***Liceul Teoretic "Grigore Moisil" Iaşi***

****

**Lucrare de atestat profesional în informatică**

**Elev: Profesor coordonator:**

**Brezuleanu Petru Giuliano Prof. Vîrgă Liliana**

***Iaşi- 2024***

**Curpins**

[Capitolul 1. Prezentarea generală a temei 3](#_Toc72855014)

[Capitolul 2. Resurse hardware și software necesare 4](#_Toc72855015)

[Capitolul 3. Realizarea aplicatiei 5](#_Toc72855016)

[3.1. Clase 6](#_Toc72855017)

[3.2 Fereastra frmMain 8](#_Toc72855018)

[3.3 Fereastra frmConectare 14](#_Toc72855019)

[3.4 Fereastra frmAdmin 17](#_Toc72855020)

[3.5 Fereastra frmClient 22](#_Toc72855021)

[3.6 Fereastra frmCont 22](#_Toc72855022)

[3.7 Baza de date 23](#_Toc72855023)

[Capitolul 4. Utilizarea aplicatiei 25](#_Toc72855024)

[Capitolul 5. Extinderi posibile 26](#_Toc72855025)

[Bibliografie 27](#_Toc72855026)

# Capitolul 1. Prezentarea generală a temei

Aplicația “IGym” a fost construită cu scopul de a ajuta in procesul de rezervare a unor abonamente sau locuri la evenimente de catre uitilizatori si de postare a acestora de catre administrator.

Prin intermediul acesteia se pot efectua urmatoarele operații:

* Crearea unui cont de utlizator
* Conectarea într-un cont existent
* Vizualizarea evenimentelor
* Vizualizarea abonamentelor
* Vizualizarea antrenorilor
* Rezervarea unui abonament
* Rezervarea unui eveniment
* Vizualizarea rezervarilor proprii
* Modificarea datelor personale ale utilizatorului
* Adaugarea evenimentelor, abonamentelor, antrenorilor de catre administrator
* Stergerea evenimentelor, abonamentelor, antrenorilor de catre administrator
* Vizualizarea rezervarilor de catre administrator
* Stergerea rezervarilor de catre administrator
* Vizualizarea clientilor de catre administrator
* Contactarea administratorului

Accesul la aplicație se face pe trei nivele de acces: vizitator, utilizator/client, administrator fiecare nivel având asociate drepturi de acces specifice:

* Vizitatorul poate viziona evenimentele si antrenorii disponibili, contacta administratorul si poate crea un cont de utilizator.
* Utilizatorul/clientul poate să rezerve un loc la un eveniment, sa rezerve un abonament, sa vizualizeze rezervarile facute de acesta si sa modifice datele personale.
* Administratorul poate să adauge si sa stearga evenimente, antrenori si abonamente, poate sa vizualizeze rezervarile facute de utilizatori/client si sa le stearga.

# Capitolul 2. Resurse hardware și software necesare

Pentru a rula în cele mai bune condiții, aplicația de “Gestiune sală fitness” are nevoie de următoarele resurse software și hardware:

* Windows 8/8.1/10
* XAMPP
* XAML
* .NET Core 3.1
* Intel(R) Core(TM) Intel(R) Core(TM) i5-8300H
* Memorie interna: 2048 MB RAM
* Memorie externa: 256 MB

# Capitolul 3. Realizarea aplicatiei

Aplicația a fost dezvoltată cu ajutorul .Net Core, Microsoft Visual C#, XAMPP si XAML.

Aplicația utilizează patru clase definite pentru:

* Eveniment
* Abonament
* Client
* Antrenor

Aplicația este constituită din cinci ferestre:

* frmMain
* frmConectare
* frmClient
* frmAdmin
* frmCont

Aplicați utilizează o bază de date MySQL - atestat, cu sase tabele:

* abonamente – tabel utilizat pentru a reține datele abonamentelor
* antrenori – tabel utilizat pentru a reține datele antrenorilor
* evenimente – tabel utilizat pentru a reține datele evenimentelor
* clienti – tabel utilizat pentru a reține datele clientilor
* rezabonamente – tabel utilizat pentru a reține rezervarile abonamentelor
* rezevenimente - tabel utilizat pentru a reține rezervarile evenimentelor

## 3.1. Clase

* Clasa Client:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Gestiune\_sala\_crossfit

{

class client

{

public static int id;

public static string nume, tel, email, parola;

public static DateTime data;

}

}

* Clasa Abonament:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Gestiune\_sala\_crossfit

{

class abonament

{

public string nume, tip, desc;

public int id,pret;

}

}

* Clasa Antrenor:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Gestiune\_sala\_crossfit

{

class antrenor

{

public string nume,desc,spec,cale;

public int id;

public DateTime varsta;

}

}

* Clasa Eveniment:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Gestiune\_sala\_crossfit

{

class eveniment

{

public string nume, desc,cale;

public DateTime? data;

public int id,pret;

}

}

## 3.2 Fereastra frmMain

Aceasta este fereastra principală a aplicației. Aceasta contine:

* un obiect de tip *Picturebox* care contine sigla platformei
* un obiect de tip *Button* prin intermediul careia se deschide fereastra de conectare
* un obiect de tip *Timer* care ajuta la derularea unor informatii automat
* patru obiecte de tip *Label* care deschid patru paneluri
* patru obiecte de tip *Panel*:
* panelul principal, home care contine:
* 2 obiecte de tip *Picturebox* pentru poze
* 2 obiecte de tip *Label* de unde se deschid 2 paneluri
* 2 obiecte de tip *Panel* unde sunt afisate informatii despre sala
* 3 obiecte de tip *Panel* unde se redau automat anumite informatii
* panelul eveniment in care se genereaza automat informatii despre evenimente
* panelul echipa in care se genereaza automat informatii despre antrenori
* panelul contact prin intermediul caruia se trimite un mesaj administratorului si care contine:
* 1 obiect de tip *Picturebox*
* 5 obiecte de tip *Label*
* 3 obiecte de tip *Textbox* in care se introduce numele, email-ul si mesajul
* 1 obiect de tip *Button* prin care se trimite mesajul

In panelul principal home care apare chiar la pornirea aplicatiei putem observa poze si informatii care se deruleaza automat, prin intermediul acestei secvente de cod:

private void timer1\_Tick(object sender, EventArgs e)

{

if (panel1.Visible == true)

{

panel1.Visible = false;

panel2.Visible = true;

}

else

{

if (panel2.Visible == true)

{

panel2.Visible = false;

panel3.Visible = true;

}

else

{

panel3.Visible = false;

panel1.Visible = true;

}

}

}

Functia care deschide fereastra frmConectare:

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

{

if (conect == null)

{

conect = new frmConectare();

conect.Show();

}

}

In panelul contact se poate trimite un mail administratorului cu ajutorul acestei secvente de cod:

private bool IsValidAddress(string mail)

