



" Prestanță. Performanță. Progres."

Coordonator: Prof. Muntean Rareș Mircea

- 1. Lupșe Darius, Clasa: a XI-a**
- 2. Trif Mihai-Alexandru, Clasa: a XI-a**

Liceul Teoretic „Emil Racoviță” Baia Mare

Cuprins

1. Introducere	5
1.1. Descrierea problemei	5
1.2. Soluția propusă	6
1.3. Public țintă	8
2. Funcționalitățile Aplicației	10
2.1. Modulul Admin	10
2.2. Modulul Profesor	11
2.3. Modulul Diriginte	12
2.4. Modulul Elev	13
2.5. Modulul de Recunoaștere Facială	13
3. Arhitectura Aplicației	15
3.1. Componente Majore	15
3.2. Relația dintre Componente	17
3.3. Fluxul Datelor	18
3.4. Avantaje Arhitecturale	19

4. Elemente Distinctive și Puncte Forte	21
4.1. Recunoaștere facială în timp real	21
4.2. Funcționare complet offline	22
4.3. Interfețe separate pe roluri	22
4.4. Automatizarea promovării	23
4.5. Resetarea prezenței	23
4.6. Acces rapid la date educaționale	24
4.7. Zero costuri de licență	24
5. Ghid de Instalare și Configurare a Aplicației	25
5.1. Cerințe preliminare	25
5.2. Instalarea Python	26
5.3. Instalarea bibliotecilor necesare	26
5.4. Instalarea MySQL / MariaDB	27
5.5. Configurarea bazei de date	27
5.6. Folderul de fotografii	28
5.7. Configurări opționale și depanare	29
5.8. Recomandări pentru utilizarea practică	29
6. Probleme Cunoscute și Soluții (Depanare)	30
6.1. Probleme la recunoaștere facială	30
6.2. Prezențele nu se actualizează	31
6.3. Logare eșuată în aplicații	32
6.4. Ferestre afișate incorect	33
6.5. Eroare la importul bibliotecilor	34
6.6. Fotografiiile nu se încarcă	34

7. Îmbunătățiri Viitoare (Roadmap)	36
7.1. Integrare cu catalogul național	36
7.2. Modul de raportare analitică	37
7.3. Sistem inteligent de notificare	37
7.4. Autentificare RFID / QR	38
7.5. Backup automat și restaurare	39
7.6. Aplicație mobile	39
7.7. Export PDF / Word / Excel	40
7.8. Înregistrare video + log biometric	40
7.9. Interfață web (Portal EduFace)	41
8. Opinia Autorului și Utilitatea Proiectului	42
9. Testimoniale (Feedback de la Utilizatori)	44
10. Concluzii Generale	45
11. Bibliografie și Resurse Consultate	47

1. Introducere

1.1. Descrierea Problemei

În școlile din România — mai ales în cele din mediul preuniversitar — procesul de gestionare a activității educaționale suferă de multiple deficiențe structurale. Multe dintre instituții continuă să utilizeze cataloage fizice, fișiere Excel neprotejate sau aplicații disparate care nu comunică între ele. Astfel, munca de zi cu zi a cadrelor didactice este îngreunată, iar accesul elevilor și părinților la date relevante este adesea limitat sau întârziat.

Problemele identificate sunt multiple și recurente:

- Lipsa centralizării digitale: informațiile sunt dispersate între registre fizice, fișiere locale sau aplicații diferite, fără o sursă unică de adevăr;
- Introducerea prezenței manual: profesorii trebuie să marcheze prezența pentru fiecare elev la fiecare oră, un proces repetitiv și predispus la erori;
- Gestionarea inefficientă a notelor și mediilor: notele sunt introduse și calculate manual, fără instrumente de analiză sau validare;
- Transparență scăzută: elevii nu au acces în timp real la evoluția lor, iar părinții sunt dependenți de ședințele cu dirigintele pentru a fi informați;
- Securitate redusă: datele elevilor (informații personale, note, absențe) sunt adesea stocate fără criptare sau autentificare corespunzătoare;

- Lipsa automatizării: promovarea elevilor, actualizarea cataloagelor sau analiza prezenței sunt activități efectuate manual, fără suport digital.

Într-un context în care digitalizarea devine tot mai importantă iar nevoia de eficiență și transparență în educație este esențială, aceste lipsuri nu mai pot fi ignorate. Educația modernă are nevoie de instrumente moderne

1.2. Descrierea Soluției Propuse

Pentru a adresa problemele menționate anterior, proiectul EduFace propune o aplicație desktop unificată destinată gestionării tuturor aspectelor administrative și educaționale dintr-o școală.

Aplicația este construită în întregime în Python și utilizează o bază de date relațională MariaDB/MySQL, oferind un mediu stabil, rapid și extensibil. Interfața grafică este realizată cu Tkinter, asigurând compatibilitate cu toate sistemele desktop moderne.

Funcționalitățile cheie ale sistemului includ:

- Gestionarea notelor și absențelor pentru fiecare elev, pe materie și pe dată;
- Promovarea automată a elevilor, cu păstrarea structurii claselor și eliminarea celor din an terminal;
- Interfață specializată pentru fiecare tip de utilizator (admin, profesor, diriginte, elev);
- Recunoaștere facială în timp real, fără servere externe, pentru marcarea automată a prezenței;

- Resetare automată a prezenței după 50 de minute, pentru cicluri orare recurente;
- Autentificare securizată cu parole criptate, roluri distincte și acces controlat la funcționalități.

