SOFTVERSKO INŽENJERSTVO

Naslov: Form aplikacija za prodaju mobilnih telefona

|  |  |
| --- | --- |
| Student: Luka Prćić  Indeks: 26120060  Smer: Informatika | Profesor: DR ŠIMON JANOŠ |

Sadržaj

[Specifikacija zahteva 3](#_Toc91449874)

[Prikaz C# okruženja 4](#_Toc91449875)

[Kod 6](#_Toc91449876)

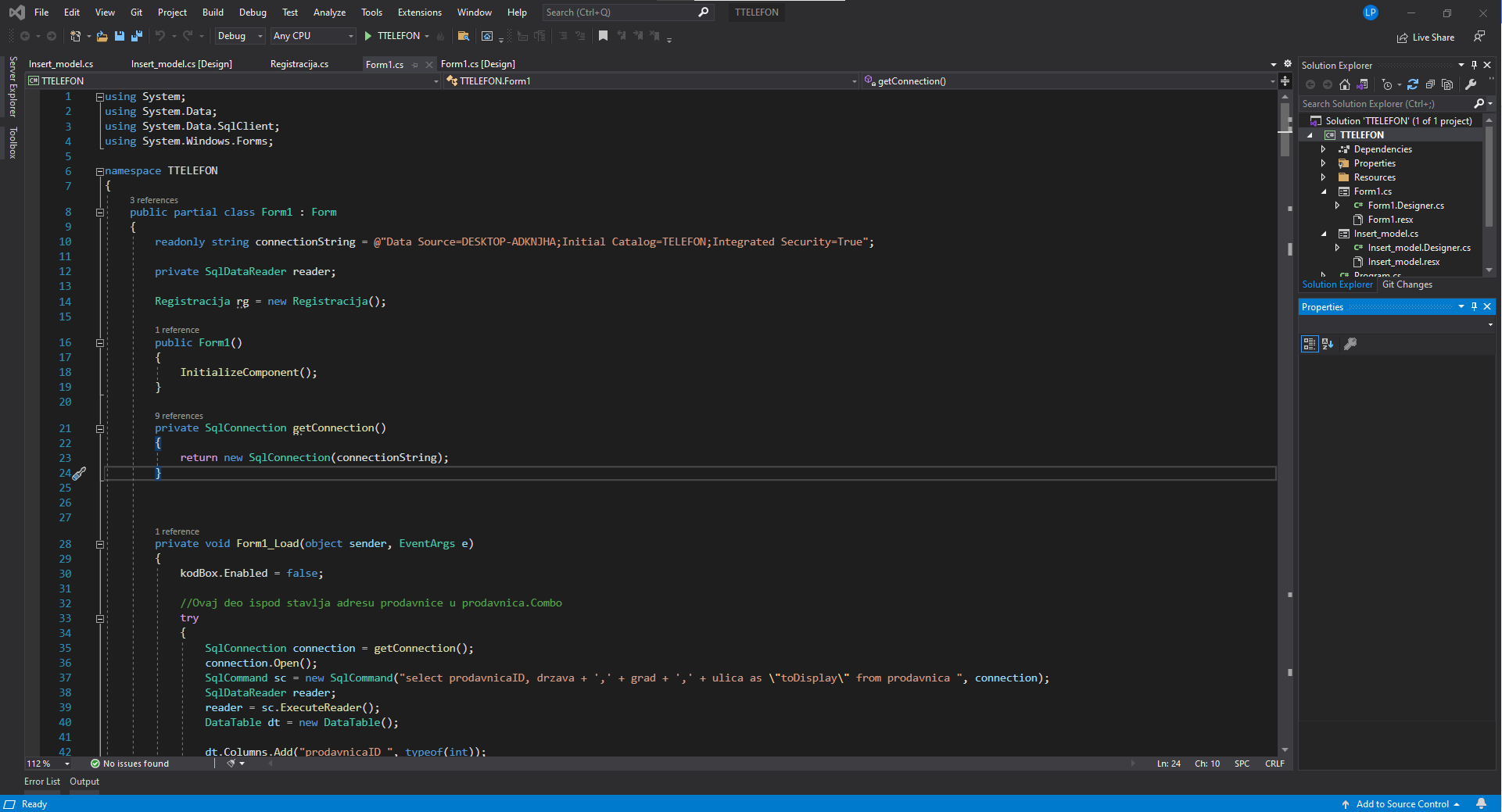
[Literatura 11](#_Toc91449877)

