

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA

Centrul de Excelență în Informatică și Tehnologii Informaționale



RAPORT

LA PRACTICA INSTRUCTIVĂ

SPECIALITATEA PROGRAMAREA ȘI ANALIZA PRODUSELOR PROGRAM

TEMA: ANILIST

A elaborat elevul: *Pîntea Ion*

Conducătorul practicii: *Jumbei Olga*

Chișinău, 2022

Cuprins

| | |
|---------------------------------------------------------------------------------|----|
| Cuprins | 2 |
| Introducere | 3 |
| Competențe..... | 4 |
| 1. Competențele practicii..... | 4 |
| 2. Competențele profesionale | 4 |
| 3. Competențe specifice | 4 |
| Listing-ul programului | 5 |
| 1. Link către întregul proiect proiectul plasat pe <i>github</i> :..... | 5 |
| 2. Câteva dintre principalele formulare și segmente de cod ale programului..... | 5 |
| Formularele și funcționalitățile programului..... | 7 |
| 1. Formularul de înregistrare, cu verificări ale datelor introduse..... | 7 |
| 2. Formularul pentru logare către programul principal | 8 |
| 3. Formularul principal, unde sunt afișate toate anime-urile | 8 |
| 4. Formularul de afișare a informațiilor despre anime-ul selectat | 9 |
| 5. Butonul „+” va adăuga anime-ul selectat în baza de date personală | 9 |
| 6. Formularul unde sunt afișate anime-urile adăugate de utilizator | 10 |
| 7. Butonul „-” va elimina anime-ul selectat din baza de date personală..... | 10 |
| Baza de date a programului | 11 |
| 1. Diagrama bazei de date | 11 |
| 2. Scenariul bazei de date | 11 |
| 3. Tabelele bazei de date | 12 |
| Concluzie..... | 13 |
| Bibliografie | 14 |

Introducere

Stagiul de practică anual oferă tânărului specialist să experimenteze realizarea unei sarcini mai complexe decât cele de până acum, realizând-o într-un colectiv, ceea ce mimează viitorul acestora, deoarece 90% din viitorii programatori vor lucra într-un astfel de anturaj.

Studentul începător se familiarizează cu programarea, de regulă, în baza elaborării și transcrierii într-un limbaj de programare a algoritmilor de rezolvare a unor probleme concrete simple. Însă, rezolvarea problemelor reale presupune elaborarea unor complexe programe mari, care includ o gamă întreagă de probleme: proiectarea sistemului, elaborarea părților componente ale algoritmului, reunirea diverselor fragmente ale programului într-un produs final, documentarea etc. În cadrul lecțiilor este posibilă examinarea doar a principiilor generale ale metodicii de elaborare a programelor și anumitor aspecte ale rezolvării problemelor. La rândul său, practica de instruire imită întregul proces de elaborare a unui complex program mare, permite studentului să evolueze în rolul de elaborator și organizator al proiectului.

Sarcinile individuale oferite studentului se formulează așa, încât să cuprindă tot materialul studiat de acesta, pentru a i se vedea competențele dobândite de-a lungul perioadei de învățare a respectivului limbaj de programare. Formularea problemei astfel că, revine pe seama conducătorului practicii, deoarece sarcina trebuie să fie suficient de complexă, cât să pună studentul în situația în care acesta trebuie să formuleze propriul său algoritm pentru a o realiza, dar și să nu-l încurce prea tare, încât să se împotmolească și să nu mai poată continua realizarea practicii până la final.

În procesul de realizare a practicii, viitorul specialist, actual student e pus în situația în care trebuie să realizeze anumite sarcini într-un timp limită, astfel că acesta caută astfel de algoritmi și își organizează timpul cât să reușească să-și realizeze sarcina deținută în limita avută, dar și să reușească să-l optimizeze cât să fie ușor de folosit și să funcționeze în orice situație avută.

Stagiul actual de practică are, ca obiective principale, următoarele puncte:

- Consolidarea cunoștințelor teoretice, obținute pe parcursul studierii limbajului și punerea acestora în practică.
- Dezvoltarea abilităților muncii de sine stătătoare, dar și lucrul în echipă.

Pentru practica actuală, am ales ca sarcină de lucru crearea unei aplicații pentru page u Anilist.

