**Liceul Teoretic “Alexandru Ioan Cuza” Corabia**

***LUCRARE PENTRU OBȚINEREA ATESTATULUI DE COMPETENȚE PROFESIONALE***

**Agenție imobiliară**

**Îndrumător: prof. Dașoveanu Laurențiu**

**Elev: Costea Corina**

**CUPRINS**

**Motivatia alegerii temei……………………….... 3**

**Cerinte hardware si software…………………… 3**

**Introducere………………………................…….. 4**

**Prezentarea aplicaţiei………………................…. 6**

**Bibliografie……………………………...............…**

MOTIVATIA ALEGERII TEMEI:

**Cerinte hardware si software**

**Hardware:**

Minimum:

-1.6 GHz processor;

-1024 MB RAM;

-5400 RPM hard-disk drive;

-DirectX 9;

**Software:**

-Microsoft Visual C# 2019

**Utilitatea aplicatiei:**

Aplicaţia poate fi utilizata de orice persoana care doreste sa-si usureze munca cu clientii, indiferent daca a mai folosit un astfel de program sau nu.

Din dorința de a cerceta cât mai multe domenii, mi-am ales această temă deoarece în ziua de azi domeniul imobiliarelor este foarte bine clasat.

Combinarea acestui domeniu cu cel al informaticii, de altfel prezent peste tot, se datorează faptului că ambele caștigă teren pe zi ce trece.

Am ales limbajul de programare C# deoarece are o interfață accesibilă și atragătoare și în același timp, oferă o multitudine de variante de creare.

**Structura aplicatiei**

Aplicatia este formata din 2 baze de date si o interfata.

Bazele de date sunt realizate cu ajutorul aplicatiei Microsoft Office Access 2007.

**INTRODUCERE**

„Agentie imobiliara” este o aplicatie care poate fi folosita pentru a prezenta imobile, dupa preferintele solicitantului, in functie de locatie, pret, suprafata etc. Scopul acestei aplicatii este de a usura cautarile unei locuinte de catre persoanele care nu au foarte mult timp la dispozitie. Este foarte usor de folosit si nu necesita cunostinte de informatica.

Clientii isi pot forma o idee despre locuintele preferate, despre preturile lor in raport cu numarul camerelor, cu aspectul, dar si cu amplasarea, criteriul pret-calitate fiind cel mai important.

Aplicatia „Agentie imobiliara” ii ofera clientului oportunitatea de a deveni vanzator, putand adauga noi proprietati alaturi de specificatiile acestora. Aplicatia ofera in acelasi timp si posibilitatea de a modifica sau sterge anumite detalii.

Cum cumpararea unei locuinte nu este numai o investitie importanta, ci si pe termen lung, trebuie luata o decizie chibzuita. Din acest motiv, trebuie sa fim foarte bine informati, pentru a fi siguri ca este o achizitie sigura si eficienta.

Daca apelam la serviciile unei agentii imobiliare, ne putem bucura de cateva avantaje:

1. **Sfaturi din partea unor profesionisti**

Pe piata imobiliara exista o multime de oferte, astfel incat poate dura ceva timp sa cauti, sa gasesti si sa vizitezi mai multe proprietati pana iti gasesti „casa ta”. Cea mai buna solutie poate fi sa apelezi la o agentie care lucreaza cu consulatanti profesionisti care cunosc piata, analizeaza dorintele tale si cauta cele mai bune optiuni, astfel aceasta aplicatie dispune de sectiunea „CONTACT”, unde puteti cere ajutor.

1. **Economie de timp si chiar bani**

Primind consultanta din partea unui profesionist economisesti timp si ai posibilitatea sa gasesti o casa pe care s-ar putea sa nu o gasesti prin alte mijloace.

**C#** este un [limbaj de programare](https://ro.wikipedia.org/wiki/Limbaj_de_programare) orientat-obiect conceput de [Microsoft](https://ro.wikipedia.org/wiki/Microsoft) la sfârșitul anilor 90. A fost conceput ca un concurent pentru [limbajul Java](https://ro.wikipedia.org/wiki/Limbajul_Java" \o "Limbajul Java). Ca și acesta, C# este un derivat al limbajului de programare [C++](https://ro.wikipedia.org/wiki/C%2B%2B).