{

try

{

MailAddress address = new MailAddress(mail);

return address.Address == mail;

}

catch

{

return false;

}

}

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

{

if(IsValidAddress(tbcemail.Text) == true && String.IsNullOrEmpty(tbcnume.Text)==false && String.IsNullOrEmpty(tbcmesaj.Text)==false)

{

MailMessage mail = new MailMessage();

mail.From = new MailAddress("igymromania@gmail.com");

mail.To.Add("igymromania@gmail.com");

mail.Subject = tbcnume.Text;

mail.Body = "Email: " + tbcemail.Text + "\n\n" + tbcmesaj.Text;

SmtpClient server = new SmtpClient("smtp.gmail.com");

server.Port = 587;

server.UseDefaultCredentials = false;

server.Credentials = new NetworkCredential("igymromania@gmail.com ", dzze yard bwmq qkax ");

server.EnableSsl = true;

server.Send(mail);

MessageBox.Show("Mesaj trimis cu succes", "Contact", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

}

}

Codul prin care se incarca in fereastra antrenorii si informatii despre acestia din baza de date pentru a putea fii vazuti de catre vizitatori:

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

{

lblhome.Font = new Font("Arial Rounded MT", 18, FontStyle.Bold);

lblevent.Font = new Font("Arial Rounded MT", 18, FontStyle.Bold);

lblcontact.Font = new Font("Arial Rounded MT", 18, FontStyle.Bold);

lblechipa.Font = new Font("Arial Rounded MT", 18, FontStyle.Italic);

nra = 0;

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"Select \* from antrenori";

//reader

MySqlDataReader r = cmd.ExecuteReader();

while (r.Read())

{

antrenori[nra] = new antrenor();

antrenori[nra].nume = r["nume"].ToString();

antrenori[nra].desc = r["descriere"].ToString();

antrenori[nra].spec = r["spec"].ToString();

antrenori[nra].id = int.Parse(r["ida"].ToString());

antrenori[nra].varsta = DateTime.Parse(r["varsta"].ToString());

antrenori[nra].cale = r["cale"].ToString();

nra++;

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

pnlant.Controls.Clear();

for (int i = 0; i < nra; i++)

{

oka = 1;

PictureBox pb = new PictureBox();

pb.Parent = pnlant;

pb.Location = new Point(15, i \* 180 + (i + 1) \* 15);

pb.Size = new Size(250, 180);

pb.SizeMode = PictureBoxSizeMode.StretchImage;

pb.ImageLocation = AppDomain.CurrentDomain.BaseDirectory + antrenori[i].cale;

pb.Click += pb\_click;

pb.Tag = i;

pb.BorderStyle = BorderStyle.FixedSingle;

}

pnlechipa.Visible = true;

pnlhome.Visible = false;

pnlcontact.Visible = false;

pnlevent.Visible = false;

antrenor a = antrenori[0];

pnldesc.Controls.Clear();

Label lbl = new Label();

#region nume

lbl.Text = a.nume;

lbl.Parent = pnldesc;

lbl.Location = new Point(0, 0);

lbl.Size = new Size(725, 60);

lbl.Font = new Font("Algeric", 30, FontStyle.Bold);

lbl.ForeColor = Color.White;

lbl.TextAlign = ContentAlignment.MiddleCenter;

#endregion

Label lbl2 = new Label();

#region specializare + varsta

lbl2.Text = "Specializare: " + a.spec + "\n" + "Varsta: " + (DateTime.Now.Year - a.varsta.Year).ToString() + " ani";

lbl2.Parent = pnldesc;

lbl2.Location = new Point(10, 65);

lbl2.AutoSize = true;

lbl2.Font = new Font("Algeric", 20);

lbl2.ForeColor = Color.White;

#endregion

Label lbl3 = new Label();

#region descriere

lbl3.Text = a.desc;

lbl3.Parent = pnldesc;

lbl3.Location = new Point(10, 140);

lbl3.Size = new Size(700, 500);

lbl3.AutoSize = false;

lbl3.Font = new Font("Algeric", 15);

lbl3.ForeColor = Color.White;

#endregion

}

## 3.3 Fereastra frmConectare

Prin intermediul acestei ferestre utilizatorul se poate conecta intr-un cont existent sau poate sa isi creeze un cont nou. Aceasta contine:

* 1 obiect de tip *Picturebox* pentru sigla
* 3 obiecte de tip *Panel*:
* panelul principal care contine:
* 2 obiecte de tip *Button* prin care se deschid celelalte paneluri
* 2 obiecte de tip *Picturebox*
* panelul pentru cont existent in care utilizatorul intra intr-un cont existent
* panelul pentru crearea unui cont nou in care utilizatorul isi creeaza un cont nou

In panelul principal care apare chiar la deschiderea ferestrei se foloseste urmatorul cod pentru a deschide celelalte paneluri:

private void btn1\_Click(object sender, EventArgs e)

{

pnlselect.Visible = false;

pnllogin.Visible = true;

pnlnew.Visible = false;

}

private void btn2\_Click(object sender, EventArgs e)

{

pnlselect.Visible = false;

pnllogin.Visible = false;

pnlnew.Visible = true;

}

Logarea se face prin intermediul codului de mai jos. Logarea este urmata instant de deschiderea ferestrei frmAdmin sau frmClient in functie de email-ul si parola introdusa.

#region login

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

{

if (tblogpass.Text == "admin" && tblogemail.Text == "igymromania@gmail.com") {

admin = new frmAdmin();

admin.Show();

this.Close();

}

else

{

if (tblogemail.TextLength > 0 && tblogpass.TextLength > 0)

{

try

{

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = "SELECT \* FROM clienti WHERE email = @pemail AND parola = @pparola";

cmd.Parameters.AddWithValue("pemail", tblogemail.Text);

cmd.Parameters.AddWithValue("pparola", tblogpass.Text);

MySqlDataReader r = cmd.ExecuteReader();

if (r.Read() && r["parola"].ToString() == tblogpass.Text)

{

client.id = int.Parse(r["idc"].ToString());

client.nume = r["nume"].ToString();

client.email = r["email"].ToString();

client.parola = r["parola"].ToString();

client.tel = r["telefon"].ToString();

clientul = new frmClient();

clientul.Show();

r.Close();

connection.Close();

this.Close();

}

else

{

MessageBox.Show("Email sau parola invalida!");

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

else

{

MessageBox.Show("Email sau parola invalida!");

}

}

}

#endregion

## 3.4 Fereastra frmAdmin

In aceasta fereastra are acces doar administratorul. Aici administratorul poate sa adauge si sa stearga un antrenor, un eveniment si un abonament. De asemenea el are acces la toate rezervarile facute de catre clienti si poate sa elimine o rezervare atat la evenimente cat si la abonamente.

