



Ministerul Educației și Cercetării al Republicii Moldova
Centrul de Excelență în Informatică și Tehnologii Informaționale
Catedra Informatică I

Raport

pentru practica tehnologică

Disciplina: Asistență pentru programarea vizuală

Tema: *Concurs de dans*

Realizat: Solomon Anastasia P-1933

(Numele, prenumele elevului, grupa)

Verificat: Nebunelea Diana

(Numele, prenumele profesorului)

Nota _____

Chișinău 2022

Cuprins

Introducere	5
Condiția sarcinii.....	5
Listingul aplicației	6
Folder-ul Classes.....	6
Folder-ul DatabaseManipulation.....	6
Fișier-ul DataAccess.cs.....	6
Fișier-ul Helper.cs.....	12
Folder-ul DatabaseTables.....	12
Fișier-ul Dancer.cs	12
Folder-ul LogIn	13
Fișierul LogInCaAdministratorForm.cs	13
Fișierul LogInForm.cs	14
Folder-ul MenuStripForms.....	15
Fișierul Categorii.cs	15
Fișierul DansatoriForm.cs	16
Fișierul TipuriDeDansForm.cs	17
Folder-ul Tasks	18
Fișierul CeaMaiTanaraDansatoareForm.cs	18
Fișierul DansatoriCuInstitutiaForm.cs	18
Fișierul DansatoriDupaCategoriiForm.cs.....	20
Fișierul DansatorTanarBatranForm.cs	20
Fișierul DateDeContactInTabelaForm.cs	21
Fișierul ExcludereDansatorForm.cs	23
Fișierul ExportaDansatoareAniForm.cs	25
Fișierul InregistrareDansatorForm.cs	27
Fișierul ListaConcursurilorCuDansatoriForm.cs	31
Fișierul OrdonareAscendentDupaVarstaForm.cs	32
Fișierul PremiiForm.cs	33
Fișierul VarstaMedieABarbatilorForm.cs	34
Fișierul ConcursDeDansForm.cs.....	35

Fișierul Program.cs	39
Prezentarea generală a aplicației	40
Interfața de logare	42
Interfața meniului principal al aplicației	43
Rezultatele testării aplicației	44
Logarea în calitate de utilizator obișnuit	44
Logarea în calitate de administrator	46
Erori întâlnite în cadrul logării calitate de administrator	47
Meniul principal	48
Bara de navigare	48
Ieșire	48
Afișarea dansatorilor	48
Afișarea categoriilor	50
Afișarea tipurilor de dans	51
Înregistrarea unui dansator nou	52
Erori întâlnite în cadrul înregistrării unui dansator nou	56
Excluderea unui dansator	58
Erori întâlnite în cadrul excluderii unui dansator nou	61
Afișarea dansatorului cel mai Tânăr / Bătrân	63
Ordonarea după vîrstă	64
Erori întâlnite în cadrul afișării dansatorilor ordonați asc după vîrstă	66
Dansatoarea cea mai Tânără	67
Afișarea listei dansatorilor, în funcție de categoria de dans selectată	68
Erori întâlnite în cadrul afișării dansatorilor după categorii	69
Vîrsta medie a bărbaților, în funcție de categoria de dans selectată	70
Erori întâlnite în cadrul afișării vîrstei medie a bărbaților	72
Exportarea dansatoarelor într-un fișier MS EXCEL	73
Erori întâlnite în cadrul exportării datelor	75
Datele de contact a dansatorilor	76
Afișarea listei dansatorilor cu instituția de învățământ	78
Afișarea listei dansatorilor cu premiile câștigate	80
Erori întâlnite în cadrul afișării listei dansatorilor cu premiile câștigate	81

Afișarea listei dansatorilor, în funcție de concursul selectat	82
Prezentarea generală a bazei de date.....	84
Diagrama E-R a bazei de date ConcursDeDans	84
Scenariul bazei de date ConcursDeDans	85
Conținutul tabelelor.....	98
Tabelul Instituții.....	98
Tabelul CategorieDansuri	99
Tabelul TipuriDans.....	99
Tabelul Dansatori.....	100
Tabelul DateDeContact.....	100
Tabelul Concursuri	101
Tabelul Acte	101
Tabelul Premii	102
Tabelul Utilizatori	102
Concluzii	103
Bibliografie / Webografie	105
Link către drive:	105

Introducere

Ca temă spre realizare la practica tehnologică, mi s-a propus realizarea unei aplicații, ce va gestiona datele unor concursuri de dans. Sarcina conține 12 itemi (2 dintre care sunt incluse de mine), unele meniuri user-friendly și de asemenea o interfață plăcută și armonioasă.

Iar baza de date ConcursDeDans este creată pentru a duce evidența despre activitatea concursurilor, organizarea structurată a datelor, lucru asupra datelor existente, precum și gestionarea eficientă a acestora.

Condiția sarcinii

Biletul 26: CONCURS DE DANS. Informația despre participanții la un concurs de dans este înregistrată într-o bază de date. Elaborați o bază de date, gestiunea căreia se va realiza dintr-o aplicație vizuală. În dependență de tematică se vor specifica criteriile de integritate pentru baza de date, administra obiecte ale bazei de date, stabili planul de întreținere a bazei de date și moduri de securizare a datelor din baza de date. Informațiile din baza de date se vor prelucra la solicitarea utilizatorului folosind meniuri, formulare și control-ere, ținând cont de următoarele funcționalități:

- 1) Înregistrează un nou participant;
- 2) Exclude înregistrarea, ce corespunde participantului cu codul introdus de la tastatură
- 3) Determină participantul cel mai în vîrstă și cel mai Tânăr, și ce tip de dans practică;
- 4) Afisează la ecran lista tuturor participanților, ordonată ascendent conform vîrstei acestora;
- 5) Determină numele celei mai tinere persoane de sex feminin;
- 6) Afisează lista participanților la categoria de dans indicată;
- 7) Determină și afisează vîrstă medie a participanților bărbați pentru fiecare categorie de dans;
- 8) Exportă într-un fișier MS Word sau MS Excel lista tuturor participantelor cu vîrstă sub 12 ani;
- 9) Creează o tabelă ce va conține date de contact al fiecărui participant;
- 10) Afisează lista participanților și instituția unde își fac studiile;
- 11) Afisează lista premiilor luate, codul, numele și prenumele participanților, în funcție de concursul selectat;
- 12) Afisează lista dansatorilor, în funcție de concursul selectat.

Listingul aplicației

Folder-ul Classes

Folder-ul DatabaseManipulation

Fișier-ul DataAccess.cs

```
using Dapper;
using System.Collections.Generic;
using System.Data;
using System.Linq;

namespace Concurs_de_dans
{
    class DataAccess
    {
        // Metoda ce preia denumirile concursurilor
        public static List<string> GetCompetitionName()
        {
            using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
            {
                List<string> output =
connection.Query<string>("dbo.selectCompetition").ToList();
                return output;
            }
        }

        // Metoda ce preia datele dansatorilor
        public static List<Dancer> GetDancers()
        {
            using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
            {
                List<Dancer> dancers =
connection.Query<Dancer>("dbo.selectDancers").ToList();
                return dancers;
            }
        }

        // Metoda ce preia denumirea categoriilor de dans
        public static List<string> GetCategoryName()
        {
            using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
            {
                List<string> output =
connection.Query<string>("dbo.selectCategory").ToList();
                return output;
            }
        }

        // Metoda ce preia denumirea tipurilor de dans
        public static List<string> GetDanceType()
        {
            using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
            {
```

```

        List<string> output =
connection.Query<string>("dbo.selectDanceType").ToList();
            return output;
        }
    }

    // Metoda ce preia datele dansatorilor, in functie de codul introdus
    public static List<Dancer> GetDancersByCode(string code)
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<Dancer> output =
connection.Query<Dancer>($"dbo.selectDancersByCode @Code = '{code}'").ToList();
            return output;
        }
    }

    // Metoda ce determina cine s-a logat
    public static void WhoLogin(string username)
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            connection.Execute($"CREATE OR ALTER VIEW currentUser (IdUser,
Username, Parola, Email, EsteAdministrator) " +
"$AS SELECT * FROM Utilizatori WHERE Username =
'{username}'");
        }
    }

    // Metoda ce selecteaza numele din vederea cu utilizatorul curent
    public static List<string> ViewLogin()
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<string> output =
connection.Query<string>("dbo.selectFromView").ToList();
            return output;
        }
    }

    // Metoda ce selecteaza numele administratorului logat
    public static List<string> GetAdministratorLogIn(string username, string
parola)
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<string> output = connection.Query<string>($"dbo.checkUser
@Username = '{username}', @Parola = '{parola}'").ToList();
            return output;
        }
    }

    // Metoda ce inregistreaza un dansator nou
    public static void InsertDancer(string numeInstitutie, string
adresaInstitutie, string denumireTipDans,

```

```

                string codDansator, string numeDansator,
string prenumeDansator,
                string sexDansator, string dataNasterii,
int experienta, string denumireConcurs,
                string email, string telefon, string
adresaDansator)
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
    {
        connection.Execute($"dbo.insertDancer @NumeInstitutie =
'{numeInstitutie}', " +
$"@AdresaInstitutie = '{adresaInstitutie}', " +
$"@DenumireTipDans = '{denumireTipDans}', "
@CodDansator = '{codDansator}', " +
$"@NumeDansator = '{numeDansator}', "
@PrenumeDansator = '{prenumeDansator}', " +
$"@SexDansator = '{sexDansator}', @DataNasterii
= '{dataNasterii}', " +
$"@Experienta = {experienta}, @DenumireConcurs =
'{denumireConcurs}', " +
$"@Email = '{email}', @Telefon = '{telefon}', "
@AdresaDansator = '{adresaDansator}');");
    }
}

// Metoda ce insereaza premiul unui dansator
public static void InsertPrize(string locul, int premiuBanesc, string
denumireConcurs)
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
    {
        connection.Execute($"dbo.insertPrize @Locul = '{locul}', "
@PremiuBanesc = {premiuBanesc}, @DenumireConcurs = '{denumireConcurs}');");
    }
}

// Metoda ce exclude un dansator
public static void DeleteDancer(string cod)
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
    {
        connection.Execute($"dbo.deleteDancer @CodDansator = '{cod}');");
    }
}

// Metoda ce preia datele celui mai tanar dansator
public static List<Dancer> GetYoungestDancer()
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
    {
        List<Dancer> output =
connection.Query<Dancer>("dbo.youngestDancer").ToList();
        return output;
    }
}

```

```

    }

    // Metoda ce preia datele celui mai in varsta dansator
    public static List<Dancer> GetOldestDancer()
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<Dancer> output =
connection.Query<Dancer>("dbo.oldestDancer").ToList();
            return output;
        }
    }

    // Metoda ce preia codul, numele, prenumele si data nasterii dansatorilor
    // din toate concursurile, ordonati ascendent dupa varsta
    public static List<Dancer> GetSortedAscDancersByDate()
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<Dancer> output =
connection.Query<Dancer>("dbo.sortAscAllDancersByDate").ToList();
            return output;
        }
    }

    // Metoda ce preia codul, numele, prenumele si data nasterii dansatorilor
    // din concursul selectat, ordonati ascendent dupa varsta
    public static List<Dancer>
GetSortedDancersAscFromOneCompetitionByDate(string denumireConcurs)
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<Dancer> output =
connection.Query<Dancer>($"dbo.sortDancersASCFromOneCompetitionByDate
@denumireConcurs = '{denumireConcurs}'").ToList();
            return output;
        }
    }

    // Metoda ce preia codul, numele, prenumele, data nasterii si sexul
    // cele mai tinere dansatoare de sex feminin
    public static List<Dancer> GetYoungestFemaleDancer()
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<Dancer> output =
connection.Query<Dancer>("dbo.youngestFemaleDancer").ToList();
            return output;
        }
    }

    // Metoda ce preia datele dansatorilor si denumirea categoriei,
    // in functie de categoria selectata
    public static List<Dancer> GetDancersByCategory(string categorie)
    {

```

```

        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
{
    List<Dancer> output =
connection.Query<Dancer>($"dbo.selectDancersByCategory @DenumireCategorie
=' {categorie} '') .ToList();
    return output;
}
}

// Metoda ce preia data nasterii dansatorilor barbati,
// in functie de categoria selectata
public static List<Dancer> GetMaleDancersByCategory(string categorie)
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
{
    List<Dancer> output =
connection.Query<Dancer>($"dbo.selectMaleDancersByCategory @DenumireCategorie
=' {categorie} '') .ToList();
    return output;
}
}

// Metoda ce ajuta la exportarea intr-un fisier MS Excel lista tuturor
// participantelor cu varsta sub o varsta introdusa
public static List<Dancer> GetFemaleDancersUnderIntroducedYears(string
dataNasterii)
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
{
    List<Dancer> output =
connection.Query<Dancer>($"dbo.selectFemaleDancersUnderIntroducedYears
@DataIntrodusa =' {dataNasterii} ') .ToList();
    return output;
}
}

// Metoda ce creeaza o vedere ce contine date de contact al fiecarui
dansator
public static List<Dancer> CreateTableContactData()
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
{
    List<Dancer> output =
connection.Query<Dancer>("dbo.createView") .ToList();
    return output;
}
}

// Metoda ce preia datele din vedere
public static List<Dancer> GetViewContactData()
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
{

```

```

        List<Dancer> output =
connection.Query<Dancer>("dbo.selectContactData").ToList();
            return output;
        }
    }

    // Metoda ce preia lista participantilor si institutia unde isi fac
studiile
    public static List<Dancer> GetDancersAndInstitution()
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<Dancer> output =
connection.Query<Dancer>("dbo.selectDancersAndInstitution").ToList();
            return output;
        }
    }

    // Metoda ce ce apreia codul, numele, prenumele, locul dansatorului,
// in functie de concursul selectat
    public static List<Dancer> GetDancerPrizeByCompetition(string
denumireConcurs)
    {
        using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
        {
            List<Dancer> output =
connection.Query<Dancer>($"dbo.selectDancerPrizeByCompetition @DenumireConcurs =
'{denumireConcurs}'").ToList();
            return output;
        }
    }

    // Metoda ce preia numele, prenumele dansatorilor, in functie de concurs
public static List<Dancer> GetDancersByCompetition(string competitionName)
{
    using ( IDbConnection connection = new
System.Data.SqlClient.SqlConnection(Helper.CnnVal("ConcursDeDansDB")))
    {
        List<Dancer> dancers =
connection.Query<Dancer>($"dbo.selectDancersByCompetitionName @CompetitionName =
'{competitionName}'").ToList();
        return dancers;
    }
}
}

```

Fișier-ul Helper.cs

```
using System.Configuration;

namespace Concurs_de_dans
{
    public static class Helper
    {
        public static string CnnVal(string name)
        {
            return ConfigurationManager.ConnectionStrings[name].ConnectionString;
        }
    }
}
```

Folder-ul DatabaseTables

Fișier-ul Dancer.cs

```
using System;

namespace Concurs_de_dans
{
    class Dancer
    {
        public string CodDansator { get; }
        public string Nume { get; }
        public string Prenume { get; }
        public string Sex { get; }
        public DateTime DataNasterii { get; }
        public int Experienta { get; }
        public string DenumireTipDans { get; }
        public string Email { get; }
        public string Telefon { get; }
        public string Adresa { get; }
        public string NumeInstitutie { get; }
        public string Locul { get; }

        // Variabila ce returneaza varsta in ani
        public string Varsta
        {
            get
            {
                var today = DateTime.Today;
                var age = today.Year - DataNasterii.Year;
                if (DataNasterii.Date > today.AddYears(-age))
                    age--;

                return $"{age}";
            }
        }

        // Variabila ce returneaza genul de tipul "Femenin"/"Masculin"
        public string GetGen
        {
            get
            {
                if (Sex.Equals("F"))
                    return "Femenin";
                else return "Masculin";
            }
        }
    }
}
```

```

        }
    }

    // Variabila ce returneaza numele si prenumele dansatorului
    public string GetNameSurname
    {
        get {
            return $"{Nume} {Prenume}";
        }
    }

    // Variabila ce returneaza numele, prenumele si varsta dansatorului
    public string GetNameSurnameAndAge
    {
        get {
            return $"{Nume} {Prenume} {Varsta} ani";
        }
    }

    // Variabila ce returneaza codul, numele, prenumele, locul dansatorului
    public string GetCodeNameSurnameAndPlace
    {
        get {
            return $"{Locul}: {CodDansator} {Nume} {Prenume}";
        }
    }
}
}

```

Folder-ul LogIn

Fișierul LogInCaAdministratorForm.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.LogIn
{
    public partial class LogInCaAdministratorForm : Form
    {
        public LogInCaAdministratorForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la meniul de logare
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new LogInForm();

            // Inchidem formularul curent, deschizand formularul de salutare
        }
    }
}

```

```

        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce deschide formularul principal din numele administratorului
    private void logInButton_Click(object sender, EventArgs e)
    {
        List<string> users =
DataAccess.GetAdministratorLogIn(usernameTextBox.Text, passwordTextBox.Text);
        if (users.Count > 0) {
            DataAccess.WhoLogin(usernameTextBox.Text);

            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        } else {
            MessageBox.Show("Username sau parola introdusa gresit!",
"Introducere gresita", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    // Metoda ce afiseaza / ascunde parola, atunci cand este apasat butonul
    private void seePasswordButton_Click(object sender, EventArgs e)
    {
        if (passwordTextBox.PasswordChar == '*') {
            passwordTextBox.PasswordChar = '\0';
        } else {
            passwordTextBox.PasswordChar = '*';
        }
    }
}
}

```

Fișierul LogInForm.cs

```

using System.Windows.Forms;

namespace Concurs_de_dans.LogIn
{
    public partial class LogInForm : Form
    {
        public LogInForm()
        {
            InitializeComponent();
        }

        // Metoda ce afiseaza formularul principal, pentru un oaspete
        private void intraCaOaspeteButton_Click(object sender, System.EventArgs e)
        {
            DataAccess.WhoLogin("Guest");

            this.Hide(); // Ascundem formularul curent
        }
    }
}

```

```

    // Cream o variabila de tip form
    var Form = new ConcursDeDansForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}

// Metoda ce afiseaza formularul de logare a administratorului
private void intraCaAdministratorButton_Click(object sender,
System.EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new LogInCaAdministratorForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}
}
}
}

```

Folder-ul MenuStripForms

Fisierul Categori.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.MenuStripForms
{
    public partial class CategoriForm : Form
    {
        public CategoriForm()
        {
            InitializeComponent();

            // Afisarea denumirilor categoriilor
            List<string> categories = DataAccess.GetCategoryName();
            categoriiListBox.DataSource = categories;
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }
    }
}

```

```
        }
    }
}
```

Fișierul DansatoriForm.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans
{
    public partial class DansatoriForm : Form
    {
        // Variabile ce duc evidenta despre dansatori
        private int contor = 0;
        List<Dancer> dancers = DataAccess.GetDancers();

        public DansatoriForm()
        {
            InitializeComponent();

            // Afisarea datelor primului dansator
            AfisareDansator(contor);
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }

        // Metoda ce afiseaza datele dansatorului precedent
        private void precedentButton_Click(object sender, EventArgs e)
        {
            if (contor > 0) {
                contor--;
                AfisareDansator(contor);
            }
        }

        // Metoda ce afiseaza datele dansatorului urmator
        private void urmatorulButton_Click(object sender, EventArgs e)
        {
            if (contor < dancers.Count - 1) {
                contor++;
                AfisareDansator(contor);
            }
        }

        // Metoda ce afiseaza datele unui dansator
    }
}
```

```

        private void AfisareDansator(int i)
    {
        if (contor < dancers.Count && contor >= 0)
        {
            codDansatorTextBox.Text = dancers[i].CodDansator;
            numeTextBox.Text = dancers[i].Nume;
            prenumeTextBox.Text = dancers[i].Prenume;
            genTextBox.Text = dancers[i].GetGen;
            varstaTextBox.Text = dancers[i].Varsta;
            experientaTextBox.Text = dancers[i].Experienta.ToString();
        }
    }
}

```

Fișierul TipuriDeDansForm.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.MenuStripForms
{
    public partial class TipuriDeDansForm : Form
    {
        public TipuriDeDansForm()
        {
            InitializeComponent();

            // Afisarea denumirilor tipurilor de dans
            List<string> danceTypes = DataAccess.GetDanceType();
            tipuriDeDansListBox.DataSource = danceTypes;
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }
    }
}

```

Folder-ul Tasks

Fișierul CeaMaiTanaraDansatoareForm.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class CeaMaiTanaraDansatoareForm : Form
    {
        List<Dancer> youngestFemaleDancer = DataAccess.GetYoungestFemaleDancer();

        public CeaMaiTanaraDansatoareForm()
        {
            InitializeComponent();

            // Afisarea datelor celei mai tinere dansatoare
            codTextBox.Text = youngestFemaleDancer[0].CodDansator;
            numeTextBox.Text = youngestFemaleDancer[0].Nume;
            prenumeTextBox.Text = youngestFemaleDancer[0].Prenume;
            varstaTextBox.Text = youngestFemaleDancer[0].Varsta;
            genTextBox.Text = youngestFemaleDancer[0].GetGen;
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }
    }
}
```

Fișierul DansatoriCuInstitutiaForm.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class DansatoriCuInstitutiaForm : Form
    {
        // Variabile ce duc evidenta despre dansatori
        private int contor = 0;
        List<Dancer> dancers = DataAccess.GetDancersAndInstitution();
        public DansatoriCuInstitutiaForm()
        {
            InitializeComponent();
        }
    }
}
```

```

        // Afisarea datelor primului dansator
        AfisareDansator(contor);
    }

    // Metoda ce intoarce la formularul principal
    private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Hide(); // Ascundem formularul curent

        // Cream o variabila de tip form
        var Form = new ConcursDeDansForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce afiseaza datele de contact dansatorului urmator
    private void urmatorulButton_Click(object sender, EventArgs e)
    {
        if (contor < dancers.Count - 1) {
            contor++;
            AfisareDansator(contor);
        }
    }

    // Metoda ce afiseaza datele de contact a dansatorului precedent
    private void precedentButton_Click(object sender, EventArgs e)
    {
        if (contor > 0) {
            contor--;
            AfisareDansator(contor);
        }
    }

    // Metoda ce afiseaza datele unui dansator
    private void AfisareDansator(int i)
    {
        if (contor < dancers.Count && contor >= 0) {
            codTextBox.Text = dancers[i].CodDansator;
            numeTextBox.Text = dancers[i].Nume;
            prenumeTextBox.Text = dancers[i].Prenume;
            institutiaTextBox.Text = dancers[i].NumeInstitutie;
        }
    }
}

```

Fișierul DansatoriDupaCategoriiForm.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class DansatoriDupaCategoriiForm : Form
    {
        public DansatoriDupaCategoriiForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }

        // Butonul ce afiseaza numele, prenumele dansatorilor in functie de
        // categoria selectata
        private void afisareButton_Click(object sender, EventArgs e)
        {
            if (denumireCategoriaComboBox.SelectedIndex == -1) {
                MessageBox.Show("Selectati categoria!", "Eroare de introducere",
                    MessageBoxButtons.OK, MessageBoxIcon.Warning);
            } else {
                List<Dancer> dancersByCategory =
                    DataAccess.GetDancersByCategory(denumireCategoriaComboBox.SelectedItem.ToString());

                dansatoriDupaCategoriaListBox.DataSource = dancersByCategory;
                dansatoriDupaCategoriaListBox.DisplayMember = "GetNameSurname";
            }
        }
    }
}
```

Fișierul DansatorTanarBatranForm.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class DansatorTanarBatranForm : Form
    {
        List<Dancer> youngestDancer = DataAccess.GetYoungestDancer();
        List<Dancer> oldestDancer = DataAccess.GetOldestDancer();
```

```

public DansatorTanarBatranForm()
{
    InitializeComponent();

    // Afisarea datelor celui mai tanar dansator
    cod1TextBox.Text = youngestDancer[0].CodDansator;
    nume1TextBox.Text = youngestDancer[0].Nume;
    prenume1TextBox.Text = youngestDancer[0].Prenume;
    tipDans1TextBox.Text = youngestDancer[0].DenumireTipDans;
    varsta1TextBox.Text = youngestDancer[0].Varsta;
}

// Metoda ce intoarce la formularul principal
private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new ConcursDeDansForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}

// Metoda ce afiseaza datele celui mai in varsta dansator
private void celMaiTanarBatranDansatorTabControl_Click(object sender,
EventArgs e)
{
    cod2TextBox.Text = oldestDancer[0].CodDansator;
    nume2TextBox.Text = oldestDancer[0].Nume;
    prenume2TextBox.Text = oldestDancer[0].Prenume;
    tipDans2TextBox.Text = oldestDancer[0].DenumireTipDans;
    varsta2TextBox.Text = oldestDancer[0].Varsta;
}
}
}

```

Fișierul DateDeContactInTabelaForm.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class DateDeContactInTabelaForm : Form
    {
        // Variabile ce duc evidenta despre dansatori
        private int contor = 0;
        List<Dancer> dancers = DataAccess.GetViewContactData();

        public DateDeContactInTabelaForm()
        {
            InitializeComponent();

            DataAccess.CreateTableContactData();
        }
    }
}

```

```

        // Afisarea datelor primului dansator
        AfisareDansator(contor);
    }

    // Metoda ce intoarce la formularul principal
    private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Hide(); // Ascundem formularul curent

        // Cream o variabila de tip form
        var Form = new ConcursDeDansForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce afiseaza datele de contact a dansatorului precedent
    private void precedentButton_Click(object sender, EventArgs e)
    {
        if (contor > 0)
        {
            contor--;
            AfisareDansator(contor);
        }
    }

    // Metoda ce afiseaza datele de contact dansatorului urmator
    private void urmatorulButton_Click(object sender, EventArgs e)
    {
        if (contor < dancers.Count - 1)
        {
            contor++;
            AfisareDansator(contor);
        }
    }

    // Metoda ce afiseaza datele unui dansator
    private void AfisareDansator(int i)
    {
        if (contor < dancers.Count && contor >= 0)
        {
            string telefonData = "";
            string email = "";

            codTextBox.Text = dancers[i].CodDansator;
            numeTextBox.Text = dancers[i].Nume;
            prenumeTextBox.Text = dancers[i].Prenume;
            // Securizam datele prin ascunderea unor caractere din datele de
            contact
            for (int j = 0; j < dancers[i].Telefon.Length; j++) {
                if (j >= 5 && j <= 12) {
                    telefonData += "***";
                } else {
                    telefonData += $"{{dancers[i].Telefon[j]}}";
                }
            }
        }
    }
}

```

```
        for (int j = 0; j < dancers[i].Email.Length; j++)
    {
        if (j < 3) {
            email += ${dancers[i].Email[j]};
        } else if (dancers[i].Email[j].Equals('@')) {
            for (int k = j; k < dancers[i].Email.Length; k++) {
                email += ${dancers[i].Email[k]};
            }
            break;
        } else {
            email += "*";
        }
    }

    telefonTextBox.Text = telefonData;
    emailTextBox.Text = email;
    adresaTextBox.Text = dancers[i].Adresa;
}

