



Univerzitet u Novom Sadu

Tehnički fakultet „Mihajlo Pupin“

Zrenjanin



SEMINARSKI RAD

Predmet: Programski prevodioci

Tema: Desktop aplikacija za evidenciju prodaje torti

Predmetni nastavnik:

Prof. dr Ljubica Kazi

Student:

Branislava Radu SI 8/20

Smer: Informacione

tehnologije -

softversko inženjerstvo

Zrenjanin, 2023.

Sadržaj

1. Uvod.....	3
2. Korisničko uputstvo	4
3. Opis implementacije	13
4. Greške programskog koda	34
5. Formalna gramatika grešaka	44
6. Zaključak.....	50
7. Literatura.....	50

1.Uvod

U ovom seminarskom radu predstavljena je desktop aplikacija za evidenciju prodaje torti. Aplikacija omogućava korisnicima da unose, čuvaju, menjaju i brišu podatke o tortama, kao i da pregledaju detaljne informacije o kupovinama, filtriraju proizvode po šifri, eksportuju podatke i prikazu najveći prihod.

Kroz naredna poglavlja detaljno ću predstaviti svrhu i funkcionalnosti ove aplikacije.

Osim toga, kroz implementaciju prikazaću ključne delove koda i tehnologije koje su omogućile kreiranje ove aplikacije.

Dalje u radu prikazaću i različite vrste greške programskog koda koje su se mogle javiti tokom razvoja ove aplikacije. Ovo će obuhvatiti leksičke, sintaksne, semantičke, runtime i logičke greške, prikazivajući kako aplikacija reaguje na njih, kao i kako ih ispraviti.

Na kraju rada, analiziraću formalnu gramatiku koda za situacije gde su namerne greške ubačene u kod.

Ovaj seminarski rad ima za cilj da uputi u funkcionalnosti ove desktop aplikacije i pomogne da se bolje razume proces njenog razvoja i moguće izazove sa kojima se susreće.

2. Korisničko uputstvo

Funkcionalnosti aplikacije su:

1. Unos i čuvanje podataka: Korisnici mogu unositi informacije o torti, uključujući šifru, naziv, ukus, oblik torte, cenu, količinu i datum prodaje.
2. Izmena podataka: Korisnici mogu ažurirati podatke o tortama u slučaju promena cena, količina ili drugih atributa proizvoda.
3. Brisanje podataka: Korisnici mogu brisati zapis iz tabeli u slučaju otkazivanje porudžbine
4. Pregled informacija o kupovinama: Aplikacija omogućava korisnicima da pregledaju informacije o svakoj kupovini.
5. Filtriranje proizvoda: Korisnici mogu filtrirati proizvode prema šifri.
6. Eksport podataka: Aplikacija omogućava korisnicima izvoz podataka.
7. Prikaz najveće zarade: Korisnici mogu videti ostavaren najveći prihod.

Dalje u radu ću detaljno opisati svaku od navedenih funkcionalnosti aplikacije.

Slika ispod prikazuje glavni prozor korisničkog interfejsa. (Slika 1.)

The screenshot displays a web application interface for managing cakes. It is divided into two main sections: 'UNOS TORTE' (Add Cake) on the left and 'PRIKAZ PRODATIH TORTI' (Display Sold Cakes) on the right. The 'UNOS TORTE' section features a background image of a cake and includes input fields for 'Šifra' (ID), 'Naziv' (Name), 'Oblik torte' (Cake Shape), 'Ukus' (Taste), 'Količina' (Quantity), 'Cena' (Price), and 'Datum prodaje' (Sale Date). It also has a 'BRISANJE' (Delete) button and a bottom row of buttons: 'UNOS', 'SNIMI', 'IZMENA', and 'ODUSTANI'. The 'PRIKAZ PRODATIH TORTI' section has a large empty box for displaying data, a 'FILTRIRAJ' (Filter) button, a 'NAJVEĆA ZARADA' (Highest Profit) button, and 'SVE' (All) and 'EXPORT XML' buttons at the bottom.

Slika 1. Korisnički interfejs

Korisnički interfejs je podeljen na dva dela.

U levom delu korisničkog interfejsa korisnici mogu unositi informacije o torti koje žele da dodaju u tabelu tj. polja za unos podataka kao što su šifra torte, naziv, ukus, količina, cena, oblik torte i datum prodaje.

Pored ovih polja, korisnicima su na raspolaganju funkcionalnosti kao što su unos novih podataka, izmena i brisanje postojećih, čuvanje podataka i odustajanje od trenutne akcije.



UNOS TORTE

Šifra:

Informacije o torti:

Naziv: Oblik torte: Ukus:

Informacije o kupovini:

Količina: Cena: Datum prodaje: ▾

Unesite šifru torte koju želite da obrišete: **BRISANJE**

UNOS **SNIMI** **IZMENA** **ODUSTANI**

Slika 2. Levi deo korisničkog interfejsa

U desnom delu korisničkog interfejsa nalazi se tabela koja prikazuje informacije o svim unetim prodanim tortama.

Osim same tabele, korisnicima su na raspolaganju funkcionalnosti za upravljenje ovim podacima kao što su: dugme za filtriranje podataka po šifri, dugme za prikaz najveće zarade, dugme za prikaz svih podataka u tabeli i dugme za ekport podataka.

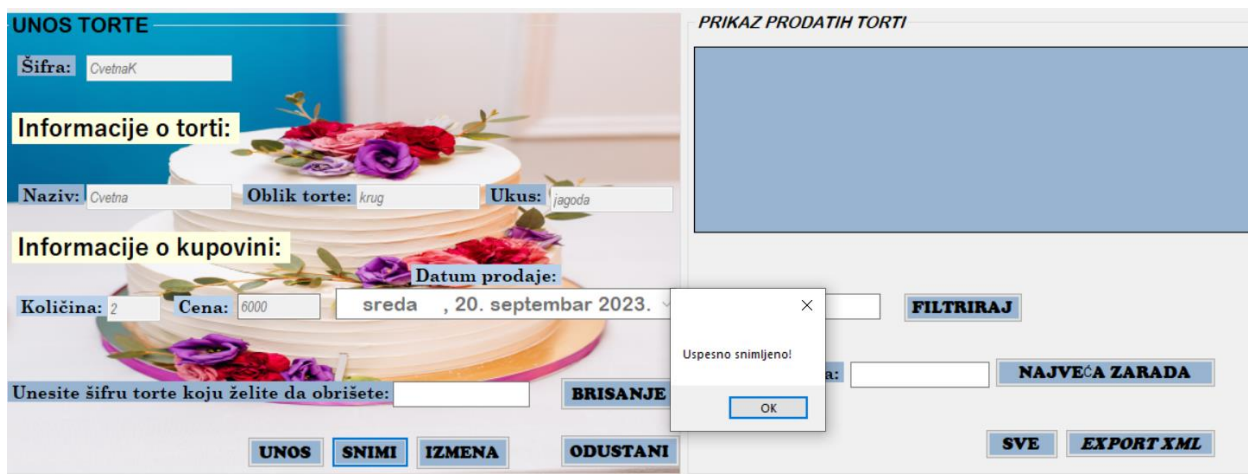
Slika 3. Desni deo korisničkog interfejsa

1. Unos i čuvanje podataka

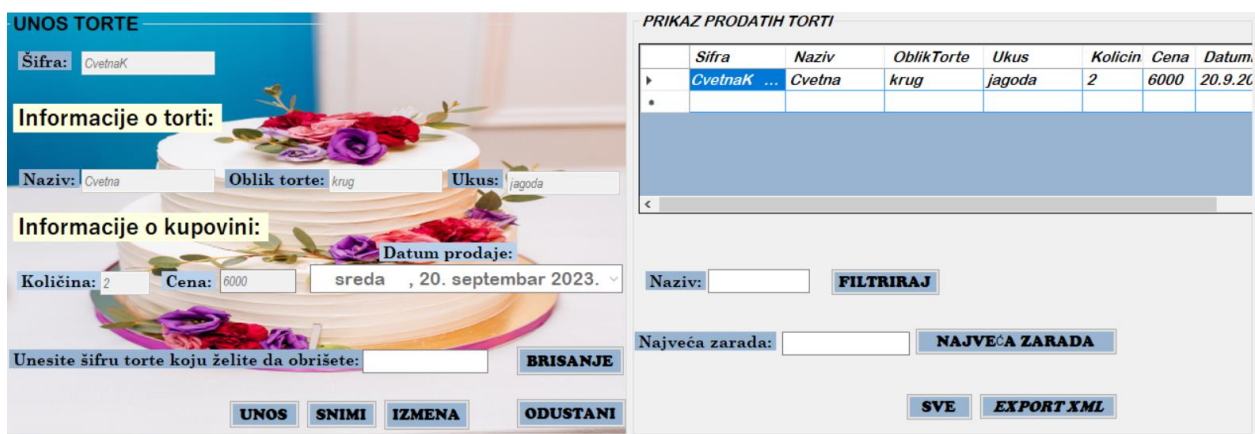
Prilikom unosa podataka u tabelu korisnik treba da popuni polja za unos i klikne na dugme “SNIMI” kako bi sačuvao te podatke.

Slika 4. Unos i čuvanje podataka

Prilikom uspešnog snimanja iskočiće poruka “Uspesno snimljeno!”.



Slika 5. Poruka o uspešnom čuvanju podataka



Slika 6. Izgled ekrana nakon unosa podataka

U slučaju da želimo da poništimo unos iz tekstualnih polja možemo učiniti klikom na dugme “ODUSTANI” i polja će biti onemogućena, ako bismo želeli da omogućimo potrebno je kliknuti dugme “UNOS”.

2. Izmena podataka

Da bismo korisnicima omogućili menjaje podataka, prvo moramo selektovati red u tabeli klikom na taj red i podaci iz tog reda će se automatski prikazati u tekstualnim poljima.

UNOS TORTE

Šifra:

Informacije o torti:

Naziv: Oblik torte: Ukus:

Informacije o kupovini:

Količina: Cena: Datum prodaje:

Unesite šifru torte koju želite da obrišete:

BRISANJE

UNOS SNIMI IZMENA ODUSTANI

PRIKAZ PRODATIH TORTI

Sifra	Naziv	OblikTorte	Ukus	Kolicin	Cena	Datum
CvetnaK ...	Cvetna	krug	jagoda	2	6000	20.9.20

Naziv: **FILTRIRAJ**

Najveća zarada: **NAJVEĆA ZARADA**

SVE **EXPORT XML**

Slika 7. Selektovan red u tabeli

Kada su podaci prikazani u tekstualnim poljima, korisnici imaju mogućnost da unesu izmene. Nakon što su uneli sve željene izmene (u ovom slučaju menja se oblik torte umesto “krug” stavlja se “kocka”) korisnici mogu sačuvati promene klikom na dugme "IZMENA". Klikom na ovo dugme ako je sve ispravno trebalo bi da iskoči poruka “Uspesno izmenjeno!” i program će automatski ažurirati podatke u tabeli sa novim vrednostima koje su korisnici uneli.

UNOS TORTE

Šifra:

Informacije o torti:

Naziv: Oblik torte: Ukus:

Informacije o kupovini:

Količina: Cena: Datum prodaje:

Unesite šifru torte koju želite da obrišete:

BRISANJE

UNOS SNIMI IZMENA ODUSTANI

PRIKAZ PRODATIH TORTI

Sifra	Naziv	OblikTorte	Ukus	Kolicin	Cena	Datum
CvetnaK ...	Cvetna	krug	jagoda	2	6000	20.9.20

Naziv: **FILTRIRAJ**

Najveća zarada: **NAJVEĆA ZARADA**

SVE **EXPORT XML**

Uspesno izmenjeno!