# Specifikacija zahteva

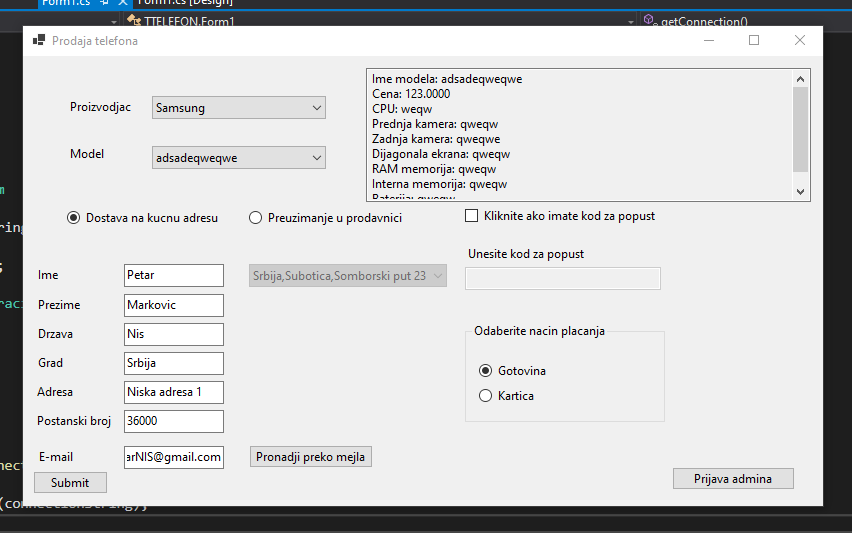
Program je namenjen za naručivanje telefona i ubacivanje novih proizvoda u bazu podataka. Za funkcionisanje ovog programa potrebna je odgovarajuća baza podataka u kojoj će se čuvati podaci o porudžbini i novi podaci koje unosi radnik prodavnice. Glavni deo programa služi kako bi mušterija videla koji su sve proizvodi na ponudi i da ima mogućnost odabira gde će naručiti svoj proizvod i da li će koristiti promo kod kako bi ostvarili neki popust.

Drugi deo aplikacije služi za insertovanje novog proizvoda u bazu podataka i ovom delu može šristupiti samo radnik čiji se podaci nalaze u bazi podataka.

