

Liceul Teoretic "Alexandru Papiu Ilarian" Dej

SOFT EDUCATIONAL ISTORIA SCLAVIEI

Lucrare de atestat profesional în informatică

Elev:

Negru Diana-Maria

Profesor coordonator:

Prof. Mihaela-Elena Capangi

Dej, 2021

Cuprins

Cuprins.....	2
Capitolul 1. Prezentarea generală a temei.....	4
Capitolul 2. Resurse hardware și software necesare.....	6
Capitolul 3. Realizarea aplicației	7
3.1. Fereastra FrmLogare	8
3.2. Fereastra FrmContNou.....	12
3.3. Fereastra FrmMain.....	16
3.4. Fereastra FrmLectii.....	25
3.5. Fereastra FrmTeste.....	29
3.6. Fereastra FrmAdaugaLectie.....	50
3.7. Fereastra FrmModificaLectie.....	58
3.8. Fereastra FrmModifOrdCapitole	65
3.9. Fereastra FrmModifOrdLectii.....	71
3.10. Fereastra FrmSergeLectie.....	78
3.11. Fereastra FrmAdaugaIntrebari.....	84
3.12. Fereastra FrmModificaIntrebari	95
3.13. Fereastra FrmSergeIntrebari.....	114
3.14. Fereastra FrmAdaugaTest	123
3.15. Fereastra FrmModificaTest	131
3.16. Fereastra FrmSergeTest.....	140
3.17. Fereastra FrmSchimbaParola.....	144
3.18. Fereastra FrmNotificari	147
3.19. Fereastra FrmJoc	154
3.20. Baze de date.....	165
3.21. Clase	167
Capitolul 4. Utilizarea aplicației	171
Capitolul 5. Extinderi posibile	184
Bibliografie	185

Capitolul 1. Prezentarea generală a temei

Introducerea tehnologiei informației în toate domeniile de activitate a influențat și sistemul de învățământ. Utilizarea calculatorului în procesul de predare – învățare – evaluare reprezintă o metodă modernă de activitate didactică, interactivă și dirijată. Impulsionarea și interesul major al copiilor pentru acest mijloc didactic, cât și datorită noutății și posibilităților multiple de utilizare, am căutat o modalitate de a crea o formă de învățare utilă și plăcută pentru copii, care să-i ajute să se dezvolte individual, iar în același timp, să fie o metodă care să ajute profesorii să transmită informația elevilor într-un mod simplu și rapid.

Aplicația “Istoria Sclaviei” este un soft educațional destinat atât elevilor care doresc să acumuleze informații într-o manieră ușoara și plăcută, care este des utilizată în ziua de astăzi de către tineri, cât și profesorilor ca și material didactic și platformă prin intermediul căreia să transmită lecțiile într-un mod ușor elevilor.

În momentul de față, aplicația cuprinde 12 lecții grupate în 6 capitole:

- Începuturile:
 - Explorarea Africii
 - Începuturile sclaviei
- Sclavul și sclavia:
 - Viața sclavilor
 - Statutul sclavilor
- Emanciparea:
 - Mișcarea pentru drepturile civile
 - Emanciparea în epoca contemporană
- Personalități emblematice:
 - Martin Luther King
 - Daisy Bates
 - John Lewis
- Literatură și artă
 - “12 ani de sclavie” de Solomon Northup
 - Muzeul Național Peleș

- Controverse
 - Black Lives Matter

Pe lângă acestea, aplicația oferă și posibilitatea testării cunoștințelor pe temele prezentate prin intermediul testelor adăugate de către profesori. Acestea sunt corectate automat de către aplicație, afișând punctajul obținut și răspunsurile corectate la care elevul a răspuns eronat.

Capitolul 2. Resurse hardware și software necesare

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

- Windows 10
- XAMPP v3.3.0
- .NetFramework 4.8
- Procesor Intel Core i3
- Memorie internă (RAM) 4GB
- Memorie externă (hard) 50GB

Capitolul 3. Realizarea aplicației

Aplicația a fost dezvoltată cu ajutorul .NetFramework, Microsoft Visual C# și XAMPP

Aplicația este construită din 19 ferestre :

- FrmLogare – fereastra destinată logării în contul de elev sau profesor
- FrmContNou – fereastra destinată creării unui nou cont
- FrmMain – fereastra principală a aplicației, în care este prezentat meniul
- FrmLectii – fereastra destinată afișării lecțiilor
- FrmTeste – fereastra destinată încărcării testelor
 - FrmAdaugaLectie – fereastra destinată profesorilor pentru a putea încărca lecții noi
 - FrmModificaLectie – fereastra destinată profesorilor pentru a putea modifica lecțiile existente
 - FrmModifOrdCapitole – fereastra destinată profesorilor pentru a putea modifica ordinea de afișare a capitolelor
 - FrmModifOrdLectii – fereastra destinată profesorilor pentru a putea modifica ordinea de afișare a lecțiilor
- FrmSergeLectie – fereastra destinată profesorilor pentru a putea șterge lecțiile existente
- FrmAdaugaIntrebari – fereastra destinată profesorilor pentru a putea adăuga întrebări pentru teste
 - FrmModificaIntrebari – fereastra destinată profesorilor pentru a putea modifica întrebările existente
 - FrmSergeIntrebari – fereastra destinată profesorilor pentru a putea șterge întrebările existente
 - FrmAdaugaTest – fereastra destinată profesorilor pentru a putea adăuga teste
 - FrmModificaTest – fereastra destinată profesorilor pentru a putea modifica testele existente
 - FrmSergeTest – fereastra destinată profesorilor pentru a putea șterge testele existente
- FrmSchimbaParola – fereastra destinată schimbării parolei
- FrmNotificari – fereastra destinată pentru vizualizarea notificărilor
- FrmJoc – fereastra destinată pentru rezolvarea unui joc de puzzle

3.1. Fereastra FrmLogare

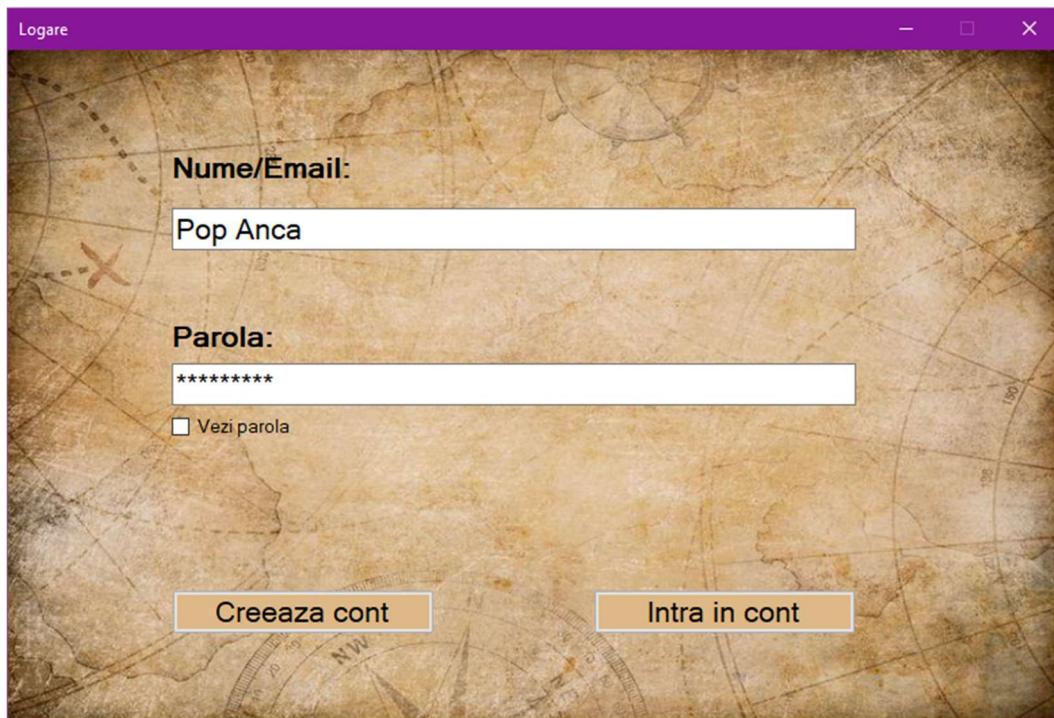
Această fereastră va fi prima care va apărea pe ecran la rularea aplicației pentru ca utilizatorul să se logheze, într-un cont deja existent. Aceste cont poate fi de două tipuri: cont de elev și cont de profesor, care diferă prin drepturile pe care le are utilizatorul.

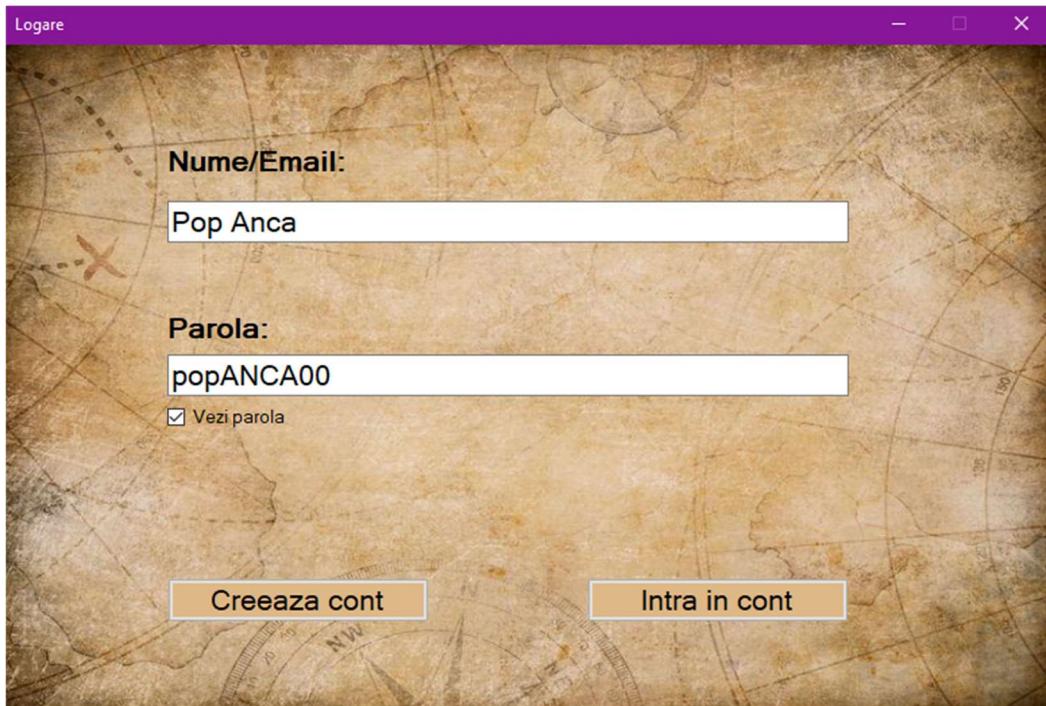
Fereastra conține:

- Două *TextBox*-uri: un *TextBox* pentru nume sau e-mail (txtNume) și un *TextBox* pentru parolă (txtParola).
- Un *CheckBox* (cbParola) care determină dacă parola introdusă de utilizator va fi vizibilă ca și caractere sau se va afișa ‘*’ pentru fiecare caracter.
- Două *Button*-uri: un *Button* pentru logarea în cont (btnLogare) și un *Button* pentru crearea unui nou cont (btnCreeazaCont).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition





Se verifică dacă datele de logare sunt corecte, iar dacă acestea corespund unui cont se încarcă fereastra FrmMain și se închide fereastra FrmLogare, utilizând următoarele secvențe de cod:

```
//se verifică dacă datele de logare sunt corecte
```

```
if(txtNume.Text!="" && txtParola.Text!="")
{
    //Se creeaza conexiunea la baza de date
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea
    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM conturi";
    int ok = 0;
    //Se executa comanda si se depune in r rezultatul
    MySqlDataReader r = cmd.ExecuteReader();
```

```

while (r.Read())
{
    Conturi x = new Conturi();
    {
        x.id = r.GetString("id");
        x.utilizator = r.GetString("utilizator");
        x.parola = r.GetString("parola");
        x.drepturi = r.GetString("drepturi");
    };

    if (x.utilizator == txtNume.Text && x.parola == txtParola.Text)
    {
        FrmMain main = new FrmMain();
        main.Tag = this;
        main.Text = x.id;
        main.Show();
        this.Hide();
        ok = 1;
    }
    else
    {
        if (x.utilizator == txtNume.Text && x.parola != txtParola.Text)
        {
            MessageBox.Show("Parola introdusa este incorecta.");
            ok = 1;
        }
    }
}

if(ok==0)
{
    MessageBox.Show("Numele de utilizator este gresit sau contul nu exista.");
}

r.Close();
conexiune.Close();
}
else
{
    MessageBox.Show("Toate campurile trebuie completeate inainte de a continua.");
}

```

//se închide fereastra FrmLogare

```
1 reference
private void FrmLogare_FormClosed(object sender, FormClosedEventArgs e)
{
    if((this.Tag as FrmContNou)! = null)
    {
        (this.Tag as FrmContNou).Close();
    }

    if ((this.Tag as FrmMain) != null)
    {
        (this.Tag as FrmMain).Close();
    }
}
```

3.2. Fereastra FrmContNou

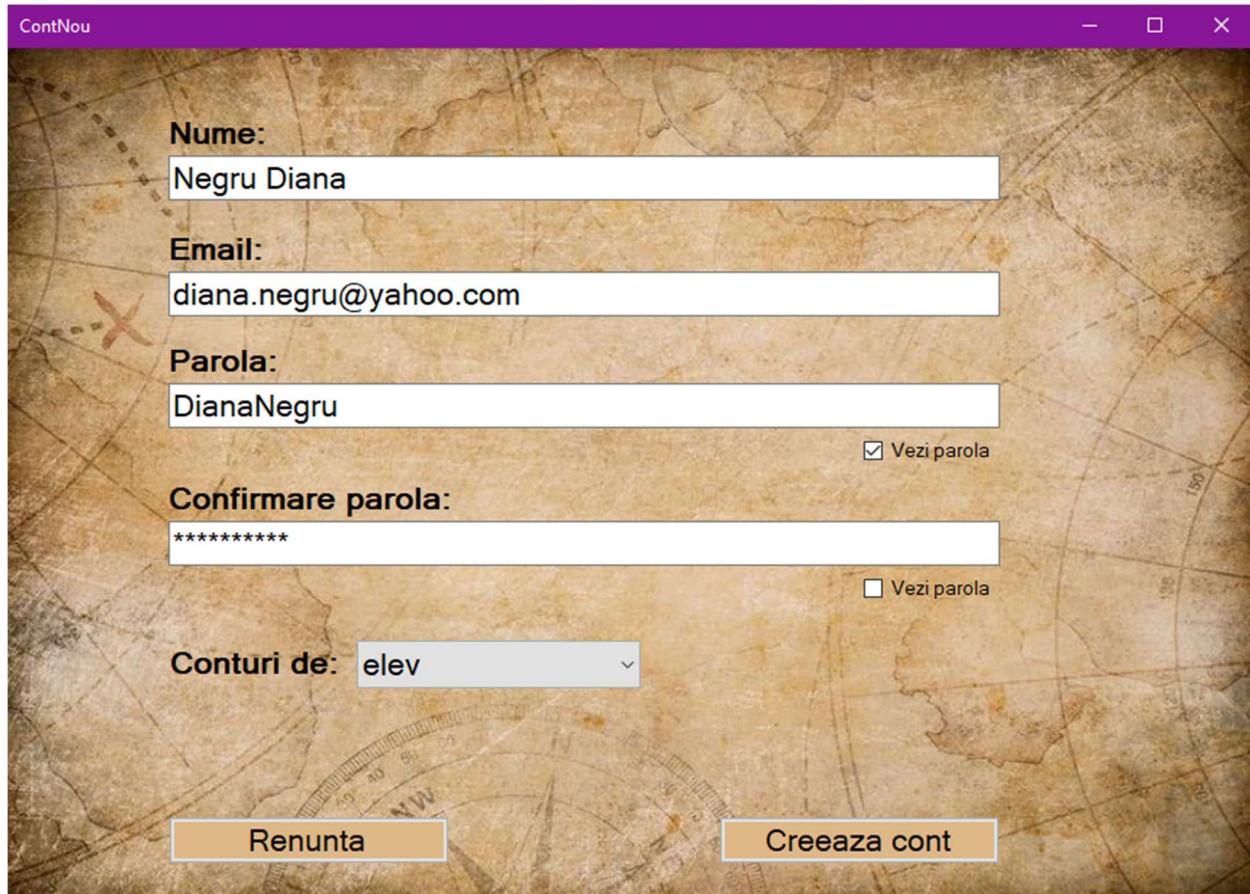
Prin intermediul acestei ferestre se poate crea un cont nou de utilizator (elev/profesor).

Fereastra conține:

- Trei *TextBox*-uri: un *TextBox* pentru nume (txtNume), un *TextBox* pentru parolă (txtParola) și un *TextBox* pentru confirmarea parolei (txtConfParola).
- Două *CheckBox*-uri (cbParola și cbConfParola) care determină dacă parola, respectiv confirmarea parolei vor fi vizibile ca și caractere sau se va afișa ‘*’ pentru fiecare caracter.
- Un *ComboBox* (cbCont) din care se va selecta tipul de cont (elev/profesor) care va fi creat.
- Două *Button*-uri: un *Button* pentru crearea contului (btnCreeazaCont) și un *Button* pentru a te întoare la pagina de logare (btnRenunta).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition



Pentru crearea contului și întoarcere la pagina de logare se utilizează următoarele secvențe de cod:

```

if(txtNume.Text!="" && txtParola.Text==txtConfParola.Text && txtParola.Text!="" &&
cbCont.SelectedIndex!= -1 && txtParola.TextLength>=8)
{
    //Se creeaza conexiunea la baza de date
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = @"INSERT INTO conturi(utilizator, parola, drepturi)
    VALUES(@utilizator, @parola, @drepturi);";
    //Se stabilesc parametrii comenzii in ordine
    cmd.Parameters.AddWithValue("utilizator", txtNume.Text);
}

```

```

cmd.Parameters.AddWithValue("pparola", txtParola.Text);
cmd.Parameters.AddWithValue("pdrepturi", cbCont.Text);

//Se executa comanda si se verifica daca rezultatul este nenul
if(cmd.ExecuteNonQuery() != 0)
{
    MessageBox.Show("Contul a fost creat cu succes!");
    FrmLogare log = new FrmLogare();
    log.Show();
    log.Tag = this;
    this.Hide();
}
else
{
    MessageBox.Show("Din cauza unor probleme, contul nu a putut fi creat. Va
rugam, sa incercati din nou.");
}
conexiune.Close();
}
else
{
    if (txtNume.Text == "" || txtConfParola.Text=="" || txtParola.Text == "" ||
cbCont.SelectedIndex == -1)
    {
        MessageBox.Show("Trebuie completate toate casutele inainte de a va putea crea
un cont!");
    }
    else
    {
        if(txtParola.TextLength<8)
        {
            MessageBox.Show("Parola trebuie sa aiba minim 8 caractere.");
        }
        else
        {
            if (txtParola.Text != txtConfParola.Text)
            {
                MessageBox.Show("Parolele nu coincid.");
            }
        }
    }
}
}

```

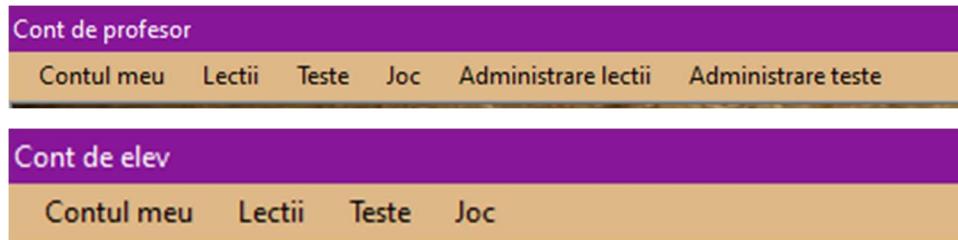
```
//se închide fereastra FrmContNou  
1 reference  
private void FrmContNou_FormClosed(object sender, FormClosedEventArgs e)  
{  
    if((this.Tag as FrmLogare)!=null)  
    {  
        (this.Tag as FrmLogare).Close();  
    }  
}
```

3.3. Fereastra FrmMain

Aceasta este fereastra principală a aplicației. Este de tipul *MdiContainer* și conține un obiect de tip *MenuStrip* prin intermediul căruia sunt deschise celelalte ferestre.

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



Încărcarea lecțiilor și a testelor din cod, preluate din baza de date și încărcarea ferestrei corespunzătoare:

→ *LECTII*

```
void menu()
{
    lectiiToolStripMenuItem.DropDownItems.Clear();

    //Se creeaza conexiunea la baza de date
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlConnection con = new MySqlConnection();
    con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    con.Open(); //Se deschide conexiunea
```

```

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = con;
cmd.CommandText = "SELECT * FROM capitole";

MySqlCommand command = new MySqlCommand();
command.Connection = conexiune;
command.CommandText = "SELECT lectii.* FROM capitole JOIN lectii ON
capitole.idC=lectii.idC ORDER BY capitole.idC";

//Se executa comanda si se depune in r rezultatul
MySqlDataReader r = cmd.ExecuteReader();
while (r.Read())
{
    Capitole c = new Capitole();
    {
        c.idC = r["idC"].ToString();
        c.titlu = r["titlu"].ToString();
    }

    lectiiToolStripMenuItem.DropDownItems.Add(c.ToString()); // se adauga capitolul
in meniu

//se creeaza o legatura intre denumirea capitolului si submeniu
ToolStripMenuItem it =
lectiiToolStripMenuItem.DropDownItems[lectiiToolStripMenuItem.DropDownItems.Count - 1] as ToolStripMenuItem;

MySqlDataReader reader = command.ExecuteReader();
while (reader.Read())
{
    Lectii l = new Lectii();
    {
        l.idL = reader["idL"].ToString();
        l.idC = reader["idC"].ToString();
        l.titlu = reader["titlu"].ToString();
        l.idProfesor = reader["idProfesor"].ToString();
        l.data_adaugare = Convert.ToDateTime(reader["data_adaugare"]);
    }

    if(c.idC==l.idC)
    {
        it.DropDownItems.Add(l.ToString());
        ToolStripMenuItem subit = it.DropDownItems[it.DropDownItems.Count - 1]
as ToolStripMenuItem;
    }
}

```

```

        subit.Click += new System.EventHandler(this.subit_Click); //eventul Click
        // se transmite prin Tag lectia care a fost aleasa la eventul Click
        subit.Tag = l.idL;
    }
}
reader.Close();
}
r.Close();
conexiune.Close();
con.Close();
}
}

private void subit_Click(object sender, EventArgs e)
{
    //id-ul testului care urmeaza sa fie incarcat
    int id = Convert.ToInt32((sender as ToolStripDropDownItem).Tag.ToString());

    FrmLectii frml = new FrmLectii();
    frml.MdiParent = this;
    frml.Tag = id;
    frml.Show();
    frml.Location = new Point(200, 0);
    frml.Text = sender.ToString();
}
}

```

→ TESTE

```

void meniuTeste()
{
    testeToolStripMenuItem.DropDownItems.Clear();

    //Se creeaza conexiunea la baza de date
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT DISTINCT capitol FROM teste"; //adaug capitolele
    pentru care exista teste
}

```

```

MySqlDataReader r = cmd.ExecuteReader();
while(r.Read())
{
    testeToolStripMenuItem.DropDownItems.Add(r["capitol"].ToString());

    MySqlConnection con = new MySqlConnection();
    con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    con.Open(); //Se deschide conexiunea

    MySqlCommand cm = new MySqlCommand();
    cm.Connection = con;
    cm.CommandText = "SELECT DISTINCT lectie FROM teste WHERE
    capitol=@pcapitol"; //adaug lectiile pentru care exista teste
    cm.Parameters.AddWithValue("pcapitol", r["capitol"].ToString());

    ToolStripMenuItem it =
    testeToolStripMenuItem.DropDownItems[testeToolStripMenuItem.DropDownItems.Count - 1] as ToolStripMenuItem;

    MySqlDataReader reader = cm.ExecuteReader();
    while(reader.Read())
    {
        it.DropDownItems.Add(reader["lectie"].ToString());

        MySqlConnection conex = new MySqlConnection();
        conex.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
        UID=root; PASSWORD=; Allow User Variables=True";
        conex.Open(); //Se deschide conexiunea

        MySqlCommand cmm = new MySqlCommand();
        cmm.Connection = conex;
        cmm.CommandText = "SELECT * FROM teste WHERE capitol=@pC AND
        lectie=@pLectie"; //adaug lectiile pentru care exista teste
        cmm.Parameters.AddWithValue("pC", r["capitol"].ToString());
        cmm.Parameters.AddWithValue("pLectie", reader["lectie"].ToString());

        ToolStripMenuItem t = it.DropDownItems[it.DropDownItems.Count - 1] as
        ToolStripMenuItem;

        MySqlDataReader rr = cmm.ExecuteReader();
        while(rr.Read())
        {
            Teste test = new Teste();
            {
                test.idTest = rr["idTest"].ToString();

```

```
        test.denumire = rr["denumire"].ToString();
    }

    t.DropDownItems.Add(test.ToString());

    ToolStripMenuItem subt = t.DropDownItems[t.DropDownItems.Count - 1] as
ToolStripMenuItem;

    subt.Click += new System.EventHandler(this.subt_Click); //eventul Click
    subt.Tag = test.idTest;

}

rr.Close();
conex.Close();
}
reader.Close();
con.Close();
}
}

private void subt_Click(object sender, EventArgs e)
{
    //id-ul testului care urmeaza sa fie incarcat
    int id = Convert.ToInt32 ((sender as ToolStripDropDownItem).Tag.ToString());

    FrmTeste frmt = new FrmTeste();
    frmt.MdiParent = this;
    frmt.Tag = id;
    frmt.Show();
    frmt.Location = new Point(300, 100);
    frmt.Text = sender.ToString();
    frmt.Size = new Size(921, 595);
}
```

Pentru încărcarea celorlalte ferestre se utilizează următoarele sevențe de cod:

```
//FrmSchimbaParola

1 reference
private void schimbaParolaToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmSchhimbaParola fsp = new FrmSchhimbaParola();
    fsp.MdiParent = this;
    fsp.Tag = id.ToString();
    fsp.Show();
    fsp.Location = new Point(400, 150);
}

//FrmNotificari

private void notificariToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmNotificari fn = new FrmNotificari();
    fn.MdiParent = this;
    fn.Tag = id.ToString();
    fn.Show();
}

//FrmAdaugaLectie

private void adaugaLectieToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmAdaugaLectie adlectie = new FrmAdaugaLectie();
    adlectie.MdiParent = this;
    adlectie.Tag = id.ToString();
    adlectie.Show();
    adlectie.Location = new Point(150, 15);
}

//FrmModificaLectie

private void modificaLectieToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmModificaLectie modifl = new FrmModificaLectie();
    modifl.MdiParent = this;
    modifl.Show();
    modifl.Location = new Point(200, 15);
}

//FrmModifOrdCapitole
```

```

private void lectiiToolStripMenuItem1_Click(object sender, EventArgs e)
{
    FrmModifOrdCapitole fmoc = new FrmModifOrdCapitole();
    fmoc.MdiParent = this;
    fmoc.Show();
    fmoc.Location = new Point(400, 100);
}

//FrmModifOrdLectii

private void lectiiToolStripMenuItem2_Click(object sender, EventArgs e)
{
    FrmModifOrdLectii fmol = new FrmModifOrdLectii();
    fmol.MdiParent = this;
    fmol.Show();
    fmol.Location = new Point(400, 100);
}

//FrmSergeLectie

private void stergeLectieToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmSergeLectie fsl = new FrmSergeLectie();
    fsl.MdiParent = this;
    fsl.Tag = id.ToString();
    fsl.Show();
    fsl.Location = new Point(200, 0);
}

//FrmAdaugaIntrebari

private void adaugaIntrebariToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmAdaugaIntrebari fai = new FrmAdaugaIntrebari();
    fai.MdiParent = this;
    fai.Tag = id.ToString();
    fai.Show();
    fai.Location = new Point(200, 0);
}

//FrmModificaIntrebari

```

```

private void ...modificaIntrebariToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmModificaIntrebari fmi = new FrmModificaIntrebari();
    fmi.MdiParent = this;
    fmi.Tag = id.ToString();
    fmi.Show();
    fmi.Location = new Point(250, 0);
}

//FrmSergeIntrebari

private void ...stergeIntrebariToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmSergeIntrebari fsi = new FrmSergeIntrebari();
    fsi.MdiParent = this;
    fsi.Tag = id.ToString();
    fsi.Show();
    fsi.Location = new Point(250, 0);
}

//FrmAdaugaTest

private void ...adaugaTestToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmAdaugaTest adtest = new FrmAdaugaTest();
    adtest.MdiParent = this;
    adtest.Tag = id.ToString();
    adtest.Show();
    adtest.Location = new Point(250, 0);
}

//FrmModificaTest

private void ...modificaTestToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmModificaTest fmt = new FrmModificaTest();
    fmt.MdiParent = this;
    fmt.Tag = id.ToString();
    fmt.Show();
    fmt.Location = new Point(250, 0);
}

//FrmSergeTest

```

```
private void stergeTestToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmSтерgeTest fst = new FrmSтерgeTest();
    fst.MdiParent = this;
    fst.Tag = id.ToString();
    fst.Show();
    fst.Location = new Point(250, 0);
}

//FrmJoc

private void jocuriToolStripMenuItem_Click(object sender, EventArgs e)
{
    FrmJoc fj = new FrmJoc();
    fj.MdiParent = this;
    fj.Show();
}
```

3.4. Fereastra FrmLectii

Prin intermediul acestei ferestre este afișată lecția pe care utilizatorul a selectat-o din meniu prezent în FrmMain.

Fereastra conține:

- Două *Label*-uri: un *Label* în care se afișează titlul capitolului și al lecției (lblDenumire) și un *Label* în care se afișează descrierea imaginii (lblDescriere).
- Un *RichTextBox* în care se afișează conținutul lecției (rtbLectie).
- Un *PictureBox* în care se afișează imaginile care aparțin lecției (pbImgLectie).
- Două *Button*-uri: unul care schimbă imaginea cu imaginea dinaintea celei afișate (btnStanga) și unul care schimbă imaginea cu imaginea care urmează celei afișate (btnDreapta).
- Un *ListBox* care nu este vizibil în care se încarcă calea tuturor imaginilor din lecție, în ordinea în care apar în baza de date (lstImg).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState

Istoria sclaviei

Cont de profesor - [Mișcarea pentru drepturile civile]

Contul meu Lectii Teste Joc Administrare lectii Administrare teste

Emanciparea. Mișcarea pentru drepturile civile

Mișcarea pentru drepturile civile a fost o luptă pentru justiție și egalitate pentru afro-americanii, care a avut loc în principal în anii 1950 și 1960. Printre liderii săi s-au numărat Martin Luther King Jr., Malcolm X, Little Rock Nine, Rosa Parks și mulți alții. Războiul civil abolise oficial sclavia, dar nu a pus capăt discriminării împotriva oamenilor de culoare - au continuat să suporte efectele devastatoare ale rasismului, în special în sud.

La mijlocul secolului al XX-lea, americanii negri avuveseră mai mult decât suficient prejudecăți și violență împotriva lor. Ei, alături de mulți americanii albi, s-au mobilizat și au început o luptă fără precedent pentru egalitate care a durat două decenii.

Pe parcursul Reconstrucției Negrii au preluat roluri de conducere ca niciodată. Aceștia au deținut funcții publice și au căutat schimbări legislative pentru egalitate și dreptul la vot, entrând într-o marginalizare a oamenilor negri, și menținând separația de cei albi și a stergeră progresul pe care îl făcuseră în timpul Reconstrucției, legile „Jim Crow” au fost stabilite în sud începând cu sfârșitul secolului al XIX-lea. Oamenii negri nu puteau folosi aceleși facilități publice ca oamenii albi, să locuască în multe din același oraș sau să meargă la aceleași școli. Căsătoria interrasială a fost ilegală, iar majoritatea oamenilor de culoare nu au putut vota deoarece nu au reușit să treacă testele de alfabetizare a alegătorilor. Înainte de cel de-al doilea război mondial, majoritatea oamenilor negri lucrau ca fermieri cu salarii reduse, muncitori din fabrici, domestici sau servitori.

La începutul anilor 1940, munca legată de război era în plină expansiune, dar majoritatea americanilor negri nu primeau locuri de muncă mai bine plătite. De asemenea, au fost descurajăți să se alăture armatei.

Bărbații și femeile negre au slujit eroic în cel de-al doilea război mondial, în ciuda suferinței de separare și discriminare în timpul desfășurării lor. Tuskegee Airmen a

Rosa Parks



Încărcarea lecției, a imaginilor din baza de date în *ListBox* și afișarea imaginilor se face utilizând următoarele secvențe de cod:

```
//Se creeaza conexiunea la baza de date
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT capitole.titlu, lectii.titlu as lTitlu, continut_lectii.continut
FROM capitole JOIN lectii ON capitole.idC=lectii.idC JOIN continut_lectii ON
lectii.idL=continut_lectii.idL WHERE lectii.idL=@pidL;";

cmd.Parameters.AddWithValue("pidL", this.Tag);

//Se executa comanda si se depune in r rezultatul
MySqlDataReader r = cmd.ExecuteReader();
while(r.Read())
{
    lblDenumire.Text = r["titlu"].ToString() + ". " + r["lTitlu"].ToString();
    rtbLectie.Text = r["continut"].ToString();
```

```

}

r.Close();

cmd.CommandText = "SELECT * FROM imagini WHERE idL=@ppidL;";
cmd.Parameters.AddWithValue("ppidL", this.Tag);
r = cmd.ExecuteReader();

while (r.Read())
{
    Imagini img = new Imagini();
    {
        img.idImg = r["idImg"].ToString();
        img.idL = r["idL"].ToString();
        img.denumire = r["denumire"].ToString();
        img.cale = r["cale"].ToString();
    }
    lstImg.Items.Add(img);
}
r.Close();
conexiune.Close();

if(lstImg.Items.Count>0)
{
    Imagini img = lstImg.Items[0] as Imagini;
    pbImgLectie.Image = new Bitmap(img.cale.ToString());
    Imagini imag = lstImg.Items[0] as Imagini;
    lblDescriere.Text = imag.denumire;
    lblDescriere.Visible = true;

    if (lstImg.Items.Count > 1)
    {
        btnDreapta.Visible = true;
    }
}
else
{
    pbImgLectie.Image = new Bitmap(@"Img/nopic.png");
    lblDescriere.Visible = false;
}

```

```
private void btnStanga_Click(object sender, EventArgs e)
{
    poz--;
    if(poz==0)
    {
        btnStanga.Visible = false;
    }
    btnDreapta.Visible = true;

    pbImgLectie.Image = new Bitmap(lstImg.Items[poz].ToString());
    Imagini imag = lstImg.Items[poz] as Imagini;
    lblDescriere.Text = imag.denumire;
}

int poz = 0;
1 reference
private void btnDreapta_Click(object sender, EventArgs e)
{
    poz++;
    btnStanga.Visible = true;

    pbImgLectie.Image = new Bitmap(lstImg.Items[poz].ToString());
    Imagini imag = lstImg.Items[poz] as Imagini;
    lblDescriere.Text = imag.denumire;

    if(poz==lstImg.Items.Count-1)
    {
        btnDreapta.Visible = false;
    }
}
```

3.5. Fereastra FrmTeste

Prin intermediul acestei ferestre este afișat testul pe care utilizatorul l-a selectat din meniuul prezent în FrmMain.