OK

Slika 8. Poruka o uspešnoj izmeni

UNOS TORTE

Šifra: CvetnaK

Informacije o torti:

Naziv: Cvetna
Oblik torte: kocka
Ukus: jagoda

Informacije o kupovini:

Količina: 2
Cena: 6000
Datum prodaje: sreda , 20. septembar 2023.

Unesite šifru torte koju želite da obrisete:

BRISANJE

UNOS

SNIMI

IZMENA

ODUSTANI

PRIKAZ PRODATIH TORTI

Sifra	Naziv	OblikTorte	Ukus	Kolicin	Cena	Datum
CvetnaK ...	Cvetna	kocka	jagoda	2	6000	20.9.2023.

Naziv:

FILTRIRAJ

Najveća zarada:

NAJVEĆA ZARADA

SVE

EXPORT XML

Slika 9. Izgled ekrana nakon izmene

3. Brisanje podataka

Prvi korak u procesu brisanja torte je unos šifre torte. U polje predviđeno za unos, korisnici trebaju upisati šifru te torte koju žele da obrišu.

Nakon što je šifra torte uneta korisnici trebaju kliknuti na dugme "BRISANJE". Klikom na ovo dugme, program će automatski pronaći tortu sa unetom šifrom i izbrisati je iz evidencije.

Nakon klika na dugme "BRISANJE" iskočiće poruka "Da li ste sigurni da želite da obrisete tortu?"

UNOS TORTE

Šifra: MikiMausP

Informacije o torti:

Naziv: Miki Maus
Oblik torte: pravougaonik
Ukus: banana

Informacije o kupovini:

Količina: 1
Cena: 4000
Datum prodaje: petak , 08. septembar

Unesite šifru torte koju želite da obrisete: MikiMausP

BRISANJE

UNOS

SNIMI

IZMENA

ODUSTANI

PRIKAZ PRODATIH TORTI

Sifra	Naziv	OblikTorte	Ukus	Kolicin	Cena	Datum
CvetnaK ...	Cvetna	kocka	jagoda	2	6000	20.9.2023.
MikiMaus...	Miki Maus	pravouga...	banana	1	4000	8.9.2023.

Naziv:

FILTRIRAJ

Najveća zarada:

NAJVEĆA ZARADA

SVE

EXPORT XML

Potvrda brisanja

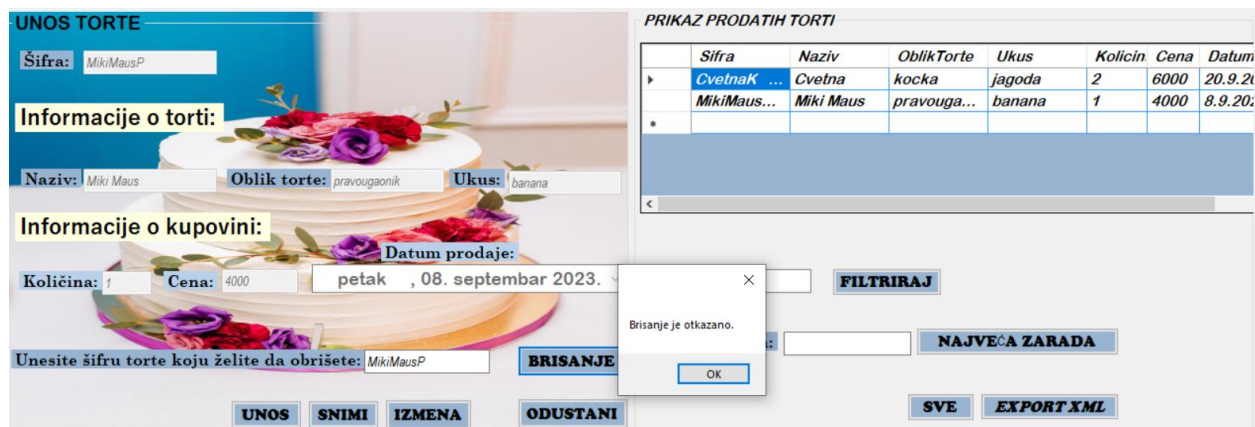
Da li ste sigurni da želite da obrisete ovu tortu?

Yes

No

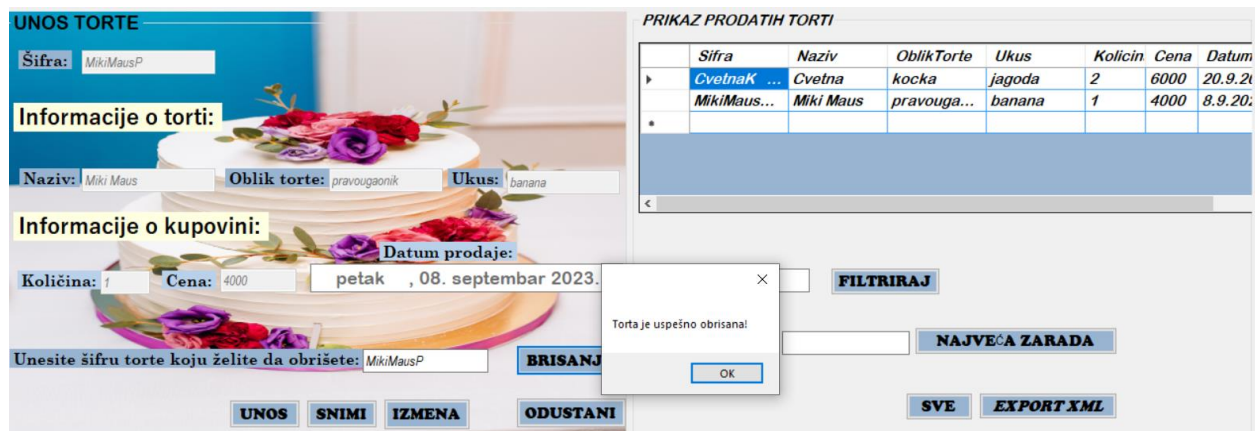
Slika 10. Poruka za potvrdu brisanja

Ako se klikne na "No" iskočiće poruka "Brisanje je otkazano."

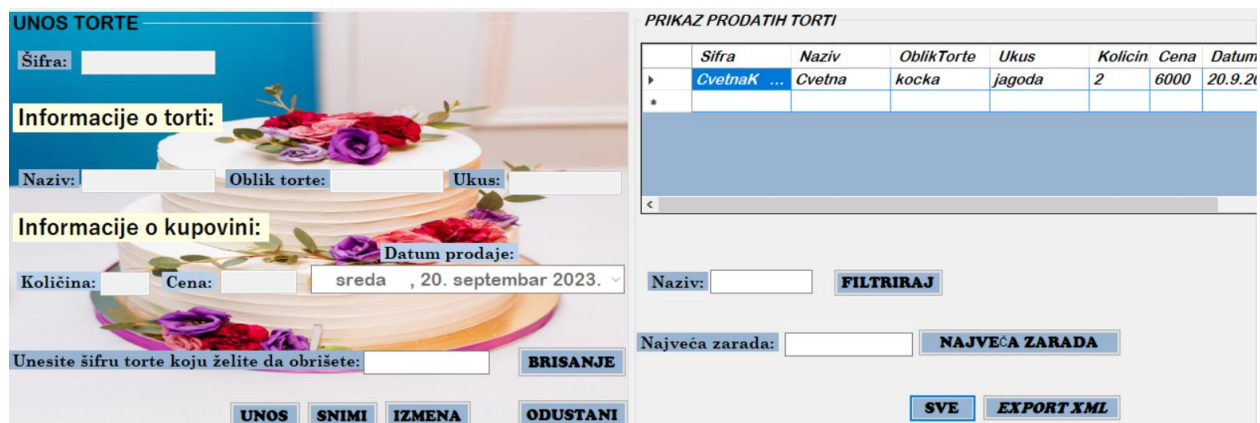


Slika 11. Poruka o otkazanom brisanju

Ako se klikne na “Yes” iskočiće poruka “Torta je uspešno obrisana!”



Slika 12. Poruka o uspešnom brisanju torte



Slika 13. Izgled ekrana nakog brisanja torte sa određenom šifrom

4. Pregled informacija o kupovinama

Klikom na dugme "SVE" korisnici mogu dobiti kompletnu pregled informacija o svim tortama koje su dosad prodane.

PRIKAZ PRODATIH TORTI

	Sifra	Naziv	OblikTorte	Ukus	Kolicina	Cena	D
▶	CvetnaK ...	Cvetna	pravoug...	jagoda	3	9000	12
	DobosP ...	Dobos	pravoug...	malina	2	6000	12
	MikiMaus...	Miki Maus	pravoug...	banana	2	6000	12
*							

<

Naziv: **FILTRIRAJ**

Najveća zarada: **NAJVEĆA ZARADA**

SVE **EXPORT XML**

Slika 14. Prikaz torti koje su prodane

5. Filtriranje proizvoda

Ukoliko korisnik želi da filtrira podatke o tortama prema određenoj šifri treba uneti šifru torte u odgovarajuće polje za unos i zatim pritisnuti dugme "FILTRIRAJ". Nakon toga, aplikacija će automatski pretražiti svoju bazu podataka i prikazati samo one torte čiji naziv odgovara unetom kriterijumu.

UNOS TORTE

Šifra:

Informacije o torti:

Naziv: Oblik torte: Ukus:

Informacije o kupovini:

Količina: Cena: Datum prodaje:

Unesite šifru torte koju želite da obrišete: **BRISANJE**

UNOS **SNIMI** **IZMENA** **ODUSTANI**

PRIKAZ PRODATIH TORTI

	Sifra	Naziv	OblikTorte	Ukus	Kolicin	Cena	Datum
▶	CvetnaK ...	Cvetna	kocka	jagoda		6000	20.9.2023
*							