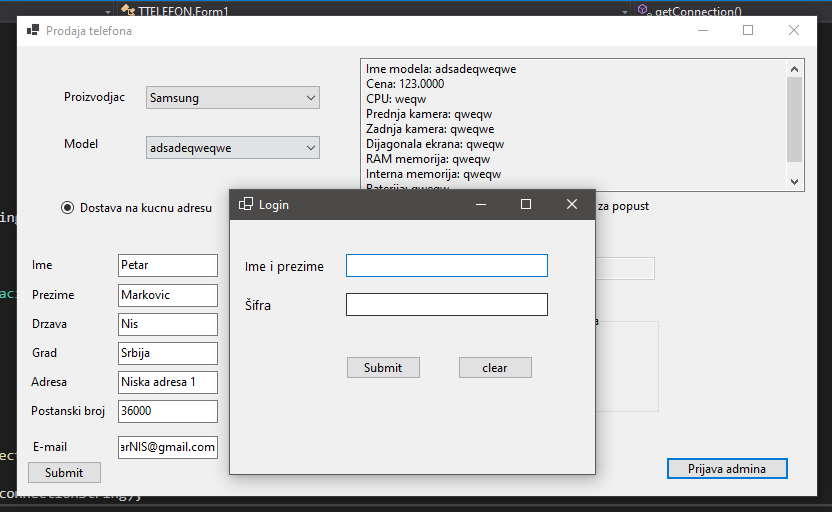
# Prikaz C# okruženja

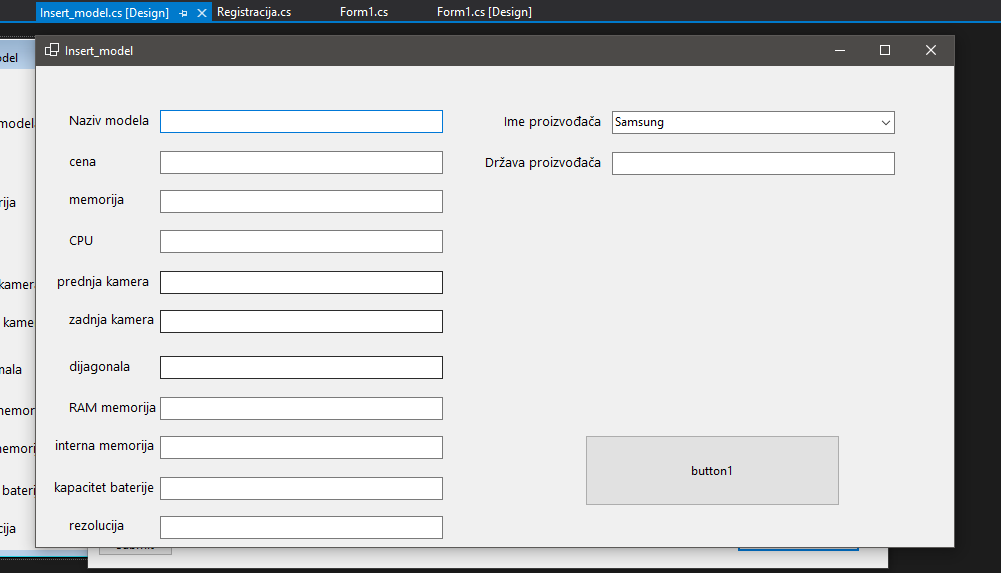


Izgled Visual Studio okruženja kada se pokrenje aplikacija

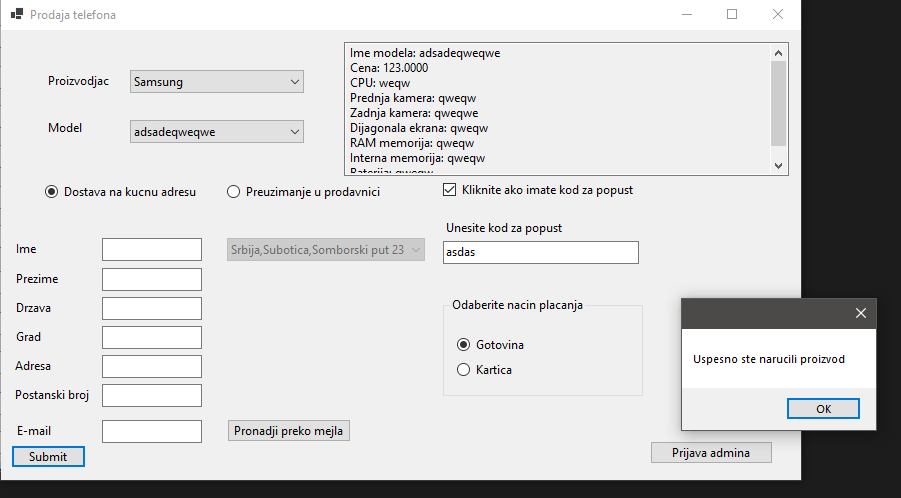


Izgled glavnog dela programa

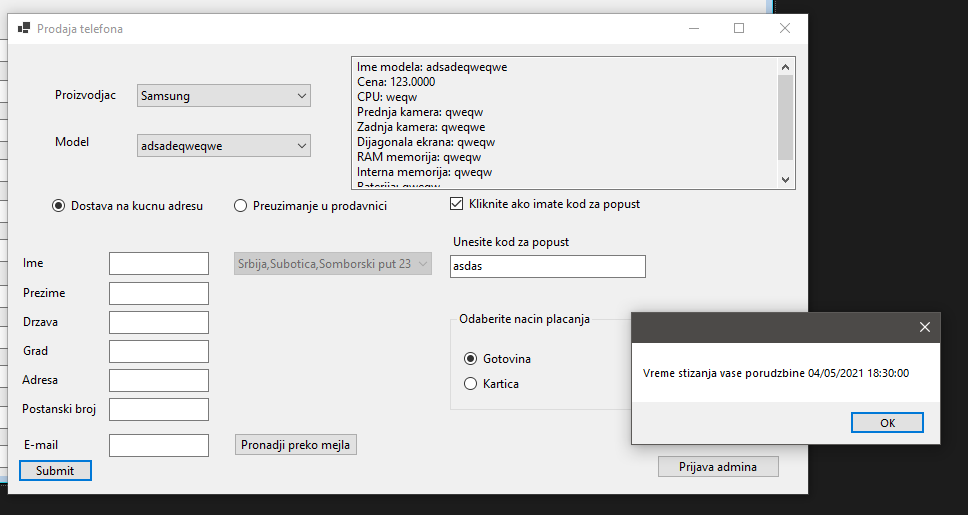
Izgled login stranice koja se aktivira klikom na dugme Prijava admina



Izgled okruženja radnika preko kojeg se unose novi modeli u ponudu



1Rezultat uspešne narudžbine prvi deo



2Rezultat uspešne narudžbine drugi deo

# Kod

**Form1 - glavni deo programa**

using System;

using System.Data;

using System.Data.SqlClient;

using System.Windows.Forms;

namespace TTELEFON

{

public partial class Form1 : Form

{

//Uspostavljamo konekciju sa bazom i programom

readonly string connectionString = @"Data Source=DESKTOP-ADKNJHA;Initial Catalog=TELEFON;Integrated Security=True";

private SqlDataReader reader;

