**Proiect la PCLP 2**

Top of Form

Bottom of Form**Profesor Coordonator**  **Studenti**

Gabriela Olteanu Robert Aldea

**Aplicaţie pentru gestionarea stocurilor**

**Profesor Coordonator** **Student**

Gabriela Olteanu Robert Aldea

**CUPRINS**

**1.INTRODUCERE...............................................................................................................4-6**

1.1.Motivația alegerii temei.................................................................................................4

1.2.Evoluția pieței ...............................................................................................................5-6

1.3.Primii jucatori de pe piață..............................................................................................6

**2.TEHNOLOGII FOLOSITE..............................................................................................6-9**

2.1. C++...............................................................................................................................6-7

2.2.Python............................................................................................................................8-9

**3.SCENARIU DE FUNCTIONARE..................................................................................10-14**

3.1 Funcţia de salvare a unui produs....................................................................................11

3.2 Funcţia de editare a unui produs....................................................................................11-12

3.3 Funcţia de căutare a unui produs....................................................................................12-13

3.4 Funcţia de ştergere a unui produs....................................................................................13

3.5 Funcţia de afişare a unui produs......................................................................................14

**4.CONCLUZIE.......................................................................................................................14**

**5.BIBLIOGRAFIE..................................................................................................................15**

**1.1 Motivarea alegerii temei**

Am ales tema „Aplicație pentru gestionarea stocurilor unui magazin” deoarece reprezintă o problemă practică și des întâlnită în domeniul comercial, unde evidența eficientă a produselor este esențială pentru funcționarea corectă a oricărui magazin. Această temă oferă oportunitatea de a aplica conceptele învățate, precum lucrul cu structuri de date, fișiere, și funcții, într-un context real și util. În plus, dezvoltarea unei astfel de aplicații în C++ îmi consolidează abilitățile de programare și gestionare eficientă resurselor, ceea ce este important pentru performanța unei aplicații de tip desktop. Alegerea temei are, așadar, atât o relevanță practică, cât și o valoare educațională ridicată.

 Această temă permite, de asemenea, simularea unor procese esențiale din activitatea unui magazin, precum adăugarea, actualizarea, căutarea și ștergerea produselor din stoc, ceea ce presupune o bună organizare logică a codului și o gândire algoritmică eficientă. Prin implementarea acestui proiect, pot fi aprofundate noțiuni importante precum manipularea fișierelor pentru salvarea persistentă a datelor sau utilizarea structurilor condiționale și iterative în contexte relevante. Totodată, aplicația poate fi extinsă cu funcționalități suplimentare, cum ar fi evidențierea produselor cu stoc minim, sortarea automată sau generarea de rapoarte, ceea ce încurajează creativitatea și dezvoltarea continuă a soluției. În concluzie, tema aleasă este una complexă, dar accesibilă, și oferă un cadru excelent pentru aplicarea practică a limbajului C++ într-un mod util și realist.

Fig 1.1 Depozit

**1.2 Evoluția pieței de închirieri auto în ultimii ani**

Evoluția pieței de aplicații pentru gestionarea stocurilor unui magazin a cunoscut un ritm accelerat în ultimii ani, pe fondul digitalizării și al cererii tot mai mari pentru soluții eficiente și automatizate. De la simple aplicații desktop folosite doar pentru evidența produselor, s-a trecut la sisteme inteligente, integrate cu baze de date, funcții de analiză predictivă și interfețe intuitive. Tehnologiile moderne, precum cloud computing-ul și inteligența artificială, au transformat aceste aplicații în instrumente esențiale pentru optimizarea lanțului de aprovizionare, reducerea pierderilor și luarea deciziilor strategice. În plus, compatibilitatea multiplatformă și integrarea cu sisteme de vânzare, facturare sau logistică au crescut semnificativ valoarea acestor soluții. Această evoluție reflectă nevoia permanentă de eficiență și control în sectorul comercial, indiferent de dimensiunea magazinului.