Fereastra conține:

- Două *ListBox*-uri: un *ListBox* care nu este vizibil și în care sunt încărcate întrebările care aparțin testului selectat (lstIntrebari), iar celălalt *ListBox*, care inițial nu este vizibil, în care se încarcă întrebările la care utilizatorul a răspuns gresit (lstIntrebariGresite), acesta devenind vizibil după apăsarea butonului de verificare.
- Trei *Button*-uri: un *Button* pentru a trece la întrebarea următoare (btnIntrUrm), un *Button* pentru a trece la întrebarea anterioară (btnIntrAnt) și un *Button* pentru corectarea testului (btnVerifica).
- Patru *Panel*-uri:

→ pnlRaspTextScurt în care se încarcă întrebările care au răspunsul de tip text scurt. Acesta conține două *Label*-uri, unul în care apare textul “Intrebare:” și unul în care apare textul “Raspuns:”. Pe lângă acestea, *Panel*-ul mai conține un *RichTextBox* în care se încarcă întrebarea (rtbIntrebareTS) și un *TextBox* în care utilizatorul va scrie răspunsul corect (txtRaspuns).

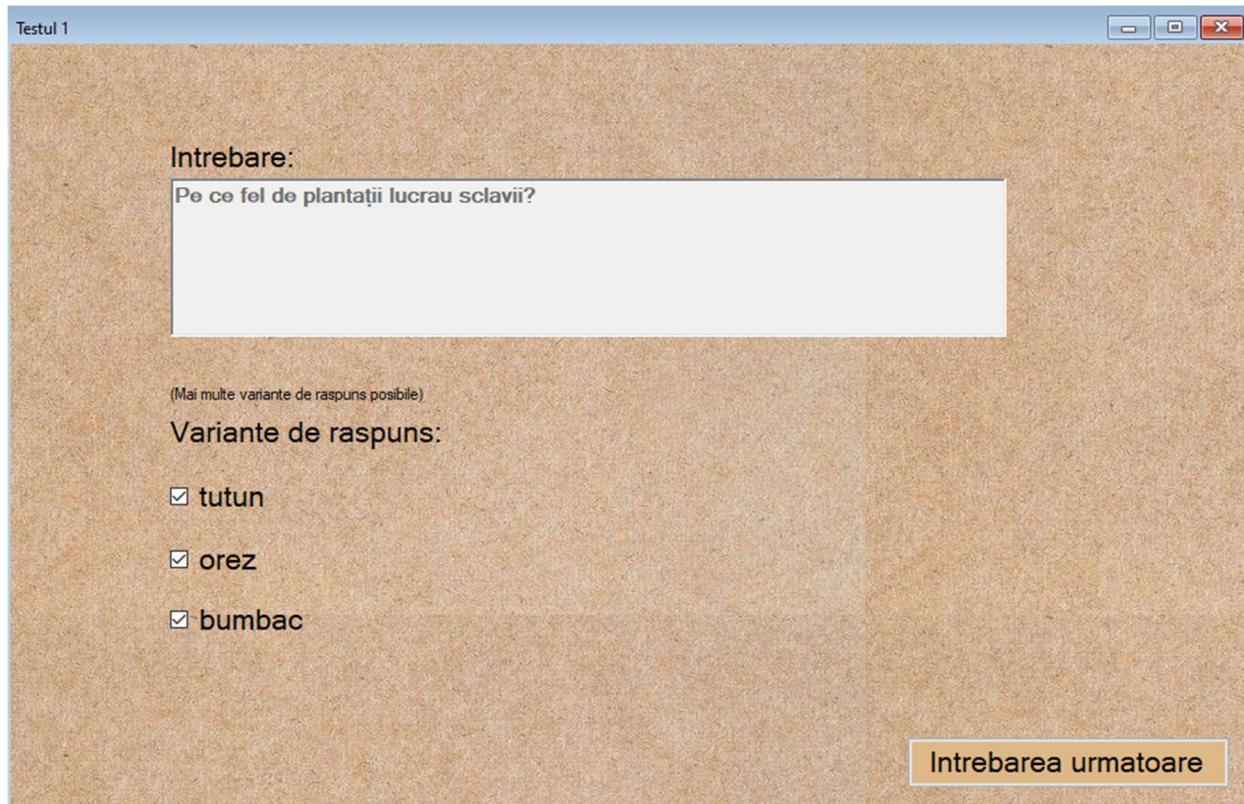
→ pnlRaspUnic în care se încarcă întrebările care au răspunsul de tip răspuns unic. Conține trei *Label*-uri în care apar texte: “Intrebare:”, “(O singura varianta de raspuns corecta)” și “Variante de raspuns:”. Acesta mai conține un *RichTextBox* în care se încarcă întrebarea (rtbIntrebareRU) și trei *RadioButton*-uri în care se încarcă variantele de răspuns (rbVar1, rbVar2, rbVar3).

→ pnlRaspunsMultiplu în care se încarcă întrebările care au răspunsul de tip răspuns multiplu. Conține trei *Label*-uri în care apar texte: “Intrebare”, “(Mai multe variante de raspuns posibile)” și “Variante de raspuns:”. Acesta mai conține un *RichTextBox* în care se încarcă întrebarea (rtbIntrebareRM) și trei *CheckBox*-uri în care se încarcă variantele de răspuns (cbVar1, cbVar2, cbVar3).

→ pnlRaspCorecte în care se încarcă răspunsul corect/răspunsurile corecte pentru întrebarea selectată din *ListBox*-ul lstIntrebariGresite. Acesta conține patru *Label*-uri, unul în care apare textul “Raspuns corect:”, iar în celelalte trei se încarcă răspunsurile corect, în funcție de numărul lor (lblRC1, lblRC2, lblRC3).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



Istoria sclaviei

Testul 1

Intrebare:
Scieți termenul la care se referă definiția: "desemnează condiția umană a persoanelor care lucrează pentru un stăpân fără remunerație și ce nu dispun de drepturi asupra propriei persoane".

Raspuns:

Intrebarea anterioara **Verifica**

Testul 1

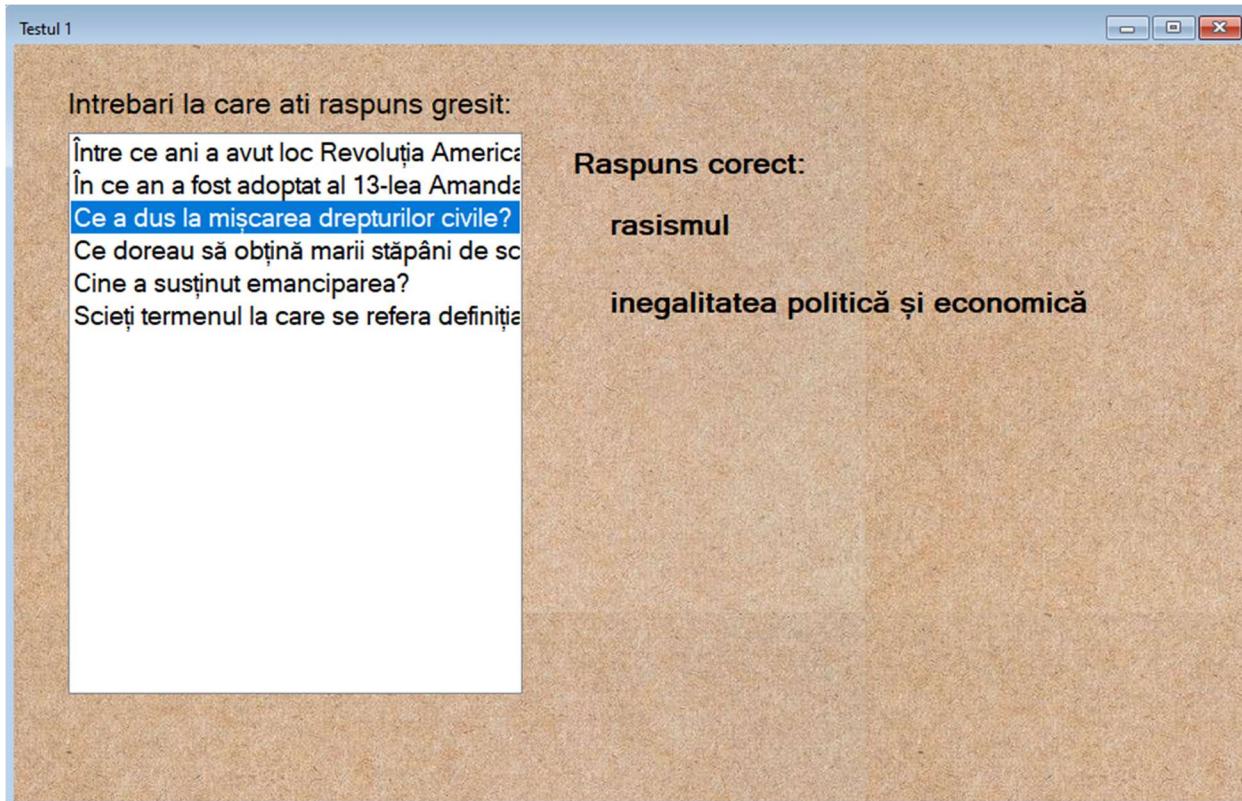
Intrebare:
Scieți termenul la care se referă definiția: "desemnează condiția umană a persoanelor care lucrează pentru un stăpân fără remunerație și ce nu dispun de drepturi asupra propriei persoane".

Raspuns:

Punctaj: 4/10

OK

Intrebarea anterioara **Verifica**



Pentru încărcarea întrebărilor în *ListBox*-uri se folosește următoarea secvență de cod:

```
void Intr()
{
    lstIntrebari.Items.Clear();

    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM continut_teste JOIN intrebare ON
    continut_teste.idIntrebare=intrebare.idIntrebare WHERE continut_teste.idTest=@pid";
    cmd.Parameters.AddWithValue("pid", this.Tag.ToString());

    MySqlDataReader r = cmd.ExecuteReader();
    while(r.Read())
    {
```

```

Intrebari intr = new Intrebari();
{
    intr.idIntrebare = r["idIntrebare"].ToString();
    intr.intrebare = r["intrebare"].ToString();
}

lstIntrebari.Items.Add(intr);
lstIntrebariGresite.Items.Add(intr);
}
r.Close();
conexiune.Close();
}

```

Pentru încărcarea întrebărilor în *Panel*-uri se folosește următoarea secvență de cod:

```

int intr = 0;
void incarcaIntr()
{
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM raspunsuri WHERE idIntrebare=@pidIntr";

    Intrebari intrebare = lstIntrebari.Items[intr] as Intrebari;
    cmd.Parameters.AddWithValue("pidIntr", intrebare.idIntrebare);

    MySqlDataReader r = cmd.ExecuteReader();
    int poz = 1;
    while(r.Read())
    {
        Raspunsuri raspuns = new Raspunsuri();
        {
            raspuns.raspuns = r["raspuns"].ToString();
            raspuns.tipRaspuns = r["tipRaspuns"].ToString();
        }

        if (raspuns.tipRaspuns == "text scurt")

```

```

{
    pnlRaspTextScurt.Visible = true;
    pnlRaspMultiplu.Visible = pnlRaspUnic.Visible = false;
    rtbIntrebareTS.Text = intrebare.intrebare;
    txtRaspuns.Text = "";
    break;
}
else
{
    if (raspuns.tipRaspuns == "unic")
    {
        pnlRaspUnic.Visible = true;
        pnlRaspTextScurt.Visible = pnlRaspMultiplu.Visible = false;
        rbVar1.Checked = false;
        rbVar2.Checked = false;
        rbVar3.Checked = false;

        if (poz == 1)
        {
            rtbIntrebareRU.Text = intrebare.intrebare;
            rbVar1.Text = r["raspuns"].ToString();
            poz++;
        }
        else
        {
            if (poz == 2)
            {
                rbVar2.Text = r["raspuns"].ToString();
                poz++;
            }
            else
            {
                rbVar3.Text = r["raspuns"].ToString();
                break;
            }
        }
    }
    else
    {
        pnlRaspMultiplu.Visible = true;
        pnlRaspTextScurt.Visible = pnlRaspUnic.Visible = false;
        cbVar1.Checked = false;
        cbVar2.Checked = false;
        cbVar3.Checked = false;

        if (poz == 1)
    }
}

```

```

        {
            rtbIntrebareRM.Text = intrebare.intrebare;
            cbVar1.Text = r["raspuns"].ToString();
            poz++;
        }
        else
        {
            if (poz == 2)
            {
                cbVar2.Text = r["raspuns"].ToString();
                poz++;
            }
            else
            {
                cbVar3.Text = r["raspuns"].ToString();
                break;
            }
        }
    }
}
conexiune.Close();

if(intr<lstIntrebari.Items.Count-1)
{
    btnIntrUrm.Visible = true;
    btnVerifica.Visible = false;
}
else
{
    btnIntrUrm.Visible = false;
    btnVerifica.Visible = true;
}

if(intr>0)
{
    btnIntrAnt.Visible = true;
}
else
{
    btnIntrAnt.Visible = false;
}
}

```

Pentru schimbarea întrebărilor și salvarea și încărcarea răspunsului dat de utilizator se utilizează următoarele secvențe de cod:

```

string[] rasp1 = new string[100];
string[] rasp2 = new string[100];
string[] rasp3 = new string[100];
string[] idIntr = new string[100];
string[] tipIntr = new string[100];

void salveazaRasp()
{
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM raspunsuri WHERE idIntrebare=@pidIntr";

    Intrebari intrebare = lstIntrebari.Items[intr] as Intrebari;
    cmd.Parameters.AddWithValue("pidIntr", intrebare.idIntrebare);

    MySqlDataReader r = cmd.ExecuteReader();
    int cate = 0;
    while (r.Read())
    {
        Raspunsuri raspuns = new Raspunsuri();
        {
            raspuns.raspuns = r["raspuns"].ToString();
            raspuns.tipRaspuns = r["tipRaspuns"].ToString();
        }

        if (cate == intr)
        {

            if (raspuns.tipRaspuns == "text scurt")
            {
                rasp1[intr] = txtRaspuns.Text;
                rasp2[intr] = "nu";
                rasp3[intr] = "nu";

                idIntr[intr] = r["idIntrebare"].ToString();
            }
        }
    }
}

```

```

        tipIntr[intr] = "text scurt";
    }
else
{
    if (raspuns.tipRaspuns == "unic")
    {
        if(rbVar1.Checked==true)
        {
            rasp1[intr] = "da";
        }
        else
        {
            rasp1[intr] = "nu";
        }

        if (rbVar2.Checked == true)
        {
            rasp2[intr] = "da";
        }
        else
        {
            rasp2[intr] = "nu";
        }

        if (rbVar3.Checked == true)
        {
            rasp3[intr] = "da";
        }
        else
        {
            rasp3[intr] = "nu";
        }

        idIntr[intr] = r["idIntrebare"].ToString();

        tipIntr[intr] = "unic";
    }

    if(raspuns.tipRaspuns == "multiplu")
    {
        if(cbVar1.Checked == true)
        {
            rasp1[intr] = "da";
        }
        else
        {
    
```

```

        rasp1[intr] = "nu";
    }

    if (cbVar2.Checked == true)
    {
        rasp2[intr] = "da";
    }
    else
    {
        rasp2[intr] = "nu";
    }

    if (cbVar3.Checked == true)
    {
        rasp3[intr] = "da";
    }
    else
    {
        rasp3[intr] = "nu";
    }

    idIntr[intr] = r["idIntrebare"].ToString();

    tipIntr[intr] = "multiplu";
}

}
else
{
    cate++;
    if (cate == intr)
    {

        if (raspuns.tipRaspuns == "text scurt")
        {
            rasp1[intr] = txtRaspuns.Text;
            rasp2[intr] = "nu";
            rasp3[intr] = "nu";

            idIntr[intr] = r["idIntrebare"].ToString();

            tipIntr[intr] = "text scurt";
        }
        else
        {
            if (raspuns.tipRaspuns == "unic")

```

```

{
    if (rbVar1.Checked == true)
    {
        rasp1[intr] = "da";
    }
    else
    {
        rasp1[intr] = "nu";
    }

    if (rbVar2.Checked == true)
    {
        rasp2[intr] = "da";
    }
    else
    {
        rasp2[intr] = "nu";
    }

    if (rbVar3.Checked == true)
    {
        rasp3[intr] = "da";
    }
    else
    {
        rasp3[intr] = "nu";
    }

    idIntr[intr] = r["idIntrebare"].ToString();

    tipIntr[intr] = "unic";
}

if (raspuns.tipRaspuns == "multiplu")
{
    if (cbVar1.Checked == true)
    {
        rasp1[intr] = "da";
    }
    else
    {
        rasp1[intr] = "nu";
    }

    if (cbVar2.Checked == true)
    {

```

```

        rasp2[intr] = "da";
    }
    else
    {
        rasp2[intr] = "nu";
    }

    if (cbVar3.Checked == true)
    {
        rasp3[intr] = "da";
    }
    else
    {
        rasp3[intr] = "nu";
    }

    idIntr[intr] = r["idIntrebare"].ToString();

    tipIntr[intr] = "multiplu";
}

}

else
{
    cate++;
}

}

r.Close();
conexiune.Close();
}

```

```
void incarcaRaspSalvat()
{
    //incarca rasp
    if (rasp1[intr]!=null && rasp2[intr]!=null && rasp3[intr]!=null && idIntr[intr]!=null)
    {
        if (tipIntr[intr]=="text scurt")
        {
            txtRaspuns.Text = rasp1[intr].ToString();
        }
        else
        {
            if (tipIntr[intr]=="unic")
            {

```

```

if (rasp1[intr]=="da")
{
    rbVar1.Checked = true;
}
else
{
    rbVar1.Checked = false;
}

if (rasp2[intr]=="da")
{
    rbVar2.Checked = true;
}
else
{
    rbVar2.Checked = false;
}

if (rasp3[intr]=="da")
{
    rbVar3.Checked = true;
}
else
{
    rbVar3.Checked = false;
}

if (tipIntr[intr]=="multiplu")
{
    if (rasp1[intr] == "da")
    {
        cbVar1.Checked = true;
    }
    else
    {
        cbVar1.Checked = false;
    }

    if (rasp2[intr] == "da")
    {
        cbVar2.Checked = true;
    }
    else
    {
        cbVar2.Checked = false;
    }
}

```

```

        }

        if (rasp3[intr] == "da")
        {
            cbVar3.Checked = true;
        }
        else
        {
            cbVar3.Checked = false;
        }
    }
}
else
{
    reset();
}
}

void reset()
{
    txtRaspuns.Text = "";
    rbVar1.Checked = rbVar2.Checked = rbVar3.Checked = false;
    cbVar1.Checked = cbVar2.Checked = cbVar3.Checked = false;
}

private void btnIntrUrm_Click(object sender, EventArgs e)
{
    if (txtRaspuns.Text != "" || (cbVar1.Checked == true || cbVar2.Checked == true ||
    cbVar3.Checked == true) || (rbVar1.Checked==true || rbVar2.Checked==true ||
    rbVar3.Checked==true))
    {
        salveazaRasp();
    }

    intr++;
    incarcaIntr();
    incarcaRaspSalvat();
}

private void btnIntrAnt_Click(object sender, EventArgs e)
{
    if (txtRaspuns.Text != "" || (cbVar1.Checked == true || cbVar2.Checked == true ||
    cbVar3.Checked == true) || (rbVar1.Checked == true || rbVar2.Checked == true ||
    rbVar3.Checked == true))
    {
}

```

```

        salveazaRasp();
    }

    intr--;
    incarcaIntr();
    incarcaRaspSalvat();
}

```

Pentru corectarea testului și afișarea întrebărilor la care s-a răspuns greșit se utilizează următoarele secvențe de cod:

```

void verificaTest()
{
    int punctaj = 0;
    for(int i=0;i<lstIntrebari.Items.Count;i++)
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT * FROM raspunsuri WHERE
idIntrebare=@pidIntr";
        Intrebari intrebare = lstIntrebari.Items[i] as Intrebari;
        cmd.Parameters.AddWithValue("pidIntr",intrebare.idIntrebare.ToString());
        cmd.Parameters.AddWithValue("pid", this.Tag.ToString());

        MySqlDataReader r = cmd.ExecuteReader();
        while(r.Read())
        {
            if (rasp1[i]!=null && rasp2[i]!=null && rasp2[3]!=null)
            {
                if (tipIntr[i]=="text scurt" && rasp1[i] == r["raspuns"].ToString())
                {
                    punctaj++;
                }
                else
                {
                    if (tipIntr[i]=="unic" || tipIntr[i]=="multiplu")
                    {
                        if (rasp1[i] == r["corect"].ToString())

```

```

    {
        r.Read();
        if (rasp2[i] == r["corect"].ToString())
        {
            r.Read();
            if (rasp3[i] == r["corect"].ToString())
            {
                punctaj++;
                lstIntrebariGresite.Items[i] = "DE STERS";
            }
            else
            {
                break;
            }
        }
        else
        {
            break;
        }
    }
    else
    {
        break;
    }
}
r.Close();
conexiune.Close();
}
MessageBox.Show("Punctaj: " + punctaj.ToString() + "/" + lstIntrebari.Items.Count);
}

```

```

private void btnVerifica_Click(object sender, EventArgs e)
{
    salveazaRasp();
    verificaTest();

    pnlRaspMultiplu.Visible = pnlRaspTextScurt.Visible = pnlRaspUnic.Visible =
    btnIntrAnt.Visible = btnIntrUrm.Visible = btnVerifica.Visible = false;

    for(int i=0;i<lstIntrebariGresite.Items.Count;i++)
    {

```

```

        if (lstIntrebariGresite.Items[i].ToString() == "DE STERS")
    {
        lstIntrebariGresite.Items.Remove(lstIntrebariGresite.Items[i]);
        i--;
    }
}

lstIntrebariGresite.Visible = true;
lblIntrGresite.Visible = true;
}

private void lstIntrebariGresite_SelectedIndexChanged(object sender, EventArgs e)
{
    if(lstIntrebariGresite.SelectedIndex!=-1)
    {
        pnlRaspCorecte.Visible = true;

        Intrebari intrebare = lstIntrebariGresite.SelectedItem as Intrebari;

        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT * FROM raspunsuri WHERE
idIntrebare=@pidIntr";
        cmd.Parameters.AddWithValue("pidIntr", intrebare.idIntrebare);

        MySqlDataReader r = cmd.ExecuteReader();
        while(r.Read())
        {
            Raspunsuri rasp = new Raspunsuri();
            {
                rasp.raspuns = r["raspuns"].ToString();
                rasp.tipRaspuns = r["tipRaspuns"].ToString();
                rasp.corect = r["corect"].ToString();
            }

            if(rasp.tipRaspuns=="text scurt")
            {
                lblRC1.Text = rasp.raspuns;
                lblRC1.Visible = true;
                lblRC2.Visible = lblRC3.Visible = false;
            }
        }
    }
}

```

```

        break;
    }
else
{
    if(rasp.tipRaspuns=="unic")
    {
        lblRC1.Visible = true;
        lblRC2.Visible = lblRC3.Visible = false;

        if(rasp.corect=="da")
        {
            lblRC1.Text = rasp.raspuns;
            break;
        }
        else
        {
            r.Read();
            rasp.raspuns = r["raspuns"].ToString();
            rasp.tipRaspuns = r["tipRaspuns"].ToString();
            rasp.corect = r["corect"].ToString();

            if(rasp.corect=="da")
            {
                lblRC1.Text = rasp.raspuns;
                break;
            }
            else
            {
                r.Read();
                rasp.raspuns = r["raspuns"].ToString();
                rasp.tipRaspuns = r["tipRaspuns"].ToString();
                rasp.corect = r["corect"].ToString();

                lblRC1.Text = rasp.raspuns;
                break;
            }
        }
    }
else
{
    if(rasp.corect=="da")
    {
        //rasp 1
        lblRC1.Visible = true;
        lblRC2.Visible = lblRC3.Visible = false;
    }
}
}

```

```

lblRC1.Text = rasp.raspuns;

//rasp 2 (corect?)
r.Read();
rasp.raspuns = r["raspuns"].ToString();
rasp.tipRaspuns = r["tipRaspuns"].ToString();
rasp.corect = r["corect"].ToString();

if(rasp.corect=="da")
{
    lblRC2.Visible = true;
    lblRC2.Text = rasp.raspuns;

    //rasp 3 (corect?)
    r.Read();
    rasp.raspuns = r["raspuns"].ToString();
    rasp.tipRaspuns = r["tipRaspuns"].ToString();
    rasp.corect = r["corect"].ToString();

    if(rasp.corect=="da")
    {
        lblRC3.Visible = true;
        lblRC3.Text = rasp.raspuns;

        break;
    }
}
else
{
    //rasp 3 (corect?)
    r.Read();
    rasp.raspuns = r["raspuns"].ToString();
    rasp.tipRaspuns = r["tipRaspuns"].ToString();
    rasp.corect = r["corect"].ToString();

    if (rasp.corect == "da")
    {
        lblRC2.Visible = true;
        lblRC2.Text = rasp.raspuns;

        break;
    }
}
}

```



```
        }
        r.Close();
        conexiune.Close();
    }
else
{
    MessageBox.Show("Selectati intrebarea la care ati raspuns gresit si doriti sa vedeti
raspunsul corect.");
}
}
```

3.6. Fereastra FrmAdaugaLectie

Prin intermediul acestei ferestre se pot adăuga lecții noi în baza de date de către profesori.

Fereastra conține:

- Trei *Panel*-uri:

→ pnlContinut în care se afișează un *ComboBox* pentru alegerea unui capitol (cbCapitole), un *Button* cu ajutorul căruia se adaugă un nou capitol în *ComboBox* (btnAdaugaCapitol) și un *Label* cu textul “Capitole:”.

→ pnlLectie în care se afișează un *ComboBox* pentru alegerea unei lecții care face parte din capitolul selectat (cbLectii), un *Button* cu ajutorul căruia se adaugă o nouă lecție (btnAdaugaLectie) și un *Label* cu textul “Lectii:”.

→ pnlContinutLectie care conține un *RichTextBox* pentru adăugarea conținutului lecției (rtbContinutLectie), un *PictureBox* pentru adăugarea imaginilor corespunzătoare lecției (pbImgLectie), un *TextBox* pentru descrierea imaginii (txtDescriereImg), două *Label*-uri, unul care este vizibil și conține un text cu indicații de utilizare și unul care nu este vizibil și conține calea imaginii care se afișează în *PictureBox* (lblCaleImg). Acest *Panel* mai conține patru *Button*-uri: btnDelImg pentru ștergerea imaginii din *PictureBox*, btnAddImg pentru adăugarea imaginii în *PictureBox*, btnSalveazaImg pentru salvarea imaginii în baza de date și btnSalveaza pentru salvarea conținutului lecției în baza de date.

- Un *OpenFileDialog* (ofd) pentru adăugarea unei imagini de pe calculator în *PictureBox*.

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState

Istoria sclaviei

Capitol: Începuturile +

Lectie: Lectie noua +

In America de Nord au ajuns peste 4 milioane de slavi negri, doar în secolul al XVIII lea. Sclavii provin din rândul triburilor africane de pe coastele continentului și mai rar din interiorul acestuia. Explicația este simplă, erau mai ușor de transportat până la vasele care-i așteptau în porturi. De obicei existau grupuri de africani sau arabi specializați în vânătoarea de sclavi. Oamenii erau pur și simplu răpiți din sămău familiilor, puși în lanțuri, obligați să mărșăluască și apoi vânduți ca animalele în târguri. Erau preferați copii și adulții tineri. Chiar și americanii au avut propriile lor "echipe" de vânători de sclavi. Odată imbarcați, sclavii erau lăsați de multe ori încale, fără aer, legați în lanțuri și cu apă și hrana insuficiente. Era un adevarat calvar, iar mulți dintre aceștia mureau pe drum, din cauza bolilor sau a condițiilor de transport. Ajuși pe continentul American, erau vânduți în târguri specializate marilor proprietari de plantăjii.

În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce marii proprietari dețineau și către 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubre și muncesc până la epuizare. Primeau doar hrana. Majoritatea sclavilor erau incurajați să se căsătorească între ei, fiindcă se credea că aşa devineau mai docili și, totodată, dădeau naștere altor sclavi. Viața acestora era însă amenințată în permanență de violență sau separare, după cum dorea stăpânul. Înclusiv în secolul al XIX lea, când țările civilizate ale Europei renunțaseră la sclavi, în Statele Unite, sclavia era în floare și, mai mult decât atât, justificată de plantăjii.

În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce marii proprietari dețineau și către 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubre și muncesc până la epuizare. Primeau doar hrana. Majoritatea sclavilor erau incurajați să se căsătorească între ei, fiindcă se credea că aşa devineau mai docili și, totodată, dădeau naștere altor sclavi. Viața acestora era însă amenințată în permanență de violență sau separare, după cum dorea stăpânul. Înclusiv în secolul al XIX lea, când țările civilizate ale Europei renunțaseră la sclavi, în Statele Unite, sclavia era în floare și, mai mult decât atât, justificată de plantăjii.

Salveaza

Imaginiile se vor putea adăuga după salvarea lectiei, cu o descriere sugestiva.

-

+

Lectia a fost salvata cu succes.

OK

NO IMAGE AVAILABLE

Descrere imagine

Salveaza imagine

Capitol: Începuturile +

Lectie: Lectie noua +

In America de Nord au ajuns peste 4 milioane de slavi negri, doar în secolul al XVIII lea. Sclavii provin din rândul triburilor africane de pe coastele continentului și mai rar din interiorul acestuia. Explicația este simplă, erau mai ușor de transportat până la vasele care-i așteptau în porturi. De obicei existau grupuri de africani sau arabi specializați în vânătoarea de sclavi. Oamenii erau pur și simplu răpiți din sămău familiilor, puși în lanțuri, obligați să mărșăluască și apoi vânduți ca animalele în târguri. Erau preferați copii și adulții tineri. Chiar și americanii au avut propriile lor "echipe" de vânători de sclavi. Odată imbarcați, sclavii erau lăsați de multe ori încale, fără aer, legați în lanțuri și cu apă și hrana insuficiente. Era un adevarat calvar, iar mulți dintre aceștia mureau pe drum, din cauza bolilor sau a condițiilor de transport. Ajuși pe continentul American, erau vânduți în târguri specializate marilor proprietari de plantăjii.

În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce marii proprietari dețineau și către 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubre și muncesc până la epuizare. Primeau doar hrana. Majoritatea sclavilor erau incurajați să se căsătorească între ei, fiindcă se credea că aşa devineau mai docili și, totodată, dădeau naștere altor sclavi. Viața acestora era însă amenințată în permanență de violență sau separare, după cum dorea stăpânul. Înclusiv în secolul al XIX lea, când țările civilizate ale Europei renunțaseră la sclavi, în Statele Unite, sclavia era în floare și, mai mult decât atât, justificată de plantăjii.

În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce marii proprietari dețineau și către 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubre și muncesc până la epuizare. Primeau doar hrana. Majoritatea sclavilor erau incurajați să se căsătorească între ei, fiindcă se credea că aşa devineau mai docili și, totodată, dădeau naștere altor sclavi. Viața acestora era însă amenințată în permanență de violență sau separare, după cum dorea stăpânul. Înclusiv în secolul al XIX lea, când țările civilizate ale Europei renunțaseră la sclavi, în Statele Unite, sclavia era în floare și, mai mult decât atât, justificată de plantăjii.

Salveaza

Imaginiile se vor putea adăuga după salvarea lectiei, cu o descriere sugestiva.