Aplicația rulează complet local și offline, fără să depindă de conexiune la internet, servere externe sau licențe comerciale. Această caracteristică o face ideală pentru utilizare în școlile din România, inclusiv în zonele cu infrastructură IT redusă.

Exemplu de utilizare integrată:

1. Profesorul se autentifică și selectează clasa și materia.
2. Interfața încarcă grila elevilor, afișând automat statusul prezenței (cu colorare în funcție de prezență/absență).
3. Profesorul adaugă note și poate marca manual absențe dacă este necesar.
4. În paralel, modulul de recunoaștere facială marchează automat elevii detectați ca prezenți.
5. Toate datele sunt salvate instant în baza de date comună și devin vizibile pentru diriginte, elevi și administrație.

1.3. Public Țintă

Aplicația EduFace a fost concepută special pentru instituțiile de învățământ preuniversitar din România, în special licee și gimnazii, dar este ușor adaptabilă și pentru alte tipuri de școli.

Categoriile de utilizatori vizate sunt:

1. Cadrele Didactice (Profesori)

Profesorii beneficiază de o interfață prietenoasă, care le permite să:

- vadă imediat prezența elevilor;
- adauge note rapid și organizat;
- consulte istoricul educațional al fiecărui elev;
- evite activitățile repetitive și consumatoare de timp.

2. Diriginții

Dirigintele clasei are o interfață dedicată pentru:

- accesul complet la notele și absențele clasei;
- realizarea de corecții (ex: ștergerea unei absențe eronate);
- evaluarea progresului fiecărui elev din clasă;
- pregătirea cataloagelor pentru raportare, ședințe sau arhivare.

3. Elevii

Fiecare elev are un cont propriu, securizat, de unde:

- poate vizualiza în timp real notele și mediile;
- poate vedea istoricul absențelor;

- poate urmări evoluția sa școlară fără a depinde de alți intermediari.

4. Administrația Școlii

Administratorii sistemului pot:

- crea conturi pentru toți utilizatorii;
- adăuga sau modifica clase și materii;
- vizualiza cataloagele complete ale oricărei clase sau elev;
- promova automat elevii;
- șterge notele/absențele incorecte și întreține baza de date.

2. Funcționalitățile Aplicației

Aplicația EduFace este structurată modular, fiecare modul fiind asociat unui rol clar definit în cadrul sistemului educațional. Fiecare rol (admin, profesor, diriginte, elev) are acces doar la funcționalitățile care îi sunt necesare, ceea ce contribuie atât la simplitatea în utilizare, cât și la securitatea sistemului.

2.1. Modulul Admin

Modulul Admin oferă acces complet asupra întregului sistem. Este destinat exclusiv administratorilor IT sau personalului școlii însărcinat cu întreținerea bazei de date și gestionarea utilizatorilor. Interfața este intuitivă și organizată pe categorii (Utilizatori, Management Clasă, Elevi).

Funcționalități principale:

- Adăugarea de entități: elevi, profesori, diriginți, clase, materii;
- Asocierea profesorilor la clase și materii (atribuirile multiple sunt permise);
- Ștergerea notelor și absențelor incorecte din orice clasă sau pentru orice elev;
- Vizualizarea completă a cataloagelor: selectezi clasa și poți consulta notele tuturor elevilor pe materii;
- Promovarea automată a elevilor la final de an, cu:
 - păstrarea literei clasei (ex: 10F → 11F),
 - ștergerea automată a datelor elevilor din clasa a XII-a (note, absențe, conturi);

- Asigurarea structurii bazei de date și menținerea integrității relațiilor dintre entități;
- Autentificare securizată, cu roluri bine definite și parole criptate.

2.2. Modulul Profesor

Acest modul este centrat pe nevoile cadrelor didactice de a administra cataloagele zilnice. Interfața este ușor de folosit, cu accent pe rapiditate și eficiență în introducerea datelor.

Funcționalități principale:

- Autentificare pe bază de cont personal și verificare automată a rolului de profesor;
- Selectarea clasei și materiei predate dintr-o listă personalizată (doar clasele atribuite acelui profesor);
- Vizualizarea grilei elevilor, afișând:
 - fotografia fiecărui elev;
 - statusul prezenței (colorare verde pentru „prezent”, alb/negru pentru „absent”);
- Adăugarea de note pentru fiecare elev, cu:
 - dată personalizabilă,
 - validare numerică;
- Vizualizarea mediilor calculate automat pentru fiecare materie;
- Marcarea manuală a absenței, dacă recunoașterea facială nu a funcționat sau elevul a fost prezent parțial;
- Acces la istoricul notelor și absențelor pentru fiecare elev;

- Feedback vizual rapid asupra performanței elevului (culori și mesaje concise).

2.3. Modulul Diriginte

Dirigintele are un rol esențial în urmărirea evoluției educaționale a întregii clase. Modulul dedicat permite accesul la toate informațiile despre elevii clasei sale, oferindu-i control asupra corectitudinii datelor.