}
}
```

Fișierul ExcludereDansatorForm.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class ExcludereDansatorForm : Form
    {
        public ExcludereDansatorForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }

        // Metoda ce afiseaza datele dansatorului care va fi exclus
        private void verificaButton_Click(object sender, EventArgs e)
        {
            List<Dancer> dancers = DataAccess.GetDancersByCode(codTextBox.Text);

            // Verificarea introducerii codului
            if (codTextBox.Text.Length == 0) {
                Clear();
            }
        }
    }
}
```

```

        MessageBox.Show("Introduceti codul!", "Eroare de introducere",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    } else { // Daca codul a fost introdus, se verifica corectitudinea
acestuia
        if (dancers.Count == 1) {
            codTextBox.Text = "";
            codDansatorTextBox.Text = dancers[0].CodDansator;
            numeTextBox.Text = dancers[0].Nume;
            prenumeTextBox.Text = dancers[0].Prenume;

            if (dancers[0].Sex.Equals("F")) {
                genTextBox.Text = "Femenin";
            } else {
                genTextBox.Text = "Masculin";
            }
            varstaTextBox.Text = dancers[0].Varsta;
            experientaTextBox.Text = dancers[0].Experienta.ToString();
        } else { // Daca nu s-a gasit dansator cu asa cod, se afiseaza un
mesaj sugestiv
            Clear();
            MessageBox.Show("Nu s-a gasit dansator cu asa cod!", "Eroare de
cautare", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

// Metoda ce exclude un dansator, conform codului introdus
private void excludeButton_Click(object sender, EventArgs e)
{
    List<Dancer> dancers;

    if (codTextBox.Text.Length != 0) {
        dancers = DataAccess.GetDancersByCode(codTextBox.Text);
    } else {
        dancers = DataAccess.GetDancersByCode(codDansatorTextBox.Text);
    }

    // Verificarea introducerii codului
    if (codTextBox.Text.Length == 0 && codDansatorTextBox.Text.Length == 0)
    {
        Clear();
        MessageBox.Show("Introduceti codul!", "Eroare de introducere",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    } else { // daca codul a fost introdus, se verifica corectitudinea
acestuia
        if (dancers.Count == 1) {
            if (codTextBox.Text.Length != 0) {
                DataAccess.DeleteDancer(codTextBox.Text);
            } else {
                DataAccess.DeleteDancer(codDansatorTextBox.Text);
            }
            Clear();
            MessageBox.Show("Dansatorul a fost exclus cu succes!",
"Excludere cu succes", MessageBoxButtons.OK, MessageBoxIcon.Information);
        } else { // Daca nu s-a gasit dansator cu asa cod, se afiseaza un
mesaj sugestiv
            Clear();
            MessageBox.Show("Nu s-a gasit dansator cu asa cod!", "Eroare de
cautare", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

```
        }
    }
}

// Metoda ce sterge valorile afisate
private void Clear()
{
    codTextBox.Text = "";
    codDansatorTextBox.Text = "";
    numeTextBox.Text = "";
    prenumeTextBox.Text = "";
    genTextBox.Text = "";
    varstaTextBox.Text = "";
    experientaTextBox.Text = "";
}

}
```

Fișierul Exportă Dansatoare Aniform.cs

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Windows.Forms;
using Excel = Microsoft.Office.Interop.Excel;

namespace Concurs_de_dans.Tasks
{
    public partial class ExportaDansatoareAniForm : Form
    {
        public ExportaDansatoareAniForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }
    }

    private void exportaButton_Click(object sender, EventArgs e)
    {
        bool isCorrect = true;
        bool isDigitOnly = true;
        foreach (char c in aniTextBox.Text) {
            if (c < '0' || c > '9') {
                isDigitOnly = false;
                break;
            }
        }
    }
}
```

```

        if (aniTextBox.Text.Length == 0) {
            MessageBox.Show("Introduceti varsta!", "Eroare de introducere",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            isCorrect = false;
        } else if (!isDigitOnly) {
            aniTextBox.Text = "";
            MessageBox.Show("Varsta contine caractere!", "Eroare de
introducere", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            isCorrect = false;
        } else if (Convert.ToInt32(aniTextBox.Text) > 40 || 
Convert.ToInt32(aniTextBox.Text) < 5) {
            aniTextBox.Text = "";
            MessageBox.Show("Varsta introdusa depaseste limitele!", "Eroare de
introducere", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            isCorrect = false;
        }

        if (isCorrect) {
            int an = DateTime.Today.Year - Convert.ToInt32(aniTextBox.Text);
            List<Dancer> dancers =
DataAccess.GetFemaleDancersUnderIntroducedYears($"{an}-01-01");
            string path = Path.GetFullPath(@"..\..\") + "Dansatoare.csv";

            // Introducem datele obtinute intr-un fisier .csv
            using (StreamWriter sw = new StreamWriter(path, false)) {
                string data;
                data = "Cod dansator,Nume,Prenume,Experienta,Varsta,,,,," +
Environment.NewLine;

                for (int i = 0; i < dancers.Count - 1; i++) {
                    data +=
$"{{dancers[i].CodDansator},{dancers[i].Nume},{dancers[i].Prenume},{dancers[i].Experienta},{dancers[i].Varsta},,,,," + Environment.NewLine;
                }
                data += $"{{dancers[dancers.Count -
1].CodDansator},{dancers[dancers.Count - 1].Nume},{dancers[dancers.Count -
1].Prenume},{dancers[dancers.Count - 1].Experienta},{dancers[dancers.Count -
1].Varsta},,,,,";

                // Scrierea datelor in .csv fisier
                sw.Write(data);
            }

            using (SaveFileDialog sfd = new SaveFileDialog() { Filter = "Excel
Workbook (*.xlsx)" }) {
                if (sfd.ShowDialog() == DialogResult.OK) {
                    try {
                        // Preluam datele din fisierul .csv intr-un fisier
.xlsx
                        Excel.Application xl = new Excel.Application();
                        Excel.Workbook wb = xl.Workbooks.Open(path);

                        Excel.Worksheet ws =
(Excel.Worksheet)wb.Worksheets.get_Item(1);
                        // Selectam UsedRange
                        Excel.Range used = ws.UsedRange;
                        // Setarea coloanelor, pentru a corespunde cu
dimensiunea textului

```

Fișierul InregistrareDansatorForm.cs

```
using System;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class InregistrareDansatorForm : Form
    {
        // Variabila ce duce evidenta despre nr. de erori aparute pe parcursul
        // introducerii datelor
        int nrGreseli;

        public InregistrareDansatorForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }

        // Metoda ce reseteaza datele introduse
        private void reseteazaButton_Click(object sender, EventArgs e)
        {
        }
    }
}
```

```

        Clear();
    }

    // Metoda ce sterge valorile introduse
    private void Clear()
    {
        numeTextBox.Text = "";
        prenumeTextBox.Text = "";
        genComboBox.SelectedIndex = -1;
        experientaTextBox.Text = "";
        dataNasteriiDatePicker.Value = DateTime.Now;

        institutiaTextBox.Text = "";
        adresaInstitutieTextBox.Text = "";

        emailTextBox.Text = "";
        telefonTextBox.Text = "";
        adresaDansatorTextBox.Text = "";

        premiuBanescTextBox.Text = "";
        loculComboBox.SelectedIndex = -1;

        tipDansComboBox.SelectedIndex = -1;
        denumireConcursComboBox.SelectedIndex = -1;
        extensiaComboBox.SelectedIndex = -1;
    }

    // Metoda ce inregistreaza un dansator nou
    private void salveazaButton_Click(object sender, EventArgs e)
    {
        Verificare();

        if (nrGreseli == 0)
        {
            string gen;

            if (genComboBox.SelectedIndex == 0) {
                gen = "F";
            } else gen = "M";

            // Introducerea datelor in baza de date
            DataAccess.InsertDancer(institutiaTextBox.Text,
adresaInstitutieTextBox.Text,
tipDansComboBox.SelectedItem.ToString(),
GenerareCodDansator(),
numeTextBox.Text, prenumeTextBox.Text, gen,
dataNasteriiDatePicker.Value.Date.ToString("yyyy-M-dd"),
Convert.ToInt32(experientaTextBox.Text),
denumireConcursComboBox.SelectedItem.ToString(),
emailTextBox.Text +
extensiaComboBox.SelectedItem.ToString(),
$"#373
{telefonTextBox.Text[0]}{telefonTextBox.Text[1]}{telefonTextBox.Text[2]}
{telefonTextBox.Text[3]}{telefonTextBox.Text[4]}
{telefonTextBox.Text[5]}{telefonTextBox.Text[6]}{telefonTextBox.Text[7]}",
adresaDansatorTextBox.Text);

            if (loculComboBox.SelectedIndex != -1) {

```

```

        DataAccess.InsertPrize(loculComboBox.SelectedItem.ToString(),
Convert.ToInt32(premiuBanescTextBox.Text),
denumireConcursComboBox.SelectedItem.ToString());
    }

    Clear();
    MessageBox.Show("Dansatorul a fost introdus cu succes!",
"Introducere cu succes", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
}

// Metoda ce verifica daca au fost introduse corect datele dansatorului
private void Verificare()
{
    string message = "";
    nrGreseli = 0;

    if (numeTextBox.Text.Length == 0) {
        nrGreseli++;
        message += "Introduceti numele!" + Environment.NewLine;
    }
    if (prenumeTextBox.Text.Length == 0) {
        nrGreseli++;
        message += "Introduceti prenumele!" + Environment.NewLine;
    }
    if (genComboBox.SelectedIndex == -1) {
        nrGreseli++;
        message += "Selectati genul!" + Environment.NewLine;
    }

    // Verificarea corectitudinii introducerii varstei si experientei
    var today = DateTime.Today;
    var age = today.Year - dataNasteriiDatePicker.Value.Year;
    if (dataNasteriiDatePicker.Value.Date > today.AddYears(-age))
        age--;

    if (age < 5) {
        nrGreseli++;
        message += "Varsta minima este de 5 ani!" + Environment.NewLine;
    } else if (age > 40) {
        nrGreseli++;
        message += "Varsta maxima este de 40 ani!" + Environment.NewLine;
    }
    if (experientaTextBox.Text.Length == 0) {
        nrGreseli++;
        message += "Introduceti experienta!" + Environment.NewLine;
    } else if (Convert.ToInt32(experientaTextBox.Text) >= age - 5) {
        nrGreseli++;
        message += "Experienta este mai mare decat varsta!" +
Environment.NewLine;
    }

    if (institutiaTextBox.Text.Length == 0) {
        nrGreseli++;
        message += "Introduceti denumirea institutiei!" +
Environment.NewLine;
    }
    if (adresaInstitutieTextBox.Text.Length == 0) {
        nrGreseli++;
    }
}

```

```

        message += "Introduceti adresa institutiei!" + Environment.NewLine;
    }

    // Verificarea corectitudinii introducerii a email
    if (emailTextBox.Text.Length == 0) {
        nrGreseli++;
        message += "Introduceti email!" + Environment.NewLine;
    }
    if (emailTextBox.Text.Contains("@gmail.com") ||
emailTextBox.Text.Contains("@mail.com") ||
emailTextBox.Text.Contains("@yahoo.com")) {
        nrGreseli++;
        message += "Nu introduceti extensia! Selectati-o din lista de
langua!" + Environment.NewLine;
    }
    if (extensiaComboBox.SelectedIndex == -1) {
        nrGreseli++;
        message += "Selectati extensia pentru email!" +
Environment.NewLine;
    }

    // Verificarea corectitudinii introducerii nr. telefon
    bool isDigitOnly = true;
    foreach (char c in telefonTextBox.Text) {
        if (c < '0' || c > '9') {
            isDigitOnly = false;
            break;
        }
    }
    if (telefonTextBox.Text.Length == 0) {
        nrGreseli++;
        message += "Introduceti nr. de telefon!" + Environment.NewLine;
    } else if (!isDigitOnly) {
        nrGreseli++;
        message += "Nr. telefon contine caractere!" + Environment.NewLine;
    } else if (telefonTextBox.Text.Length != 8) {
        nrGreseli++;
        message += "Nr. telefon introdus gresit (ex: 69375842)!" +
Environment.NewLine;
    }
    if (adresaDansatorTextBox.Text.Length == 0) {
        nrGreseli++;
        message += "Introduceti adresa dansatorului!" +
Environment.NewLine;
    }

    if (tipDansComboBox.SelectedIndex == -1) {
        nrGreseli++;
        message += "Selectati tipul de dans practicat de dansator!" +
Environment.NewLine;
    }
    if (denumireConcursComboBox.SelectedIndex == -1) {
        nrGreseli++;
        message += "Selectati concursul la care participa dansatorul!" +
Environment.NewLine;
    }

    // Daca dansatorul are un loc ocupat, verificam corectitudinea
introducerii datelor

```

```

        bool isDigitOnlyForPrize = true;
        foreach (char c in premiuBanescTextBox.Text) {
            if (c < '0' || c > '9') {
                isDigitOnlyForPrize = false;
                break;
            }
        }

        if (loculComboBox.SelectedIndex != -1 &&
premiuBanescTextBox.Text.Length == 0) {
            nrGreseli++;
            message += "Introduceti si premiul banesc!" + Environment.NewLine;
        } else if (loculComboBox.SelectedIndex == -1 &&
premiuBanescTextBox.Text.Length != 0) {
            nrGreseli++;
            message += "Introduceti si locul ocupat!" + Environment.NewLine;
        } else if (!isDigitOnlyForPrize) {
            nrGreseli++;
            message += "Premiul banesc contine caractere!" +
Environment.NewLine;
        }

        if (nrGreseli != 0)
        {
            MessageBox.Show(message, "Atentie", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
        }
    }

    // Metoda ce genereaza codul unui dansator
    private string GenerareCodDansator()
    {
        Random random = new Random();
        int randomNumber = random.Next(10, 99);
        string cod;

        cod =
$"{numeTextBox.Text[0]}{prenumeTextBox.Text[0]}{telefonTextBox.Text[6]}{telefonText
Box.Text[7]}{randomNumber}";
        return cod;
    }
}
}

```

Fișierul ListaConcursurilorCuDansatoriForm.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class ListaConcursurilorCuDansatoriForm : Form
    {
        public ListaConcursurilorCuDansatoriForm()
        {
            InitializeComponent();
        }
    }
}

```

```

        // Controlerul ce preia denumirile concursurilor
        concursuriListBox.DataSource = DataAccess.GetCompetitionName();
    }

    // Metoda ce intoarce la formularul principal
    private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Hide(); // Ascundem formularul curent

        // Cream o variabila de tip form
        var Form = new ConcursDeDansForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Butonul ce afiseaza participatnii, in functie de concursul ales
    private void afiseazaParticipantiButton_Click(object sender, EventArgs e)
    {
        List<Dancer> dancers =
DataAccess.GetDancersByCompetition(concursuriListBox.SelectedItem.ToString());

        participantiListBox.DataSource = dancers;
        participantiListBox.DisplayMember = "GetNameSurname";
    }
}
}

```

Fișierul OrdonareAscendentDupaVarstaForm.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class OrdonareAscendentDupaVarstaForm : Form
    {
        public OrdonareAscendentDupaVarstaForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }

        // Metoda ce afiseaza datele dansatorilor, sortati ascendent dupa varsta,
    }
}

```

```

// in functie de concurs
private void afisareButton_Click(object sender, EventArgs e)
{
    List<Dancer> sortedDancers;

    if (denumireConcursComboBox.SelectedIndex == -1) {
        MessageBox.Show("Selectati concursul dupa care se vor sorta
dansatorii!", "Eroare de introducere", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
    } else {
        if (denumireConcursComboBox.SelectedIndex == 0) {
            sortedDancers = DataAccess.GetSortedAscDancersByDate();
            dansatoriASCLListBox.DataSource = sortedDancers;
        } else {
            sortedDancers =
DataAccess.GetSortedDancersAscFromOneCompetitionByDate(denumireConcursComboBox.Sele
ctedItem.ToString());
            dansatoriASCLListBox.DataSource = sortedDancers;
        }
        dansatoriASCLListBox.DisplayMember = "GetNameSurnameAndAge";
    }
}
}
}

```

Fișierul PremiiForm.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans
{
    public partial class PremiiForm : Form
    {
        public PremiiForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }

        // Metoda ce afiseaza datele dansatorilor, sortati ascendent dupa varsta,
        // in functie de concurs
        private void afisareButton_Click(object sender, EventArgs e)
        {
            List<Dancer> dancersPrises;

```

```
        if (denumireConcursComboBox.SelectedIndex == -1) {
            MessageBox.Show("Selectati concursul dupa care vor fi afisate premiile!", "Eroare de introducere", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        } else {
            dancersPrises =
DataAccess.GetDancerPrizeByCompetition(denumireConcursComboBox.SelectedItem.ToString());
            premiiConcursListBox.DataSource = dancersPrises;
            premiiConcursListBox.DisplayMember = "GetCodeNameSurnameAndPlace";
        }
    }
}
```

Fisierul VarstaMedieABarbatilorForm.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans.Tasks
{
    public partial class VarstaMedieABarbatilorForm : Form
    {
        public VarstaMedieABarbatilorForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la formularul principal
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new ConcursDeDansForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }

        // Metoda ce afiseaza varsta medie a barbatilor dintr-o categorie
        a selectata
        private void afisareButton_Click(object sender, EventArgs e)
        {
            double varstaMedie = 0;
            varstaMedieCircularProgressBar.Maximum = 40;

            if (denumireCategorieComboBox.SelectedIndex == -1) {
                MessageBox.Show("Selectati categoria!", "Eroare de introducere",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
            } else {
                List<Dancer> maleDancers =
                DataAccess.GetMaleDancersByCategory(denumireCategorieComboBox.SelectedItem.ToString
                ());
            }
        }
    }
}

```

```
        varstaMedie += Convert.ToDouble(i.Varsta);
    }

    varstaMedie /= maleDancers.Count;
    if (varstaMedie > 0) {
        aniLabel.Text = Math.Round(varstaMedie, 0).ToString() + " ani";
        varstaMedieCircularProgressBar.Value =
Convert.ToInt32(varstaMedie);
    } else {
        aniLabel.Text = "0 ani";
        varstaMedieCircularProgressBar.Value = Convert.ToInt32(0);
    }
}
}
```

Fișierul ConcursDeDansForm.cs

```
using Concurs_de_dans.LogIn;
using Concurs_de_dans.MenuStripForms;
using Concurs_de_dans.Tasks;
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace Concurs_de_dans
{
    public partial class ConcursDeDansForm : Form
    {
        // Variabila ce determina cine a s-a logat
        List<string> whoEntered = DataAccess.ViewLogin();

        public ConcursDeDansForm()
        {
            InitializeComponent();
        }

        // Metoda ce intoarce la meniul de logare
        private void inapoiToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new LogInForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }

        // Metoda ce afiseaza formularul cu dansatori
        private void dansatoriToolStripMenuItem_Click(object sender, EventArgs e)
        {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
```

```

        var Form = new DansatoriForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce afiseaza formularul cu categorii
    private void categoriiToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Hide(); // Ascundem formularul curent

        // Cream o variabila de tip form
        var Form = new CategoriForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce afiseaza formularul cu tipuri de dans
    private void tipDeDansuriToolStripMenuItem_Click(object sender, EventArgs e)
    {
        this.Hide(); // Ascundem formularul curent

        // Cream o variabila de tip form
        var Form = new TipuriDeDansForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce afiseaza formularul cu inregistrare
    private void inregistreazaButton_Click(object sender, EventArgs e)
    {
        if (whoEntered[0].Equals("False")) {
            MessageBox.Show("Nu aveti destule drepturi pentru a insera un dansator nou!", "Oaspete", MessageBoxButtons.OK, MessageBoxIcon.Error);
        } else {
            this.Hide(); // Ascundem formularul curent

            // Cream o variabila de tip form
            var Form = new InregistrareDansatorForm();

            // Inchidem formularul curent, deschizand formularul de salutare
            Form.Closed += (s, args) => this.Close();
            Form.Show();
        }
    }

    // Metoda ce afiseaza formularul cu excludere
    private void excludeButton_Click(object sender, EventArgs e)
    {
        if (whoEntered[0].Equals("False")) {
            MessageBox.Show("Nu aveti destule drepturi pentru a exclude un dansator!", "Oaspete", MessageBoxButtons.OK, MessageBoxIcon.Error);
        } else {

```

```

        this.Hide(); // Ascundem formularul curent

        // Cream o variabila de tip form
        var Form = new ExcludereDansatorForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }
}

// Metoda ce afiseaza formularul cu dansatorul cl mai tanar / in varsta,
// si ce tip de dans practica
private void selecteazaCelMaiInVarstaTanarButton_Click(object sender,
EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new DansatorTanarBatranForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}

// Metoda ce afiseaza formularul dansatori ordonati ascendent dupa varsta
private void ordonareDupaVarsaButton_Click(object sender, EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new OrdonareAscendentDupaVarstaForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}

// Metoda ce afiseaza formularul cu cea mai tanara dansatoare
private void ceaMaiTanaraFemeieButton_Click(object sender, EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new CeaMaiTanaraDansatoareForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}

// Metoda ce afiseaza formularul cu dansatori, in functie de categoria
selectata
private void dansatoriDupaCategorieButton_Click(object sender, EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form

```

```

        var Form = new DansatoriDupaCategoriiform();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce afiseaza formularul cu varsta medie a barbatilor
    private void varstamediaBarbatiCategorieButton_Click(object sender,
EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new VarstaMedieABarbatilorForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}

// Metoda ce afiseaza formularul care exporta dansatoarele sub un anumit nr
de ani
private void exportaButton_Click(object sender, EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new ExportaDansatoareAniForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}

// Metoda ce afiseaza formularul care creeaza o vedere,
// ce contine date de contact a dansatorilor
private void dateDeContactButton_Click(object sender, EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new DateDeContactInTabelaForm();

    // Inchidem formularul curent, deschizand formularul de salutare
    Form.Closed += (s, args) => this.Close();
    Form.Show();
}

// Metoda ce afiseaza formularul cu dansatorii si institutia unde isi fac
studiile
private void participantiStudiileButton_Click(object sender, EventArgs e)
{
    this.Hide(); // Ascundem formularul curent

    // Cream o variabila de tip form
    var Form = new DansatoriCuInstitutiaForm();

    // Inchidem formularul curent, deschizand formularul de salutare

```

```

        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce afiseaza formularul cu premii
    private void afisarepremiiButton_Click(object sender, EventArgs e)
    {
        this.Hide(); // Ascundem formularul curent

        // Cream o variabila de tip form
        var Form = new PremiiForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }

    // Metoda ce afiseaza formularul cu concursuri si dansatorii ce participa
    la acesta
    private void ConcursuriParticipantiButton_Click(object sender, EventArgs e)
    {
        this.Hide(); // Ascundem formularul curent

        // Cream o variabila de tip form
        var Form = new ListaConcursurilorCuDansatoriForm();

        // Inchidem formularul curent, deschizand formularul de salutare
        Form.Closed += (s, args) => this.Close();
        Form.Show();
    }
}
}

```

Fișierul Program.cs

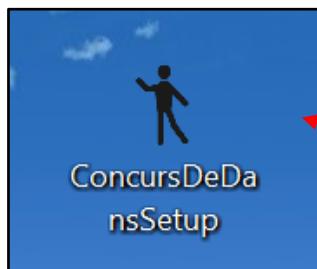
```

using Concurs_de_dans.LogIn;
using System;
using System.Windows.Forms;

namespace Concurs_de_dans
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new LogInForm());
        }
    }
}

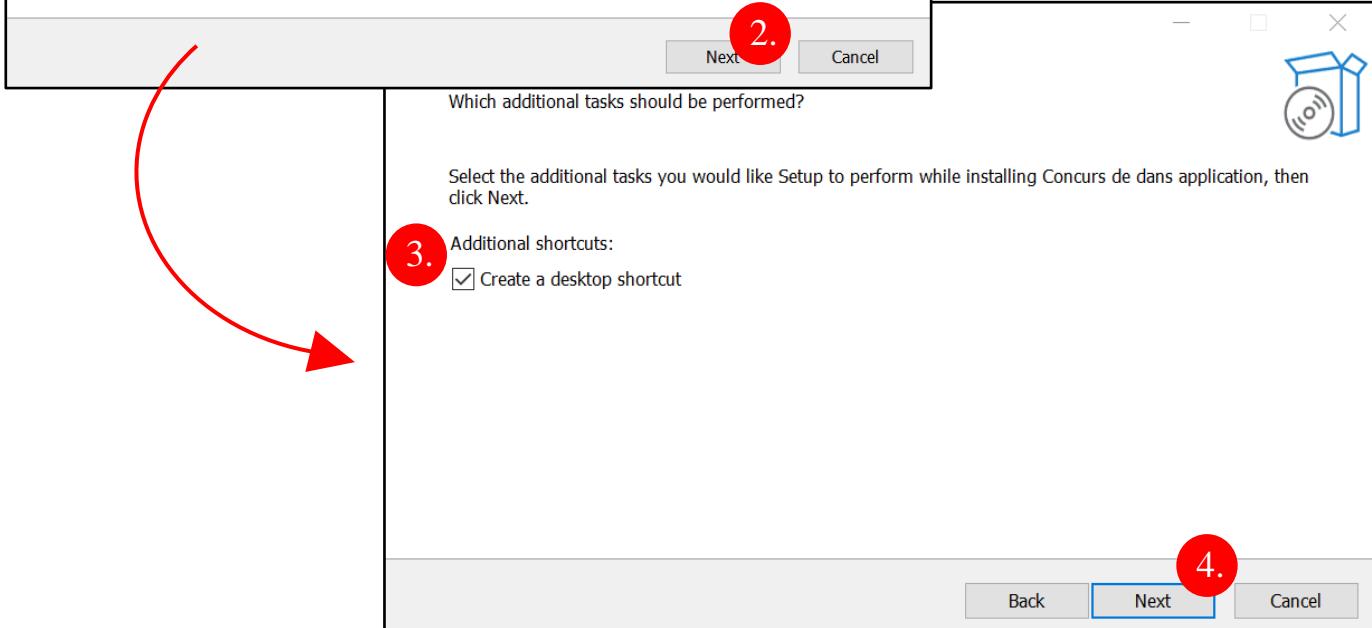
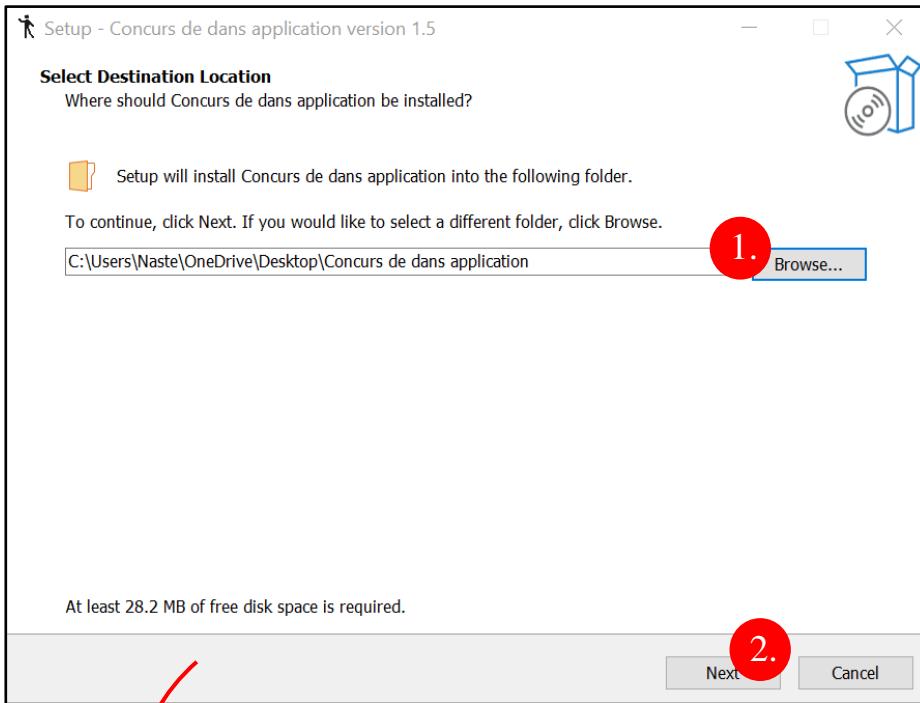
```