Registracija rg = new Registracija();

public Form1()

{

InitializeComponent();

}

//Metoda koju cemo pozivati svaki put kad nam treba otvorena konekcija sa bazom

private SqlConnection getConnection()

{

return new SqlConnection(connectionString);

}

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

{

kodBox.Enabled = false;

//Ovaj deo ispod stavlja adresu prodavnice u prodavnica.Combo

try

{

SqlConnection connection = getConnection();

connection.Open();

SqlCommand sc = new SqlCommand("select prodavnicaID, drzava + ',' + grad + ',' + ulica as \"toDisplay\" from prodavnica ", connection);

SqlDataReader reader;

reader = sc.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("prodavnicaID ", typeof(int));

dt.Columns.Add("toDisplay ", typeof(string));

dt.Load(reader);

prodavnicaCombo.ValueMember = "prodavnicaID";

prodavnicaCombo.DisplayMember = "toDisplay";

prodavnicaCombo.DataSource = dt;

connection.Close();

}

catch (Exception err)

{

MessageBox.Show("Exception: " + err.ToString());

}

//Ovaj deo ispod uzima i stavlja podatke u comboBox1 tj stavlja naziv proizvodjaca telefona u comboBox

try

{

SqlConnection connection = getConnection();

connection.Open();

SqlCommand sc = new SqlCommand("select proizvodjac\_ime, proizvodjacID from proizvodjac", connection);

SqlDataReader reader;

reader = sc.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("proizvodjac\_ime ", typeof(string));

dt.Columns.Add("proizvodjacID ", typeof(int));

dt.Load(reader);

comboBox1.ValueMember = "proizvodjacID";

comboBox1.DisplayMember = "proizvodjac\_ime";

comboBox1.DataSource = dt;

connection.Close();

}

catch (Exception err)

{

MessageBox.Show("Exception: " + err.Message);

}

}

//Menja ime modela u comboBox2 na osnovu ime proizvodjaca u comboBox1

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

var proizvodjacId = (int)comboBox1.SelectedValue;

string query = "select model\_naziv from model where proizvodjac\_ID= " + proizvodjacId;

try

{

var connection = getConnection();

connection.Open();

SqlCommand sc = new SqlCommand(query, connection);

SqlDataReader reader;

reader = sc.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("model\_naziv ", typeof(string));

dt.Load(reader);

comboBox2.ValueMember = "model\_naziv";

comboBox2.DisplayMember = "model\_naziv";

comboBox2.DataSource = dt;

connection.Close();

}

catch (Exception err)

{

MessageBox.Show("Exception: " + err.Message);

}

}

//Menja mogucnost pristupa tekstualnim poljima ukoliko je kucaRadio checkovan pristup combo boxu prodavnice nece biti moguc i obrnuto

private void kucaRadio\_CheckedChanged(object sender, EventArgs e)

{

drzavaBox.Enabled = false;

gradBox.Enabled = false;

adresaBox.Enabled = false;

postanskiBox.Enabled = false;

prodavnicaCombo.Enabled = true;

if (prodavnicaRadio.Checked == true)

{

drzavaBox.Enabled = false;

gradBox.Enabled = false;

adresaBox.Enabled = false;

postanskiBox.Enabled = false;

prodavnicaCombo.Enabled = true;

}

else if (kucaRadio.Checked == true)

{

prodavnicaCombo.Enabled = false;

drzavaBox.Enabled = true;

gradBox.Enabled = true;

adresaBox.Enabled = true;

postanskiBox.Enabled = true;

}

}

//Omogucava pristup tekstualnom polju za unos koda i sprecava unos koda ukoliko nije cekirano check box

private void kod\_checkBox\_CheckedChanged(object sender, EventArgs e)

{

if (kod\_checkBox.Checked == true)

{

kodBox.Enabled = true;

}

else if (kod\_checkBox.Checked == false)

{

kodBox.Enabled = false;

}

}

//Dugme koje aktivira veliki deo programa. Aktivacijom dugmeta pravi se porudzbina i unose se svi relevantni podaci u sve odgovarajuce tabele

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

{

try

{

//PROMOKOD UZIMA IZ BAZE AKO JE VALIDAN TJ NALAZI SE U BAZI

SqlConnection promo\_kod\_konekcija = getConnection();

string promoKodQuery = ("SELECT promo\_kodID from promo\_kod where kod = @promo\_kodID and datum\_isteka\_koda > getdate()");

SqlCommand promoKodCommand = new SqlCommand(promoKodQuery, promo\_kod\_konekcija);

promoKodCommand.Parameters.AddWithValue("@promo\_kodID", kodBox.Text);

int? promoKodID = null;

promo\_kod\_konekcija.Open();