Creat ca instrument de dezvoltare pentru arhitectura .NET, limbajul ofera o modalitate facila si eficienta de a scrie programe pentru sistemul Windows, internet, componente software etc.

Limbajul C# contine mai multe facilitati novatoare, dintre care cele mai importante se refera la suportul incorporat pentru componente software. C# dispune de facilitati care implementeaza direct elementele care alcatuiesc componentele software, cum ar fi proprietatile, metodele si evenimentele. Poate cea mai importanta facilitate de care dispune C# este posibilitatea de a lucra intr-un mediu cu limbaj mixt.

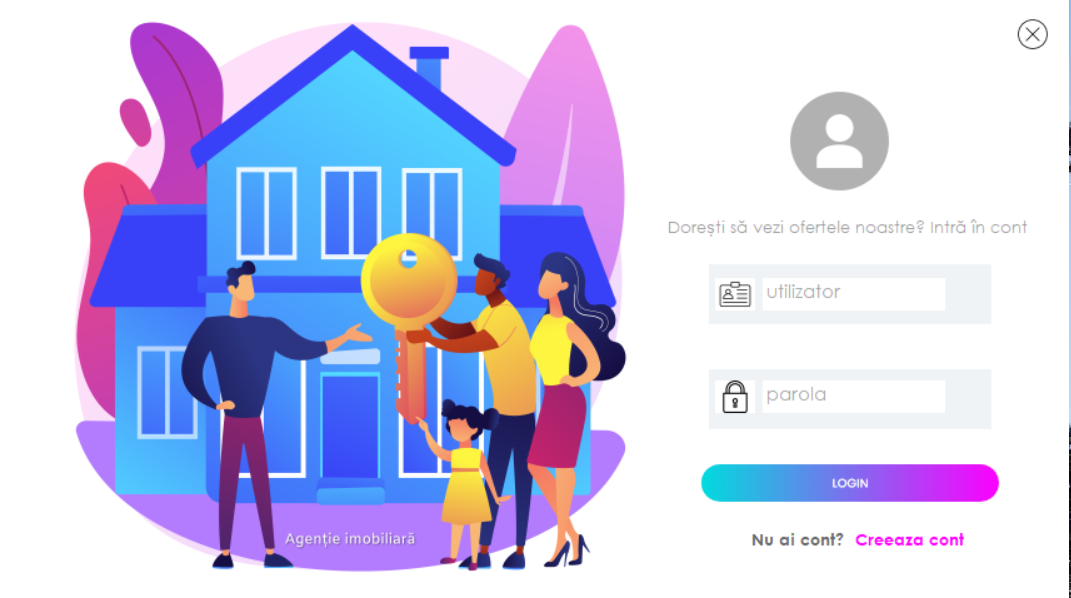
**PREZENTAREA APLICATIEI**

**Structura aplicatiei**

Aplicatia este formata dintr-o baza de date si o interfata.

Baza de date este realizata cu ajutorul aplicatiei Microsoft Office Access 2007.

Dupa ce rulam programul apare aceasta pagina:



Pentru a accesa ofertele si a ajunge la pagina principala, trebuie completate campurile **„utilizator”** si **„parola”**. Daca acestea nu corespund cu cele din baza de date, se va afisa mesajul „Utilizator sau parola incorecte”. In acest caz, utiliatorul are posibilitatea de a-si creea propriul cont accesand butonul **„Creeaza cont”**, care are urmatorul cod:

private void label3\_Click(object sender, EventArgs e)