Funcționalități principale:

- Autentificare securizată ca diriginte (rol distinct în baza de date);
- Acces la catalogul complet al clasei proprii, cu afișarea notelor și absențelor pentru toți elevii;
- Vizualizarea grilei elevilor, similară celei pentru profesori;
- Ștergerea de note sau absențe incorecte prin selecție ghidată (materie, elev, dată);
- Filtrare și afișare notițe pe elev, materie sau dată;
- Calcularea mediei generale pe elev, afișată în listă;
- Acces rapid la date utile pentru întocmirea rapoartelor semestriale sau pentru discuții cu părinții.

2.4. Modulul Elev

Elevii beneficiază de o interfață simplificată și securizată, cu acces doar la informațiile proprii. Acest modul este esențial pentru creșterea transparenței și a implicării elevilor în propria educație.

Funcționalități principale:

- Autentificare cu user și parolă înregistrate în baza de date;
- Vizualizarea notelor pe materii, sortate cronologic;
- Afișarea mediei pe fiecare materie;
- Istoric complet al absențelor, cu dată și materie;
- Design curat, fără funcționalități redundante sau opțiuni avansate;
- Acces offline dacă aplicația este rulată local pe un sistem disponibil elevilor.

2.5. Modulul de Recunoaștere Facială

Este componenta inovatoare a aplicației, concepută pentru a automatiza procesul de prezență fără intervenție umană. Folosește biblioteca face_recognition și OpenCV, oferind identificare biometrică precisă în timp real.

Funcționalități principale:

- Detecție facială în timp real prin camera web (externă sau integrată);
- Marcarea automată a prezenței în baza de date dacă fața corespunde unui elev;

- Resetarea prezenței după 50 de minute, pentru a permite un nou ciclu de prezență;
- Log jurnal cu toate evenimentele detectate (prezențe, necunoscuți, erori);
- Capacitate de rulare în mod headless (fără GUI), pentru a fi folosit pe sisteme de supraveghere fără ecran;
- Feedback vizual instant în interfață (rectanglu pe față, nume, timp marcaj);
- Compatibilitate cu mai multe camere, cu opțiunea de schimbare rapidă între ele;
- Funcție automată de detectare a necunoscuților, cu logare în jurnal și posibilă extindere pentru alerte viitoare.

3. Arhitectura Aplicației

3.1. Componente Majore

Arhitectura aplicației EduFace este gândită pentru modularitate, claritate și scalabilitate. Fiecare rol are un fișier Python dedicat, care comunică cu aceeași bază de date relațională. Aplicația nu necesită server web, browser sau servicii externe, ci rulează complet local, ca aplicație desktop.

1. Baza de date (MariaDB / MySQL)

- Reprezintă nucleul sistemului.
- Conține tabelele: users, students, teachers, head_teachers, classes, subjects, grades, attendance_current, attendance_history, teacher_assignments, user_roles.
- Structura este normalizată, folosind chei străine pentru menținerea integrității relaționale.
- Este locală și poate fi administrată cu instrumente precum phpMyAdmin sau DBeaver.

2. Backend-ul (Python)

Fiecare fișier Python reprezintă un modul dedicat unui rol:

- admin.py — pentru administratori;
- catalog_profesori.py — pentru profesori;
- catalog_diriginti.py — pentru diriginți;

- `elevi.py` — pentru interfața elevului;
- `face.py` — pentru recunoaștere facială și prezență automată;
- `catalog.py` — meniu central care lansează toate modulele.

Toate aceste fișiere folosesc biblioteca `pymysql` pentru conectarea la baza de date.

3. Interfața grafică (Tkinter)

- Realizată cu Tkinter, un GUI toolkit simplu și eficient inclus în Python.
- Nu necesită instalare suplimentară.
- Design intuitiv, cu ferestre, butoane, etichete și liste (Treeview).
- Fiecare modul are propria fereastră, fără interdependențe vizuale între ele.

4. Modulul de Recunoaștere Facială (`face.py`)

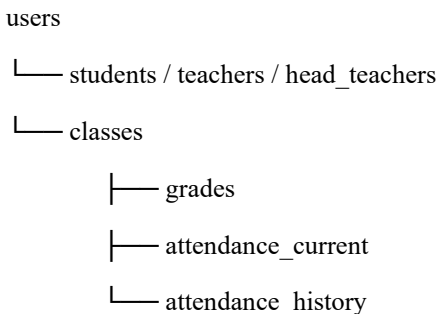
- Bazat pe biblioteca `face_recognition` + `OpenCV` + `numpy`.
- Deschide camera și detectează fețele în timp real.
- Compară encodările fețelor cu cele salvate în fișierele foto.
- Actualizează automat tabela `attendance_current`.

3.2. Relația dintre Componente

Comunicarea între componente nu se face prin schimb de date între fișiere Python, ci exclusiv prin intermediul bazei de date. Această abordare oferă:

- Separare logică completă — fiecare modul poate fi rulat sau modificat independent;
- Simplitate în întreținere — codul este compartimentat și ușor de urmărit;
- Flexibilitate pentru extindere — viitoare aplicații mobile/web pot accesa aceeași bază.

Schema relațională simplificată (logică):



De exemplu:

- students are un user_id și un class_id;
- grades are un student_id și un subject;
- teacher_assignments mapează teacher_id la class_id + subject_id.

3.3. Fluxul Datelor

❖ Pornirea aplicației

Fiecare utilizator deschide aplicația aferentă rolului său (ex: profesorul deschide `catalog_profesori.py`).

