**UNIVERSITATEA TEHNICĂ DE CONSTRUCȚII BUCUREȘTI**

**FACULTATEA DE HIDROTEHNICĂ**

**SPECIALIZAREA: AUTOMATICĂ SI INFORMATICĂ APLICATĂ**

**Proiect la disciplina: Programarea calculatoarelor si limbaje de programare II**

**Titlu proiect: Sistem Gestionare Utilizatori în C++**

**Student:** Tudor Ion Cornel  
**AIA I Grupa 2**

**Anul universitar 2024 – 2025**

**Cuprins**

1. Introducere
2. Motivarea alegerii temei
3. Obiective
4. Tehnologii utilizate
5. Explicații cod sursă
6. Capturi de ecran
7. Bibliografie

**1. Introducere**

Proiectul constă în dezvoltarea unei aplicații de tip consolă, realizată în limbajul de programare C++, care gestionează un set de utilizatori. Programul permite adăugarea, afișarea, modificarea și ștergerea acestora, toate printr-un meniu interactiv.

Această aplicație simulează o componentă simplificată a unui sistem real de administrare a utilizatorilor, fiind întâlnită în numeroase aplicații moderne precum platforme de învățare, aplicații de business, sau sisteme de operare.

**2. Motivarea alegerii temei**

Am ales această temă deoarece este una dintre cele mai comune funcționalități în orice aplicație: gestionarea utilizatorilor.

Motivul principal a fost dorința de a înțelege fundamentele OOP (programare orientată pe obiecte) și de a aplica în practică lucrul cu vectori, clase, metode și meniuri interactive în consolă.

Tema este potrivită pentru nivelul meu actual și oferă o bază solidă pentru viitoare extensii ale proiectului, cum ar fi: salvare în fișiere, autentificare, sau crearea unei interfețe grafice.

**3. Obiective**

 Crearea unei clase Utilizator care conține atributele: id, nume, email.

 Implementarea unui meniu text cu opțiuni clare.

 Gestionarea unei liste dinamice de utilizatori cu ajutorul containerului vector.

 Aplicarea principiilor OOP: abstractizare, modularitate, reutilizare.

 Utilizarea funcției system("cls") pentru curățarea ecranului, și cin.ignore() pentru manipularea inputului corect.

**4. Tehnologii utilizate**

 **Limbaj:** C++

 **Container STL:** std::vector

 **Tip interfață:** Consolă (text-based)

 **Mediu de dezvoltare:** Visual Studio Code

 **Compilator:** g++ (MinGW sau altul)

 **Sistem de operare:** Windows 10/11

**5. Structura Aplicatiei**

Aplicația este împărțită în 3 mari secțiuni:

1. Definirea clasei Utilizator
2. Vectorul utilizatori și funcțiile principale
3. Funcția main() care gestionează meniul principal și controlul aplicației

**6. Descrierea detaliata a functiilor**

### **Clasa Utilizator**

class Utilizator {

public:

int id;

string nume;

string email;

Utilizator(int i, string n, string e) : id(i), nume(n), email(e) {}

};

### **Funcția adaugaUtilizator()**

Solicită de la utilizator ID-ul, numele și emailul, și creează un obiect Utilizator pe care îl adaugă în vector.

### **Funcția afiseazaUtilizatori()**

Parcurge vectorul și afișează fiecare utilizator într-un format ordonat.

### **Funcția modificaUtilizator()**

Caută utilizatorul după ID și actualizează informațiile introduse de utilizatorul aplicației

### **Funcția stergeUtilizator()**

Folosește un iterator pentru a identifica și elimina utilizatorul cu ID-ul dorit din vector.

### **Meniul principal**

Se află în main() și permite utilizatorului să navigheze între opțiuni:

1. Adaugă utilizator

2. Afișează utilizatori

3. Modifică utilizator

4. Șterge utilizator

0. Ieșire

**7. Interfata Aplicatiei**

Interfața este text-based, dar este structurată elegant cu separatoare și titluri:

===== SISTEM GESTIONARE UTILIZATORI =====

1. Adauga utilizator

2. Afiseaza utilizatori

3. Modifica utilizator

4. Sterge utilizator

0. Iesire

Se folosește system("cls") pentru curățarea ecranului și pauza() pentru oprirea temporară a execuției.