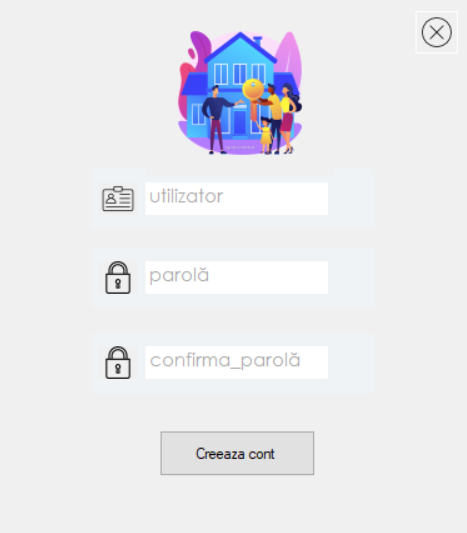
{

register reg = new register();

reg.Show();

}

Pagina de creeare cont este urmatoarea si trebuie completata dupa preferinte:



Parola introdusa trebuie sa fie cuprinsa intre 5 si 10 caractere, daca nu corespunde, se va afisa unul dintre mesajele:

* „Parola este prea scurta”
* „Parola este prea lunga”

Pentru a creea contul trebuie apasat butonul **„Creeaza cont”** cu urmatorul cod:

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

{

con.Open();

string userpass = "SELECT \* FROM cont where utilizator = '"+userregister.Text+"' and parola = '"+passregister.Text+"'";

OleDbCommand cmd = new OleDbCommand(userpass, con);

OleDbDataAdapter adapter = new OleDbDataAdapter(cmd);

DataTable ds = new DataTable();

adapter.Fill(ds);

if (ds.Rows.Count == 1)

{

MessageBox.Show("Contul exista deja.");

}

else

{

if (string.Equals(passregister.Text, pass2register.Text))

{

//parola

if (passregister.Text.Length <= 4)

{

MessageBox.Show("Parola este prea scurta.");

}

else if (passregister.Text.Length >= 10)

{

MessageBox.Show("Parola este prea lunga.");

}

else

{

conn.Open();

OleDbCommand cmdd = conn.CreateCommand();

cmdd.CommandType = CommandType.Text;

cmdd.CommandText = "insert into cont values('" + userregister.Text + "', '" + passregister.Text + "')";

cmdd.ExecuteNonQuery();

conn.Close();

MessageBox.Show("Contul a fost creat cu succes. Acum te poti conecta");

}

}

else

{

MessageBox.Show("Parolele nu se potrives.");

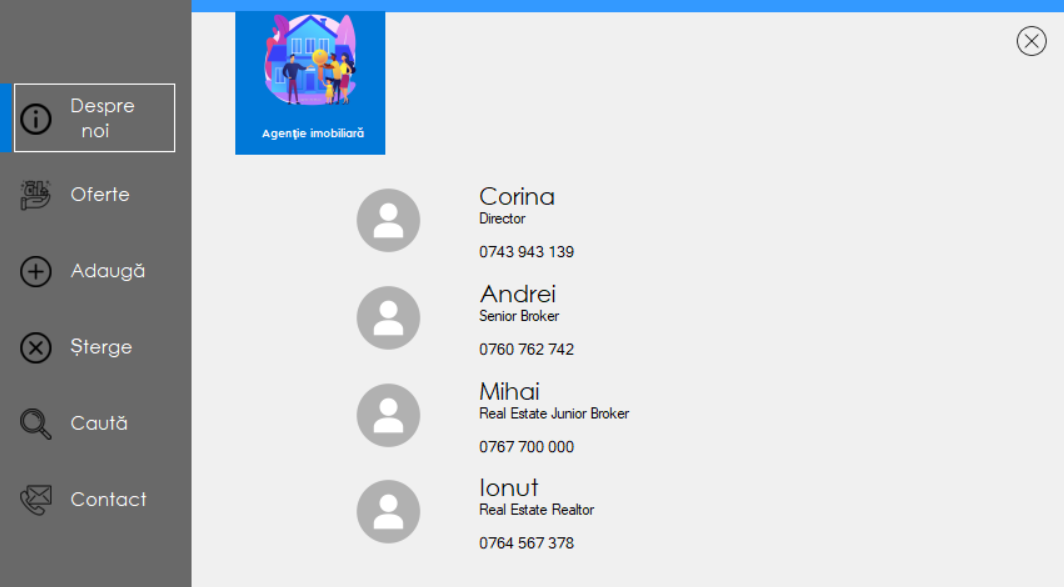
}

}

con.Close();