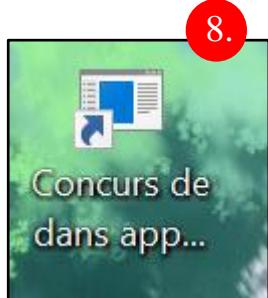
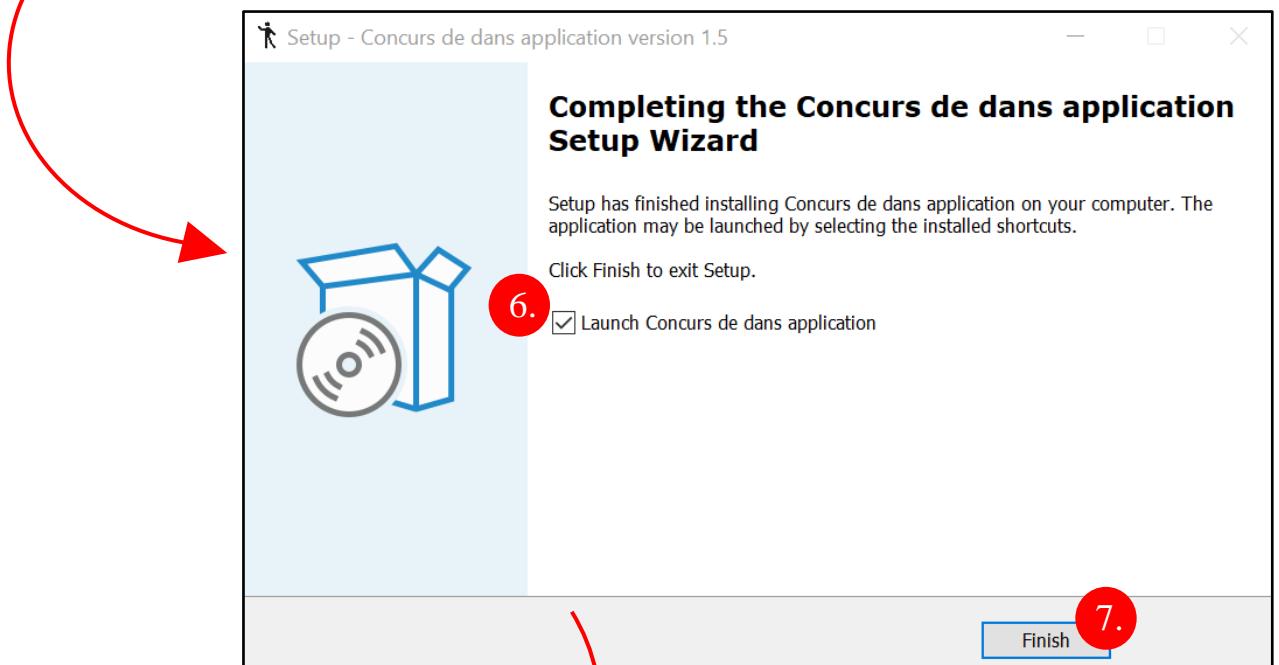
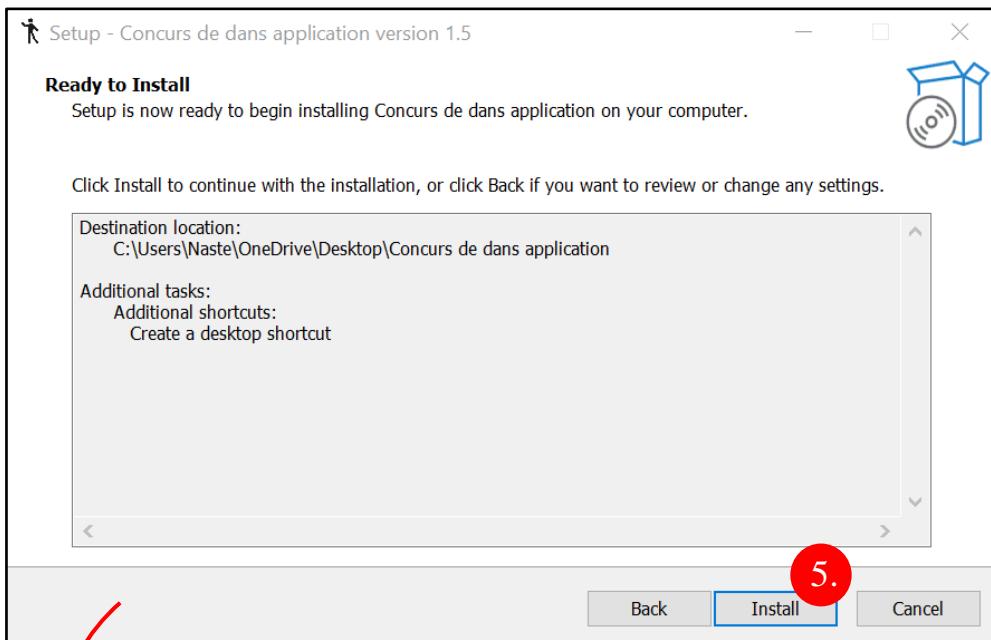
Prezentarea generală a aplicației



Pentru a instala aplicația, rulăm fișierul ConcursDeDansSetup.exe

După, acceptăm instalarea. În continuare, urmărim pașii ce urmează:





După ce instalăm și rulăm aplicația, ne se va deschide interfața de logare:

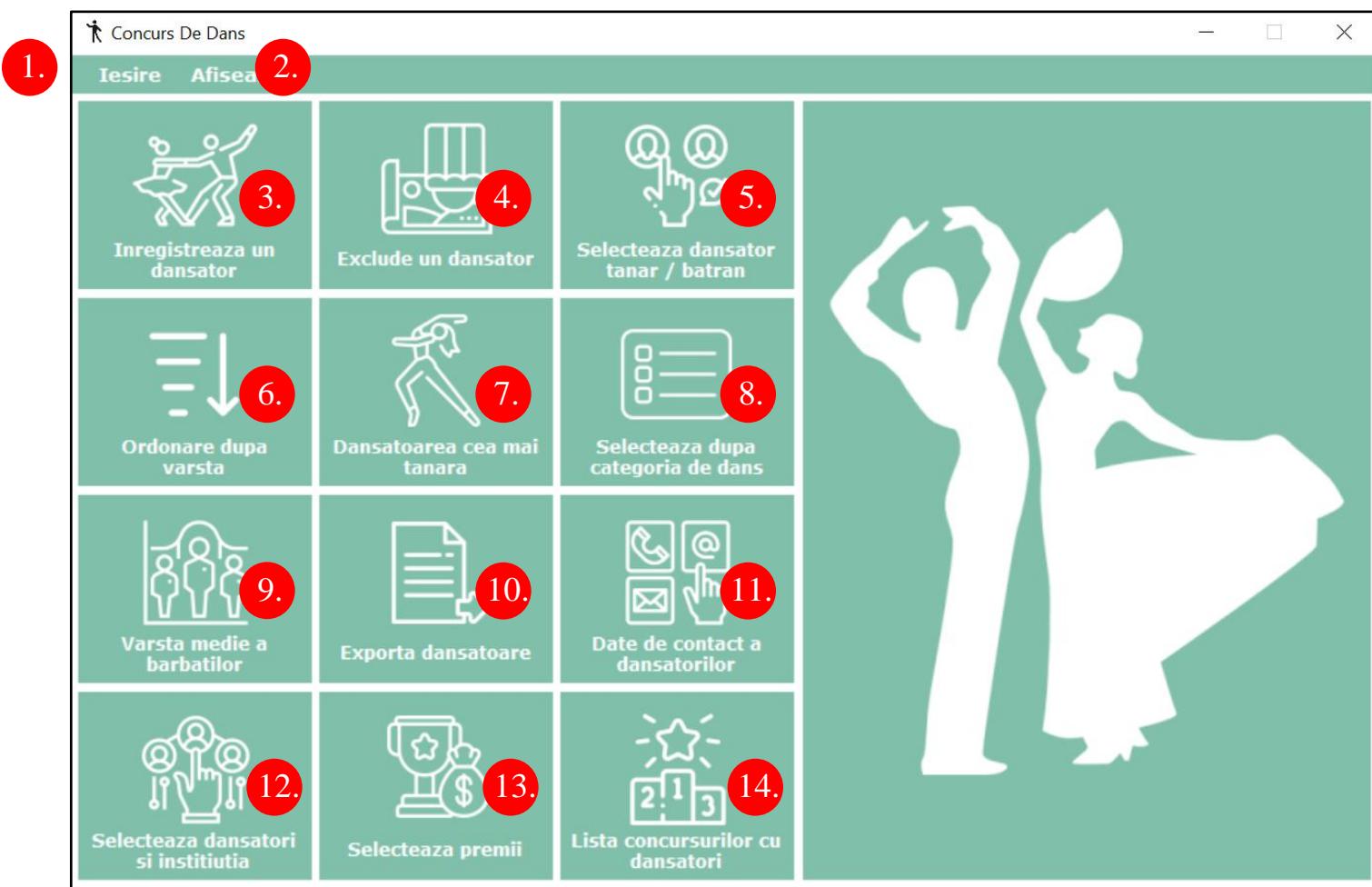


Interfața de logare



1. Logare în calitate de oaspete;
2. Logare în calitate de administrator.

Interfața meniului principal al aplicației

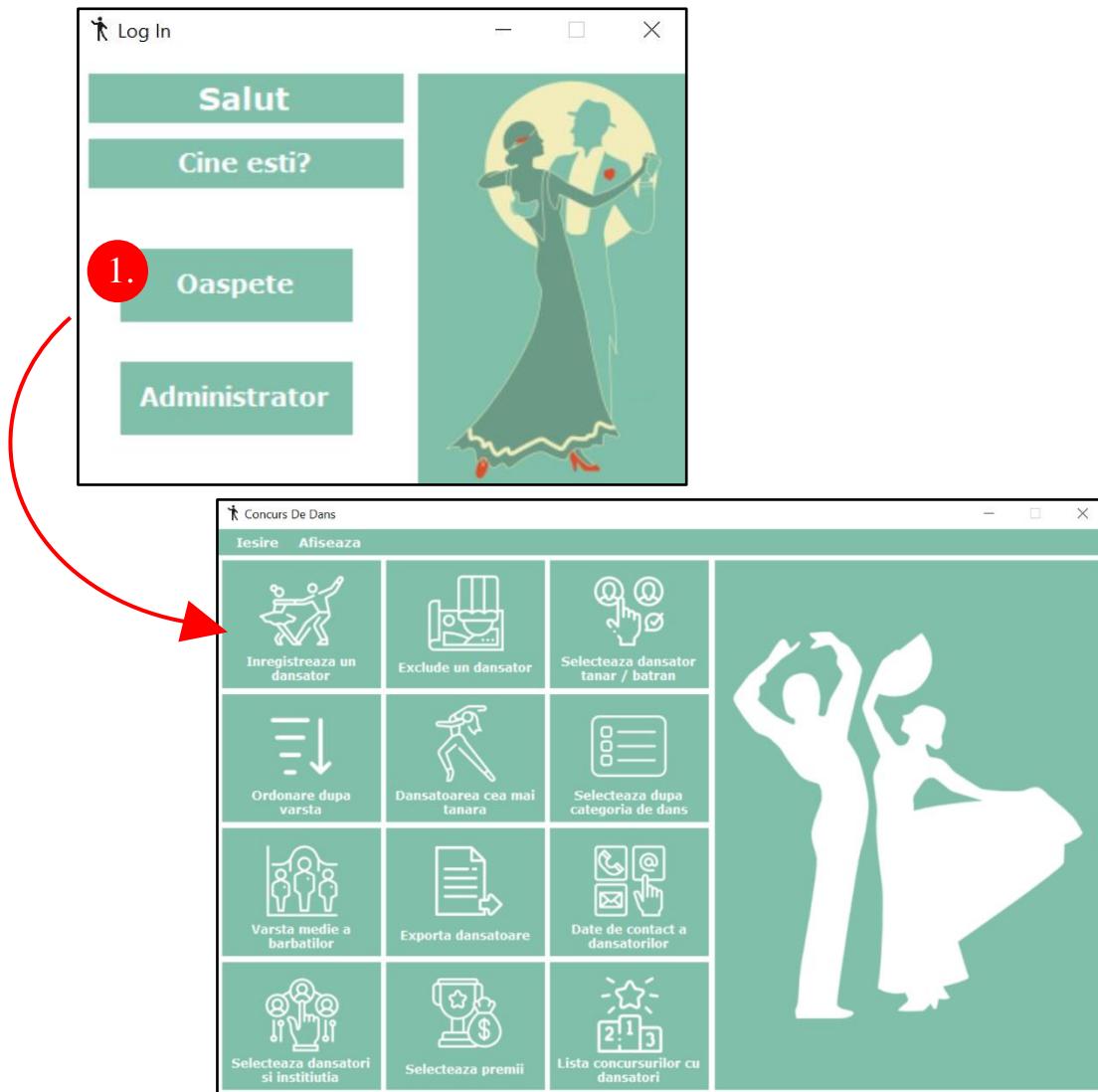


1. Revenirea la meniul de logare;
2. Afisarea datelor cele mai importante din baza de date;
3. Înregistrează un nou participant;
4. Exclude înregistrarea, ce corespunde participantului cu codul introdus de la tastatură
5. Determină participantul cel mai în vîrstă și cel mai Tânăr, și ce tip de dans practică;
6. Afisează la ecran lista tuturor participanților, ordonată ascendent conform vîrstei acestora;
7. Determină numele celei mai tinere persoane de sex feminin;
8. Afisează lista participanților la categoria de dans indicată;
9. Determină și afisează vîrsta medie a bărbaților pentru fiecare categorie de dans;
10. Exportă într-un fișier MS Word sau MS Excel lista tuturor participantelor cu vîrsta sub 12 ani;
11. Creează o tabelă ce va conține date de contact al fiecărui participant;
12. Afisează lista participanților și instituția unde își fac studiile;

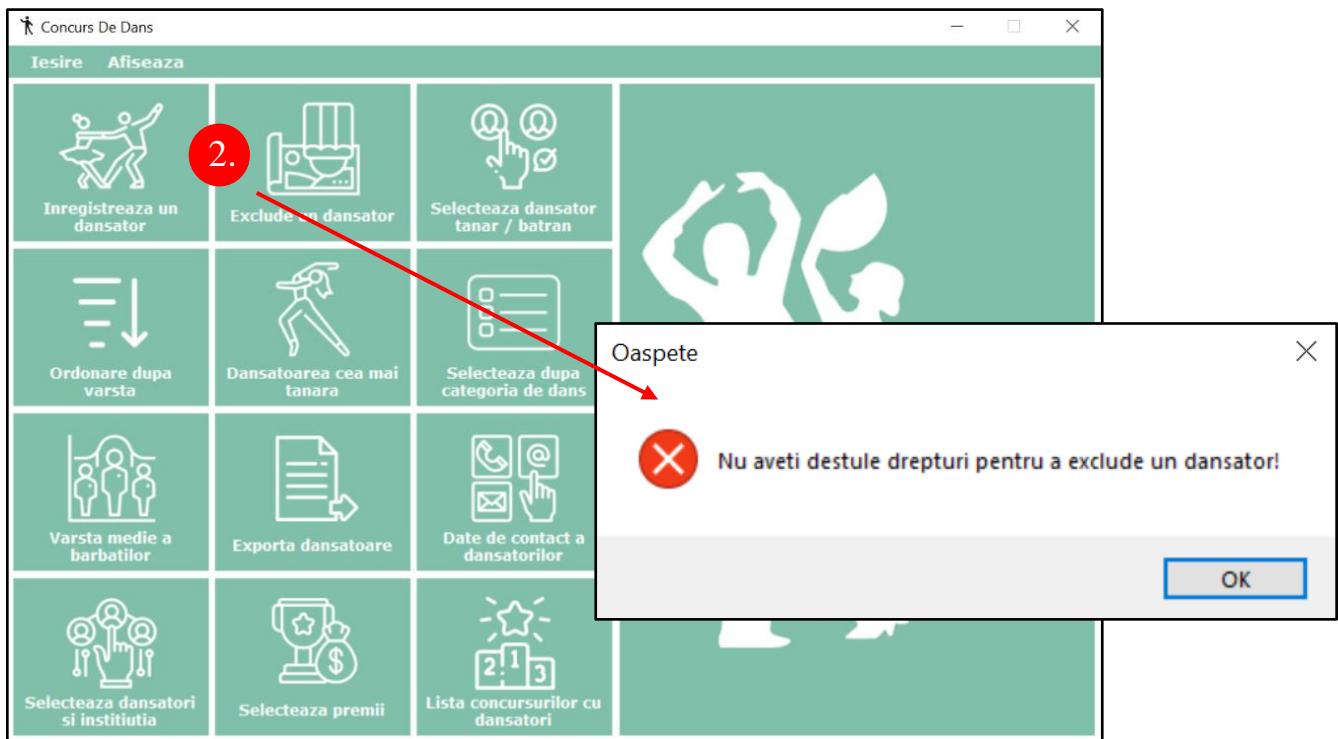
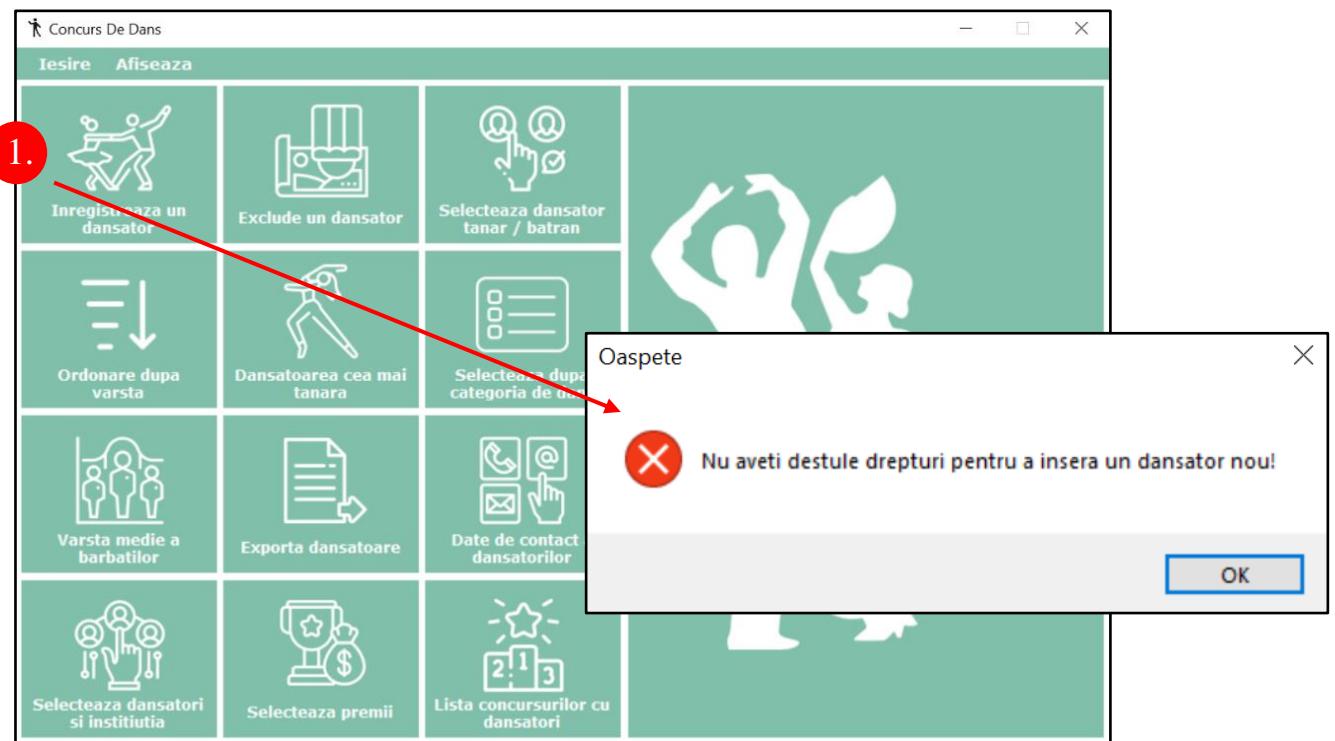
13. Afisează lista premiilor luate, codul, numele și prenumele participanților, în funcție de concursul selectat;
14. Afisează lista dansatorilor, în funcție de concursul selectat.

Rezultatele testării aplicației

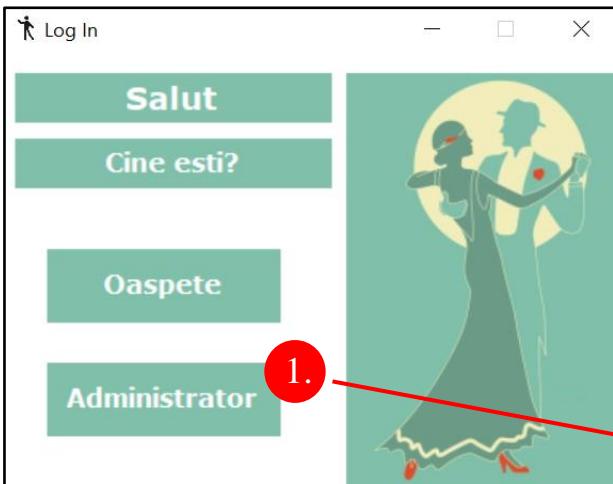
Logarea în calitate de utilizator obișnuit



Atunci când ne logăm în calitate de un utilizator obișnuit, unele acțiuni sunt blocate, precum înregistrarea unui dansator nou și ștergerea unui prezent, pentru a asigura securitatea bazelor de date:



Logarea în calitate de administrator



Revenire la meniul de logare

2.

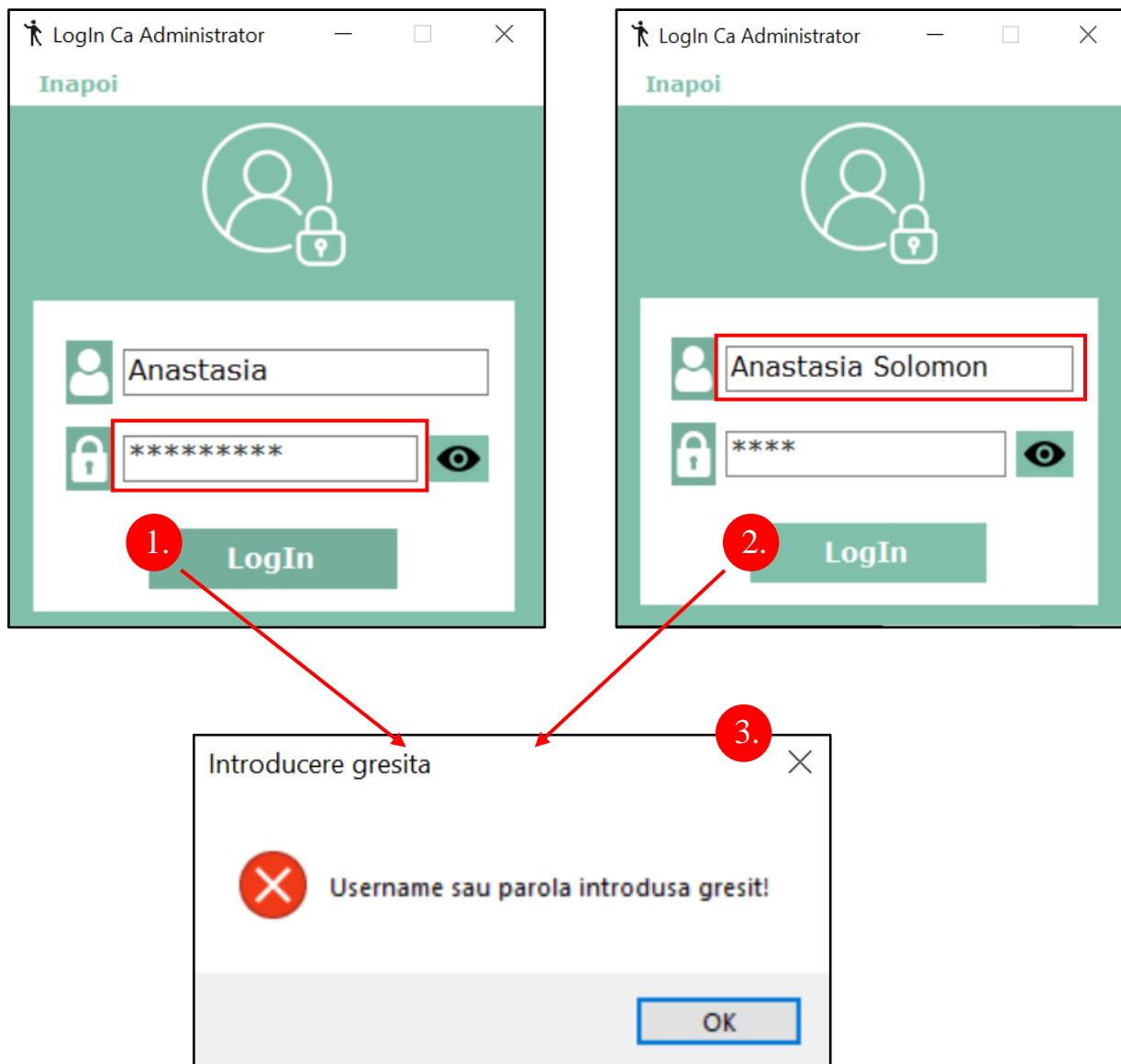
3.

4.

În caz că dorim să vedem parola, acest buton ne vine în ajutor

În cazul în care numele utilizatorului și parola este introdusă corect, va apărea meniul principal, la fel ca și în cazul unui oaspete, însă cu posibilitatea introducerii și excluderii unui dansator.

