**1. Introducere în Java**

În acest proiect, am folosit intens System.out.println() pentru a afisa mesaje și pentru a interacționa cu utilizatorul prin consola. Este un element esențial pentru afisarea meniurilor, mesajelor de eroare sau succes și altele. De asemenea, am folosit cel mai frecvent două tipuri de variabile: int pentru a reprezenta valori numerice și String pentru gestionarea textului, ambele fiind fundamentale în implementarea logicii programului. Acestea au constituit baza majorității claselor.

**2. Introducerea în Java**

Input-ul este o componentă esențială în orice aplicație interactivă. În aplicația mea, acesta permite utilizatorilor să adauge date importante, precum nume, parole, emailuri, postări și altele. Am utilizat input-ul pentru a facilita procesul de înregistrare și interacțiune cu utilizatorii. Structurile de control repetitiv, precum buclele for și condițiile if, sunt folosite frecvent pentru a parcurge listele și a valida datele introduse. Acestea au fost esențiale pentru a asigura corectitudinea logicii și pentru a preveni erorile în program.

**3. Colecții Java**

Am folosit colecțiile List și ArrayList pentru a stoca și manipula utilizatori și postări. Metodele .contains() și alte metode de căutare mi-au permis să localizez rapid un utilizator sau o postare după anumite criterii (de exemplu, căutarea unui utilizator sau a unei postări după titlu). Am utilizat și metodele de adăugare pentru a salva utilizatorii și postările pe durata executării aplicației.

**4. Clase Java**

Am creat un număr considerabil de clase în Java, fiecare urmând principiile fundamentale ale orientării pe obiect. Aceste clase sunt stocate în pachetul models și sunt echipate cu metode relevante, precum getteri, setteri și constructori. Am implementat funcționalități care sunt specifice fiecărei clase, pentru a asigura o structură clară și modulară. Fiecare clasă a fost proiectată astfel încât să fie ușor de folosit și extins pe măsură ce cerințele aplicației evoluează.

**5. Moștenire în Java**

Am folosit moștenirea pentru a crea o ierarhie între utilizatori. Clasa abstractă User servește drept bază pentru clasele derivate StandardUser și AdminUser. Acestea împărtășesc un set comun de caracteristici și comportamente, dar fiecare are și comportamente specifice, precum drepturile administrative ale AdminUser. De asemenea, nu a fost necesară moștenirea altor clase, deoarece am creat variabile și metode la nivelul claselor deja existente.

**6. Interfețe în Java**

Am implementat patru interfețe pentru a organiza și a simplifica codul. Acestea sunt legate de postări, gestionarea fișierelor JSON, utilizatori și serviciile utilizatorilor. Interfețele au oferit un mod clar și modular de a organiza funcționalitățile aplicației. Implementarea acestora a făcut ca programarea să fie mai ușoară și mai bine structurată, fiind simplu să identific funcțiile relevante și să le folosesc fără a depinde de implementarea completă a claselor.

**7. Teste**

Testele au fost implementate pentru a mă asigura că aplicația nu conține erori grave care ar putea afecta performanța sau integritatea datelor. Am prioritizat testele în cazul funcționalităților esențiale, precum înregistrarea utilizatorilor sau salvarea postărilor, pentru a preveni posibile defecte care ar putea duce la pierderea datelor sau funcționarea defectuoasă a aplicației. Aceste teste sunt esențiale pentru menținerea unui standard ridicat de calitate.

**8. Persistența datelor**

Pentru persistenta datelor, am ales să folosesc fișiere JSON în locul unei baze de date. JSON este un format de date ușor de utilizat, portabil și acceptat de majoritatea limbajelor de programare. Fișierele JSON sunt rapide și simplu de citit, iar formatul lor asigură compatibilitate între diferite sisteme de operare. Am folosit trei fișiere JSON pentru a salva datele utilizatorilor: standard\_users.json, admin\_users.json și posts.json. Fiecare fișier conține informații specifice: standard\_users.json stochează utilizatorii care nu au privilegii administrative, admin\_users.json conține datele despre utilizatorii administratori, iar posts.json stochează postările, inclusiv comentariile și like-urile (cu protecția confidențialității datelor privind utilizatorii care dau like-uri).

9. Diagramele UML

Diagramele UML pe care le am facut din Intellij Idea sunt incluse in folderul diagrame uml din repo, dar si in acest document mai jos.