Principalele obiecte in aceasta fereastra sunt:

* 5 obiecte de tip *Button*
* 5 obiecte de tip *Panel* care sunt deschise de cele 5 butoane
* panel pentru adaugarea unui antrenor nou
* panel pentru adaugarea unui eveniment nou
* panel pentru adaugarea unui abonament nou
* panel in care se pot vedea in obiecte de tip *DataGridView* abonamentele, evenimentele si antrenorii, de asemeanea se pot sterge
* panel in care se pot vedea si se pot sterge rezervarile facute de clienti atat la abonamente cat si la evenimente

Codul prin care se creeaza un abonament nou este urmatorul:

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

{

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

DialogResult result = MessageBox.Show("Sigur doriti sa creati un abonament nou?", "Adaugare abonament", buttons);

if (result == DialogResult.Yes)

{

try

{

string tip = "";

if (rbpersonal.Checked == true)

{

tip = "personal";

}

if (rbgrup.Checked == true)

{

tip = "grup";

}

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"INSERT INTO abonamente(nume,tip,descriere,pret) VALUES(@pnume,@ptip,@pdescriere,@ppret);";

//stabilirea parametrilor

cmd.Parameters.AddWithValue("pnume", tbnumeab.Text);

cmd.Parameters.AddWithValue("pdescriere", tbdescab.Text);

cmd.Parameters.AddWithValue("ppret", tbpretab.Text);

cmd.Parameters.AddWithValue("ptip", tip);

//verificare

if (cmd.ExecuteNonQuery() != 0)

{

MessageBox.Show("Abonament creat!", "Creare abonament", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

Prin coduri asemanatoare se face adaugarea unui eveniment si a unui antrenor.

Astfel se genereaza tabelele in care administratorul poate sa vada evenimentele, abonamentele si antrenorii intr-un obiect de tip *DataGridView* si eventual sa le stearga:

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

{

pnlstergere.Visible = true;

pnlstergere.BringToFront();

dtant.Rows.Clear();

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"Select \* from antrenori";

//reader

MySqlDataReader r = cmd.ExecuteReader();

string[] s = new string[100];

while (r.Read())

{

s[0] = r["ida"].ToString();

s[1] = r["nume"].ToString();

s[2] = r["spec"].ToString();

s[3] = r["varsta"].ToString();

dtant.Rows.Add(s);

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

dtab.Rows.Clear();

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"Select \* from abonamente";

//reader

MySqlDataReader r = cmd.ExecuteReader();

string[] s = new string[100];

while (r.Read())

{

s[0] = r["idab"].ToString();

s[1] = r["nume"].ToString();

s[2] = r["tip"].ToString();

s[3] = r["pret"].ToString();

dtab.Rows.Add(s);

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

dtevent.Rows.Clear();

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"Select \* from evenimente";

//reader

MySqlDataReader r = cmd.ExecuteReader();

string[] s = new string[100];

while (r.Read())

{

s[0] = r["ide"].ToString();

s[1] = r["nume"].ToString();

s[2] = r["data"].ToString();

s[3] = r["pret"].ToString();

dtevent.Rows.Add(s);

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

## 3.5 Fereastra frmClient

In aceasta fereastra are acces utilizatorul. Aceasta fereastra apare dupa efectuarea logarii sau crearea unui cont nou. Din aceasta fereastra utilizatorul poate sa rezerva un abonament sau un eveniment si poate sa vada rezervarile facute de acesta.

Principalele obiecte ale acestei ferestre sunt:

* 3 obiecte de tip *Button* care deschid 3 obiecte de tip *Panel*
* 1 obiect de tip *Button* care deschide o fereastra noua
* 4 obiecte de tip *Pannel*:
* panelul principal de unde apar butoanele si descrierile acestora
* panelul de rezervare a abonamentelor in care utilizatorul poate vizualiza si rezerva abonamentele
* panelul de rezervare a evenimentelor in care utilizatorul poate vizualiza si rezerva un loc la un eveniment
* panelul de vizualizare a rezervarilor facute de utilizator

Codul prin care se genereaza obiecte de tip *DataGridView* in care utilizatorul poate sa isi vada propriile rezervari facute este urmatorul:

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

{

pnlrez.Visible = true;

pnlevent.Visible = false;

pnlmain.Visible = false;

pnlab.Visible = false;

dtev.Rows.Clear();

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"Select \*,evenimente.pret as pretev, evenimente.nume as numeevent from rezevenimente inner join clienti on rezevenimente.idc = clienti.idc inner join evenimente on rezevenimente.ide = evenimente.ide where rezevenimente.idc = @idc";

cmd.Parameters.AddWithValue("idc", client.id);

//reader

MySqlDataReader r = cmd.ExecuteReader();

string[] s = new string[100];

while (r.Read())

{

s[0] = r["idre"].ToString();

s[1] = r["numeevent"].ToString();

s[2] = r["pretev"].ToString();

dtev.Rows.Add(s);

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

dtab.Rows.Clear();

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"Select \*,abonamente.pret as pretab, abonamente.nume as numeab from rezabonamente inner join clienti on rezabonamente.idc = clienti.idc inner join abonamente on rezabonamente.idab = abonamente.idab where rezabonamente.idc = @idc";

cmd.Parameters.AddWithValue("idc", client.id);

//reader

MySqlDataReader r = cmd.ExecuteReader();

string[] s = new string[100];

while (r.Read())

{

s[0] = r["idra"].ToString();

s[1] = r["numeab"].ToString();

s[2] = r["pretab"].ToString();

dtab.Rows.Add(s);

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

Dupa ce sunt generate intr-un panel separat abonamentele, unde utilizatorul le poate vizualiza, acesta poate sa rezerva un abonament, proces ce se realizeaza prin intermediul codului urmator:

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

{

bool adauga = true;

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"Select \* from rezabonamente where idc = @idc and idab = @idab";

cmd.Parameters.AddWithValue("idc", client.id);

cmd.Parameters.AddWithValue("idab", abonamente[int.Parse((sender as Button).Tag.ToString())].id);

//reader

MySqlDataReader r = cmd.ExecuteReader();

if (r.Read())

{

adauga = false;

}

}

catch

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

if (adauga)

{

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"INSERT INTO rezabonamente(idc,idab) VALUES(@idc,@idab);";

//stabilirea parametrilor

cmd.Parameters.AddWithValue("idc", client.id);

cmd.Parameters.AddWithValue("idab", abonamente[int.Parse((sender as Button).Tag.ToString())].id);

//verificare

if (cmd.ExecuteNonQuery() != 0)