Competențe

1. Competențele practice

1. Competențe de învățare sine stătătoare.
2. Competențe de autocunoaștere și autorealizare.
3. Competențe interpersoanel, civile, morale
4. Competențe de bază în matematică, științe și tehnologie
5. Competențe digitale, în domeniul tehnologiei informației și comunicațiilor (TIC)
6. Competențe culturale, interculturale (de a recepta și a crea valori)
7. Competențe antreprenoriale, spirit de inițiativă și antreprenoriat
8. Competențe de acționare strategică

2. Competențele profesionale

1. Elaborarea aplicațiilor, utilizând elementele programării orientate pe obiecte
2. Conceperea produselor multimedia.
3. Modelarea, proiectarea și implementarea sistemelor informatice.
4. Aplicarea fundamentelor teoretice în activitatea profesională
5. Proiectarea algoritmilor și codificarea lui într-un limbaj de programare
6. Respectarea dreptului de autor asupra resurselor digitale, a normelor de etică și securitate informațională
7. Identificarea principiilor de funcționare, clasificare și proiectare a conexiunii între echipamentele rețetelor de calculatoare

3. Competențe specifice

1. Analiza structurală a problemei în studiu
2. Utilizarea tipurilor structurate de date adecvate pentru organizarea și prelucrarea datelor
3. Utilizarea limbajelor SQL și C# la elaborarea sarcinilor
4. Aplicarea tehnicilor de testare și depănare a programelor elaborate
5. Gestionarea informației utilizând resursele Internet
6. Proiectarea descendentă a algoritmilor.

Listing-ul programului

Pentru accesarea mai eficientă a întregului *listing* a programului creat, cât și primirea de actualizări curente și la zi am ales să plasez întregul program creat în visual studio, utilizant limbajul C# pe platforma *github*, care oferă utilizatorului modalitatea de a vedea direct pe site-ul acestora fiecare folder, fișier și codul din acesta, dar și ca ceilalți utilizatori interesați de proiect să poată oferi sugestii, testa programul și face îmbunătățiri.

1. Link către întregul proiect proiectul plasat pe *github*:

<https://github.com/probablymattq/Anilist>

2. Câteva dintre principalele formulare și segmente de cod ale programului

- Clasa unde am creat fiecare linie de cod în sql și apel către baza de date -

<https://github.com/probablymattq/Anilist/blob/master/Practic%C4%83/DataAcces.cs>

```
public List<AnimePersonal> GetAnimePersonal(int idanime, string email)
{
    using (IDbConnection connection = new System.Data.SqlClient.SqlConnection(Helper.CnnVal("AnilistDB")))
    {
        var output = connection.Query<AnimePersonal>($"select * from ANIMEDATAPERSONAL where idanime = '{idanime}' and email_user = '{email}'").ToList();
        return output;
    }
}
```

- UserControl-ul main unde sunt afișare anime-urile din baza de date -

<https://github.com/probablymattq/Anilist/blob/master/Practic%C4%83/Main.cs>

```
for (int j = 0; j < Controls.Count; j++)
{
    if (Controls[j].GetType().ToString() == "System.Windows.Forms.PictureBox")
    {
        PictureBox c = (PictureBox)Controls[j];
        String image = db.GetAnime().Where(id => id.idanime == i).First().animeimage;
        String namepb = Controls[j].Name.ToString();

        AnimeData.Add(new AnimeDataAcces { PictureBoxName = namepb, ClickedAnimeId = i });
        c.ImageLocation = image;
        c.SizeMode = PictureBoxSizeMode.StretchImage;
        c.LoadAsync();
        i++;

        Controls[j].Click += new EventHandler(ShowActionSelector);
    }
}
```

- UserControl-ul UserProfile unde sunt afișare anime-urile selectate de utilizator -

<https://github.com/probablymattq/Anilist/blob/master/Practic%C4%83/UserProfile.cs>

```

if (Controls[j].GetType().ToString() == "System.Windows.Forms.PictureBox")
{
    PictureBox c = (PictureBox)Controls[j];
    String image = db.GetAnimePersonal2(email)[j / 2].animeimage;
    String namepb = Controls[j].Name.ToString();

    AnimeData2.Add(new AnimeDataAcces { PictureBoxName = namepb, ClickedAnimeId = db.GetAnimePersonal2(email)[j / 2].idanime });
    c.ImageLocation = image;
    c.SizeMode = PictureBoxSizeMode.StretchImage;
    c.LoadAsync();
    Controls[j].Visible = true;

    Controls[j].Click += new EventHandler(ShowActionSelector);
}