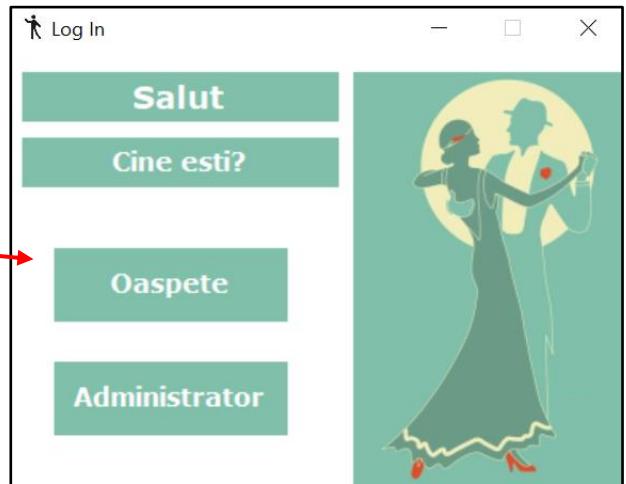
Erori întâlnite în cadrul logării calitate de administrator



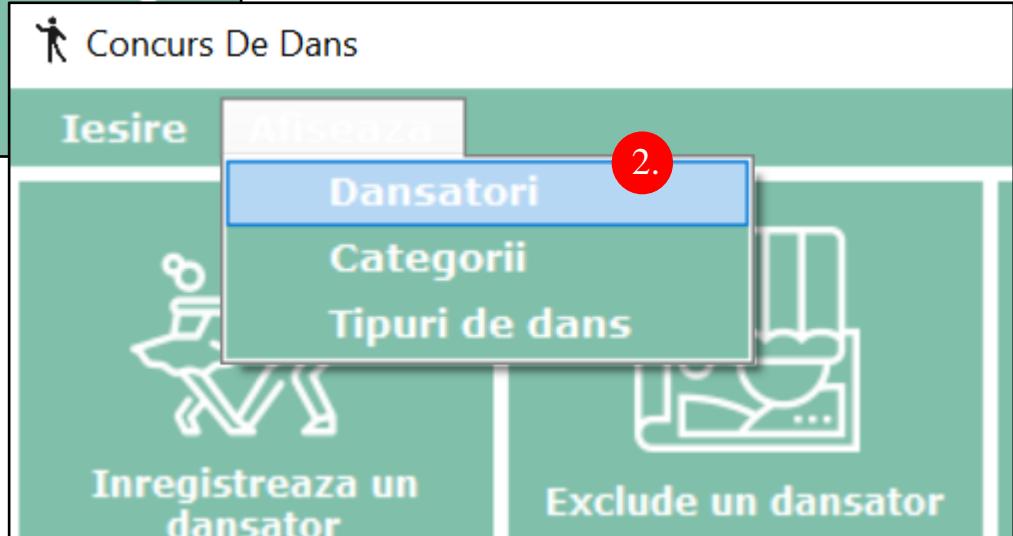
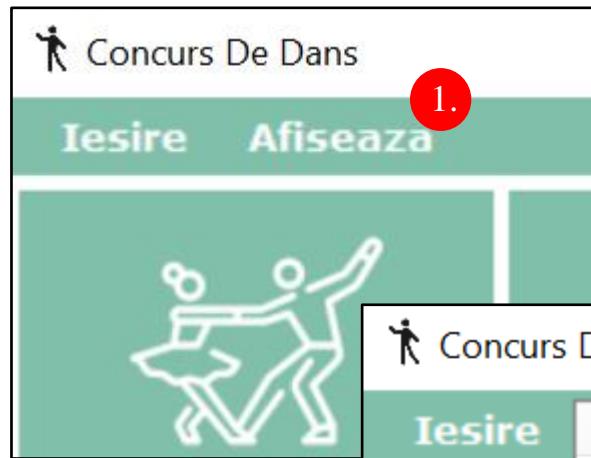
Meniul principal

Bara de navigare

Ieșire



Afișarea dansatorilor



Revenire la meniul principal

3.

Dansatori

Inapoi

Cod	MR3716	Gen	Femenin
Nume	Mccallum	Varsta	7 Ani
Prenume	Ryley	Experienta	2 Ani

Precedent Urmatorul



4.

Dansatori

Inapoi

Cod	DD1783	Gen	Femenin
Nume	Davidson	Varsta	7 Ani
Prenume	Darcie-Mae	Experienta	1 Ani

Precedent Urmatorul



Afișarea dansatorului precedent

Afișarea dansatorului următor

5.

Dansatori

Inapoi

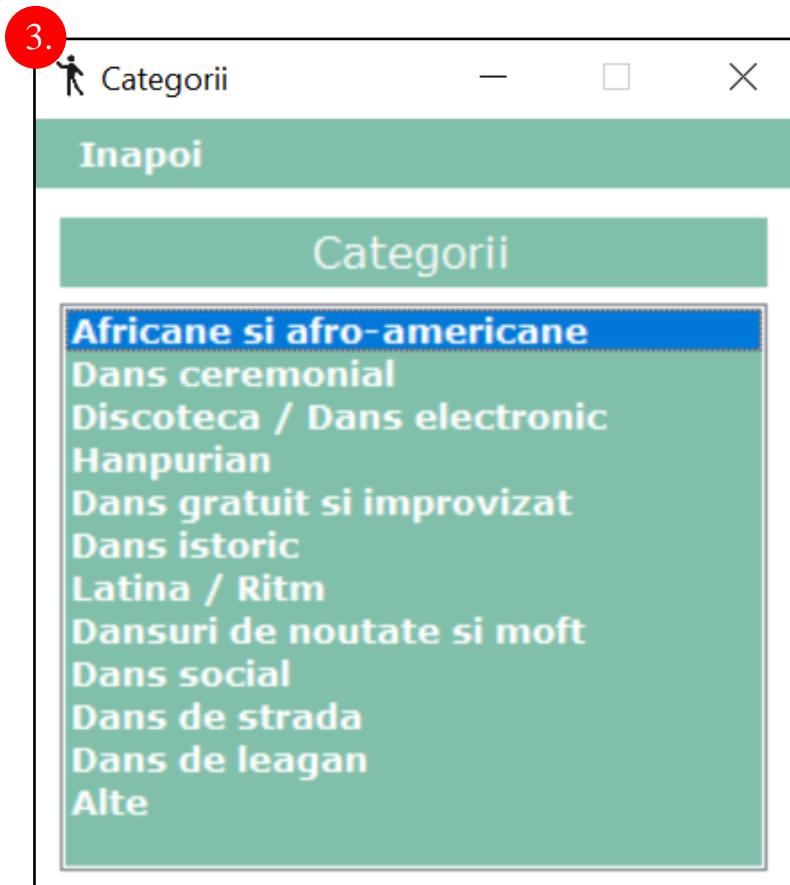
Cod	MR3716	Gen	Femenin
Nume	Mccallum	Varsta	7 Ani
Prenume	Ryley	Experienta	2 Ani

Precedent Urmatorul



Afișarea categoriilor

Revenire la meniu principal



Afișarea tipurilor de dans

Revenir la meniu principal



This screenshot shows a list of dance types. At the top, there is a header 'Tipuri De Dans' with a back button and a close button. Below this is a section titled 'Tipuri de dansuri' containing a list of dance names. The first item in the list, 'Bolojo', is highlighted with a blue selection bar. A red circle with the number '3.' is positioned above the 'Tipuri De Dans' header. The list includes the following items:

- Bolojo
- Boge
- Pas cu pas
- Kompa
- Dans de jazz
- Haka
- Kagura
- Dansuri rituale din China
- Dans Cham
- Drametse Ngacham
- Dans Kumha Pyakhan
- Dans profetic
- Boogaloo

Înregistrarea unui dansator nou

Pentru a înregistra un dansator nou, ne logăm în calitate de administrator (vezi la subiectul „logarea ca administrator”). După urmărim pașii:

The image shows two screenshots of a software application for managing dance contests. The top screenshot is a main menu with a red box around the 'Inregistreaza un dansator' button, which is step 1. A red arrow points down to the second screenshot, which is a detailed registration form. The second screenshot has numbered circles (2 through 19) corresponding to the steps below.

Step	Field / Action
1.	Inregistreaza un dansator (button)
2.	Inapoi (button)
3.	Nume (text input)
4.	Prenume (text input)
5.	Genul (dropdown menu)
6.	Tipul dansului practicat (dropdown menu)
7.	Data nasterii (date input)
8.	Experienta (text input)
9.	Institutia (text input)
10.	Adresa (text input)
11.	Email (text input)
12.	Extensia (text input)
13.	Telefon (+373) (text input)
14.	Adresa dansatorului (text input)
15.	Denumirea (dropdown menu)
16.	Locul (dropdown menu)
17.	Premiu banesc (text input)
18.	Reseteaza (button)
19.	Salveaza (button)

2. Revenire la meniul principal;
3. Numele dansatorului nou;
4. Prenumele dansatorului nou;

5. Genul dansatorului (selectat dintr-o listă);
6. Tipul de dans practicat de dansator (selectat dintr-o listă);
7. Data nașterii a dansatorului (selectată dintr-o listă sau indicată manual);
8. Experiența acumulată de către dansator;
9. Instituția de învățământ în care învață curent (sau pe care a absolvit);
10. Adresa instituției;
11. Email a dansatorului fără extensie;
12. Extensia pentru email (selectată dintr-o listă) ;
13. Numărul de telefon a dansatorului;
14. Adresa dansatorului nou;
15. Denumirea concursului la care participă (selectată dintr-o listă);
16. Locul ocupat în concurs – optional (selectat dintr-o listă) ;
17. Premiul bănesc primit de către acest dansator.

Introducem datele:

Inregistrare Dansator

Inapoi

Dansator

Nume Solomon	Prenume Anastasia
Genul Femenin	Tipul dansului practicat Flamenco

Institutia de invatamant

Institutia CEITI	Data nasterei 3/ 4/2003
Adresa Chisinau	Experienta 8 Ani

Date de contact

Email anastasia	Extensia @gmail.com
Telefon +373 60243654	
Adresa dansatorului Chisinau	

Concurs

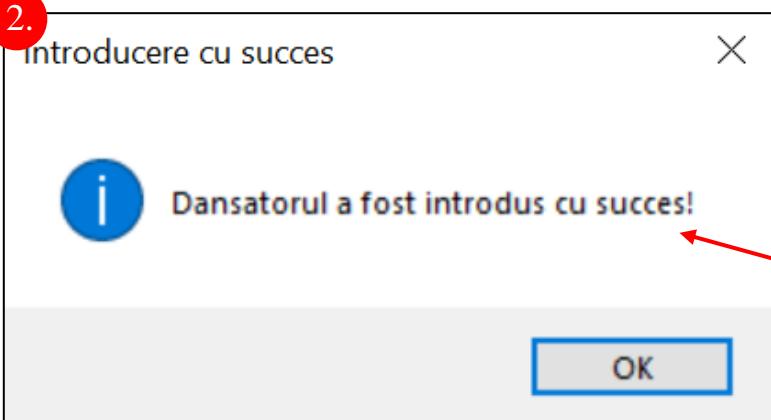
Denumirea Fantezie si Talent
--

Premiu (optional)

Locul I	Premiu banesc 2000 Lei
-------------------	----------------------------------

Reseteaza **Salveaza** 1.

2.



În cazul în care datele sunt introduse corect, se va afișa un mesaj sugestiv

Baza de date înainte de înregistrarea unui dansator nou:

CodDansator	Nume	Prenume	Sex	DataNasterii	Experienta	Email	Telefon
PA9176	Piper	Antony	M	1983-06-26	32	aibrahim@gmail.com	+373 562 28 191
HB4085	Hanna	Brooklyn	M	1986-01-17	28	maratb@gmail.com	+373 788 20 440
SC4892	Shah	Chenai	M	1987-11-04	26	tubesteak@gmail.com	+373 562 05 948

Adresa	NumelInstitutie	Adresa	DenumireTipDans	DenumireCategorie
Balti	Universitatea Pedagogica de Stat "Ion Creanga"	Tiraspol	Kagura	Dans ceremonial
Balti	Universitatea de Stat "Grigore T. Popovici"	Balti	Dans baroc	Dans istoric
Chisinau	Institutul Internațional de Management "IMI-NOVA"	Tiraspol	Dansuri rituale din China	Dans ceremonial

Locul	PremiuBanesc	Nume	Prenume
Mentiune	500.00	Elliott	Cairon
Mentiune	100.00	Holman	Greg
Mentiune	1000.00	Neale	Eduard

Baza de date după înregistrarea unui dansator nou:

CodDansator	Nume	Prenume	Sex	DataNasterii	Experienta	Email	Telefon
PA9176	Piper	Antony	M	1983-06-26	32	aibrahim@gmail.com	+373 562 28 191
HB4085	Hanna	Brooklyn	M	1986-01-17	28	maratb@gmail.com	+373 788 20 440
SC4892	Shah	Chenai	M	1987-11-04	26	tubesteak@gmail.com	+373 562 05 948
SA5470	Solomon	Anastasia	F	2003-03-04	8	anastasia@gmail.com	+373 602 43 654

Adresa	NumeInstitutie	Adresa	DenumireTipDans	DenumireCategorie
Balti	Universitatea Pedagogica de Stat "Ion Creanga"	Tiraspol	Kagura	Dans ceremonial
Balti	Universitatea de Stat "Grigore Tocmescu"	Balti	Dans baroc	Dans istoric
Chisinau	Institutul Internațional de Management "IMI-NOVA"	Tiraspol	Dansuri rituale din China	Dans ceremonial
Chisinau	CEITI	Chisinau	Flamenco	Alte

Locul	PremiuBanesc	Nume	Prenume
Mentiune	500.00	Elliott	Cairon
Mentiune	100.00	Holman	Greg
Mentiune	1000.00	Neale	Eduard
I	2000.00	Solomon	Anastasia

În caz că ne-am răzgândit să înregistram datele dansatorului curent, putem reseta toate câmpurile.

1.

2.

Erori întâlnite în cadrul înregistrării unui dansator nou

Pentru a demonstra toate erorile împreună, nu introducem nici un câmp, însă dacă se va omite un câmp (unele câmpuri), lista va fi diferită. De asemenea, sunt tratate erorile de format (de ex nu putem introduce caracter în experiență, nr. de telefon etc.).

The screenshot shows a registration form for a dancer ('Dansator') with several fields highlighted in red, indicating validation errors. A red circle with the number '1.' is positioned in the bottom right corner of the form area.

Dansator

- Nume**: Red box around the input field.
- Prenume**: Red box around the input field.
- Genul**: Red box around the dropdown menu.
- Tipul dansului practicat**: Red box around the dropdown menu.
- Institutia de invatamant**

 - Institutia**: Red box around the input field.
 - Data nasterii**: Red box around the date input field containing '5/24/2022'.
 - Experienta**: Red box around the input field.
 - Ani**: Red box around the dropdown menu.

Date de contact

 - Email**: Red box around the input field.
 - Extensia**: Red box around the dropdown menu.
 - Telefon**: Red box around the input field.
 - +373**: Red box around the country code.
 - Adresa dansatorului**: Red box around the input field.

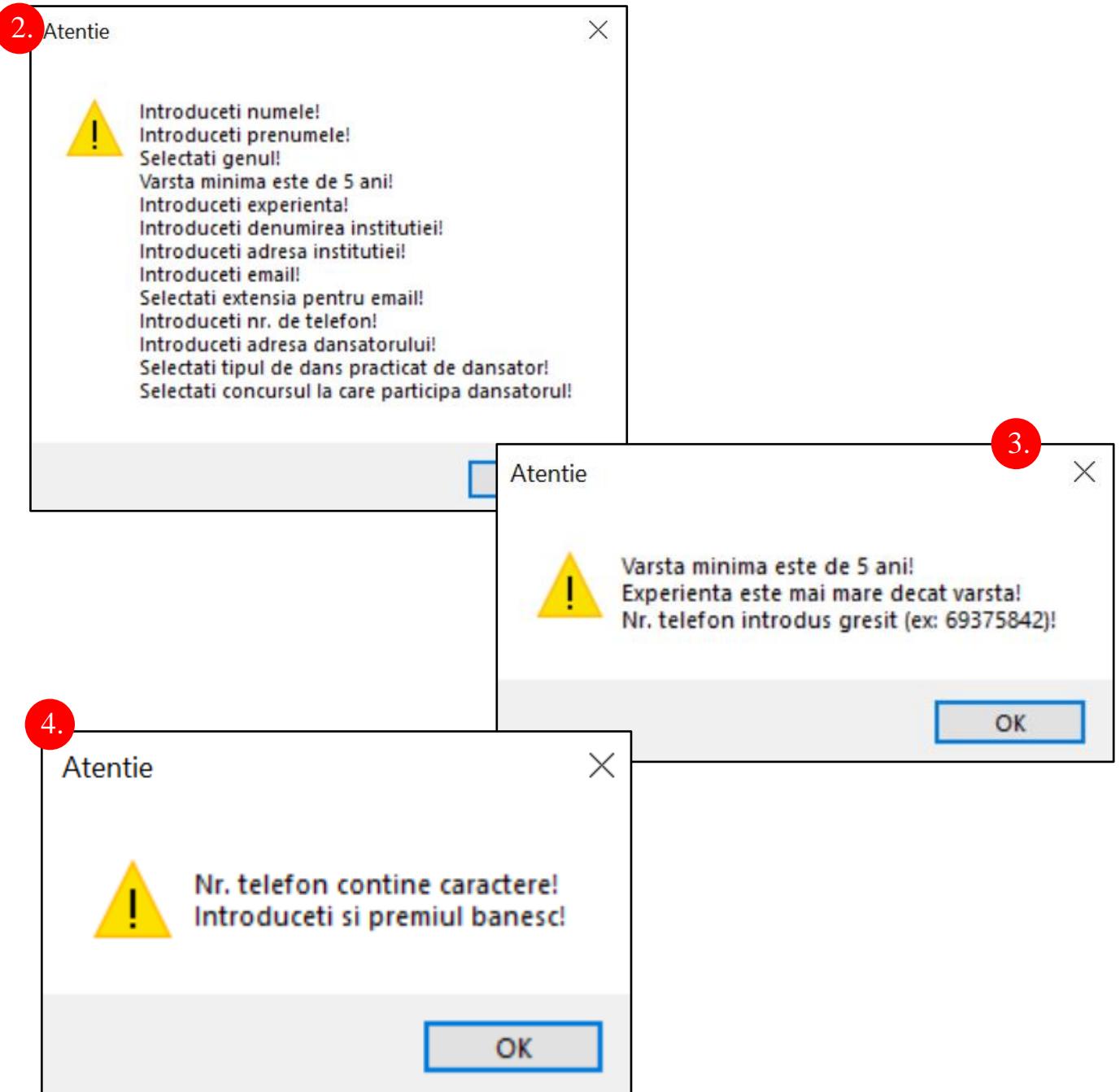
Concurs

 - Denumirea**: Red box around the dropdown menu.

Premiu (optional)

 - Locul**: Red box around the dropdown menu.
 - Premiu banesc**: Red box around the input field.
 - Lei**: Red box around the dropdown menu.
 - Reseteaza**
 - Salveaza**

Tipuri de mesaje:



Excluderea unui dansator

Pentru a exclude un dansator, ne logăm în calitate de administrator (vezi la subiectul „logarea ca administrator”). După urmărim pașii:

2. Revenire la meniul principal;
3. Codul dansatorului ce va fi exclus;
4. Zona de afişare a datelor;
5. Verificarea codului introdus;
6. Excluderea dansatorului.

Introducem codul dansatorului ce dorim să-l excludem:

1.

2.

3.

4.

5.

Observăm că a fost găsit dansatorul cu codul introdus

Cod	Gen
SC4892	Masculin

Nume	Varsta	Ani
Shah	34	Ani

Prenume	Experienta	Ani
Chenai	26	Ani

Excludere cu succes

Dansatorul a fost exclus cu succes!

OK

Baza de date înainte de excluderea dansatorului:

CodDansator	Nume	Prenume	Sex	DataNasterii	Experienta	Email	Telefon	Adresa
PA9176	Piper	Antony	M	1983-06-26	32	aibrahim@gmail.com	+373 562 28 191	Balti
HB4085	Hanna	Brooklyn	M	1986-01-17	28	maratb@gmail.com	+373 788 20 440	Balti
SC4892	Shah	Chenai	M	1987-11-04	26	tubesteak@gmail.com	+373 562 05 948	Chisinau
SA5470	Solomon	Anastasia	F	2003-03-04	8	anastasia@gmail.com	+373 602 43 654	Chisinau

NumelInstitutie	Adresa	DenumireTipDans	DenumireCategorie
Universitatea Pedagogica de Stat "Ion Creanga"	Tiraspol	Kagura	Dans ceremonial
Universitatea de Stat "Grigore Tamblac"	Balti	Dans baroc	Dans istoric
Institutul International de Management "IMI-NOVA"	Tiraspol	Dansuri rituale din China	Dans ceremonial
CEITI	Chisinau	Flamenco	Alte

Baza de date după excluderea dansatorului:

CodDansator	Nume	Prenume	Sex	DataNasterii	Experienta	Email	Telefon	Adresa
PA9176	Piper	Antony	M	1983-06-26	32	aibrahim@gmail.com	+373 562 28 191	Balti
HB4085	Hanna	Brooklyn	M	1986-01-17	28	maratb@gmail.com	+373 788 20 440	Balti
SA5470	Solomon	Anastasia	F	2003-03-04	8	anastasia@gmail.com	+373 602 43 654	Chisinau

NumelInstitutie	Adresa	DenumireTipDans	DenumireCategorie
Universitatea Pedagogica de Stat "Ion Creanga"	Tiraspol	Kagura	Dans ceremonial
Universitatea de Stat "Grigore Tamblac"	Balti	Dans baroc	Dans istoric
CEITI	Chisinau	Flamenco	Alte

Erori întâlnite în cadrul excluderii unui dansator nou

The screenshot shows a window titled 'Excludere Dansator'. In the top-left corner, there's a green bar with the word 'Inapoi'. Below it, a text input field contains the value 'reger' (marked with a red circle labeled '1.'). To the right of the input fields for 'Cod' and 'Gen', there's a modal dialog box titled 'Eroare de cautare' (Search error) containing the message 'Nu s-a gasit dansator cu asa cod!' (No dancer was found with this code!) next to a red 'X' icon. A red arrow points from the 'reger' input field to this error message.

The screenshot shows a window titled 'Excludere Dansator'. In the top-left corner, there's a green bar with the word 'Inapoi'. Below it, a text input field is empty (marked with a red circle labeled '1.'). To the right of the input fields for 'Cod' and 'Gen', there's a modal dialog box titled 'Eroare de introducere' (Input error) containing the message 'Introduceti codul!' (Enter the code!) next to a yellow warning triangle icon. A red arrow points from the empty 'Cod' input field to this error message.

Excludere Dansator

Inapoi

1.

Codul dansatorului

Cod Gen

Nume Varsta Ani

Prenume Experienta Ani

Verifica 2.

Eroare de introducere 3.

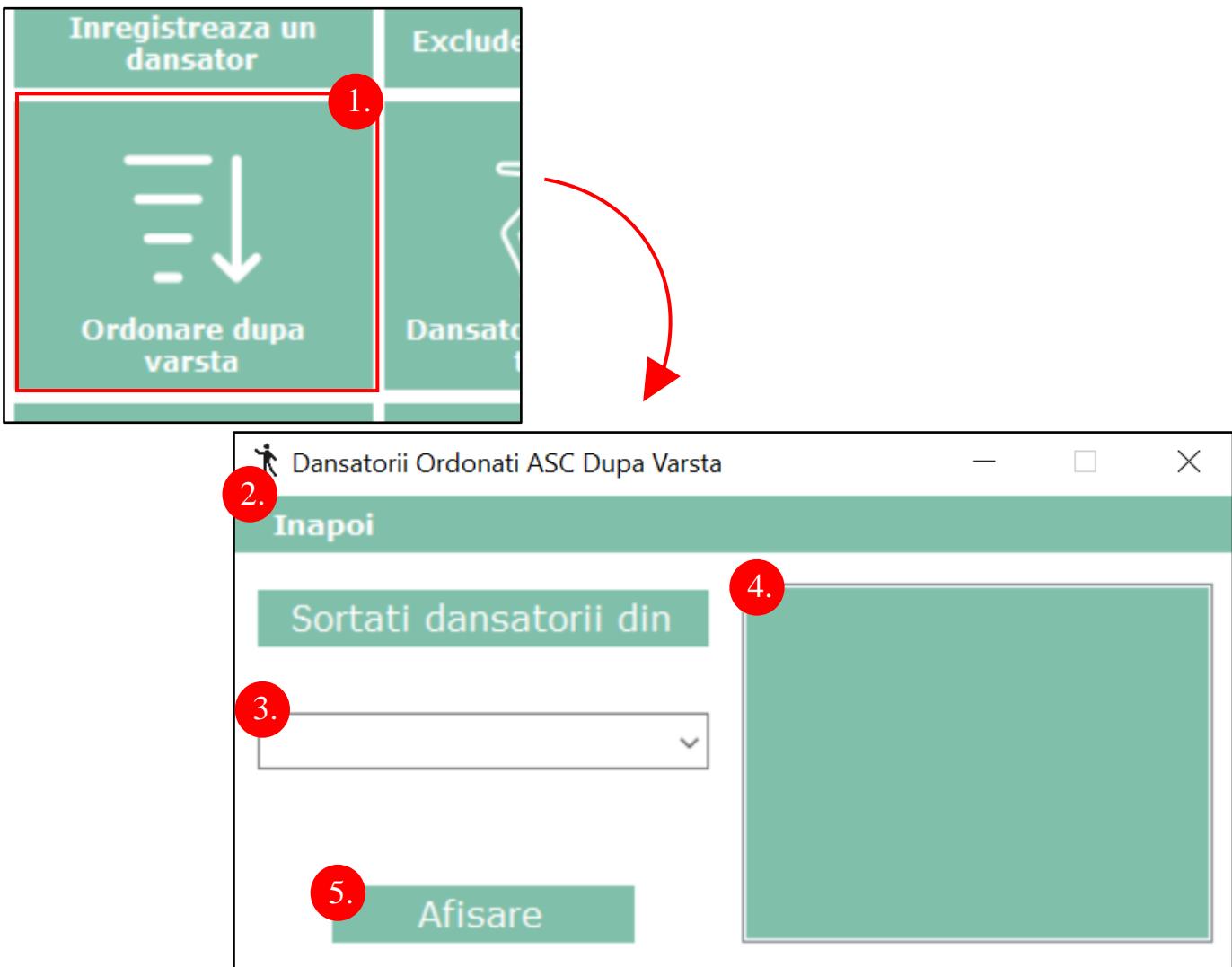
Introduceti codul!

Afișarea dansatorului cel mai Tânăr / bătrân

The diagram illustrates a user interface flow for selecting the youngest or oldest dancer. It consists of three main windows:

- Main Menu Window:** Shows a dancing couple icon and the text "de un dansator". A red box highlights the instruction "Selecteaza dansator tanar / batran" (Select the youngest / oldest dancer) with a red circle labeled "1.". A red arrow points from this window to the second window.
- First Selection Window:** Titled "Dansatorul cel mai tanar / batran". It has two tabs: "Cel mai tanar" (selected) and "Cel mai in varsta". A red circle labeled "2." is on the "Inapoi" (Back) button. A red box highlights the data fields: "Codul dansatorului" (EA9201), "Nume" (Erickson), "Varsta" (6 Ani), "Prenume" (Alissa), and "Tip dans" (Dans solo). A red arrow points from this window to the third window.
- Second Selection Window:** Titled "Dansatorul cel mai tanar / batran". It also has "Cel mai tanar" and "Cel mai in varsta" tabs. A red circle labeled "3." is on the "Inapoi" button. A red box highlights the data fields: "Codul dansatorului" (DM6019), "Nume" (Delgado), "Varsta" (39 Ani), "Prenume" (Mariah), and "Tip dans" (Lambada). A red circle labeled "4." is on the "Cel mai in varsta" tab.

Ordonarea după vârstă



2. Revenire la meniul principal;
3. Denumirea concursului selectată de către utilizator dintr-o listă;
4. Zona de afișarea dansatorilor din cadrul unui concurs, ordonați ascendent după vârstă;
5. Afișarea datelor dansatorilor.

Dansatorii Ordonati ASC Dupa Varsta

Inapoi

Sortati dansatorii din

Toate Concursurile 1.

Afisare 2.

Dansatorii Ordonati ASC Dupa Varsta

Inapoi

Sortati dansatorii din

Toate Concursurile 3.