{

MailMessage mail = new MailMessage();

mail.From = new MailAddress("igymromania@gmail.com");

mail.To.Add(client.email);

mail.Subject = "Rezervare facuta cu succes";

mail.Body = "Ati rezervat cu succes abonamentul: " + abonamente[int.Parse((sender as Button).Tag.ToString())].nume + '\n' + "Pretul este de: " + abonamente[int.Parse((sender as Button).Tag.ToString())].pret + " lei/luna." + '\n' +'\n' + '\n'+ "IGym \nAdresa: Iaşi, Strada Petre Andrei 9, Iași 700496 \nTelefon: 0771 518 956";

SmtpClient server = new SmtpClient("smtp.gmail.com");

server.Port = 587;

server.UseDefaultCredentials = false;

server.Credentials = new NetworkCredential("igymromania@gmail.com", "dzze yard bwmq qkax ");

server.EnableSsl = true;

server.Send(mail);

MessageBox.Show("Abonament rezervat cu succes!", "Rezervare abonament", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

else

{

MessageBox.Show("Aveti deja rezervare facuta pentru acest abonament!", "Rezervare abonament", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

## 3.6 Fereastra frmCont

Din aceasta fereastra utilizatorul conectat poate sa isi modifice datele de utilizator, precum: numele, numarul de telefon si parola.

Codul prin care se modifica datele este urmatorul:

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

{

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

DialogResult result = MessageBox.Show("Sigur doriti sa modificati datele?", "Modificare date", buttons);

if (result == DialogResult.Yes)

{

try

{

//creare conexiune

MySqlConnection connection = new MySqlConnection();

connection.ConnectionString = @"Server=localhost; DATABASE=IGym; UID=root; PASSWORD=; allow User Variables=True";

connection.Open();

//creare comanda

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = connection;

cmd.CommandText = @"UPDATE clienti SET `nume`=@nume,`telefon`=@telefon where idc=@idc";

cmd.Parameters.AddWithValue("nume", tbnume.Text);

cmd.Parameters.AddWithValue("telefon", tbtel.Text);

cmd.Parameters.AddWithValue("idc", client.id);

if (cmd.ExecuteNonQuery() != 0)

{

MessageBox.Show("Date modificate!", "Date modificate", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

catch (Exception ex)

{

MessageBox.Show("A aparut o eroare, va rugam sa contactati un administrator!", "Eroare", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

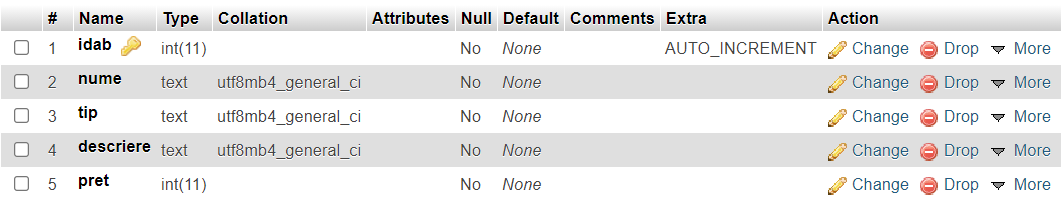
}

}

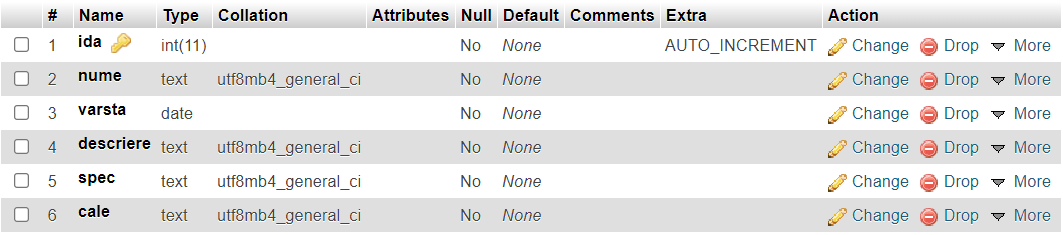
## 3.7 Baza de date

Aplicați utilizează o bază de date MySQL - atestat, cu sase tabele:

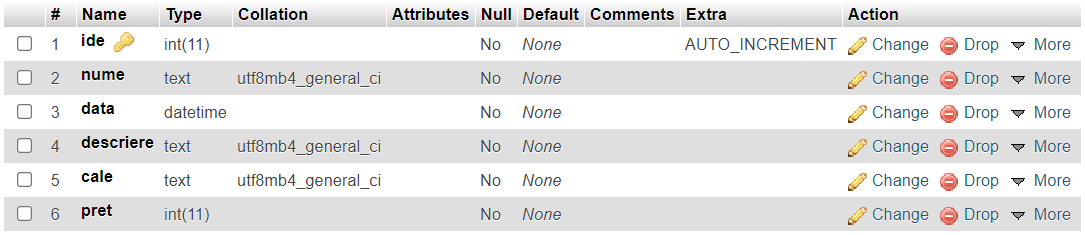
* abonamente – tabel utilizat pentru a reține datele abonamentelor



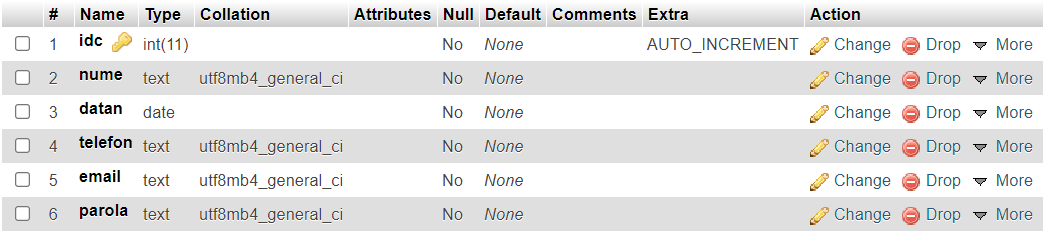
* antrenori – tabel utilizat pentru a reține datele antrenorilor



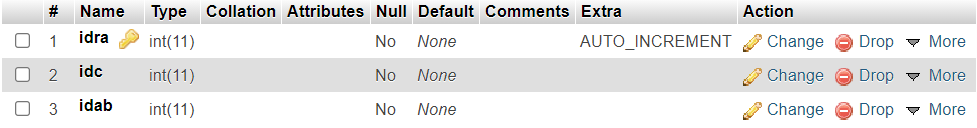
* evenimente – tabel utilizat pentru a reține datele evenimentelor



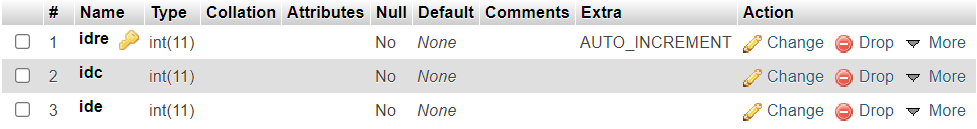
* clienti– tabel utilizat pentru a reține datele clientilor