❖ Autentificarea

Se face prin tabelul `users`, folosind parole criptate cu `bcrypt`. Se verifică rolul utilizatorului, apoi se continuă doar dacă are permisiunea aferentă modulului respectiv.

❖ Obținerea datelor

După autentificare:

- Profesorii văd doar clasele și materiile alocate;
- Diriginții văd doar clasa proprie;
- Elevii văd doar datele proprii;
- Adminul are acces la tot.

❖ Recunoaștere facială (în paralel)

- `face.py` este rulat pe un alt sistem (sau în fundal) și detectează fețele elevilor;
- Actualizează tabela `attendance_current` pentru a marca prezența automat.

❖ Actualizări și vizualizare

Toate datele introduse de profesori sau detectate de face.py sunt vizibile imediat în modulele diriginte/elev/admin, fără sincronizare suplimentară.

3.4. Avantaje Arhitecturale

❖ Modularitate

Codul este împărțit în module independente. Un bug sau o modificare într-un modul (ex: elevi.py) nu afectează restul sistemului.

❖ Securitate

- Accesul este bazat pe roluri;
- Parolele sunt criptate;
- Nu există transfer de date prin rețea, ceea ce reduce riscul de interceptare.

❖ Scalabilitate

- Baza de date poate fi mutată pe un server de rețea (intern sau online);
- Aplicația poate fi integrată cu viitoare platforme web sau mobile;
- Se pot adăuga funcționalități fără a afecta structura principală.

❖ **Portabilitate**

- Funcționează pe orice sistem cu Python: Windows, Linux, Mac;
- Nu necesită server web, doar Python și MySQL/MariaDB;
- Poate fi rulat de pe stick USB, laptop personal sau desktop dedicat.

4. Elemente Distinctive și Puncte Forte

Aplicația EduFace se remarcă printr-o serie de caracteristici care o diferențiază net de soluțiile existente în prezent, fie ele software comerciale sau instrumente improvizate în școli (ex. fișiere Excel, Google Sheets, cataloage în PDF).

4.1. Recunoaștere facială în timp real, fără servere externe

Modulul de recunoaștere facială integrează bibliotecile `face_recognition` și `OpenCV` pentru a detecta automat prezența elevilor în timpul orei. Procesul este complet local și se desfășoară astfel:

- Camera web captează imaginea elevului;
- Sistemul identifică fața și o compară cu imaginile stocate local;
- Dacă există o potrivire, elevul este marcat automat ca „prezent” în baza de date.

Avantaje:

- Elimină complet necesitatea de a marca manual prezența;
- Reduce erorile umane și tentativele de fraudă (ex: răspunsul altui coleg);
- Nu necesită servere externe, cloud sau internet — totul se face offline, local.

4.2. Funcționare complet autonomă, fără dependență de internet

Toate modulele aplicației rulează local, iar baza de date este găzduită pe același calculator sau rețea internă. Niciun modul nu necesită conexiune la internet pentru a funcționa.

Avantaje:

- Ideal pentru școlile din mediul rural sau din zone cu infrastructură slabă;
- Evită riscurile de downtime cauzate de lipsa semnalului;
- Securitate crescută — datele nu sunt expuse online.

4.3. Roluri clar delimitate și interfețe dedicate

Fiecare categorie de utilizator beneficiază de un fișier Python separat, cu o interfață creată special pentru nevoile sale:

- `admin.py` — managementul general al sistemului;
- `catalog_profesori.py` — pentru profesori;
- `catalog_diriginti.py` — pentru diriginți;
- `elevi.py` — pentru elevi;
- `face.py` — pentru recunoașterea facială.
-

Avantaje:

- Claritate: fiecare utilizator vede doar ceea ce îi este relevant;
- Securitate: rolurile sunt verificate la autentificare;
- Ușurință în folosire: ferestre simple, aerisite și intuitive.

4.4. Automatizarea promovării elevilor și profesorilor

Modulul de promovare permite administratorului să promoveze elevii la sfârșitul anului școlar cu un singur click. Aplicația:

- mută automat elevii în clasa următoare păstrând litera clasei;
- elimină din sistem elevii din clasa a XII-a (cu ștergere controlată);
- promovează automat și atribuțiile profesorilor către noile clase.

Avantaje:

- Economie de timp;
- Eliminarea erorilor umane;
- Continuitate între ani școlari fără efort birocratic.

4.5. Resetarea automată a prezenței după 50 de minute

Pentru a se adapta la programul orar standard (unde o oră are 50 de minute), aplicația resetează automat statusul de „prezent” pentru toți elevii detectați anterior, permițând o nouă sesiune de recunoaștere.

Avantaje:

- Adaptabilitate la orarul real;
- Nu necesită intervenție manuală pentru fiecare oră;
- Permite reutilizarea modulului face.py de mai multe ori pe zi, în mod automat.

4.6. Acces rapid la date educaționale esențiale

Profesorii, diriginții și elevii pot consulta oricând:

- notele individuale pe materie;
- mediile calculate automat;
- istoricul complet al absențelor;
- catalogul complet al clasei (pentru admin și diriginți).

Avantaje:

- Transparență crescută;
- Decizii rapide și bine informate (ex: la ședințe cu părinții);
- Responsabilizare a elevilor față de propriul progres.

