

# Prodavnica

Opis projektnog zadatka:

Projekat je osmišljen tako da simulira najjednostavniji vid online prodavnice. Korisniku su ponuđena dva combo boxa, koja međusobno zavise jedan od drugog, i jedan text box u koji se unosi broj komada željenog artikla. Jedan combo box sadrži listu svih proizvođača a drugi listu svih artikala.

The screenshot shows a web form titled "Form1" with the following elements:

- Three labels: "PROIZVODJAC", "PROIZVOD", and "KOM.".
- Two dropdown menus: the first is labeled "Izaberite proizvođača" and the second is labeled "Izaberite proizvod".
- A text input field for the quantity, labeled "KOM.", with a small "1" inside.
- A button labeled "Dodaj u korpu".
- A button labeled "Izaberite sledeći proizvod".
- A section titled "U korpi" containing a large empty rectangular box.
- A button labeled "Isprazni korpu".
- A counter displaying "0".
- A button labeled "PLACANJE".

Na slici takođe možemo videti 4 dugmeta. Dugme “dodaj u korpu” uzima podatke koje je korisnik uneo, prebacuje ih u list box i dozvoljava korisniku da bira još stvari koje će ubaciti u virtuelnu korpu. Kako je malopre napomenuto da combo boxevi zavise jedan od drugog mora postojati dugme “izaberite sledeći proizvod” koje služi da se korisnik vrati na početak odabira proizvoda bez bacanja istog u korpu.

Form1

PROIZVODJAC    PROIZVOD    KOM.

Izaberite proizvođača    Izaberite proizvod   

Dodaj u korpu    Izaberite sledeći proizvod

U korpi

FIFA sličice 50din. 1kom  
Pavlaka 40din. 3kom  
Torba 15000din. 5kom

PLACANJE

Isprazni korpu    75170

Možemo primetiti da labela ispod list boxa ispisuje sveukupnu cenu unetih proizvoda. Dugme “isprazni korpu” kako mi i samo ime govori, prazni datu korpu i cenu resetuje na 0.

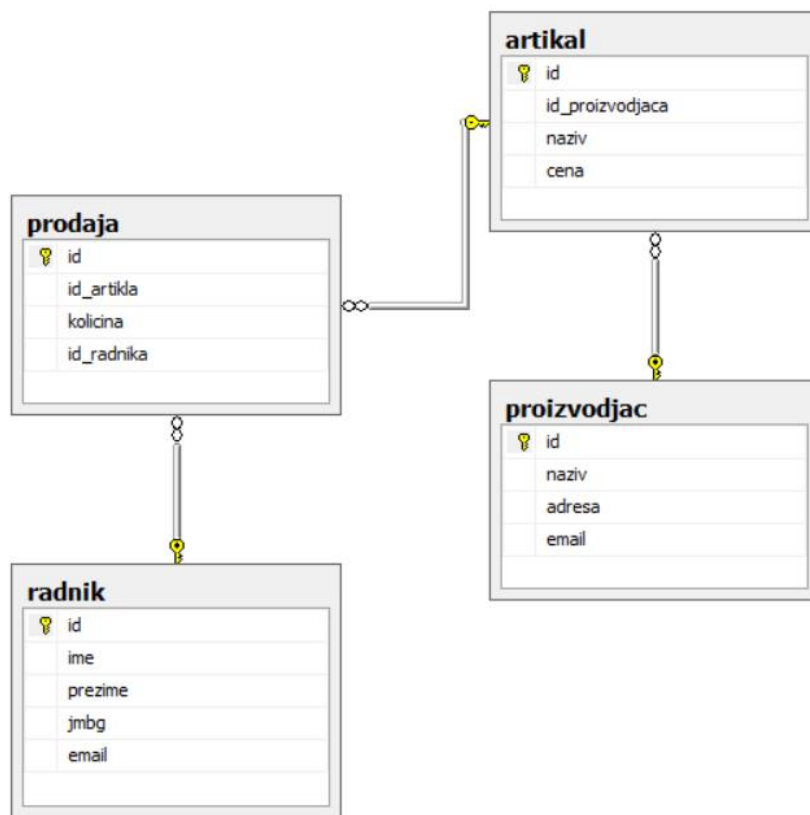
Nakon završenog procesa kupovine našem korisniku ostaje još da plati odabrane proizvode. Klikom an dugme “placanje”, dobija se poruka koja nas obaveštava da smo kupili određene proizvode i potrošili određenu količinu novca.