```

- Formularul principal, unde sunt afișate cele 2 UserControl-uri -

<https://github.com/probablymattq/Anilist/blob/master/Practic%C4%83/Form1.cs>

```

private void Form1_Load(object sender, EventArgs e)
{
    userProfile1.Hide();
    main1.Show();
    pictureBox3.Hide();
    pictureBox2.Show();

    DataAcces db = new DataAcces();
    label1.Text = "Current session: " + db.GetCurrentSession().First().username;
}

```

- Formularul care prezintă informație despre anime-ul selectat -

<https://github.com/probablymattq/Anilist/blob/master/Practic%C4%83/AnimeInformation.cs>

```

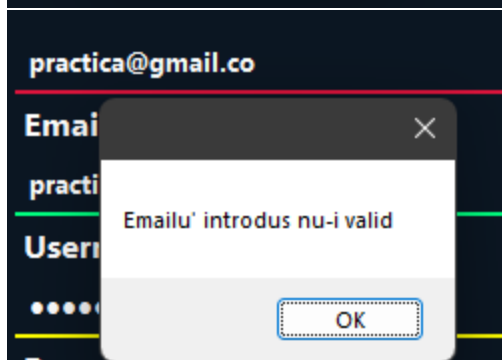
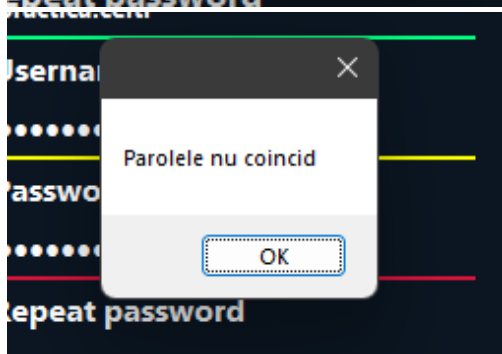
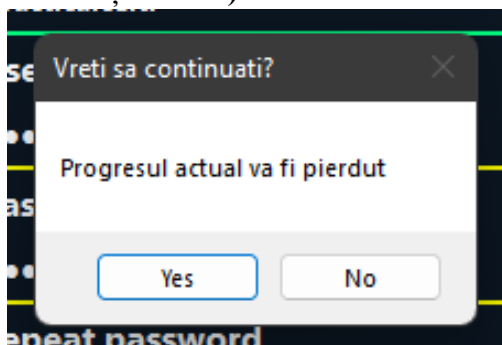
var id1 = Main.AnimeData.Where(a => a.PictureBoxName == Main.PB).First().ClickedAnimeId;
var anime = db.GetAnime().Where(a => a.idanime == id1).First();

label15.Text = anime.numeanime;
label14.Text = anime.nrepisoade.ToString();
label13.Text = anime.status;
label12.Text = anime.dataaparitie.Split(' ')[0].ToString();
label11.Text = db.GetStudio(anime.idstudio);
label10.Text = db.GetGen(anime.idgen);
label19.Text = db.GetSezon(anime.idsezon);

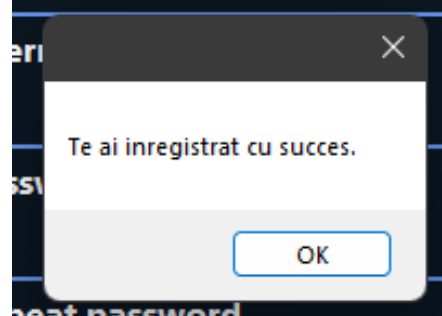
```