-

+



Descriere imagine

Salveaza imagine

Pentru adăugarea unei lecții în baza de date se utilizează următoarea secvență de cod:

```

if (rtbContinutLectie.Text != "" && cbCapitole.Text != "" && cbLectii.Text != "")
{
    btnAddImg.Enabled = true;
    btnDelImg.Enabled = true;

    verificare = 1;

    //Se creeaza conexiunea la baza de date
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM capitole"; // se cauta capitolul

    int ok = 0; //determina daca exista sau nu capitolul

    MySqlDataReader r = cmd.ExecuteReader();
    while (r.Read())
    {
        Capitole capitol = new Capitole();
        {
            capitol.idC = r["idC"].ToString();
            capitol.titlu = r["titlu"].ToString();
        }

        if (capitol.titlu == cbCapitole.Text)
        {
            ok = 1;
            break;
        }
    }
    r.Close();

    if (ok == 0) //daca nu exista capitolul, se adauga in baza de date
    {
        cmd.CommandText = @"INSERT INTO capitole (titlu) VALUES (@ptitlu);";
    }
}

```

```

        cmd.Parameters.AddWithValue("ptitlu", cbCapitole.Text);
        cmd.ExecuteNonQuery();
    }

    Capitole c = new Capitole();

    cmd.CommandText = "SELECT * FROM capitole"; //se determina id-ul capitolului
    in care urmeaza sa fie adaugata lectia
    r = cmd.ExecuteReader();
    while(r.Read())
    {

        c.idC = r.GetString("idC");
        c.titlu = r.GetString("titlu");

        if(c.titlu==cbCapitole.Text)
        {
            break;
        }
    }
    r.Close();

    int exista = 0;
    //se verifica daca exista lectia deja
    cmd.CommandText = "SELECT * FROM lectii WHERE idC=@idC";
    //cmd.Parameters.AddWithValue("titlu", cbLectii.Text);
    cmd.Parameters.AddWithValue("idC",c.idC);
    r = cmd.ExecuteReader();
    while(r.Read())
    {
        if (r["titlu"].ToString()==cbLectii.Text)
        {
            MessageBox.Show("Lectia selectata exista deja. Daca doriti sa o modificati, va
rugam intrati la modificar lectie sau alegeti alt titlu pentru lectie.");
            exista++;
            break;
        }
    }
    r.Close();

    if(exista==0)
    {
        //se adauga lectia in baza de date
        cmd.CommandText = @"INSERT INTO lectii (idC, titlu, idProfesor,
data_adaugare) VALUES (@pidC, @pptitlu, @pidProfesor, now());";
    }
}

```

```

cmd.Parameters.AddWithValue("pidC", c.idC);
cmd.Parameters.AddWithValue("pptitlu", cbLectii.Text);
cmd.Parameters.AddWithValue("pidProfesor", this.Tag);
cmd.ExecuteNonQuery();

cmd.CommandText = "SELECT * FROM lectii"; // se determina id-ul lectiei
r = cmd.ExecuteReader();

Lectii lectie = new Lectii();
while (r.Read())
{
    lectie.idL = r.GetString("idL");
    lectie.idC = r.GetString("idC");
    lectie.titlu = r.GetString("titlu");
    lectie.idProfesor = r.GetString("idProfesor");
    lectie.data_adaugare = Convert.ToDateTime(r["data_adaugare"]);

    if (lectie.titlu == cbLectii.Text)
    {
        break;
    }
}
r.Close();

//se adauga continutul lectiei in baza de date
cmd.CommandText = @"INSERT INTO continut_lectii (idL, continut) VALUES
(@pidL, @pcontinut);";

cmd.Parameters.AddWithValue("pidL", lectie.idL);
cmd.Parameters.AddWithValue("pcontinut", rtbContinutLectie.Text);
cmd.ExecuteNonQuery();

//se adauga notificarea pentru adaugarea lectiei
cmd.CommandText = @"INSERT INTO notificari (idCont,tipNotificare,idLectie)
VALUES (@idCont,@ptip,@idL)";

MySqlConnection con = new MySqlConnection();
con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
con.Open(); //Se deschide conexiunea

MySqlCommand cm = new MySqlCommand();
cm.Connection = con;
cm.CommandText = "SELECT * FROM conturi";
MySqlDataReader read = cm.ExecuteReader();

```

```

while(read.Read())
{
    cmd.Parameters.AddWithValue("idCont", read["id"].ToString());
    cmd.Parameters.AddWithValue("ptip", "adaugareLectie");
    cmd.Parameters.AddWithValue("idL", lectie.idL);
    cmd.ExecuteNonQuery();
    cmd.Parameters.Clear();
}
read.Close();
con.Close();
conexiune.Close();

MessageBox.Show("Lectia a fost salvata cu succes.");

btnSalveazaImg.Enabled = true;
txtDescriereImg.Text = "Descriere imagine";
txtDescriereImg.Enabled = true;
rtbContinutLectie.Enabled = false;
btnSalveaza.Enabled = false;
}
}
else
{
    MessageBox.Show("Campurile cu denumirea capitolului, a lectiei si continutul lectie
trebuie completate inainte de a salva lectia.");
}

```

Pentru salvarea imaginii în baza de date se folosește următoarea secvență de cod:

```

if(txtDescriereImg.Text!="Descriere imagine" && lblCaleImg.Text!="Img/nopic.png")
{
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM lectii"; // se determina id-ul lectiei

    //Se executa comanda si se depune in r rezultatul
    MySqlDataReader r = cmd.ExecuteReader();
    Lectii lectie = new Lectii();
}

```

```

while (r.Read())
{
    lectie.idL = r.GetString("idL");
    lectie.idC = r.GetString("idC");
    lectie.titlu = r.GetString("titlu");
    lectie.idProfesor = r.GetString("idProfesor");
    lectie.data_adaugare = Convert.ToDateTime(r["data_adaugare"]);

    if(lectie.titlu==cbLectii.Text)
    {
        break;
    }
}
r.Close();

//se adauga imaginea in baza de date
cmd.CommandText = @"INSERT INTO imagini (idL, denumire, cale) VALUES
(@pidL, @pdenumire, @pcale)";
cmd.Parameters.AddWithValue("pidL",lectie.idL);
cmd.Parameters.AddWithValue("pdenumire", txtDescriereImg.Text);
cmd.Parameters.AddWithValue("pcale", lblCaleImg.Text);

cmd.ExecuteNonQuery();

conexiune.Close();

lblCaleImg.Text = "Img/nopic.png";
pbImgLectie.Image = new Bitmap(lblCaleImg.Text);
txtDescriereImg.Text = "Descriere imagine";

btnAddImg.Enabled = btnDelImg.Enabled = true;

MessageBox.Show("Imaginea a fost salvata cu succes!");
}

else
{
    MessageBox.Show("Trebuie sa adaugati o imagine si sa adaugati o descriere acesteia
inainte de a o salva");
}

```

Pentru adăugarea și ștergerea imaginilor din *PictureBox* se utilizează următoarele secvențe de cod:

```
//se incarca imaginea din calculator in PictureBox
1 reference
private void btnAddImg_Click(object sender, EventArgs e)
{
    //se deschide OpenFileDialog

    if (ofd.ShowDialog() == DialogResult.OK)
    {
        lblCaleImg.Text = "Img/" + ofd.SafeFileName;
        if (!File.Exists(lblCaleImg.Text))
        {
            File.Copy(ofd.FileName, lblCaleImg.Text);
        }
        pbImgLectie.Image = new Bitmap(lblCaleImg.Text);
    }
}

//se inlocuieste imaginea din PictureBox cu nopic
1 reference
private void btnDelImg_Click(object sender, EventArgs e)
{
    lblCaleImg.Text = "Img/nopic.png";
    pbImgLectie.Image = new Bitmap(lblCaleImg.Text);
}
```

3.7. Fereastra FrmModificaLectie

Prin intermediul acestei ferestre se pot modifica lecțiile de către profesori.

Fereastra conține:

- Cinci *Panel-uri*:

→ pnlCapitol în care se afișează un *ComboBox* pentru alegerea unui capitol (cbCapitole) și un *Label* cu textul “Capitole:”.

→ pnlLectie în care se afișează un *ComboBox* pentru alegerea unei lecții care face parte din capitolul selectat (cbLectii) și un *Label* cu textul “Lectii:”.

→ pnlContinutLectie care conține un *RichTextBox* pentru conținutul lecției (rtbContinutLectie), trei *Button-uri*: btnSalveaza pentru salvarea conținutului lecției modificat, btnAdauga pentru afișarea *Panel-ului* pnlAdaugaImg și btnStere pentru afișarea *Panel-ului* pnlStereImg. Aceasta mai conține și un *Label* care nu este vizibil pentru calea imaginii care se încarcă în *PictureBox* (lblCaleImg).

→ pnlStereImg care se afișează pentru ca utilizatorul să steargă imagini din baza de date care corespund lecției pe care acesta o modifică. Acest *Panel* conține un *PictureBox* pentru afișarea imaginilor (pbStereImg), un *ListBox* care nu este vizibil în care se încarcă din cod calea tuturor imaginilor care aparțin lecției sub formă de *Class* (lstImagini), un *TextBox* care este *Enable = false* și în care se încarcă descrierea imaginii (txtDescriere) și trei *Button-uri*: btnStereImg pentru ștergerea imaginii din baza de date, iar btnSt și btnDr pentru trecerea de la o imagine la alta.

→ pnlAdaugaImg care se afișează pentru ca utilizatorul să adauge imagini noi. Acest *Panel* conține un *PictureBox* pentru afișarea imaginilor (pbImgLectie), un *TextBox* care în care se adaugă descrierea imaginii (txtDescriere) și trei *Button-uri*: btnSalveazaImg pentru salvarea noii imagini, btnDellImg pentru ștergerea imaginii din *PictureBox* și btnAddImg pentru adăugarea unei imagini din calculator în *PictureBox*.

- Un *OpenFileDialog* (ofd) pentru adăugarea unei imagini de pe calculator în *PictureBox*.

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState

Istoria sclaviei

ModificaLectie

Capitol: Emanciparea

Lectie: Mișcarea pentru drepturile civile

Mișcarea pentru drepturile civile a fost o luptă pentru justiție și egalitate pentru afro-americanii, care a avut loc în principal în anii 1950 și 1960. Printre liderii săi s-au numărat Martin Luther King Jr., Malcolm X, Little Rock Nine, Rosa Parks și mulți alții. Războiul civil abolise oficial sclavia, dar nu a pus capăt discriminării împotriva oamenilor de culoare - au continuat să suporte efectele devastatoare ale racismului, în special în sud.

La mijlocul secolului al XX-lea, americanii negri avuseseră mai mult decât suficient prejudecăți și violență împotriva lor. Ei, alături de mulți americani albi, s-au mobilizat și au început o luptă fără precedent pentru egalitate care a durat două decenii.

Pe parcursul Reconstrucției Negrii au preluat roluri de conducere ca niciodată. Aceștia au deținut funcții publice și au căutat schimbări legislative pentru egalitate și dreptul la vot, entrându-și în marginaliza oamenii negri, a-i menținând separați de cei albi și a ștergând progresul pe care îl făcuseră în timpul Reconstrucției, legile „Jim Crow” au fost stabilite în sud începând cu sfârșitul secolului al XIX-lea. Oamenii negri nu puteau

Salveaza

Adauga imagine Sterge imagine



Martin Luther King Jr.

Sterge

ModificaLectie

Capitol: Emanciparea

Lectie: Mișcarea pentru drepturile civile

Mișcarea pentru drepturile civile a fost o luptă pentru justiție și egalitate pentru afro-americanii, care a avut loc în principal în anii 1950 și 1960. Printre liderii săi s-au numărat Martin Luther King Jr., Malcolm X, Little Rock Nine, Rosa Parks și mulți alții. Războiul civil abolise oficial sclavia, dar nu a pus capăt discriminării împotriva oamenilor de culoare - au continuat să suporte efectele devastatoare ale racismului, în special în sud.

La mijlocul secolului al XX-lea, americanii negri avuseseră mai mult decât suficient prejudecăți și violență împotriva lor. Ei, alături de mulți americani albi, s-au mobilizat și au început o luptă fără precedent pentru egalitate care a durat două decenii.

Pe parcursul Reconstrucției Negrii au preluat roluri de conducere ca niciodată. Aceștia au deținut funcții publice și au căutat schimbări legislative pentru egalitate și dreptul la vot, entrându-și în marginaliza oamenii negri, a-i menținând separați de cei albi și a ștergând progresul pe care îl făcuseră în timpul Reconstrucției, legile „Jim Crow” au fost stabilite în sud începând cu sfârșitul secolului al XIX-lea. Oamenii negri nu puteau

Salveaza

Adauga imagine Sterge imagine



NO IMAGE AVAILABLE

Descriere imagine

Salveaza

Pentru încărcarea lectiei din baza de date se utilizează următoarea secvență de cod:

```

if (cbLectii.SelectedIndex != -1)
{
    pnlContinutLectie.Visible = true;

    //Se creeaza conexiunea la baza de date
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM lectii";

    //Se executa comanda si se depune in r rezultatul
    MySqlDataReader r = cmd.ExecuteReader();

    Lectii lectie = new Lectii();

    while(r.Read())
    {
        lectie.idL = r["idL"].ToString();
        lectie.idC = r["idC"].ToString();
        lectie.titlu = r["titlu"].ToString();
        lectie.idProfesor = r["idProfesor"].ToString();
        lectie.data_adaugare = Convert.ToDateTime(r["data_adaugare"]);

        if(lectie.titlu==cbLectii.Text)
        {
            this.Tag = lectie.idL;
            break;
        }
    }
    r.Close();

    cmd.CommandText = "SELECT * FROM continut_lectii ";
    r = cmd.ExecuteReader();
    while(r.Read())
    {
        if (lectie.idL == r["idL"].ToString())
        {
            rtbContinutLectie.Text = r["continut"].ToString();
        }
    }
}

```

```

        break;
    }
}
r.Close();
conexiune.Close();
}
else
{
    MessageBox.Show("Selectati o lectie deja existenta, sau adaugati una noua
introducand titlul acesteia si apasand pe butonul +");
}

```

Pentru operațiile cu imagini se utilizează următoarele secvențe de cod:

```

private void btnAddImg_Click(object sender, EventArgs e)
{
    if (ofd.ShowDialog() == DialogResult.OK)
    {
        lblCaleImg.Text = "Img/" + ofd.SafeFileName;
        if (! File.Exists(lblCaleImg.Text))
        {
            File.Copy(ofd.FileName, lblCaleImg.Text);
        }
        pbImgLectie.Image = new Bitmap(lblCaleImg.Text);
    }
}

```

```

private void btnDelImg_Click(object sender, EventArgs e)
{
    lblCaleImg.Text = "Img/nopic.png";
    pbImgLectie.Image = new Bitmap(lblCaleImg.Text);
}

```

```

private void btnSalveazaImg_Click(object sender, EventArgs e)
{
    if (txtDescriereImg.Text != "Descriere imagine" && lblCaleImg.Text !=
"Img/nopic.png")
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea
    }
}

```

```

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;

cmd.CommandText = @"INSERT INTO imagini (idL, denumire, cale) VALUES
(@pidL, @pdenumire, @pcale)";
cmd.Parameters.AddWithValue("pidL", this.Tag.ToString());
cmd.Parameters.AddWithValue("pdenumire", txtDescriereImg.Text);
cmd.Parameters.AddWithValue("pcale", lblCaleImg.Text);

cmd.ExecuteNonQuery();

conexiune.Close();

lblCaleImg.Text = "Img/nopic.png";
pbImgLectie.Image = new Bitmap(lblCaleImg.Text);
txtDescriereImg.Text = "Descriere imagine";

MessageBox.Show("Imaginea a fost salvata cu succes!");
}
else
{
    MessageBox.Show("Trebuie sa adaugati o imagine si sa adaugati o descriere acesteia
inainte de a o salva");
}
}

private void IncarcaImg()
{
    Imagini img = lstImagini.Items[imgcurrenta] as Imagini;
    txtDescriere.Text = img.denumire;
    pbStergeImg.Image = new Bitmap(img.cale);
}

```

```

private void btnSt_Click(object sender, EventArgs e)
{
    imgcurrenta--;
    if (imgcurrenta >= 0)
    {
        IncarcaImg();
    }
    else
    {
        imgcurrenta = lstImagini.Items.Count - 1;
        IncarcaImg();
    }
}

```

```

int imgcurrenta = 0;
1 reference
private void btnDr_Click(object sender, EventArgs e)
{
    imgcurrenta++;
    if (imgcurrenta <= lstImagini.Items.Count - 1)
    {
        IncarcaImg();
    }
    else
    {
        imgcurrenta = 0;
        IncarcaImg();
    }
}

```

```

private void btnSergeImg_Click(object sender, EventArgs e)
{
    if (imgcurrenta > -1)
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;

        cmd.CommandText = "DELETE FROM imagini WHERE idL=@pidL";
        cmd.Parameters.AddWithValue("pidL", this.Tag);
        cmd.ExecuteNonQuery();
    }
}

```

```
conexiune.Close();

lstImagini.Items.Remove(lstImagini.Items[imgcurenta]);
MessageBox.Show("Imaginea a fost stearsa cu succes.");

if(imgcurenta+1<=lstImagini.Items.Count-1)
{
    imgcurenta++;
    IncarcaImg();
}
else
{
    if(imgcurenta-1>=0)
    {
        imgcurenta--;
        IncarcaImg();
    }
    else
    {
        pnlSergeImg.Visible = false;
        MessageBox.Show("Nu mai exista imagini pe care sa le puteti sterge.");
    }
}
}
```

3.8. Fereastra FrmModifOrdCapitole

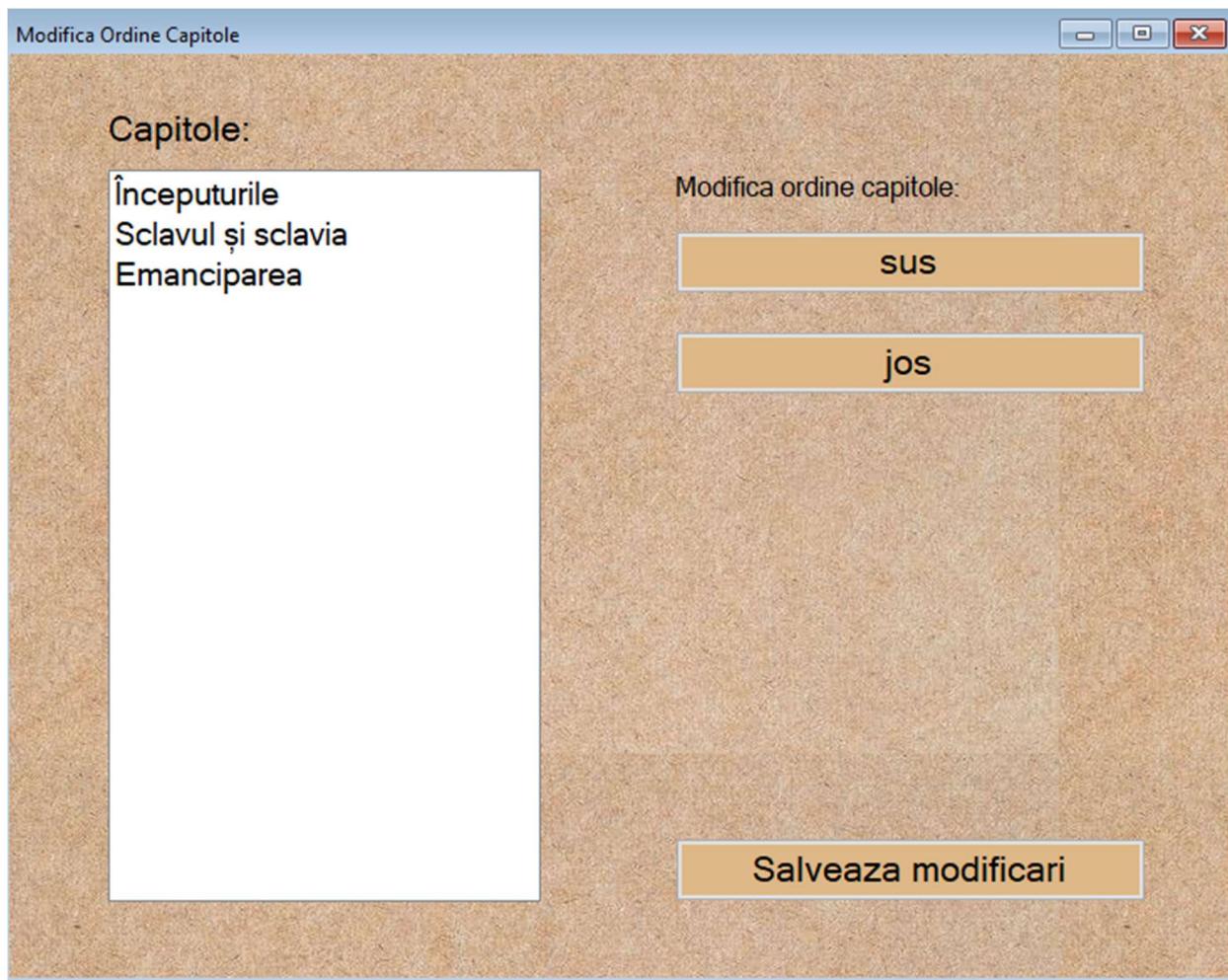
Prin intermediul acestei ferestre se poate modifica ordinea de afișare a capitolelor de către profesori.

Fereastra conține:

- Două *ListBox*-uri: lstCapitole în care se încarcă titlurile capitolelor din baza de date și lstIdc în care se încarcă id-ul capitolelor din baza de date, în aceeași ordine cu cele ale capitolelor care corespund.
- Trei *Button*-uri: btnSalveazaModif prin intermediul căruia se salvează modificările făcute, btnSus și btnJos prin care se modifică ordinea capitolelor în *ListBox*.
- Două *Label*-uri cu textele: “Capitole:” și “Modifica ordine capitole:”.

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



Pentru încărcarea capitolelor din baza de date se utilizează următoarea secvență de cod:

```
lstCapitole.Items.Clear();
lstIdc.Items.Clear();

MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea
MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM capitole";

MySqlDataReader r = cmd.ExecuteReader();
while(r.Read())
```

```

{
    Capitole c = new Capitole();
    {
        c.idC = r["idC"].ToString();
        c.titlu = r["titlu"].ToString();
    }
    lstCapitole.Items.Add(c);
}
r.Close();
conexiune.Close();

```

Pentru modificarea ordinii capitolilor și salvarea acesteia se utilizează următoarele secvențe de cod:

```

private void btnSus_Click(object sender, EventArgs e)
{
    if (lstCapitole.SelectedIndex > 0)
    {
        int index = lstCapitole.SelectedIndex;
        Capitole capitol = lstCapitole.SelectedItem as Capitole;

        lstCapitole.Items[index] = lstCapitole.Items[index - 1];
        lstCapitole.Items[index - 1] = capitol;

        lstCapitole.SelectedIndex = index - 1;
        lstCapitole.Update();
        lstCapitole.Focus();
    }
    else
    {
        if (lstCapitole.SelectedIndex == -1)
        {
            MessageBox.Show("Selectati capitolul pe care doriti sa il mutati cu o pozitie mai sus.");
        }
        else
        {
            MessageBox.Show("Nu puteti efectua aceasta modificare.");
        }
    }
}

```

```

private void btnJos_Click(object sender, EventArgs e)
{
    if (lstCapitole.SelectedIndex < lstCapitole.Items.Count - 1 &&
lstCapitole.SelectedIndex != -1)
    {
        int index = lstCapitole.SelectedIndex;
        Capitole capitol = lstCapitole.SelectedItem as Capitole;

        lstCapitole.Items[index] = lstCapitole.Items[index + 1];
        lstCapitole.Items[index + 1] = capitol;

        lstCapitole.SelectedIndex = index + 1;
        lstCapitole.Update();
        lstCapitole.Focus();
    }
    else
    {
        if (lstCapitole.SelectedIndex == -1)
        {
            MessageBox.Show("Selectati capitolul pe care doriti sa il mutati cu o pozitie mai
jos.");
        }
        else
        {
            MessageBox.Show("Nu puteti efectua aceasta modificare.");
        }
    }
}

```

```

private void btnSalveazaModif_Click(object sender, EventArgs e)
{
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM capitole";

    int index = 0;

    MySqlDataReader read = cmd.ExecuteReader();
    while(read.Read())
    {

```

```

Capitole c = lstCapitole.Items[index] as Capitole;
lstIdc.Items.Add(c);

MySqlConnection con = new MySqlConnection();
con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
con.Open(); //Se deschide conexiunea

MySqlCommand cmm = new MySqlCommand();
cmm.Connection = con;
cmm.CommandText = "UPDATE capitole SET titlu=@ptitlu WHERE idC=@pid";
cmm.Parameters.AddWithValue("ptitlu", c.titlu);
cmm.Parameters.AddWithValue("pid", read["idC"].ToString());
cmm.ExecuteNonQuery();
con.Close();
index++;
}
read.Close();

cmd.CommandText = "SELECT * FROM capitole";
MySqlDataReader r = cmd.ExecuteReader();
index = 0;
while(r.Read())
{
    MySqlConnection con = new MySqlConnection();
    con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
    con.Open(); //Se deschide conexiunea

    MySqlCommand cmm = new MySqlCommand();
    cmm.Connection = con;
    cmm.CommandText = "UPDATE capitole SET idC=@pidC WHERE titlu=@titlu";
    Capitole capitol = lstIdc.Items[index] as Capitole;
    cmm.Parameters.AddWithValue("pidC", capitol.idC);
    cmm.Parameters.AddWithValue("titlu", capitol.titlu);
    cmm.ExecuteNonQuery();
    con.Close();
    index++;
}
r.Close();
conexiune.Close();

```

```
    MessageBox.Show("Modificările au fost salvate cu succes.");
    this.Close();
}
```

3.9. Fereastra FrmModifOrdLectii

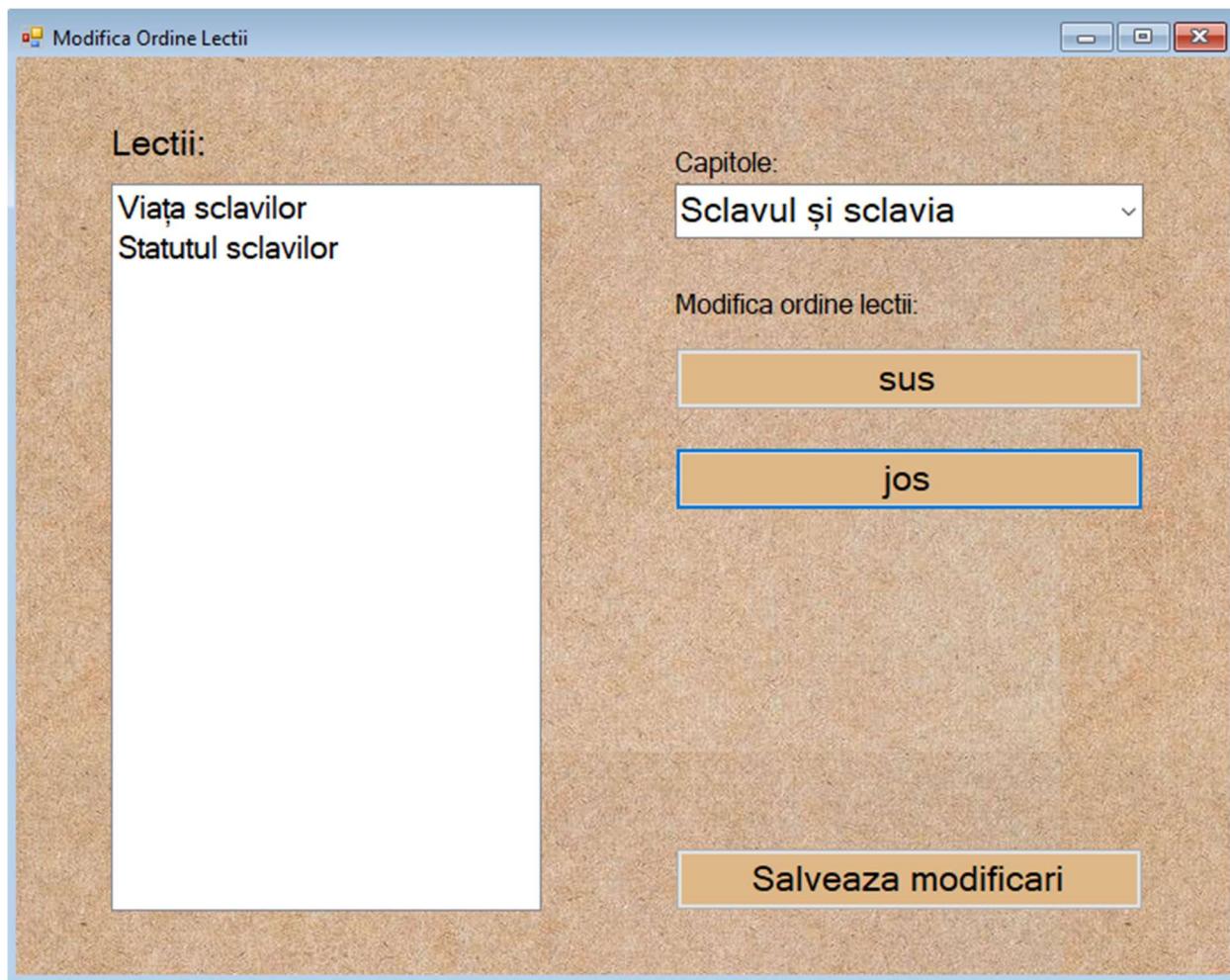
Prin intermediul acestei ferestre se poate modifica ordinea de afișare a lecțiilor de către profesori.

Fereastra conține:

- Două *ListBox*-uri: lstLectii în care se încarcă titlurile lecțiilor care aparțin capitolului selectat și lstIdL în care se încarcă id-ul lecțiilor din baza de date, în aceeași ordine cu cele ale lecțiilor care corespund.
- Trei *Button*-uri: btnSalveazaModif prin intermediul căruia se salvează modificările făcute, btnSus și btnJos prin care se modifică ordinea lecțiilor în *ListBox*.
- Trei *Label*-uri cu textele: “Lectii:”, “Capitole:” și “Modifica ordine lectii:”.
- Un *ComboBox* în care se încarcă titlurile capitolelor.

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



Pentru încărcarea capitolelor și a lecțiilor din baza de date se utilizează următoarea secvență de cod:

```
//Încărcarea capitolelor  
lstLectii.Items.Clear();  
lstIdL.Items.Clear();  
cbCapitole.Items.Clear();  
  
MySqlConnection conexiune = new MySqlConnection();  
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;  
UID=root; PASSWORD=; Allow User Variables=True";  
conexiune.Open(); //Se deschide conexiunea  
  
MySqlCommand cmd = new MySqlCommand();
```

```

cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM capitole";

MySqlDataReader r = cmd.ExecuteReader();
while (r.Read())
{
    Capitole c = new Capitole();
    {
        c.idC = r["idC"].ToString();
        c.titlu = r["titlu"].ToString();
    }
    cbCapitole.Items.Add(c);
}
r.Close();
conexiune.Close();

```

//Încărcarea lectiilor după ce s-a selectat un capitol

```

private void cbCapitole_SelectedIndexChanged(object sender, EventArgs e)
{
    if(cbCapitole.SelectedIndex!=-1)
    {
        lstLectii.Items.Clear();
        lstIdL.Items.Clear();

        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT * FROM lectii WHERE idC=@pidC";
        Capitole capitol = cbCapitole.SelectedItem as Capitole;
        cmd.Parameters.AddWithValue("pidC", capitol.idC);

        this.Tag = capitol.idC;

        MySqlDataReader r = cmd.ExecuteReader();
        while (r.Read())
        {
            Lectii l = new Lectii();
            {
                // l.id = r["id"].ToString();
                l.idL = r["idL"].ToString();
                l.idC = r["idC"].ToString();
            }
            lstLectii.Items.Add(l);
        }
    }
}

```

```
        l.titlu = r["titlu"].ToString();
        l.idProfesor = r["idProfesor"].ToString();
        l.data_adaugare = Convert.ToDateTime( r["data_adaugare"]);
    }
    lstLectii.Items.Add(l);
    lstIdL.Items.Add(l);
}
r.Close();
conexiune.Close();
}
```

Pentru salvarea modificărilor și a ordinii lecțiilor se utilizează următoarea secvență de cod:

```
private void btnSus_Click(object sender, EventArgs e)
{
    if (lstLectii.SelectedIndex > 0)
    {
        int index = lstLectii.SelectedIndex;
        Lectii lectie = lstLectii.SelectedItem as Lectii;

        lstLectii.Items[index] = lstLectii.Items[index - 1];
        lstLectii.Items[index - 1] = lectie;

        lstLectii.SelectedIndex = index - 1;
        lstLectii.Update();
        lstLectii.Focus();
    }
    else
    {
        if (lstLectii.SelectedIndex == -1)
        {
            MessageBox.Show("Selectati lectia pe care doriti sa o mutati cu o pozitie mai sus.");
        }
        else
        {
            MessageBox.Show("Nu puteti efectua aceasta modificare.");
        }
    }
}
```

```

private void btnJos_Click(object sender, EventArgs e)
{
    if (lstLectii.SelectedIndex < lstLectii.Items.Count - 1 && lstLectii.SelectedIndex != -1)
    {
        int index = lstLectii.SelectedIndex;
        Lectii lectie = lstLectii.SelectedItem as Lectii;

        lstLectii.Items[index] = lstLectii.Items[index + 1];
        lstLectii.Items[index + 1] = lectie;

        lstLectii.SelectedIndex = index + 1;
        lstLectii.Update();
        lstLectii.Focus();
    }
    else
    {
        if (lstLectii.SelectedIndex == -1)
        {
            MessageBox.Show("Selectati lectia pe care doriti sa o mutati cu o pozitie mai
jos.");
        }
        else
        {
            MessageBox.Show("Nu puteti efectua aceasta modificare.");
        }
    }
}

```

```

private void btnSalveazaModif_Click(object sender, EventArgs e)
{
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM lectii WHERE idC=@pidC";
    cmd.Parameters.AddWithValue("pidC", this.Tag);

    int index = 0;

    MySqlDataReader read = cmd.ExecuteReader();
    while (read.Read())
    {

```

```

Lectii l = lstLectii.Items[index] as Lectii;
Lectii ll = lstIdL.Items[index] as Lectii;
//lstIdL.Items.Add(l);

MySqlConnection con = new MySqlConnection();
con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
con.Open(); //Se deschide conexiunea

MySqlCommand cmm = new MySqlCommand();
cmm.Connection = con;
cmm.CommandText = "UPDATE lectii SET titlu=@ptitlu, idProfesor=@idProf,
data_adaugare=@pdata WHERE idL=@pid";
//cmm.Parameters.AddWithValue("pidC", l.idC);
cmm.Parameters.AddWithValue("ptitlu", l.titlu);
cmm.Parameters.AddWithValue("idProf", l.idProfesor);
cmm.Parameters.AddWithValue("pdata", l.data_adaugare);
cmm.Parameters.AddWithValue("pid", ll.idL);
//cmm.Parameters.AddWithValue("pid", l.id);
cmm.ExecuteNonQuery();
con.Close();
index++;
}
read.Close();

cmd.CommandText = "SELECT * FROM lectii WHERE idC=@pid";
cmd.Parameters.AddWithValue("pid", this.Tag);
MySqlDataReader r = cmd.ExecuteReader();
index = 0;
while (r.Read())
{
    MySqlConnection con = new MySqlConnection();
    con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
    con.Open(); //Se deschide conexiunea

    MySqlCommand cmm = new MySqlCommand();
    cmm.Connection = con;
    cmm.CommandText = "UPDATE lectii SET idL=@pidL WHERE titlu=@titlu";
    Lectii lectie = lstLectii.Items[index] as Lectii;
    cmm.Parameters.AddWithValue("pidL", lectie.idL);
    cmm.Parameters.AddWithValue("titlu", lectie.titlu);
    cmm.ExecuteNonQuery();
    con.Close();
    index++;
}

```

```
r.Close();
conexiune.Close();

MessageBox.Show("Modificările au fost salvate cu succes.");
this.Close();
}
```

3.10. Fereastra FrmStergeLectie

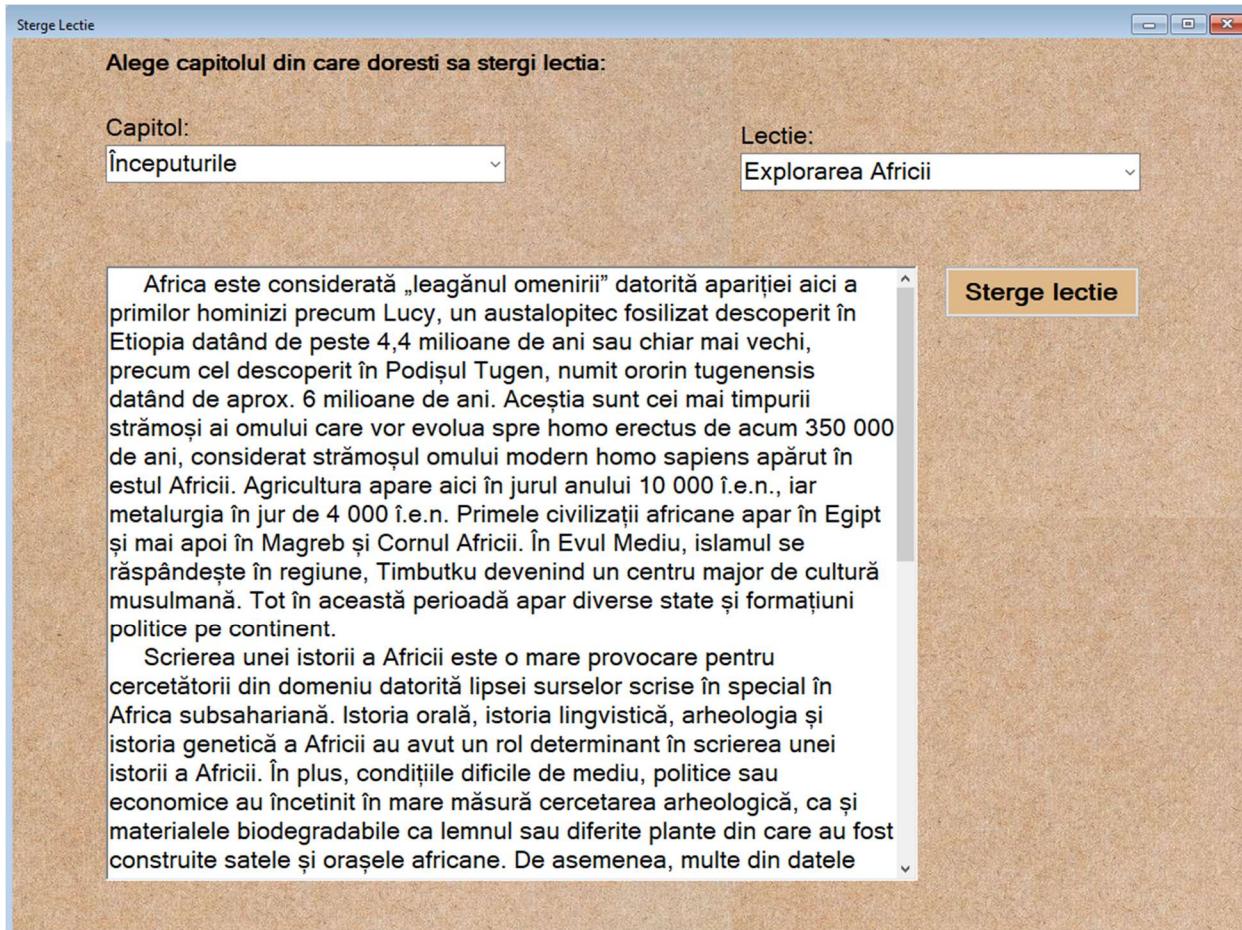
Prin intermediul acestei ferestre se pot șterge lecții din baza de date de către profesori.