**1.3 Primii jucatori de pe piata**

Primii jucători de pe piața aplicațiilor de gestionare a stocurilor au fost companii specializate în dezvoltarea de software pentru afaceri, precum SAP, Oracle și Microsoft, care au integrat funcționalități de management al inventarului în soluțiile lor ERP (Enterprise Resource Planning). Aceste companii au pus bazele unor sisteme robuste, destinate inițial marilor corporații, oferind funcții complexe de urmărire a stocurilor, automatizare a comenzilor și raportare avansată. Ulterior, odată cu dezvoltarea tehnologiei și apariția nevoilor din partea afacerilor mici și mijlocii, pe piață au apărut și alți furnizori importanți precum Zoho Inventory, TradeGecko (acum parte din QuickBooks), sau inFlow Inventory, care au adus soluții mai accesibile, cu interfețe prietenoase și funcționalități adaptate diverselor industrii. Acești pionieri au influențat semnificativ standardele actuale ale pieței, punând accent pe integrare, accesibilitate și scalabilitate.

# **A red square with white text and a cloud with a pie chart AI-generated content may be incorrect.**

**Fig 1.2 Oracle Management Cloud**

# **2.Tehnologii Folosite**

În cadrul dezvoltarii acestui proiect am utilizat predominant limbajul de programare C++. Interfaţa grafică am realizat-o cu ajutorul limbajului Python pe care l-am sincronizat cu funcţiile din limbajul C++ cu ajutorul unei file de tip .dll (Dynamic Link Library).

## **2.1 C++**

Limbajul de programare C++ este unul dintre cele mai influente și utilizate limbaje din istoria informaticii moderne. Acesta a fost creat la începutul anilor 1980 de către informaticianul danez **Bjarne Stroustrup**, în cadrul faimoaselor laboratoare **Bell Labs** din Statele Unite. Stroustrup a început să lucreze la acest limbaj ca o extensie a limbajului C, cu intenția de a adăuga facilități de programare orientată pe obiect, fără a compromite viteza și eficiența care făcuseră din C un limbaj atât de popular în dezvoltarea de sisteme. Inițial, limbajul s-a numit „C with Classes”, dar pe măsură ce a evoluat și a primit tot mai multe funcționalități, în 1983 a fost redenumit oficial **C++**, un nume care simbolizează îmbunătățirea sa față de C, prin folosirea operatorului de incrementare ++ din C.

C++ este un limbaj complex, care combină mai multe paradigme de programare. Deși păstrează stilul procedural caracteristic lui C, el adaugă suport pentru programarea orientată pe obiect, un model de gândire software care permite structurarea logică a codului în jurul entităților numite „obiecte”. Aceste obiecte sunt instanțieri ale unor clase, care pot conține atât date, cât și metode pentru manipularea acelor date. C++ oferă concepte esențiale precum **încapsularea**, **moștenirea** și **polimorfismul**, care permit dezvoltatorilor să creeze aplicații scalabile, reutilizabile și ușor de întreținut.

Un alt aspect definitoriu al limbajului C++ este controlul precis asupra resurselor de sistem. Programatorii pot gestiona manual memoria prin utilizarea pointerilor, ceea ce permite optimizarea fină a performanței, în special în aplicații critice cum ar fi jocurile video, software-ul industrial, sistemele embedded sau motoarele de baze de date. Totodată, limbajul permite utilizarea șabloanelor (templates), prin care se poate scrie cod generic, reutilizabil pentru mai multe tipuri de date, consolidând astfel principiile programării generice.

De-a lungul timpului, C++ a trecut prin mai multe standardizări. Prima versiune oficială standard a fost lansată în 1998 (C++98), urmată de actualizări semnificative precum C++03, C++11 (care a adus îmbunătățiri majore precum lambda expressions, auto, smart pointers), C++14, C++17 și C++20. Fiecare versiune a adăugat noi caracteristici care au modernizat limbajul, menținându-l competitiv și actual.

Astăzi, C++ rămâne un limbaj esențial în industrie și educație, fiind folosit în dezvoltarea de aplicații software critice, în inteligența artificială, în dezvoltarea sistemelor de operare, în programarea de jocuri și simulatoare, dar și în domeniul financiar, acolo unde performanța este vitală. De asemenea, numeroase alte limbaje moderne, precum Java, C#, sau Rust, au fost influențate într-o măsură mai mică sau mai mare de conceptele introduse sau popularizate de C++.