Afisare

Erickson Alissa 6 ani
Mccallum Ryley 7 ani
Goulding Hira 7 ani
Preston Maci 7 ani
Davidson Darcie-Mae 7 ani
Ibarra Emry 8 ani
Bains Brenna 8 ani
Kirkland Abigale 8 ani
Arias Krista 9 ani
Neale Eduard 10 ani

Dansatorii Ordonati ASC Dupa Varsta

Inapoi

Sortati dansatorii din

Fantezie si Talent 4.

Afisare 5.

Dansatorii Ordonati ASC Dupa Varsta

Inapoi

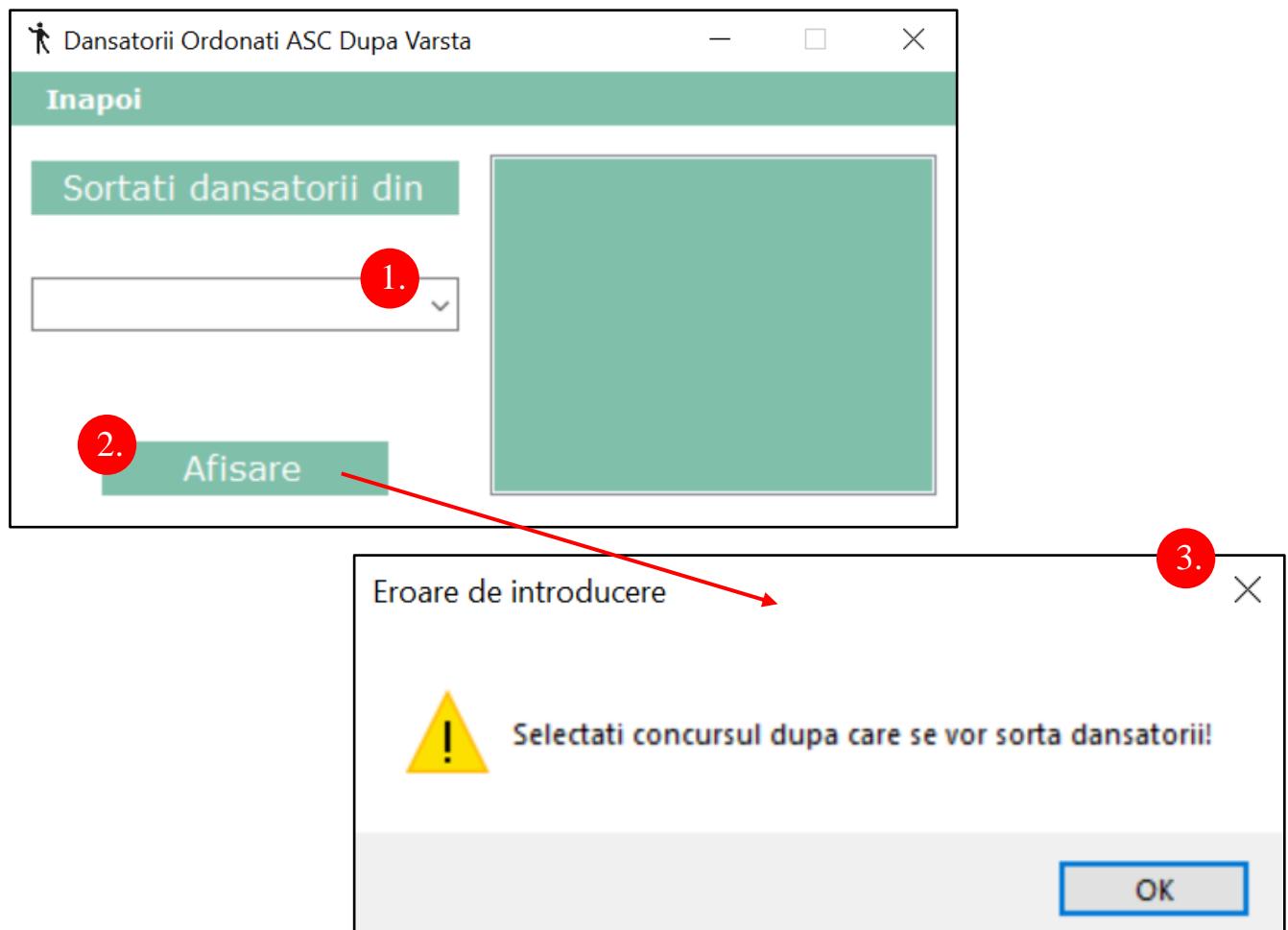
Sortati dansatorii din

Fantezie si Talent 6.

Afisare

Solomon Anastasia 19 ani
Mcpherson Emilis 23 ani
Grey Jett 23 ani
Wilkerson Harvir 24 ani
Holman Greg 27 ani
Flower Franciszek 29 ani

Erori întâlnite în cadrul afișării dansatorilor ordonați asc după vârstă



Dansatoarea cea mai Tânără

1.

2.

Revenire la meniul principal

Inapoi

Cea Mai Tanara Dansatoare

—

X

Codul dansatoarei EA9201

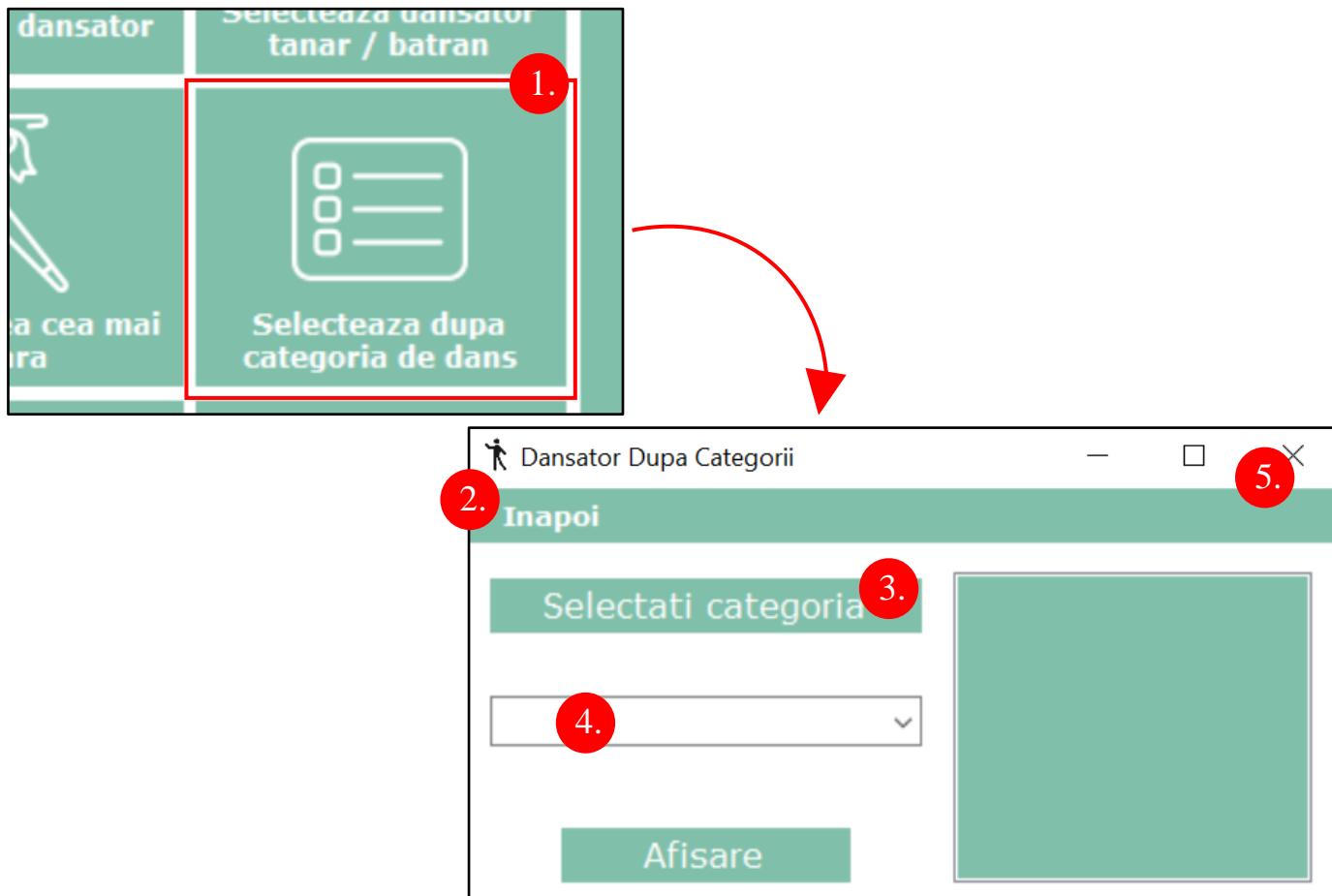
Nume Erickson

Varsta 6 Ani

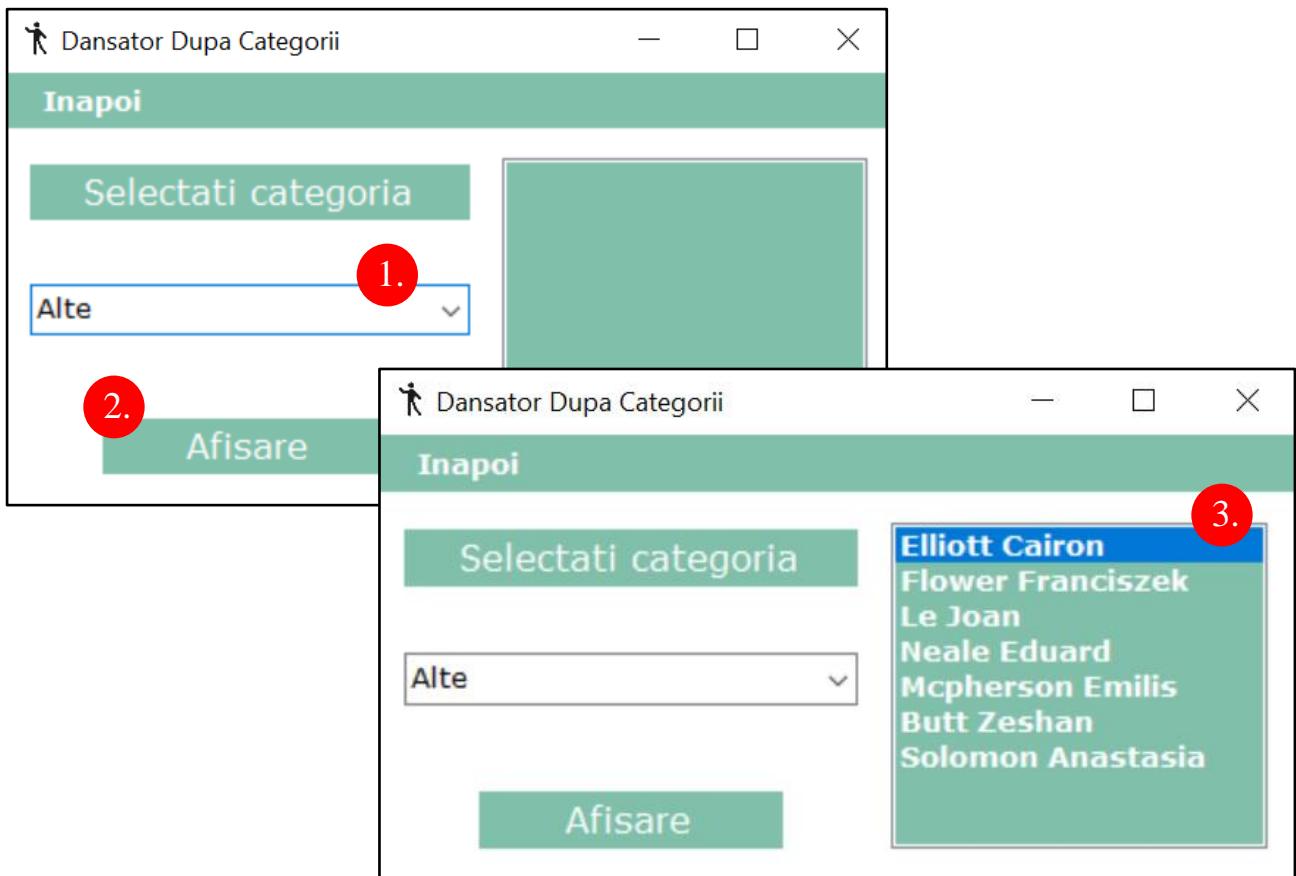
Prenume Alissa

Genul Femenin

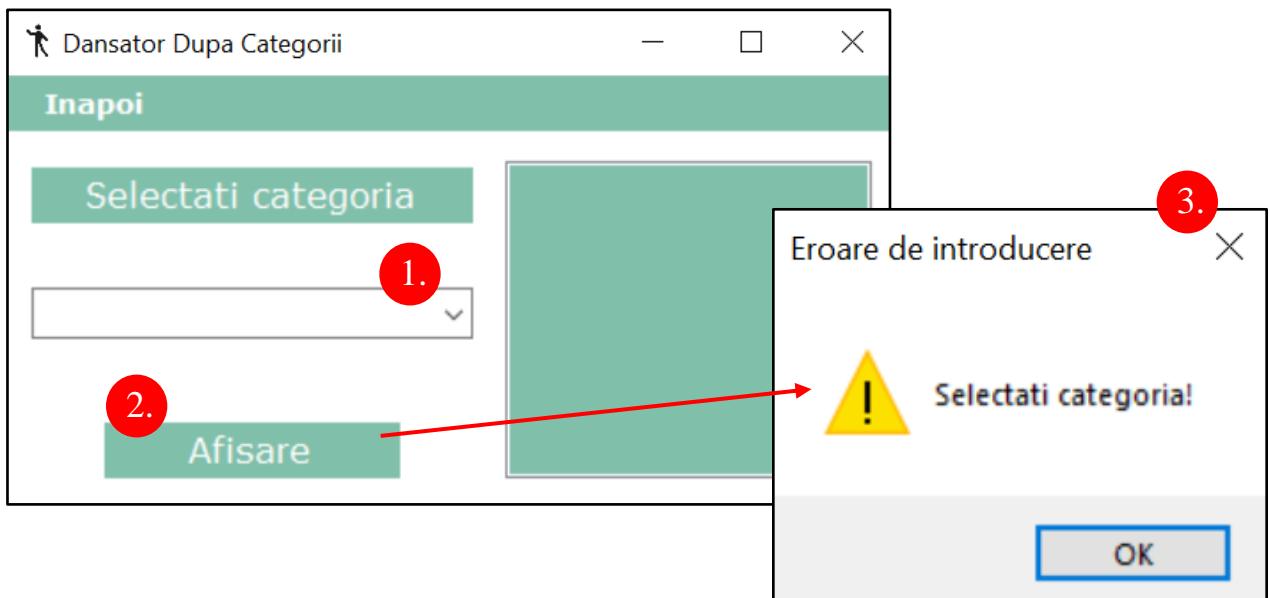
Afișarea listei dansatorilor, în funcție de categoria de dans selectată



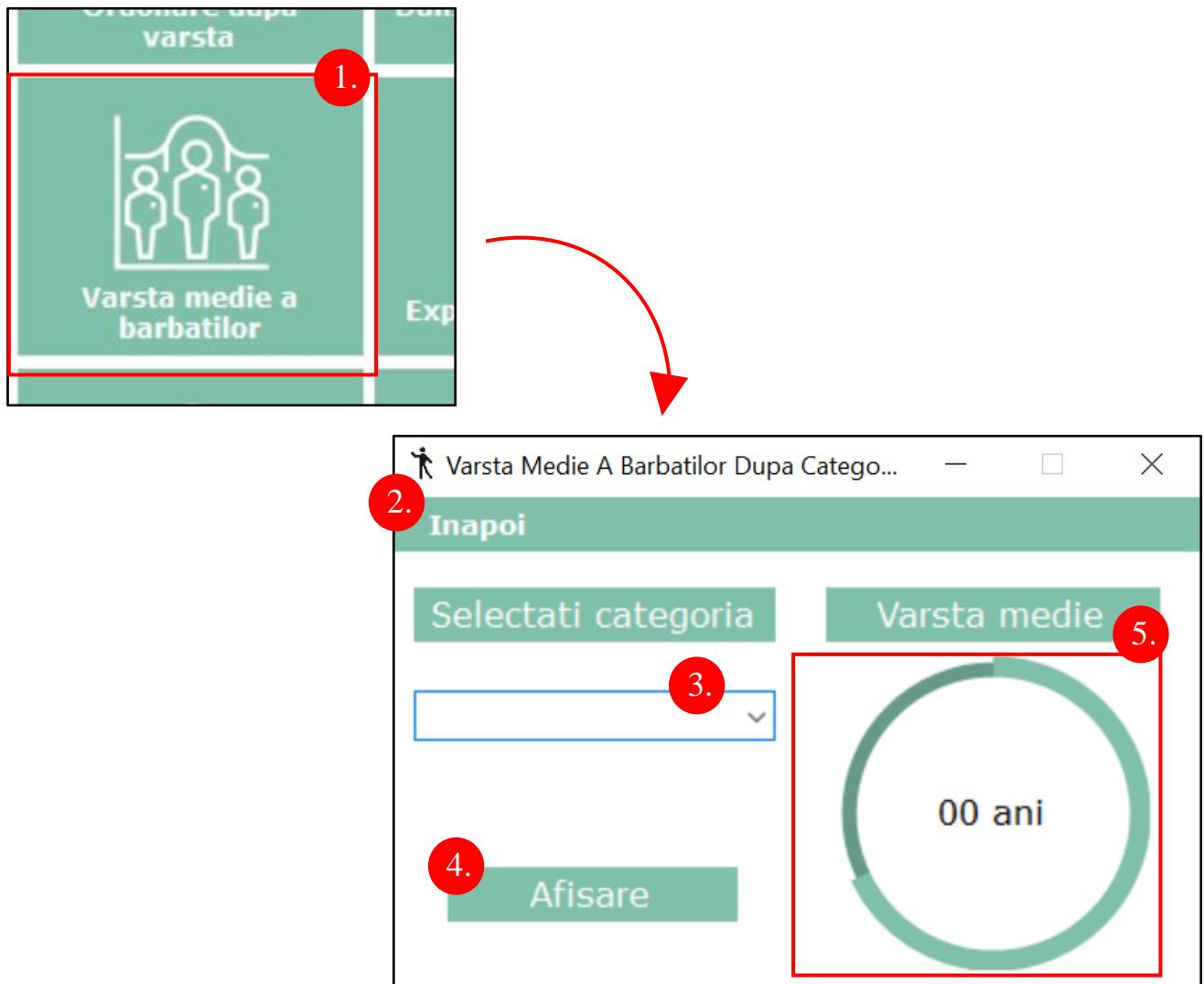
2. Revenire la meniul principal;
3. Denumirea categoriei selectată de către utilizator dintr-o listă;
4. Afisarea datelor dansatorilor;
5. Zona de afișarea dansatorilor din cadrul unui concurs, ordonați ascendent după vârstă.



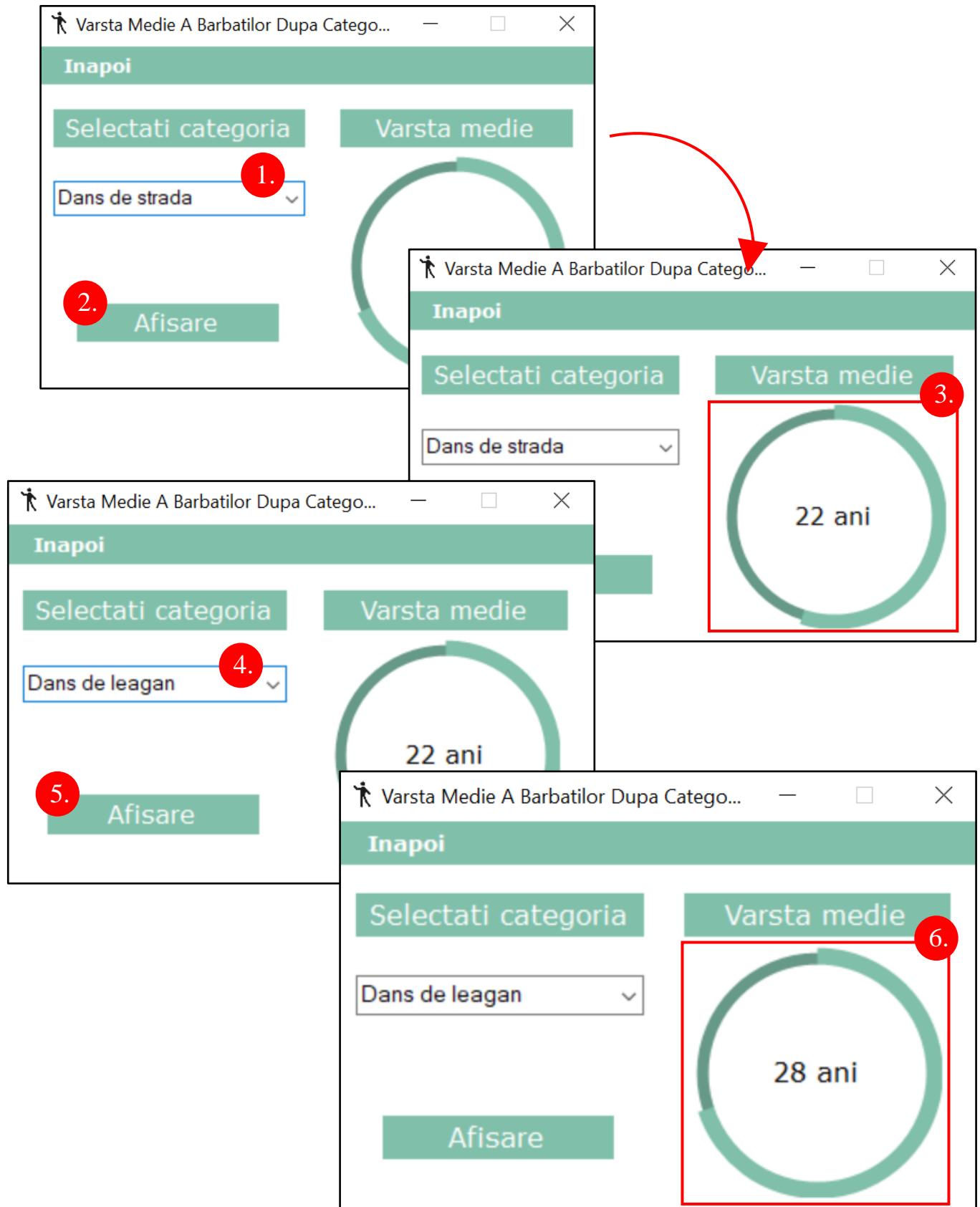
Erori întâlnite în cadrul afișării dansatorilor după categorii



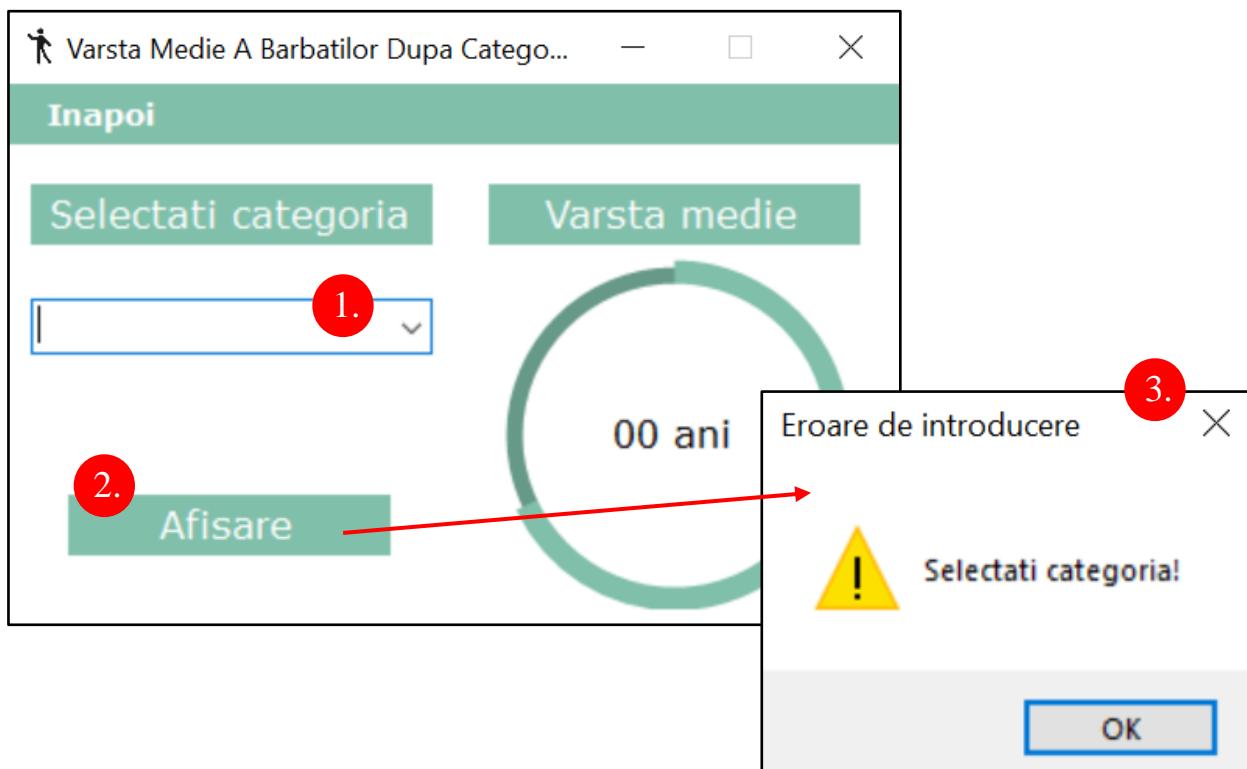
Vârsta medie a bărbaților, în funcție de categoria de dans selectată



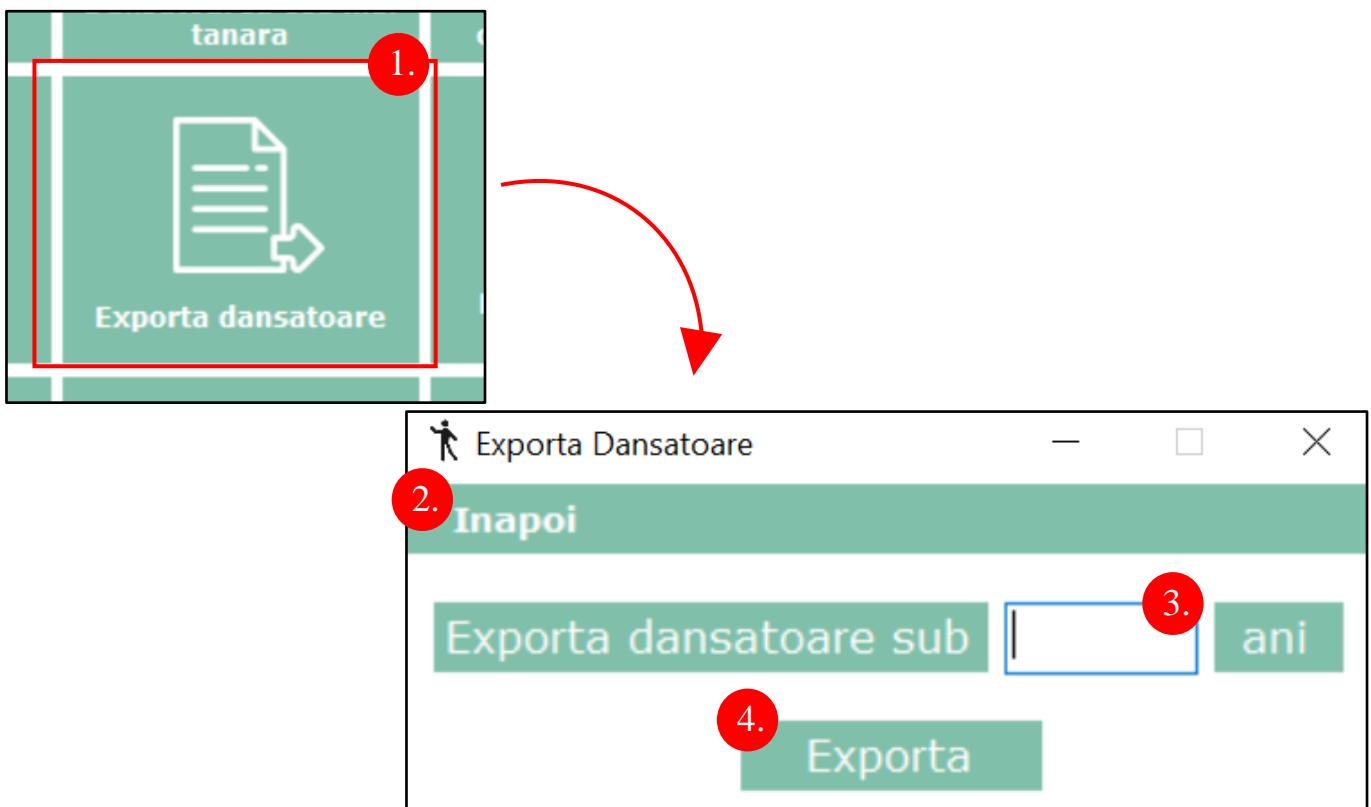
2. Revenire la meniul principal;
3. Denumirea categoriei selectată de către utilizator dintr-o listă;
4. Zona de afișare a vârstei medie a bărbaților din categoria selectată;
5. Afișarea vârstei medii.