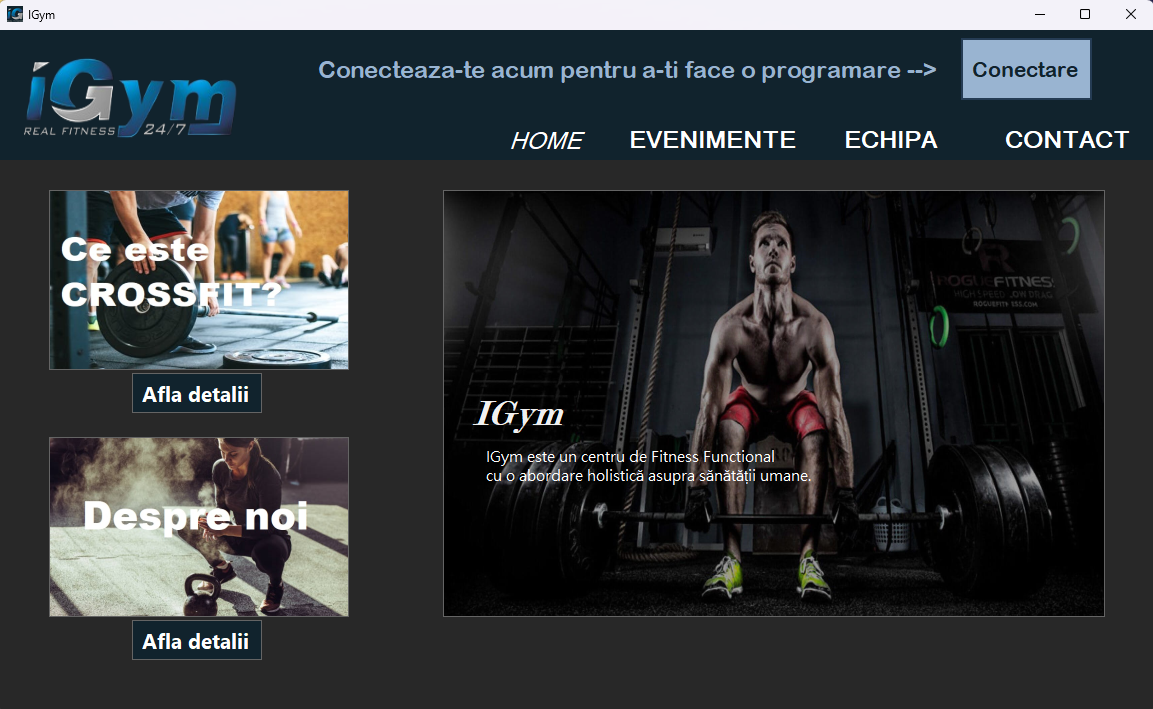
* rezabonamente – tabel utilizat pentru a reține rezervarile abonamentelor



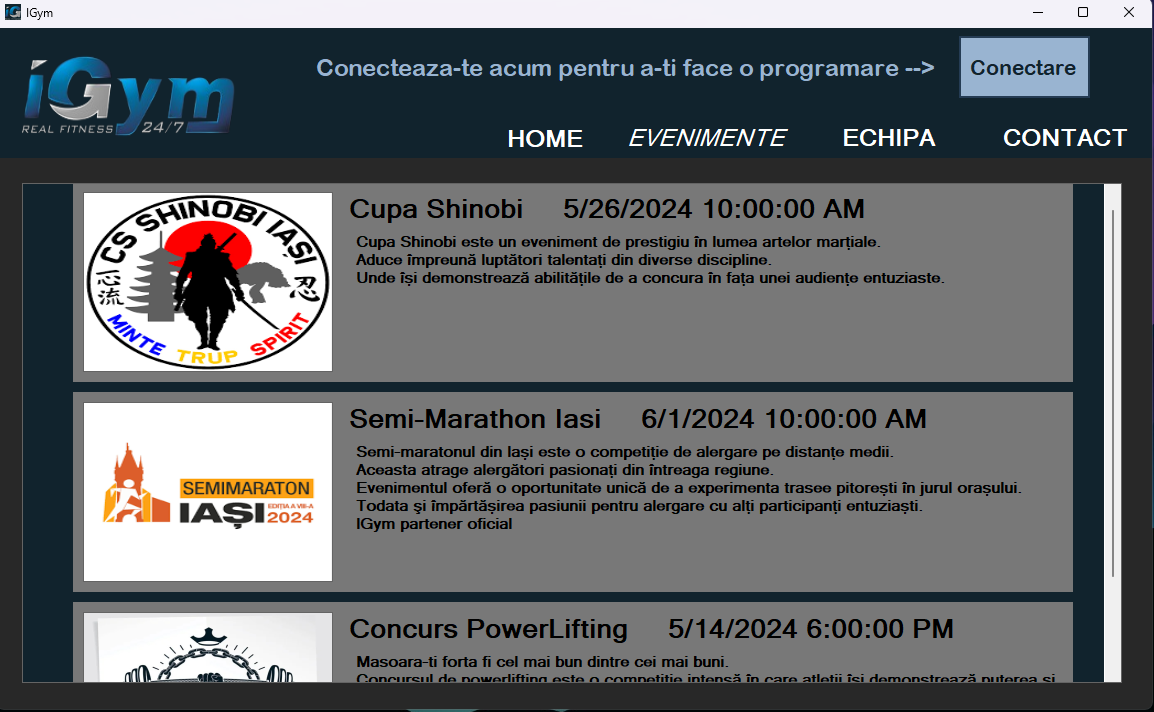
* rezevenimente- tabel utilizat pentru a reține rezervarile evenimentelor