Formularele și funcționalitățile programului

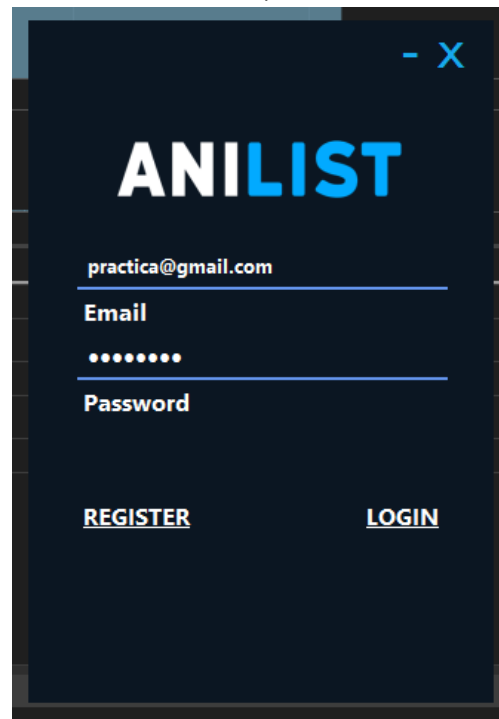
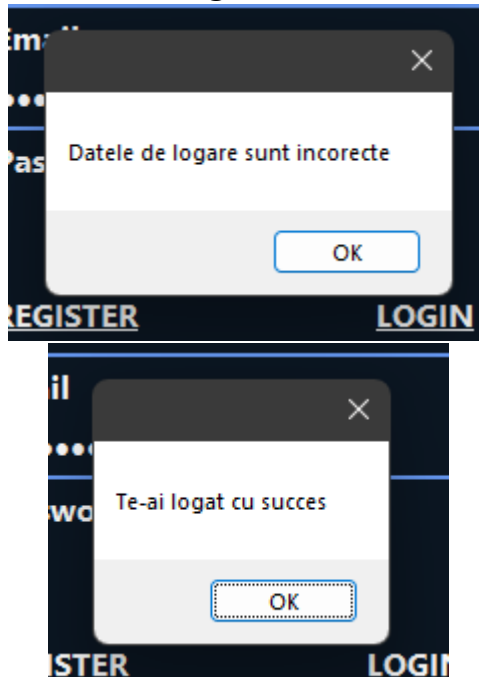
1. Formularul de înregistrare, cu verificări ale datelor introduse (email-uri acceptate, coinciderea parolelor și nivel de caractere maxim și minim)



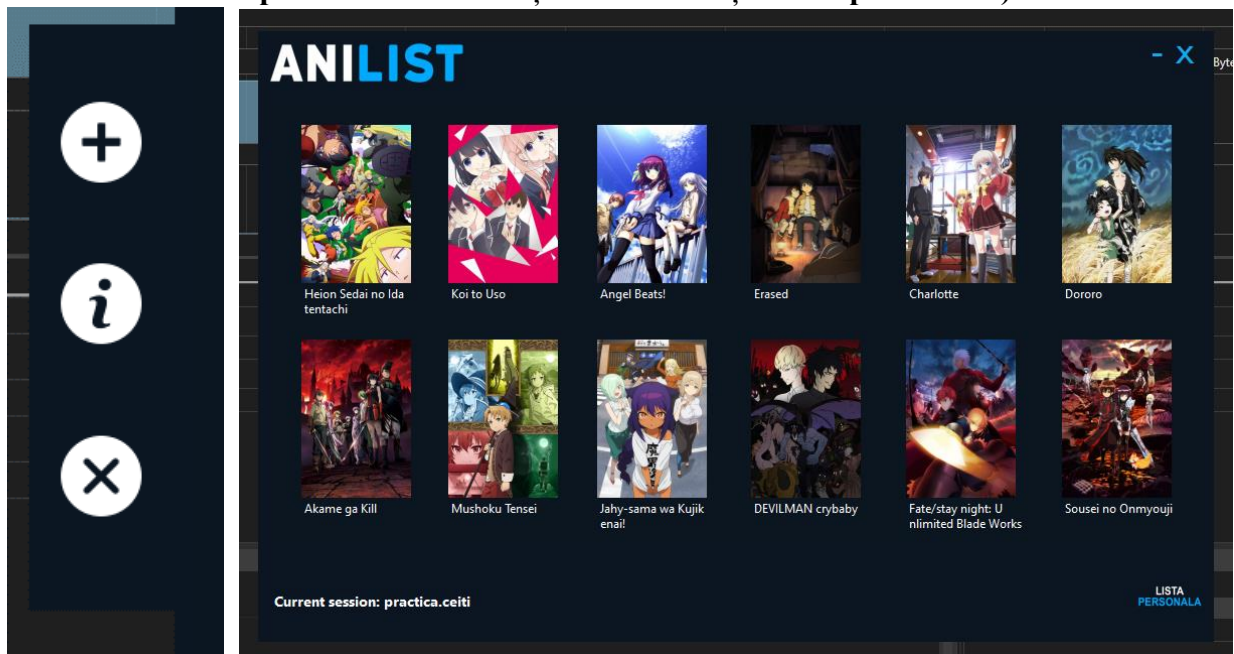
The registration form for ANILIST is displayed on a dark background. It features the ANILIST logo at the top. Below the logo, there are four input fields: "Email" (containing "practica@gmail.com"), "Username" (containing "practica.ceiti"), "Password" (masked with dots), and "Repeat password" (also masked with dots). At the bottom of the form, there are two buttons: "REGISTER" and "LOGIN". A close button (X) is located in the top right corner of the form area.