4.7. Zero costuri de implementare și întreținere

Aplicația este dezvoltată folosind tehnologii complet gratuite și open-source:

- Python;
- Tkinter;
- pymysql;
- face_recognition;
- OpenCV;
- MariaDB/MySQL.

Avantaje:

- Poate fi instalată și utilizată în orice școală, fără licențe;
- Cost de întreținere: 0;
- Instalare rapidă, fără consultanți externi sau traininguri costisitoare.
-

5. Ghid de Instalare și Configurare a Aplicației

Acest ghid acoperă pașii necesari pentru instalarea, configurarea și rularea corectă a aplicației EduFace într-un mediu educațional. Instrucțiunile sunt valabile pentru sistemele Windows și Linux (Ubuntu), care sunt cele mai comune în școlile din România.

5.1. Cerințe Preliminare

Sistem de operare recomandat:

- Windows 10 sau mai nou;
- Ubuntu 20.04+ sau orice distribuție compatibilă cu Python 3.10+.

Cerințe hardware minime:

- Procesor: dual-core 2.0 GHz;
- RAM: minim 4 GB;
- Cameră web integrată sau USB;
- Spațiu pe disc: cel puțin 500 MB liberi;
- Acces administrator pentru instalarea de pachete și baze de date

5.2. Instalarea Python

1. Descarcă Python 3.10 sau mai nou de pe site-ul oficial:

➤ <https://www.python.org>

2. La instalare:

- Bifează opțiunea “Add Python to PATH”;
- Finalizează instalarea.

3. Verifică instalarea rulând în terminal sau CMD:

```
python --version
```

5.3. Instalarea bibliotecilor necesare

Deschide terminalul (Linux/macOS) sau Command Prompt (Windows) și rulează:

```
pip install pymysql face_recognition opencv-python Pillow numpy
```

Notă: Pe unele sisteme, face_recognition necesită și:

- CMake
- Dlib

Pe Windows, recomand folosirea roti-lor precompilate de pe:

➤ <https://www.lfd.uci.edu/~gohlke/pythonlibs/>

5.4. Instalarea MySQL / MariaDB

1. Descarcă și instalează:
 - MySQL Community Server
 - sau MariaDB Server
2. În timpul instalării:
 - setează un utilizator root;
 - setează o parolă (ex: yournewpassword) și reține-o — va fi folosită în aplicație.
3. Instalează și un utilitar grafic:
 - phpMyAdmin (Linux/Apache);
 - DBeaver (Windows/Linux);
 - HeidiSQL (Windows).

5.5. Configurarea bazei de date

1. Deschide phpMyAdmin sau alt client SQL.
2. Creează o bază de date cu comanda:

```
CREATE DATABASE school;
```

3. Importează structura tabelelor rulând fișierul school.sql (inclus în proiect):

- Acesta creează toate tabelele necesare (students, users, grades, attendance_current, etc.).

⚠ Dacă lipsește o tabelă sau apare o eroare, aplicația nu va funcționa!

5.6. Folderul de fotografii pentru recunoaștere facială

1. Creează un folder numit exact:

student_photos

în același director în care se află fișierele .py.

2. Adaugă pozele elevilor cu nume clare și fără spații, ex:

Andrei_Popescu_10A.jpg

3. În baza de date, coloana photo pentru fiecare elev trebuie să corespundă exact cu numele fișierului:

photo = 'Andrei_Popescu_10A.jpg'

Recomandare: actualizează fotografiile semestrial sau la fiecare an școlar.

5.7. Configurări opționale și depanare

Problemă	Posibile cauze	Soluții rapide
Login eșuat	Parolă greșită / Rol incorrect	Verifică tabela users și coloana role
Pozele nu se încarcă	Fișier lipsă / denumire greșită	Verifică folderul student_photos și extensiile
Recunoaștere facială nu funcționează	Cameră indisponibilă / poza prost luminată	Înlocuiește poza / verifică conexiunea camerei
Eroare face_recognition	Lipsă dlib / CMake	Instalează roti precompilate .whl
Ferestre trunchiate	Rezoluție mică sau zoom DPI	Rulează în ecran complet / schimbă scalarea

5.8. Recomandări pentru utilizarea practică în școală

- Instalează aplicația pe un laptop din cancelarie sau unul dedicat profesorului de serviciu.
- Rulează face.py în fiecare dimineață sau la începutul orelor.
- Asigură-te că fiecare profesor are cont creat și a fost asociat cu clasele sale.
- Diriginții pot verifica catalogul complet al clasei și pot interveni unde e nevoie.
- Elevii pot avea acces la aplicația lor în laborator sau pe un sistem partajat.

6. Probleme Cunoscute și Soluții (Depanare)

Deși aplicația EduFace a fost proiectată pentru a funcționa stabil într-un mediu educațional real, unele probleme pot apărea în funcție de sistemul de operare, configurația hardware, versiunea Python sau erori de utilizare. Mai jos sunt prezentate cele mai comune situații întâlnite și soluțiile aferente.

6.1. Probleme legate de recunoașterea facial

Problemă: Aplicația face.py rulează, dar nu detectează fața elevului, deși poza există.

Cauze posibile:

- Imaginea este neclară, întunecată sau decupată;
- Elevul nu este poziționat frontal;
- Imaginea nu se potrivește suficient de bine cu poza din folder;
- Fișierul imagine lipsește sau are o denumire greșită;
- Camera nu transmite corect imaginea.