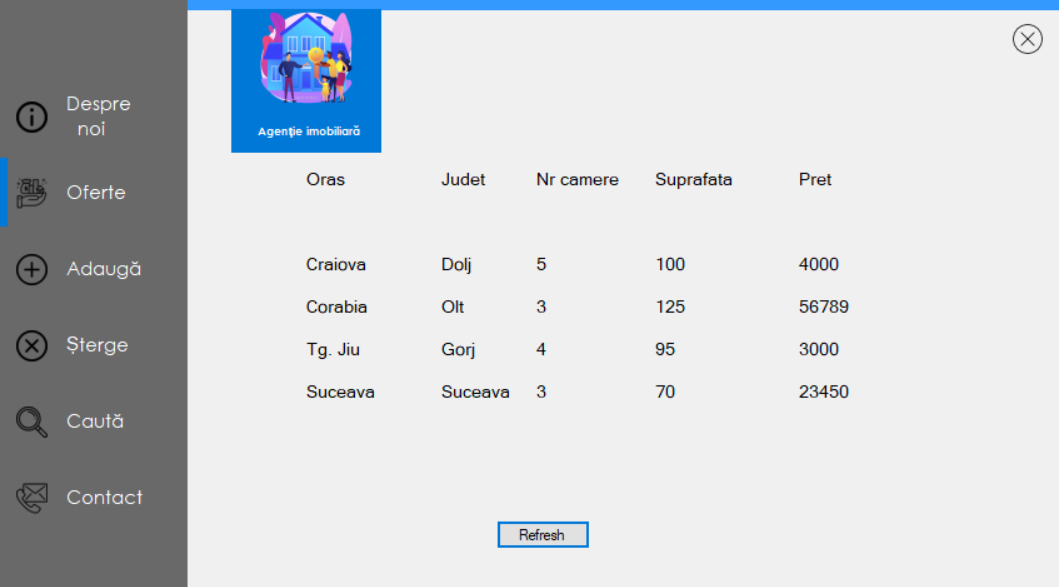
}

Dupa ce contul a fost creeat, putem accesa pagina principala, care contine 6 sectiuni, acestea fiind: **„Despre noi”, „Oferte”, „Adauga”, „Sterge”, „Cauta” si „Contact”.**



Pagina **„Despre noi”** contine detalii despre angajatii nostri.

In pagina **„Oferte”** gasim locuintele alaturi de specificatiile acestora.



Pentru a vizualiza ofertele trebuie sa apasam butonul **“Refresh”** cu urmatorul cod:

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

{

con.Open();

string cautare = "SELECT \* FROM Table1";

//OleDbCommand cmd = con.CreateCommand();

OleDbCommand cmd = new OleDbCommand(cautare, con);

// DataTable dt = new DataTable();

OleDbDataAdapter adapter = new OleDbDataAdapter(cmd);

DataSet ds = new DataSet();

adapter.Fill(ds, "Table1");

label1.Text = "Oras" + "\n" + "\n" + "\n";

label2.Text = "Judet" + "\n" + "\n" + "\n";

label3.Text = "Nr camere" + "\n" + "\n" + "\n";

label4.Text = "Suprafata" + "\n" + "\n" + "\n";

label5.Text = "Pret" + "\n" + "\n" + "\n";

foreach (DataRow r in ds.Tables["Table1"].Rows)

{

label1.Text = label1.Text + "\n" + r["Oras"] + "\n";

label2.Text = label2.Text + "\n" + r["Judet"] + "\n";

label3.Text = label3.Text + "\n" + r["NrCamere"] + "\n";

label4.Text = label4.Text + "\n" + r["Suprafata"] + "\n";

label5.Text = label5.Text + "\n" + r["Pret"] + "\n";