Erori întâlnite în cadrul afișării vârstei medie a bărbaților

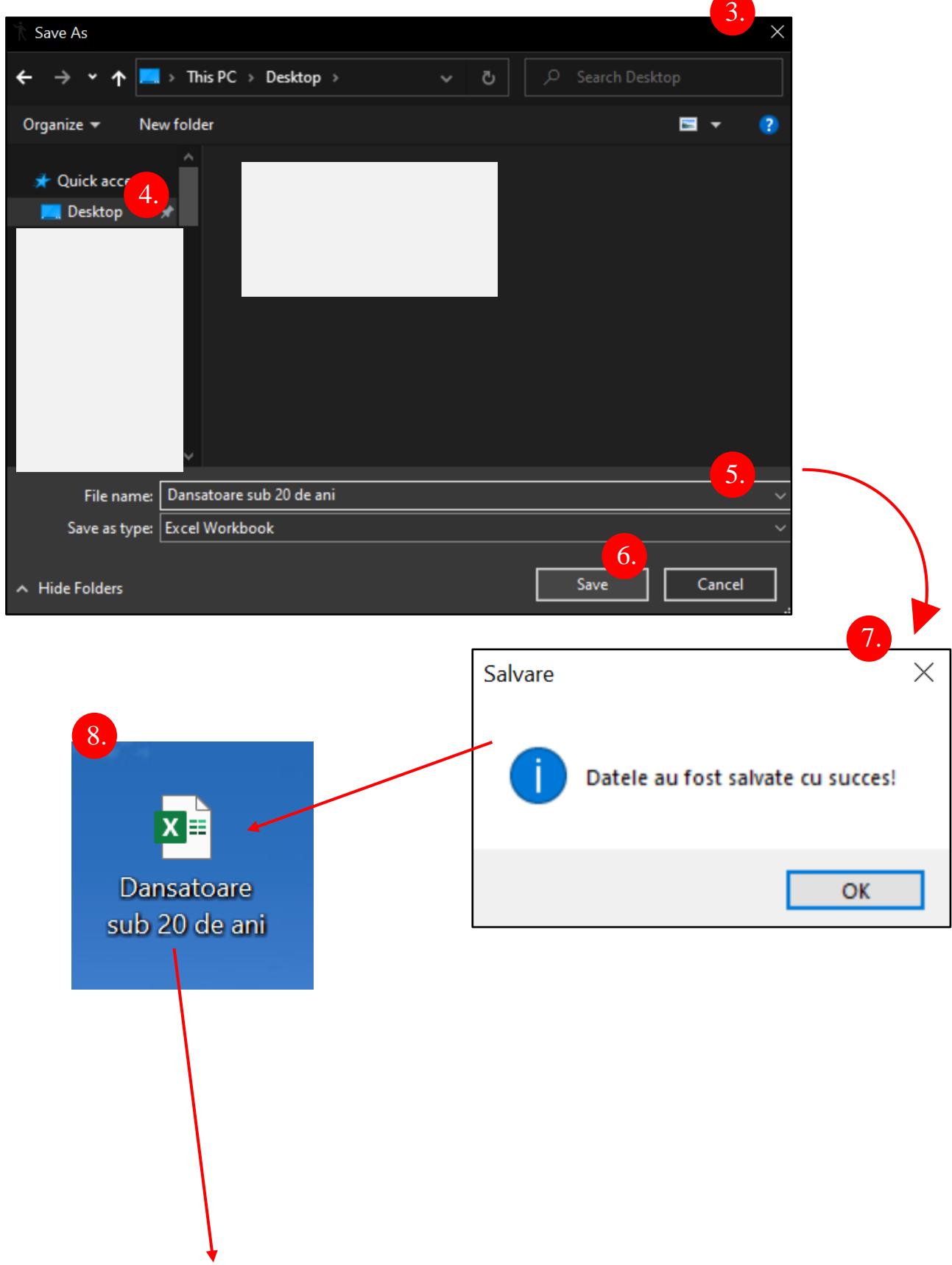


Exportarea dansatoarelor într-un fișier MS EXCEL



2. Revenire la meniul principal;
3. Caseta pentru introducerea vîrstei dansatoarei;
4. Exportarea datelor.

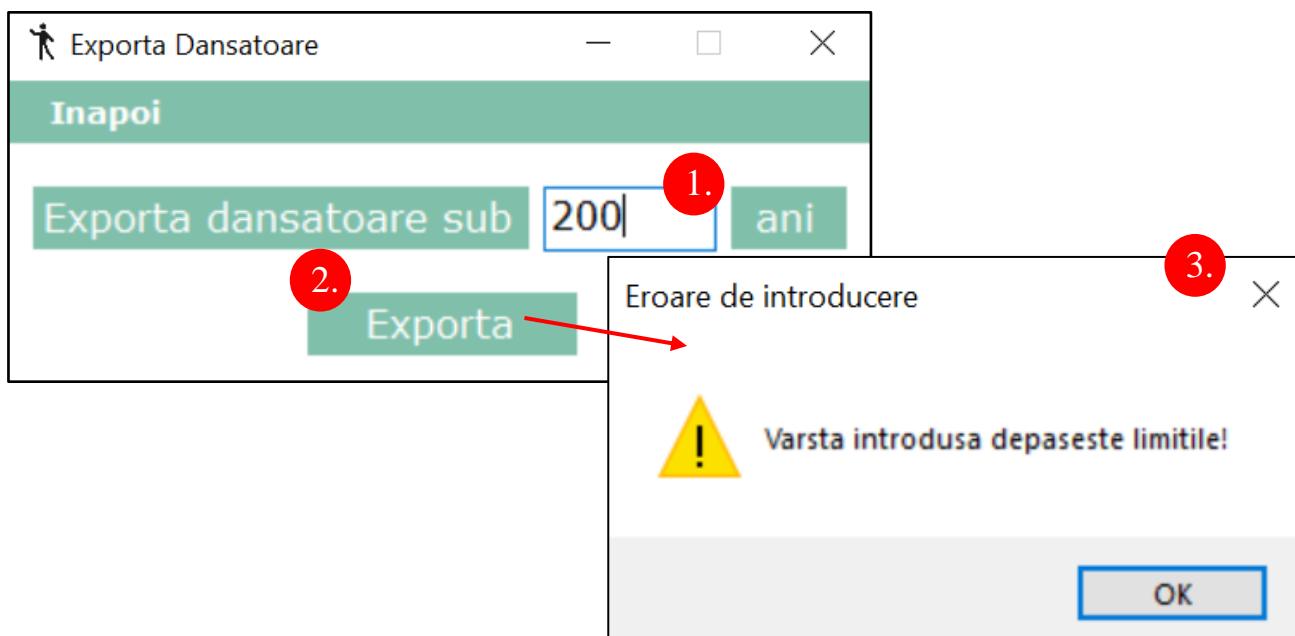


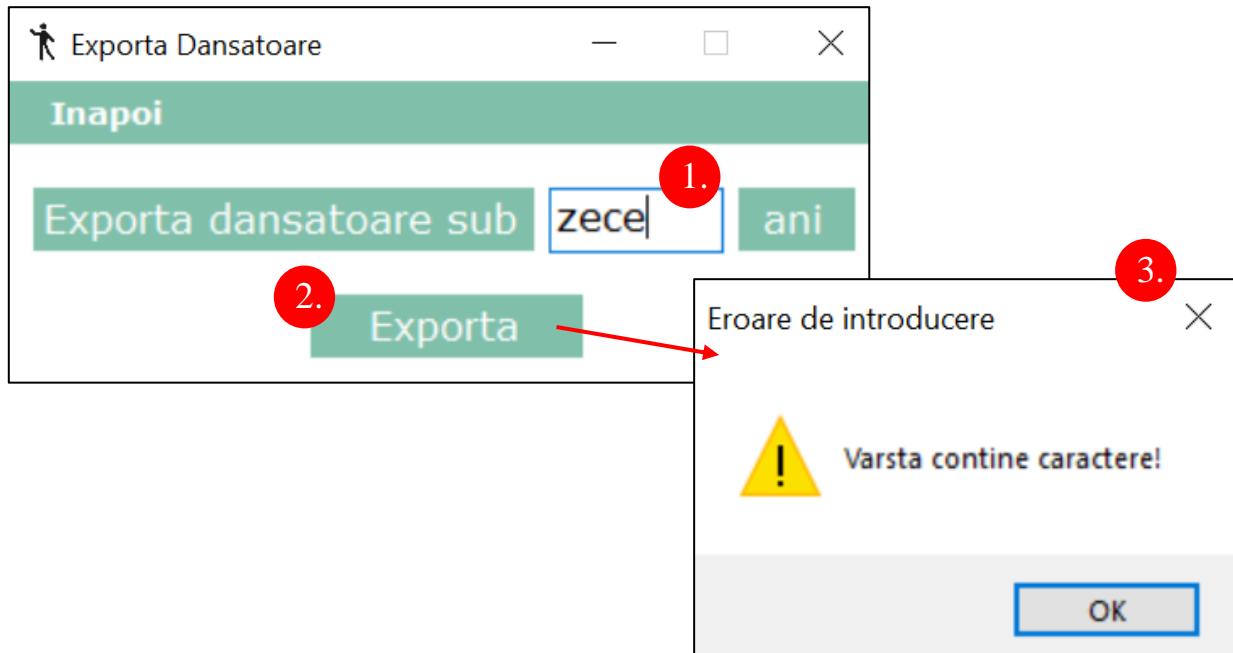


9.

Cod dansator	Nume	Prenume	Experienta	Varsta
MR3716	Mccallum	Ryley	2	7
DD1783	Davidson	Darcie-Mae	1	7
IE1634	Ibarra	Emyr	3	8
MP7371	Mccormick	Phillipa	11	17
KA2531	Kirkland	Abigale	1	8
BB2610	Bains	Brenna	2	8
KZ8756	Kaiser	Zahra	4	12
DN0348	Dickinson	Nia	7	12
NK4000	Nicholls	Kristen	6	11
AK2274	Arias	Krista	2	9
CL0815	Campos	Leila	4	11
HK7613	Harper	Karim	5	10
MM4671	Meza	Malika	2	10
GH9753	Goulding	Hira	1	7
EA9201	Erickson	Alissa	2	6
SA5470	Solomon	Anastasia	8	19

Erori întâlnite în cadrul exportării datelor





Datele de contact a dansatorilor

1.

2.

3.

4.

5.

2. Revenire la meniul principal;

3. Zona de afişare a datelor de contact;

4. Afişarea datelor de contact a dansatorului următor;

5. Afişarea datelor de contact a dansatorului precedent.

Cod	Nume	Telefon
MR3716	Mccallum	+373 *****37
Adresa	Prenume	Email
Chisinau	Ryley	koh***@gmail.com

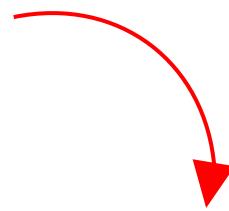
Inapoi

Date de contact

Cod	Nume	Telefon
MR3716	Mccallum	+373 *****37
Adresa	Prenume	Email
Chisinau	Ryley	koh***@gmail.com

Precedent Urmatorul

1.



Inapoi

Date de contact

Cod	Nume	Telefon
BB2610	Bains	+373 *****26
Adresa	Prenume	Email
Truseni	Brenna	fuk****@gmail.com

2. Precedent Urmatorul

Inapoi

Date de contact

Cod	Nume	Telefon
MR3716	Mccallum	+373 *****37
Adresa	Prenume	Email
Chisinau	Ryley	koh***@gmail.com

Precedent Urmatorul

3.

Afișarea listei dansatorilor cu instituția de învățământ

The screenshot shows a software application window titled "Dansatori Cu Institutia". The window has a green header bar with the title and a red "X" button. Below the header is a blue navigation bar with the text "Inapoi" (Back) and two red circular numbered labels: "2." on the left and "3." on the right. The main content area contains four input fields arranged in a grid:

Nume	Codul
Mccallum	MR3716
Prenume	Institutia
Ryley	Liceul Teoretic "Pro Succes"

At the bottom of the window are two green buttons: "Precedent" (Previous) on the left with a red "5." label above it, and "Urmatorul" (Next) on the right with a red "4." label above it.

2. Revenire la meniul principal;
3. Zona de afișare dansatorului și instituției;
4. Afisarea dansatorului și instituției următoare;
5. Afisarea dansatorului și instituției precedente.

Dansatori Cu Institutia

Inapoi

Nume	Codul
Mccallum	MR3716
Prenume	Institutia
Ryley	Liceul Teoretic "Pro Succes"

Precedent Urmatorul

1.

Dansatori Cu Institutia

Inapoi

Nume	Codul
Bains	BB2610
Prenume	Institutia
Brenna	Liceul Teoretic cu Profil Sportiv "Gloria"

Precedent Urmatorul

2.

Dansatori Cu Institutia

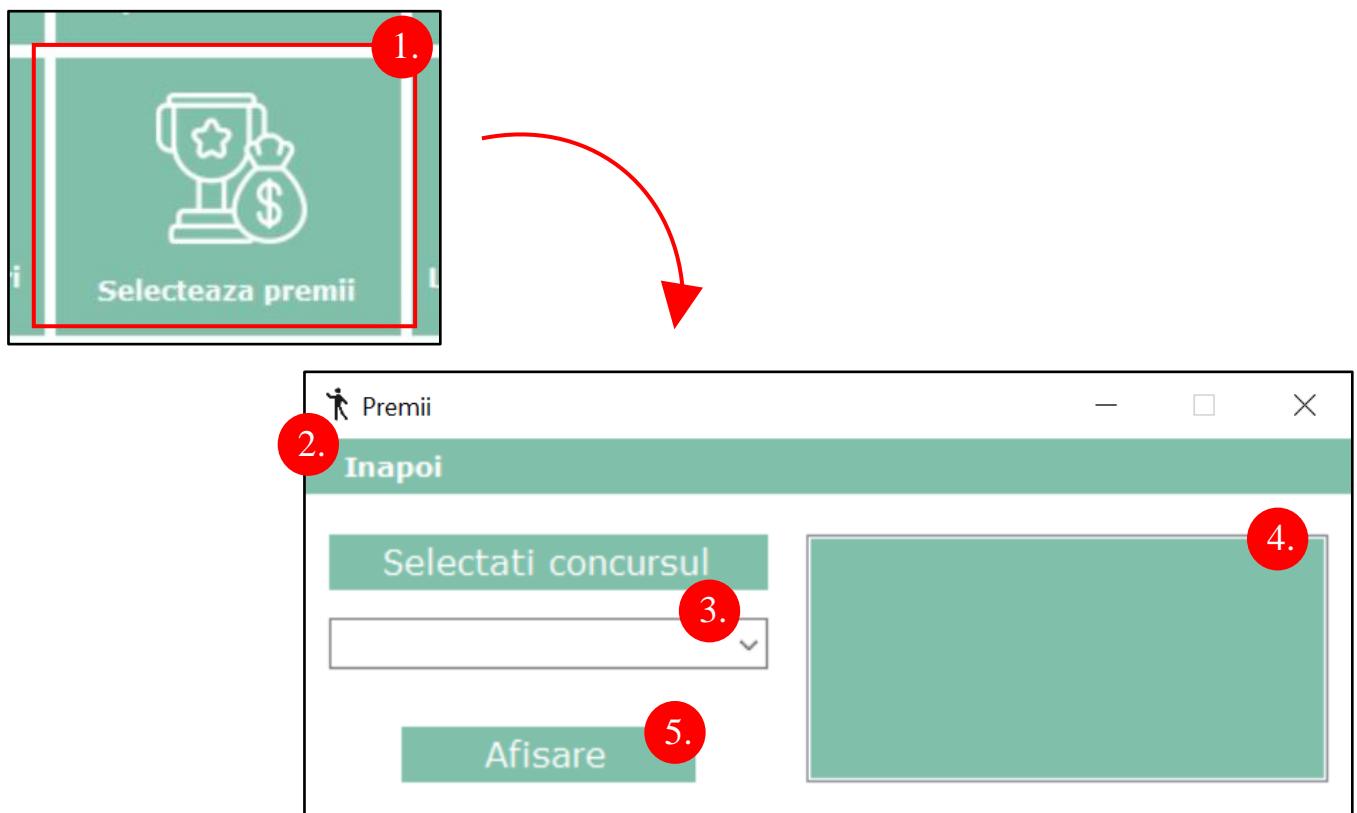
Inapoi

Nume	Codul
Mccallum	MR3716
Prenume	Institutia
Ryley	Liceul Teoretic "Pro Succes"

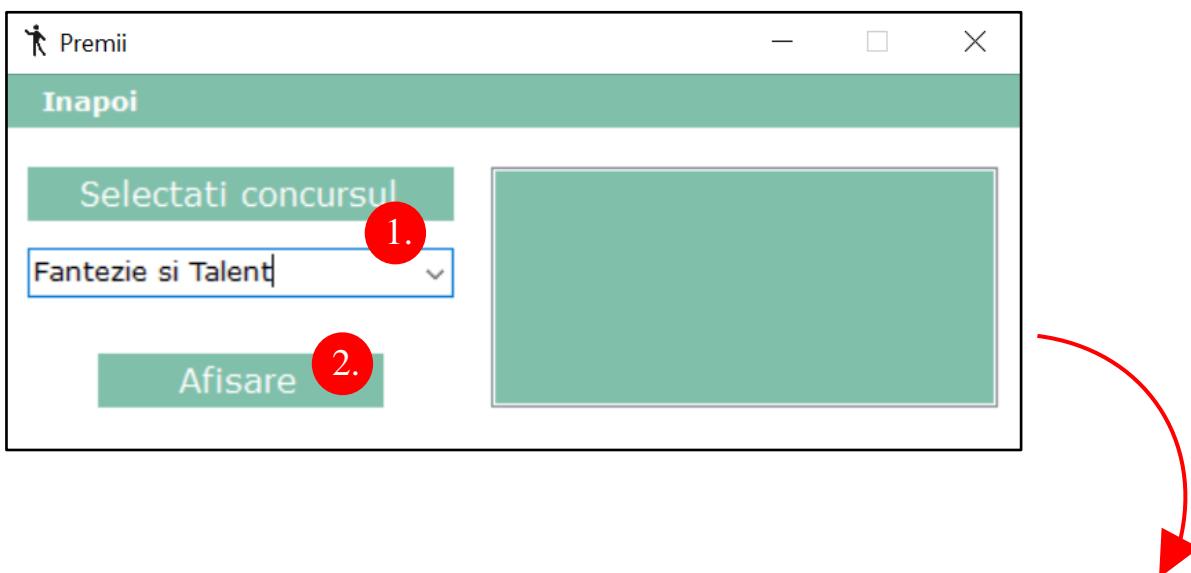
Precedent Urmatorul

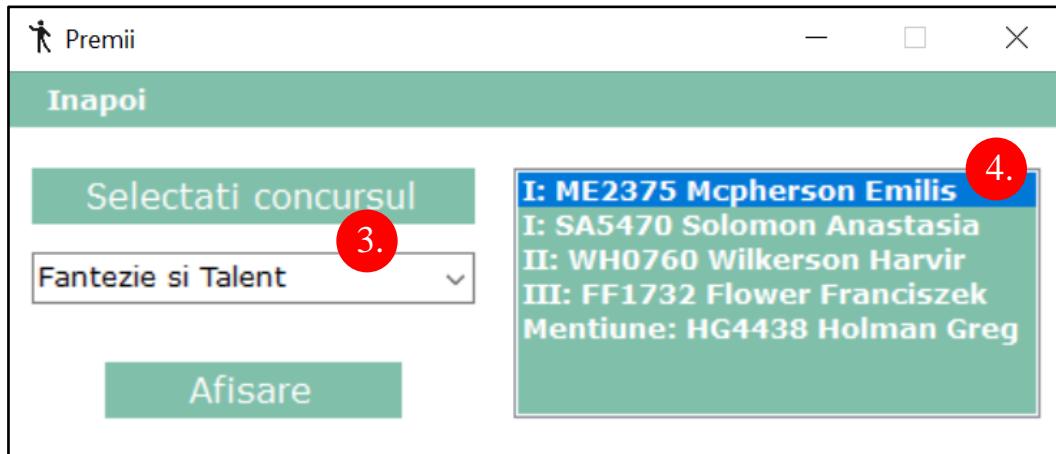
3.

Afișarea listei dansatorilor cu premiile câștigate

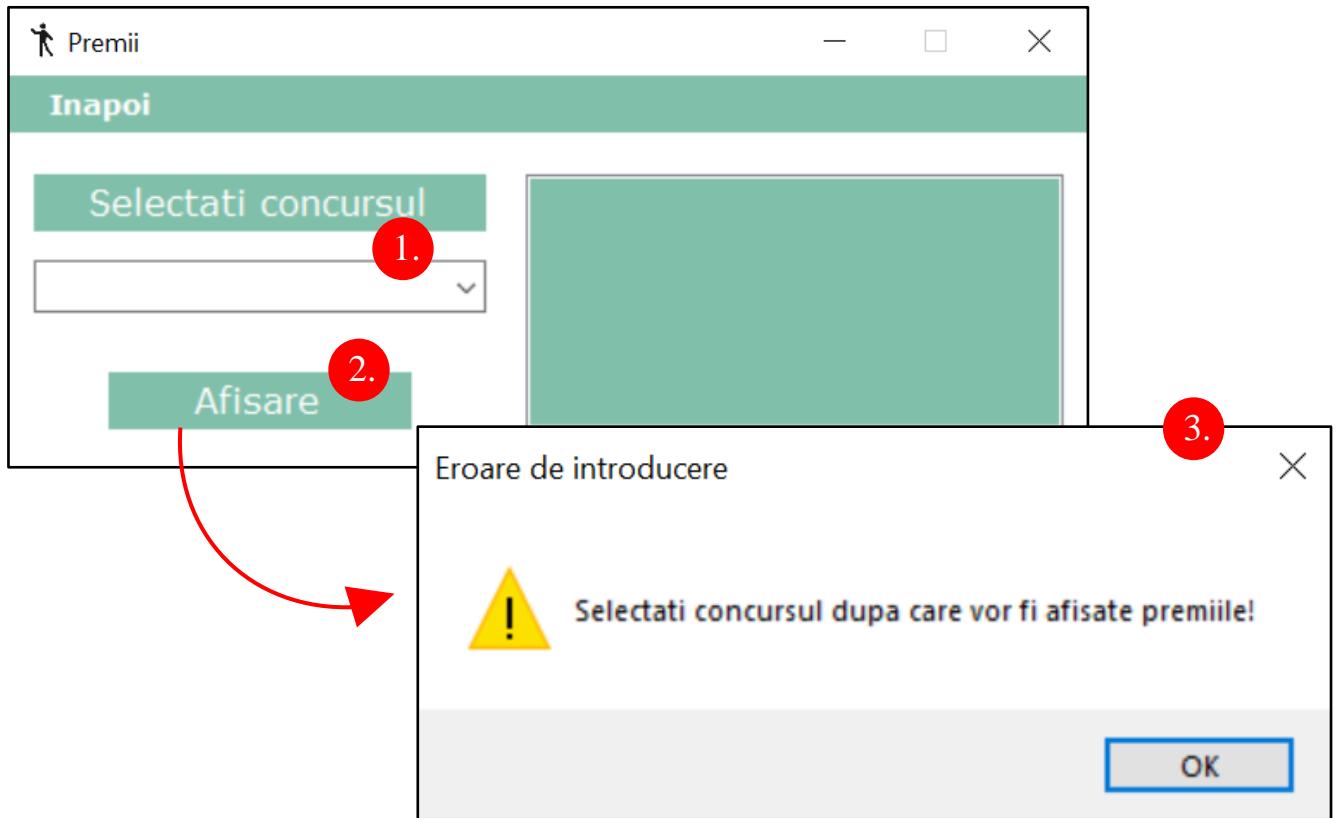


2. Revenire la meniul principal;
3. Denumirea concursului selectată de către utilizator dintr-o listă;
4. Zona de afișare dansatorului și premiului;
5. Afișarea datelor dansatorului și premiului său.