<

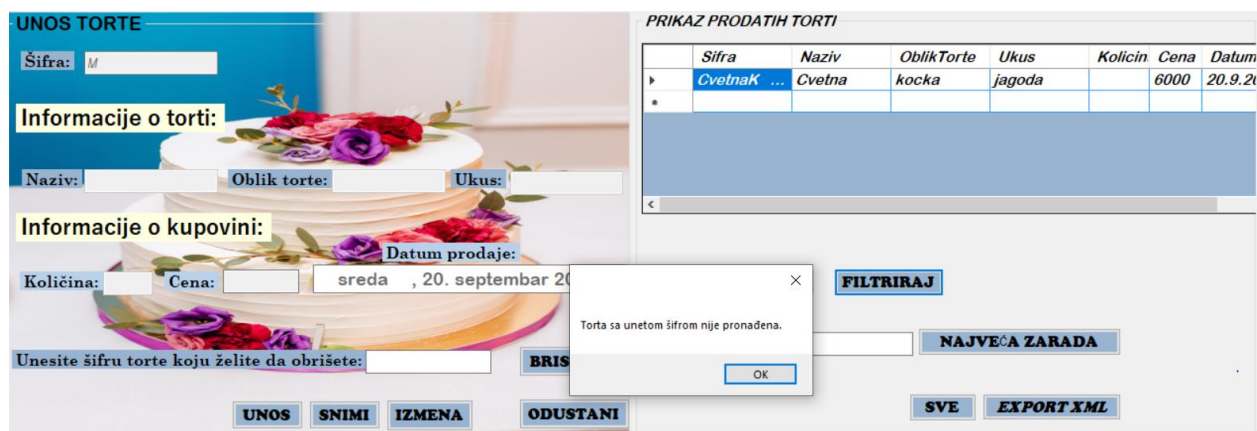
Naziv: **FILTRIRAJ**

Najveća zarada: **NAJVEĆA ZARADA**

SVE **EXPORT XML**

Slika 15. Filtriranje torte prema šifri

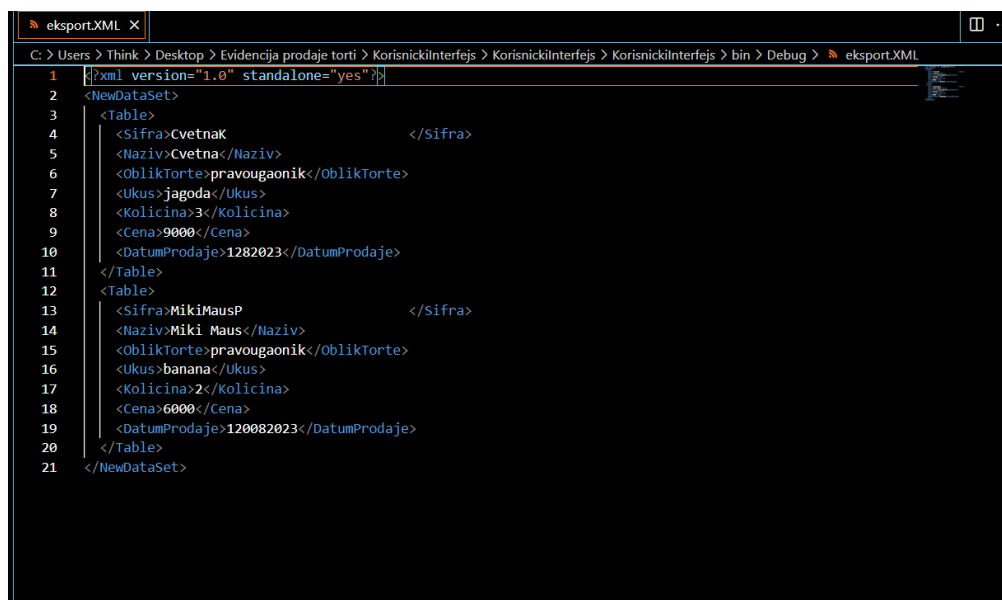
Ako se unese šifra torte koja se ne nalazi u bazi iskočiće poruka "Torta sa unetom šifrom ne postoji".



Slika 16 .Filtiranje torte sa šifrom koja ne postoji u bazi

6. Eksport podataka

Kad korisnik klikne na dugme "EXPORT XML" aplikacija će automatski generisati XML datoteku koja sadrži sve informacije o prodanim tortama.



Slika 17. Prikaz XML datoteke sa podacima o prodanim torti.

7. Prikaz najveće zarade

Kada korisnik klikne na dugme "NAJVEĆA ZARADA" prikazaće se najveća zarada.

The screenshot shows a web application interface for managing cakes. On the left, the 'UNOS TORTE' (Add Cake) form is visible, featuring a background image of a decorated cake. The form includes fields for 'Šifra' (ID), 'Naziv' (Name), 'Oblik torte' (Cake Shape), 'Ukus' (Flavor), 'Količina' (Quantity), 'Cena' (Price), and 'Datum prodaje' (Sale Date). Below these are buttons for 'BRISANJE' (Delete), 'UNOS', 'SNIMI' (Save), 'IZMENA' (Edit), and 'ODUSTANI' (Cancel). On the right, the 'PRIKAZ PRODATIH TORTI' (Display Sold Cakes) table is shown. It contains two rows of data. Below the table are buttons for 'FILTIRAJ' (Filter), 'Najveća zarada: 6000' (Maximum Profit: 6000), 'NAJVEĆA ZARADA' (Maximum Profit), 'SVE' (All), and 'EXPORT XML'.

Sifra	Naziv	OblikTorte	Ukus	Kolicin	Cena	Datum
CvetnaK ...	Cvetna	kocka	jagoda	2	6000	20.9.2023.
MikiK ...	Miki Maus	kocka	malina	1	3000	3.9.2023.

Slika 18. Prikaz najveće zarade

3. Opis implementacije

Prvi korak u razvoju aplikacije je kreiranje baze podataka koja sadrži podatke o torti.

Kreirana je baza podataka "Torte" u programu SQL Management Server Studio 2019.

Baza podataka "Torte" sadrži tabelu "Torta" sa atributima: Sifra, Naziv, Ukus, Cena, Oblik Torte, Kolicina, Datum Prodaje.

```
USE [master]
```

```
GO
```

```
CREATE DATABASE [Tortee]
```

```
GO
```

```
USE [Tortee]
```

```
GO
```

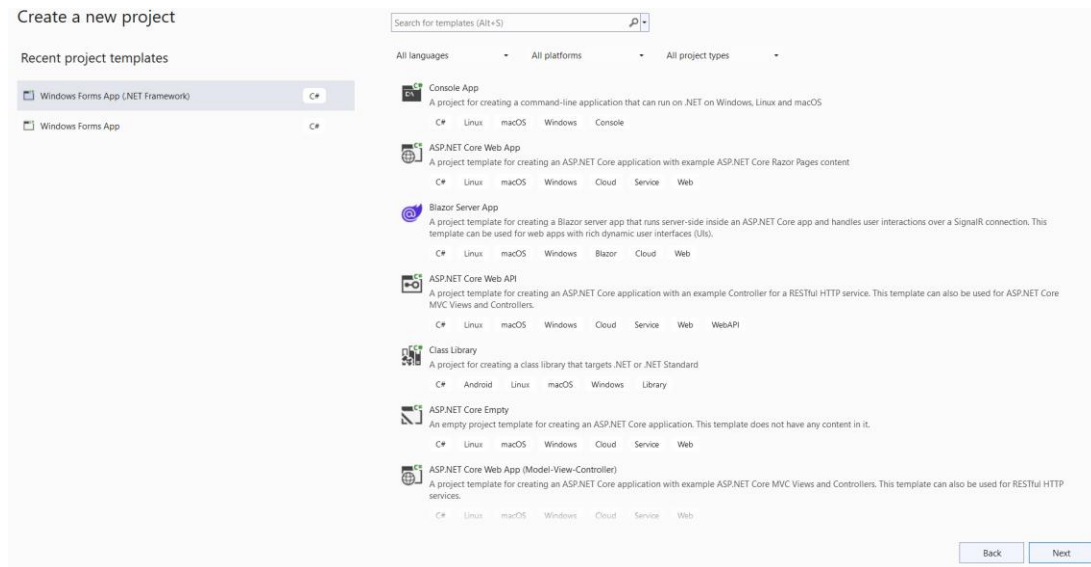
```
CREATE TABLE [dbo].[Torta](
    [Sifra] [Char] (30) NOT NULL,
    [Naziv] [nvarchar](50) NOT NULL,
    [OblikTorte] [nvarchar](50) NOT NULL,
    [Ukus] [nvarchar](50) NOT NULL,
    [Kolicina] [int] NOT NULL,
    [Cena] [int] NOT NULL,
    [DatumProdaje] date NOT NULL,
)
GO

ALTER TABLE [dbo].[Torta]
ADD CONSTRAINT [PK_Torta] PRIMARY KEY CLUSTERED
(
    [Sifra] ASC
)
GO
```

Listing 1. Kreiranje baze podataka "Torte"

Nakon kreiranja baze podataka treba kreirati izgled aplikacije.

Otvoriti Visual Studio 2022 -> Create a new Project->Izabrati Windows Forms App (.NET Framework)



Slika 19. Kreiranje novog projekta

Zatim, uneti ime projekta → Create

Configure your new project

Windows Forms App (.NET Framework) C# Windows Desktop

Project name

WindowsFormsApp13

Location

C:\Users\Think\source\repos

Solution name ⓘ

WindowsFormsApp13

☐ Place solution and project in the same directory

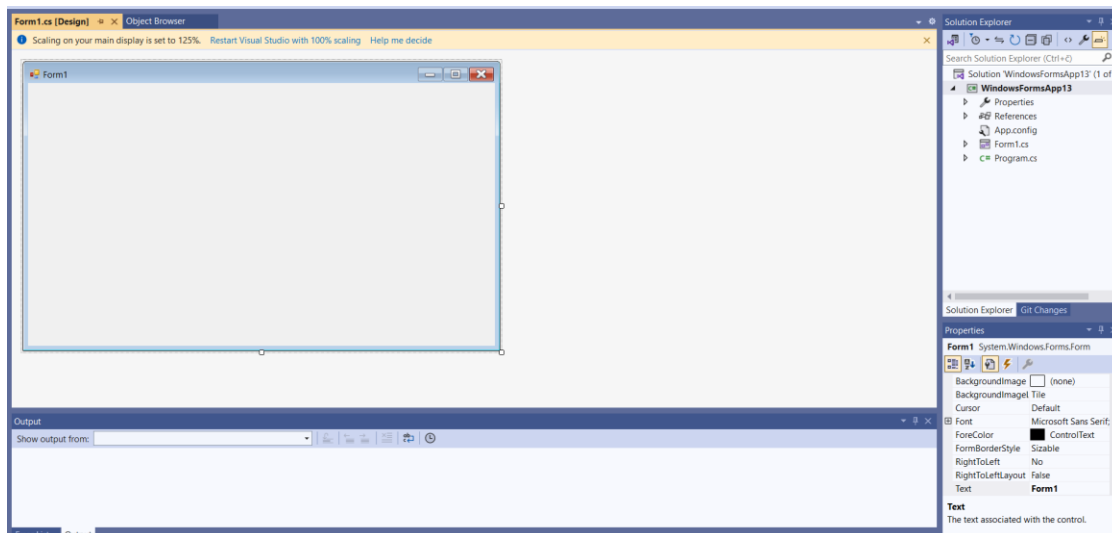
Framework

.NET Framework 4.7.2

Project will be created in "C:\Users\Think\source\repos\WindowsFormsApp13\WindowsFormsApp13\"

Slika 20. Unos imena projekta

Prilikom kreiranja prikazuje se prozor koji trenutno izgleda ovako. (Slika 21.)



Slika 21. Trenutan prozor prilikom kreiranja projekta

Da bi trenuti prozor prilagodili izgledu prozora sa slike ispod treba dodati određene elemente iz Toolbox-a(alatke). (Slika 22.)