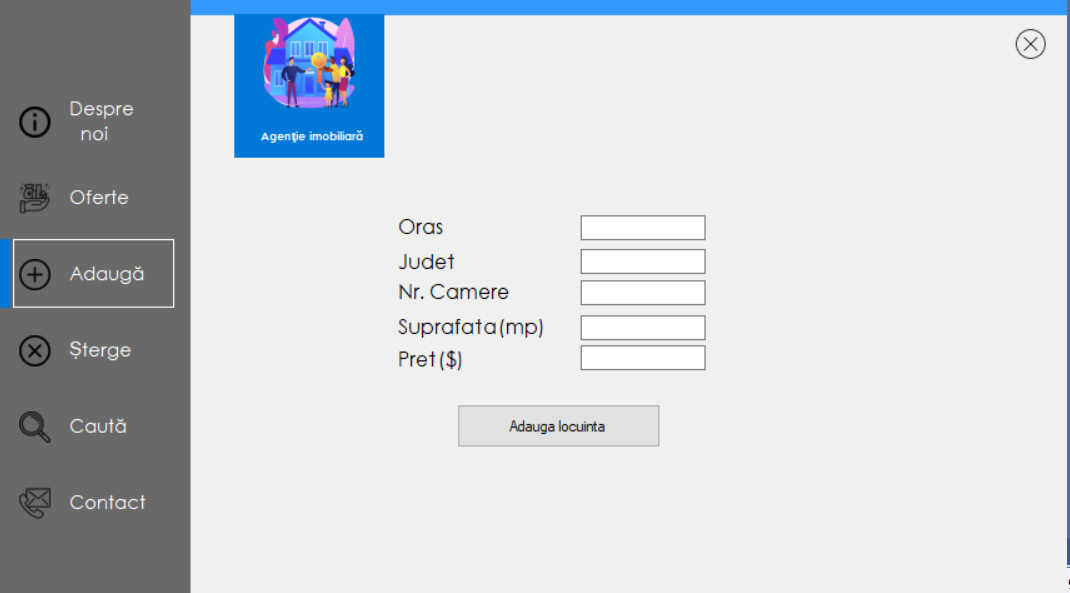
}

con.Close();

}

Pentru a actualiza informatiile, acest buton trebuie apasat in fiecare pagina.

In pagina **“Adauga”** putem adauga noi locuinte completand urmatoarele campuri: **“Oras”, “Judet”, “Nr. Camera”, “Suprafata”, “Pret”.**



Pentru a finaliza trebuie sa apasam butonul **“Adauga locuinta”** care are codul:

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

{

con.Open();

OleDbCommand cmd = con.CreateCommand();

cmd.CommandType = CommandType.Text;

cmd.CommandText = "insert into table1 values('"+addoras.Text+"', '"+ addjudet.Text+ "', '" + addnrcamere.Text + "', '" + addsuprafata.Text + "', '" + addpret.Text + "')";

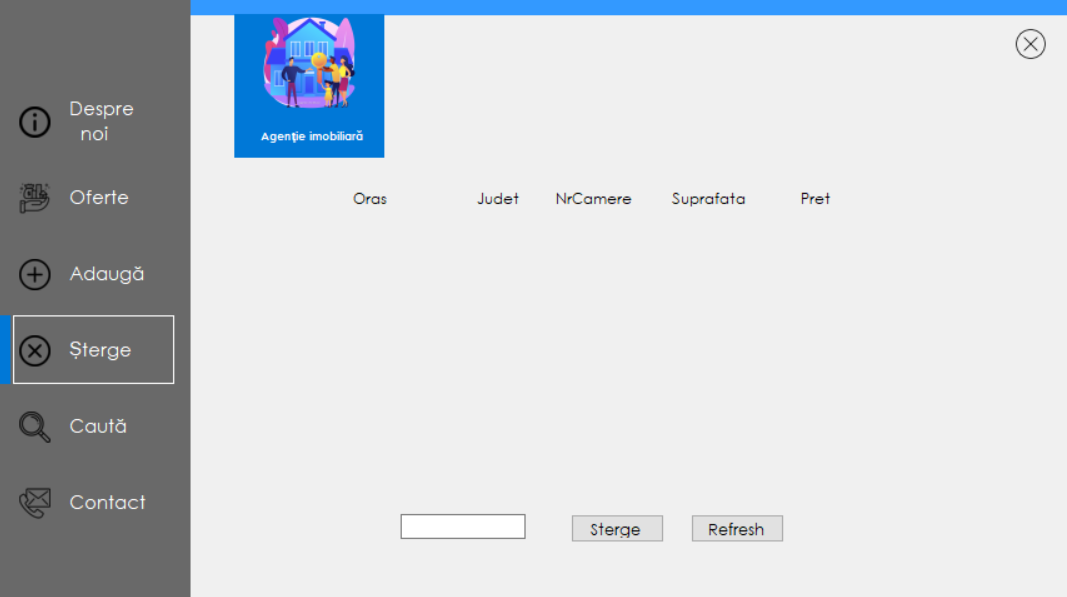
cmd.ExecuteNonQuery();