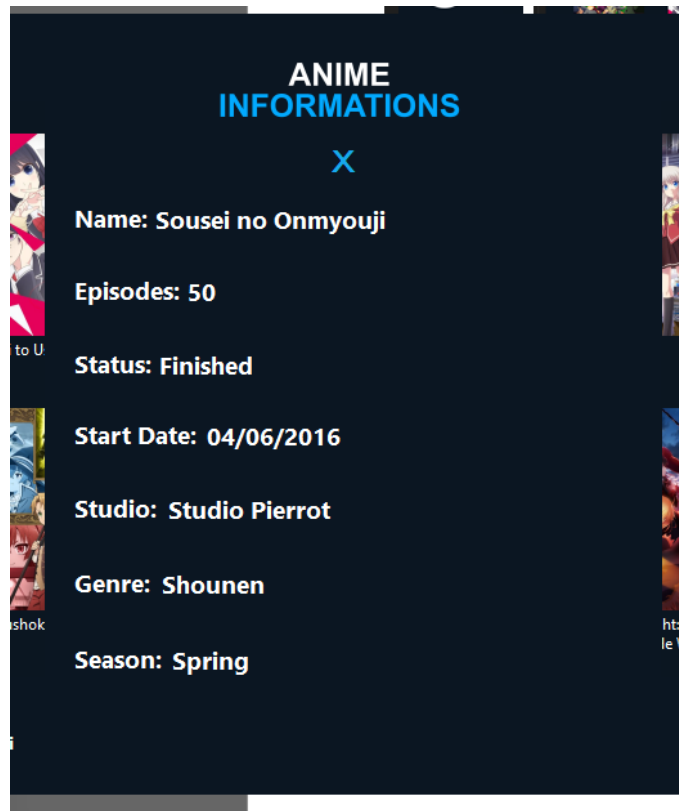
2. Formularul pentru logare către programul principal (funcționează doar dacă datele introduse se regăsesc în baza de date)



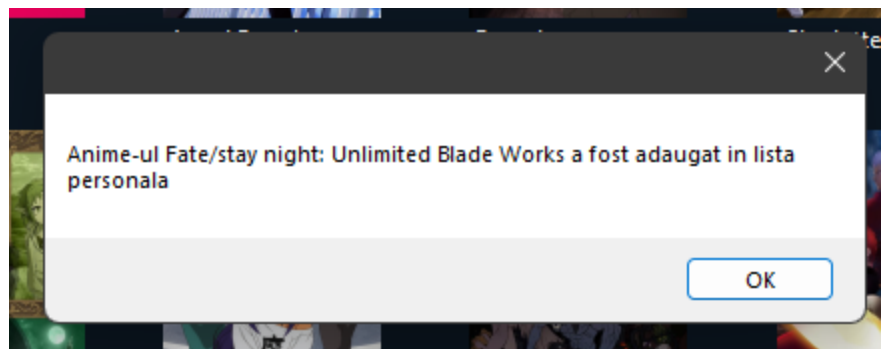
3. Formularul principal, unde sunt afișate toate anime-urile (cu opțiuni de adăugare în baza de date personală sau de afișare a informațiilor despre acestea)