UNOS TORTE

Šifra:

Informacije o torti:

Naziv: Oblik torte: Ukus:

Informacije o kupovini:

Količina: Cena: Datum prodaje:

Unesite šifru torte koju želite da obrišete: **BRISANJE**

UNOS SNIMI IZMENA ODUSTANI

PRIKAZ PRODATIH TORTI

	Šifra	Naziv	OblikTorte	Ukus	Kolicin	Cena	Datum
▶	CvetnaK ...	Cvetna	kocka	jaboda	2	6000	20.9.2
*	MikiK ...	Miki Maus	kocka	malina	1	3000	3.9.20

FILTRIRAJ

Najveća zarada: **NAJVEĆA ZARADA**

SVE EXPORT XML

Slika 22. Izgled korisničkog interfejsa

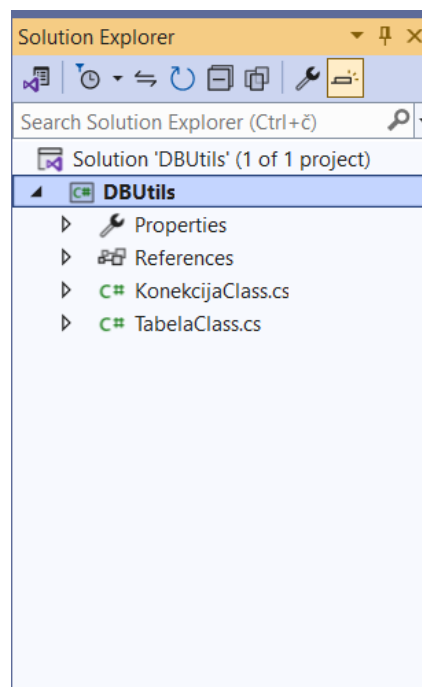
Nakon što smo sredili izgled prozora aplikacije, aplikaciju treba povezati sa bazom podataka, to radimo na sledeći način:

Da bismo omogućili vezu sa bazom podataka u okviru aplikacije moramo dodati klasu Parametri.cs gde ćemo postaviti odgovarajuće parametre koji se koriste za definisanje veze sa bazom podataka (naziv servera , naziv baze) i putanja za čuvanje XML datoteke.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 //
6 using System.Windows.Forms;
7
8 namespace KorisnickiInterfejs
9 {
10     1 reference
11     public static class Parametri
12     {
13         public static string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\MSSQL2008; Initial Catalog=Torteel; Integrated Security=True;";
14         public static string putanjaXML = Application.StartupPath + "\\eksport.XML";
15     }
16 }
17
```

Slika 23. Klasa Parametri.cs

Kada se uspostavi veza sa bazom podataka, potrebno je kreirati projekat DBUtils.



Slika 24. Projekat DBUtils

DBUtils sadrži dve klase: KonekcijaClass.cs i TabelaClass.cs.

1. KonekcijaClass.cs

Klasa KonekcijaClass.cs predstavlja osnovu za upravljanje vezom sa bazom podataka u aplikaciji. Sadrži atribut za čuvanje informacija o vezi, kao što su putanja do baze, naziv baze, i naziv instance DBMS-a. Takođe, sadrži konstruktore za kreiranje objekata klase sa odgovarajućim parametrima, metodu za otvaranje veze sa bazom, metodu za dobijanje same veze, i metodu za zatvaranje veze. Ova klasa omogućava aplikaciji da uspostavi komunikaciju sa bazom podataka i izvršava SQL upite.

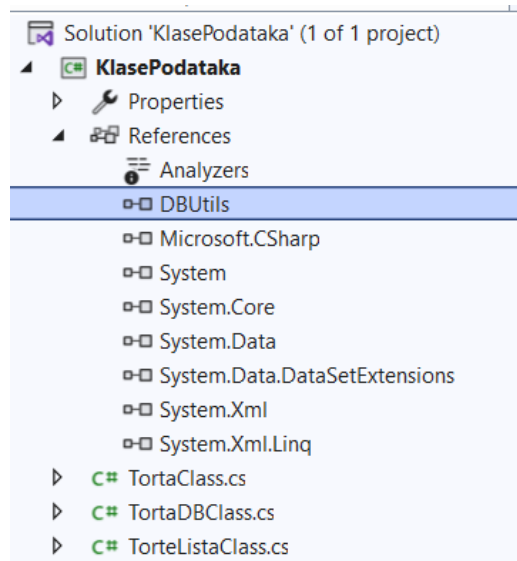
2. TabelaClass.cs

TabelaClass.cs je klasa koja omogućava manipulaciju podacima u tabelama baze podataka. Ova klasa sadrži atribut za rad sa određenom tabelom, kao što je naziv tabele, SqlDataAdapter za izvršavanje SQL upita, DataSet za čuvanje podataka iz tabele i objekat klase KonekcijaClass za upravljanje vezom sa bazom podataka. Takođe, klasa sadrži različite konstruktore za inicijalizaciju objekata, metode za izvođenje SQL upita i ažuriranje podataka u tabeli.

Sledeće što je potrebno uraditi jeste uključiti biblioteku DBUtils.dll u projekat KlasePodataka.sln. Da bismo dobili biblioteku DBUtils.dll potrebno je uraditi Build projekta.

Priključenje datoteke se radi na sledeći način:

References → Add references → DBUtils.dll → Add.



Slika 25. Dodavanje biblioteke DBUtils.dll

Klasa podataka sadrži tri klase:

1. Klasa TorteListaClass služi kao kolekcija za upravljanje objektima klase TortaClass u aplikaciji. Ova klasa omogućava dodavanje, brisanje i izmenu torti u listi.
2. Klasa TortaClass služi kao šablon za kreiranje objekata koji predstavljaju tortu u aplikaciji. Svaka instanca ove klase predstavlja jedan zapis o torti i ima attribute koji čuvaju različite informacije o toj torti, kao što su šifra, naziv, cena, datum prodaje, oblik torte, ukus i količina.
3. Klasa TortaDBClass omogućava interakciju sa bazom podataka uključujući prikaz svih torti, pretragu torti po nazivu, unos novih torti, brisanje postojećih torti i izmenu informacija o tortama. Takođe, omogućava i dobijanje informacija o torti sa najvećom cenom.

U okviru klase TortaDBClass imamo implemetirane sledeće metode:

1. DajSveTorte().

Ova metoda izvršava upit nad bazom podataka da bi prikazala sve informacije o tortama.

Prvo se uspostavlja veza sa bazom pomoću definisanog stringa za konekciju, a zatim se izvršava SQL upit SELECT koji selektuje sve redove iz tabele "Torta". Rezultati upita se popunjavaju u DataSet.

```
public DataSet DajSveTorte()
{
    DataSet podaciDataSet = new DataSet();
    string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS; Initial
Catalog=Tortee; Integrated Security=True;";
    using (SqlConnection konekcija = new SqlConnection(stringKonekcije))
    {
        string upit = "Select * from Torta";

        using (SqlDataAdapter adapter = new SqlDataAdapter(upit, konekcija))
        {
            adapter.Fill(podaciDataSet);
        }
    }

    return podaciDataSet;
}
```

Listing 2. Kod za prikazivanje podataka o tortama iz baze

2. DajTortuPremaSifri()

Ova metoda služi za dobijanje informacija o određenoj torti iz baze podataka na osnovu šifre.

Prvo se uspostavlja veza sa bazom pomoću definisanog stringa za konekciju, a zatim se izvršava SQL upit SELECT koji traži tortu sa šifrom koja se poklapa sa prosleđenim parametrom. Ako se pronađe odgovarajući rezultat funkcija vraća ovu instancu sa svim informacijama o traženoj torti. Na kraju, veza sa bazom podataka se zatvara.

```
public TortaClass DajTortuPremaSifri(string sifraTorteParametar)
{
    TortaClass torta = null;

    string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS;
Initial Catalog=Tortee; Integrated Security=True;";
    using (SqlConnection konekcija = new SqlConnection(stringKonekcije))
    {
        try
        {
            konekcija.Open();

            string upit = "SELECT * FROM Torta WHERE Sifra = @Sifra";
            SqlCommand komanda = new SqlCommand(upit, konekcija);
            komanda.Parameters.AddWithValue("@Sifra", sifraTorteParametar);

            using (SqlDataReader reader = komanda.ExecuteReader())
            {
                if (reader.Read())
                {
                    torta = new TortaClass();
                    torta.Sifra = reader["Sifra"].ToString();
                    torta.Naziv = reader["Naziv"].ToString();
                    torta.Ukus = reader["Ukus"].ToString();
                    torta.OblikTorte = reader["OblikTorte"].ToString();
                    torta.Cena = reader["Cena"].ToString();
                    string datumIzBaze = reader["DatumProdaje"].ToString();
                    DateTime datum = DateTime.ParseExact(datumIzBaze, "dd/MM/yyyy",
CultureInfo.InvariantCulture);
                    torta.DatumProdaje = datum;
                }
            }
        }
        catch { }
    }
}
```

```

        torta.Kolicina = reader["Kolicina"].ToString();

    }
}
}
catch (Exception ex)
{

}
finally
{
    konekcija.Close();
}
}

return torta;
}

```

Listing 3. Kod za prikaz torte sa određenom šifrom iz baze

3. NajvecaZarada()

Ova metoda služi za dobijanje najveće zarade.

Prvo se uspostavlja veza sa bazom pomoću definisanog stringa za konekciju, a zatim se izvršava SQL upit SELECT koji traži maksimalnu vrednost u koloni "Cena" tabele "Torta", i zatim vraća maksimalnu vrednost kao decimalni broj. Ako je upit uspešan, funkcija će vratiti najveću cenu iz tabele "Torta". Na kraju, veza sa bazom podataka se zatvara.

```

public decimal NajvecaZarada()
{
    decimal najvecaZarada = 0;
    string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS;
Initial Catalog=Tortee; Integrated Security=True;";

    using (SqlConnection konekcija = new SqlConnection(stringKonekcije))
    {
        try
        {
            konekcija.Open();
            string upit = "SELECT TOP 1 Cena AS NajvecaZarada FROM Torta";

```

```

SqlCommand komanda = new SqlCommand(upit, konekcija);

object rezultat = komanda.ExecuteScalar();

if (rezultat != null && rezultat != DBNull.Value)
{
    najvecaZarada = Convert.ToDecimal(rezultat);
}
}
catch (Exception ex)
{
}
finally
{
    konekcija.Close();
}
}

return najvecaZarada;
}

```

Listing 4. Kod za prikaz najveće zarade iz baze

4. SnimiNovuTortu()

Ova metoda omogućava dodavanje nove torte u bazu.

Prvo se uspostavlja veza sa bazom podataka, zatim se koristi SQL upit INSERT za unos nove torte u tabelu "Torta". Nakon toga, upit se izvršava koristeći ExecuteNonQuery() funkciju, koja vraća broj redova koji su ažurirani. Ako je broj ažuriranih redova veći od nule, funkcija vraća true, što znači da je unos uspeo. U suprotnom, vraća false. Veza sa bazom podataka se automatski zatvara nakon izvršenja upita.

```

public bool SnimiNovuTortu(TortaClass novaTortaObjectParametar)
{
    bool uspeh = false;

    string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS; Initial
Catalog=Tortee; Integrated Security=True;";

```

```

using (SqlConnection konekcija = new SqlConnection(stringKonekcije))
{
    string upit= "INSERT INTO Torta(Sifra, Naziv, OblikTorte, Ukus, Kolicina, Cena,
DatumProdaje) " +
                "VALUES (@Sifra, @Naziv, @OblikTorte, @Ukus, @Kolicina, @Cena,
@DatumProdaje)";

    using (SqlCommand komanda = new SqlCommand(upit, konekcija))
    {
        komanda.Parameters.AddWithValue("@Sifra", novaTortaObjectParametar.Sifra);
        komanda.Parameters.AddWithValue("@Naziv", novaTortaObjectParametar.Naziv);
        komanda.Parameters.AddWithValue("@OblikTorte",
novaTortaObjectParametar.OblikTorte);
        komanda.Parameters.AddWithValue("@Ukus", novaTortaObjectParametar.Ukus);
        komanda.Parameters.AddWithValue("@Kolicina",
novaTortaObjectParametar.Kolicina);
        komanda.Parameters.AddWithValue("@Cena", novaTortaObjectParametar.Cena);
        komanda.Parameters.AddWithValue("@DatumProdaje",
novaTortaObjectParametar.DatumProdaje);

        konekcija.Open();

        int rowsAffected = komanda.ExecuteNonQuery();
        uspeh = rowsAffected > 0;
    }
}

return uspeh;
}

```

Listing 5. Kod za upis podataka u bazu

5. ObrisiTortu()

Ova metoda omogućava brisanje torte iz baze podataka na osnovu zadate šifre torte koja se nalazi u objektu tortaZaBrisanjeObjectParametar.

Prvo se uspostavlja veza sa bazom podataka, a zatim se koristi SQL upit kako bi se uklonila torta iz tabele "Torta" sa odgovarajućom šifrom.

Parametar @Sifra u upitu se postavlja na osnovu šifre torte iz objekta tortaZaBrisanjeObjectParametar. Nakon izvršenja upita, funkcija vraća true ako je uspešno obrisana barem jedna torta, u suprotnom vraća false. Veza sa bazom podataka se automatski zatvara nakon izvršenja upita.