# Capitolul 4. Utilizarea aplicatiei

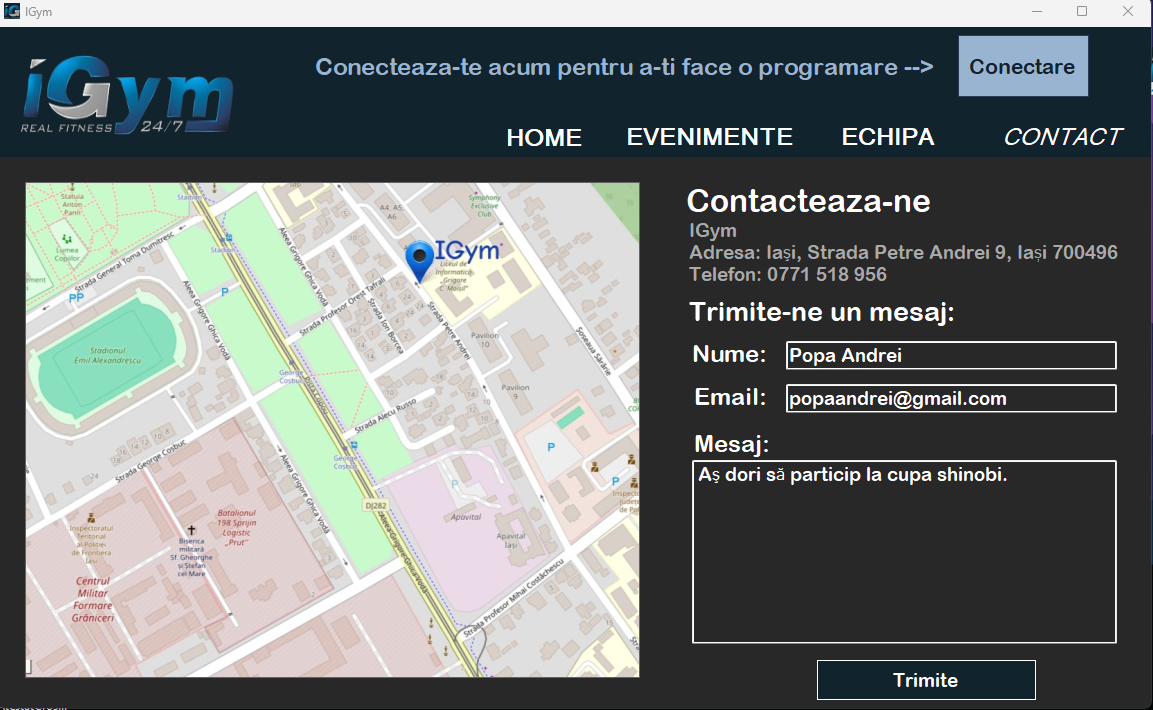
La deschiderea aplicatiei este afisata fereastra *IGym* si panelul principal home, unde vizitatorul poate sa vada unele informatii despre crossfit si despre sala. Din panelul main se poate deschide fereastra *Conectare*, prin apasarea butonului *Conectare.*

Cand se apasa pe *EVENIMENTE* vizitatorul poate sa vada evenimentele disponibile si informatii despre acestea.



Cand se apasa pe *ECHIPA* vizitatorul poate sa vada antrenorii, iar cand apasa pe poza unui antrenor apar niste informatii despre acesta.

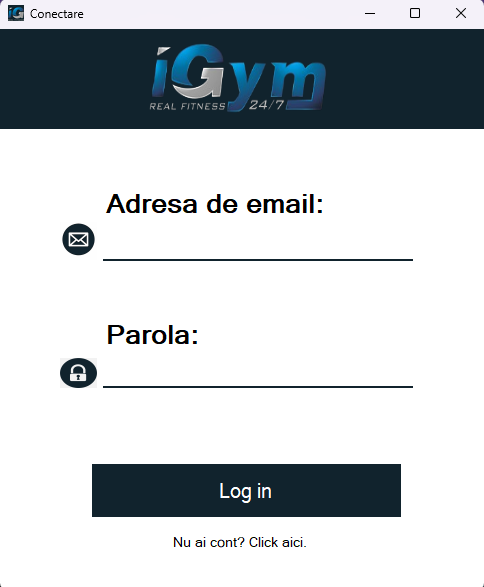
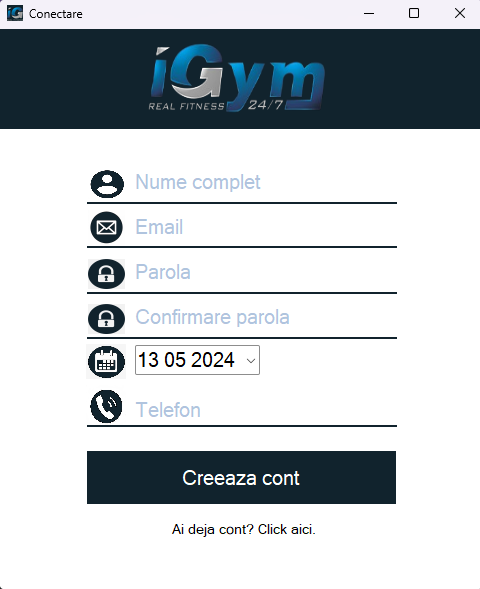


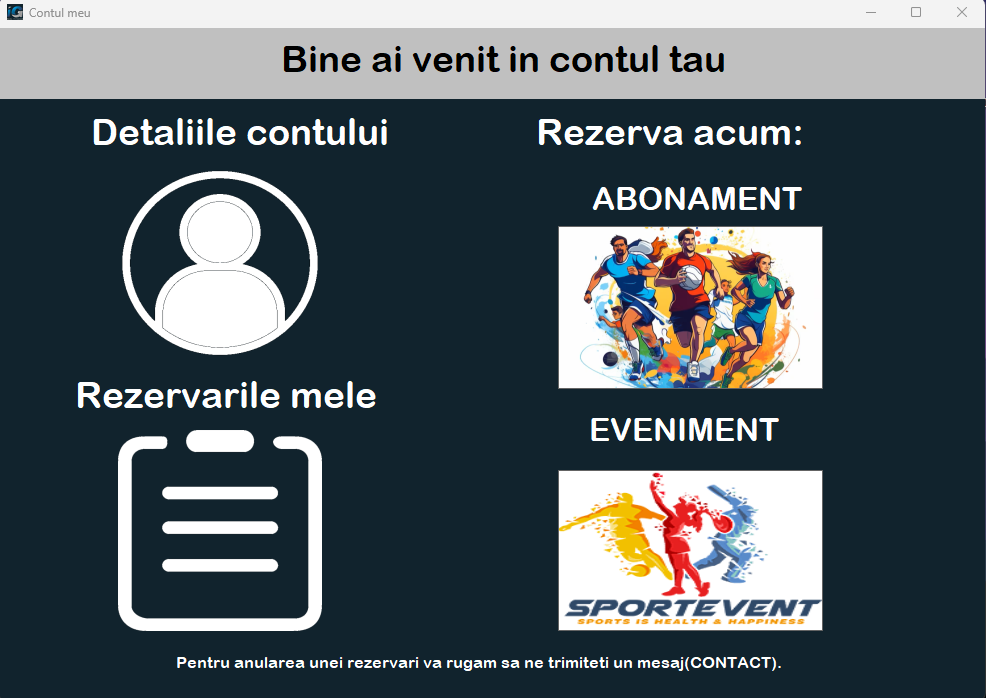
La apasarea butonului *CONTACT* se deschide panelul in care vizitatorul poate sa trimita un mesaj administratorului. Mesajul va fi trimis pe email-ul administratorului.