În concluzie, C++ este un limbaj care combină eficiența și puterea lui C cu structura și expresivitatea programării orientate pe obiect. Complexitatea sa îl face mai dificil de învățat pentru începători, însă versatilitatea, viteza și robustețea îl recomandă ca fiind un instrument indispensabil pentru orice programator care dorește să înțeleagă fundamentele dezvoltării software moderne.

A blue hexagon with white letters and white text

AI-generated content may be incorrect.

Fig 2.1 – Logo C++

**2.2 Python**

Python este un limbaj de programare de nivel înalt, interpretat și foarte popular datorită simplității și ușurinței în utilizare. A fost creat de către **Guido van Rossum** în anul 1989, în timp ce lucra la Centrul de Cercetare pentru Matematică și Informatică (CWI) din Olanda. Van Rossum și-a dorit să dezvolte un limbaj care să fie ușor de citit și scris, cu o sintaxă clară și intuitivă, care să permită programatorilor să se concentreze mai mult pe rezolvarea problemelor și mai puțin pe detaliile tehnice ale codului.

Python a fost lansat oficial în 1991 și, de atunci, a evoluat continuu prin numeroase versiuni și actualizări. Spre deosebire de limbaje precum C++ sau Java, Python pune accent pe lizibilitatea codului și folosește indentarea pentru a delimita blocurile de cod, ceea ce îl face deosebit de prietenos pentru începători. De asemenea, fiind un limbaj interpretat, programatorii pot testa rapid bucăți de cod fără a fi nevoie să compileze întregul program, facilitând dezvoltarea și depanarea.

Caracteristicile esențiale ale Python includ suportul pentru programarea procedurală, orientată pe obiect și funcțională, ceea ce îl face foarte flexibil pentru o gamă largă de aplicații. Python dispune de o bibliotecă standard foarte bogată, care oferă funcții și module pentru diverse domenii, de la manipularea fișierelor și lucrul cu rețele, până la procesarea datelor și dezvoltarea web. Această extensibilitate este completată de o comunitate largă și activă, care a dezvoltat numeroase pachete și framework-uri, cum ar fi Django pentru web, NumPy și Pandas pentru știința datelor, TensorFlow pentru inteligența artificială și multe altele.

Un alt avantaj major al Python este portabilitatea – programele scrise în Python pot rula pe aproape orice sistem de operare fără modificări semnificative ale codului. Aceasta, împreună cu natura sa open-source și cu faptul că este gratuit, a contribuit la răspândirea rapidă a limbajului în diverse domenii, inclusiv educație, cercetare, industrie și dezvoltare de software.

Python este utilizat pe scară largă în aplicații precum dezvoltarea de site-uri web, automatizarea sarcinilor repetitive, analiza datelor, machine learning, dezvoltarea de jocuri și chiar în robotică. Popularitatea sa a crescut exponențial în ultimii ani, fiind unul dintre cele mai cerute limbaje în industrie și educație.

În concluzie, Python este un limbaj modern, accesibil și extrem de versatil, ideal atât pentru începători, cât și pentru programatori experimentați. Simplitatea sa, combinată cu puterea și flexibilitatea oferite de ecosistemul său vast, îl fac un instrument valoros în dezvoltarea software contemporană.

A logo of a python

AI-generated content may be incorrect.

Fig 2.2 – Logo Python

# **3. Scenariu de functionare**

La pornirea aplicaţiei utilizatorul vede toate comenzile aplicaţiei precum si “baza de date” care contine toate produsele din depozitul magazinului.

A screenshot of a computer

AI-generated content may be incorrect.

Fig 3.1 – Pagina principală

**3.1 Funcţia de salvare a unui produs**

In partea de sus a paginii putem regăsii 3 cămpuri de tip “text input”. Odata completate cele 3 campuri putem folosi prima funcţie a aplicaţiei, funcţia „save\_product” care este apelată la apasarea butonului „Salvează”, aceasta funcţie deschide un fisier de text cu numele “products” în care scrie si salvează datele introduse de utilizator în cele 3 câmpuri.

A screen shot of a computer code

AI-generated content may be incorrect.