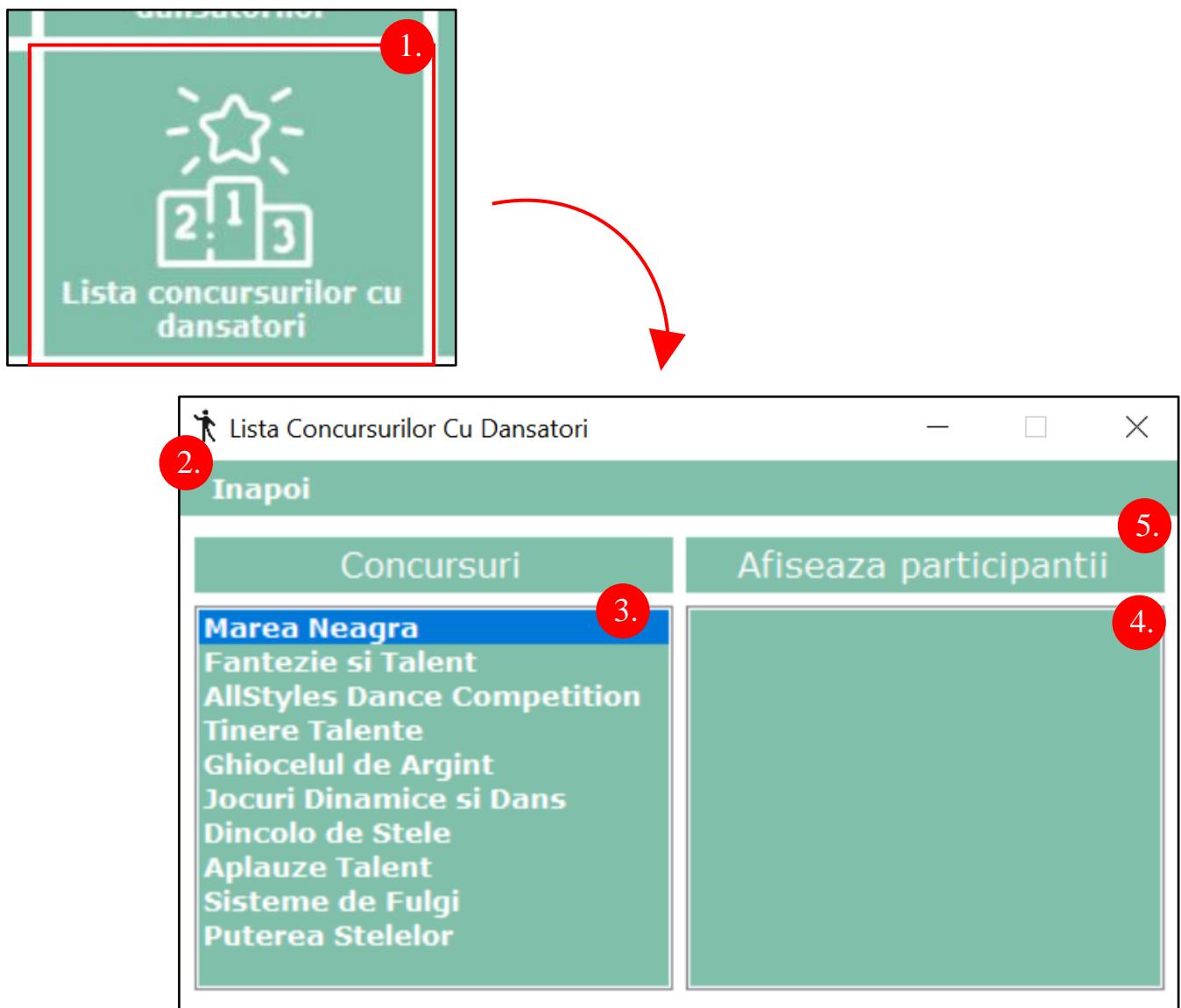




Erori întâlnite în cadrul afișării listei dansatorilor cu premiile câștigate



Afișarea listei dansatorilor, în funcție de concursul selectat



2. Revenire la meniul principal;
3. Zona de afișare a denumirii concursurilor, ce va fi selectat pentru a afișa datele dansatorilor;
4. Zona de afișare a datelor dansatorilor;
5. Afișarea datelor dansatorilor.

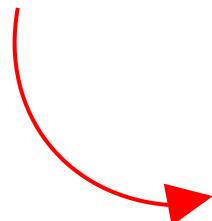
 Lista Concursurilor Cu Dansatori

Inapoi

Concursuri	Afiseaza participantii
Marea Neagra	
Fantezie si Talent	
AllStyles Dance Competition	
Tinere Talente	
Ghiocelul de Argint	
Jocuri Dinamice si Dans	
Dincolo de Stele	
Aplauze Talent	
Sisteme de Fulgi	
Puterea Stelelor	

1.

2.



 Lista Concursurilor Cu Dansatori

Inapoi

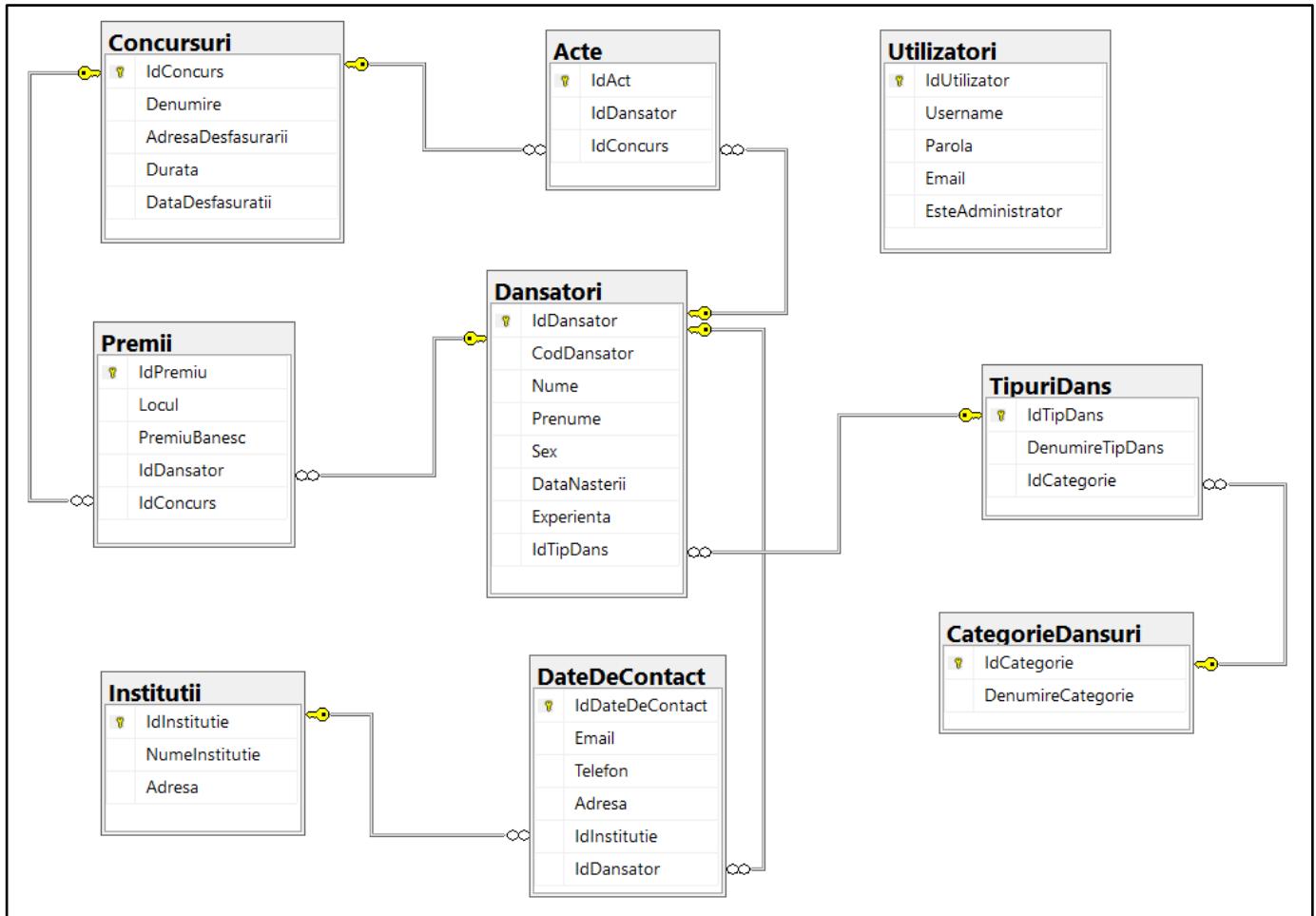
Concursuri	Afiseaza participantii
Marea Neagra	
Fantezie si Talent	
AllStyles Dance Competition	
Tinere Talente	
Ghiocelul de Argint	
Jocuri Dinamice si Dans	
Dincolo de Stele	
Aplauze Talent	
Sisteme de Fulgi	
Puterea Stelelor	

Wilkerson Harvir	3.
Mcpherson Emilia	
Flower Franciszek	
Holman Greg	
Grey Jett	
Solomon Anastasia	

Prezentarea generală a bazei de date

Baza de date ConcursDeDans este creată pentru a duce evidență despre activitatea concursurilor, organizarea structurată a datelor, lucru asupra datelor existente, precum și gestionarea eficientă a acestora.

Diagrama E-R a bazei de date ConcursDeDans



Scenariul bazei de date ConcursDeDans

```
--Utilizarea bazei de date 'MASTER'
USE MASTER
GO

--Stergerea BD 'ConcursDeDans' in caz ca exista
IF EXISTS (SELECT * FROM sys.databases WHERE NAME = 'ConcursDeDans')
BEGIN
    ALTER DATABASE ConcursDeDans SET SINGLE_USER WITH ROLLBACK IMMEDIATE
    DROP DATABASE ConcursDeDans
END
GO

--Crearea bazei de date 'ConcursDeDans'
CREATE DATABASE ConcursDeDans
GO

--Utilizarea bazei de date 'ConcursDeDans'
USE ConcursDeDans
GO

--Crearea tabelului INSTITUTII
CREATE TABLE Institutii (
    IdInstitutie INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    NumeInstitutie VARCHAR(100) NOT NULL,
    Adresa VARCHAR(50) NOT NULL,
)
GO

--Crearea tabelului CATEGORIE_DANSURI
CREATE TABLE CategorieDansuri (
    IdCategorie INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    DenumireCategorie VARCHAR(50) NOT NULL,
)
GO

--Crearea tabelului TIP_DANSURI
CREATE TABLE TipuriDans (
    IdTipDans INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    DenumireTipDans VARCHAR(50) NOT NULL,
    IdCategorie INT NOT NULL,
)
GO

--Crearea cheii straine intre tabelul CategorieDansuri si TipuriDans
ALTER TABLE TipuriDans WITH CHECK ADD CONSTRAINT FK_TipuriDans_CategorieDansuri
FOREIGN KEY(IdCategorie)
REFERENCES CategorieDansuri(IdCategorie)
ON DELETE CASCADE
GO

--Crearea tabelului DANSATORI
CREATE TABLE Dansatori (
    IdDansator INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    CodDansator VARCHAR(6) NOT NULL,
    Nume VARCHAR(20) NOT NULL,
    Prenume VARCHAR(20) NOT NULL,
)
```

```

        Sex VARCHAR(1) NOT NULL,
        DataNasterii DATE NOT NULL,
        Experienta INT NOT NULL,
        IdTipDans INT NOT NULL,
    )
GO

--Crearea cheii straine intre tabelul Dansuri si TipuriDans
ALTER TABLE Dansatori WITH CHECK ADD CONSTRAINT FK_Dansatori_TipuriDans
FOREIGN KEY(IdTipDans)
REFERENCES TipuriDans(IdTipDans)
ON DELETE CASCADE
GO

--Crearea tabelului DATE_DE_CONTACT
CREATE TABLE DateDeContact (
    IdDateDeContact INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    Email VARCHAR(50) NOT NULL,
    Telefon VARCHAR(15) NOT NULL,
    Adresa VARCHAR(50) NOT NULL,
    IdInstitutie INT NOT NULL,
    IdDansator INT NOT NULL,
)
GO

--Crearea cheii straine intre tabelul Institutii si DateDeContact
ALTER TABLE DateDeContact WITH CHECK ADD CONSTRAINT FK_DateDeContact_Institutii
FOREIGN KEY(IdInstitutie)
REFERENCES Institutii(IdInstitutie)
ON DELETE CASCADE
GO

--Crearea cheii straine intre tabelul Dansatori si DateDeContact
ALTER TABLE DateDeContact WITH CHECK ADD CONSTRAINT FK_DateDeContact_Dansatori
FOREIGN KEY(IdDansator)
REFERENCES Dansatori(IdDansator)
ON DELETE CASCADE
GO

--Crearea tabelului CONCURSURI
CREATE TABLE Concursuri (
    IdConcurs INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    Denumire VARCHAR(100) NOT NULL,
    AdresaDesfasurarii VARCHAR(50) NOT NULL,
    Durata TIME NOT NULL,
    DataDesfasuratii DATE NOT NULL,
)
GO

--Crearea tabelului ACTE
CREATE TABLE Acte (
    IdAct INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    IdDansator INT NOT NULL,
    IdConcurs INT NOT NULL,
)
GO

--Crearea cheii straine intre tabelul Dansatori si Acte
ALTER TABLE Acte WITH CHECK ADD CONSTRAINT FK_Acte_Dansatori

```

```

FOREIGN KEY(IdDansator)
REFERENCES Dansatori(IdDansator)
ON DELETE CASCADE
GO

--Crearea cheii straine intre tabelul Concursuri si Acte
ALTER TABLE Acte WITH CHECK ADD CONSTRAINT FK_Acte_Concursuri
FOREIGN KEY(IdConcurs)
REFERENCES Concursuri(IdConcurs)
ON DELETE CASCADE
GO

--Crearea tabelului PREMII
CREATE TABLE Premii (
    IdPremiu INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    Locul VARCHAR(20) NOT NULL,
    PremiuBanesc MONEY NOT NULL,
    IdDansator INT NOT NULL,
    IdConcurs INT NOT NULL,
)
GO

--Crearea tabelului Utilizatori
CREATE TABLE Utilizatori (
    IdUtilizator INT IDENTITY(1, 1) NOT NULL PRIMARY KEY,
    Username VARCHAR(50) NOT NULL,
    Parola VARCHAR(25),
    Email VARCHAR(50),
    EsteAdministrator BIT NOT NULL,
)
GO

--Crearea cheii straine intre tabelul Dansatori si Premii
ALTER TABLE Premii WITH CHECK ADD CONSTRAINT FK_Premii_Dansatori
FOREIGN KEY(IdDansator)
REFERENCES Dansatori(IdDansator)
ON DELETE CASCADE
GO

--Crearea cheii straine intre tabelul Concursuri si Premii
ALTER TABLE Premii WITH CHECK ADD CONSTRAINT FK_Premii_Concursuri
FOREIGN KEY(IdConcurs)
REFERENCES Concursuri(IdConcurs)
ON DELETE CASCADE
GO

--Inserarea datelor in tabelul INSTITUTII
INSERT INTO Institutii (NumeInstitutie, Adresa) VALUES
    ('Liceul Teoretic "Pro Succes"', 'Chisinau'),
    ('Liceul Teoretic cu Profil Sportiv "Gloria"', 'Balti'),
    ('Liceul Teoretic "Rambam"', 'Tiraspol'),
    ('Gimnaziul "Galata"', 'Cahul'),
    ('Gimnaziul "Decebal"', 'Chisinau'),
    ('Liceul Teoretic cu Profil de Arte "Elena Alistar"', 'Edinet'),
    ('Liceul Teoretic "Iulia Hasdeu"', 'Cahul'),
    ('Institutia Publica Liceul Teoretic "Liviu Rebreanu"', 'Cimislia'),
    ('Gimnaziul "Nicolae H. Costin"', 'Cimislia'),
    ('Gimnaziul nr.49', 'Chisinau'),
    ('Institutia Publica Liceul Teoretic "Tudor Vladimirescu"', 'Cahul'),

```

```

('Liceul Teoretic "Mircea cel Batran"', 'Edinet'),
('Liceul Teoretic "Bogdan Petriceicu Hasdeu"', 'Cahul'),
('Institutia Publica Liceul Teoretic "Traian"', 'Tiraspol'),
('Liceul Teoretic "Mihai Grecu"', 'Chisinau'),
('Institutia Publica Liceul Teoretic "Petru Movila"', 'Balti'),
('Liceul Teoretic "Nicolae Iorga"', 'Chisinau'),
('Liceul Teoretic "Mihai Eminescu"', 'Cahul'),
('Colegiul de Informatica', 'Cimislia'),
('Colegiul Financiar - Bancar', 'Balti'),
('Colegiul de Microelectronica si Tehnica de Calcul', 'Cimislia'),
('Colegiul de Ecologie', 'Edinet'),
('Colegiul de Industrie Usoara', 'Chisinau'),
('Colegiul Politehnic', 'Chisinau'),
('Colegiul de Transporturi', 'Balti'),
('Colegiul Tehnologic', 'Cimislia'),
('Colegiul de Constructii', 'Chisinau'),
('Colegiul Tehnic Feroviar', 'Balti'),
('Academia de Muzica, Teatru si Arte Plastice', 'Tiraspol'),
('Academia de Studii Economice', 'Edinet'),
('Universitatea Agrara de Stat', 'Cimislia'),
('Universitatea Pedagogica de Stat "Ion Creanga"', 'Tiraspol'),
('Universitatea de Stat de Medicina si Farmacie "Nicolae Testemitanu"', 'Cahul'),
('Universitatea Cooperativist - Comerciala', 'Cahul'),
('Universitatea de Stat', 'Edinet'),
('Universitatea Tehnica', 'Cahul'),
('Academia de Studii Economice', 'Cimislia'),
('Universitatea de Stat "Alecu Russo"', 'Cahul'),
('Universitatea de Stat "Bogdan Petriceicu Hasdeu"', 'Tiraspol'),
('Universitatea de Stat de Educatie Fizica si Sport', 'Cahul'),
('Academia de Politie "Stefan cel Mare"', 'Edinet'),
('Academia Militara a Fortelor Armate "Alexandru cel Bun"', 'Tiraspol'),
('Academia de Administrare Publica', 'Balti'),
('Universitatea de Stat "Grigore Tocmblac"', 'Balti'),
('Universitatea Libera Internationala', 'Chisinau'),
('Institutul de Stiinte Penale si Criminologie Aplicata', 'Cahul'),
('Institutul International de Management "IMI-NOVA"', 'Tiraspol'),
('Universitatea de Studii Politice si Economice Europene', 'Tiraspol'),
('Academia de Teologie Ortodoxa', 'Balti'),
('Scoala Profesionala nr.1', 'Chisinau');

```

GO

```

--Inserarea datelor in tabelul CATEGORIE_DANSURI
INSERT INTO CategorieDansuri (DenumireCategorie) VALUES
('Africane si afro-americane'),
('Dans ceremonial'),
('Discoteca / Dans electronic'),
('Hanpurian'),
('Dans gratuit si improvizat'),
('Dans istoric'),
('Latina / Ritm'),
('Dansuri de noutate si moft'),
('Dans social'),
('Dans de strada'),
('Dans de leagan'),
('Alte');

```

GO

```

--Inserarea datelor in tabelul TIP_DANSURI
INSERT INTO TipuriDans (DenumireTipDans, IdCategorie) VALUES

```

```

('Bolojo', 1),
('Boge', 1),
('Pas cu pas', 1),
('Kompa', 1),
('Dans de jazz', 1),
('Haka', 2),
('Kagura', 2),
('Dansuri rituale din China', 2),
('Dans Cham', 2),
('Drametse Ngacham', 2),
('Dans Kumha Pyakhan', 2),
('Dans profetic', 2),
('Boogaloo', 3),
('Cucui', 3),
('Taierea formelor', 3),
('Waacking', 3),
('Pas liber', 3),
('Hustle', 3),
('Dans Adhunik Hanpurian', 4),
('Contact improvizatie', 5),
('Improvizatie de dans', 5),
('Dans extatic', 5),
('Dans gratuit', 5),
('Balet / Franta', 6),
('Dans baroc', 6),
('Dans medieval', 6),
('Dans Polka', 6),
('Rumba', 7),
('Cha Cha', 7),
('Salsa', 7),
('Lambada', 8),
('Madison', 8),
('Dans solo', 9),
('Dans hip-hop', 10),
('Dansuri de club', 10),
('Breakdancing', 10),
('Uprock', 10),
('Dansul electro', 10),
('Balboa', 11),
('Dansul blues', 11),
('Bugg', 11),
('Jive', 11),
('Colegial Shag', 11),
('Single Swing', 11),
('Dansul acro', 12),
('Dansul focului', 12),
('K-Pop', 12),
('Dans liric', 12),
('Flamenco', 12),
('Dans vintage', 12);

GO

--Inserarea datelor in tabelul DANSATORI
INSERT INTO Dansatori (CodDansator, Nume, Prenume, Sex, DataNasterii, Experienta, IdTipDans) VALUES
    ('MR3716', 'Mccallum', 'Ryley', 'F', '2014-12-24', 2, 1),
    ('DD1783', 'Davidson', 'Darcie-Mae', 'F', '2014-06-20', 1, 17),
    ('RM3483', 'Rush', 'Manal', 'M', '2008-09-15', 6, 16),
    ('HC6318', 'Holland', 'Cain', 'M', '2008-06-24', 7, 28),

```

```

('IE1634', 'Ibarra', 'Emyr', 'F', '2013-10-10', 3, 30),
('CB7590', 'Corona', 'Bryce', 'M', '2002-06-10', 10, 18),
('HW3802', 'Huffman', 'Wade', 'M', '1991-04-01', 22, 38),
('IK4894', 'Irvine', 'Kyran', 'M', '2007-09-16', 4, 29),
('OK4361', 'ONeill', 'Kian', 'M', '2006-09-30', 7, 2),
('MP7371', 'McCormick', 'Phillipa', 'F', '2004-09-26', 11, 19),
('ES0047', 'Espinosa', 'Sonny', 'M', '2000-01-09', 2, 15),
('PD8341', 'Perkins', 'Derry', 'M', '2008-08-11', 5, 35),
('KA2531', 'Kirkland', 'Abigale', 'F', '2013-06-01', 1, 3),
('BB2610', 'Bains', 'Brenna', 'F', '2013-09-24', 2, 39),
('EC5281', 'Elliott', 'Cairon', 'M', '2001-09-25', 6, 50),
('HG4438', 'Holman', 'Greg', 'M', '1994-07-08', 18, 36),
('FF1732', 'Flower', 'Franciszek', 'M', '1993-03-19', 15, 49),
('BL7829', 'Bob', 'Loki', 'M', '2004-03-10', 2, 37),
('LJ8261', 'Le', 'Joan', 'M', '2004-01-09', 10, 48),
('JC5174', 'Jacobs', 'Charis', 'M', '1999-01-23', 3, 27),
('KZ8756', 'Kaiser', 'Zahra', 'F', '2010-05-21', 4, 4),
('WH0760', 'Wilkerson', 'Harvir', 'M', '1997-11-13', 5, 40),
('DM6019', 'Delgado', 'Mariah', 'F', '1983-05-17', 29, 31),
('GP3356', 'Guest', 'Padraig', 'F', '1993-03-21', 10, 43),
('DA0875', 'Derrick', 'Ariella', 'F', '1992-11-24', 11, 5),
('DN0348', 'Dickinson', 'Nia', 'F', '2009-09-25', 7, 12),
('MM5613', 'Mcghee', 'Mai', 'F', '1994-05-10', 12, 6),
('KR3891', 'Knowles', 'Reiss', 'M', '1990-06-07', 15, 20),
('QC6450', 'Quintero', 'Cristiano', 'M', '1992-10-25', 21, 13),
('NK4000', 'Nicholls', 'Kristen', 'F', '2011-01-15', 6, 44),
('GJ1936', 'Grey', 'Jett', 'M', '1998-10-16', 14, 14),
('MK0419', 'McCaffrey', 'Keelan', 'M', '1989-11-19', 23, 41),
('AK2274', 'Arias', 'Krista', 'F', '2013-05-01', 2, 21),
('NE8215', 'Neale', 'Eduard', 'M', '2011-11-08', 6, 46),
('BF2981', 'Beech', 'Flora', 'F', '1988-06-21', 24, 42),
('WC3210', 'Wilde', 'Colton', 'M', '1988-08-16', 20, 26),
('CL0815', 'Campos', 'Leila', 'F', '2011-05-13', 4, 22),
('HK7613', 'Harper', 'Karim', 'F', '2011-06-21', 5, 32),
('MM8033', 'Mendez', 'Miriam', 'F', '1997-09-09', 16, 9),
('RK2137', 'Redman', 'Karson', 'M', '2010-07-08', 7, 10),
('MM4671', 'Meza', 'Malika', 'F', '2011-06-21', 2, 24),
('GH9753', 'Goulding', 'Hira', 'F', '2014-10-22', 1, 34),
('PM4781', 'Preston', 'Maci', 'M', '2014-10-05', 1, 23),
('ME2375', 'Mcpherson', 'Emilis', 'F', '1998-10-24', 15, 45),
('BZ8349', 'Butt', 'Zeshan', 'M', '1987-11-20', 27, 47),
('JN9339', 'Joyce', 'Naseem', 'M', '1996-08-08', 18, 11),
('EA9201', 'Erickson', 'Alissa', 'F', '2015-08-21', 2, 33),
('PA9176', 'Piper', 'Antony', 'M', '1983-06-26', 32, 7),
('HB4085', 'Hanna', 'Brooklyn', 'M', '1986-01-17', 28, 25),
('SC4892', 'Shah', 'Chenai', 'M', '1987-11-04', 26, 8);

```

GO

```

--Inserarea datelor in tabelul DATE_DE_CONTACT
INSERT INTO DateDeContact (Email, Telefon, Adresa, IdInstitutie, IdDansator) VALUES
    ('kohlis@gmail.com', '+373 562 31 237', 'Chisinau', 1, 1),
    ('fukuchi@gmail.com', '+373 562 52 426', 'Truseni', 2, 14),
    ('mgreen@mail.com', '+373 680 74 925', 'Cricova', 3, 13),
    ('sacraver@yahoo.com', '+373 640 72 956', 'Chisinau', 28, 27),
    ('fatelk@mail.com', '+373 562 27 132', 'Bacoi', 35, 36),
    ('fallorn@gmail.com', '+373 562 22 983', 'Floresti', 34, 45),
    ('kingma@mail.com', '+373 776 29 903', 'Floresti', 4, 26),
    ('oster@yahoo.com', '+373 562 72 844', 'Bacoi', 40, 16),
    ('afeldspar@gmail.com', '+373 562 23 908', 'Cricova', 15, 37),

```

```

('novanet@yahoo.com', '+373 562 46 693', 'Chisinau', 38, 46),
('johnbob@yahoo.com', '+373 562 95 329', 'Ialoveni', 41, 35),
('qrcza@gmail.com', '+373 637 19 047', 'Balti', 13, 43),
('fwitness@gmail.com', '+373 608 69 892', 'Balti', 5, 47),
('frode@mail.com', '+373 776 31 333', 'Balti', 31, 24),
('mjewell@gmail.com', '+373 624 40 782', 'Cricova', 6, 34),
('jtorkbob@yahoo.com', '+373 645 32 252', 'Balti', 23, 15),
('kudra@gmail.com', '+373 562 89 723', 'Truseni', 29, 44),
('torgox@mail.com', '+373 562 86 817', 'Chisinau', 7, 2),
('keijser@gmail.com', '+373 782 05 383', 'Floresti', 11, 12),
('geekoid@gmail.com', '+373 635 44 508', 'Cricova', 33, 25),
('avalon@gmail.com', '+373 562 84 938', 'Ialoveni', 39, 28),
('aibrahim@gmail.com', '+373 562 28 191', 'Balti', 32, 48),
('osrin@yahoo.com', '+373 646 94 463', 'Truseni', 17, 4),
('jsbach@gmail.com', '+373 562 37 575', 'Bacioi', 22, 6),
('kalpol@gmail.com', '+373 711 81 138', 'Chisinau', 49, 7),
('ahuillet@gmail.com', '+373 562 64 760', 'Ialoveni', 45, 23),
('jdhedden@gmail.com', '+373 562 79 334', 'Cricova', 8, 3),
('paley@gmail.com', '+373 621 21 017', 'Balti', 36, 17),
('stomv@yahoo.com', '+373 638 37 382', 'Chisinau', 25, 19),
('privcan@gmail.com', '+373 654 99 419', 'Floresti', 37, 31),
('rgarton@gmail.com', '+373 647 36 276', 'Balti', 9, 38),
('seanq@mail.com', '+373 562 80 216', 'Truseni', 10, 5),
('maratb@gmail.com', '+373 788 20 440', 'Balti', 44, 49),
('jelmer@mail.com', '+373 636 38 978', 'Ialoveni', 24, 18),
('attwood@gmail.com', '+373 562 13 980', 'Cricova', 48, 39),
('tubestea@gmail.com', '+373 562 05 948', 'Chisinau', 47, 50),
('mosses@gmail.com', '+373 562 05 948', 'Chisinau', 10, 8),
('sopwith@gmail.com', '+373 777 52 946', 'Bacioi', 12, 41),
('vganesh@gmail.com', '+373 562 61 664', 'Cricova', 42, 29),
('valdez@mail.com', '+373 790 90 897', 'Truseni', 14, 42),
('timtroyr@gmail.com', '+373 562 32 022', 'Chisinau', 16, 33),
('scarlet@yahoo.com', '+373 792 20 421', 'Balti', 18, 40),
('kostas@mail.com', '+373 628 65 343', 'Ialoveni', 26, 9),
('delpino@yahoo.ca', '+373 562 08 707', 'Truseni', 43, 22),
('pjackson@yahoo.com', '+373 760 64 704', 'Balti', 46, 32),
('frostman@gmail.com', '+373 717 48 740', 'Cricova', 19, 30),
('pereinar@gmail.com', '+373 786 24 451', 'Chisinau', 30, 20),
('alastair@mail.com', '+373 562 22 373', 'Bacioi', 20, 10),
('dwheeler@gmail.com', '+373 562 85 287', 'Truseni', 21, 21),
('bjornk@gmail.com', '+373 562 42 600', 'Chisinau', 50, 11);