1. **Testare și scenarii de utilizare**

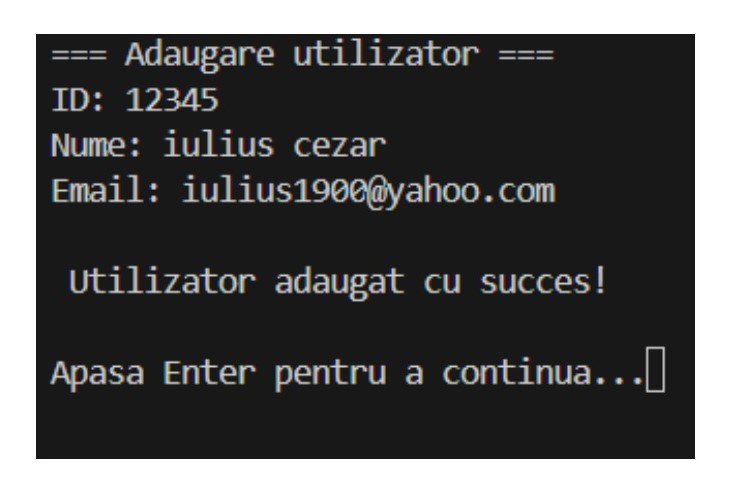
Au fost testate următoarele scenarii:

* Adăugarea mai multor utilizatori consecutiv
* Modificarea datelor unui utilizator existent
* Ștergerea unui utilizator inexistent (programul gestionează elegant eroarea)
* Afișarea unei liste goale (mesaj de avertizare)

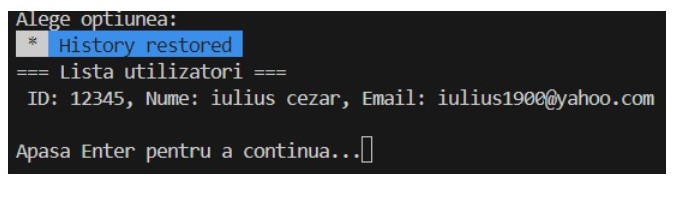
Programul s-a comportat corect în toate cazurile de test, fără erori de compilare sau execuție.

**9.Capturi de ecran**

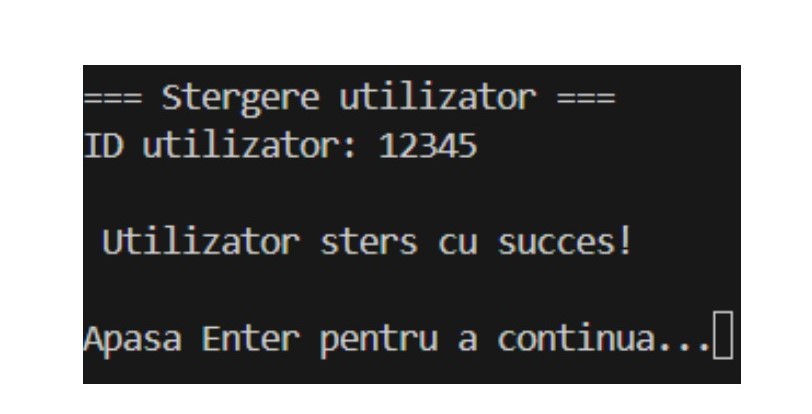
**Adaugare Utilizator**



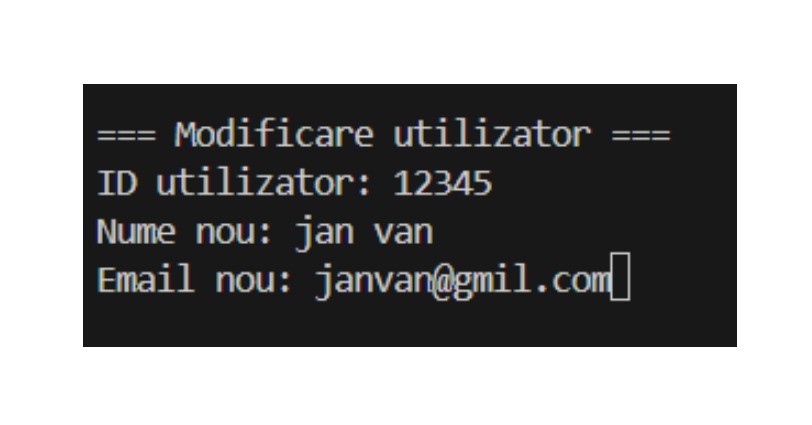
**Afisare Utilizator**



**Stergere Utilizator**



**Modificare Utilizator**



**10.Concluzii**

Acest proiect a reprezentat o ocazie excelentă de a exersa principiile OOP și de a învăța cum să creezi aplicații interactive în C++.

Am reușit să implementez un sistem funcțional, modular și intuitiv. Codul este ușor de întreținut și poate fi extins cu funcționalități noi.

Pe viitor, intenționez să adaug:

* Salvarea datelor în fișier .txt sau .csv
* Căutare după nume/email
* Autentificare și parole
* Interfață grafică (ex. cu SFML sau Qt)

**11.Bibliografie**

* [cplusplus.com](https://cplusplus.com) – Documentație oficială C++
* ChatGPT – pentru explicații și organizare a documentației
* Lecții de laborator și exemple discutate la curs
* Forumuri StackOverflow