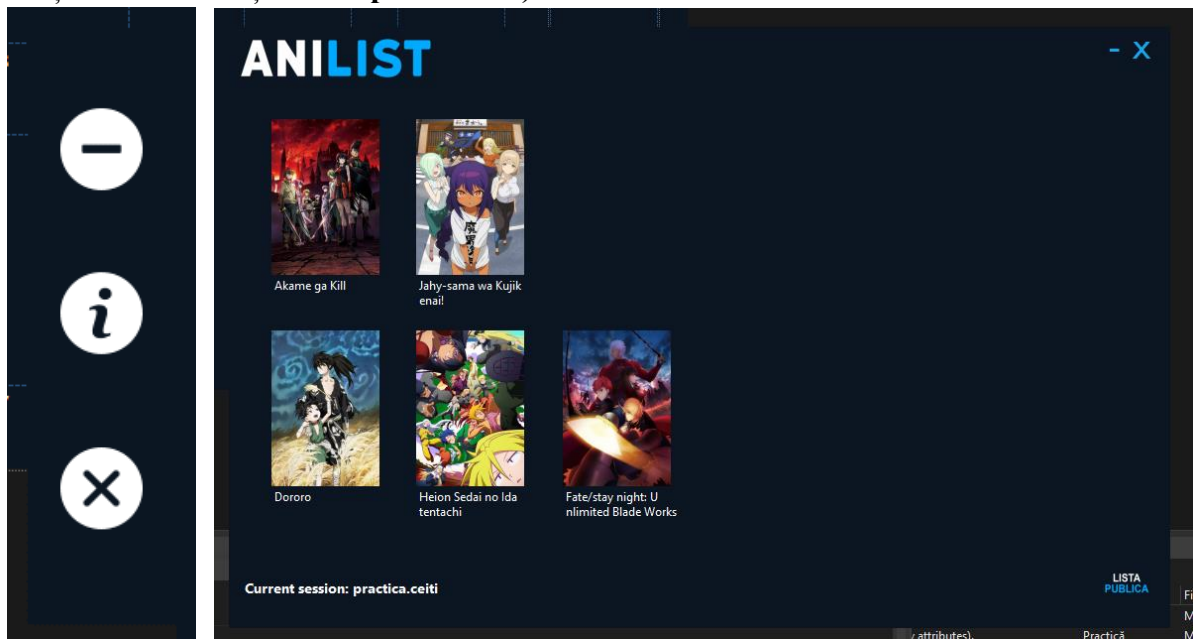
4. Formularul de afișare a informațiilor despre anime-ul selectat, extrase din baza de date, afișate apăsând pe butonul „i”



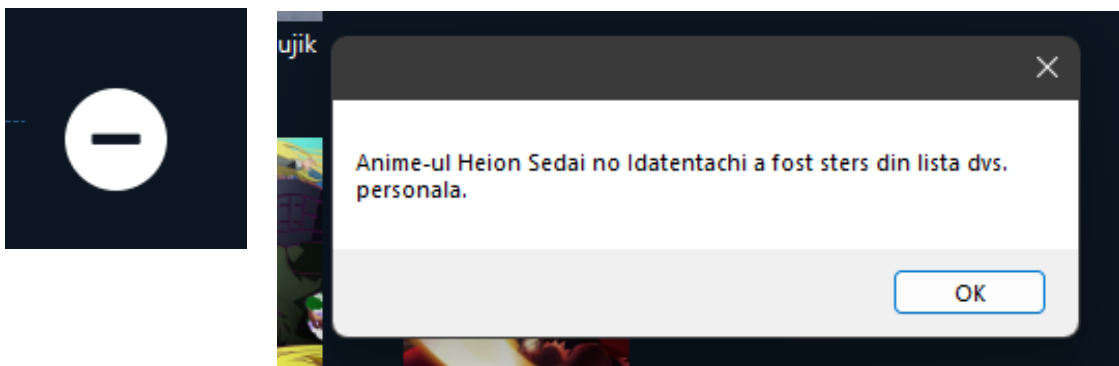
5. Butonul „+” va adăuga anime-ul selectat în baza de date personală, urmând ca la redeschiderea programului să fie afișat în usercontrol-ul UserProfile



6. Formularul unde sunt afișate anime-urile adăugate de utilizator în baza de date personală a acestuia. (cu opțiuni de eliminare din baza de date personală sau de afișare a informațiilor despre acestea)