Fig 3.2 – Funcţia salvare

**3.2 Funcţia de editare a unui produs**

Imediat sub butonul “Salvează” regăsim butonul “Selectează pentru editare”. Odată ce am apasat pe un produs din lista de mai jos şi apăsăm pe acest buton, datele respectivului produs vor fi afişate în cele 3 câmpuri text din partea de sus a ecranului, loc unde datele produsului pot fi editate. După editare se apasă butonul “Salvează” iar datele produsului sunt modificate în baza de date, dar şi în aplicaţie în timp real.

A screenshot of a computer

AI-generated content may be incorrect.

Fig 3.3 – Interfaţa grafică pentru editare

Această funcţie oferă variabilei selected\_name numele produsului pe care utilizatorul doreşte să îl modifice astfel butonul “Salvează” apelează funcţia “update\_product”. Funcţia “update\_prduct” caută în fişierul “products.txt” produsul pe care utilizatorul doreşte să îl modifice, îl copiază în fişierul “temp.txt” alături de celelalte produse nemodificare, în final fisierul iniţial este şters iar cel nou este redenumit.

A computer screen shot of a program code

AI-generated content may be incorrect.

Fig 3.4 – Funcţia editare

**3.3 Funcţia de căutare a unui produs**

Pentru a utiliza funcţia de căutare a unui produs trebuie introdu in primul camp de tip text numele produsului cautat, apoi utilizatorul trebuie să apese butonul “Caută produs”. Butonul apelează funcţia „show\_product”, funcţie care parcurge “baza de date” rând cu rând până la găsirea produsului căutat. Dupa găsirea produsului pe ecran este afişat un mesaj cu detaliile produsului, în cazul în care produsul nu se află în baza de date, pe ecran este afişat un mesaj care informează utiliyatorul acest lucru.

A computer screen with colorful code

AI-generated content may be incorrect.

Fig 3.5 – Funcţia afişare

**3.4 Funcţia de ştergere a unui produs**

Pentru a utiliza funcţia de ştergere a unui produs utilizatorul trebuie să selecteze un produs din listă apoi să apese pe butonul “Ştergere produs selectat”, butonul apelează funcţia „delete\_product\_by\_name” care parcurge tot fişierul “products.txt” şi copiază in fişierul “temp.txt” toate produsele excluzând produsul pe care utilizatorul doreşte să îl şteargă, după parcurgerea întregului fişier, fişierul “products.txt” este şters iar fişierul “temp.txt” este redenumit in “products.txt”.

A screen shot of a computer program

AI-generated content may be incorrect.

Fig 3.6 – Funcţia ştergere

**3.5 Funcţia de afişare a produselor**

Funcţia de afişare a produselor este folosită de catre lista din centrul aplicaţiei, acesta foloseşte funcţia “read\_all\_products” care citeşte fişieril “products.txt”

A screen shot of a computer program

AI-generated content may be incorrect.

Fig 3.7 – Funcţia citire

**4.CONCLUZIE**

În concluzie, realizarea acestui proiect a reprezentat o oportunitate valoroasă de aplicare practică a cunoștințelor dobândite în cadrul materiei „Limbaje de programare”, utilizând limbajul C++ pentru dezvoltarea unei aplicații utile și relevante – un sistem de gestionare a stocurilor unui magazin. Tema aleasă a permis consolidarea unor concepte fundamentale precum manipularea fișierelor, lucrul cu structuri de date, dar și dezvoltarea gândirii algoritmice și a abilităților de organizare logică a codului. Prin abordarea unei probleme reale din sfera comerțului, proiectul a demonstrat importanța digitalizării în gestionarea eficientă a resurselor și a evidențiat rolul pe care limbajele de programare, precum C++, îl joacă în automatizarea și optimizarea proceselor. Totodată, cercetarea teoretică privind evoluția pieței aplicațiilor de acest tip și a limbajelor C++ și Python a oferit un context larg, care completează partea practică și subliniază relevanța acestui domeniu într-o societate tot mai orientată către tehnologie.

**5.BIBLIOGRAFIA**

1. https://en.wikipedia.org/wiki/C%2B%2B

2.https://en.wikipedia.org/wiki/Python\_(programming\_language)

3. https://gmdhsoftware.com/ro/inventory-management-software/

4. <https://stackoverflow.com/questions/61951238/how-to-load-a-c-c-dll-in-python>

5.Materiale universale primite la laboratoare și cursuri.