con.Close();

MessageBox.Show("Datele au fost adaugate cu succes.");

}

In pagina **“Sterge”** putem sterge o locuinta in functie de oras.



Pentru a sterge apasam butonul **“Sterge”** avand urmatorul cod:

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

{

con.Open();

string stergere;

stergere = "delete \* from Table1 where Oras = '" + orasdesters.Text + "'";

OleDbCommand cmd = new OleDbCommand(stergere, con);

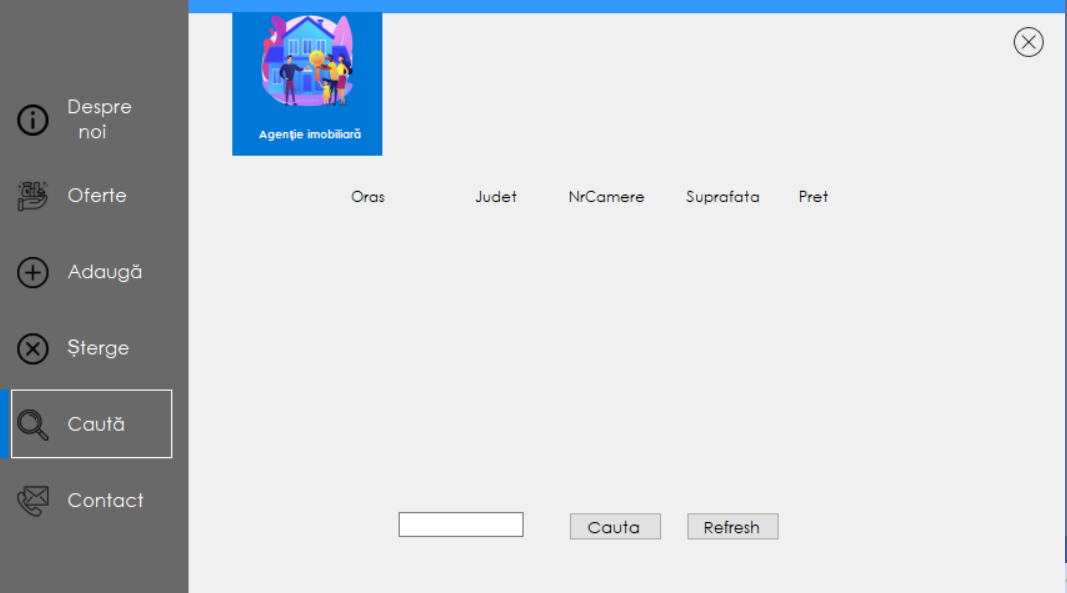
cmd.ExecuteNonQuery();

MessageBox.Show("Locuinta a fost eliminata!");

orasdesters.Text = "";

con.Close(); }

In pagina **“Cautare”** putem cauta o locuinta in functie de orasul dorit:



Pentru a finaliza cautarea se apasa butonul **“Cauta”** avand urmatorul cod:

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

{

con.Open();

string cautare = "SELECT \* FROM Table1 where Oras = '"; cautare += cautabox.Text +"'";