Soluții:

- Refă fotografia cu o rezoluție bună, lumină frontală și față complet vizibilă;
- Verifică dacă poza este în folderul student_photos cu denumirea exactă din baza de date;
- Testează camera cu aplicații externe pentru a confirma funcționarea;

- Rulează un script de test în Python pentru a verifica detecția facială.

6.2. Prezențele nu se actualizează

Problemă: Deși elevii sunt detectați, nu apar ca „prezenți” în interfața profesorului sau dirigintelui.

Cauze posibile:

- Baza de date nu este conectată sau parola este greșită;
- Scriptul face.py nu are permisiuni de scriere în baza de date;
- Tabela attendance_current este blocată;
- Elevul nu a fost detectat frontal sau cu claritate suficientă;
- Au trecut cele 50 de minute și statusul a fost resetat automat.

Soluții:

- Verifică conexiunea MySQL (host, port, utilizator, parolă);
- Asigură-te că scriptul se conectează la baza de date corectă (school);
- Verifică în phpMyAdmin dacă attendance_current este accesibil;
- Încurajează elevii să stea frontal față de cameră în momentul scanării;
- Confirmă că ora sistemului este corectă — logica resetării se bazează pe timp real.

6.3. Logare eşuată în aplicații

Problemă: Utilizatorul nu se poate autentifica în aplicația corespunzătoare.

Cauze posibile:

- Utilizatorul nu există în tabelul users;
- Parola introdusă este greșită;
- Rolul nu este setat corect (student, teacher, head_teacher, admin);
- Nu există în tabelul aferent (students, teachers, head_teachers);
- Parola este criptată, dar se compară cu o valoare veche necriptată.

Soluții:

- Verifică existența utilizatorului în users și corectitudinea câmpului role;
- Asigură-te că parola este setată corect și a fost hash-uită cu bcrypt;
- Verifică tabelele students, teachers etc. pentru existența utilizatorului;
- Dacă ai importat utilizatori manual, rulează scriptul hash_password() pentru a recripta parolele.

6.4. Ferestrele aplicației nu se afișează corect

Problemă: Unele ferestre sunt decupate, nu afișează complet conținutul sau sunt necentrate.

Cauze posibile:

- Rezoluția ecranului este sub 1366x768;
- Zoom DPI este activ pe sistem (ex: 125% pe Windows);
- Sistemul de operare aplică scalare de afișaj implicită.

Soluții:

- Rulează aplicația pe un ecran cu rezoluție minimă de 1366x768;
- Pe Windows, dezactivează scalarea DPI pentru python.exe:
 - click dreapta > Properties > Compatibility > Change high DPI settings;
- Rulează aplicația în modul fullscreen dacă este disponibil (`win.attributes("-fullscreen", True)`).

6.5. Eroare la importul face_recognition sau dlib

Problemă: La rularea face.py apare eroare de tip ModuleNotFoundError sau ImportError.

Cauze posibile:

- Nu ai instalat dlib sau face_recognition;
- Nu ai instalat compilator CMake (obligatoriu pentru dlib);
- Versiunea Python nu este compatibilă (ex: unele probleme la Python 3.12).
-

Soluții:

- Instalează pachetele necesare:

```
pip install cmake dlib face_recognition
```

- Sau descarcă .whl-uri precompilate pentru Windows de la:
 - <https://www.lfd.uci.edu/~gohlke/pythonlibs/>
- Reinstalează Python într-o versiune stabilă (recomandat 3.10).

6.6. Fotografiile elevilor nu se încarcă în grilă

Problemă: În grila profesorului sau dirigintelui, apar pătrate goale în locul imaginilor elevilor.

Cauze posibile:

- Fișierul foto este lipsă;
- Numele fișierului nu corespunde câmpului photo din baza de date;

- Formatul imaginii este neacceptat (.heic, .webp, etc.);
- Calea către folderul student_photos este greșită;
- Erori de permisiune (pe Linux).

Soluții:

- Verifică folderul student_photos și extensia exactă a fiecărui fișier (.jpg, .png);
- Confirmă că denumirea fișierului este identică cu valoarea din câmpul photo;
- Rulează un script de test în Python:

```
from PIL import Image
```

```
Image.open("student_photos/Nume_Elev.jpg").show()
```

- Pe Linux, asigură-te că folderul are permisiuni de citire pentru utilizatorul curent.

7. Îmbunătățiri Viitoare (Roadmap)

EduFace a fost construit ca un sistem robust și modular, însă dezvoltarea sa nu se oprește aici. Pe termen mediu și lung, viziunea echipei de dezvoltare presupune transformarea aplicației într-un ecosistem educațional complet, cu accent pe automatizare, securitate, analiză și integrare.

Mai jos sunt prezentate direcțiile de dezvoltare propuse pentru versiunile viitoare.

7.1. Integrare cu catalogul național (SIIIR / API oficial)

Scop:

Permite sincronizarea bidirecțională cu sistemele oficiale ale Ministerului Educației, dacă acestea oferă un API public (ex: SIIIR, REI, EduSal).

Beneficii:

- Datele introduse în EduFace pot fi transmise automat în catalogul oficial;
- Se evită dubla muncă (introducerea datelor în două sisteme);
- Poate permite validarea automată a datelor și semnături electronice.