```

public bool ObrisiTortu(TortaClass tortaZaBrisanjeObjectParametar)
{
    bool uspeh = false;
    string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS;
Initial Catalog=Tortee; Integrated Security=True;";

    string upit = "DELETE FROM Torta WHERE Sifra=@Sifra";

    using (SqlConnection konekcija = new SqlConnection(stringKonekcije))
    {
        using (SqlCommand komanda = new SqlCommand(upit, konekcija))
        {
            komanda.Parameters.AddWithValue("@Sifra",
tortaZaBrisanjeObjectParametar.Sifra);

            try
            {
                konekcija.Open();
                int brojRedova = komanda.ExecuteNonQuery();
                uspeh = brojRedova > 0;
            }
            catch (Exception ex)
            {
                uspeh = false;
            }
        }
    }

    return uspeh;
}

```

Listing 6. Kod za brisanje torte iz baze

6. IzmeniTortu()

Ova metoda omogućava ažuriranje informacija o torti u bazi podataka.

Na osnovu stare torte, identifikovane šifrom staraTorta, ažurira se naziv, oblik, ukus, količina, cena i datum prodaje sa vrednostima nove torte, novaTorta. Koristi se SQL UPDATE upit za izmenu podataka u tabeli "Torta" na osnovu šifre stare torte. Ako je

uspešno ažuriran barem jedan red u bazi, funkcija vraća true, inače vraća false. Ova funkcija omogućava promenu podataka o torti u bazi podataka na osnovu šifre.

```
public bool IzmeniTortu(TortaClass staraTorta, TortaClass novaTorta)
{
    bool uspeh = false;

    string upit = "UPDATE Torta SET Naziv=@NoviNaziv,
OblikTorte=@NoviOblikTorte, Ukus=@NoviUkus, Kolicina=@NovaKolicina,
Cena=@NovaCena, DatumProdaje=@NoviDatumProdaje WHERE Sifra=@StaraSifra";
    string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS;
Initial Catalog=Tortee; Integrated Security=True;";

    using (SqlConnection konekcija = new SqlConnection(stringKonekcije))
    {
        konekcija.Open();

        using (SqlCommand komanda = new SqlCommand(upit, konekcija))
        {
            komanda.Parameters.AddWithValue("@NoviNaziv", novaTorta.Naziv);
            komanda.Parameters.AddWithValue("@NoviOblikTorte",
novaTorta.OblikTorte);
            komanda.Parameters.AddWithValue("@NoviUkus", novaTorta.Ukus);
            komanda.Parameters.AddWithValue("@NovaKolicina", novaTorta.Kolicina);
            komanda.Parameters.AddWithValue("@NovaCena", novaTorta.Cena);
            komanda.Parameters.AddWithValue("@NoviDatumProdaje",
novaTorta.DatumProdaje);
            komanda.Parameters.AddWithValue("@StaraSifra", staraTorta.Sifra);
            komanda.Parameters.AddWithValue("@datum", DateTime.Parse("2023-09-
20"));
            int brojAžuriranihRedova = komanda.ExecuteNonQuery();

            if (brojAžuriranihRedova > 0)
            {
                uspeh = true;
            }
            else
            {
                uspeh = false;
            }
        }
    }

    return uspeh;
}
```

Listing 7. Kod za izmenu torte u bazi

Kad završimo sa kreiranjem projekta KlasePodataka potrebno je biblioteku KlasePodataka.dll uključiti u aplikaciju na isti način kao što smo biblioteku DBUtils.dll.

Sledeći korak je implementacija funkcionalnosti dugmadi koja će omogućiti korisnicima da rukuju podacima o tortama. Dvoklikom na dugmad otvaramo prozor za implementaciju određenih akcija. Ovaj korak omogućava da definišemo konkretne operacije koje će se izvršiti kada korisnici kliknu na dugmad.

1. Implementacija dugmeta "UNOS"

Prilikom unosa podataka pozivaju se dve metode i fokus na određeno polje.

```
private void btnUnos_Click(object sender, EventArgs e)
{
    IsprazniKontrole();
    AktivirajKontrole();
    OznakaTextBox.Focus();
}
```

Listing 8. Implementacija dugmeta "UNOS"

IsprazniKontrole();

Ova funkcija se poziva kako bi se ispraznila tekstualna polja na korisničkom interfejsu.

```
private void IsprazniKontrole()
{
    OznakaTextBox.Text = "";
    NazivTextBox.Text = "";
    KolicinaTextBox.Text = "";
    OblikTorteTextBox.Text = "";
    DatumProdajeTextBox.Text = "";
    CenaTextBox.Text = "";
    UkusTextBox.Text = "";
    BrisanjeOznakaTextBox.Text = "";
}
```

Listing 9. Kod za metodu koja prazni tekstualna polja

i AktivirajKontrole();

Ova funkcija se poziva kako bi se omogućile ili aktivirale određena tekstualna polja na korisničkom interfejsu.

```
private void AktivirajKontrole()
{
    OznakaTextBox.Enabled = true;
    NazivTextBox.Enabled = true;

    KolicinaTextBox.Enabled = true;
    UkusTextBox.Enabled = true;

    CenaTextBox.Enabled = true;
    DatumProdajeTextBox.Enabled = true;

    BrisanjeOznakaTextBox.Enabled = true;
    OblikTorteTextBox.Enabled = true;
```

Listing 10. Kod za aktivaciju tekstualnih polja

2. Implementacija dugmeta "SNIMI"

Preuzimaju se vrednosti sa korisničkog interfejsa, zatim se koristi funkcija za snimanje u bazu. Nakon snimanja, prikazuje se poruka korisniku, a određeni elementi na interfejsu se onemogućavaju kako bi se sprečilo dalje menjanje podataka.

```
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdaje.Value;

    ObjekatTorte.OblikTorte = OblikTorteTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;
```

```

bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
if (uspesnoSnimljeno)
{
    poruka = "Uspesno snimljeno!";
}
else
{
    poruka = "Nije uspesno snimljeno!";
}
DeaktivirajKontrole();
MessageBox.Show(poruka);
}

```

Listing 11. Implementacija dugmeta "SNIMI"

3. Implementacija dugmeta "ODUSTANI"

Ukoliko korisnik želi da odustane od unos potrebno je da klikne na dugme "ODUSTANI"

Pozivaju se dve metode IsprazniKontrole(); koja se koristi kako bi se ispraznila tekstualna polja i

```

private void btnOdustani_Click(object sender, EventArgs e)
{
    IsprazniKontrole();
    DeaktivirajKontrole();
}

```

Listing 12. Implementacija dugmeta "ODUSTANI"

Funkcija "DeaktivirajKontrole" služi za onemogućavanje tekstualnih polja nakon što su podaci uneti ili ažurirani.

```

private void DeaktivirajKontrole()
{
    OznakaTextBox.Enabled = false;
    NazivTextBox.Enabled = false;
    DatumProdajeTextBox.Enabled = false;
    OblikTorteTextBox.Enabled = false;
    CenaTextBox.Enabled = false;
    UkusTextBox.Enabled = false;
    KolicinaTextBox.Enabled = false;
}

```

```
OblikTorteTextBox.Enabled = false;  
}
```

Listing 13. Implementacija metode koja onemogućuje tekstualna polja

4. Implementacija dugmeta "IZMENA"

Izmena torte funkcinoše na sledeći način:

Korisnik selektuje red u DataGridView.

Elementu pod nazivom " SpisakTortiDataGridView" dodali smo funkciju SpisakTortiDataGridView_CellClick koja omogućava korisnicima da pregledaju detalje torte selektovane klikom na određeni red u tabeli.

Kada korisnik klikne na bilo koji red u tabeli, funkcija proverava da li je odabrani red validan (e.RowIndex >= 0).

Ako jeste, uzima vrednosti iz odgovarajućih ćelija u tom redu i prikazuje ih u odgovarajućim tekstualnim poljima.

```
private void SpisakTortiDataGridView_CellClick(object sender,  
DataGridViewCellEventArgs e)  
{  
    if (e.RowIndex >= 0)  
    {  
        DataGridViewRow row = this.SpisakTortiDataGridView.Rows[e.RowIndex];  
  
        OznakaTextBox.Text = row.Cells[0].Value.ToString();  
        NazivTextBox.Text = row.Cells[1].Value.ToString();  
        OblikTorteTextBox.Text = row.Cells[2].Value.ToString();  
        UkusTextBox.Text = row.Cells[3].Value.ToString();  
        KolicinaTextBox.Text = row.Cells[4].Value.ToString();  
        CenaTextBox.Text = row.Cells[5].Value.ToString();  
        DatumProdajeTextBox.Text = row.Cells[6].Value.ToString();  
    }  
}
```

Listing 14. Implementacija metode koja omogućuje popunjavanje tekstualnih polja za selektovani red u tabeli

Kada korisnik izvrši željene izmene potrebno je da klikne dugme "Izmena".

Prvo se pripremaju objekti za staru i novu tortu, gde se šifra torte koristi za identifikaciju. Nakon što korisnik unese nove podatke u tekstualna polja funkcija poziva metodu za izmenu podataka.

Rezultat izmene se prikazuje korisniku putem poruke, a tekstualna polja se deaktiviraju kako bi se onemogućila dalja izmena dok se ne klikne na dugme za unos ili izmenu novih podataka.

```
private void IzmenaButton_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass staraTorta = new TortaClass
    {
        Sifra = OznakaTextBox.Text
    };

    TortaClass novaTorta = new TortaClass
    {
        Sifra = OznakaTextBox.Text,
        Ukus = UkusTextBox.Text,
        Kolicina = KolicinaTextBox.Text,
        OblikTorte = OblikTorteTextBox.Text,
        DatumProdaje = DatumProdaje.Value,
        Naziv = NazivTextBox.Text,
        Cena = CenaTextBox.Text
    };

    bool uspesnoIzmenjeno = TortaDBObject.IzmeniTortu(staraTorta, novaTorta);

    if (uspesnoIzmenjeno)
    {
        poruka = "Uspesno izmenjeno!";
    }
    else
    {
        poruka = "Nije uspesno izmenjeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}
```

Listing 15. Implementacija dugmeta "IZMENA"

5. Implementacija dugmeta "BRISANJE"

Zatim imamo funkcionalnost brisanja.

Ova funkcija omogućava korisnicima brisanje informacija o torti na osnovu unete šifre. Nakon što korisnik unese šifru i potvrdi brisanje, funkcija će izbrisati podatke o toj torti iz baze podataka i obavestiti korisnika o uspešnom ili neuspešnom brisanju.