SqlDataReader reader = promoKodCommand.ExecuteReader();

//Ukoliko je reader nesto procitao odnosno promoKodQuery je pronasao podatak koji odgovara kveriju reader ce procitati i izvrsiti zadatu komandu

if (reader.HasRows)

{

reader.Read();

promoKodID = reader.GetInt32("promo\_kodID");

}

//Zatvaramo konekciju

promo\_kod\_konekcija.Close();

//TIP PLACANJA

//Dodeljuje vrednost varijabli tipPlacanja koja ce se kasnije prikazati u Message boxu za narudzbinu

//U zavisnosti koji tip placanja kupac zeli dodelice se taj ID varijabli tipPlacanja

int tipPlacanja = 0;

if (GotovinaRadio.Checked)

{

tipPlacanja = 1003;

}

else if (KarticaRadio.Checked)

{

tipPlacanja = 1002;

}

//OVAJ DEO INSERTUJE KUPCA

//Insertuje sve podatke koje kupac upise u program i unese ih u bazu

var connection = getConnection();

var command = new SqlCommand

{

Connection = connection,

CommandText = "INSERT INTO kupac " +

"(ime, prezime,Email) output INSERTED.kupacID VALUES " +

"(@ime, @prezime,@mail)"

};

command.Parameters.Clear();

command.Parameters.AddWithValue("@ime", imeBox.Text);

command.Parameters.AddWithValue("@prezime", prezimeBox.Text);

command.Parameters.AddWithValue("@mail", mailBox.Text);

connection.Open();

int kupacID = (int)command.ExecuteScalar();

//Isto kao i za kupca samo sto sacuva podatke o adresi kupca i unese ih u adresa tabelu

command.CommandText = "INSERT INTO adresa " +

"(drzava,grad,ulica,zip,kupacID) output INSERTED.adresaID VALUES " +

"(@drzava,@grad,@ulica,@zip,@kupacID)";

command.Parameters.Clear();

command.Parameters.AddWithValue("@drzava", drzavaBox.Text);

command.Parameters.AddWithValue("@grad", gradBox.Text);

command.Parameters.AddWithValue("@ulica", adresaBox.Text);

command.Parameters.AddWithValue("@zip", postanskiBox.Text);

command.Parameters.AddWithValue("@kupacID", kupacID);

int result = command.ExecuteNonQuery();

int adresaID = (int)command.ExecuteScalar();

//Metoda isprazni() ce isprazniti sva tekstualna polja nakon izvrsenja komande.

isprazni();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//OVAJ DEO INSERTUJE PORUDZBINU U TABELU PORUDZBINA

var porudzbina\_command = new SqlCommand

{

Connection = connection,

CommandText = "INSERT INTO porudzbina " +

"(kupac\_id, tip\_placanjaID,promo\_kodID,datum\_narucivanja) output INSERTED.porudzbinaID VALUES " +

"(@kupacID,@tip\_placanjaID, @promo\_kodID,getdate())"

};

porudzbina\_command.Parameters.Clear();

porudzbina\_command.Parameters.AddWithValue("@kupacID", kupacID);

porudzbina\_command.Parameters.AddWithValue("@tip\_placanjaID", tipPlacanja);

//Proverava da li je promoKodId jedank sa null ukoliko jeste dobice vrednost null sto znaci da kupac nije iskoristio promo kod

if (promoKodID != null)

{

porudzbina\_command.Parameters.AddWithValue("@promo\_kodID", promoKodID);

}

else

{

porudzbina\_command.Parameters.AddWithValue("@promo\_kodID", DBNull.Value);

}

int porudzbinaID = (int)porudzbina\_command.ExecuteScalar();

//PORUDZBINA\_PRODAVNICA

//Insertuje u pordzbina\_prodavnica ako je chekovan kucaRadio

//U zavisnosti od toga da li je kuppac odabrao dostavu na kucnu adresu ili dostavu u prodavnici dva razlicita kverija ce se izvrsiti

if (kucaRadio.Checked)

{

porudzbina\_command.CommandText = "INSERT INTO porudzbina\_adresa " +

"(adresaID,porudzbinaID,vreme\_isporuke) VALUES " +

"(@adresaID,@porudzbinaID,getdate()+20 )";

porudzbina\_command.Parameters.Clear();

porudzbina\_command.Parameters.AddWithValue("@adresaID", adresaID);

porudzbina\_command.Parameters.AddWithValue("@porudzbinaID", porudzbinaID);

int porudzbina\_result = porudzbina\_command.ExecuteNonQuery();

}

//PORUDZBINA\_PRODAVNICA

//Insertuje u pordzbina\_adresa ako je chekovan prodavnicaRadio

else if (prodavnicaRadio.Checked)