Fereastra conține:

- Un *Label* care conține textul: “Alege capitolul din care doresti sa stergi lectia:”.
- Trei *Panel*-uri:
 - pnlCapitol în care se afișează un *ComboBox* pentru alegerea unui capitol (cbapitole) și un *Label* cu textul “Capitole:”.
 - pnlLectie în care se afișează un *ComboBox* pentru alegerea unei lecții care face parte din capitolul selectat (cbLectii) și un *Label* cu textul “Lectii:”.
 - pnlContinut care conține un *RichTextBox* în care se afișează conținutul lecției selectate (rtbLectie) și un *Button* cu ajutorul căreia se șterge lecția din baza de date (btnStergeLectie).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



Pentru încărcarea capitolelor și a lecțiilor, cât și conținutul lecției selectate și ștergerea lecției din baza de date se utilizează următoarele secvențe de cod:

//Încărcarea capitolelor

```
//Se creeaza conexiunea la baza de date
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM capitole";

//Se executa comanda si se depune in r rezultatul
MySqlDataReader r = cmd.ExecuteReader();
```

```

while (r.Read())
{
    Capitole capitol = new Capitole();
    {
        capitol.idC = r.GetString("idC");
        capitol.titlu = r.GetString("titlu");
    }

    cbCapitole.Items.Add(capitol);

}
r.Close();
conexiune.Close();

```

//Încărcarea lecțiilor

```

private void cbCapitole_SelectedIndexChanged(object sender, EventArgs e)
{
    if (cbCapitole.SelectedIndex != -1)
    {
        cbLectii.Text = "";
        cbLectii.Items.Clear();

        int ok = 0;

        Capitole capitol = cbCapitole.SelectedItem as Capitole;

        //Se creeaza conexiunea la baza de date
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT * FROM lectii WHERE
idProfesor=@pidProfesor";

        cmd.Parameters.AddWithValue("pidProfesor", this.Tag);

        //Se executa comanda si se depune in r rezultatul
        MySqlDataReader r = cmd.ExecuteReader();
        while (r.Read())
        {

```

```

Lectii lectie = new Lectii();
{
    lectie.idL = r.GetString("idL");
    lectie.idC = r.GetString("idC");
    lectie.titlu = r.GetString("titlu");
    lectie.idProfesor = r.GetString("idProfesor");
    lectie.data_adaugare = Convert.ToDateTime(r["data_adaugare"]);
}

if (lectie.idC == capitol.idC)
{
    cbLectii.Items.Add(lectie);
    ok++;
}
r.Close();
conexiune.Close();

if(ok!=0)
{
    pnlLectie.Visible = true;
}
else
{
    MessageBox.Show("Nu aveti lectii adaugate in acest capitol.");
    pnlLectie.Visible = false;
}
}
else
{
    MessageBox.Show("Selectati un capitol deja existent, sau adaugati unul nou
introducand titlul acestuia si apasand pe butonul +");
}
}

```

//Încărcarea conținutului lecției

```

private void cbLectii_SelectedIndexChanged(object sender, EventArgs e)
{
    if (cbLectii.SelectedIndex != -1)
    {

        pnlContinut.Visible = true;

        //Se creeaza conexiunea la baza de date
        MySqlConnection conexiune = new MySqlConnection();

```

```

conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

Lectii l = cbLectii.SelectedItem as Lectii;

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM continut_lectii WHERE idL=@pidL";

cmd.Parameters.AddWithValue("pidL", l.idL);

//Se executa comanda si se depune in r rezultatul
MySqlDataReader r = cmd.ExecuteReader();
while(r.Read())
{
    Continut_lectii cl = new Continut_lectii();
    {
        cl.id = r["id"].ToString();
        cl.idL = r["idL"].ToString();
        cl.continut = r["continut"].ToString();
    }

    rtbLectie.Text = cl.ToString();
    rtbLectie.Tag = cl.idL;
}

}
else
{
    MessageBox.Show("Selectati lectia pe care doriti sa o stergeti");
}
}

```

//Ștergerea lectiei din baza de date

```

private void btnStereLectie_Click(object sender, EventArgs e)
{
    //Se creeaza conexiunea la baza de date
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();

```

```
cmd.Connection = conexiune;
cmd.CommandText = "DELETE FROM lectii WHERE idL=@pidL";
cmd.Parameters.AddWithValue("pidL", rtbLectie.Tag);
cmd.ExecuteNonQuery();

cmd.CommandText = "DELETE FROM continut_lectii WHERE idL=@ppidL";
cmd.Parameters.AddWithValue("ppidL", rtbLectie.Tag);
cmd.ExecuteNonQuery();

cmd.CommandText = "DELETE FROM imagini WHERE idL=@idL";
cmd.Parameters.AddWithValue("idL", rtbLectie.Tag);
cmd.ExecuteNonQuery();

conexiune.Close();

MessageBox.Show("Lectia a fost stearsa cu succes!");
rtbLectie.Text = "";
rtbLectie.Tag = null;
cbLectii.Text = "";
cbLectii.Items.Clear();
pnlContinut.Visible = false;
}
```

3.11. Fereastra FrmAdaugaIntrebari

Prin intermediul acestei ferestre profesorii pot adăuga întrebări în baza de date, întrebări care vor fi folosite pentru crearea testelor și pentru jocul de puzzle.

Fereastra conține:

- Două *Label*-uri cu textele: “Selecteaza capitolul din care sa faca parte intrebarea:” și “Selecteaza lectia din care sa faca parte intrebarea.”.
- Patru *Panel*-uri:

→ pnlIntrebare care conține două *Label*-uri, un *RichTextBox* în care să se adauge întrebarea (rtbIntrebare) și trei *RadioButton*-uri care corespund tipului de răspuns pe care utilizatorul dorește să-l aibă întrebarea (rbTextScurt, rbRaspUnic, rbRaspMultiplu).

→ pnlTextScurt care conține două *Label*-uri și un *TextBox* în care să se adauge răspunsul corect (txtRaspTextScurt).

→ pnlRaspUnic care conține cinci *Label*-uri, trei *TextBox*-uri, câte unul pentru fiecare variantă de răspuns (txtV1, txtV2, txtV3) și trei *RadioButton*-uri, fiecare fiind corespondentul unei variante de răspuns, utilizatorul selectând varianta corectă de răspuns cu ajutorul acestora (rbVar1, rbVar2, rbVar3).

→ pnlRaspMultiplu care conține cinci *Label*-uri, trei *TextBox*-uri, câte unul pentru fiecare variantă de răspuns (txtMV1, txtMV2, txtMV3) și trei *CheckBox*-uri, fiecare fiind corespondentul unei variante de răspuns, iar utilizatorul selectează varianta corectă/variantele corecte de răspuns cu ajutorul acestora (cbV1, cbV2, cbV3).

- Un *Button* cu ajutorul căruia se salvează întrebarea în baza de date (btnAdIntr).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- *BackgroundImage*
- *BackgroundImageLayout*
- *FormBorderStyle*
- *StartPosition*
- *WindowState*

Istoria sclaviei

Adauga Intrebări

Selecteaza capitolul din care sa faca parte intrebarea: Sclavul și sclavia

Selecteaza lectia din care sa faca parte intrebarea: Statutul sclavilor

Selecteaza tipul de raspuns al intrebarii:

Text scurt Raspuns unic Raspuns multiplu

Intrebare:

În ce an a fost adoptat al 13-lea Amendament?

Adaugati un raspuns scurt care poate fi verificat. In caz contrar, raspunsul va fi corectat gresit, iar elevului nu i se va acorda punctajul pentru raspuns, chiar daca acesta este corect.

Raspuns: 1865

Adauga intrebare

Adauga Intrebări

Selecteaza capitolul din care sa faca parte intrebarea: Emanciparea

Selecteaza lectia din care sa faca parte intrebarea: Mișcarea pentru drepturile civile

Selecteaza tipul de raspuns al intrebarii:

Text scurt Raspuns unic Raspuns multiplu

Intrebare:

Când au fost stabilite legile "Jim Crow"?

Variante de raspuns:

varianta 1
la sfârșitul secolului XVIII

varianta 2
la începutul secolului XX

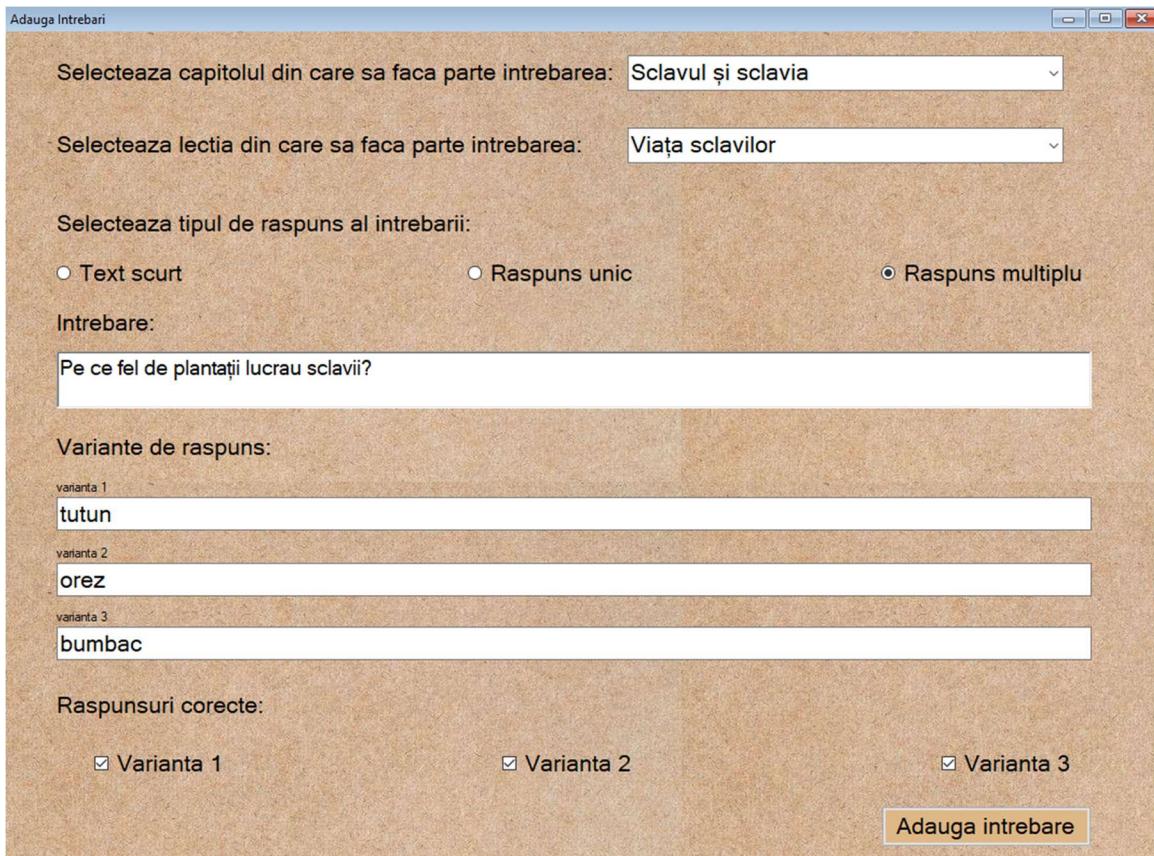
varianta 3
la sfârșitul secolului XIX

Raspuns corect:

varianta 1 varianta 2 varianta 3

Adauga intrebare

Istoria sclaviei



Pentru încărcarea capitolelor și lecțiilor am utilizat următoarele secvențe de cod:

```
//Încărcarea capitolelor și a tuturor lecțiilor

//Se incarca capitoile in ComboBox
cbCapitole.Items.Add("General");

//Se creeaza conexiunea la baza de date
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM capitole";
```

```

//Se executa comanda si se depune in r rezultatul

MySqlDataReader r = cmd.ExecuteReader();
while (r.Read())
{
    Capitole capitol = new Capitole();
    {
        capitol.idC = r.GetString("idC");
        capitol.titlu = r.GetString("titlu");
    }

    cbCapitole.Items.Add(capitol);

}

r.Close();
conexiune.Close();

//se adauga lectiile in ComboBox
cbLectii.Items.Add("General");

//Se creeaza conexiunea la baza de date
conexiune.Open(); //Se deschide conexiunea

cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM lectii";

//Se executa comanda si se depune in r rezultatul
r = cmd.ExecuteReader();
while (r.Read())
{
    Lectii lectie = new Lectii();
    {
        lectie.idL = r.GetString("idL");
        lectie.idC = r.GetString("idC");
        lectie.titlu = r.GetString("titlu");
        lectie.idProfesor = r.GetString("idProfesor");
        lectie.data_adaugare = Convert.ToDateTime(r["data_adaugare"]);
    }
    cbLectii.Items.Add(lectie);
}
r.Close();
conexiune.Close();

```

//Încărcarea lectiilor după ce s-a selectat un anumit capitol

```

private void cbCapitole_SelectedIndexChanged(object sender, EventArgs e)
{
    if(cbCapitole.SelectedIndex!=-1 && cbCapitole.Text!="General")
    {
        cbLectii.Items.Clear();
        cbLectii.Items.Add("General");

        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT * FROM lectii";

        Capitole c = cbCapitole.SelectedItem as Capitole;

        MySqlDataReader r = cmd.ExecuteReader();
        while(r.Read())
        {
            Lectii l = new Lectii();
            {
                l.idL = r["idL"].ToString();
                l.idC = r["idC"].ToString();
                l.titlu = r["titlu"].ToString();
            }

            if(c.idC==l.idC)
            {
                cbLectii.Items.Add(l);
            }
        }
        r.Close();
        conexiune.Close();
    }
}

```

Pentru salvarea întrebărilor s-a utilizat următoarea secvență de cod:

```

private void btnAdIntr_Click(object sender, EventArgs e)
{
    if(cbCapitole.Text=="")
    {
        cbCapitole.Text = "General";
    }
    if(cbLectii.Text=="")
    {
        cbLectii.Text = "General";
    }

    if(rtbIntrebare.Text!="" && ((txtV1.Text!="" && txtV2.Text!="" && txtV3.Text!=""
&& (rbVar1.Checked==true || rbVar2.Checked==true || rbVar3.Checked==true)) ||
(txtMV1.Text!="" && txtMV2.Text!="" && txtMV3.Text!="" && (cbV1.Checked==true ||
cbV2.Checked==true || cbV3.Checked==true)) || txtRaspTextScurt.Text!=""))
    {
        //Se creeaza conexiunea la baza de date
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = @"INSERT INTO intrebari
(idProfesor,intrebare,capitol,lectie) VALUES (@pidProfesor, @pintrebare, @pcapitol,
@plectie)";

        cmd.Parameters.AddWithValue("pidProfesor", this.Tag);
        cmd.Parameters.AddWithValue("pintrebare", rtbIntrebare.Text);
        cmd.Parameters.AddWithValue("pcapitol", cbCapitole.Text);
        cmd.Parameters.AddWithValue("plectie", cbLectii.Text);
        cmd.ExecuteNonQuery();

        cmd.CommandText = "SELECT * FROM intrebari";
        MySqlDataReader r = cmd.ExecuteReader();

        while (r.Read())
        {
            if (r["idProfesor"].ToString() == this.Tag.ToString() && r["intrebare"].ToString()
== rtbIntrebare.Text)
            {
                idIntr = r["idIntrebare"].ToString();
                break;
            }
        }
    }
}

```

```

r.Close();

if (rbTextScurt.Checked == true)
{
    cmd.CommandText = @"INSERT INTO raspunsuri (idIntrebare, raspuns,
tipRaspuns,corect) VALUES (@ppidIntrebare, @praspuns, @ptipRaspuns, @pcorrect)";

    cmd.Parameters.AddWithValue("ppidIntrebare", idIntr);
    cmd.Parameters.AddWithValue("praspuns", txtRaspTextScurt.Text);
    cmd.Parameters.AddWithValue("ptipRaspuns", "text scurt");
    cmd.Parameters.AddWithValue("pcorrect", "da");

    cmd.ExecuteNonQuery();
}

if (rbRaspUnic.Checked == true)
{
    //se adauga prima varianta de raspuns
    cmd.CommandText = @"INSERT INTO raspunsuri (idIntrebare, raspuns,
tipRaspuns,corect) VALUES (@pp1idIntrebare, @p1raspuns, @p1tipRaspuns, @p1correct)";

    cmd.Parameters.AddWithValue("pp1idIntrebare", idIntr);
    cmd.Parameters.AddWithValue("p1raspuns", txtV1.Text);
    cmd.Parameters.AddWithValue("p1tipRaspuns", "unic");
    if (rbVar1.Checked == true)
    {
        cmd.Parameters.AddWithValue("p1correct", "da");
    }
    else
    {
        cmd.Parameters.AddWithValue("p1correct", "nu");
    }

    cmd.ExecuteNonQuery();

    //se adauga a doua varianta de raspuns
    cmd.CommandText = @"INSERT INTO raspunsuri (idIntrebare, raspuns,
tipRaspuns,corect) VALUES (@pp2idIntrebare, @p2raspuns, @p2tipRaspuns, @p2correct)";

    cmd.Parameters.AddWithValue("pp2idIntrebare", idIntr);
    cmd.Parameters.AddWithValue("p2raspuns", txtV2.Text);
    cmd.Parameters.AddWithValue("p2tipRaspuns", "unic");
    if (rbVar2.Checked == true)
    {
        cmd.Parameters.AddWithValue("p2correct", "da");
    }
}

```

```

else
{
    cmd.Parameters.AddWithValue("p2corect", "nu");
}

cmd.ExecuteNonQuery();

//se adauga a treia varianta de raspuns
cmd.CommandText = @"INSERT INTO raspunsuri (idIntrebare, raspuns,
tipRaspuns,corect) VALUES (@pp3idIntrebare, @p3raspuns, @p3tipRaspuns, @p3corect)";

cmd.Parameters.AddWithValue("pp3idIntrebare", idIntr);
cmd.Parameters.AddWithValue("p3raspuns", txtV3.Text);
cmd.Parameters.AddWithValue("p3tipRaspuns", "unic");
if (rbVar3.Checked == true)
{
    cmd.Parameters.AddWithValue("p3corect", "da");
}
else
{
    cmd.Parameters.AddWithValue("p3corect", "nu");
}

cmd.ExecuteNonQuery();
}

if (rbRaspMultiplu.Checked == true)
{
    //se adauga prima varianta de raspuns
    cmd.CommandText = @"INSERT INTO raspunsuri (idIntrebare, raspuns,
tipRaspuns,corect) VALUES (@ppidIntrebare, @praspuns, @ptipRaspuns, @pcorrect)";

    cmd.Parameters.AddWithValue("ppidIntrebare", idIntr);
    cmd.Parameters.AddWithValue("praspuns", txtMV1.Text);
    cmd.Parameters.AddWithValue("ptipRaspuns", "multiplu");
    if (cbV1.Checked == true)
    {
        cmd.Parameters.AddWithValue("pcorrect", "da");
    }
    else
    {
        cmd.Parameters.AddWithValue("pcorrect", "nu");
    }

    cmd.ExecuteNonQuery();
}

```

```

//se adauga a doua varianta de raspuns
cmd.CommandText = @"INSERT INTO raspunsuri (idIntrebare, raspuns,
tipRaspuns,corect) VALUES (@pp2idIntrebare, @p2raspuns, @p2tipRaspuns, @p2corect)";

cmd.Parameters.AddWithValue("pp2idIntrebare", idIntr);
cmd.Parameters.AddWithValue("p2raspuns", txtMV2.Text);
cmd.Parameters.AddWithValue("p2tipRaspuns", "multiplu");
if (cbV2.Checked == true)
{
    cmd.Parameters.AddWithValue("p2corect", "da");
}
else
{
    cmd.Parameters.AddWithValue("p2corect", "nu");
}

cmd.ExecuteNonQuery();

//se adauga a treia varianta de raspuns
cmd.CommandText = @"INSERT INTO raspunsuri (idIntrebare, raspuns,
tipRaspuns,corect) VALUES (@pp3idIntrebare, @p3raspuns, @p3tipRaspuns, @p3corect)";

cmd.Parameters.AddWithValue("pp3idIntrebare", idIntr);
cmd.Parameters.AddWithValue("p3raspuns", txtMV3.Text);
cmd.Parameters.AddWithValue("p3tipRaspuns", "multiplu");
if (cbV3.Checked == true)
{
    cmd.Parameters.AddWithValue("p3corect", "da");
}
else
{
    cmd.Parameters.AddWithValue("p3corect", "nu");
}

cmd.ExecuteNonQuery();
}

conexiune.Close();
MessageBox.Show("Intrebarea a fost adaugata cu succes!");

//se adauga notificare pentru adaugare intrebare
MySqlConnection c = new MySqlConnection();
c.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
c.Open(); //Se deschide conexiunea

MySqlCommand cmdd = new MySqlCommand();

```

```

cmdd.Connection = c;
cmdd.CommandText = @"INSERT INTO notificari
(idCont,tipNotificare,idIntrebare) VALUES (@idCont,@ptip,@idIntr)";

MySqlConnection conn = new MySqlConnection();
conn.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conn.Open(); //Se deschide conexiunea

MySqlCommand cmm = new MySqlCommand();
cmm.Connection = conn;
cmm.CommandText = "SELECT * FROM conturi";
MySqlDataReader read = cmm.ExecuteReader();
while (read.Read())
{
    cmdd.Parameters.AddWithValue("idCont", read["id"].ToString());
    cmdd.Parameters.AddWithValue("ptip", "adaugaIntrebare");
    cmdd.Parameters.AddWithValue("idIntr", idIntr);
    cmdd.ExecuteNonQuery();
    cmdd.Parameters.Clear();
}
read.Close();
conn.Close();

FrmAdaugaIntrebari_Load(sender,e); //dupa adaugarea intrebării în bază de date, tot
continutul ferestrei apare precum la încarcarea ferestrei
}

else
{
    MessageBox.Show("Pentru a salva întrebarea, toate campurile trebuie completate
OBLIGATORIU.");
}
}
}

```


3.12. Fereastra FrmModificaIntrebari

Prin intermediul acestei ferestre profesorii pot modifica întrebările din baza de date.

Fereastra conține:

- Un *Label* cu textul: “Intrebări:”.
- Cinci *Panel*-uri:

→ pnlIntrebare care conține un *Label* și un *RichTextBox* în care să se adauge întrebarea (rtbIntrebare).

→ pnlTextScurt care conține două *Label*-uri și un *TextBox* în care să se adauge răspunsul corect (txtRaspTextScurt).

→ pnlRaspUnic care conține cinci *Label*-uri, trei *RichTextBox*-uri, câte unul pentru fiecare variantă de răspuns (rtbVar1, rtbVar2, rtbVar3) și trei *RadioButton*-uri, fiecare fiind corespondentul unei variante de răspuns, cu ajutorul cărora utilizatorul poate modifica varianta corectă de răspuns (rbVar1, rbVar2, rbVar3).

→ pnlRaspMultiplu care conține cinci *Label*-uri, trei *RichTextBox*-uri, câte unul pentru fiecare variantă de răspuns (rtbMV1, rtbMV2, rtbMV3) și trei *CheckBox*-uri, fiecare fiind corespondentul unei variante de răspuns, cu ajutorul cărora utilizatorul poate modifica varianta corectă/variantele corecte de răspuns (cbV1, cbV2, cbV3).

→ pnlModificaCL care conține două *label*-uri, două *ComboBox*-uri, unul în care se încarcă capitolele și unul în care se încarcă lecțiile (cbCapitole, cbLectii) și un *Button* pentru salvarea capitolului și lecției din care face parte întrebarea (btnSalveazaModif).

- Un *ListBox* în care se încarcă lista întrebărilor din baza de date (lstIntrebari).
- Două *Button*-uri: unul pentru a se afișa *Panel*-ul de modificare a capitolului și lecției (btnModifCL) și un *Button* pentru salvarea modificărilor făcute (btnModifIntr).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- *BackgroundImage*
- *BackgroundImageLayout*
- *FormBorderStyle*
- *StartPosition*
- *WindowState*

Istoria sclaviei

FrmModificaIntrebari

Intrebări:

Scîeți termenul la care se referă definiția:
Pe ce fel de plantații lucrau sclavi?
Ce doreau să obțină marii stăpâni de sclavi?
În ce an a avut loc Revoluția Americană?
În ce an a fost adoptat al 13-lea Amendament?
Ce a dus la mișcarea drepturilor civile?
Cine a susținut emanciparea?
Prin adoptarea celui de al 13-lea Amendament,
Ce a reprezentat "Mișcarea pentru drepturi civile"?
Când au fost stabilite legile "Jim Crow"?

Intrebare:

Când au fost stabilite legile "Jim Crow"?

Variante de răspuns:

variantă 1
la sfârșitul secolului XVIII

variantă 2
la începutul secolului XX

variantă 3
la sfârșitul secolului XIX

Răspuns corect:

la sfârșitul secolului XVIII
 la începutul secolului XX
 la sfârșitul secolului XIX

Modifica capitol/lectie **Modifica intrebare**

FrmModificaIntrebari

Intrebări:

Scîeți termenul la care se referă definiția:
Pe ce fel de plantații lucrau sclavi?
Ce doreau să obțină marii stăpâni de sclavi?
În ce an a avut loc Revoluția Americană?
În ce an a fost adoptat al 13-lea Amendament?
Ce a dus la mișcarea drepturilor civile?
Cine a susținut emanciparea?
Prin adoptarea celui de al 13-lea Amendament,
Ce a reprezentat "Mișcarea pentru drepturile civile"?
Când au fost stabilite legile "Jim Crow"?

Intrebare:

Când au fost stabilite legile "Jim Crow"?

Selecteaza capitolul:

Emanciparea

Selecteaza lectia:

Mișcarea pentru drepturile civile

Salveaza modificari **Modifica capitol/lectie** **Modifica intrebare**

Pentru încărcarea întrebărilor, a capitolelor și a lecțiilor din baza de date se utilizează următoarele linii de cod:

```
//incarcare intrebari in ListBox + resetare lista
lstIntrebari.Items.Clear();

//Se creeaza conexiunea la baza de date
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM intrebari WHERE
idProfesor=@pidProfesor";
cmd.Parameters.AddWithValue("pidProfesor", this.Tag.ToString());

MySqlDataReader r = cmd.ExecuteReader();
while(r.Read())
{
    Intrebari intr = new Intrebari();
    {
        intr.idIntrebare = r["idIntrebare"].ToString();
        intr.idProfesor = r["idProfesor"].ToString();
        intr.intrebare = r["intrebare"].ToString();
        intr.capitol = r["capitol"].ToString();
        intr.lectie = r["lectie"].ToString();
    }

    lstIntrebari.Items.Add(intr);
}
conexiune.Close();

//Încărcarea datelor legate de întrebarea selectată

private void lstIntrebari_SelectedIndexChanged(object sender, EventArgs e)
{
    if(lstIntrebari.SelectedIndex!=-1)
    {
        Intrebari intr = lstIntrebari.SelectedItem as Intrebari;

        btnModifCL.Visible = btnModifIntr.Visible = true;
    }
}
```

```

pnlIntrebare.Visible = true;
rtbIntrebare.Tag = intr.idIntrebare.ToString();
rtbIntrebare.Text = intr.intrebare.ToString();

//se incarca raspunsurile
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM raspunsuri WHERE
idIntrebare=@pidIntrebare";
cmd.Parameters.AddWithValue("pidIntrebare", rtbIntrebare.Tag.ToString());

int ok = 1;

MySqlDataReader r = cmd.ExecuteReader();
while(r.Read())
{
    Raspunsuri rasp = new Raspunsuri();
    {
        rasp.idRaspuns = r["idRaspuns"].ToString();
        rasp.idIntrebare = r["idIntrebare"].ToString();
        rasp.raspuns = r["raspuns"].ToString();
        rasp.tipRaspuns = r["tipRaspuns"].ToString();
        rasp.corect = r["corect"].ToString();
    }

    if(rasp.tipRaspuns == "text scurt")
    {
        pnlTextScurt.Visible = true;
        pnlRaspUnic.Visible = pnlRaspMultiplu.Visible = pnlModifCL.Visible = false;

        txtRaspTextScurt.Text = rasp.raspuns.ToString();

    }
    else
    {
        if(rasp.tipRaspuns == "unic")
        {
            pnlRaspUnic.Visible = true;
            pnlTextScurt.Visible = pnlRaspMultiplu.Visible = pnlModifCL.Visible =
false;
        }
    }
}

```

```
if(ok==1)
{
    rtbVar1.Text = rasp.raspuns.ToString();
    rtbVar1.Tag = rasp.idRaspuns.ToString();
    rbVar1.Text = rtbVar1.Text.ToString();
    if(rasp.corect=="da")
    {
        rbVar1.Checked = true;
    }
    else
    {
        rbVar1.Checked = false;
    }
}
else
{
    if(ok==2)
    {
        rtbVar2.Text = rasp.raspuns.ToString();
        rtbVar2.Tag = rasp.idRaspuns.ToString();
        rbVar2.Text = rtbVar2.Text.ToString();
        if (rasp.corect=="da")
        {
            rbVar2.Checked = true;
        }
        else
        {
            rbVar2.Checked = false;
        }
    }
    else
    {
        if(ok==3)
        {
            rtbVar3.Text = rasp.raspuns.ToString();
            rtbVar3.Tag = rasp.idRaspuns.ToString();
            rbVar3.Text = rtbVar3.Text.ToString();
            if (rasp.corect=="da")
            {
                rbVar3.Checked = true;
            }
            else
            {
                rbVar3.Checked = false;
            }
        }
    }
}
```

```

        }
    }
    ok++;
}
else
{
    pnlRaspMultiplu.Visible = true;
    pnlTextScurt.Visible = pnlRaspUnic.Visible = pnlModifCL.Visible = false;

    if(ok==1)
    {
        rtbMV1.Text = rasp.raspuns.ToString();
        rtbMV1.Tag = rasp.idRaspuns.ToString();
        cbV1.Text = rtbMV1.Text;
        if(rasp.corect=="da")
        {
            cbV1.Checked = true;
        }
        else
        {
            cbV1.Checked = false;
        }
    }
    else
    {
        if (ok == 2)
        {
            rtbMV2.Text = rasp.raspuns.ToString();
            rtbMV2.Tag = rasp.idRaspuns.ToString();
            cbV2.Text = rtbMV2.Text;
            if (rasp.corect == "da")
            {
                cbV2.Checked = true;
            }
            else
            {
                cbV2.Checked = false;
            }
        }
        else
        {
            if (ok == 3)
            {
                rtbMV3.Text = rasp.raspuns.ToString();
                rtbMV3.Tag = rasp.idRaspuns.ToString();
            }
        }
    }
}

```

```

        cbV3.Text = rtbMV3.Text;
        if (rasp.corect == "da")
        {
            cbV3.Checked = true;
        }
        else
        {
            cbV3.Checked = false;
        }
    }

    ok++;
}

}

conexiune.Close();

}
}

```

//Încărcarea capitolului și lectiei din care face parte întrebarea

```

string id;
private void btnModifCL_Click(object sender, EventArgs e)
{
    pnlModifCL.Visible = true;

    cbLectii.Items.Clear();
    cbCapitole.Items.Clear();

    btnSalveazaModif.Visible = true;

    Intrebari intr = lstIntrebari.SelectedItem as Intrebari;
    if(intr.capitol.ToString()!="General")
    {
        cbCapitole.Text = intr.capitol.ToString();

        //adaug capitolul "General"
        Capitole capitol = new Capitole();
        {
            capitol.titlu = "General";
            capitol.idC = "0";
        }
    }
}