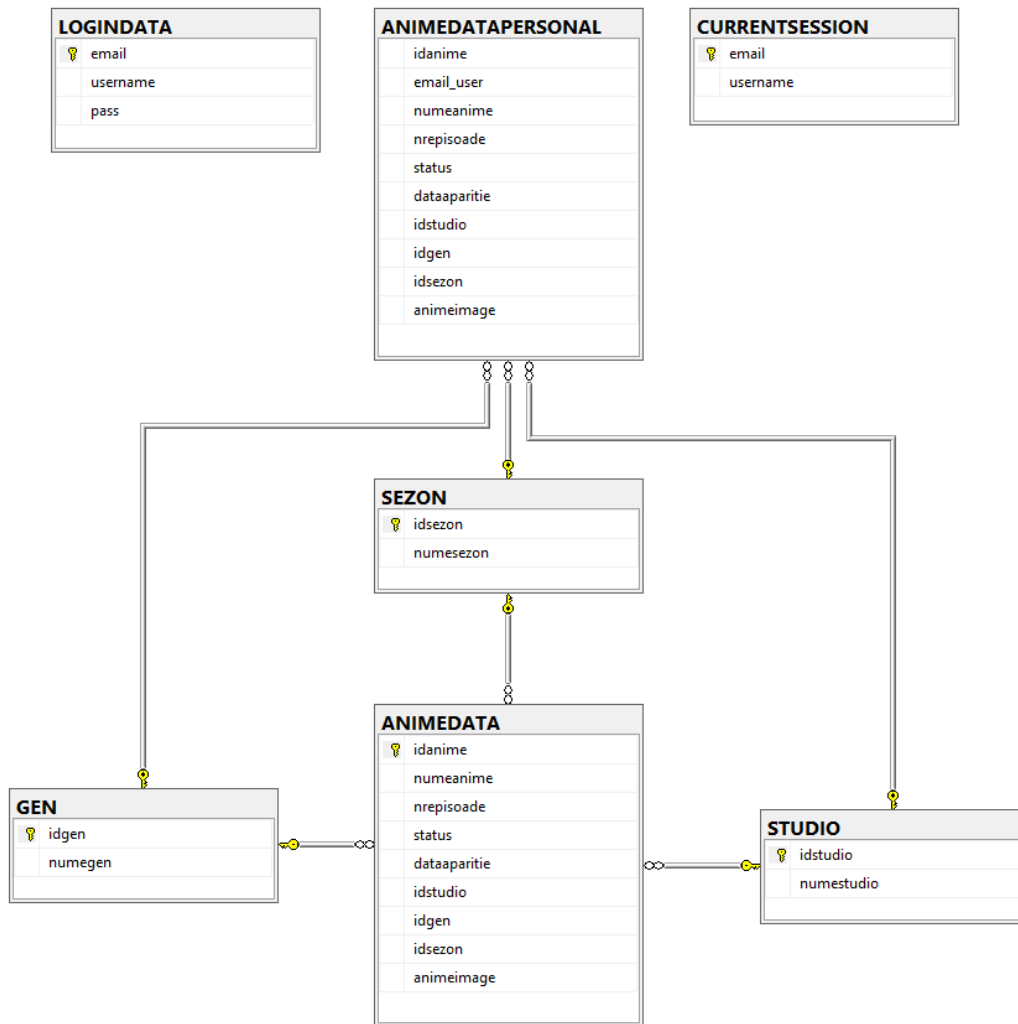


7. Butonul „-” va elimina anime-ul selectat din baza de date personală, urmând ca la redeschiderea programului să nu mai fie afișat respectivul în usercontrol-ul UserProfile.



Baza de date a programului

1. Diagrama bazei de date



2. Scenariul bazei de date

```
CREATE TABLE LOGINDATA(  
email varchar(50) not null primary key,  
username varchar(30) not null,  
pass varchar(30) not null);
```

```
CREATE TABLE CURRENTSESSION(  
email varchar(50) not null primary key,  
username varchar(30) not null);
```

```
CREATE TABLE STUDIO  
(idstudio INT NOT NULL PRIMARY KEY,  
numestudio VARCHAR(30) NOT NULL,);
```

```

CREATE TABLE GEN
(idgen INT NOT NULL PRIMARY KEY,
numegen VARCHAR(30) NOT NULL,);















CREATE TABLE SEZON
(idsezon INT NOT NULL PRIMARY KEY,
numesezon VARCHAR(6) NOT NULL);

CREATE TABLE ANIMEDATA
(idanime INT NOT NULL PRIMARY KEY,
numeanime VARCHAR(50) NOT NULL,
nrepisoade INT,
status CHAR(16) NOT NULL,
dataaparitie DATE,
idstudio INT NOT NULL,
idgen INT NOT NULL,
idsezon INT,
animeimage varchar(300),
CHECK (idsezon >= 1 AND idsezon <=4),
CHECK (nrepisoade >= 0));
ALTER TABLE ANIMEDATA ADD FOREIGN KEY (idstudio) REFERENCES STUDIO(idstudio);
ALTER TABLE ANIMEDATA ADD FOREIGN KEY (idgen) REFERENCES GEN(idgen);
ALTER TABLE ANIMEDATA ADD FOREIGN KEY (idsezon) REFERENCES SEZON(idsezon);

CREATE TABLE ANIMEDATAPERSONAL
(idanime INT NOT NULL,
email_user VARCHAR(50) NOT NULL,
numeanime VARCHAR(50) NOT NULL,
nrepisoade INT,
status CHAR(16) NOT NULL,
dataaparitie DATE,
idstudio INT NOT NULL,
idgen INT NOT NULL,
idsezon INT,
animeimage varchar(300),
CHECK (idsezon >= 1 AND idsezon <=4),
CHECK (nrepisoade >= 0));
ALTER TABLE ANIMEDATAPERSONAL ADD FOREIGN KEY (idstudio) REFERENCES STUDIO(idstudio);
ALTER TABLE ANIMEDATAPERSONAL ADD FOREIGN KEY (idgen) REFERENCES GEN(idgen);
ALTER TABLE ANIMEDATAPERSONAL ADD FOREIGN KEY (idsezon) REFERENCES SEZON(idsezon);