{

porudzbina\_command.CommandText = "INSERT INTO porudzbina\_prodavnica " +

"(datum\_stizanja\_proizvoda,prodavnicaID,porudzbinaID) VALUES " +

"(getdate()+20,@prodavnicaID,@porudzbinaID )";

porudzbina\_command.Parameters.Clear();

//ovde parametri

int prodavnicaID = (int)prodavnicaCombo.SelectedValue;

porudzbina\_command.Parameters.AddWithValue("@porudzbinaID", porudzbinaID);

porudzbina\_command.Parameters.AddWithValue("@prodavnicaID", prodavnicaID);

int porudzbina\_result = porudzbina\_command.ExecuteNonQuery();

}

//STAVKA

var stavka\_command = new SqlCommand

{

Connection = connection,

CommandText = "INSERT INTO stavka " +

"(br\_komada,proizvodjacID,model\_naziv,porudzbinaID) VALUES " +

"(1, @proizvodjacID,@model\_naziv,@porudzbinaID)"

};

stavka\_command.Parameters.Clear();

var proizvodjacID = comboBox1.SelectedValue;

stavka\_command.Parameters.AddWithValue("@proizvodjacID", proizvodjacID);

stavka\_command.Parameters.AddWithValue("@porudzbinaID", porudzbinaID);

stavka\_command.Parameters.AddWithValue("@model\_naziv", comboBox2.Text);

int stavka\_result = stavka\_command.ExecuteNonQuery();

connection.Close();

if (result > 0)

{

MessageBox.Show("Uspesno ste narucili proizvod");

//Ponovo u zavisnosti od toga da li kupac zeli dostavu na kucnu adresu ili u prodavnici razlicito vreme ce mu iskociti na kraju u vidu MessageBoxa

if (kucaRadio.Checked)

{

SqlConnection poruka\_konekcija = getConnection();

string poruka\_kveri = ("SELECT vreme\_isporuke from porudzbina\_adresa");

SqlCommand poruka\_command = new SqlCommand(poruka\_kveri, poruka\_konekcija);

poruka\_konekcija.Open();

SqlDataReader poruka\_reader = poruka\_command.ExecuteReader();

if (poruka\_reader.HasRows)

{

poruka\_reader.Read();

var vreme\_isporuke = poruka\_reader.GetDateTime("vreme\_isporuke").ToString();

MessageBox.Show(string.Format("Vreme stizanja vase porudzbine {0}", vreme\_isporuke));

}

poruka\_konekcija.Close();

}

else if (prodavnicaRadio.Checked)

{

SqlConnection poruka\_konekcija = getConnection();

string poruka\_kveri = ("SELECT datum\_stizanja\_proizvoda from porudzbina\_prodavnica");

SqlCommand poruka\_command = new SqlCommand(poruka\_kveri, poruka\_konekcija);

poruka\_konekcija.Open();

SqlDataReader poruka\_reader = poruka\_command.ExecuteReader();

if (poruka\_reader.HasRows)

{

poruka\_reader.Read();

var vreme\_isporuke = poruka\_reader.GetDateTime("datum\_stizanja\_proizvoda ").ToString();

MessageBox.Show(string.Format("Vreme stizanja vase porudzbine {0}", vreme\_isporuke));

}

poruka\_konekcija.Close();

}

}

//Poruka koja ce iskociti ako nesto nije ispunjeno na odgovarajuci nacin

else

{

MessageBox.Show("Doslo je do greske niste narucili proizvod");

}

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message);

}

}

//Svaki put kada kupac izabere drugi model u text boxu ce se ispisati svi podaci o tom modelu i na osnovu tih podataka kupac moze odluciti sta zeli kupiti

private void comboBox2\_SelectedIndexChanged(object sender, EventArgs e)

{

var connection = getConnection();

var proizvodjacId = (int)comboBox1.SelectedValue;

var model\_naziv = comboBox2.SelectedValue;

var command = new SqlCommand

{

Connection = connection,

CommandText = "SELECT \* from model where proizvodjac\_ID = " + proizvodjacId + "and model\_naziv = '" + model\_naziv + "'"

};

try

{

connection.Open();

int result = command.ExecuteNonQuery();

reader = command.ExecuteReader(CommandBehavior.SingleRow);

if (reader.Read())