```

```

        cbCapitole.Items.Add("capitol");
    }
else
{
    Capitole cp = new Capitole();
    {
        cp.titlu = "General";
        cp.idC = "0";
        cbCapitole.Items.Add(cp);
    }

    cbCapitole.Text = "General";
}

if(intr.lectie.ToString()!="General")
{
    cbLectii.Text = intr.lectie.ToString();

    //adaug lectia "General"
    Lectii lectie = new Lectii();
    {
        lectie.titlu = "General";
        lectie.idL = "0";
        lectie.idProfesor = "0";
    }

    cbLectii.Items.Add(lectie);
}
else
{
    Lectii lt = new Lectii();
    {
        lt.titlu = "General";
        lt.idL = "0";
        lt.idProfesor = "0";
    }

    cbLectii.Items.Add(lt);
}

cbLectii.Text = "General";
}

MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";

```

```

conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM capitole";

MySqlDataReader r = cmd.ExecuteReader();
while(r.Read())
{
    Capitole c = new Capitole();
    {
        c.idC = r["idC"].ToString();
        c.titlu = r["titlu"].ToString();
    }
    cbCapitole.Items.Add(c);

    if(c.titlu==cbCapitole.Text)
    {
        id = c.idC;
    }
}
conexiune.Close();

if(cbCapitole.Text!="General")
{
    incarcaLectii(id);
}
else
{
    cbLectii.Items.Clear();

    Lectii l = new Lectii();
    {
        l.titlu = "General";
        l.idL = "0";
        l.idProfesor = "0";
    }

    cbLectii.Items.Add(l);

    cbLectii.Text = "General";
}
}

```

Pentru salvarea modificărilor se utilizează următoarele secvențe de cod:

```
//Salvarea modificărilor legate de capitolul și lecția în care se încadrează întrebarea

private void btnSalveazaModif_Click(object sender, EventArgs e)
{
    if (cbLectii.Text != "" && cbCapitole.Text != "")
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "UPDATE intrebari SET capitol=@pcapitol, lectie=@plectie
WHERE idIntrebare=@pidIntrebare ";
        cmd.Parameters.AddWithValue("pcapitol", cbCapitole.Text);
        cmd.Parameters.AddWithValue("plectie", cbLectii.Text);
        cmd.Parameters.AddWithValue("pidIntrebare", rtbIntrebare.Tag.ToString());
        cmd.ExecuteNonQuery();

        MessageBox.Show("Modificările au fost salvate cu succes.");
        pnlModifCL.Visible = false;
    }
    else
    {
        MessageBox.Show("Completați toate campurile înainte de a salva modificările.");
    }
}
```

//Salvarea modificărilor pentru întrebare/răspunsuri

```
private void btnModifIntr_Click(object sender, EventArgs e)
{
    int invalid = 0;
    if(rtbIntrebare.Text!="")
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
```

```

        cmd.Connection = conexiune;
        cmd.CommandText = "UPDATE intrebari SET intrebare=@pintr WHERE
idIntrebare=@pidIntr";
        cmd.Parameters.AddWithValue("pidIntr", rtbIntrebare.Tag.ToString());
        cmd.Parameters.AddWithValue("pintr", rtbIntrebare.Text);
        cmd.ExecuteNonQuery();
        conexiune.Close();
    }
    else
    {
        MessageBox.Show("Trebuie completeate toate campurile inainte de a salva
modificările.");
        invalid = 1;
    }

    if(invalid==0 && pnlTextScurt.Visible==true && txtRaspTextScurt.Text!="")
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "UPDATE raspunsuri SET raspuns=@praspuns WHERE
idIntrebare=@pidIntr";
        cmd.Parameters.AddWithValue("pidIntr", rtbIntrebare.Tag.ToString());
        cmd.Parameters.AddWithValue("praspuns", txtRaspTextScurt.Text);
        cmd.ExecuteNonQuery();
        conexiune.Close();

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

    //se adauga notificare pentru modificarea intrebării
    MySqlConnection c = new MySqlConnection();
    c.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
    c.Open(); //Se deschide conexiunea

    MySqlCommand cmdd = new MySqlCommand();
    cmdd.Connection = c;
    cmdd.CommandText = @"INSERT INTO notificari
(idCont,tipNotificare,idIntrebare) VALUES (@idCont,@ptip,@idIntr)";

    MySqlConnection conn = new MySqlConnection();
    conn.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;

```

```

UID=root; PASSWORD=; Allow User Variables=True";
conn.Open(); //Se deschide conexiunea

MySqlCommand cmm = new MySqlCommand();
cmm.Connection = conn;
cmm.CommandText = "SELECT * FROM conturi";
MySqlDataReader read = cmm.ExecuteReader();
while (read.Read())
{
    cmdd.Parameters.AddWithValue("idCont", read["id"].ToString());
    cmdd.Parameters.AddWithValue("ptip", "modifIntrebare");
    cmdd.Parameters.AddWithValue("idIntr", rtbIntrebare.Tag.ToString());
    cmdd.ExecuteNonQuery();
    cmdd.Parameters.Clear();
}
read.Close();
conn.Close();
}
else
{
    if(invalid==0 && pnlRaspUnic.Visible==true && rtbVar1.Text!="" &&
rtbVar2.Text!="" && rtbVar3.Text!="" && (rbVar1.Checked==true || rbVar2.Checked==true
|| rbVar3.Checked==true))
    {
        MySqlConnection con = new MySqlConnection();
        con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
        con.Open(); //Se deschide conexiunea

        MySqlCommand comm = new MySqlCommand();
        comm.Connection = con;
        comm.CommandText = "SELECT * FROM raspunsuri WHERE
idIntrebare=@pid";
        comm.Parameters.AddWithValue("pid", rtbIntrebare.Tag.ToString());

        //update tabel din baza de date
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;

        MySqlDataReader r = comm.ExecuteReader();
    }
}

```

```

while(r.Read())
{
    Raspunsuri rasp = new Raspunsuri();
    {
        rasp.idRaspuns = r["idRaspuns"].ToString();
        rasp.idIntrebare = r["idIntrebare"].ToString();
        rasp.raspuns = r["raspuns"].ToString();
        rasp.tipRaspuns = r["tipRaspuns"].ToString();
        rasp.corect = r["corect"].ToString();
    }

    if(rasp.idRaspuns==rtbVar1.Tag.ToString())
    {
        cmd.CommandText = "UPDATE raspunsuri SET raspuns=@pr1,
corect=@c1 WHERE idIntrebare=@id1 AND idRaspuns=@idr1";
        cmd.Parameters.AddWithValue("id1", rtbIntrebare.Tag.ToString());
        cmd.Parameters.AddWithValue("pr1", rtbVar1.Text);
        cmd.Parameters.AddWithValue("idr1", rtbVar1.Tag.ToString());
        if (rbVar1.Checked == true)
        {
            cmd.Parameters.AddWithValue("c1", "da");
            //MessageBox.Show("da");
        }
        else
        {
            cmd.Parameters.AddWithValue("c1", "nu");
            //MessageBox.Show("nu");
        }
    }

    //MessageBox.Show(rasp.idRaspuns.ToString() + " " +
rtbVar1.Tag.ToString());
}
else
{
    if(rasp.idRaspuns==rtbVar2.Tag.ToString())
    {
        cmd.CommandText = "UPDATE raspunsuri SET raspuns=@pr2,
corect=@c2 WHERE idIntrebare=@id2 AND idRaspuns=@idr2";
        cmd.Parameters.AddWithValue("id2", rtbIntrebare.Tag.ToString());
        cmd.Parameters.AddWithValue("pr2", rtbVar2.Text);
        cmd.Parameters.AddWithValue("idr2", rtbVar2.Tag.ToString());
        if (rbVar2.Checked == true)
        {
            cmd.Parameters.AddWithValue("c2", "da");

        }
    }

    //MessageBox.Show("da");
}

```

```

    }
    else
    {
        cmd.Parameters.AddWithValue("c2", "nu");

        //MessageBox.Show("nu");
    }

    //MessageBox.Show(rasp.idRaspuns.ToString() + " " +
rtbVar2.Tag.ToString());
}
else
{
    if(rasp.idRaspuns==rtbVar3.Tag.ToString())
    {
        cmd.CommandText = "UPDATE raspunsuri SET raspuns=@pr3,
corect=@c3 WHERE idIntrebare=@id3 AND idRaspuns=@idr3";
        cmd.Parameters.AddWithValue("id3", rtbIntrebare.Tag.ToString());
        cmd.Parameters.AddWithValue("pr3", rtbVar3.Text);
        cmd.Parameters.AddWithValue("idr3", rtbVar3.Tag.ToString());
        if (rbVar3.Checked == true)
        {
            cmd.Parameters.AddWithValue("c3", "da");
            //MessageBox.Show("da");
        }
        else
        {
            cmd.Parameters.AddWithValue("c3", "nu");
            //MessageBox.Show("nu");
        }
    }

    //MessageBox.Show(rasp.idRaspuns.ToString() + " " +
rtbVar3.Tag.ToString());
}
}

cmd.ExecuteNonQuery();
}
conexiune.Close();
con.Close();
MessageBox.Show("Datele au fost modificate cu succes.");

//se adauga notificare pentru modificarea intrebarii
MySqlConnection c = new MySqlConnection();
c.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";

```

```
c.Open(); //Se deschide conexiunea

MySqlCommand cmdd = new MySqlCommand();
cmdd.Connection = c;
cmdd.CommandText = @"INSERT INTO notificari
(idCont,tipNotificare,idIntrebare) VALUES (@idCont,@ptip,@idIntr)";

MySqlConnection conn = new MySqlConnection();
conn.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conn.Open(); //Se deschide conexiunea

MySqlCommand cmm = new MySqlCommand();
cmm.Connection = conn;
cmm.CommandText = "SELECT * FROM conturi";
MySqlDataReader read = cmm.ExecuteReader();
while (read.Read())
{
    cmdd.Parameters.AddWithValue("idCont", read["id"].ToString());
    cmdd.Parameters.AddWithValue("ptip", "modifIntrebare");
    cmdd.Parameters.AddWithValue("idIntr", rtbIntrebare.Tag.ToString());
    cmdd.ExecuteNonQuery();
    cmdd.Parameters.Clear();
}
read.Close();
conn.Close();
}

else
{
    if(invalid==0 && pnlRaspMultiplu.Visible==true && rtbMV1.Text!="" &&
rtbMV2.Text!="" && rtbMV3.Text!="" && (cbV1.Checked==true || cbV2.Checked==true ||
cbV3.Checked==true))
    {
        MySqlConnection con = new MySqlConnection();
        con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
        con.Open(); //Se deschide conexiunea

        MySqlCommand comm = new MySqlCommand();
        comm.Connection = con;
        comm.CommandText = "SELECT * FROM raspunsuri WHERE
idIntrebare=@pid";
        comm.Parameters.AddWithValue("pid", rtbIntrebare.Tag.ToString());

        //update tabel din baza de date
        MySqlConnection conexiune = new MySqlConnection();
    }
}
```

```

conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;

MySqlDataReader r = comm.ExecuteReader();
while (r.Read())
{
    Raspunsuri rasp = new Raspunsuri();
    {
        rasp.idRaspuns = r["idRaspuns"].ToString();
        rasp.idIntrebare = r["idIntrebare"].ToString();
        rasp.raspuns = r["raspuns"].ToString();
        rasp.tipRaspuns = r["tipRaspuns"].ToString();
        rasp.corect = r["corect"].ToString();
    }

    if(rasp.idRaspuns==rtbMV1.Tag.ToString())
    {
        cmd.CommandText = "UPDATE raspunsuri SET raspuns=@rm1,
corect=@cm1 WHERE idIntrebare=@idm1 AND idRaspuns=@idrm1";
        cmd.Parameters.AddWithValue("rm1", rtbMV1.Text);
        if(cbV1.Checked==true)
        {
            cmd.Parameters.AddWithValue("cm1", "da");
        }
        else
        {
            cmd.Parameters.AddWithValue("cm1", "nu");
        }
        cmd.Parameters.AddWithValue("idm1", rtbIntrebare.Tag.ToString());
        cmd.Parameters.AddWithValue("idrm1", rtbMV1.Tag.ToString());
    }
    else
    {
        if (rasp.idRaspuns == rtbMV2.Tag.ToString())
        {
            cmd.CommandText = "UPDATE raspunsuri SET raspuns=@rm2,
corect=@cm2 WHERE idIntrebare=@idm2 AND idRaspuns=@idrm2";
            cmd.Parameters.AddWithValue("rm2", rtbMV2.Text);
            if (cbV2.Checked == true)
            {
                cmd.Parameters.AddWithValue("cm2", "da");
            }
        }
    }
}

```

```

        }
        else
        {
            cmd.Parameters.AddWithValue("cm2", "nu");
        }
        cmd.Parameters.AddWithValue("idm2", rtbIntrebare.Tag.ToString());
        cmd.Parameters.AddWithValue("idrm2", rtbMV2.Tag.ToString());

    }
    else
    {
        if (rasp.idRaspuns == rtbMV3.Tag.ToString())
        {
            cmd.CommandText = "UPDATE raspunsuri SET raspuns=@rm3,
corect=@cm3 WHERE idIntrebare=@idm3 AND idRaspuns=@idrm3";
            cmd.Parameters.AddWithValue("rm3", rtbMV3.Text);
            if (cbV3.Checked == true)
            {
                cmd.Parameters.AddWithValue("cm3", "da");
            }
            else
            {
                cmd.Parameters.AddWithValue("cm3", "nu");
            }
            cmd.Parameters.AddWithValue("idm3", rtbIntrebare.Tag.ToString());
            cmd.Parameters.AddWithValue("idrm3", rtbMV3.Tag.ToString());
        }
    }
}
cmd.ExecuteNonQuery();
}
conexiune.Close();
con.Close();
MessageBox.Show("Datele au fost modificate cu succes.");

//se adauga notificare pentru modificarea intrebarii
MySqlConnection c = new MySqlConnection();
c.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
c.Open(); //Se deschide conexiunea

MySqlCommand cmdd = new MySqlCommand();
cmdd.Connection = c;
cmdd.CommandText = @"INSERT INTO notificari
(idCont,tipNotificare,idIntrebare) VALUES (@idCont,@ptip,@idIntr)";

```


3.13. Fereastra FrmStergeIntrebari

Prin intermediul acestei ferestre profesorii pot șterge întrebări din baza de date.

Fereastra conține:

- Un *Label*-uri cu textul: “Intrebări:”.
- Patru *Panel*-uri:

→ pnlIntrebare care conține un *Label* și un *RichTextBox* în care să se încarcă întrebarea (rtbIntrebare).

→ pnlTextScurt care conține două *Label*-uri și un *TextBox* în care să se încarcă răspunsul corect (txtRaspTextScurt).

→ pnlRaspUnic care conține cinci *Label*-uri, trei *RichTextBox*-uri, câte unul pentru fiecare variantă de răspuns (rtbVar1, rtbVar2, rtbVar3) și trei *RadioButton*-uri (rbVar1, rbVar2, rbVar3).

→ pnlRaspMultiplu care conține cinci *Label*-uri, trei *RichTextBox*-uri, câte unul pentru fiecare variantă de răspuns (rtbMV1, rtbMV2, rtbMV3) și trei *CheckBox*-uri (cbV1, cbV2, cbV3).

- Un *Button* cu ajutorul căruia se șterge întrebarea din baza de date (btnStergeIntr).
- Un *ListBox* în care se încarcă toate întrebările existente în baza de date (lstIntrebari).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState

Istoria sclaviei

Sterge Intrebări

Intrebări:

Scięti termenul la care se refera definiția:
Pe ce fel de plantații lucrau sclavi?
Ce doreau să obțină marii stăpâni de sclavi?
Între ce ani a avut loc Revoluția Americană?
În ce an a fost adoptat al 13-lea Amendment?
Ce a dus la mișcarea drepturilor civile?
Cine a susținut emanciparea?
Prin adoptarea celui de al 13-lea Amendment
Ce a reprezentat "Mișcarea pentru drepturi civile"?
Când au fost stabilite legile "Jim Crow"?

Intrebare:

Scięti termenul la care se refera definiția: "desemnează condiția umană a persoanelor care lucrează pentru un stăpân fără remunerare și ce nu dispun de drepturi asupra propriei persoane".

Raspuns:

sclavia

Sterge intrebare

Sterge Intrebări

Intrebări:

Scięti termenul la care se refera definiția:
Pe ce fel de plantații lucrau sclavi?
Ce doreau să obțină marii stăpâni de sclavi?
Între ce ani a avut loc Revoluția Americană?
În ce an a fost adoptat al 13-lea Amendment?
Ce a dus la mișcarea drepturilor civile?
Cine a susținut emanciparea?
Prin adoptarea celui de al 13-lea Amendment
Ce a reprezentat "Mișcarea pentru drepturi civile"?
Când au fost stabilite legile "Jim Crow"?

Intrebare:

Ce a dus la mișcarea drepturilor civile?

Variante de raspuns:

varianta 1
rasismul

varianta 2
egalitatea dintre indivizi

varianta 3
inegalitatea politică și economică

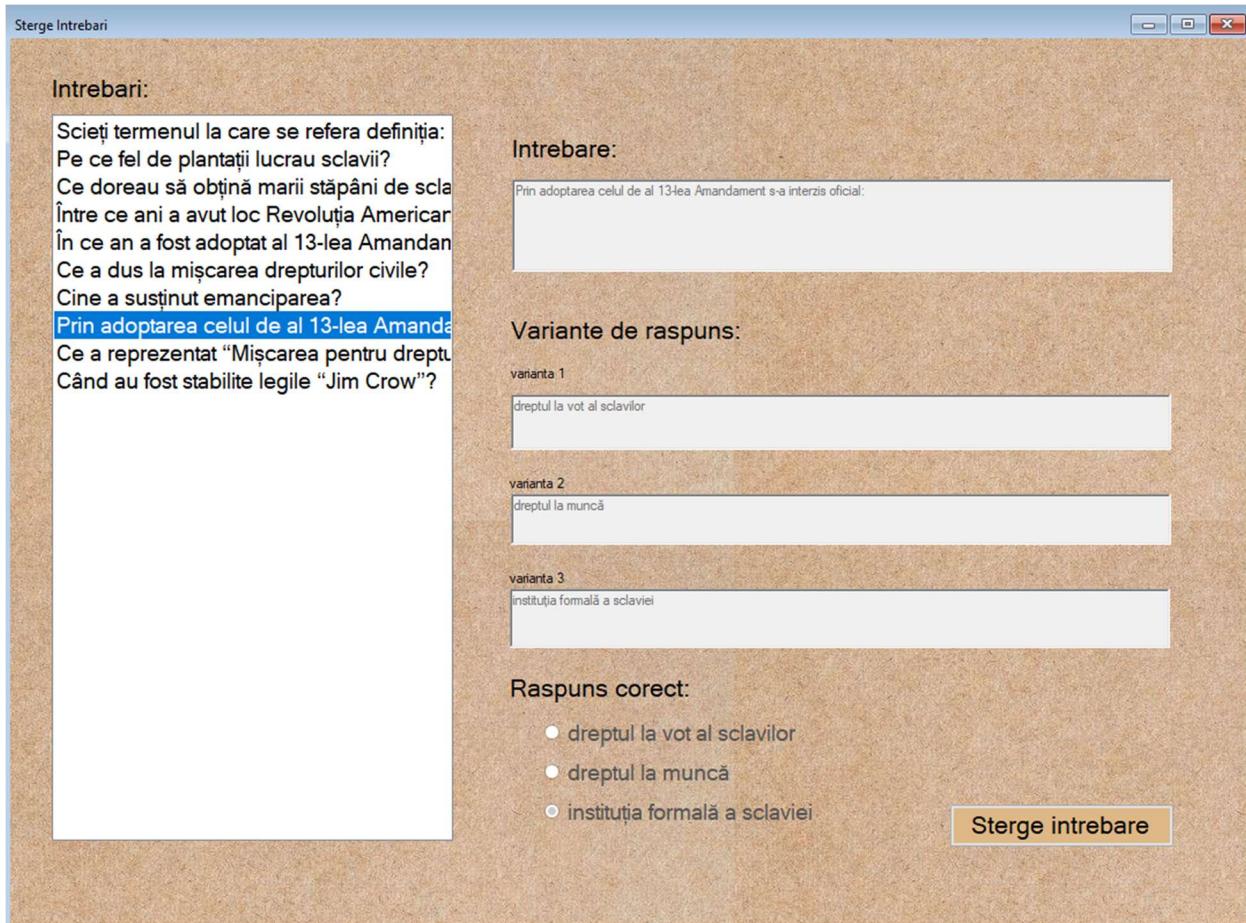
Raspunsuri corecte:

racismul

egalitatea dintre indivizi

inegalitatea politică și economică

Sterge intrebare



Pentru încărcarea întrebărilor din baza de date și ștergerea acestora se utilizează următoarele secvențe de cod:

```
//Încărcarea întrebărilor în ListBox

//Se creeaza conexiunea la baza de date
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM intrebari";

MySqlDataReader r = cmd.ExecuteReader();
```

```

int ok = 0;

while (r.Read())
{
    Intrebari intr = new Intrebari();
    {
        intr.idIntrebare = r["idIntrebare"].ToString();
        intr.idProfesor = r["idProfesor"].ToString();
        intr.intrebare = r["intrebare"].ToString();
        intr.capitol = r["capitol"].ToString();
        intr.lectie = r["lectie"].ToString();
    }

    if (intr.idProfesor == this.Tag.ToString())
    {
        lstIntrebari.Items.Add(intr);
        ok++;
    }
}
r.Close();

if (ok == 0)
{
    MessageBox.Show("Nu aveti incarcate intrebari pe care sa le puteti sterge.");
    this.Close();
}
conexiune.Close();

```

//Încărcarea întrebării selectate și răspunsul/răspunsurile

```

private void lstIntrebari_SelectedIndexChanged(object sender, EventArgs e)
{
    if (lstIntrebari.SelectedIndex != -1)
    {
        btnStergIntr.Visible = true;

        Intrebari intr = lstIntrebari.SelectedItem as Intrebari;

        pnlIntrebare.Visible = true;
        rtbIntrebare.Tag = intr.idIntrebare.ToString();
        //MessageBox.Show(rtbIntrebare.Tag.ToString());
        rtbIntrebare.Text = intr.intrebare.ToString();

        //se incarca raspunsurile
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;

```

```

DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM raspunsuri WHERE
idIntrebare=@pidIntrebare";
cmd.Parameters.AddWithValue("pidIntrebare", rtbIntrebare.Tag.ToString());

int ok = 1;

MySqlDataReader r = cmd.ExecuteReader();
while (r.Read())
{
    Raspunsuri rasp = new Raspunsuri();
    {
        rasp.idRaspuns = r["idRaspuns"].ToString();
        rasp.idIntrebare = r["idIntrebare"].ToString();
        rasp.raspuns = r["raspuns"].ToString();
        rasp.tipRaspuns = r["tipRaspuns"].ToString();
        rasp.corect = r["corect"].ToString();
    }

    if (rasp.tipRaspuns == "text scurt")
    {
        pnlTextScurt.Visible = true;
        pnlRaspUnic.Visible = pnlRaspMultiplu.Visible = false;

        txtRaspTextScurt.Text = rasp.raspuns.ToString();

    }
    else
    {
        if (rasp.tipRaspuns == "unic")
        {
            pnlRaspUnic.Visible = true;
            pnlTextScurt.Visible = pnlRaspMultiplu.Visible = false;

            if (ok == 1)
            {
                rtbVar1.Text = rasp.raspuns.ToString();
                rtbVar1.Tag = rasp.idRaspuns.ToString();
                rbVar1.Text = rtbVar1.Text.ToString();
                if (rasp.corect == "da")
                {
                    rbVar1.Checked = true;
                }
            }
        }
    }
}

```

```

        }
    else
    {
        rbVar1.Checked = false;
    }
}
else
{
    if (ok == 2)
    {
        rtbVar2.Text = rasp.raspuns.ToString();
        rtbVar2.Tag = rasp.idRaspuns.ToString();
        rbVar2.Text = rtbVar2.Text.ToString();
        if (rasp.corect == "da")
        {
            rbVar2.Checked = true;
        }
        else
        {
            rbVar2.Checked = false;
        }
    }
    else
    {
        if (ok == 3)
        {
            rtbVar3.Text = rasp.raspuns.ToString();
            rtbVar3.Tag = rasp.idRaspuns.ToString();
            rbVar3.Text = rtbVar3.Text.ToString();
            if (rasp.corect == "da")
            {
                rbVar3.Checked = true;
            }
            else
            {
                rbVar3.Checked = false;
            }
        }
    }
    ok++;
}
else
{
    pnlRaspMultiplu.Visible = true;
    pnlTextScurt.Visible = pnlRaspUnic.Visible = false;
}

```

```
if (ok == 1)
{
    rtbMV1.Text = rasp.raspuns.ToString();
    rtbMV1.Tag = rasp.idRaspuns.ToString();
    cbV1.Text = rtbMV1.Text;
    if (rasp.corect == "da")
    {
        cbV1.Checked = true;
    }
    else
    {
        cbV1.Checked = false;
    }
}
else
{
    if (ok == 2)
    {
        rtbMV2.Text = rasp.raspuns.ToString();
        rtbMV2.Tag = rasp.idRaspuns.ToString();
        cbV2.Text = rtbMV2.Text;
        if (rasp.corect == "da")
        {
            cbV2.Checked = true;
        }
        else
        {
            cbV2.Checked = false;
        }
    }
    else
    {
        if (ok == 3)
        {
            rtbMV3.Text = rasp.raspuns.ToString();
            rtbMV3.Tag = rasp.idRaspuns.ToString();
            cbV3.Text = rtbMV3.Text;
            if (rasp.corect == "da")
            {
                cbV3.Checked = true;
            }
            else
            {
                cbV3.Checked = false;
            }
        }
    }
}
```

```

        }
    }

        ok++;
    }

}
conexiune.Close();
}

}

```

//Ștergerea întrebării din baza de date

```

private void btnStergIntr_Click(object sender, EventArgs e)
{
    if (rtbIntrebare.Text != "" && lstIntrebari.SelectedIndex != -1)
    {
        //MessageBox.Show(rtbIntrebare.Tag.ToString());

        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "DELETE FROM intrebari WHERE idIntrebare=@pidIntr";

        cmd.Parameters.AddWithValue("pidIntr", rtbIntrebare.Tag.ToString());
        cmd.ExecuteNonQuery();

        cmd.CommandText = "DELETE FROM raspunsuri WHERE
idIntrebare=@ppidIntr";

        cmd.Parameters.AddWithValue("ppidIntr", rtbIntrebare.Tag.ToString());
        cmd.ExecuteNonQuery();

        cmd.CommandText = "DELETE FROM teste WHERE idIntrebare=@idIntr";

        cmd.Parameters.AddWithValue("idIntr", rtbIntrebare.Tag.ToString());
        cmd.ExecuteNonQuery();

        MessageBox.Show("Intrebarea a fost stearsa cu succes!");
        pnlIntrebare.Visible = pnlRaspMultiplu.Visible = pnlRaspUnic.Visible =
pnlTextScurt.Visible = false;
    }
}

```

```
    btnStergIntr.Visible = false;  
  
    FrmStergeIntrebari_Load(sender, e);  
}  
}
```

3.14. Fereastra FrmAdaugaTest

Prin intermediul acestei ferestre profesorii pot crea teste noi.

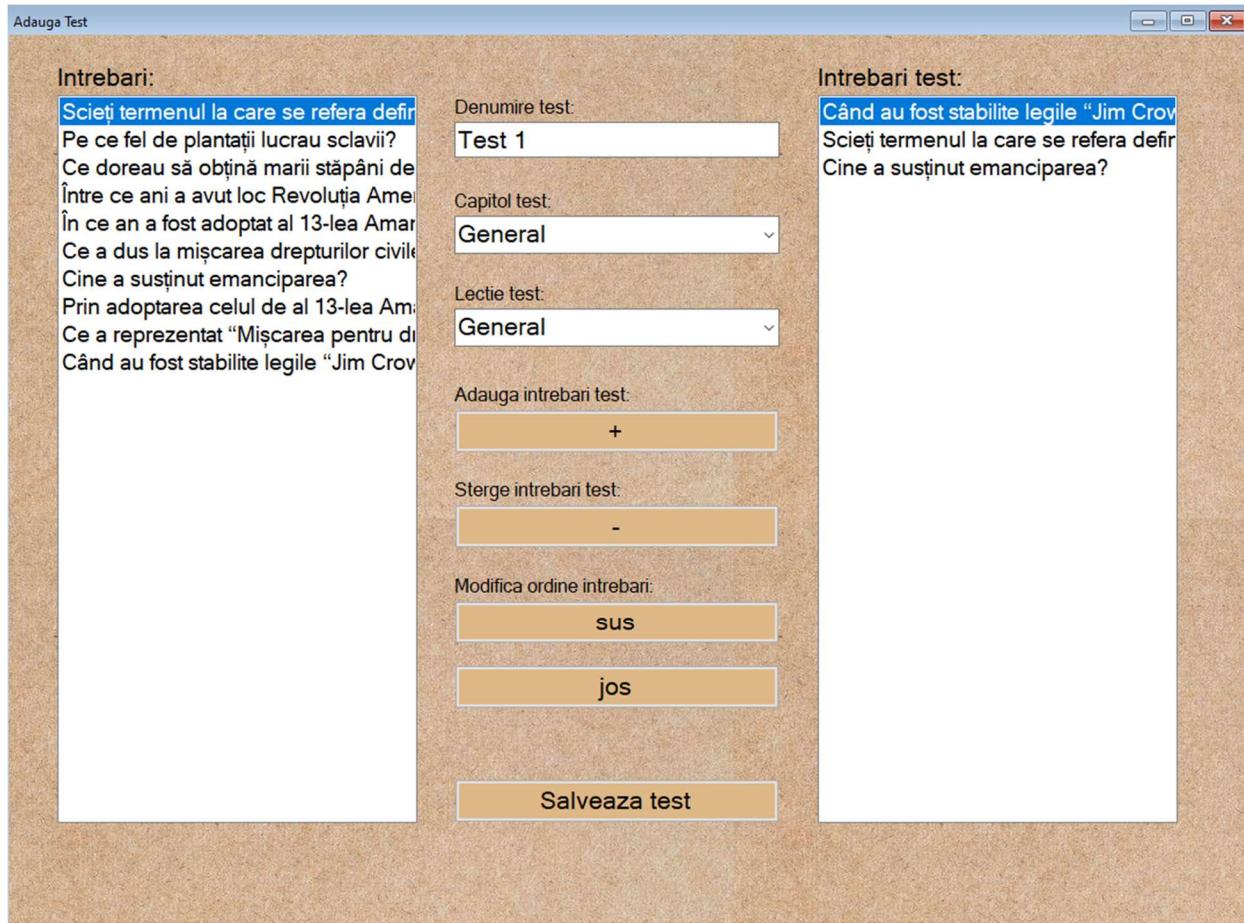
Fereastra conține:

- Două *ListBox*-uri: lstIntrebari în care se încarcă toate întrebările și lstIntrTest în care profesorul adaugă întrebările care vor face parte din test.
- Opt *Label*-uri în care apar indicații de utilizare a ferestrei.
- Un *TextBox* în care se adaugă denumirea testului (txtNumeTest).
- Două *ComboBox*-uri în care se încarcă capitolele (cbCapitole) și lecțiile (cbLectii).
- Cinci *Button*-uri: btnAdaugaIntrebare și btnStergeIntrebare prin intermediul cărora se adaugă sau se sterge o întrebare din lstIntrTest, btnSus și btnJos prin intermediul cărora se modifică ordinea în care vor apărea întrebările la test și btnSalveaza prin intermediul căruia se salvează testul.