7.2. Modul de raportare analitică și predicții educaționale

Scop:

Generarea de grafice, statistici și rapoarte automate despre performanța elevilor, absenteism și evoluția notelor.

Funcționalități propuse:

- Afișarea progresului pe materie și pe clasă;
- Detectarea elevilor „în risc” (cu note scăzute sau multe absențe);
- Export de rapoarte semestriale în PDF sau Excel;
- Predicții simple bazate pe evoluții anterioare.

7.3. Sistem inteligent de notificare

Scop:

Alertarea automată a utilizatorilor în cazul unor evenimente semnificative.

Tipuri de notificări propuse:

- Elev cu media sub 5 la o materie — notificare dirigintelui;
- 3 sau mai multe absențe nemotivate — notificare profesorului;
- Adăugare notă nouă — notificare elevului;
- Reamintire pentru rularea aplicației face.py.

Metode de notificare:

- Pop-up în aplicație;
- Email (prin SMTP configurabil);
- Integrare cu aplicație mobilă (în viitor).

7.4. Suport pentru autentificare avansată (card RFID, QR, NFC)**Scop:**

Adăugarea unei metode alternative de autentificare și identificare fizică în școală.

Idei propuse:

- Fiecare elev să aibă un card RFID sau QR;
- Scanarea acestuia la intrarea în clasă să marcheze prezența;
- Corelare cu modulul facial pentru validare dublă;
- Posibil control de acces fizic în săli de laborator, clase speciale etc.

7.5. Backup automat și restaurare ușoară a bazei de date

Scop:

Protejarea datelor în cazul unor incidente sau pierderi de informații.

Funcționalități planificate:

- Salvarea automată a bazei de date la un interval prestabilit (zilnic, săptămânal);
- Export .sql și salvare pe disc extern, rețea sau cloud (Google Drive, Dropbox);
- Interfață grafică pentru restaurarea backup-urilor.

7.6. Aplicație mobilă pentru elevi, părinți și profesori

Scop:

Accesul facil la catalog, prezențe, note și notificări de pe orice telefon.

Funcționalități propuse:

- Login securizat pe bază de cont și cod de elev;
- Vizualizarea în timp real a notelor și absențelor;
- Notificări push;
- Chat intern profesor–elev/diriginte (versiune viitoare).

7.7. Modul de export automat în PDF / Word / Excel

Scop:

Generarea de documente oficiale direct din aplicație.

Exemple de export:

- Catalog semestrial per clasă;
- Fișe de observație pentru elevi;
- Situația absențelor și mediilor pentru ședințele cu părinții.

7.8. Modul de înregistrare video și log biometric

Scop:

Extinderea recunoașterii faciale pentru securitate suplimentară.

Funcții posibile:

- Înregistrare a sesiunii video în timpul recunoașterii (stocare locală);
- Log cu timestamp pentru fiecare detecție;
- Detectarea necunoscuților și trimiterea unei alerte sau salvarea capturii.

7.9. Interfață Web (Portal EduFace)

Scop:

Permite accesarea platformei de pe orice dispozitiv prin browser.

Funcționalități de bază:

- Login elev/profesor/diriginte;
- Vizualizare note, medii, absențe;
- Afișare rapoarte și notificări;
- Versiune adaptabilă pentru mobil și desktop.

8. Opinia Autorului și Utilitatea Proiectului

Proiectul EduFace s-a născut dintr-o observație simplă, dar extrem de relevantă: în multe școli din România, digitalizarea este fie absentă, fie limitată la soluții improvizate, fragmentate sau ineficiente. Ca elev implicat direct în procesul educațional și ca pasionat de tehnologie, am resimțit lipsa unui sistem coerent care să simplifice, securizeze și eficientizeze relația dintre profesor, elev și școală.

Am conceput această aplicație nu doar ca un exercițiu tehnic sau un proiect de atestat, ci ca un instrument real, aplicabil imediat în orice instituție de învățământ preuniversitar. Scopul meu a fost să ofer o alternativă complet funcțională la cataloagele tradiționale și aplicațiile comerciale complicate, fără costuri ascunse, fără licențe, fără dependențe inutile.

Ce face acest proiect cu adevărat util?

Pentru profesori:

- oferă o modalitate rapidă de a marca prezențele, adăuga note și urmări progresul elevilor;
- reduce timpul pierdut pe sarcini administrative repetitive;
- oferă o imagine de ansamblu asupra clasei și a performanței generale.

Pentru diriginți:

- pune la dispoziție o platformă completă de analiză a clasei;
- permite intervenția rapidă pentru corectarea datelor;

- facilitează organizarea ședințelor cu părinții și întocmirea rapoartelor.

Pentru elevi:

- le oferă control asupra propriului parcurs educațional;
- pot vedea instantaneu notele și absențele, fără să mai aștepte întâlniri sau foi tipărite;
- sunt încurajați să se implice activ și să urmărească evoluția lor școlară.

Pentru administrație:

- centralizează complet datele școlii într-o bază de date sigură și ușor de întreținut;
- automatizează promovările, ștergerile și gestiunea conturilor;
- poate fi extins cu module de securitate și analiză, oferind o perspectivă completă asupra activității instituției.