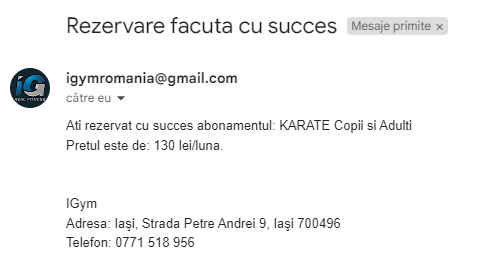
Dupa apasarea butonului *Conectare* se deschide fereastra *Conectare* unde vizitatorul poate sa aleaga daca doreste sa se conecteze intr-un cont existent sau sa isi creeze un cont nou.



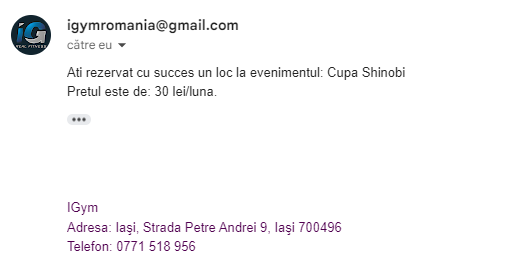


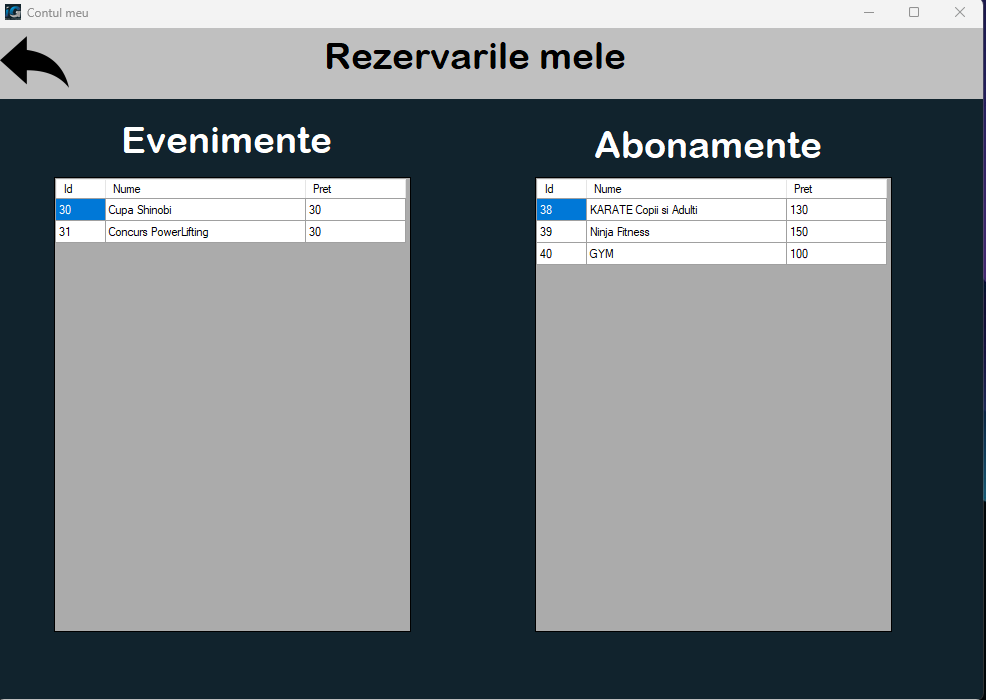
Dupa realizarea conectarii se afiseaza fereastra *Client*, daca conectarea s-a efectuat cu un cont de utilizator. In aceasta fereastra, initial apare panelul principal cu 4 butoane care redirectioneaza utilizatorul la panelul sau fereastra dorita.

Daca utilizatorul decide sa rezerve un abonament va apasa pe imaginea de sub *Rezerva acum: - un abonament* si se va afisa panelul din care se poate face rezervare pentru un abonament. ( rezervarea se face de pe butonul +, dupa apare un mesaj in care apare rezervara facut cu success, iar apoi pe adresa de email va fi trimis automat un mesaj in care se confirma rezervarea abonamentului)

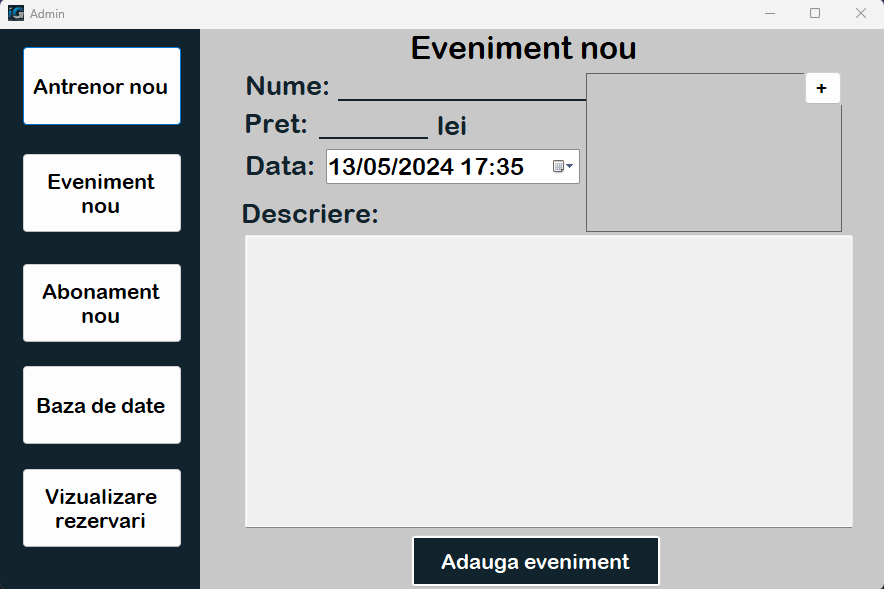


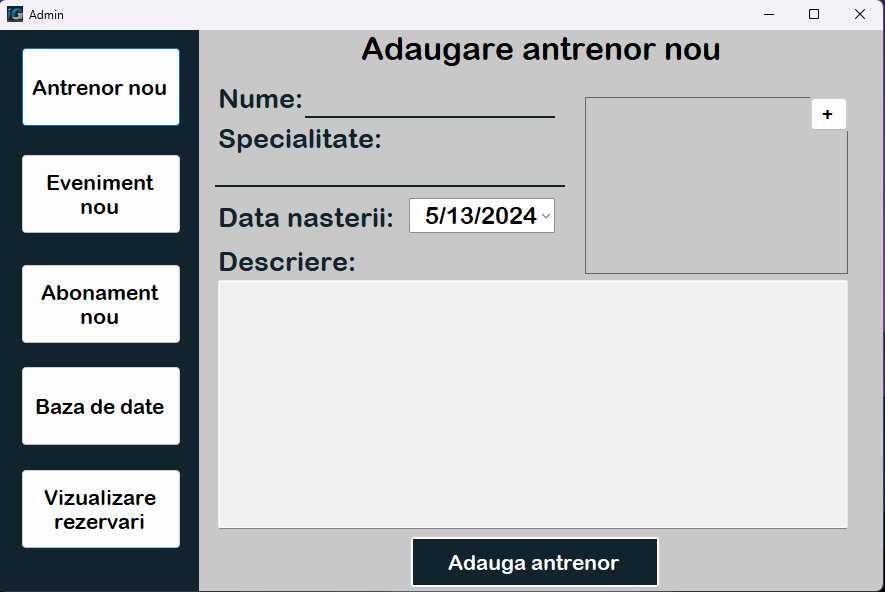
In mod asemanator functioneaza si rezervarea evenimentelor.