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState

Istoria sclaviei



Pentru operațiile cu întrebări din lstIntrTest se utilizează următoarele secvențe de cod:

```
//Butonul “+”

private void btnAdaugaIntrebare_Click(object sender, EventArgs e)
{
    if(lstIntrebări.SelectedIndex!=-1)
    {
        Intrebări intr = lstIntrebări.SelectedItem as Intrebări;

        int ok = 0;
        for(int i=0;i<lstIntrTest.Items.Count;i++)
        {
            if (lstIntrTest.Items[i] == intr)
            {
                ok = 1;
                break;
            }
        }
    }
}
```

```

        }
    }
    if(ok==0)
    {
        lstIntrTest.Items.Add(intr);
    }
    else
    {
        MessageBox.Show("Intrebarea selectata a fost deja adaugata la test");
    }

}
else
{
    MessageBox.Show("Selectati intrebare pe care doriti sa o adaugati la test.");
}

}

```

//Butonul “-”

```

private void btnStergeIntrebare_Click(object sender, EventArgs e)
{
    if(lstIntrTest.SelectedIndex!=-1)
    {
        lstIntrTest.Items.Remove(lstIntrTest.SelectedItem);
    }
    else
    {
        MessageBox.Show("Selectati intrebare pe care doriti sa o stergeti.");
    }
}

```

//Butonul “sus”

```

private void btnSus_Click(object sender, EventArgs e)
{
    if(lstIntrTest.SelectedIndex > 0)
    {
        int index = lstIntrTest.SelectedIndex;
        Intrebari intr = lstIntrTest.SelectedItem as Intrebari;

        lstIntrTest.Items[index] = lstIntrTest.Items[index - 1];
        lstIntrTest.Items[index - 1] = intr;
    }
}

```

```

        lstIntrTest.SelectedIndex = index - 1;
        lstIntrTest.Update();
        lstIntrTest.Focus();
    }
    else
    {
        if (lstIntrTest.SelectedIndex == -1)
        {
            MessageBox.Show("Selectati intrebare pe care doriti sa o mutati cu o pozitie mai
jos.");
        }
        else
        {
            MessageBox.Show("Nu puteti efectua aceasta modificare.");
        }
    }
}

//Butonul "jos"

private void btnJos_Click(object sender, EventArgs e)
{
    if (lstIntrTest.SelectedIndex < lstIntrTest.Items.Count - 1 &&
lstIntrTest.SelectedIndex != -1)
    {
        int index = lstIntrTest.SelectedIndex;
        Intrebari intr = lstIntrTest.SelectedItem as Intrebari;

        lstIntrTest.Items[index] = lstIntrTest.Items[index + 1];
        lstIntrTest.Items[index + 1] = intr;

        lstIntrTest.SelectedIndex = index + 1;
        lstIntrTest.Update();
        lstIntrTest.Focus();
    }
    else
    {
        if (lstIntrTest.SelectedIndex == -1)
        {
            MessageBox.Show("Selectati intrebare pe care doriti sa o mutati cu o pozitie mai
jos.");
        }
        else
        {
            MessageBox.Show("Nu puteti efectua aceasta modificare.");
        }
    }
}

```

```

        }
    }
}
```

Pentru salvarea testului se utilizează următoarea secvență de cod:

```

string idTest;

private void btnSalveaza_Click(object sender, EventArgs e)
{
    if(lstIntrTest.Items.Count!=0 && txtNumeTest.Text!="")
    {
        int ok = 0;
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT denumire FROM teste";
        MySqlDataReader r = cmd.ExecuteReader();
        while(r.Read())
        {
            if (r["denumire"].ToString()==txtNumeTest.Text)
            {
                ok = 1;
                break;
            }
        }
        r.Close();

        if(ok==0)
        {
            cmd.CommandText = "INSERT INTO teste (denumire, idProfesor,
data_adaugare, capitol, lectie) VALUES (@pdenum, @pidProf, @pdataAd, @pCapitol,
@plectie)";

            cmd.Parameters.AddWithValue("pdenum", txtNumeTest.Text);
            cmd.Parameters.AddWithValue("pidProf", this.Tag.ToString());
            cmd.Parameters.AddWithValue("pdataAd", "now()");
        }
    }
}
```

```

if (cbCapitole.Text != "")
{
    cmd.Parameters.AddWithValue("pCapitol", cbCapitole.Text);
}
else
{
    cmd.Parameters.AddWithValue("pCapitol", "General");
}

if (cbLectii.Text != "")
{
    cmd.Parameters.AddWithValue("plectie", cbLectii.Text);
}
else
{
    cmd.Parameters.AddWithValue("plectie", "General");
}
cmd.ExecuteNonQuery();

cmd.CommandText = "SELECT * FROM teste WHERE
denumire=@pdenumire";
cmd.Parameters.AddWithValue("pdenumire", txtNumeTest.Text);
r = cmd.ExecuteReader();

while (r.Read())
{
    idTest = r["idTest"].ToString();
}
r.Close();

}
conexiune.Close();

if (ok==0)
{

for (int i=0;i<lstIntrTest.Items.Count;i++)
{
    MySqlConnection con = new MySqlConnection();
    con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    con.Open(); //Se deschide conexiunea

    MySqlCommand cm = new MySqlCommand();
    cm.Connection = con;
}

```

```

        cm.CommandText = "INSERT INTO continut_teste (idTest, idIntrebare)
VALUES (@pidTest, @pidIntrebare)";
        cm.Parameters.AddWithValue("pidTest", idTest.ToString());
        Intrebari intr = lstIntrTest.Items[i] as Intrebari;
        cm.Parameters.AddWithValue("pidIntrebare", intr.idIntrebare.ToString());
        cm.ExecuteNonQuery();
        con.Close();
    }

    MessageBox.Show("Testul a fost salvat cu succes.");

    //se adauga notificare pentru adaugarea testului
    MySqlConnection c = new MySqlConnection();
    c.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    c.Open(); //Se deschide conexiunea

    MySqlCommand cmdd = new MySqlCommand();
    cmdd.Connection = c;
    cmdd.CommandText = @"INSERT INTO notificari (idCont,tipNotificare,idTest)
VALUES (@idCont,@ptip,@idT)";

    MySqlConnection conn = new MySqlConnection();
    conn.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conn.Open(); //Se deschide conexiunea

    MySqlCommand cmm = new MySqlCommand();
    cmm.Connection = conn;
    cmm.CommandText = "SELECT * FROM conturi";
    MySqlDataReader read = cmm.ExecuteReader();
    while (read.Read())
    {
        cmdd.Parameters.AddWithValue("idCont", read["id"].ToString());
        cmdd.Parameters.AddWithValue("ptip", "adaugareTest");
        cmdd.Parameters.AddWithValue("idT", idTest.ToString());
        cmdd.ExecuteNonQuery();
        cmdd.Parameters.Clear();
    }
    read.Close();
    conn.Close();
    c.Close();

    FrmAdaugaTest_Load(sender, e);
}
else

```

```
{  
    MessageBox.Show("Exista deja un test cu acest nume. Va rugam sa alegeti alt  
nume.");  
}  
conexiune.Close();  
  
}  
else  
{  
    MessageBox.Show("Campul cu denumirea testului si lista intrebarilor pentru test  
sunt obligatorii. Va rugam sa le completati inainte de a salva testul.");  
}  
}
```

3.15. Fereastra FrmModificaTest

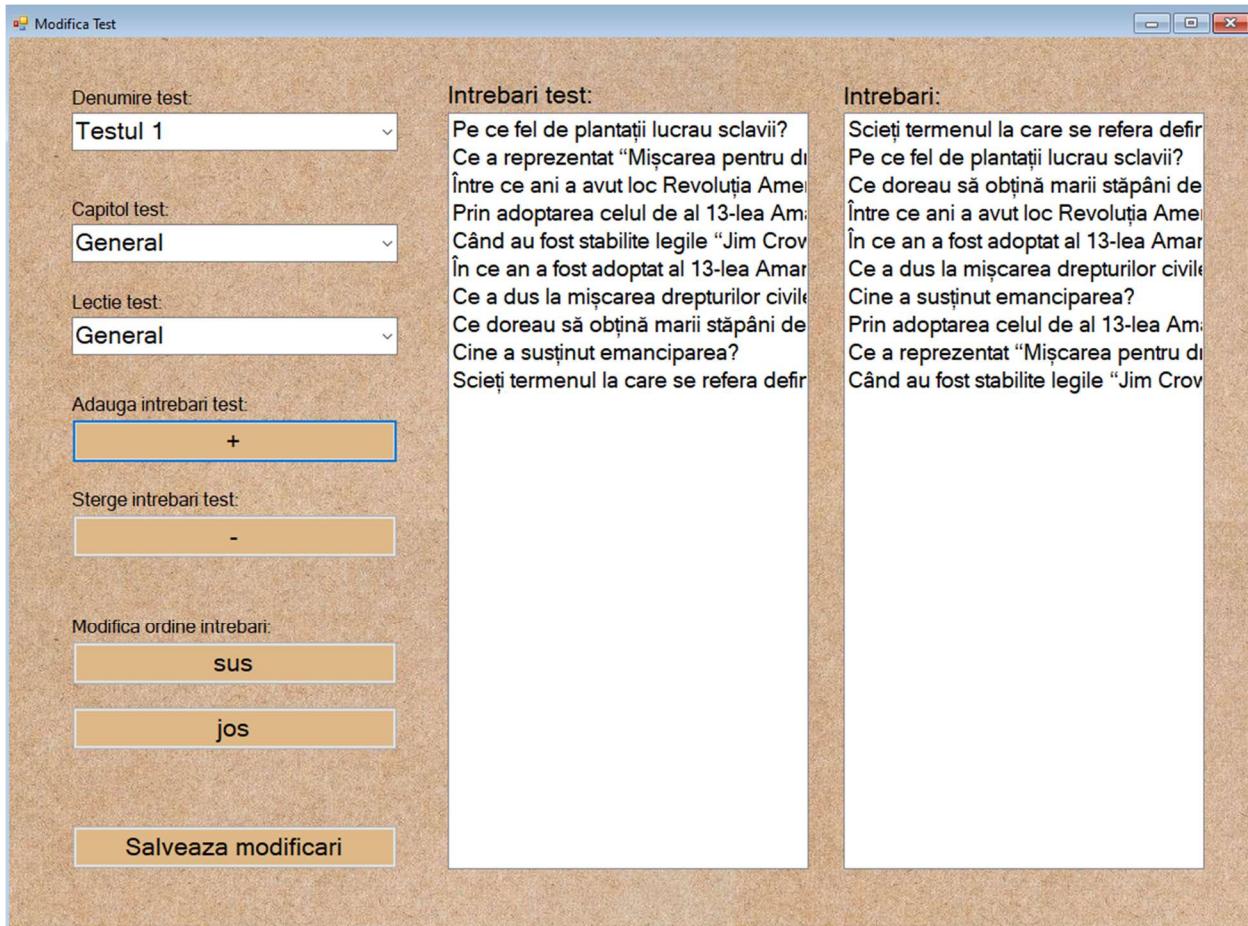
Prin intermediul acestei ferestre profesorii pot modifica testelete existente.

Fereastra conține:

- Trei *ListBox*-uri: lstIntrebari în care se încarcă toate întrebările, lstIntrTest în care profesorul adaugă întrebările care vor face parte din test și lstID care nu este vizibil și în care se încarcă id-ul din baza de date al întrebărilor.
- Trei *Label*-uri în care apar indicații de utilizare a ferestrei.
- Un *ComboBox* în care se afișează denumirile testelor (cbTeste).
- Un *Panel* care conține: două *ComboBox*-uri în care se încarcă capitolele (cbCapitole) și lecțiile (cbLectii), cinci *Button*-uri (btnAdaugaIntrebare și btnStergeIntrebare prin intermediul cărora se adaugă sau se sterge o întrebare din lstIntrTest, btnSus și btnJos prin intermediul cărora se modifică ordinea în care vor apărea întrebările la test și btnSalveaza prin intermediul căruia se salvează modificările) și cinci *Label*-uri.

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



Pentru încărcarea testelor, a tuturor întrebărilor și a întrebărilor care aparțin testului selectat, cât și pentru salvarea modificărilor se utilizează următoarele secvențe de cod:

```
//Încărcarea tuturor întrebărilor și a testelor
```

```
//se adauga intrebarile in ListBox
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM intrebari";

MySqlDataReader r = cmd.ExecuteReader();
while (r.Read())
```

```

{
    Intrebari intr = new Intrebari();
    {
        intr.idIntrebare = r["idIntrebare"].ToString();
        intr.idProfesor = r["idProfesor"].ToString();
        intr.intrebare = r["intrebare"].ToString();
        intr.capitol = r["capitol"].ToString();
        intr.lectie = r["lectie"].ToString();
    }
    lstIntrebari.Items.Add(intr);
}
r.Close();

cmd.CommandText = "SELECT * FROM teste";
int ok = 0;

r = cmd.ExecuteReader();
while(r.Read())
{
    Teste t = new Teste();
    {
        t.idTest = r["idTest"].ToString();
        t.denumire = r["denumire"].ToString();
        t.idProfesor = r["idProfesor"].ToString();
        //t.data_adaugare = r["data_adaugare"].ToString();
        t.capitol = r["capitol"].ToString();
        t.lectie = r["lectie"].ToString();
    }

    if(t.idProfesor==this.Tag.ToString())
    {
        ok++;
        cbTeste.Items.Add(t);
    }
}

if(ok==0)
{
    MessageBox.Show("Nu aveti teste pe care sa le puteti modifica.");
    this.Close();
}

conexiune.Close();

```

//Încărcarea întrebărilor care aparțin testului selectat

```

private void cbTeste_SelectedIndexChanged(object sender, EventArgs e)
{
    if(cbTeste.SelectedIndex!=-1)
    {
        pnlControl.Visible = true;

        lstIntrTest.Items.Clear();

        Teste t = cbTeste.SelectedItem as Teste;

        //se incarca testul ales
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT * FROM teste WHERE idTest=@pidTest";
        cmd.Parameters.AddWithValue("pidTest", t.idTest.ToString());

        MySqlDataReader r = cmd.ExecuteReader();
        while(r.Read())
        {
            Teste test = new Teste();
            {
                test.idTest = r["idTest"].ToString();
                test.denumire = r["denumire"].ToString();
                test.idProfesor = r["idProfesor"].ToString();
                //test.data_adaugare = r["data_adaugare"].ToString();
                test.capitol = r["capitol"].ToString();
                test.lectie = r["lectie"].ToString();
            }
            denumCap = test.capitol;
            idTest = test.idTest;

            cbCapitole.Text = test.capitol;
            cbLectii.Text = test.lectie;

        }
        r.Close();

        //se adauga capitolele in ComboBox
        cbCapitole.Items.Clear();

        cmd.CommandText = "SELECT * FROM capitole";
    }
}

```

```

r = cmd.ExecuteReader();
while (r.Read())
{
    Capitole c = new Capitole();
    {
        c.idC = r["idC"].ToString();
        c.titlu = r["titlu"].ToString();
    }
    cbCapitole.Items.Add(c);

    if(c.titlu==denumCap)
    {
        idCapitol = c.idC;
    }
}
r.Close();

//se incarca intrebarile
cmd.CommandText = "SELECT intrebari.*, continut_teste.id FROM intrebari JOIN
continut_teste ON intrebari.idIntrebare=continut_teste.idIntrebare WHERE idTest=@pidT";
cmd.Parameters.AddWithValue("pidT",idTest);

r = cmd.ExecuteReader();
while(r.Read())
{
    Intrebari intr = new Intrebari();
    {
        intr.idIntrebare = r["idIntrebare"].ToString();
        intr.idProfesor = r["idProfesor"].ToString();
        intr.intrebare = r["intrebare"].ToString();
        intr.capitol = r["capitol"].ToString();
        intr.lectie = r["lectie"].ToString();
    }

    lstIntrTest.Items.Add(intr);
    lstID.Items.Add(r["id"].ToString());
}

conexiune.Close();

//adaug capitolul "General"
Capitole capitol = new Capitole();
{
    capitol.titlu = "General";
}

```

```

        capitol.idC = "0";
    }
    cbCapitole.Items.Add(capitol);

    //se adauga lectiile
    //Capitole cp = cbCapitole. as Capitole;
    incarcaLectii(idCapitol);

}
else
{
    MessageBox.Show("Selectati testul pe care doriti sa-l modificati");
}
}

```

//Salvarea modificărilor

```

private void btnSalveazaModif_Click(object sender, EventArgs e)
{
    if(cbTeste.Text!="" && cbLectii.Text!="" && cbCapitole.Text!="" &&
    lstIntrTest.Items.Count!=0)
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "UPDATE teste SET denumire=@pdent, capitol=@pcapitol,
lectie=@plectie WHERE idTest=@pid";
        cmd.Parameters.AddWithValue("pdent", cbTeste.Text);
        cmd.Parameters.AddWithValue("pcapitol", cbCapitole.Text);
        cmd.Parameters.AddWithValue("plectie", cbLectii.Text);
        cmd.Parameters.AddWithValue("pid", idTest);
        cmd.ExecuteNonQuery();

        cmd.CommandText = "SELECT * FROM continut_teste WHERE
idTest=@pidTest";
        cmd.Parameters.AddWithValue("pidTest", idTest);

        MySqlDataReader r = cmd.ExecuteReader();
        int i = 0;

        while(r.Read())
        {

```

```

MySqlConnection con = new MySqlConnection();
con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
con.Open(); //Se deschide conexiunea

MySqlCommand cm = new MySqlCommand();
cm.Connection = con;

if(i<lstIntrTest.Items.Count && i < lstID.Items.Count)
{
    cm.CommandText = "UPDATE continut_teste SET idIntrebare=@pidIntr
WHERE idTest=@pidTest AND id=@pidd";
    Intrebari intr = lstIntrTest.Items[i] as Intrebari;
    //MessageBox.Show(intr.idIntrebare.ToString() + " " + i.ToString());
    cm.Parameters.AddWithValue("pidIntr", intr.idIntrebare);
    cm.Parameters.AddWithValue("pidTest", idTest);
    cm.Parameters.AddWithValue("pidd", lstID.Items[i]);
    poz = i;
    idInreg = r["id"].ToString();
    cm.ExecuteNonQuery();
}
else
{
    //Intrebari intrebare = lstIntrTest.Items[i] as Intrebari;
    cm.CommandText = "DELETE FROM continut_teste WHERE id>@id AND
idTest=@idT";
    cm.Parameters.AddWithValue("id", idInreg.ToString());
    cm.Parameters.AddWithValue("idT", idTest);
    cm.ExecuteNonQuery();
}

con.Close();
i++;
}
r.Close();
conexiune.Close();

//verific daca au ramas elem

if (poz<lstIntrTest.Items.Count)
{
    for (int k = poz + 1; k < lstIntrTest.Items.Count; k++)
    {
        MySqlConnection conex = new MySqlConnection();
        conex.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";

```

```

conex.Open(); //Se deschide conexiunea

MySqlCommand cmm = new MySqlCommand();
cmm.Connection = conex;
cmm.CommandText = "INSERT INTO continut_teste (idTest,idIntrebare)
VALUES (@pidTest, @pidIntr)";
cmm.Parameters.AddWithValue("pidTest", idTest);
Intrebari intr = lstIntrTest.Items[k] as Intrebari;
//MessageBox.Show(intr.ToString() + " " + k.ToString());
cmm.Parameters.AddWithValue("pidIntr", intr.idIntrebare.ToString());
cmm.ExecuteNonQuery();
conex.Close();
}

}

if(stop==0)
{
    stop = 1;
    btnSalveazaModif_Click(sender, e);
}
else
{
    MessageBox.Show("Modificările au fost salvate.");
    //se adauga notificare pentru modificarea testului
    MySqlConnection c = new MySqlConnection();
    c.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    c.Open(); //Se deschide conexiunea

    MySqlCommand cmdd = new MySqlCommand();
    cmdd.Connection = c;
    cmdd.CommandText = @"INSERT INTO notificari (idCont,tipNotificare,idTest)
VALUES (@idCont,@ptip,@idT)";

    MySqlConnection conn = new MySqlConnection();
    conn.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conn.Open(); //Se deschide conexiunea

    MySqlCommand cmm = new MySqlCommand();
    cmm.Connection = conn;
    cmm.CommandText = "SELECT * FROM conturi";
    MySqlDataReader read = cmm.ExecuteReader();
    while (read.Read())
    {
        cmdd.Parameters.AddWithValue("idCont", read["id"].ToString());
}

```

```
        cmdd.Parameters.AddWithValue("ptip", "modifTest");
        cmdd.Parameters.AddWithValue("idT", idTest);
        cmdd.ExecuteNonQuery();
        cmdd.Parameters.Clear();
    }
    read.Close();
    conn.Close();
    stop = 0;
    FrmModificaTest_Load(sender, e);
}

}
else
{
    MessageBox.Show("Toate campurile trebuie completate inainte de a salva
modificările.");
    this.Close();
}
}
```

3.16. Fereastra FrmSergeTest

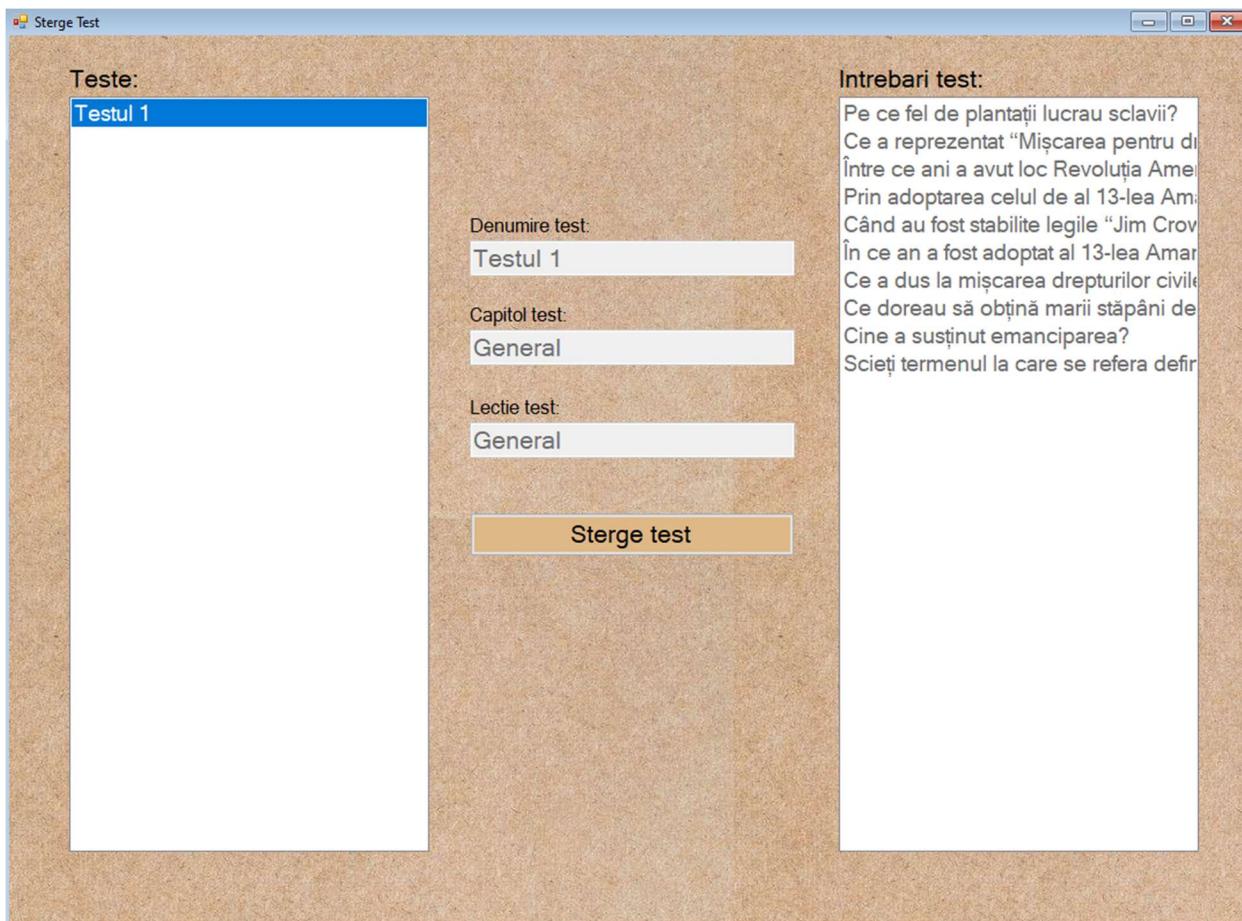
Prin intermediul acestei ferestre profesorii pot șterge testele pe care ei le-au încărcat.

Fereastra conține:

- Un *ListBox* în care se încarcă teste (lstTeste).
- Un *Label* cu textul “Teste”.
- Un *Panel* care conține trei *TextBox*-uri în care se încarcă denumirea testului (txtDenumTest), capitolul (txtCapitol) și lecția (txtLectie), acestea fiind *Enable=false*. Mai conține un *Label* cu textul “Intrebări test”, un *ListBox* în care se încarcă întrebările pe care le conține testul (lstIntrTest) și un *Button* pentru ștergerea lecției (btnSalveazaModif).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



Pentru încărcarea testelor, a întrebărilor pe care le conține testul selectat și pentru ștergerea testului din baza de date se utilizează următoarele secvențe de cod:

```
//Încărcarea testelor

//se adauga testele in ListBox
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM teste WHERE idProfesor=@idProf";
cmd.Parameters.AddWithValue("idProf", this.Tag.ToString());
```

```

MySqlDataReader r = cmd.ExecuteReader();
while (r.Read())
{
    Teste test = new Teste();
    {
        test.idTest = r["idTest"].ToString();
        test.denumire = r["denumire"].ToString();
        test.idProfesor = r["idProfesor"].ToString();
        test.capitol = r["capitol"].ToString();
        test.lectie = r["lectie"].ToString();
    }
    lstTeste.Items.Add(test);
}
r.Close();
conexiune.Close();

```

//Încărcarea întrebărilor

```

private void lstTeste_SelectedIndexChanged(object sender, EventArgs e)
{
    if(lstTeste.SelectedIndex!=-1)
    {
        pnlDetalii.Visible = true;

        Teste test = lstTeste.SelectedItem as Teste;
        txtDenumTest.Text = test.denumire;
        txtCapitol.Text = test.capitol;
        txtLectie.Text = test.lectie;

        //se adauga intrebarile in ListBox
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT * FROM continut_teste JOIN intrebari ON
continut_teste.idIntrebare=intrebari.idIntrebare WHERE idTest=@pidTest";
        cmd.Parameters.AddWithValue("pidTest", test.idTest);
        MySqlDataReader r = cmd.ExecuteReader();
        while(r.Read())
        {
            lstIntrTest.Items.Add(r["intrebare"].ToString());
        }
        r.Close();
    }
}

```

```

        conexiune.Close();

    }
    else
    {
        MessageBox.Show("Selectati testul pe care doriti sa il stergeti.");
    }
}

//Stergerea testului

private void btnSalveazaModif_Click(object sender, EventArgs e)
{
    Teste test = lstTeste.SelectedItem as Teste;

    //se sterge testul din baza de date
    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "DELETE FROM teste WHERE idTest=@pidTest";
    cmd.Parameters.AddWithValue("pidTest", test.idTest);
    cmd.ExecuteNonQuery();

    cmd.CommandText = "DELETE FROM continut_teste WHERE idTest=@pidT";
    cmd.Parameters.AddWithValue("pidT", test.idTest);
    cmd.ExecuteNonQuery();

    conexiune.Close();
    MessageBox.Show("Testul a fost sters cu succes.");
}

FrmStergeTest_Load(sender, e);
}

```

3.17. Fereastra FrmSchimbaParola

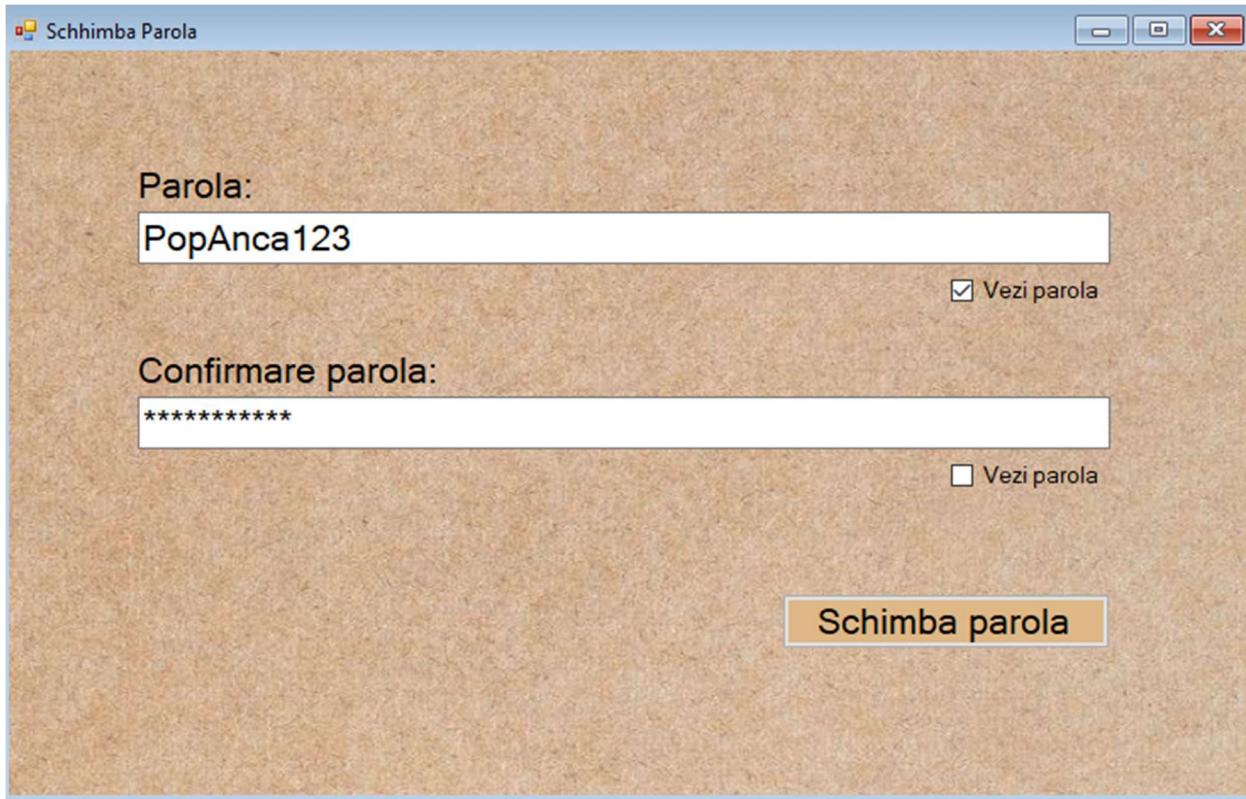
Prin intermediul acestei ferestre utilizatorul poate să schimbe parola.

Fereastra conține:

- Două *Label*-uri cu textele: “Parola:” și “Confirmare parola:”
- Două *TextBox*-uri în care utilizatorul să completeze parola nouă (txtParola) și confirmarea acesteia (txtConfParola).
- Două *CheckBox*-uri care determină dacă parola/ confirmarea parolei sunt vizibile sau dacă se afișează caracterul ‘*’ în locul fiecărui caracter pe care îl conține parola (cbParola, cbConfParola).
- Un *Button* pentru salvarea noii parole (btnSchimbaParola).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



Pentru modificarea parolei se utilizează următoarea secvență de cod:

```
private void btnSchimbaParola_Click(object sender, EventArgs e)
{
    if(txtParola.Text==txtConfParola.Text && txtParola.TextLength >= 8)
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "UPDATE conturi SET parola=@pparola WHERE id=@pid";
        cmd.Parameters.AddWithValue("pparola", txtParola.Text);
        cmd.Parameters.AddWithValue("pid", this.Tag.ToString());
        cmd.ExecuteNonQuery();
        conexiune.Close();
        MessageBox.Show("Parola a fost modificata cu succes.");
        this.Close();
    }
}
```

```
else
{
    if(txtParola.Text != txtConfParola.Text)
    {
        MessageBox.Show("Parolele nu corespund.");
    }
    else
    {

        MessageBox.Show("Parola trebuie sa contina minim 8 caractere.");
    }
}
```

3.18. Fereastra FrmNotificari

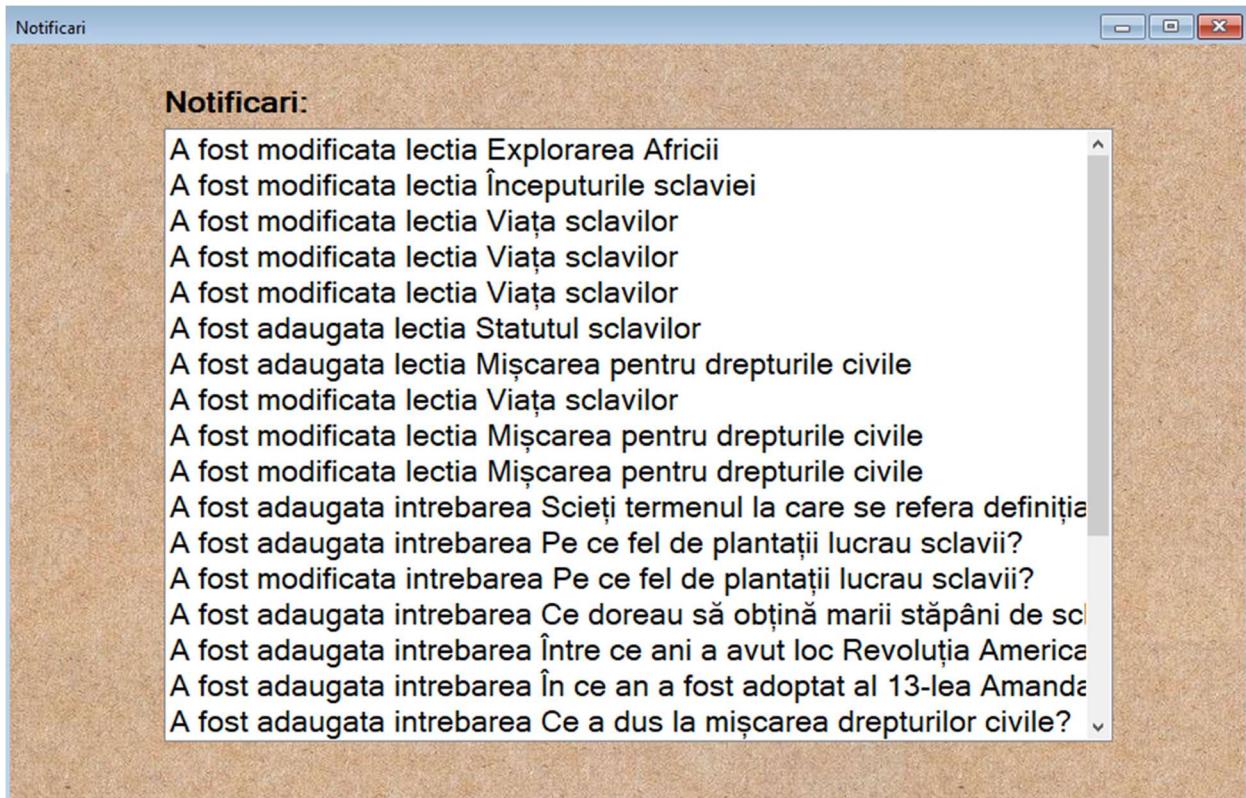
Prin intermediul acestei ferestre utilizatorul va primi o notificare pentru adăugarea unei lecții, modificarea acesteia sau adăugarea și modificarea testelor, respectiv adăugarea și modificarea întrebărilor (doar pentru profesori).

Fereastra conține:

- Un *Label*-uri cu textul “Notificari:”.
- Două *ListBox*-uri: lstidNotif în care se încarcă id-ul notificării din baza de date și lstNotif în care se încarcă notificările.