```

3. Tabelele bazei de date

-   **dbo.ANIMEDATA**
-   **dbo.ANIMEDATAPERSONAL**
-   **dbo.CURRENTSESSION**
-   **dbo.GEN**
-   **dbo.LOGINDATA**
-   **dbo.SEZON**
-   **dbo.STUDIO**

- Tabelele bazei de date folosite se împart în 2 categorii: pentru logare (login data și currentsession) și stocare de anime (animedata, animedatapersonal, gen, sezon, studio), fiecare dintre acestea servind la buna funcționare a programului).

Concluzie

În urma finalizării practicii, pot să trag mai multe concluzii, deci voi încerca să le expun cât mai clar și elocvent posibil.

În primul rând, cel mai important lucru consider că a fost că am obținut niște *experiență* despre cum ar trebui să fie pe viitor funcția de programator și să mi-am întărit cunoștințele dobândite până în momentul de față.

Cu ajutorul acestei practici am putut înțelege mai bine cum lucrează programul cu fișiere. Am înțeles cum e mai bine să fac programul ca să fie mai eficient și cât mai mic în volum. La fel am înțeles că trebuie să te atârni destul de serios față de aceea ce faci (de program), pentru a-l face mai plăcut la folosire și mai ușor de înțeles.

Desigur, uneori am întâmpinat unele greutăți, în special în momentul când trebuia să găsesc un algoritm bun sau să găsesc noi metode din limbajul `c#` pe care să le integrez cât mai eficient posibil în cadrul codului meu, fără a fi nevoie să scrii prea mult cod de umplură, care în final doar să îngreuneze funcționalitatea programului, și să mărească dimensiunea lui degeaba.

În cadrul lecțiilor este posibilă examinarea doar a principiilor generale ale metodicii de elaborare a programelor și anumitor aspecte ale rezolvării problemelor. La rândul său, practica de instruire imită întregul process de elaborare a unui complex program mare, permite studentului să evolueze în rolul de elaborator și organizator al proiectului.

Practica de specialitate constituie o primă lucrare de sine stătătoare a studenților și după rezultatele realizării ei se pot face concluzii cu privire la pregătirea studentului pentru a activa în domeniul programării, de a evidenția studenții cei mai capabili pentru instruirea ulterioară în calitate de specialist în programare.

Ca concluzie finală pot spune cert că această practică a avut un impact destul de bun asupra atârării mele față de acest limbaj de programare, și m-a ajutat să-mi aprofundez cunoștințele.

Bibliografie

1. G. Damian-Timoşenco, E. Ionaşcu-Cuciuc, Ghid de elaborare a curriculumului profesional,
2. Stackoverflow - <https://stackoverflow.com/>
3. Wikipedia - <https://www.wikipedia.org/>
4. GeeksforGeeks - <https://www.geeksforgeeks.org/>
5. W3schools - <https://www.w3schools.com/>
6. CodeProject - <https://www.codeproject.com/>
7. Windows Forms Docs - <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms?view=windowsdesktop-6.0>
8. How to connect C# to SQL - <https://www.youtube.com/watch?v=Et2khGnrIgc>