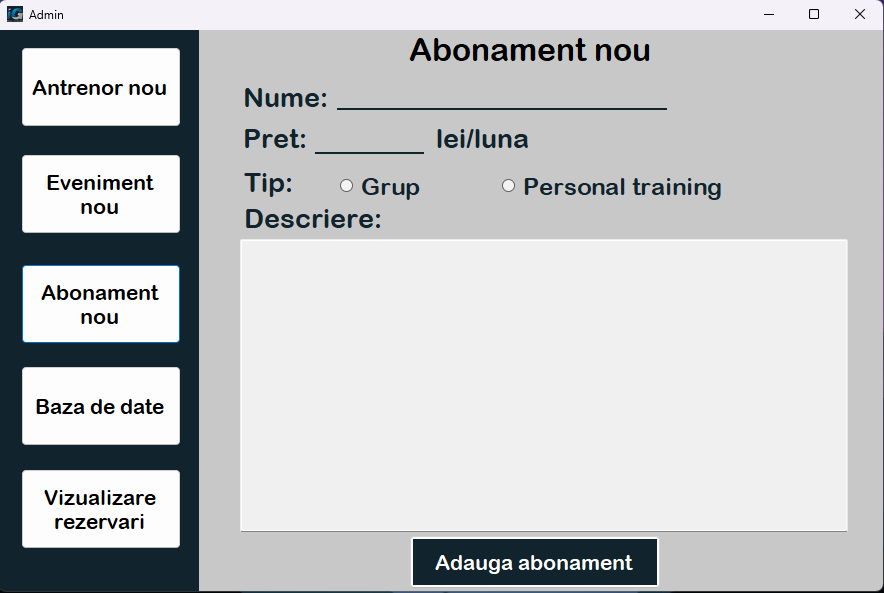


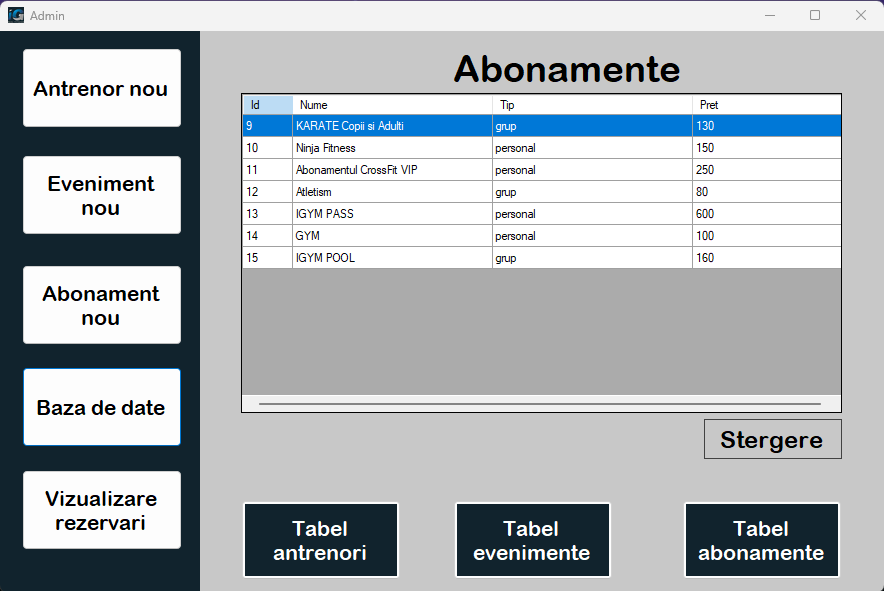
Cand se apasa pe imaginea de sub *Rezervarile mele* apare un panel in care in doua obiecte de tip *DataGridView* utilizatorul poate sa vada rezervarile facute de el insusi.

La apasarea imaginii de sub *Detaliile contului* se deschide fereastra *frmCont* unde utilizatorul poate sa vada datele personale si sa le modifice daca doreste.

Daca la realizarea logarii se introduce adresa de email si parola administratorului atunci se deschide fereastra *Admin* de unde administratorul poate sa faca diferite operatii. Daca administratorul apasa pe butoanele *Antrenor nou, Eveniment nou, Abonament nou* vor aparea panelul de adaugare specific fiecarui buton.





La apasarea butonului *Baze de date* apare un panel in care administratorul poate sa vada in obiecte de tip *DataGridView* abonamntele, evenimentele si antrenorii si de asemenea poata sa ii stearga. Butoanele  *Tabel antrenori, Tabel evenimente, Tabel abonamente* fac sa apara in panel tabelul corespunzator.

La apasarea butonului *Vizualizare rezervari* administratorul poate sa vada toate rezervarile facute de clienti si poata sa le stearga daca doreste.

# Capitolul 5. Extinderi posibile

Aceasta aplicatie ofera toate functionalitatile necesare unui program de gestiune sala de crossfit. Utilizatorii pot vizualiza evenimentele, abonamentele si antrenorii. De asemenea utilizatorii pot sa rezerve un loc la un eveniment sau un tip de abonament. Dupa rezervare utilizatorii vor primi un email in care se confirma crearea rezervarii.

Vizitatorii pot sa vada care sunt antrenorii si evenimentele disponibile fara a crea un cont de utilizator, avand posibilitatea sa ia legatura cu administratorul prin intermediu *CONTACTULUI,* administratorul primid un email de la vizitator.

Administratorii pot sa adauge si sa stearga abonamente, evenimente si antrenori. De asemenea ei pot sa vizualizeze toate rezervarile facute de utilizatori sis a elimine o rezervare daca este necesar

Asadar programul permite comunicarea intre administrator si utilzator intr-un mod simplu si eficient, lucru care permite o buna functionare a acesteia.

Totusi, un element care poate asigura in continuare integritatea programului este adaugarea pentru administrator de a sterge un cont de utilizator.

# Bibliografie

Edx (n.d.). Retrieved from **https://www.edx.org/learn/xampp**

W3school (n.d.). Retrieved from https://www.w3schools.com/cs/index.php

Microsoft. (n.d.). Retrieved from https://dotnet.microsoft.com/en-us/learn/csharp

Microsoft. (n.d.). Retrieved from https://docs.microsoft.com/en-us/dotnet/csharp/

Stackoverflow. (n.d.). Retrieved from https://stackoverflow.com/