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState



La apăsarea unei notificări legate de lecții sau teste, se va deschide fereastra FrmLectii sau FrmTeste. Pentru aceste operații, cât și pentru încărcarea notificărilor din baza de date se utilizează următoarele secvențe de cod:

```
//Se încarcă notificările

lstNotif.Items.Clear();
lstidNotif.Items.Clear();

//Se creeaza conexiunea la baza de date
MySqlConnection conexiune = new MySqlConnection();
conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
conexiune.Open(); //Se deschide conexiunea

MySqlCommand cmd = new MySqlCommand();
cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM notificari WHERE idCont=@pid";
cmd.Parameters.AddWithValue("pid", this.Tag.ToString());

MySqlDataReader r = cmd.ExecuteReader();
while(r.Read())
{
    Notificari notif = new Notificari();
    {
        notif.id = r["id"].ToString();
        notif.idCont = r["idCont"].ToString();
        notif.tipNotificare = r["tipNotificare"].ToString();
        notif.idLectie = r["idLectie"].ToString();
        notif.idTest = r["idTest"].ToString();
        notif.idIntrebare = r["idIntrebare"].ToString();

        if (notif.tipNotificare == "adaugareLectie" || notif.tipNotificare=="modifLectie")
        {
            if(notif.tipNotificare == "adaugareLectie")
            {
                notif.text = "A fost adaugata lectia ";
            }
            else
            {
                notif.text = "A fost modificata lectia ";
            }
        }
    }
}

MySqlConnection con = new MySqlConnection();
```

```

con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
con.Open(); //Se deschide conexiunea

MySqlCommand cm = new MySqlCommand();
cm.Connection = con;
cm.CommandText = "SELECT * FROM lectii WHERE idL=@pid";
cm.Parameters.AddWithValue("pid", notif.idLectie);
MySqlDataReader read = cm.ExecuteReader();
while(read.Read())
{
    notif.text += read["titlu"].ToString();
    break;
}
read.Close();
cm.Parameters.Clear();
con.Close();
}

else
{
    if(notif.tipNotificare=="adaugareTest" || notif.tipNotificare=="modifTest")
    {
        if(notif.tipNotificare== "adaugareTest")
        {
            notif.text = "A fost adaugat testul ";
        }
        else
        {
            notif.text = "A fost modificat testul ";
        }
    }

    MySqlConnection con = new MySqlConnection();
    con.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
    con.Open(); //Se deschide conexiunea

    MySqlCommand cm = new MySqlCommand();
    cm.Connection = con;
    cm.CommandText = "SELECT * FROM teste WHERE idTest=@pid";
    cm.Parameters.AddWithValue("pid", notif.idTest);
    MySqlDataReader read = cm.ExecuteReader();
    while (read.Read())
    {
        notif.text += read["denumire"].ToString();
        break;
    }
}

```

```

        read.Close();
        cm.Parameters.Clear();
        con.Close();
    }
    else
    {
        //verific daca contul este de profesor
        MySqlConnection conn = new MySqlConnection();
        conn.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conn.Open(); //Se deschide conexiunea

        MySqlCommand cmdd = new MySqlCommand();
        cmdd.Connection = conn;
        cmdd.CommandText = "SELECT * FROM conturi WHERE id=@idCont";
        cmdd.Parameters.AddWithValue("idCont", this.Tag.ToString());
        MySqlDataReader rr = cmdd.ExecuteReader();
        rr.Read();
        int drept = 0;
        if (rr["drepturi"].ToString() == "profesor")
        {
            drept = 1;
        }
        rr.Close();
        conn.Close();

        if (drept == 1)
        {
            if (notif.tipNotificare == "adaugaIntrebare")
            {
                notif.text = "A fost adaugata intrebarea ";
            }
            else
            {
                notif.text = "A fost modificata intrebarea ";
            }
        }

        MySqlConnection con = new MySqlConnection();
        con.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        con.Open(); //Se deschide conexiunea

        MySqlCommand cm = new MySqlCommand();
        cm.Connection = con;
        cm.CommandText = "SELECT * FROM intrebari WHERE
idIntrebare=@pid";
    }
}

```

```

        cm.Parameters.AddWithValue("pid", notif.idIntrebare);
        MySqlDataReader read = cm.ExecuteReader();
        while (read.Read())
        {
            notif.text += read["intrebare"].ToString();
            break;
        }
        read.Close();
        cm.Parameters.Clear();
        con.Close();

    }

}

if(notif.ToString()!=null)
{
    lstNotif.Items.Add(notif.ToString());
    lstidNotif.Items.Add(notif.id);
}

if(lstNotif.Items.Count==0)
{
    MessageBox.Show("Nu aveti notificari noi.");
    this.Close();
}

```

//Se încarcă fereastra corespunzătoare notificării pe care am apăsat

```

private void lstNotif_SelectedIndexChanged(object sender, EventArgs e)
{
    if(lstNotif.SelectedIndex!=-1)
    {
        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();

```

```

cmd.Connection = conexiune;
cmd.CommandText = "SELECT * FROM notificari WHERE idCont=@pid AND
id=@pidNotif";
cmd.Parameters.AddWithValue("pid", this.Tag.ToString());
cmd.Parameters.AddWithValue("pidNotif",
lstidNotif.Items[lstNotif.SelectedIndex]);
MySqlDataReader r = cmd.ExecuteReader();
r.Read();
if (r["tipNotificare"].ToString() == "adaugareLectie" ||
r["tipNotificare"].ToString() == "modifLectie")
{
    FrmLectii fl = new FrmLectii();
    fl.MdiParent = this.MdiParent;
    fl.Tag = r["idLectie"].ToString();
    fl.Show();
    this.Close();
}
else
{
    if (r["tipNotificare"].ToString() == "adaugareTest" || r["tipNotificare"].ToString()
== "modifTest")
    {
        FrmTeste ft = new FrmTeste();
        ft.MdiParent = this.MdiParent;
        ft.Tag = r["idTest"].ToString();
        ft.Show();
        this.Close();
    }
}

MySqlConnection con = new MySqlConnection();
con.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
UID=root; PASSWORD=; Allow User Variables=True";
con.Open(); //Se deschide conexiunea

MySqlCommand cmm = new MySqlCommand();
cmm.Connection = con;
cmm.CommandText = "DELETE FROM notificari WHERE id=@pidNotif";
cmm.Parameters.AddWithValue("pidNotif",
lstidNotif.Items[lstNotif.SelectedIndex]);
cmm.ExecuteNonQuery();
con.Close();

r.Close();
conexiune.Close();

```

```
        FrmNotificari_Load(sender, e);  
    }  
}
```

3.19. Fereastra FrmJoc

Prin intermediul acestei ferestre utilizatorul își poate testa cunoștiințele într-un mod distractiv, rezolvând un joc de puzzle, dar în același timp răspunzând la întrebări din lecții, astfel testându-și cunoștiințele.

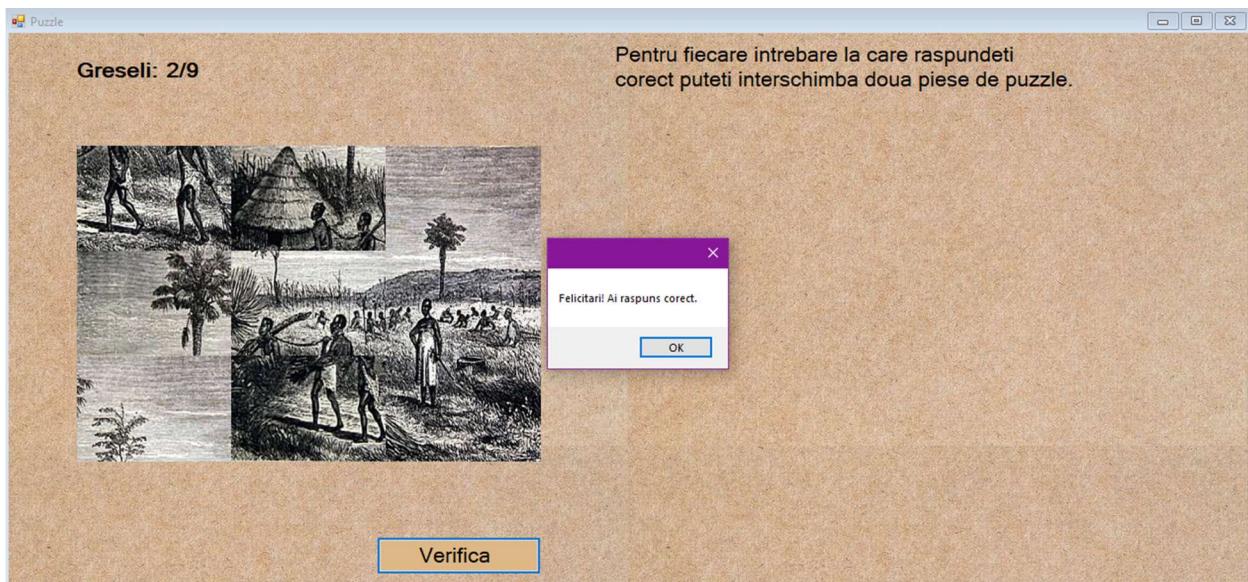
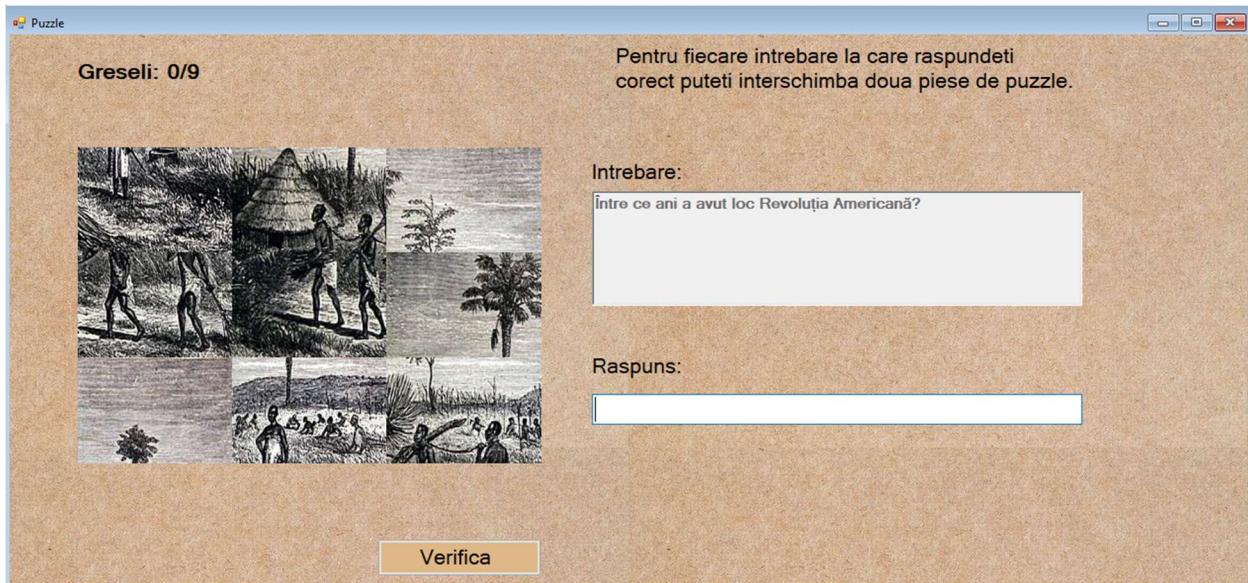
Fereastra conține:

- Trei *Label*-uri: lblGreseli în care se afișează numărul întrebărilor la care s-a răspuns greșit din 9, care este numărul maxim de greșeli admise și două *Label*-uri în care se afișează informații despre modul de utilizare.
- Nouă *PictureBox*-uri în care se afișează piesele de puzzle (o imagine tăiată în nouă bucăți) într-o ordine aleatorie.
- Un *Button* pentru a verifica dacă răspunsul la întrebare este corect sau nu.
- Un *ListBox* pentru încărcarea întrebărilor din baza de date (lstIntrebari).
- Patru *Panel*-uri:
 - pnlIntrebari care conține celelalte trei *Panel*-uri
 - pnlRaspTextScurt care conține două *Label*-uri, un *RichTextBox* pentru încărcarea întrebării (rtbIntrebareTS) și un *TextBox* pentru a scrie răspunsul (txtRaspuns).
 - pnlRaspUnic care conține un *RichTextBox* pentru încărcarea întrebării (rtbIntrebareRU), trei *Label*-uri și trei *RadioButton*-uri pentru alegerea răspunsului corect (rbVar1, rbVar2, rbVar3).
 - pnlRaspMultiplu care conține trei *Label*-uri, un *RichTextBox* pentru încărcarea întrebării (rtbIntrebareRM) și trei *CheckBox*-uri pentru alegerea răspunsului corect/ răspunsurilor corecte (cbVar1, cbVar2, cbVar3).

Pentru această fereastră s-au stabilit următoarele proprietăți:

- BackgroundImage
- BackgroundImageLayout
- FormBorderStyle
- StartPosition
- WindowState

Istoria sclaviei



Pentru încărcarea, verificarea și modificarea puzzle-ului se utilizează următoarele secvențe de cod:

```
void Afisare()
{
    for (int i = 1; i <= 9; i++)
    {
        (Controls["pb" + i] as PictureBox).Image = imagini[(int)(Controls["pb" + i] as
PictureBox).Tag];
    }
}
```

```

}

void IncarcaPuzzle()
{
    Random r = new Random();
    int folder = r.Next(1, 3);
    for (int i = 1; i <= 9; i++)
    {
        imagini[i] = new Bitmap(@"Puzzle\" + folder.ToString() + @"\\" + i + ".jpg");
        (Controls["pb" + i] as PictureBox).Tag = i;
    }
    Afisare();

    for (int i=1;i<=9;i++)
    {
        int nou = r.Next(1, i);
        int aux = (int)(Controls["pb" + nou] as PictureBox).Tag;
        (Controls["pb" + nou] as PictureBox).Tag = (Controls["pb" + i] as PictureBox).Tag;
        (Controls["pb" + i] as PictureBox).Tag = aux;
    }
    Afisare();
}

bool Verificare()
{
    int ok = 0;
    for (int i = 1; i <= 9; i++)
    {
        if ((int)(Controls["pb" + i] as PictureBox).Tag == i)
        {
            ok++;
        }
    }

    if (ok == 9)
    {
        return true;
    }
    else
    {
        return false;
    }
}

private void pb_Click(object sender, EventArgs e)
{
}

```

```

if (selectat == null)
{
    selectat = (sender as PictureBox);
}
else
{
    int aux = (int)selectat.Tag;
    selectat.Tag = (int)(sender as PictureBox).Tag;
    (sender as PictureBox).Tag = aux;
    selectat = null;

    Afisare();
    if (Verificare())
    {
        MessageBox.Show("Felicitari! Puzzle-ul este rezolvat corect!");
    }

    incarcaIntrebare();
}
}

```

Pentru încărcarea întrebărilor, verificarea răspunsurilor și oprirea jocului se utilizează următoarele secvențe de cod:

```

void IncarcareListaIntrebari()
{
    lstIntrebari.Items.Clear();

    MySqlConnection conexiune = new MySqlConnection();
    conexiune.ConnectionString = @"SERVER=localhost; DATABASE=softeducational;
    UID=root; PASSWORD=; Allow User Variables=True";
    conexiune.Open(); //Se deschide conexiunea

    MySqlCommand cmd = new MySqlCommand();
    cmd.Connection = conexiune;
    cmd.CommandText = "SELECT * FROM intrebari";

    MySqlDataReader r = cmd.ExecuteReader();
    while(r.Read())
    {

```

```

        Intrebari intr = new Intrebari();
    {
        intr.idIntrebare = r["idIntrebare"].ToString();
        intr.intrebare = r["intrebare"].ToString();
    }
    lstIntrebari.Items.Add(intr);
}
r.Close();
conexiune.Close();
}

void incarcaIntrebare()
{
    pnlIntrebari.Visible = true;

    pb1.Enabled = pb2.Enabled = pb3.Enabled = false;
    pb4.Enabled = pb5.Enabled = pb6.Enabled = false;
    pb7.Enabled = pb8.Enabled = pb9.Enabled = false;

    Random r = new Random();
    if(lstIntrebari.Items.Count>0)
    {
        int i = r.Next(0, lstIntrebari.Items.Count - 1);

        MySqlConnection conexiune = new MySqlConnection();
        conexiune.ConnectionString = @"SERVER=localhost;
DATABASE=softeducational; UID=root; PASSWORD=; Allow User Variables=True";
        conexiune.Open(); //Se deschide conexiunea

        MySqlCommand cmd = new MySqlCommand();
        cmd.Connection = conexiune;
        cmd.CommandText = "SELECT * FROM raspunsuri WHERE
idIntrebare=@pidIntr";
        Intrebari intr = lstIntrebari.Items[i] as Intrebari;
        cmd.Parameters.AddWithValue("pidIntr", intr.idIntrebare);

        MySqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {
            if (reader["tipRaspuns"].ToString() == "text scurt")
            {
                pnlRaspTextScurt.Visible = true;
                pnlRaspUnic.Visible = pnlRaspMultiplu.Visible = false;

                rtbIntrebareTS.Text = intr.intrebare;
            }
        }
    }
}

```

```

txtRaspuns.Tag = reader["raspuns"].ToString();
break;
}
else
{
    if (reader["tipRaspuns"].ToString() == "unic")
    {
        pnlRaspUnic.Visible = true;
        pnlRaspTextScurt.Visible = pnlRaspMultiplu.Visible = false;

        rtbIntrebareRU.Text = intr.intrebare;

        rbVar1.Text = reader["raspuns"].ToString();
        rbVar1.Tag = reader["corect"].ToString();

        reader.Read();
        rbVar2.Text = reader["raspuns"].ToString();
        rbVar2.Tag = reader["corect"].ToString();

        reader.Read();
        rbVar3.Text = reader["raspuns"].ToString();
        rbVar3.Tag = reader["corect"].ToString();

        break;
    }
    else
    {
        pnlRaspMultiplu.Visible = true;
        pnlRaspUnic.Visible = pnlRaspTextScurt.Visible = false;

        rtbIntrebareRM.Text = intr.intrebare;

        cbVar1.Text = reader["raspuns"].ToString();
        cbVar1.Tag = reader["corect"].ToString();

        reader.Read();
        cbVar2.Text = reader["raspuns"].ToString();
        cbVar2.Tag = reader["corect"].ToString();

        reader.Read();
        cbVar3.Text = reader["raspuns"].ToString();
        cbVar3.Tag = reader["corect"].ToString();

        break;
    }
}

```

```

        }

    reader.Close();
    conexiune.Close();

    lstIntrebari.Items.Remove(lstIntrebari.Items[i]);
}
else
{
    MessageBox.Show("Au intervenit niste probleme la incarcarea jocului. Va rugam
reveniti mai tarziu.");
    this.Close();
}
}

int gresite = 0;
void StopJoc()
{
    if(gresite==9)
    {
        MessageBox.Show("Ai pierdut! Mai invata, apoi hai sa ne jucam!");
        this.Close();
    }
}

private void btnVerifica_Click(object sender, EventArgs e)
{
    corect = verificaIntrebare();

    if(corect==true)
    {
        MessageBox.Show("Felicitari! Ai raspuns corect.");

        pb1.Enabled = pb2.Enabled = pb3.Enabled = true;
        pb4.Enabled = pb5.Enabled = pb6.Enabled = true;
        pb7.Enabled = pb8.Enabled = pb9.Enabled = true;
    }
    else
    {
        incarcaIntrebare();
    }
}

bool verificaIntrebare()
{

```

```

if(pnlRaspTextScurt.Visible==true)
{
    if(txtRaspuns.Text!="")
    {
        if(txtRaspuns.Text==txtRaspuns.Tag.ToString())
        {
            pnlIntrebari.Visible = false;
            reset();
            return true;
        }
    }
    else
    {
        MessageBox.Show("Raspuns gresit.");
        gresite++;
        lblGreseli.Text = gresite.ToString() + "/9";
        pnlIntrebari.Visible = false;
        StopJoc();
        reset();
        return false;
    }
}
else
{
    MessageBox.Show("Trebuie sa raspundeti la intrebare.");
}
}
else
{
    if(pnlRaspUnic.Visible==true)
    {
        if(rbVar1.Checked==true || rbVar2.Checked==true || rbVar3.Checked==true)
        {
            if (rbVar1.Checked == true && rbVar1.Tag.ToString() == "da")
            {
                pnlIntrebari.Visible = false;
                reset();
                return true;
            }
        }
        else
        {
            if (rbVar2.Checked == true && rbVar2.Tag.ToString() == "da")
            {
                pnlIntrebari.Visible = false;
                reset();
                return true;
            }
        }
    }
}

```

```

        else
        {
            if (rbVar3.Checked == true && rbVar3.Tag.ToString() == "da")
            {
                pnlIntrebari.Visible = false;
                reset();
                return true;
            }
            else
            {

                MessageBox.Show("Raspuns gresit.");
                gresite++;
                lblGreseli.Text = gresite.ToString() + "/9";
                pnlIntrebari.Visible = false;
                StopJoc();
                reset();
                return false;
            }
        }
    }
    else
    {
        MessageBox.Show("Trebui sa raspundeti la intrebarile");
    }
}
else
{
    if(cbVar1.Checked==true || cbVar2.Checked==true || cbVar3.Checked==true)
    {
        if(cbVar1.Checked==true && cbVar1.Tag.ToString()=="da" &&
cbVar2.Checked==false && cbVar2.Tag.ToString()=="nu" && cbVar3.Checked==false &&
cbVar3.Tag.ToString()=="nu")
        {
            pnlIntrebari.Visible = false;
            reset();
            return true;
        }
        else
        {
            if (cbVar1.Checked == false && cbVar1.Tag.ToString() == "nu" &&
cbVar2.Checked == true && cbVar2.Tag.ToString() == "da" && cbVar3.Checked == false &&
cbVar3.Tag.ToString() == "nu")
            {
                pnlIntrebari.Visible = false;
            }
        }
    }
}

```

```

        reset();
        return true;
    }
    else
    {
        if (cbVar1.Checked == false && cbVar1.Tag.ToString() == "nu" &&
cbVar2.Checked == false && cbVar2.Tag.ToString() == "nu" && cbVar3.Checked == true
&& cbVar3.Tag.ToString() == "da")
        {
            pnlIntrebari.Visible = false;
            reset();
            return true;
        }
        else
        {
            if (cbVar1.Checked == true && cbVar1.Tag.ToString() == "da" &&
cbVar2.Checked == true && cbVar2.Tag.ToString() == "da" && cbVar3.Checked == false
&& cbVar3.Tag.ToString() == "nu")
            {
                pnlIntrebari.Visible = false;
                reset();
                return true;
            }
            else
            {
                if (cbVar1.Checked == true && cbVar1.Tag.ToString() == "da" &&
cbVar2.Checked == false && cbVar2.Tag.ToString() == "nu" && cbVar3.Checked == true
&& cbVar3.Tag.ToString() == "da")
                {
                    pnlIntrebari.Visible = false;
                    reset();
                    return true;
                }
                else
                {
                    if (cbVar1.Checked == false && cbVar1.Tag.ToString() == "nu"
&& cbVar2.Checked == true && cbVar2.Tag.ToString() == "da" && cbVar3.Checked ==
true && cbVar3.Tag.ToString() == "da")
                    {
                        pnlIntrebari.Visible = false;
                        reset();
                        return true;
                    }
                    else
                    {
                        if (cbVar1.Checked == true && cbVar1.Tag.ToString() == "da"

```

```
void reset()
{
    pnlIntrebari.Visible = false;
    pnlRaspMultiplu.Visible = pnlRaspTextScurt.Visible = pnlRaspUnic.Visible = false;

    rtbIntrebareRM.Text = rtbIntrebareRU.Text = rtbIntrebareTS.Text = "";
    rbVar1.Checked = rbVar2.Checked = rbVar3.Checked = false;
    cbVar1.Checked = cbVar2.Checked = cbVar3.Checked = false;

    rbVar1.Checked = rbVar2.Checked = rbVar3.Checked = false;
    cbVar1.Checked = cbVar2.Checked = cbVar3.Checked = false;
    txtRaspuns.Text = "";
}
```

3.20. Baze de date

softeducational capitole	softeducational lectii	softeducational continut_lectii	softeducational imagini
# idC : int(11) @ Titlu : text	# idL : int(11) # idC : int(11) @ Titlu : text # idProfesor : int(11) @ data_adaugare : date	# id : int(11) # idL : int(11) @ continut : longtext	# idImg : int(11) # idL : int(11) @ denumire : text @ cale : text

softeducational intrebari	softeducational raspunsuri	softeducational teste	softeducational continut_teste
# idIntrebare : int(11) # idProfesor : int(11) @ intrebare : text @ capitol : text @ lectie : text	# idRaspuns : int(11) # idIntrebare : int(11) @ raspuns : text @ tipRaspuns : text @ corect : text	# idTest : int(11) @ denumire : text # idProfesor : int(11) @ data_adaugare : date @ capitol : text @ lectie : text	# id : int(11) # idTest : int(11) # idIntrebare : int(11)

softeducational conturi	softeducational notificari
# id : int(11) @ utilizator : text @ email : text @ parola : text @ drepturi : text @ ultimaLogare : date	# id : int(11) # idCont : int(11) @ tipNotificare : text # idLectie : int(11) # idTest : int(11) # idIntrebare : int(11)

Exemple de interogări utilizate în aplicație:

1. ADĂUGAREA ÎN BAZA DE DATE

```
cmd.CommandText = @"INSERT INTO capitole (titlu) VALUES (@ptitlu);";
cmd.Parameters.AddWithValue("ptitlu", cbCapitole.Text);
cmd.ExecuteNonQuery();
```

2. STERGEREA DIN BAZA DE DATE

```
cmd.CommandText = "DELETE FROM lectii WHERE idL=@pidL";
```

```
cmd.Parameters.AddWithValue("pidL", rtbLectie.Tag);
cmd.ExecuteNonQuery();
```

3. CĂUTAREA ÎN BAZA DE DATE

```
cmd.CommandText = "SELECT * FROM imagini WHERE idL=@ppidL;";
cmd.Parameters.AddWithValue("ppidL", this.Tag);
r = cmd.ExecuteReader();
```

3.21. Clase

```
class Conturi
{
    public string id;
    public string utilizator;
    public string email;
    public string parola;
    public string drepturi;
    public DateTime ultimaLogare;
}

internal class Capitole
{
    public string idC;
    public string titlu;

    1 reference
    public override string ToString()
    {
        return titlu;
    }
}

internal class Continut_lectii
{
    public string id;
    public string idL;
    public string continut;

    1 reference
    public override string ToString()
    {
        return continut;
    }
}
```

```
internal class Continut_teste
{
    public string id;
    public string idTest;
    public string idIntrebare;
}

internal class Imagini
{
    public string idImg;
    public string idL;
    public string denumire;
    public string cale;

    0 references
    public override string ToString()
    {
        return cale;
    }
}

internal class Intrebari
{
    public string idIntrebare;
    public string idProfesor;
    public string intrebare;
    public string capitol;
    public string lectie;

    0 references
    public override string ToString()
    {
        return intrebare;
    }
}
```

```
internal class Lectii
{
    public string idL;
    public string idC;
    public string titlu;
    public string idProfesor;
    public DateTime data_adaugare;

    1 reference
    public override string ToString()
    {
        return titlu;
    }
}

internal class Notificari
{
    public string id;
    public string idCont;
    public string tipNotificare;
    public string idLectie;
    public string idTest;
    public string idIntrebare;
    public string text;

    2 references
    public override string ToString()
    {
        return text;
    }
}

internal class Raspunsuri
{
    public string idRaspuns;
    public string idIntrebare;
    public string raspuns;
    public string tipRaspuns;
    public string corect;

    0 references
    public override string ToString()
    {
        return raspuns;
    }
}
```

```
internal class Teste
{
    public string idTest;
    public string denumire;
    public string idProfesor;
    public string data_adaugare;
    public string capitol;
    public string lectie;

    1 reference
    public override string ToString()
    {
        return denumire;
    }
}
```

Capitolul 4. Utilizarea aplicației

The image displays two windows of a software application with a nautical map background.

Logare (Login Window):

- Nume/Email:** Pop Anca
- Parola:** Vezi parola
- Creeaza cont** (Create Account)
- Intra in cont** (Log In)

ContNou (New Account Window):

- Nume:** Muresan Iustin
- Email:** iustin.muresan@yahoo.com
- Parola:** MuresanIustin Vezi parola
- Confirmare parola:** Vezi parola
- Conturi de:** elev
- Renunță** (Cancel)
- Creează cont** (Create Account)

Istoria sclaviei

Cont de profesor - [FrmLectii]

Contul meu Lectii Teste Joc Administrare lectii Administrare teste

Începuturile. Începuturile sclaviei

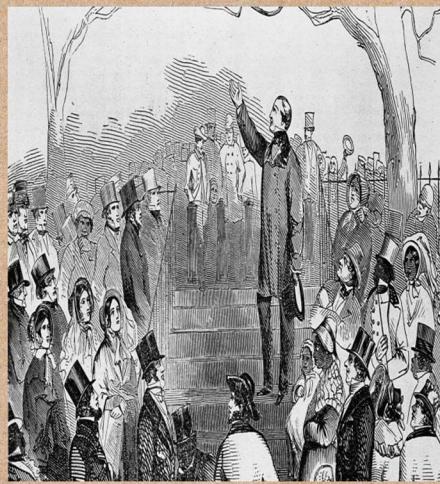
Sclavia (din termenul medieval slavus < slavus) desemnează condiția umană a persoanelor (sclavilor) care lucrează pentru un stăpân fără remunerare și ce nu dispun de drepturi asupra propriei persoane. Din punct de vedere istoric, sclavia înseamnă, prin definiție, negarea egalității între oameni, iar filozofic sclavii erau considerați o specie aparte și inferioară.

Istoria sclaviei se întinde pe mai multe culturi, naționalități și religii din cele mai vechi timpuri până în prezent. Cu toate acestea, pozițiile sociale, economice și juridice ale sclavilor au diferit foarte mult în diferite sisteme de sclavia în diferite perioade și locuri. Dovezile sclaviei sunt anterioare înregistrărilor scrise; practica a existat de multe, dacă nu chiar în majoritatea culturilor. Sclavia a avut loc în civilizații vechi ca Sumer, precum și în aproape orice altă civilizație antică, inclusiv Egiptul antic, China antică, Imperiul Akkadian, Asiria, Babilonia, Persia, Grecia antică, India antică Imperiul Akkadian, Asiria, Babilonia, Persia, Grecia antică, India antică, Imperiul Roman, Califatul arab islamic și Sultanatul, Nubia și civilizațiile precolumbiene ale Americii. Sclavia antică reprezintă un amestec de sclavie datorie, pedeapsă pentru crimă, înrobirea prizonierilor de război, abandonarea copiilor, și nașterea copiilor sclavi.

Începuturile sclaviei pe continental Americană de Nord e legată de secolul al XVII-lea, de întemeierea primelor colonii. Sclavia populației afro-americanăe a avut un anumit impact în evoluția poporului american. Tema sclaviei, a mișcării abolitioniste și a procesului de emancipare a foștilor sclavi a fost intens analizată și dezbatută de-a lungul timpului în contextul american dintr-un amplu ansamblu de puncte de vedere,

Sclavia nu a fost considerată nici necesară și nici profitabilă în primii ani de existență a coloniilor americane. Slujitorii angajați de funcționari erau în primă fază cei care îndeplineau diferite munci în interiorul coloniilor. Sclavia în America a

Mișcarea aboliționistă



Cont de profesor - [FrmLectii]

Contul meu Lectii Teste Joc Administrare lectii Administrare teste

Începuturile. Începuturile sclaviei

și a procesului de emancipare a foștilor sclavi a fost intens analizata și dezbatuta de-a lungul timpului în contextul american dintr-un amplu ansamblu de puncte de vedere,

Sclavia nu a fost considerată nici necesară și nici profitabilă în primii ani de existență a coloniilor americane. Slujitorii angajați de funcționari erau în primă fază cei care îndeplineau diferite munci în interiorul coloniilor. Sclavia în America a început, inițial, prin deportarea în colonia Virginia a diferitor indivizi care au încălcăt legea în Metropola britanică. În general infractorii din Anglia puteau fi pedepsiți prin 3 moduri: bicuirea și trimiterea înapoi în parohia de unde au venit, înrolarea forțată în expediții militare sau deportarea în colonii. Odată ajunși în colonii, acești deportați trebuiau să presteze diferite munci pentru coloniști, în schimbul cărora erau slab remunerati.

Riscul unei rebeliuni în interiorul coloniei era unul foarte crescut. Ambiția proprietarilor de teren de a face profituri cât mai repede și cât mai ușor a instaurat cu pași repezi, sclavia în statul Virginia. Cei slujeau de mai mult timp pe plantații de tutun din Virginia au devenit treptat subiecții unor pedepse din ce în ce mai aspre în caz de fugă. În Virginia, din 1620, astfel de lucrători au ajuns să fie vânduți și cumpărați ca mărfuri. Odată ce sclavia a ajuns să fie acceptată ca sistem de lucru, nici autoritatea centrală nu mai dorea să renunțe la avantajele care provineau din menținerea ei (acoperirea necesarului de forță de muncă, supraproducție și piețe libere).

Istoria sclavilor afro-americani a început oficial pe 20 august 1619, atunci când o navă olandeză a adus în Jamestown (Virginia) 20 de sclavi de origine africană. Oficial, acești primi sclavi afro-americani au fost desemnați inițial ca fiind ucenici. Proprietarul unei plantații de tutun, John Rolfe, consimnează evenimentul în jurnalul său: „Pe la sfârșitul lui august a venit un olandez și ne-a vândut 20 de negri”. În

Sclavi care lucrau pe plantații



Istoria sclaviei

Testul 1

Intrebare:
Pe ce fel de plantații lucrau sclavii?

(Mai multe variante de răspuns posibile)

Variante de răspuns:

- tutun
- orez
- bumbac

Intrebarea urmatoare

Testul 1

Intrebare:
Scieți termenul la care se referă definiția: "desemnează condiția umană a persoanelor care lucrează pentru un stăpân fără remunerare și ce nu dispun de drepturi asupra propriei persoane".

Răspuns:

Punctaj: 3/10

OK

Intrebarea anterioara **Verifica**

Istoria sclaviei

Testul 1

Intrebari la care ati raspuns gresit:

Ce a reprezentat "Mișcarea pentru drepturile omului și ale libertății"?
Între ce ani a avut loc Revoluția Americană?
În ce an a fost adoptat al 13-lea Amendament?
Ce a dus la mișcarea drepturilor civile?

Ce doreau să obțină marii stăpâni de sclavi?

Cine a susținut emanciparea?

Sciți termenul la care se referă definiția: "de la un om către altul"

Raspuns corect:

profit maxim cu cheltuieli minime pe mâna de lucru

Puzzle

Greseli: 0/9

Pentru fiecare intrebare la care raspundeti corect puteti interschimba doua piese de puzzle.

Intrebare:
Pe ce fel de plantații lucrau sclavii?

(Mai multe variante de raspuns posibile)

Variante de raspuns:

tutun
 orez
 bumbac

Verifica

Istoria sclaviei

Greseli: 0/9

Pentru fiecare intrebare la care raspundeti corect puteti interschimba doua piese de puzzle.



A modal window displays the message: "Felicitari! Ai raspuns corect." with an "OK" button.

Verifica

Greseli: 0/9

Pentru fiecare intrebare la care raspundeti corect puteti interschimba doua piese de puzzle.



Intrebare:
Ce a reprezentat "Mișcarea pentru drepturile civile"?

Raspuns gresit.
(din 3 răspuns posibile)

raspuns:

o luptă pentru pierderea drepturilor pe care le au sclavii
 o luptă pentru justiție
 o luptă pentru egalitate

Verifica

Istoria sclaviei

Puzzle

Greseli: 1/9

Pentru fiecare întrebare la care raspundeti corect puteti interschimba doua piese de puzzle.



