućava otpuštanje zahvaćanja prozora mišem i "SendMessage()" koja šalje poruku sustavu da izvrši određenu akciju. Ta funkcionalnost omogućava korisniku da pomakne aplikaciju putem miša povlačenjem naslovnog panela.

# Izvori

**https://www.youtube.com/**

[**https://www.wikipedia.org/**](https://www.wikipedia.org/)

[**https://stackoverflow.com/**](https://stackoverflow.com/)

[**https://openai.com/blog/chatgpt**](https://openai.com/blog/chatgpt)

**https://www.freecodecamp.org/**