```
private void BrisanjeButton_Click(object sender, EventArgs e)
{
    string sifraTorte = BrisanjeOznakaTextBox.Text;

    if (string.IsNullOrEmpty(sifraTorte))
    {
        MessageBox.Show("Unesite ispravnu šifru torte za brisanje.");
        return;
    }

    DialogResult poruka = MessageBox.Show("Da li ste sigurni da želite da obrišete ovu tortu?",
    "Potvrda brisanja", MessageBoxButtons.YesNo);

    if (poruka == DialogResult.Yes)
    {
        TortaClass tortaZaBrisanje = new TortaClass
        {
            Sifra = sifraTorte
        };

        bool uspesnoBrisanje = TortaDBObject.ObrisiTortu(tortaZaBrisanje);

        if (uspesnoBrisanje)
        {
            MessageBox.Show("Torta je uspešno obrisana!");
            IsprazniKontrola();
        }
        else
        {
            MessageBox.Show("Nije uspelo brisanje torte.");
        }
    }
    else
    {
        MessageBox.Show("Brisanje je otkazano.");
    }
}
```

```
}
```

Listing 16. Implementacija dugmeta "BRISANJE"

6. Implementacija dugmeta "FILTRIRAJ"

Sledeća funkcionalnost je filtriranje.

Ova funkcija omogućava korisnicima filtriranje informacija o tortama na osnovu unete šifre torte.

Korisnik unosi šifru torte, zatim se pretražuje baza podataka kako bi se pronašla tortu sa odgovarajućim nazivom. Ako se torta pronađe, podaci o njoj se zatim prikazuju u tabeli .

```
private void FiltrirajButton_Click(object sender, EventArgs e)
{
    string sifraTorte = FilterTextBox.Text;
    TortaClass torta = TortaDBObject.DajTortuPremaSifri(sifraTorte);

    if (torta != null)
    {
        DataTable tabela = new DataTable("Torta");
        tabela.Columns.Add("Sifra");
        tabela.Columns.Add("Naziv");
        tabela.Columns.Add("OblikTorte");
        tabela.Columns.Add("Ukus");
        tabela.Columns.Add("Kolicina");
        tabela.Columns.Add("Cena");
        tabela.Columns.Add("DatumProdaje");

        tabela.Rows.Add(torta.Sifra, torta.Naziv, torta.OblikTorte, torta.Ukus, torta.Kolicina,
            torta.Cena, torta.DatumProdaje);

        DataSet noviDataSet = new DataSet();
        noviDataSet.Tables.Add(tabela);
        PrikaziTabeluPodataka(noviDataSet);
    }
    else
    {
        MessageBox.Show("Torta sa unetom šifrom nije pronađena.");
    }
}
```


Listing 17. Implementacija dugmeta "FILTRIRAJ"

7. Implementacija dugmeta "NAJVEĆA ZARADA"

Zatim sledi opis funkcionalnosti najveće zarade.

Ova funkcija omogućava korisnicima da pronađu i prikažu najveću zaradu među tortama u evidenciji.

Klikom na dugme "Najveća Zarada" pretražuju se podaci. Nakon što se najveća zarada pronađe, vrednost se prikazuje u odgovarajuće tekstualno polje.

```
private void NajvecaZaradaButton_Click(object sender, EventArgs e)
{
    decimal najvecaZarada = TortaDBObject.NajvecaZarada();

    NajvecaZaradaTextBox.Text = najvecaZarada.ToString();
}
```

Listing 18. Implementacija dugmeta "NAJVEĆA ZARADA"

8. Implementacija dugmeta "EXPORT XML"

Zatim sledi opis eksportvanje u XML.

Funkcija SnimiXML koristi se za eksportovanje unetih podataka u tabeli.

```
private void btnExportXML_Click(object sender, EventArgs e)
{
    SnimiXML(podaciDataSet.Tables[0], Parametri.putanjaXML);
    MessageBox.Show("Uspesno realizovan eksport podataka!");
}
```

Listing 19. Implementacija dugmeta "EXPORT XML"

9. Implementacija dugmeta "SVE"

I na kraju opis funkcije koja omogućava prikazivanje prodanih torti.

```
private void btnSve_Click(object sender, EventArgs e)
{
    podaciDataSet = TortaDBObject.DajSveTorte();
    PrikaziTabeluPodataka(podaciDataSet);
}
```

Listing 20. Implementacija dugmeta "SVE"

4. Greške programskog koda

```
1 reference
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdajeTextBox.Text;
    ObjekatTorte.OblikTorte = OblikTorteTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = "Uspesno snimljeno!";
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}
```

Slika 26. Kod bez grešaka

```

private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdaje.Value;

    ObjekatTorte.OblikTorte = UkusTextBox.Text;
    ObjekatTorte.Ukus = OblikTorteTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = " ";
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}

```

Slika 27. Kod sa četiri vrste greške

Unutar metode btnSnimi_Click(object sender, EventArgs e) pronađene su četiri vrste grešaka:

1. Leksička greška

```

1 reference
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdajeTextBox.Text;
    ObjekatTorte.OblikTorte = OblikTorteTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

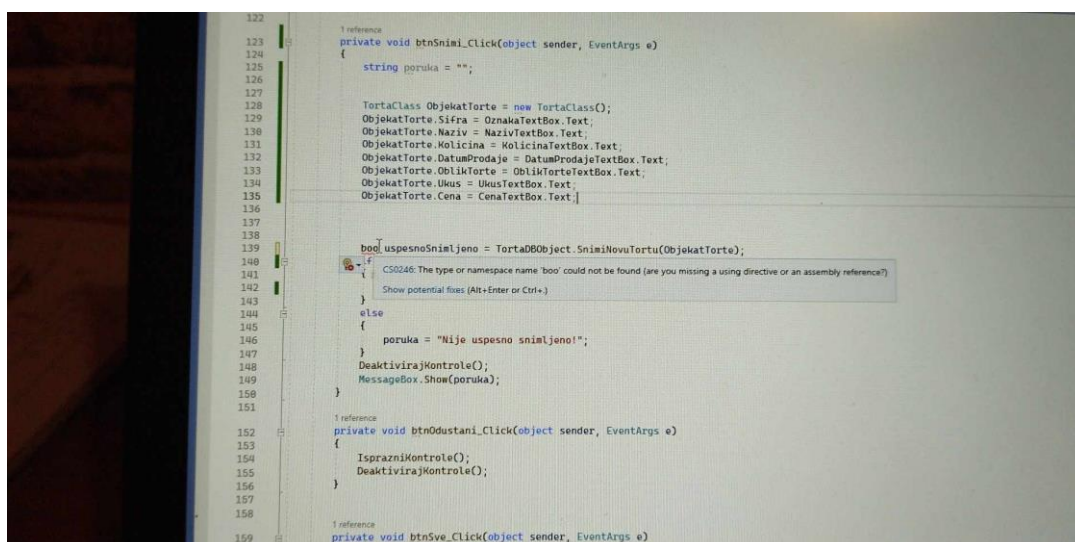
    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = "Uspesno snimljeno!";
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}

```

Slika 28. Leksička greška

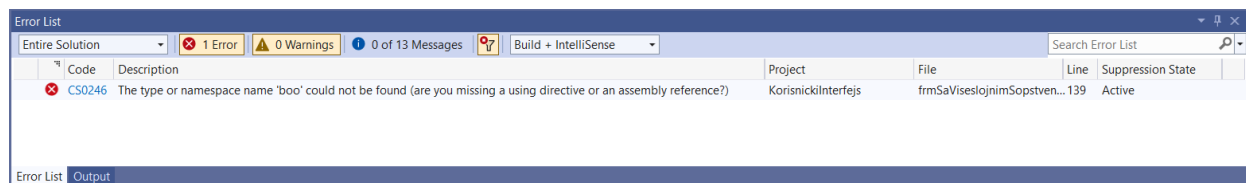
Objašnjenje greške: Neispravan oblik rezervisane reči. Umesto rezervisane reči "bool" napisano je "boo".

Reakcija editora:



Slika 29. Reakcija editora usled leksičke greške

Reakcija kompajlera:



Slika 30. Reakcija kompajlera usled leksičke greške

Korekcija greške:

Da bi se ispravila ova greška, potrebno je zameniti neispravan oblik "boo" sa ispravnom rezervisanom reči "bool".

```

1 reference
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdajeTextBox.Text;
    ObjekatTorte.OblikTorte = OblikTorteTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = "Uspesno snimljeno!";
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}

```

Slika 31. Otklonjena leksička greška

2. Sintaksna greška

```

1 reference
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdajeTextBox.Text;
    ObjekatTorte.OblikTorte = OblikTorteTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

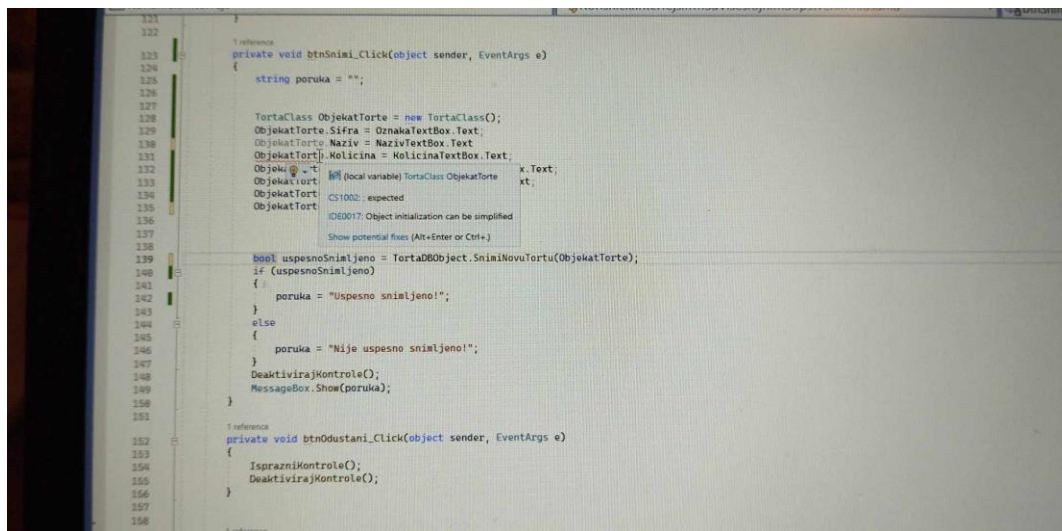
    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = "Uspesno snimljeno!";
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}

```

Slika 32. Sintaksna greška

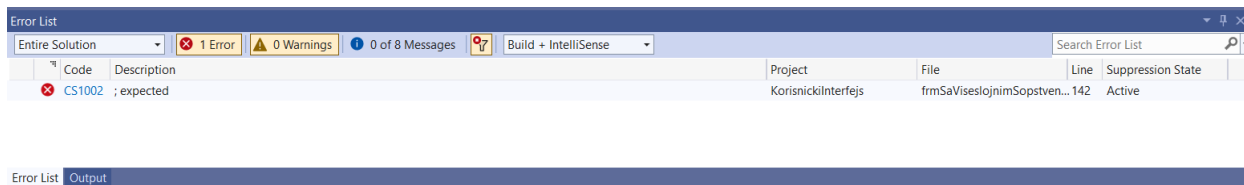
Objašnjenje greške: Neispravan oblik naredbe –nedostaje specijalni simbol (;) na kraju naredbe.

Reakcija editora:



Slika 33. Reakcija editora usled sintaksne greške

Reakcija kompajlera:



Slika 34. Reakcija kompajlera usled sintaksne greške

Korekcija greške:

Da bi se rešila ovu greška, potrebno je dodati ";" na kraj naredbe.