Intrebare:

În ce an a fost adoptat al 13-lea Amendament?

Raspuns:

Verifica

AdaugaLectie

Capitol: Începuturile +/-

Lectie: Lectie noua +/-

In America de Nord au ajuns peste 4 milioane de sclavi negri, doar în secolul al XVIII lea. Sclavi provineau din rândul triburilor africane de pe coastele continentului și mai rar din interiorul acestuia. Explicația este simplă, erau mai ușor de transportat până la vasele care-i aşteptau în porturi. De obicei existau grupuri de africani sau arabi specializați în vânătoarea de sclavi. Oamenii erau pur și simplu răpiți din sâmbuful familiilor, puși în lanțuri, obligați să mărșăuiască și apoi vânduți ca animalele în targuri. Erau preferați copiii și adulții tineri. Chiar și americani au avut propriile lor "echipe" de vânători de sclavi. Odată îmbarcați, sclavii erau lăsați de multe ori în cale, fără aer, legați în lanțuri și cu apă și hrana insuficiente. Era un adevarat calvar, iar mulți dintre aceștia murdeau pe drum, din cauza bolilor sau a condițiilor de transport. Ajunși pe continentul American, erau vânduți în targuri specializate marilor proprietari de plantații.

În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce marii proprietari dețineau și câte 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubre și munceau până la epuizare. Primeau doar hrana. Majoritatea sclavilor erau incurațiați să se căsătorească între ei, fiindcă se credea că aşa devineau mai docili și, totodată, dădeau naștere altor sclavi. Viața acestora era însă amenințată în permanență de violență sau separare, după cum dorea stăpânul. Înclusiv în secolul al XIX lea, când țările civilizate ale Europei renunțaseră la sclavi, în Statele Unite, sclavia era în floare și, mai mult decât atât, justificată de

Imaginiile se vor putea adăuga după salvarea lectiei, cu o descriere sugestiva.

- +

Lectia a fost salvata cu succes.

OK

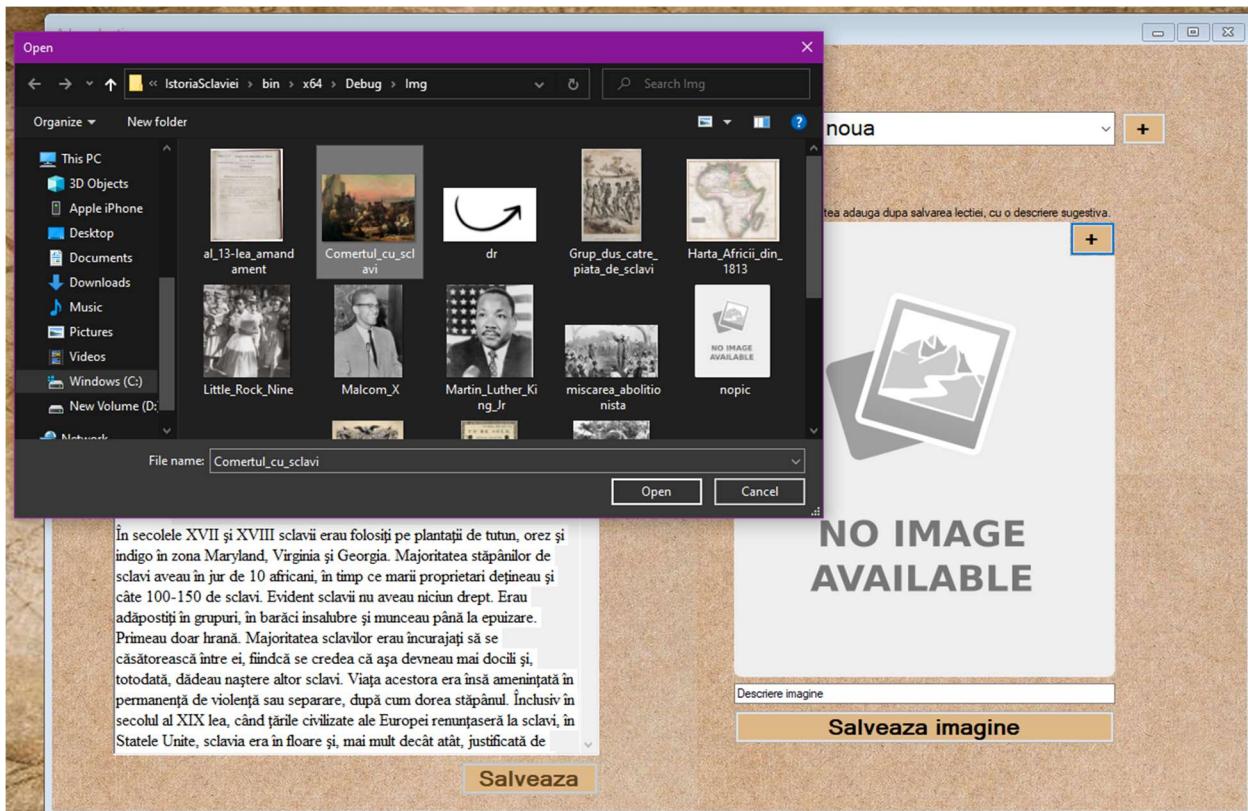
NO IMAGE AVAILABLE

Descriere imagine

Salveaza imagine

Salveaza

Istoria sclaviei



AdaugaLectie

Capitol: Începuturile **Lectie:** Lectie noua

In America de Nord au ajuns peste 4 milioane de sclavi negri, doar în secolul al XVIII lea. Sclavii provineau din rândul triburilor africane de pe coastele continentului și mai rar din interiorul acestuia. Explicația este simplă, erau mai ușor de transportat până la vasile care-i așteptau în porturi. De obicei existau grupuri de africani sau arabi specializați în vânătoarea de sclavi. Oamenii erau pur și simplu răpiți din sâni familiilor, puși în lanțuri, obligați să mărșătuiască și apoi vânduți ca animalele în targuri. Erau preferați copii și adulții tineri. Chine și americanii au avut propriile lor "echipe" de vânători de sclavi. Odată imbarcați, sclavii erau lăsați de multe ori în cale, fără aer, legați în lanțuri și cu apă și hrană insuficiente. Era un adevarat calvar, iar mulți dintre aceștia mureau pe drum, din cauza bolilor sau a condițiilor de transport. Ajunși pe continentul American, erau vânduți în targuri specializate marilor proprietari de plantații.

În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce mari proprietari detineau și căte 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubri și munceau până la epuizare. Primeau doar hrană. Majoritatea sclavilor erau incurați să se căsătorească între ei, fiindcă se credea că aşa devineau mai docili și, totodată, dădeau naștere altor sclavi. Viața acestora era însă amenințată în permanență de violență sau separare, după cum dorea stăpânul. Înclusiv în secolul al XIX lea, când țările civilizate ale Europei renunțaseră la sclavi, în Statele Unite, sclavia era în floare și, mai mult decât atât, justificată de

Salveaza

Salveaza imagine

Istoria sclaviei

AdaugaLectie

Capitol: Începuturile Lectie: Lectie noua

In America de Nord au ajuns peste 4 milioane de sclavi negri, doar în secolul al XVIII lea. Sclavii provineau din rândul triburilor africane de pe coastele continentului și mai rar din interiorul acestuia. Explicația este simplă, erau mai ușor de transportat până la vasele care-i așteptau în porturi. De obicei existau grupuri de africani sau arabi specializați în vânătoarea de sclavi. Oamenii erau pur și simplu răpiți din sănul familiilor, puși în lanțuri, obligați să mărșăuiască și apoi vânduți ca animalele în târguri. Erau preferați copii și adulții tineri.. Chiar și americanii au avut propriile lor "echipe" de vânători de sclavi. Odată îmbarcați, sclavii erau lăsați de multe ori în cale, fără aer, legați în lanțuri și cu apă și hrana insuficiente. Era un adevărat calvar, iar mulți dintre aceștia mureau pe drum, din cauza bolilor sau a condițiilor de transport. Ajunși pe continentul American, erau vânduți în târguri specializate marilor proprietari de plantăjii.

În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce marii proprietari dețineau și câte 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubre și nuncneau până la epuizare. Primeau doar hrana. Majoritatea sclavilor erau incurajați să căsătorească între ei, fiindcă se credea că aşa devineau mai docili și, totodată, dădeau naștere altor sclavi. Viața acestora era însă amenințată în permanență de violență sau separare, după cum dorea stăpânul. Inclusiv în secolul al XIX lea, când țările civilizate ale Europei renunțaseră la sclavi, în Statele Unite, sclavia era în floare și, mai mult decât atât, justificată de

Salveaza

Imaginiile se vor putea adăuga după salvarea lectiei, cu o descriere sugestiva.

+

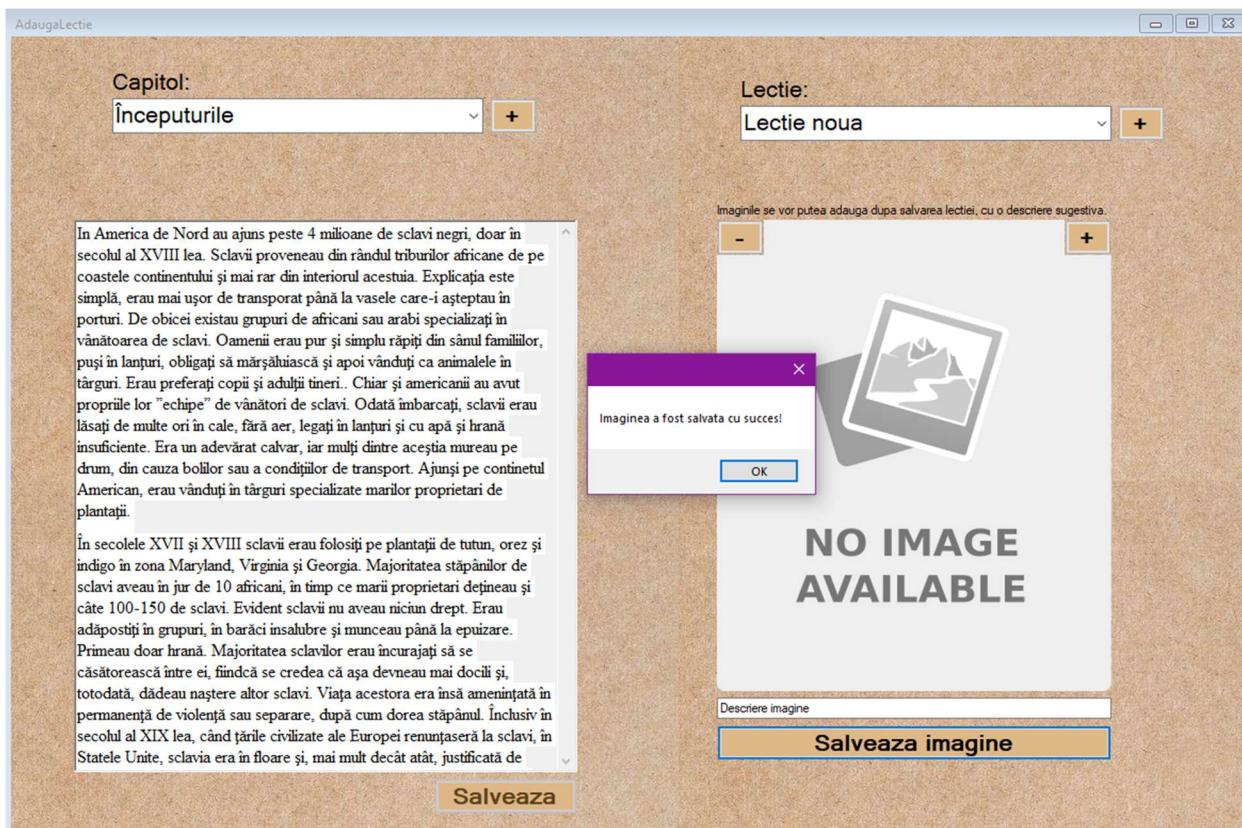
Imaginea a fost salvată cu succes!

OK

NO IMAGE AVAILABLE

Descriere imagine

Salveaza imagine



ModificaLectie

Capitol: Începuturile Lectie: Lectie noua

Adauga imagine Sterge imagine

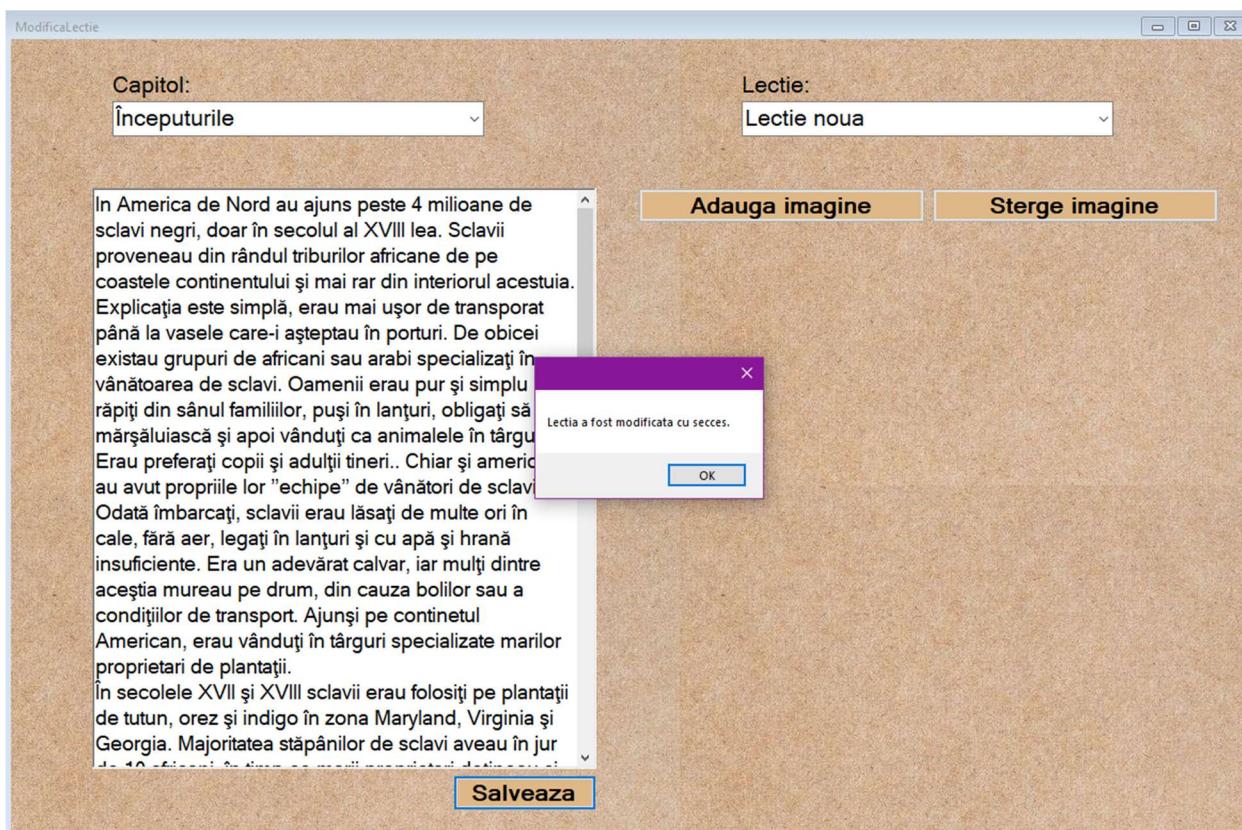
In America de Nord au ajuns peste 4 milioane de sclavi negri, doar în secolul al XVIII lea. Sclavii provineau din rândul triburilor africane de pe coastele continentului și mai rar din interiorul acestuia. Explicația este simplă, erau mai ușor de transportat până la vasele care-i așteptau în porturi. De obicei existau grupuri de africani sau arabi specializați în vânătoarea de sclavi. Oamenii erau pur și simplu răpiți din sănul familiilor, puși în lanțuri, obligați să mărșăuiască și apoi vânduți ca animalele în târguri. Erau preferați copii și adulții tineri.. Chiar și americanii au avut propriile lor "echipe" de vânători de sclavi. Odată îmbarcați, sclavii erau lăsați de multe ori în cale, fără aer, legați în lanțuri și cu apă și hrana insuficiente. Era un adevărat calvar, iar mulți dintre aceștia mureau pe drum, din cauza bolilor sau a condițiilor de transport. Ajunși pe continentul American, erau vânduți în târguri specializate marilor proprietari de plantăjii.

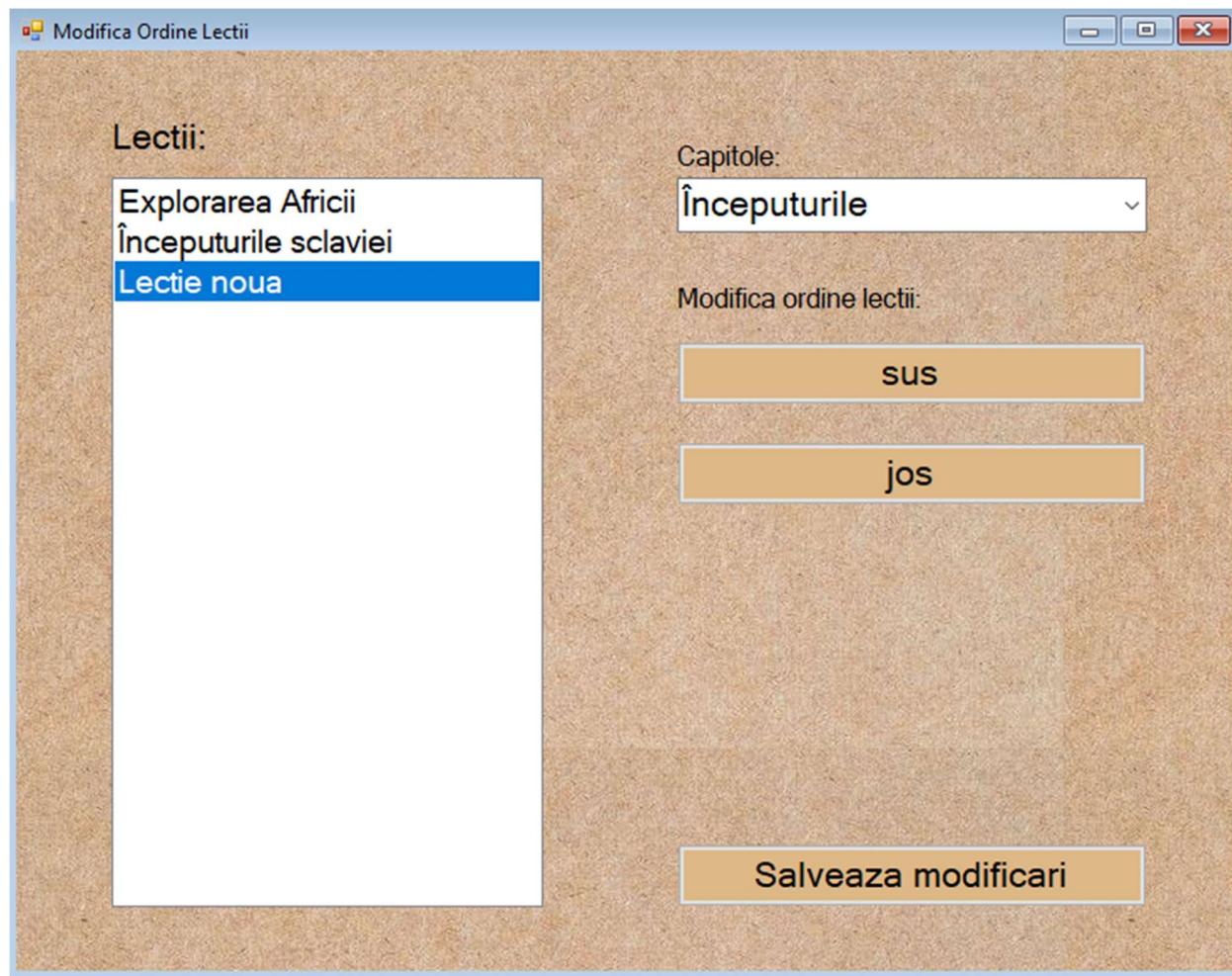
În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce marii proprietari dețineau și câte 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubre și nuncneau până la epuizare. Primeau doar hrana. Majoritatea sclavilor erau incurajați să căsătorească între ei, fiindcă se credea că aşa devineau mai docili și, totodată, dădeau naștere altor sclavi. Viața acestora era însă amenințată în permanență de violență sau separare, după cum dorea stăpânul. Inclusiv în secolul al XIX lea, când țările civilizate ale Europei renunțaseră la sclavi, în Statele Unite, sclavia era în floare și, mai mult decât atât, justificată de

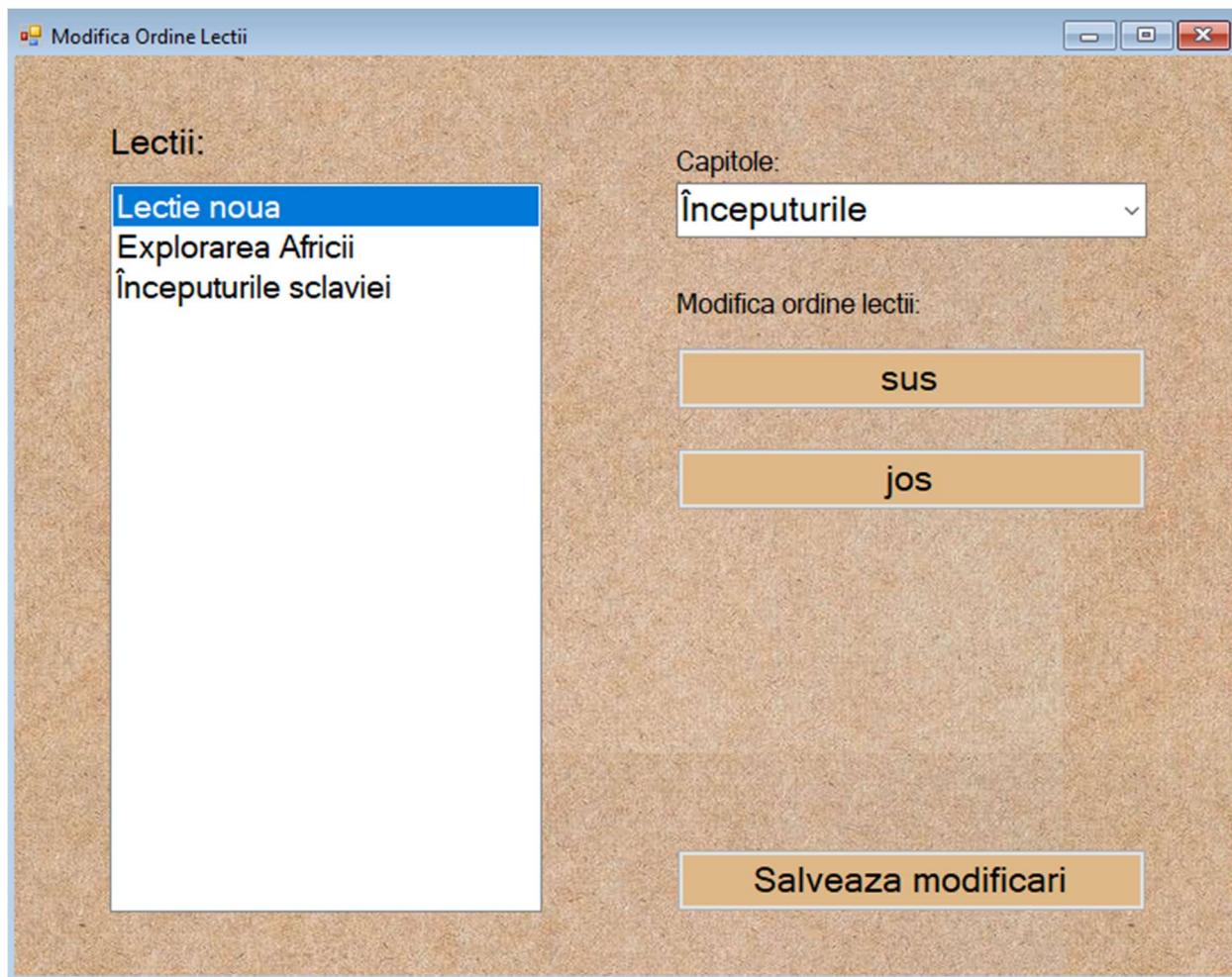
Salveaza

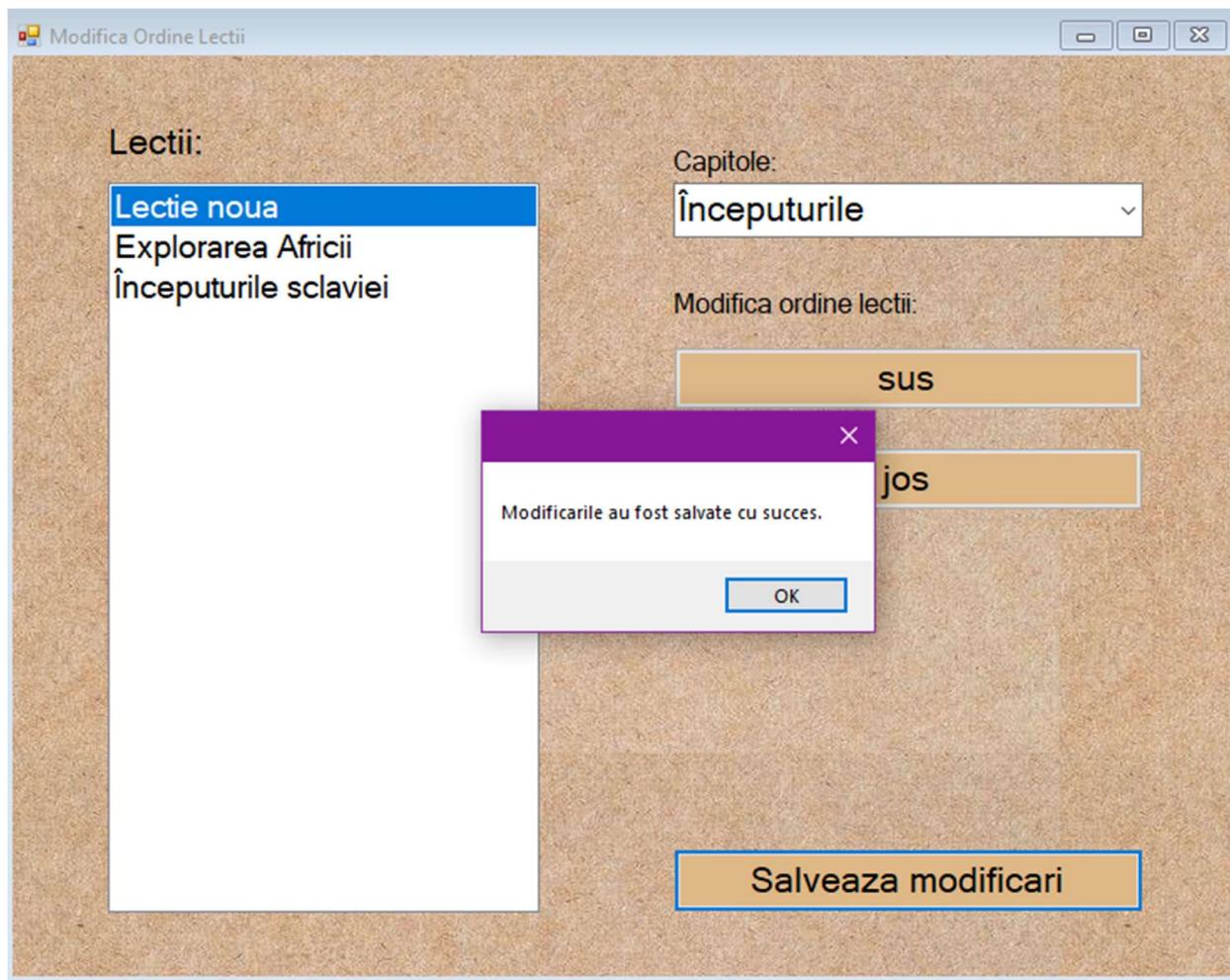
Lectia a fost modificata cu succes.

OK









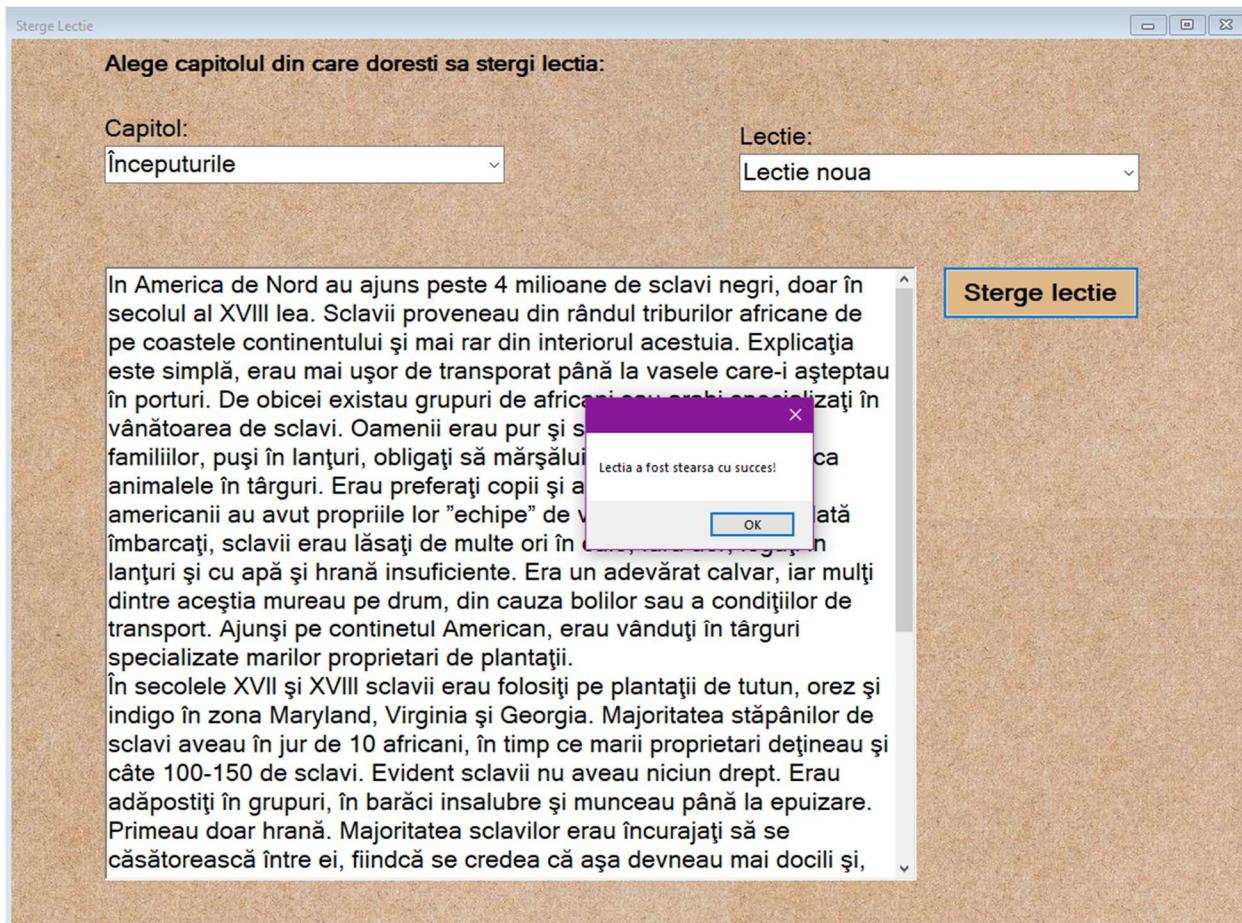
Sterge Lectie

Alege capitolul din care doresti sa stergi lectia:

Capitol: Începuturile Lectie: Lectie noua

In America de Nord au ajuns peste 4 milioane de sclavi negri, doar în secolul al XVIII lea. Sclavii provineau din rândul triburilor africane de pe coastele continentului și mai rar din interiorul acestuia. Explicația este simplă, erau mai ușor de transportat până la vasele care-i așteptau în porturi. De obicei existau grupuri de africani sau arabi specializați în vânătoarea de sclavi. Oamenii erau pur și simplu răpiți din sănul familiilor, puși în lanțuri, obligați să mărșăluască și apoi vânduți ca animalele în târguri. Erau preferați copii și adulții tineri.. Chiar și americanii au avut propriile lor "echipe" de vânători de sclavi. Odată îmbarcați, sclavii erau lăsați de multe ori în cale, fără aer, legați în lanțuri și cu apă și hrană insuficiente. Era un adevărat calvar, iar mulți dintre aceștia mureau pe drum, din cauza bolilor sau a condițiilor de transport. Ajunși pe continentul American, erau vânduți în târguri specializate marilor proprietari de plantații.
În secolele XVII și XVIII sclavii erau folosiți pe plantații de tutun, orez și indigo în zona Maryland, Virginia și Georgia. Majoritatea stăpânilor de sclavi aveau în jur de 10 africani, în timp ce marii proprietari detineau și câte 100-150 de sclavi. Evident sclavii nu aveau niciun drept. Erau adăpostiți în grupuri, în barăci insalubre și munceau până la epuizare. Primeau doar hrană. Majoritatea sclavilor erau încurajați să se căsătorească între ei, fiindcă se credea că aşa devineau mai docili și,

Sterge lectie



Din imaginile anterioare rezultă funcționalitatea aplicației: lecții și teste.

Toate aceste ferestre au un design unitar, pentru a facilita accesul la aplicație chiar și pentru un utilizator nefamiliarizat cu aceasta. În acest sens fiecare formular conține: informații, imagini și butoane pentru a trece de la un set de informații la altul.

Capitolul 5. Extinderi posibile

La proiect s-ar mai putea utiliza:

- Adăugarea unui nou joc, de tipul “Snake”, iar la fiecare întrebare la care se răspunde corect la întâmpinarea unui obstacol să se mai adauge un cerc la șarpe.
- Profesorii să poată adăuga puzzle-uri noi.
- Să se trimită notificări pe email.
- Să existe clase pentru elevi, astfel încât profesorul să se poată organiza mai ușor.

Bibliografie

- “Manualul de informatica intensiv pentru clasa a XII-a” de Vlad Tudor Huțanu și Carmen Popescu
- “Curs de C#. Programare in Visual C# 2008 Express Edition” de Constantin Gălățan și Susana Gălățan
- <https://stackoverflow.com/>
- <https://learn.microsoft.com/en-us/dotnet/csharp/>