```

GO

```

--Inserarea datelor in tabelul CONCURSURI
INSERT INTO Concursuri (Denumire, AdresaDesfasurarii, Durata, DataDesfasuratii)
VALUES
    ('Marea Neagra', 'Chisinau', '01:30', '2021-10-30'),
    ('Fantezie si Talent', 'Causeni', '02:30', '2021-10-13'),
    ('AllStyles Dance Competition', 'Drochia', '02:00', '2021-06-24'),
    ('Tinere Talente', 'Straseni', '01:30', '2021-05-29'),
    ('Ghiocelul de Argint', 'Orhei', '02:00', '2021-06-09'),
    ('Jocuri Dinamice si Dans', 'Vulcanesti', '02:30', '2021-05-27'),
    ('Dincolo de Stele', 'Ceadir-Lunga', '01:30', '2021-10-10'),
    ('Aplauze Talent', 'Chisinau', '01:30', '2021-10-09'),
    ('Sisteme de Fulgi', 'Chisinau', '02:30', '2021-12-30'),
    ('Puterea Stelelor', 'Chisinau', '02:00', '2021-08-20');

```

GO

```

--Inserarea datelor in tabelul ACTE
INSERT INTO Acte (IdDansator, IdConcurs) VALUES
(4, 1),
(22, 2),
(39, 4),
(3, 3),
(49, 6),
(40, 8),
(21, 10),
(37, 9),
(50, 7),
(38, 3),
(2, 3),
(1, 6),
(36, 5),
(48, 3),
(23, 9),
(10, 10),
(20, 1),
(41, 5),
(46, 1),
(11, 7),
(19, 9),
(28, 8),
(6, 5),
(18, 6),
(34, 7),
(5, 10),
(27, 10),
(35, 8),
(29, 1),
(24, 9),
(45, 10),
(47, 1),
(15, 4),
(12, 8),
(44, 2),
(9, 5),
(26, 7),
(30, 3),
(14, 6),
(17, 2),
(33, 4),
(32, 7),
(13, 9),
(25, 4),
(43, 8),
(7, 4),
(16, 2),
(42, 6),
(8, 5),
(31, 2);

GO

--Inserarea datelor in tabelul PREMII
INSERT INTO Premii (Locul, PremiuBanesc, IdDansator, IdConcurs) VALUES
('I', 1000, 4, 1),
('I', 1500, 8, 5),
('I', 3000, 2, 3),

```

```

('I', 2500, 42, 6),
('I', 2000, 23, 9),
('I', 1000, 39, 4),
('I', 1500, 44, 2),
('I', 3000, 28, 8),
('I', 2500, 47, 10),
('I', 2000, 11, 7),

('II', 2000, 9, 5),
('II', 500, 22, 2),
('II', 1000, 35, 8),
('II', 1750, 20, 1),
('II', 2000, 49, 6),
('II', 2000, 37, 9),
('II', 500, 30, 3),
('II', 1000, 5, 10),
('II', 1750, 50, 7),
('II', 2000, 33, 4),

('III', 750, 7, 4),
('III', 1500, 40, 8),
('III', 1500, 36, 5),
('III', 750, 32, 7),
('III', 1500, 17, 2);

GO

--Inserarea datelor in tabelul ADMINISTRATORI
INSERT INTO Utilizatori(Username, Parola, Email, EsteAdministrator) VALUES
    ('Guest', NULL, NULL, 0),
    ('Anastasia', '1234', 'exemplu@gmail.com', 1),
    ('Lucas', '4321', 'lucas@gmail.com', 1);
GO

--PROCEDURI
--Procedura ce selecteaza denumirea concursurilor
CREATE OR ALTER PROCEDURE selectCompetition
AS
    SELECT Denumire FROM Concursuri
GO

--Procedura ce selecteaza dansatori
CREATE OR ALTER PROCEDURE selectDancers
AS
    SELECT * FROM Dansatori
GO

--Procedura ce selecteaza denumirea categoriilor de dans
CREATE OR ALTER PROCEDURE selectCategory
AS
    SELECT DenumireCategorie FROM CategorieDansuri
GO

--Procedura ce selecteaza denumirea tipurilor de dans

```

```

CREATE OR ALTER PROCEDURE selectDanceType
AS
    SELECT DenumireTipDans FROM TipuriDans
GO

--Procedura ce selecteaza participantii in functie de codul personal
CREATE OR ALTER PROCEDURE selectDancersByCode
@Code VARCHAR(6)
AS
    SELECT * FROM Dansatori
    WHERE CodDansator = @Code
GO

--Procedura ce afiseaza username-ul din vedere
CREATE OR ALTER PROCEDURE selectFromView
AS
    SELECT EsteAdministrator FROM currentUser
GO

--Procedura ce verifica daca username si parola introdusa,
--corespunde cu datele din tabelul utilizatori
CREATE OR ALTER PROCEDURE checkUser
@Username VARCHAR(50),
@Parola VARCHAR(25)
AS
    SELECT Username FROM Utilizatori
    WHERE Username = @Username AND
          Parola = @Parola
GO

--ITEMUL 1
--Procedura ce inregistreaza un dansator nou
CREATE OR ALTER PROCEDURE insertDancer
@NumeInstitutie VARCHAR(100),
@AdresaInstitutie VARCHAR(50),

@DenumireTipDans VARCHAR(50),

@CodDansator VARCHAR(6),
@NumeDansator VARCHAR(20),
@PrenumeDansator VARCHAR(20),
@SexDansator VARCHAR(1),
@DataNasterii DATE,
@Experienta INT,

@DenumireConcurs VARCHAR(100),

@email VARCHAR(50),
@Telefon VARCHAR(15),
@AdresaDansator VARCHAR(50)
AS
    DECLARE @IdInstitutie INT,
            @IdDansator INT

    IF NOT EXISTS((SELECT NumeInstitutie, Adresa FROM Institutii
                   WHERE NumeInstitutie = @NumeInstitutie AND
                         Adresa = @AdresaInstitutie))
    BEGIN
        INSERT INTO Institutii (NumeInstitutie, Adresa) VALUES

```

```

        (@NumeInstitutie, @AdresaInstitutie);

    SET @IdInstitutie = (SELECT MAX(IdInstitutie) FROM Institutii);
END
ELSE
BEGIN
    SET @IdInstitutie = (SELECT IdInstitutie FROM Institutii
                          WHERE NumeInstitutie = @NumeInstitutie AND
                                Adresa = @AdresaInstitutie);
END

    INSERT INTO Dansatori (CodDansator, Nume, Prenume, Sex, DataNasterii,
Experienta, IdTipDans) VALUES
    (@CodDansator, @NumeDansator, @PrenumeDansator, @SexDansator,
@DataNasterii, @Experienta, (SELECT IdTipDans FROM TipuriDans
                               WHERE DenumireTipDans = @DenumireTipDans));

    SET @IdDansator = (SELECT MAX(IdDansator) FROM Dansatori);

    INSERT INTO DateDeContact (Email, Telefon, Adresa, IdInstitutie, IdDansator)
VALUES
    (@Email, @Telefon, @AdresaDansator, @IdInstitutie, @IdDansator);

    INSERT INTO Acte (IdDansator, IdConcurs) VALUES
    (@IdDansator, (SELECT IdConcurs FROM Concursuri
                   WHERE Denumire = @DenumireConcurs));
GO

--Procedura ce insereaza premiul unui dansator
CREATE OR ALTER PROCEDURE insertPrize
@Locul VARCHAR(20),
@PremiuBanesc MONEY,
@DenumireConcurs VARCHAR(100)
AS
    DECLARE @IdDansator INT,
            @IdConcurs INT

    INSERT INTO Premii (Locul, PremiuBanesc, IdDansator, IdConcurs) VALUES
    (@Locul, @PremiuBanesc, (SELECT MAX(IdDansator) FROM Dansatori),
     (SELECT IdConcurs FROM Concursuri
      WHERE Denumire = @DenumireConcurs));
GO

--ITEMUL 2
--Procedura ce exclude un dansator, in functie de codul introdus
CREATE OR ALTER PROCEDURE deleteDancer
@CodDansator VARCHAR(60)
AS
    DELETE FROM Dansatori
    WHERE CodDansator = @CodDansator
GO

--ITEMUL 3
--Procedura ce selecteaza participantul cel mai tanar si ce tip de dans practica
CREATE OR ALTER PROCEDURE youngestDancer
AS
    SELECT CodDansator, Nume, Prenume, DataNasterii, DenumireTipDans FROM Dansatori
    INNER JOIN TipuriDans

```

```

        ON Dansatori.IdTipDans = TipuriDans.IdTipDans
    WHERE DataNasterii = (SELECT MAX(DataNasterii) FROM Dansatori)
GO

--Procedura ce selecteaza participantul cel mai in varsta si ce tip de dans
practica
CREATE OR ALTER PROCEDURE oldestDancer
AS
    SELECT CodDansator, Nume, Prenume, DataNasterii, DenumireTipDans FROM Dansatori
        INNER JOIN TipuriDans
            ON Dansatori.IdTipDans = TipuriDans.IdTipDans
    WHERE DataNasterii = (SELECT MIN(DataNasterii) FROM Dansatori)
GO

--ITEMUL 4
--Procedura ce ordoneaza ascendent toti dansatorii, dupa varsta
CREATE OR ALTER PROCEDURE sortAscAllDancersByDate
AS
    SELECT Nume, Prenume, DataNasterii FROM Dansatori
        ORDER BY DataNasterii DESC
GO

--Procedura ce ordoneaza ascendent dansatorii dintr-un concurs, dupa varsta
CREATE OR ALTER PROCEDURE sortDancersASCFromOneCompetitionByDate
@DenumireConcurs VARCHAR(100)
AS
    SELECT Nume, Prenume, DataNasterii FROM Dansatori
        INNER JOIN Acte
            ON Acte.IdDansator = Dansatori.IdDansator
                INNER JOIN Concursuri
                    ON Concursuri.IdConcurs = Acte.IdConcurs
    WHERE Denumire = @DenumireConcurs
        ORDER BY DataNasterii DESC
GO

--ITEMUL 5
--Procedura ce selecteaza codul, numele, prenumele, data nasterii si genul
--celei mai tinere persoane de sex feminin
CREATE OR ALTER PROCEDURE youngestFemaleDancer
AS
    SELECT CodDansator, Nume, Prenume, DataNasterii, Sex FROM Dansatori
    WHERE DataNasterii = (SELECT MAX(DataNasterii) FROM Dansatori) AND
        Sex = 'F'
GO

--ITEMUL 6
--Procedura ce selecteaza lista participantilor la categoria de dans indicata
CREATE OR ALTER PROCEDURE selectDancersByCategory
@DenumireCategoria VARCHAR(50)
AS
    SELECT Dansatori.*, CategorieDansuri.DenumireCategoria FROM Dansatori
        INNER JOIN TipuriDans
            ON Dansatori.IdTipDans = TipuriDans.IdTipDans
                INNER JOIN CategorieDansuri
                    ON CategorieDansuri.IdCategoria = TipuriDans.IdCategoria
    WHERE CategorieDansuri.DenumireCategoria = @DenumireCategoria
GO

--ITEMUL 7

```

```

--Procedura ce afiseaza dansatorii barbati pentru fiecare categorie de dans
CREATE OR ALTER PROCEDURE selectMaleDancersByCategory
@DenumireCategoria VARCHAR(50)
AS
    SELECT DataNasterii FROM Dansatori
        INNER JOIN TipuriDans
            ON Dansatori.IdTipDans = TipuriDans.IdTipDans
                INNER JOIN CategorieDansuri
                    ON CategorieDansuri.IdCategorie = TipuriDans.IdCategorie
    WHERE CategorieDansuri.DenumireCategoria = @DenumireCategoria AND
        Sex = 'M'
GO

--ITEMUL 8
--Procedura ce ajuta la exportarea intr-un fisier MS Excel lista tuturor
--participantelor cu varsta sub o varsta introdusa
CREATE OR ALTER PROCEDURE selectFemaleDancersUnderIntroducedYears
@DataIntrodusa DATE
AS
    SELECT CodDansator, Nume, Prenume, Experienta, DataNasterii FROM Dansatori
    WHERE DataNasterii > @DataIntrodusa AND
        Sex = 'F'
GO

--ITEMUL 9
--Vederea ce creeaza o vedere ce contine date de contact al fiecarui dansator
CREATE OR ALTER PROCEDURE createView
AS
    GO
    CREATE OR ALTER VIEW selectDateDeContact (CodDansator, Nume, Prenume, Email,
    Telefon, Adresa)
        AS SELECT CodDansator, Nume, Prenume, Email, Telefon, Adresa FROM DateDeContact
            INNER JOIN Dansatori
                ON Dansatori.IdDansator = DateDeContact.IdDansator
GO

--Procedura ce preia datele din vedere
CREATE OR ALTER PROCEDURE selectContactData
AS
    SELECT * FROM selectDateDeContact
GO

--ITEMUL 10
--Procedura ce afiseaza lista participantilor si institutia unde isi fac studiile
CREATE OR ALTER PROCEDURE selectDancersAndInstitution
AS
    SELECT CodDansator, Nume, Prenume, NumeInstitutie FROM Dansatori
        INNER JOIN DateDeContact
            ON Dansatori.IdDansator = DateDeContact.IdDansator
                INNER JOIN Institutii
                    ON DateDeContact.IdInstitutie = Institutii.IdInstitutie
GO

--ITEMUL 11
--Procedura ce afiseaza codul, numele, prenumele, locul dansatorului,
--in functie de concursul selectat
CREATE OR ALTER PROCEDURE selectDancerPrizeByCompetition
@DenumireConcurs VARCHAR(100)
AS

```

```

SELECT CodDansator, Nume, Prenume, Locul FROM Dansatori
    INNER JOIN Premii
        ON Dansatori.IdDansator = Premii.IdDansator
            INNER JOIN Concursuri
                ON Concursuri.IdConcurs = Premii.IdConcurs
WHERE Denumire = @DenumireConcurs
ORDER BY Locul
GO

--ITEMUL 12
--Procedura ce selecteaza participantii in functie de denumirea concursului
CREATE OR ALTER PROCEDURE selectDancersByCompetitionName
@CompetitionName VARCHAR(100)
AS
    SELECT Nume, Prenume FROM Dansatori
        INNER JOIN Acte
            ON Acte.IdDansator = Dansatori.IdDansator
                INNER JOIN Concursuri
                    ON Concursuri.IdConcurs = Acte.IdConcurs
WHERE Concursuri.Denumire = @CompetitionName
GO

```

Conținutul tabelelor

Tabelul Instituții

Tabelul Instituții inițial conține 50 de înregistrări, însă, pentru a economisi spațiu, vor fi indicate primele 10 înregistrări (mai detaliat datele sunt indicate în „scenariul bazei de date ConcursDeDans”).

	IdInstitutie	NumelInstitutie	Adresa
1	1	Liceul Teoretic "Pro Succes"	Chisinau
2	2	Liceul Teoretic cu Profil Sportiv "Gloria"	Balti
3	3	Liceul Teoretic "Rambam"	Tiraspol
4	4	Gimnaziul "Galata"	Cahul
5	5	Gimnaziul "Decebal"	Chisinau
6	6	Liceul Teoretic cu Profil de Arte "Elena Alistar"	Edinet
7	7	Liceul Teoretic "Iulia Hasdeu"	Cahul
8	8	Institutia Publica Liceul Teoretic "Liviu Rebreanu"	Cimișlia
9	9	Gimnaziul "Nicolae H. Costin"	Cimișlia
10	10	Gimnaziul nr.49	Chisinau

Tabelul CategorieDansuri

Tabelul CategorieDansuri inițial conține 12 de înregistrări, însă, pentru a economisi spațiu, vor fi indicate primele 10 înregistrări (mai detaliat datele sunt indicate în „scenariul bazei de date ConcursDeDans”).

	IdCategorie	DenumireCategorie
1	1	Africane si afro-americane
2	2	Dans ceremonial
3	3	Discoteca / Dans electronic
4	4	Hanpurian
5	5	Dans gratuit si improvizat
6	6	Dans istoric
7	7	Latina / Ritm
8	8	Dansuri de noutate si moft
9	9	Dans social
10	10	Dans de strada

Tabelul TipuriDans

Tabelul TipuriDans inițial conține 50 de înregistrări, însă, pentru a economisi spațiu, vor fi indicate primele 10 înregistrări (mai detaliat datele sunt indicate în „scenariul bazei de date ConcursDeDans”).

	IdTipDans	DenumireTipDans	IdCategorie
1	1	Bolojo	1
2	2	Boge	1
3	3	Pas cu pas	1
4	4	Kompa	1
5	5	Dans de jazz	1
6	6	Haka	2
7	7	Kagura	2
8	8	Dansuri rituale din China	2
9	9	Dans Cham	2
10	10	Drametse Ngacham	2

Tabelul Dansatori

Tabelul Dansatori inițial conține 50 de înregistrări, însă, pentru a economisi spațiu, vor fi indicate primele 10 înregistrări (mai detaliat datele sunt indicate în „scenariul bazei de date ConcursDeDans”).

	IdDansator	CodDansator	Nume	Prenume	Sex	DataNasterii	Experienta	IdTipDans
1	1	MR3716	Mccallum	Ryley	F	2014-12-24	2	1
2	2	DD1783	Davidson	Darcie-Mae	F	2014-06-20	1	17
3	3	RM3483	Rush	Manal	M	2008-09-15	6	16
4	4	HC6318	Holland	Cain	M	2008-06-24	7	28
5	5	IE1634	Ibarra	Emyr	F	2013-10-10	3	30
6	6	CB7590	Corona	Bryce	M	2002-06-10	10	18
7	7	HW3802	Huffman	Wade	M	1991-04-01	22	38
8	8	IK4894	Irvine	Kyran	M	2007-09-16	4	29
9	9	OK4361	ONeill	Kian	M	2006-09-30	7	2
10	10	MP7371	Mccormick	Phillipa	F	2004-09-26	11	19

Tabelul DateDeContact

Tabelul DateDeContact inițial conține 50 de înregistrări, însă, pentru a economisi spațiu, vor fi indicate primele 10 înregistrări (mai detaliat datele sunt indicate în „scenariul bazei de date ConcursDeDans”).

	IdDateDeContact	Email	Telefon	Adresa	IdInstitutie	IdDansator
1	1	kohlis@gmail.com	+373 562 31 237	Chisinau	1	1
2	2	fukuchi@gmail.com	+373 562 52 426	Truseni	2	14
3	3	mgreen@mail.com	+373 680 74 925	Cricova	3	13
4	4	sacraver@yahoo.com	+373 640 72 956	Chisinau	28	27
5	5	fatelk@mail.com	+373 562 27 132	Bacioi	35	36
6	6	fallorn@gmail.com	+373 562 22 983	Floresti	34	45
7	7	kingma@mail.com	+373 776 29 903	Floresti	4	26
8	8	oster@yahoo.com	+373 562 72 844	Bacioi	40	16
9	9	afeldspar@gmail.com	+373 562 23 908	Cricova	15	37
10	10	novanet@yahoo.com	+373 562 46 693	Chisinau	38	46

Tabelul Concursuri

	IdConcurs	Denumire	AdresaDesfasurarii	Durata	DataDesfasurarii
1	1	Marea Neagra	Chisinau	01:30:00.0000000	2021-10-30
2	2	Fantezie si Talent	Causeni	02:30:00.0000000	2021-10-13
3	3	AllStyles Dance Competition	Drochia	02:00:00.0000000	2021-06-24
4	4	Tinere Talente	Straseni	01:30:00.0000000	2021-05-29
5	5	Ghiocelul de Argint	Orhei	02:00:00.0000000	2021-06-09
6	6	Jocuri Dinamice si Dans	Vulcanesti	02:30:00.0000000	2021-05-27
7	7	Dincolo de Stele	Ceadir-Lunga	01:30:00.0000000	2021-10-10
8	8	Aplauze Talent	Chisinau	01:30:00.0000000	2021-10-09
9	9	Sisteme de Fulgi	Chisinau	02:30:00.0000000	2021-12-30
10	10	Puterea Stelelor	Chisinau	02:00:00.0000000	2021-08-20

Tabelul Acte

Tabelul Acte inițial conține 50 de înregistrări, însă, pentru a economisi spațiu, vor fi indicate primele 10 înregistrări (mai detaliat datele sunt indicate în „scenariul bazei de date ConcursDeDans”).

	IdAct	IdDansator	IdConcurs
1	1	4	1
2	2	22	2
3	3	39	4
4	4	3	3
5	5	49	6
6	6	40	8
7	7	21	10
8	8	37	9
9	9	50	7
10	10	38	3

Tabelul Premii

Tabelul Premii inițial conține 30 de înregistrări, însă, pentru a economisi spațiu, vor fi indicate primele 10 înregistrări (mai detaliat datele sunt indicate în „scenariul bazei de date ConcursDeDans”).

	IdPremiu	Locul	PremiuBanesc	IdDansator	IdConcurs
1	1	I	1000.00	4	1
2	2	I	1500.00	8	5
3	3	I	3000.00	2	3
4	4	I	2500.00	42	6
5	5	I	2000.00	23	9
6	6	I	1000.00	39	4
7	7	I	1500.00	44	2
8	8	I	3000.00	28	8
9	9	I	2500.00	47	10
10	10	I	2000.00	11	7

Tabelul Utilizatori

	IdUtilizator	Username	Parola	Email	EsteAdministrator
1	1	Guest	NULL	NULL	0
2	2	Anastasia	1234	exemplu@gmail.com	1
3	3	Lucas	4321	lucas@gmail.com	1

Concluzii

Consider că, practica efectuată reprezintă o experiență, care îmi va fi utilă nu doar la moment, ci și în viitor. Pot spune cu încredere că practica a oferit pentru mine șansa de a testa cunoștințele acumulate pe parcursul acestui an și, de asemenea, ce e mai important, oferă oportunitatea de a studia noi cunoștințe, acumulate din diferite resurse care sunt deschise pentru orice student, și implementarea acestora în viitor, la muncă.

Planul de activitate pe perioada de practică, a fost formată astfel:

După ce mi s-a fost prezentată sarcina, am început să lucrez asupra reprezentarea generală a bazei de date și a aplicației. După am început să implementez această reprezentare. Sarcina a fost repartizată astfel: crearea unei aplicații vizuale care a fost realizată în limbajul de programare C#, prin intermediul mediului de dezvoltare „Visual Studio”, crearea bazei de date ce va interacționa cu aplicație și utilizator, conexiunea la această bază de date, crearea diagramei E-R pentru baza de date, completarea agendei și realizarea documentației în MS Word. Cel mai captivant lucru a fost faptul că am fost în stare să adăugăm orice ne dormi, astfel utilizând creativitatea la maxim.

Practica am început cu studierea materialelor prezentate pe Moodle, de către conducătorul stagiu de practică, doamna Diana Nebunelea. De asemenea, pentru această sarcină, am împărțit sarcina astfel, pentru a mă încadra în timpul oferit pentru practică, pentru a lucra eficient și destul de rapid. După analizarea materialului, am început lucru asupra creării schemei abstracte a aplicației și a bazei de date, iar după, și aplicarea acelor scheme în proiect. Apoi, am început să lucrez asupra scrierii codului conform sarcinilor propuse, cât și implementarea sarcinilor proprii, care mi s-au părut interesante de realizat. Aplicația creată este bazată pe concursuri de dans. Datele necesare programului sunt preluate din baza de date „ConcursDeDans”. După care acestea sunt prelucrate de către program, pentru a fi realizate subpunctele din meniu. De asemenea, în baza de date se păstrează toate datele și modificările efectuate de către aplicație.

Datorită meniului specific, aplicația rulează până când utilizatorul nu va închide total aplicația. Astfel, un subpunct poate fi vizitat de mai multe ori, iar utilizatorul (administratorul) se poate deplasa cu ușurință prin aplicație. De asemenea, pentru ca aplicația să funcționeze mai bine, am eliminat multe excepții, pot spune chiar toate, care pot apărea pe parcursul rulării acesteia. După finisarea elaborării aplicației și a bazei de date, am început să lucrez asupra realizarea crearea diagramei Entitate-Relație a bazei de date și a raportului.

Practica dată, mi-a oferit posibilitatea de a lucra la un proiect mai complex, ce permite implementarea propriilor fantezii. Datorită unui program mai complicat, am înțeles mai bine cum să structurez datele și să optimizez codul, pentru a exclude toate erorile care pot apărea pe parcurs.

Crearea unei interfețe de logare pentru utilizatori diferiți, crearea unui meniu plăcut și user-friendly, inserarea/excluderea/preluarea datelor din baza de date, prelucrarea datelor și altele au fost prezente în aplicație, conform condițiilor și imaginației personale.

Pe parcurs ce îndeplineam cerințele condiției, trebuia să fiu atentă la posibilele erori și să le corectez înainte ca acestea să apară. Elaborarea aplicației date, mi-a oferit cunoștințe noi în programare și utilizarea mai eficientă a codului, care îmi vor fi utile în viitoarea mea carieră de programator.

În concluzii pot spune că mi-a plăcut să lucrez cu limbajul de programare C# și cu aplicațiile vizuale, sunt mulțumită de practica pe care am realizat-o. Consider că o astfel de practică la sfârșitul unui an de studii este necesară pentru verificarea cunoștințelor unui viitor programator, ce îl va motiva să fie mai responsabil, mai concentrat asupra condiției propuse și îl va motiva spre noi succese.

Bibliografie / Webografie

- Platforma Moodle: <https://moodle1.ceiti.md/course/view.php?id=2180>
- Conexiunea la baze de date:
<https://www.youtube.com/watch?v=Et2khGnrIqc&list=LL&index=4&t=1s>,
<https://www.connectionstrings.com/sql-server/>
- Imagini: <https://icon-icons.com>, <https://www.flaticon.com>
- Inversarea culorilor într-o imagine: <https://pinetools.com/invert-image-colors>
- Evidențierea codului: <http://hilite.me>
- Exportarea datei dintr-un fișier .csv în MS Excel:
<https://www.syncfusion.com/blogs/post/export-data-from-sql-server-to-excel-in-c-sharp.aspx>
- Lista categoriilor de dans: https://wikicro.icu/wiki/List_of_dance_style_categories

Link către drive:

<https://drive.google.com/drive/folders/1gfhd8ceBjIQTc2hm7FfWHldLnCl4mdPJ>