{

string naziv = (string)reader["model\_naziv"].ToString();

string cena = (string)reader["cena"].ToString();

string CPU = (string)reader["CPU"].ToString();

string pkamera = (string)reader["prednja\_kamera"].ToString();

string zkamera = (string)reader["zadnja\_kamera"].ToString();

string dekrana = (string)reader["dijagonala\_ekrana"].ToString();

string rmemorija = (string)reader["RAM\_memorija"].ToString();

string imemorija = (string)reader["interna\_memorija"].ToString();

string baterija = (string)reader["kapacitet\_baterije"].ToString();

podaciTextBox.Text = "Ime modela: " + naziv + " \r\nCena: " + cena + "\r\nCPU: " + CPU + "\r\nPrednja kamera: " + pkamera + "\r\nZadnja kamera: " + zkamera + "\r\nDijagonala ekrana: " + dekrana

+ "\r\nRAM memorija: " + rmemorija + " \r\nInterna memorija: " + imemorija + "\r\nBaterija: " + baterija;

}

else

{

MessageBox.Show("Nema zeljenog modela");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

//Klikom na dugme pronadji ispisace se svi podaci o musteriji ukoliko je ta musterija narucila nesto preko aplikacije koristeci isti mejl

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

{

var connection = getConnection();

try

{

var command = new SqlCommand

{

Connection = connection,

CommandText = "SELECT \* FROM kupac WHERE Email = @Email"

};

command.Parameters.Clear();

command.Parameters.AddWithValue("@Email", mailBox.Text);

var customerID = 0;

var success = false;

connection.Open();

var reader = command.ExecuteReader();

if (reader.Read())

{

success = true;

imeBox.Text = reader["ime"].ToString();

prezimeBox.Text = reader["prezime"].ToString();

customerID = Convert.ToInt32(reader["kupacID"].ToString());

}

reader.Close();

if (success)

{

command.Parameters.Clear();

command.CommandText = "SELECT \* FROM adresa WHERE kupacID = @kupacID";

command.Parameters.AddWithValue("@kupacID", customerID);

reader = command.ExecuteReader();

success = false;

if (reader.Read())

{

success = true;

drzavaBox.Text = reader["grad"].ToString();

gradBox.Text = reader["drzava"].ToString(); //savetovao bi ti da koristis reader.getString Int Double sta god ovde funkciju i

// punis sa indexom po redu koji ti ocitao

adresaBox.Text = reader["ulica"].ToString();

postanskiBox.Text = reader["zip"].ToString();

}

if (success)

{

MessageBox.Show("Success");

}

else

{

MessageBox.Show(" address select Error");

}

}

else

{

MessageBox.Show("E-mail entered doesn't exist");

}

}

catch (SqlException ex)

{

MessageBox.Show(ex.Message);

}

connection.Close();

}

//Telo metode isprazni kojom se brisu svi uneti podaci

private void isprazni()

{

imeBox.Clear();

prezimeBox.Clear();

mailBox.Clear();

drzavaBox.Clear();

adresaBox.Clear();

postanskiBox.Clear();

gradBox.Clear();

}

//Klikom na dugme Prijava admina prikazace se prozor namenjen radniku ili adminu koji ima mogucnost unosa novih modela u prodavnicu

private void Registracija\_btn\_Click(object sender, EventArgs e)

{

rg.Show();

}

}

}

**Registracija- login radnika**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace TTELEFON

{

public partial class Registracija : Form

{

public Registracija()

{

InitializeComponent();

}

readonly string connectionString = @"Data Source=DESKTOP-ADKNJHA;Initial Catalog=TELEFON;Integrated Security=True";

private SqlConnection getConnection()

{

return new SqlConnection(connectionString);

}

//Unose se podaci u dva tekstualna polja i zatim se proveravaju ti podaci da li se nalaze u bazi. Ukoliko se nalaze u bazi podataka prikazace se sledeci insert prozor

private void Registracija\_btn\_Click(object sender, EventArgs e)

{

string ime\_prezime, sifra;

SqlConnection connection = getConnection();

try

{

string querry = ("SELECT \* from radnik where ime\_prezime = '" + ime\_radnika\_box.Text.Trim() + "' and sifra = '" + sifra\_radnika\_box.Text.Trim()+"'");

SqlDataAdapter sda = new SqlDataAdapter(querry,connection);

DataTable dtable = new DataTable();

sda.Fill(dtable);

if (dtable.Rows.Count > 0)

{

ime\_prezime = ime\_radnika\_box.Text;

sifra = sifra\_radnika\_box.Text;

Insert\_model im = new Insert\_model();

im.Show();

this.Hide();

}

else

{

MessageBox.Show("Invalid login details","Error",MessageBoxButtons.OK,MessageBoxIcon.Error);

ime\_radnika\_box.Clear();

sifra\_radnika\_box.Clear();

ime\_radnika\_box.Focus();

}

}

catch (SqlException ex)

{

MessageBox.Show(ex.ToString());

}

finally

{

connection.Close();