OleDbCommand cmd = new OleDbCommand(cautare, con);

OleDbDataAdapter adapter = new OleDbDataAdapter(cmd);

DataSet ds = new DataSet();

adapter.Fill(ds, "Table1");

label1.Text = "Oras" + "\n" + "\n";

label2.Text = "Judet" + "\n" + "\n";

label3.Text = "Nr camere" + "\n" + "\n";

label4.Text = "Suprafata" + "\n" + "\n";

label5.Text = "Pret" + "\n" + "\n";

foreach (DataRow r in ds.Tables["Table1"].Rows)

{

label1.Text = label1.Text + "\n" + r["Oras"] + "\n";

label2.Text = label2.Text + "\n" + r["Judet"] + "\n";

label3.Text = label3.Text + "\n" + r["NrCamere"] + "\n";

label4.Text = label4.Text + "\n" + r["Suprafata"] + "\n";

label5.Text = label5.Text + "\n" + r["Pret"] + "\n";

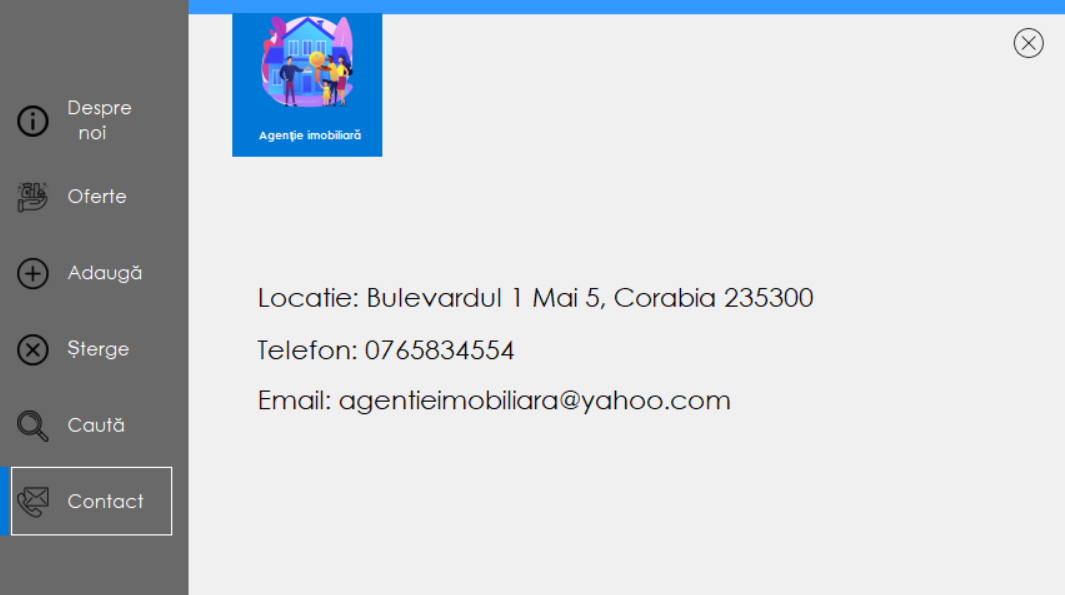
}

cautabox.Text = "";

con.Close();

}

In pagina **“Contact”** gasim datele de contact ale agentiei, precum adresa si numarul de telefon.



**Posibilitati de dezvoltare:**

Programul are scopul de a economisi timp si bani pentru viitorii cumparatori sau vanzatori ai locuintelor.

Un element ce ar aduce un plus aplicatiei, ar fi sortarea in functie de numarul de camere, suprafata sau pret. Acest lucru ar usura mult mai mult cautarile.

Un alt element care ar ajuta ulitizatorii, ar fi adaugare imaginilor locuintelor pe care acestia urmeaza sa le cumpere.

**Bibliografie**

-Manual „Curs de C# Programare in Visual C# 2008 Express Edition” de Constantin Galatan si Susana Galatan

- https://visualstudio.microsoft.com/vs/