Hvala na ukazanom poverenju.  
Iznos za placanje je 75170

OK

Istovremeno se na SQL serveru vrši komanda koja upisuje podatke o kupljenim proizvodima u tabelu koja prati koliko se kojih proizvoda uzelo.

Model baze podataka:



Ova baza sadrži 4 tabele. Malopre pomenuta tabela koja vodi zapis o tome koliko je kojih artikala uzeto nosi naziv “prodaja”. Ona nam pored broja prodatog artikla govori koji radnik je odgovoran za određenu transakciju. Tabela “proizvodjac” nosi podatke o svim proizvođačima koji uvoze svoje article u prodavnicu dok ja tabela “artikal” povezana sa proizvođačima i nosi podatke o svim artiklima jednog određenog proizvođača. Tabela “radnik”, kao i tablea “prodaja” za korisnika nije bitna, a ona nosi podatke o radnicima koji su korisnicima omogućili usluge aplikacije. Svaka veza u datoj bazi je 1 prema više.

## Tekst kreiranja baze:

```
create database prodza
use prodza

create table artikal(
id int identity(1,1) primary key,
id_proizvodjaca int,
naziv nvarchar(50),
cena int,
foreign key (id_proizvodjaca) references proizvodjac(id)
)

create table proizvodjac(
id int identity(1,1) primary key,
naziv nvarchar(50),
adresa nvarchar(50),
email nvarchar(50)
)

drop table radnik(
id int identity(1,1) primary key,
ime nvarchar(50),
prezime nvarchar(50),
jmbg nvarchar(15),
email nvarchar(50)
)

create table prodaja(
id int identity(1,1) primary key,
id_artikla int,
kolicina int,
id_radnika int,
foreign key (id_artikla) references artikal(id),
foreign key (id_radnika) references radnik(id)
)

insert into proizvodjac values
('Frikom', 'Bakionicka 56', 'frikom@gmail.com'),
('Panini', 'Zrmanjska 420', 'panini@gmail.com'),
('Imlek', 'Zaplanjska 3', 'imlek@gmail.com'),
('Versace', 'Narodnih heroja 20', 'oridjidji@gmail.com'),
('Gucci', 'Vatromira Petrovica 76', 'dobromicuci@gmail.com'),
('Rolex', 'Decanska 8', 'rol@gmail.com')

insert into artikal values
(1, 'Sladoled u kornetu', 150),
(1, 'Sladoled na stapicu', 120),
(1, 'Smrznuti grasak', 200),
(2, 'FIFA slicice', 50),
(2, 'UEFA CL slicice', 70),
(3, 'Jogurt', 80),
(3, 'Pavlaka', 40),
```

```

(3, 'Mleko', 100),
(3, 'Kiselo mleko', 50),
(4, 'Parfem', 5000),
(4, 'Torba', 15000),
(4, 'Rob', 1000000),
(5, 'Rob', 5000000),
(6, 'Rob', 100000),
(5, 'Torba', 20000),
(5, 'Sat', 123456),
(6, 'Sat', 123456)

insert into radnik values
('Milos', 'Rakic', null, null),
('Igor', 'Janjic', null, null)

```

## Kod:

```

using
System;

using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using System.Collections;

namespace Prodavnica_Projekat
{
    public partial class Form1 : Form
    {
        string cs = @"Data source = DESKTOP-G6UU3BE; Initial catalog = prodza;
Integrated security = true";
        string skomanda;
        DataTable tabela, radnici;
        SqlConnection veza;
        List<Komid> korpica = new List<Komid>();
        int cena;
        public Form1()