}

}

//Ukoliko neko zeli na lak nacin izbrisati podatke koje su uneli klknuce dugme clear

private void butto\_clear\_Click(object sender, EventArgs e)

{

ime\_radnika\_box.Clear();

sifra\_radnika\_box.Clear();

}

}

}

**Insert\_model- forma za unosenje modela**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

namespace TTELEFON

{

public partial class Insert\_model : Form

{

private SqlDataReader reader;

public Insert\_model()

{

InitializeComponent();

}

readonly string connectionString = @"Data Source=DESKTOP-ADKNJHA;Initial Catalog=TELEFON;Integrated Security=True";

private SqlConnection getConnection()

{

return new SqlConnection(connectionString);

}

//Prikazace nam samo formu za insertovanje ukoliko smo uneli odgovarajuce podatke

private void Insert\_model\_Load(object sender, EventArgs e)

{

try

{

SqlConnection connection = getConnection();

connection.Open();

SqlCommand sc = new SqlCommand("select proizvodjac\_ime, proizvodjacID from proizvodjac", connection);

SqlDataReader reader;

reader = sc.ExecuteReader();

DataTable dt = new DataTable();

dt.Columns.Add("proizvodjac\_ime ", typeof(string));

dt.Columns.Add("proizvodjacID ", typeof(int));

dt.Load(reader);

comboBox\_proizvodjac.ValueMember = "proizvodjacID";

comboBox\_proizvodjac.DisplayMember = "proizvodjac\_ime";

comboBox\_proizvodjac.DataSource = dt;

connection.Close();

}

catch (Exception err)

{

MessageBox.Show("Exception: " + err.Message);

}

}

//Klikom da dugme jedan unose se podaci u tabelu model i zatim se taj isti model moze izabrati za kupovinu u Form1 delu programa

private void button1\_Click(object sender, EventArgs e)

{

var connection = getConnection();

var command = new SqlCommand

{

Connection = connection,

CommandText = "INSERT INTO model " +

"(model\_naziv,cena,memorija,CPU,prednja\_kamera,zadnja\_kamera,dijagonala\_ekrana,RAM\_memorija,interna\_memorija,kapacitet\_baterije,rezolucija,proizvodjac\_ID) values "

+ "(@model\_naziv,@cena,@memorija,@CPU,@prednja\_kamera,@zadnja\_kamera,@dijagonala\_ekrana,@RAM\_memorija,@interna\_memorija,@kapacitet\_baterije,@rezolucija,@proizvodjac\_ID)"

};

var proizvodjacID = (int)comboBox\_proizvodjac.SelectedValue;

command.Parameters.Clear();

command.Parameters.AddWithValue("@model\_naziv", naziv\_mod\_txtBox.Text);

command.Parameters.AddWithValue("@cena", cena\_txtBox.Text);

command.Parameters.AddWithValue("@memorija", memorija\_txtBox.Text);

command.Parameters.AddWithValue("@CPU", CPU\_txtBox.Text);

command.Parameters.AddWithValue("@prednja\_kamera", prednja\_kamera\_txtBox.Text);

command.Parameters.AddWithValue("@zadnja\_kamera", zadnja\_kamera\_txtBox.Text);

command.Parameters.AddWithValue("@dijagonala\_ekrana", dijagonala\_txtBox.Text);

command.Parameters.AddWithValue("@RAM\_memorija", RAM\_txtBox.Text);

command.Parameters.AddWithValue("@interna\_memorija", interna\_txtBox.Text);

command.Parameters.AddWithValue("@kapacitet\_baterije", kapacitet\_txtBox.Text);

command.Parameters.AddWithValue("@rezolucija", rezolucija\_txtBox.Text);

command.Parameters.AddWithValue("@proizvodjac\_ID", proizvodjacID);

connection.Open();

int result = command.ExecuteNonQuery();

if (result > 0)

{

MessageBox.Show("Uspesno ste uneli novi proizvod");

}

else

{

MessageBox.Show("Doslo je do greske niste uneli proizvod");

}

//Brisu se uneti podaci iz tekstualnih polja na kraju izvrsavanja programa

proizvodjac\_drzava\_txtBox.Clear();

connection.Close();

naziv\_mod\_txtBox.Clear();

cena\_txtBox.Clear();

memorija\_txtBox.Clear();

CPU\_txtBox.Clear();

prednja\_kamera\_txtBox.Clear();

zadnja\_kamera\_txtBox.Clear();

dijagonala\_txtBox.Clear();

RAM\_txtBox.Clear();

interna\_txtBox.Clear();

kapacitet\_txtBox.Clear();

rezolucija\_txtBox.Clear();

}

}

}

# Literatura

Youtube tutorials:

* <https://www.w3schools.com/cs/index.php>