```

1 reference
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdajeTextBox.Text;
    ObjekatTorte.OblikTorte = OblikTorteTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = 0;
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}

```

Slika 35. Otklonjena sintaksna greška

3. Semantička greška

```

1 reference
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdajeTextBox.Text;
    ObjekatTorte.OblikTorte = OblikTorteTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

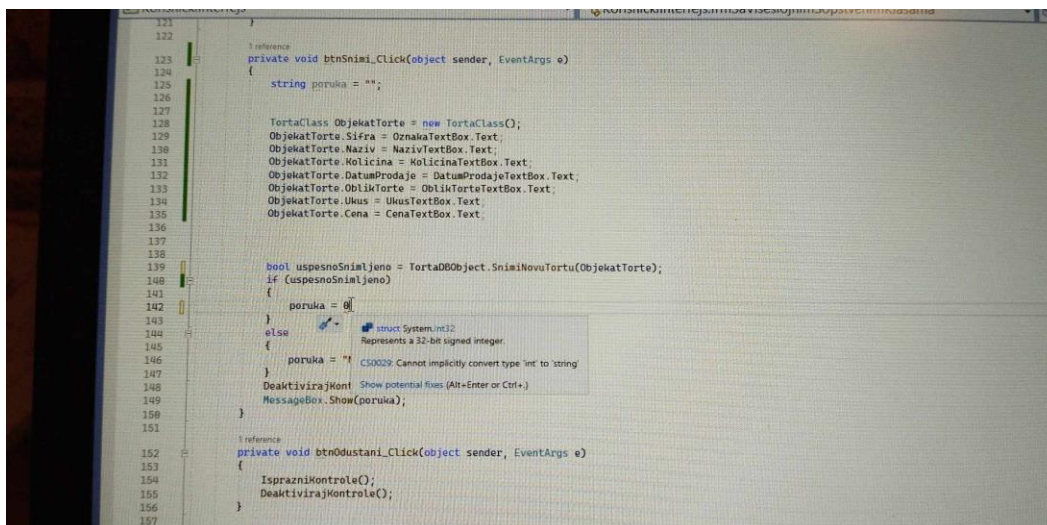
    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = 0;
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}

```

Slika 36. Semantička greška

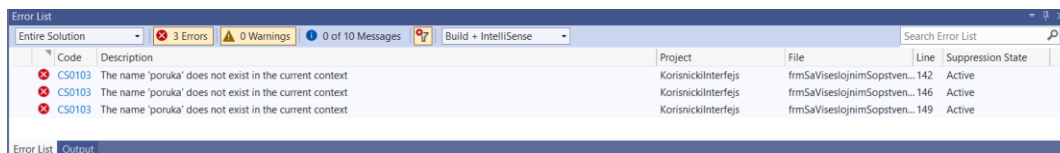
Objašnjenje greške: Vrednost se dodeljuje promenljivoj koja nije istog tipa kao i vrednost, a nije urađen typ casting

Reakcija editora:



Slika 37. Reakcija editora usled semnatičke greške

Reakcija kompajlera:



Slika 38. Reakcija kompajlera usled semantičke greške

Korekcija greške:

Usklađivanje tipova promenljive i vrednosti koja se dodeljuje. Npr. ako je promenljiva poruka deklarisan kao string, trebali bi dodeliti string vrednost, kao što je "0", "Uspesno snimljeno!", ili bilo koji drugi string koji odgovara kontekstu poruke.


```

1 reference
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdajeTextBox.Text;
    ObjekatTorte.OblikTorte = OblikTorteTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = "Uspesno snimljeno!";
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}

```

Slika 39. Otklonjena semantička greška

4. Logička greška

```

1 reference
private void btnSnimi_Click(object sender, EventArgs e)
{
    string poruka = "";

    TortaClass ObjekatTorte = new TortaClass();
    ObjekatTorte.Sifra = OznakaTextBox.Text;
    ObjekatTorte.Naziv = NazivTextBox.Text;
    ObjekatTorte.Kolicina = KolicinaTextBox.Text;
    ObjekatTorte.DatumProdaje = DatumProdaje.Value;

    ObjekatTorte.OblikTorte = UkusTextBox.Text;
    ObjekatTorte.Ukus = UkusTextBox.Text;
    ObjekatTorte.Cena = CenaTextBox.Text;

    bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
    if (uspesnoSnimljeno)
    {
        poruka = "Uspesno snimljeno!";
    }
    else
    {
        poruka = "Nije uspesno snimljeno!";
    }
    DeaktivirajKontrolu();
    MessageBox.Show(poruka);
}

```

Slika 40. Logička greška

UNOS TORTE

Šifra:

Informacije o torti:

Naziv: Oblik torte: Ukus:

Informacije o kupovini:

Količina: Cena: Datum prodaje:

Unesite šifru torte koju želite da obrišete:

BRISANJE

UNOS SNIMI IZMENA ODUSTANI

PRIKAZ PRODATIH TORTI

Sifra	Naziv	OblikTorte	Ukus	Kolicina	Cena
CvetnaK	Cvetna	jagoda	jagoda	3	12000

FILTRIRAJ

Najveća zarada: **NAJVEĆA ZARADA**

SVE EXPORT XML

Slika 41. Logička greška

Objašnjenje greške: Logička greška je nastala zato što atributu oblik torte postavljamo vrednost za ukus. Prilikom ovog tipa greške kompajler i editor neće prikazati grešku.

Korekcija greške:

Da bismo ispravili logičku grešku treba za atributu oblik torte postaviti vrednost oblik torte, a ne vrednost za ukus.

5. Runtime greška

```

0 references
public bool SnimiNovuTortu(TortaClass novaTortaObjectParametar)
{
    bool uspeh = false;

    string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS; Initial Catalog=Tort; Integrated Security=True;";

    using (SqlConnection konekcija = new SqlConnection(stringKonekcije))
    {
        string upit = "INSERT INTO Torta(Sifra, Naziv, OblikTorte, Ukus, Kolicina, Cena, DatumProdaje) " +
            "VALUES (@Sifra, @Naziv, @OblikTorte, @Ukus, @Kolicina, @Cena, @DatumProdaje)";

        using (SqlCommand komanda = new SqlCommand(upit, konekcija))
        {
            komanda.Parameters.AddWithValue("@Sifra", novaTortaObjectParametar.Sifra);
            komanda.Parameters.AddWithValue("@Naziv", novaTortaObjectParametar.Naziv);
            komanda.Parameters.AddWithValue("@OblikTorte", novaTortaObjectParametar.OblikTorte);
            komanda.Parameters.AddWithValue("@Ukus", novaTortaObjectParametar.Ukus);
            komanda.Parameters.AddWithValue("@Kolicina", novaTortaObjectParametar.Kolicina);
            komanda.Parameters.AddWithValue("@Cena", novaTortaObjectParametar.Cena);
            komanda.Parameters.AddWithValue("@DatumProdaje", novaTortaObjectParametar.DatumProdaje);

            konekcija.Open();

            int rowsAffected = komanda.ExecuteNonQuery();
            uspeh = rowsAffected > 0;
        }
    }

    return uspeh;
}

```

Slika 42. Runtime greška

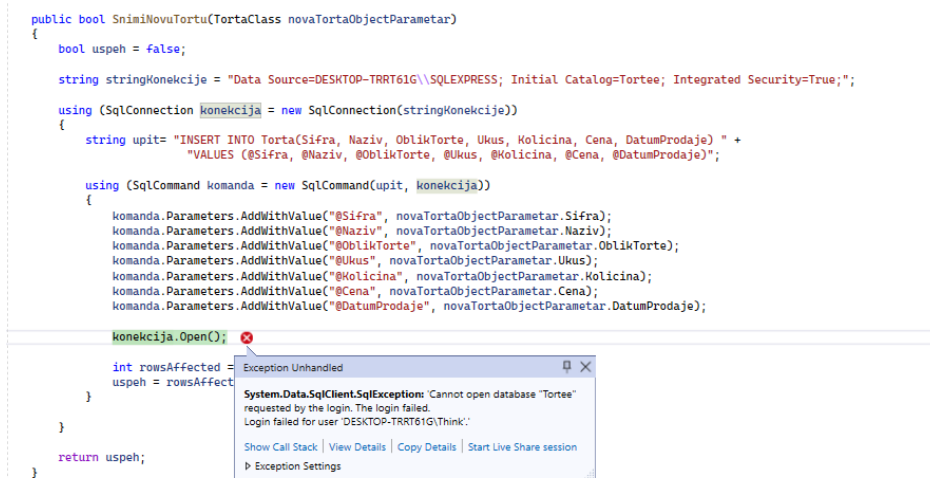
Objašnjenje greške:

Runtime greške su greške koje se dešavaju tokom izvršavanja programa i mogu uzrokovati prekid izvršavanja programa.

Nastala je zbog nepravilno napisanog naziva baze podataka u stringu konekcije prilikom otvaranja veze ka SQL Server. Ova greška se javlja kada pokušam da izvršim operaciju upisa podataka u bazu ali SQL Server ne može pronaći ili povezati se sa bazom podataka sa zadatim nazivom.

Prilikom ovog tipa greške kompajler i editor neće prikazati grešku.

Nakon pokušaja da se upišu podaci u tabelu tj. bazu korišćenjem dugmeta SNIMI izbaciće nas iz programa, podaci se neće upisati u tabelu i pojaviće se ova poruka. (Slika)



Slika 43. Runtime greška

Korekcija greške:

Ispravka ove greške zahteva da se tačno navede naziv baze podataka u string konekcije umesto naziva Tortee treba napisati Torte jer je ovo ispravan naziv baze. Prilikom upisa podataka u tabelu klikom na dugme SNIMI podaci će se upisati u tabelu.

```

0 references
public bool SnimiNovuTortu(TortaClass novaTortaObjectParametar)
{
    bool uspeh = false;

    string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS; Initial Catalog=Torte; Integrated Security=True;";

    using (SqlConnection konekcija = new SqlConnection(stringKonekcije))
    {
        string upit= "INSERT INTO Torte(Sifra, Naziv, OblikTorte, Ukus, Kolicina, Cena, DatumProdaje) * +
        "VALUES (@Sifra, @Naziv, @OblikTorte, @Ukus, @Kolicina, @Cena, @DatumProdaje)";

        using (SqlCommand komanda = new SqlCommand(upit, konekcija))
        {
            komanda.Parameters.AddWithValue("@Sifra", novaTortaObjectParametar.Sifra);
            komanda.Parameters.AddWithValue("@Naziv", novaTortaObjectParametar.Naziv);
            komanda.Parameters.AddWithValue("@OblikTorte", novaTortaObjectParametar.OblikTorte);
            komanda.Parameters.AddWithValue("@Ukus", novaTortaObjectParametar.Ukus);
            komanda.Parameters.AddWithValue("@Kolicina", novaTortaObjectParametar.Kolicina);
            komanda.Parameters.AddWithValue("@Cena", novaTortaObjectParametar.Cena);
            komanda.Parameters.AddWithValue("@DatumProdaje", novaTortaObjectParametar.DatumProdaje);

            konekcija.Open();

            int rowsAffected = komanda.ExecuteNonQuery();
            uspeh = rowsAffected > 0;
        }
    }

    return uspeh;
}

```

Slika 44. Otklonjena runtime greška

5. Formalna gramatika grešaka

1. **Leksička greška**- neispravan oblik rezervisane reči. Umesto rezervisane reči "bool" napisano je "boo".

boo uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);
--

Ispravno:

bool uspesnoSnimljeno = TortaDBObject.SnimiNovuTortu(ObjekatTorte);

Formalna gramatika

Sta rtn i si mb ol	<naredba_definicije _promenljive>
Te	"(", ")", ";", "bool", ".", "=",

rm ina li	
Va rija ble	<naredba_definicije_promenljive>,<deklaracija_promenljive>,<dodela_vrednosti>,<simbol_kraja_naredbe>, <tip_podatka>,<praznina>,<promenljiva>,<simbol_dodele_vrednosti>,<izraz_za_dodavanje_objekta_u_tabelu>,<objekat>,<funkcija_za_snimanje_podataka>,<naziv_funkcije>,<pocetak_funkcije>,<parametar_funkcije>,<kraj_funkcije>,<simbol_pristupa_funkciji>
Pra vil a	1. <naredba_definicije_promenljive>=<deklaracija_promenljive>,<dodela_vrednosti>,<simbol_kraja_naredbe>; 2.<deklaracija_promenljive>=<tip_podatka>,<praznina>,<promenljiva>; 3.<dodela_vrednosti>=<simbol_dodele_vrednosti>,<izraz_za_dodavanje_objekta_u_tabelu>; 4. <izraz_za_dodavanje_objekta_u_tabelu>=<objekat><simbol_pristupa_funkciji>,<funkcija_za_snimanje_podataka>; 5.<funkcija_za_snimanje_podataka>=<naziv_funkcije>,<pocetak_funkcije>,<parametar_funkcije>,<kraj_funkcije>; 6. <tip_podatka>="bool"; 7. <praznina>=" "; 8. <promenljiva>=<identifikator>; 9. <identifikator>={<slovo> } [<broj>] 10.<naziv_funkcije>=<identifikator>; 11. <slovo>="a" "b" "c"... 12. <broj>="1" "2" "3"... 13. <simbol_dodele_vrednosti>="="; 14. <simbol_pristupa_funkciji>="."; 15. <pocetak_funkcije>="("; 16. <kraj_funkcije>=")"; 17. <simbol_kraja_naredbe>=","; 18.<objekat>=<identifikator>; 19.<parametar_funkcije>=<identifikator>;

2. Sintaksna greška- neispravan oblik naredbe –nedostaje specijalni simbol (;) na kraju naredbe.