```

```

{
    InitializeComponent();
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    comboBox2.Items.Clear();
    for (int i = 0; i < tabela.Rows.Count; i++)
    {
        if((string)comboBox1.SelectedItem ==
tabela.Rows[i][5].ToString())
            comboBox2.Items.Add(tabela.Rows[i][2].ToString());
    }
}

private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    comboBox1.Items.Clear();
    for (int i = 0; i < tabela.Rows.Count; i++)
    {
        if ((string)comboBox2.SelectedItem ==
tabela.Rows[i][2].ToString())
            comboBox1.Items.Add(tabela.Rows[i][5].ToString());
    }
}

void rewind()
{
    comboBox1.Items.Clear();
    comboBox2.Items.Clear();
    comboBox1.Text = "Izaberite proizvodjaca";
    comboBox2.Text = "Izaberite proizvod";
    for (int i = 0; i < tabela.Rows.Count; i++)
    {
        if (!comboBox1.Items.Contains(tabela.Rows[i][5].ToString()))
        {
            comboBox1.Items.Add(tabela.Rows[i][5].ToString());
        }
    }
    for (int i = 0; i < tabela.Rows.Count; i++)
    {
        if (!comboBox2.Items.Contains(tabela.Rows[i][2].ToString()))
        {

```

```

        comboBox2.Items.Add(tabela.Rows[i][2].ToString());
    }

}

private void button1_Click(object sender, EventArgs e)
{
    rewind();
}

private void dodaj_Click(object sender, EventArgs e)
{
    string aoikjwdoiajwdoi = "";
    for (int i = 0; i < tabela.Rows.Count; i++)
    {
        if(tabela.Rows[i][2].ToString() ==
(string)comboBox2.SelectedItem)
        {
            korpica.Add(new Komid(Convert.ToInt32(tabela.Rows[i][0]),
Convert.ToInt32(textBox1.Text)));
            aoikjwdoiajwdoi = comboBox2.SelectedItem + " " +
tabela.Rows[i][3].ToString() + "din." + " " + textBox1.Text + "kom";
            cena += Convert.ToInt32(tabela.Rows[i][3]) *
Convert.ToInt32(textBox1.Text);
            break;
        }
    }
    label5.Text = cena.ToString();
    listBox1.Items.Add(aoikjwdoiajwdoi);
    rewind();
    textBox1.Text = "";
}

private void plati_Click(object sender, EventArgs e)
{
    Random rnd = new Random();
    MessageBox.Show($"Hvala na ukazanom poverenju.\n Iznos za placanje je
{cena}");
    veza = new SqlConnection(cs);
    for (int i = 0; i < korpica.Count; i++)
    {

```

```

        SqlCommand komanda = new SqlCommand($"insert into prodaja
values({korpica[i].id}, {korpica[i].kom},{rnd.Next(1,3)})",veza);
        veza.Open();
        komanda.ExecuteNonQuery();
        veza.Close();
    }
    rewind();
    korpica.Clear();
    listBox1.Items.Clear();
    cena = 0;
    label5.Text = "0";

}

private void button2_Click(object sender, EventArgs e)
{
    korpica.Clear();
    cena = 0;
    label5.Text = "0";
}

private void Form1_Load(object sender, EventArgs e)
{
    veza = new SqlConnection(cs);
    skomanda = "select * from artikal join proizvođjac on proizvođjac.id
= artikal.id_proizvođjaca";
    SqlDataAdapter radnik = new SqlDataAdapter("select * from radnik",
veza);

    SqlDataAdapter adapter = new SqlDataAdapter(skomanda, veza);
    tabela = new DataTable();
    radnici = new DataTable();
    adapter.Fill(tabela);
    radnik.Fill(radnici);
    for (int i = 0; i < tabela.Rows.Count; i++)
    {
        if (!comboBox1.Items.Contains(tabela.Rows[i][5].ToString())) {
            comboBox1.Items.Add(tabela.Rows[i][5].ToString());
        }

    }

    for (int i = 0; i < tabela.Rows.Count; i++)
    {
        if (!comboBox2.Items.Contains(tabela.Rows[i][2].ToString()))

```



```
        {
            comboBox2.Items.Add(tabela.Rows[i][2].ToString());
        }

    }

}

public class Komid
{
    public int kom;
    public int id;
    public Komid(int koma, int ida)
    {
        this.kom = koma;
        this.id = ida;
    }
}

}
```