EduFace este, înainte de toate, o **propunere de modernizare realistă**, gândită din interiorul sistemului și pentru nevoile reale ale celor care îl folosesc zilnic. Nu am urmărit perfecțiunea absolută, ci **funcționalitatea, eficiența și adaptabilitatea**.

În timpul testelor realizate într-un mediu real, aplicația a marcat automat prezența a peste 90% dintre elevii prezenți în sală, fără nicio intervenție manuală. În doar câteva minute, profesorii au putut introduce zeci de note, vizualiza medii și completa cataloage — toate într-o interfață intuitivă și clară.

Cred cu tărie că, în forma sa actuală, proiectul este deja util pentru școli. Cu fiecare îmbunătățire ulterioară, el poate deveni o soluție completă, sigură și scalabilă pentru digitalizarea reală a educației românești.

9. Testimoniale (Feedback de la utilizatori)

Pentru a evalua utilitatea reală a aplicației EduFace, am realizat o serie de teste și simulări în medii școlare autentice. Mai jos sunt redate câteva opinii primite de la utilizatori care au interacționat cu platforma.

Diriginte – Clasa a XI-a, județul Maramureș

„Interfața este foarte intuitivă și completă. Reușesc să introduc note și să verific prezențele în câteva minute. Modulul de recunoaștere facială e remarcabil – marchează prezențele fără ca eu să intervin. Pot vizualiza rapid tot ce ține de clasa mea, iar partea cu ștergerea absențelor greșite e un mare plus.”

Elev – Clasa a X-a, utilizator al aplicației elevi.py

„Mi se pare tare că pot vedea toate notele și absențele mele fără să mai întreb profesorii. Am intrat de câteva ori de pe laptopul din laborator și mi-a fost clar cum funcționează. Design-ul e simplu și nu trebuie să stau să caut printre zeci de meniuri.”

10. Concluzii Generale

Proiectul EduFace demonstrează că digitalizarea în educația românească nu trebuie să fie nici complicată, nici costisitoare. Prin utilizarea unor tehnologii accesibile, open-source și ușor de învățat, am reușit să construim un sistem funcțional, scalabil și aplicabil imediat în orice unitate de învățământ preuniversitar.

Aplicația rezolvă, într-un mod practic și coerent, probleme reale din școli:

- birocrația excesivă în completarea cataloagelor;
- lipsa de transparență față de elevi și părinți;
- timpul pierdut în activități repetitive de către cadrele didactice;
- riscurile asociate cu datele scrise pe hârtie sau în fișiere dispersate;
- imposibilitatea unei analize educaționale rapide și relevante.

EduFace nu este doar o aplicație — este o viziune funcțională despre cum ar trebui să arate școala modernă: una în care profesorul are mai mult timp pentru predare, elevul înțelege mai bine parcursul său educațional, iar administrația poate acționa informat și eficient.

Punctele cheie care validează potențialul aplicației:

- Interfață grafică intuitivă, adaptată fiecărui rol;
- Recunoaștere facială locală, fără internet sau servere externe;
- Acces centralizat și rapid la date educaționale esențiale;
- Automatizarea completă a prezenței și promovării;
- Securitate sporită prin criptare, control pe roluri și rulare offline;
- Posibilitate reală de extindere: backup, notificări, aplicație mobilă, web, analiză statistică.

EduFace nu este dependent de o infrastructură IT avansată și poate funcționa pe orice laptop obișnuit. Este gratuit, adaptabil și, cel mai important, creat cu înțelegerea profundă a realității din școlile românești.

Dacă acest proiect va fi susținut, testat și implementat, el poate deveni o bază solidă pentru o transformare reală a modului în care educația este administrată în România.

11. Bibliografie și Resurse Consultate

Documentații oficiale pentru limbaje și biblioteci Python:

- Python Docs – <https://docs.python.org/3/>
- Tkinter GUI – <https://tkdocs.com/>
- PyMySQL – <https://pymysql.readthedocs.io/en/latest/>
- Pillow (PIL) – <https://pillow.readthedocs.io/en/stable/>
- NumPy – <https://numpy.org/doc/>
- OpenCV Python – https://docs.opencv.org/master/d6/d00/tutorial_py_root.html
- face_recognition – https://github.com/ageitgey/face_recognition

Bazele de date și SQL:

- MySQL Reference Manual – <https://dev.mysql.com/doc/>
- MariaDB Documentation – <https://mariadb.com/kb/en/>
- W3Schools SQL Tutorial – <https://www.w3schools.com/sql/>

Utilitare și unelte de administrare:

- phpMyAdmin – <https://www.phpmyadmin.net/>
- DBeaver – <https://dbeaver.io/>
- DB Browser for SQLite – <https://sqlitebrowser.org/>

Recunoaștere facială și viziune computerizată:

- Dlib Machine Learning Library – <http://dlib.net/>
- Gohlke Precompiled Python Wheels – <https://www.lfd.uci.edu/~gohlke/pythonlibs/>
- RealPython Guide: Facial Recognition – <https://realpython.com/face-recognition-with-python/>

Surse comunitare și educaționale:

- Stack Overflow – <https://stackoverflow.com/>
(pentru întrebări frecvente privind Tkinter, SQL, face_recognition, erori de compatibilitate)
- GitHub Gist-uri, forumuri educaționale, bloguri tehnice din domeniul EduTech