```
ObjekatTorte.Kolicina = KolicinaTextBox.Text
```

Ispravno:

```
ObjekatTorte.Kolicina = KolicinaTextBox.Text;
```

Formalna gramatika

Star tni sim bol	<naredba_definicije >
Ter min ali	"," , ".", "=", "Text"
Vari jabl e	<naredba_definicije>,<pristup_atributu>,<dodela_vrednosti>,<simbol_kraja_naredbe>,<pristup_tekstualnom_sadrzaju>,<objekat>,<simbol_za_izlistavanje>,<referenca_na_tekstualni_sadrzaj>,<atribut_kolicina>,<tekstualni_sadrzaj>
Prav ila	<ol style="list-style-type: none"> 1. <naredba_definicije>=<pristup_atributu>,<dodela_vrednosti>,<simbol_kraja_naredbe>; 2. <pristup_atributu>=<objekat>,<simbol_za_izlistavanje>,<atribut_kolicina>; 3. <dodela_vrednosti>=<simbol_dodele_vrednosti>,<pristup_tekstualnom_sadrzaju>; 4. <pristup_tekstualnom_sadrzaju>=<referenca_na_tekstualni_sadrzaj>,<simbol_za_izlistavanje>,<tekstualni_sadrzaj>; 5. <identifikator>={<slovo> } [<broj>] 6. <slovo>="a" "b" "c"... 7. <broj>="1" "2" "3"... 8. <simbol_dodele_vrednosti>="="; 9. <simbol_za_izlistavanje>="."; 10. <simbol_kraja_naredbe>=","; 11. <objekat>=<identifikator>; 12. <atribut_kolicina>=<identifikator>; 13. <referenca_na_tekstualni_sadrzaj>=<identifikator>; 14. <tekstualni_sadrzaj>="Text";

3. **Semantička greška** -Vrednost se dodeljuje promenljivoj koja nije istog tipa kao i vrednost, a nije urađen typ casting

```
poruka = 0;
```

Ispravno:

```
poruka = "Uspesno snimljeno!";
```

Formalna gramatika

Startni simbol	<naredba_definicije_vrednosti>
Terminali	"=", ",", " ", """,
Varijable	<promenljiva>, <dodela_vrednosti>, <izraz_stringa>, <simbol_kraja_naredbe>, <navodnik>, <niz_karaktera>, <praznina>
Pravila	1. <naredba_definicije_vrednosti>=<dodela_vrednosti>,<izraz_stringa>; 2. <dodela_vrednosti>=<simbol_dodele_vrednosti>,<izraz_stringa>; 3. <izraz_stringa>=<navodnik>,<niz_karaktera><navodnik>; 4. <navodnik>=" " "; 5. <praznina>=" " "; 6. <promenljiva>=<identifikator>; 7. <identifikator>={<slovo> }[[<broj>] 8. <slovo>="a" "b" "c" "... 9. <broj>="1" "2" "3" "... 10. <simbol_dodele_vrednosti>="=""; 11. <simbol_kraja_naredbe>=""; 12. <niz_karaktera>=<identifikator>;

4. **Logička greška** - nastala zato što atributu oblik torte postavljamo vrednost za ukus.

ObjekatTorte.OblikTorte = UkusTextBox.Text;

Ispravno:

ObjekatTorte.OblikTorte = UkusTextBox.Text;

Formalna gramatika

Star tni sim bol	<naredba_definicije>
Ter min ali	"," , ". ", "=", "Text"
Vari jabl e	<naredba_definicije>, <pristup_atributu>, <dodela_vrednosti>, <simbol_kraja_naredbe>, <pristup_tekstualnom_sadrzaju>, <objekat>, <simbol_za_izlistavanje>, <referenca_na_tekstualni_sadrzaj>, <atribut_oblik_torte>, <tekstualni_sadrzaj>
Prav ila	1. <naredba_definicije_promenljive>=<pristup_atributu>,<dodela_vrednosti><simbol_kraja

	_naredbe>; 2.<pristup_atributu>=<objekat>,<simbol_za_izlistavanje>,<atribut_oblik_torte>; 3.<dodela_vrednosti>=<simbol_dodele_vrednosti>,<pristup_tekstualnom_sadrzaju>; 4.<pristup_tekstualnom_sadrzaju>=<referenca_na_tekstualni_sadrzaj>,<simbol_za_izlistavanje>,<tekstualni_sadrzaj>; 5.<identifikator>={<slovo> }[[<broj>] 6.<slovo>="a" "b" "c"... 7.<broj>="1" "2" "3"... 8.<simbol_dodele_vrednosti>="="; 9.<simbol_za_izlistavanje>="."; 10.<simbol_kraja_naredbe>=","; 11.<objekat>=<identifikator>; 12.<atribut_oblik_torte>=<identifikator>; 13.<referenca_na_tekstualni_sadrzaj>=<identifikator>; 14.<tekstualni_sadrzaj>="Text";
--	--

5. Runtime greške – nesipravan naziv baze podataka

```
string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS; Initial
Catalog=Tort; Integrated Security=True;;";
```

Ispravno:

```
string stringKonekcije = "Data Source=DESKTOP-TRRT61G\\SQLEXPRESS; Initial
Catalog=Tortee; Integrated Security=True;;";
```

St art ni si m bo l	<naredba_definicije_promenljive>
Te rm in ali	","; "string", "=", "\\","SQLEXPRESS","Initial Catalog","Data Source","Integrated Security","True", """,

V ari ja bl e	<naredba_definicije_promenljive>,<deklaracija_promenljive>,<dodela_vrednosti>,<simbol_kraja_naredbe>,<tip_podatka>,<praznina>,<promenljiva>,<simbol_dodele_vrednosti>,<izraz_stringa>,<veza_sa_bazom_gde_se_nalazi_SQL_Server>,<izvor_podataka>,<naziv_racunara>,<simbol_razdvajanja_naziva_racunara_i_instance>,<naziv_instance>,<pocetni_katalog>,<naziv_baze>,<veza_sa_bazom_naziv_baze_sa_kojom_se_povezuje>,<SQL_Server_autentifikacija>,<mogucnost_autentifikacije>,<navodnik>,<string>,<veza_sa_bazom_specifikacija_nacina_autentifikacije>
Pr av ila	1. <naredba_definicije_promenljive>=<deklaracija_promenljive>,<dodela_vrednosti>,<simbol_kraja_naredbe>; 2.<deklaracija_promenljive>=<tip_podatka>,<praznina>,<promenljiva>; 3.<dodela_vrednosti>=<simbol_dodele_vrednosti>,<string>; 4.<izraz_stringa>=<navodnik>,<veza_sa_bazom_gde_se_nalazi_SQL_Server>,<veza_sa_bazom_naziv_baze_sa_kojom_se_povezuje>,<veza_sa_bazom_specifikacija_nacina_autentifikacije>,<navodnik>; 5. <veza_sa_bazom_gde_se_nalazi_SQL_Server>=<izvor_podataka>,<simbol_dodele_vrednosti>,<naziv_racunara>,<simbol_razdvajanja_naziva_racunara_i_instance>,<naziv_instance>,<simbol_kraja_naredbe>; 6.<veza_sa_bazom_naziv_baze_sa_kojom_se_povezuje>=<pocetni_katalog>,<simbol_dodele_vrednosti>,<naziv_baze>,<simbol_kraja_naredbe>; 7.<veza_sa_bazom_specifikacija_nacina_autentifikacije>=<SQL_Server_autentifikacija>,<simbol_dodele_vrednosti>,<mogucnost_autentifikacije>; 8.<tip_podatka>=<"string">; 9.<praznina>=<" ">; 10.<promenljiva>=<identifikator>; 11.<identifikator>={<slovo> }[[<broj>] 12. 13.<slovo>=<"a" "b" "c" ...>; 14.<broj>=<"1" "2" "3" ...>; 15.<simbol_dodele_vrednosti>=<"=">; 16.<simbol_kraja_naredbe>=<">; 17.<navodnik>=<">; 18.<izvor_podataka>=<"Data Source">; 19.<naziv_racunara>=<identifikator>; 20.<simbol_razdvajanja_naziva_racunara_i_instance>=<"\">; 21.<naziv_instance>=<identifikator>; 22.<pocetni_katalog>=<"Initial Catalog">;

	23.<naziv_baze>=<identifikator>; 24.<SQL_Server_autentifikacija>=" Integrated Security "; <mogucnost_autentifikacije>" True ";
--	--

6. Zaključak

U ovom seminarskom radu predstavili smo desktop aplikaciju za evidenciju prodaje torti, čija je namena da omogući korisnicima proces unosa, čuvanja, izmene i brisanja podataka o tortama, kao i pregled detaljnih informacija o kupovinama, filtriranje proizvoda po šifri, eksportovanje podataka i prikazivanje najvećeg prihoda. Kroz detaljan opis svrhe i funkcionalnosti aplikacije, istražili smo ključne delove koda i tehnologije koje su omogućile njeno stvaranje.

Tokom razmatranja različitih vrsta grešaka u programskom kodu, uključujući leksičke, sintaksne, semantičke, runtime i logičke greške, analizirali smo kako aplikacija reaguje na ove greške i kako ih ispraviti. Takođe smo istražili formalnu gramatiku koda, fokusirajući se na situacije gde su namerne greške ubačene u kod.

Aplikaciju za evidenciju prodaje torti možemo unaprediti dodavanjem slika torti, detaljnijih informacija o proizvodima, funkcionalnosti koje uključuju automatsko praćenje prodaje po danima i izračun zarade. Takođe, implementacija vizualizacija i grafikona za pregled prodaje i zarade može korisnicima olakšati analizu podataka.

7. Literatura

1. https://sr.wikipedia.org/sr-el/C_Sharp
2. http://www.tfzr.uns.ac.rs/Content/files/0/PP_sk2022_23_zbirka_zadataka.pdf
3. <http://147.91.177.109/Predmet/programski-prevodioci/obaveze-studenata-sk-202223>
4. https://sr.wikipedia.org/wiki/Microsoft_Visual_Studio