



Universitatea Tehnică „Gheorghe Asachi” din Iași
Facultatea de Automatică și Calculatoare
Domeniul: Calculatoare și Tehnologia Informației
Specializarea: Tehnologia Informației
Disciplina: Extragerea Cunoștințelor din Baze de Date

Gestiunea sesiunilor de consiliere psihologică online

Cadru didactic coordonator

Ș.I.dr.ing. Cristian Aflori

Student
Popa Andrei, 1409B

An universitar 2024-2025

1. Descrierea proiectului

Proiectul realizat are ca obiectiv dezvoltarea unei aplicații pentru gestionarea activităților specifice consilierii psihologice în mediul online. Aplicația răspunde nevoilor atât ale pacienților care solicită suport psihologic, cât și ale terapeuților care desfășoară sesiuni de terapie, facilitând un cadru organizat, sigur și scalabil pentru interacțiune și analiză.

Platforma dezvoltată permite înregistrarea utilizatorilor în roluri diferite (terapeut sau pacient), stabilirea programărilor pentru sesiuni, desfășurarea acestora și colectarea de feedback la final. Datele sunt structurate în mod logic în cadrul unui sistem de baze de date MongoDB, fiind gândite pentru a permite operații avansate de interogare, filtrare, sortare și agregare.

Scopul final al proiectului este de a crea un ecosistem informatic coerent și extensibil care să poată fi utilizat atât în cadrul unor clinici de specialitate, cât și în contexte educaționale sau de cercetare, oferind o soluție viabilă pentru digitalizarea completă a procesului de consiliere psihologică. Aplicația este optimizată pentru analiza performanțelor terapeutice și pentru automatizarea unor activități administrative, menținând în același timp o structură ușor de extins și integrat cu alte servicii digitale.

2. Structura bazei de date

Pentru implementarea aplicației, au fost definite patru colecții principale în baza de date MongoDB: *users*, *appointments*, *sessions* și *feedbacks*. Fiecare colecție îndeplinește un rol bine definit și include atât câmpuri de tip scalar, cât și structuri complexe precum vectori sau subdocumente. Mai jos sunt descrise detaliat structurile fiecărei colecții și tipurile de date asociate:

➤ Colecția *users*

Această colecție gestionează informațiile despre toți utilizatorii platformei, atât terapeuți cât și pacienți; permite căutări în subdocumente, verificarea rolurilor și filtrarea pe orașe sau specializări.

- *user_id* (String) – identificator unic
- *name* (String) – numele complet al utilizatorului
- *email* (String) – adresa de email, unică în cadrul sistemului
- *role* (String) – poate fi „therapist” sau „patient”
- *specialties* (Array de String) – lista de competențe pentru terapeuți (opțional)
- *address* (Subdocument):
 - *street* (String)

- *city* (String)
- *zip* (String)
- *created_at* (Date) – data înregistrării utilizatorului

➤ **Colecția *appointments***

Această colecție conține programările planificate între terapeuți și pacienți; evidențiază istoria evoluției unei programări printr-un vector de subdocumente, fiind utilă pentru urmărirea schimbărilor de stare.

- *appointment_id* (String) – identificator unic
- *therapist_id, patient_id* (String) – referințe către users
- *date* (Date) – data și ora programării
- *notes* (String) – comentarii despre motivul consultului
- *history* (Array de Subdocumente):
 - *status* (String) – ex. „pending”, „confirmed”, „completed”
 - *date* (Date) – data la care s-a schimbat statusul

➤ **Colecția *sessions***

Această colecție include informații despre sesiunile efective de consiliere, permite analize statistice asupra conținutului și impactului emoțional al sesiunilor prin agregări și interogări multiple.

- *session_id* (String) – identificator unic
- *appointment_id* (String) – legătura către programarea corespunzătoare
- *duration_min* (Int) – durata sesiunii în minute
- *topics_discussed* (Array de String) – teme discutate în cadrul sesiunii
- *topic_details* (Array de Subdocumente):
 - *topic* (String)
 - *depth* (String) – nivelul de detaliu abordat
- *mood_score* (Double) – scor emoțional evaluat
- *summary* (String) – rezumat text al sesiuni

➤ **Colecția *feedbacks***

Această colecție stochează opiniile pacienților legate de sesiunile desfășurate, fiind baza pentru calculul ratingului mediu pe terapeut și analiza calitativă a interacțiunii.

- *feedback_id* (String) – identificator unic
- *session_id* (String) – legătura către sesiunea evaluată
- *rating* (Int) – notă acordată (1–5)
- *comment* (String) – opinie textuală
- *submitted_at* (Date) – data trimiterii formularului

Structura acestor colecții a fost construită pentru a permite atât operații CRUD uzuale, cât și interogări avansate, corelații multiple și procesări complexe ale datelor prin pipeline-uri agregate.

3. Descrierea funcționalităților & Operații CRUD

Platforma a fost proiectată pentru a permite operații complete asupra colecțiilor, acoperind toate funcționalitățile cerute de specificația temei. Acestea includ atât operații CRUD (create, read, update, delete), cât și interogări complexe de tip agregat, filtrări, căutări avansate și relaționări între colecții.

• **Operații de inserare (*insertOne*, *insertMany*)**

Pentru toate cele patru colecții (*users*, *appointments*, *sessions*, *feedbacks*) au fost dezvoltate scripturi pentru inserări individuale și multiple. Datele sunt realiste și corelate, permițând rularea interogărilor ulterioare. Inserările includ atât documente simple, cât și structuri complexe cu vectori și subdocumente, reflectând scenarii reale de utilizare.

• **Operații de actualizare (*updateOne*, *updateMany*)**

Scripturile de actualizare au fost concepute progresiv: de la modificarea unui singur câmp (*\$set*) până la actualizarea elementelor din vectori de subdocumente utilizând *arrayFilters*. De exemplu, actualizarea statusului din istoric într-o programare sau modificarea unui anumit *depth* din *topic_details*. De asemenea, s-a folosit operatorul *\$push* pentru extinderea vectorilor.

• **Operații de ștergere (*deleteOne*, *deleteMany*)**

Au fost implementate atât ștergeri individuale, cât și multiple, cu filtre care evită ștergerea accidentală a datelor importante. Printre exemple se numără

ștergerea pacienților dintr-un anumit oraș, ștergerea sesiunilor cu scoruri mici sau a feedbackurilor fără dată de trimitere. Aceste ștergeri ajută la curățarea și normalizarea bazei de date.

- **Căutări simple (find, sort, limit, projection)**

Au fost dezvoltate scripturi pentru căutări după atribute directe, în vectori sau subdocumente. De exemplu, căutarea tuturor pacienților dintr-un anumit oraș, a terapeuților specializați pe o temă sau a sesiunilor în funcție de mood_score. Căutările sunt optimizate cu proiecții și sortări relevante.

- **Căutări paginate (find + skip + limit)**

Pentru a gestiona volume mari de date, au fost implementate interogări cu paginare. Acestea sunt utile în afișarea controlată a informațiilor despre utilizatori, programări sau feedbackuri în interfețe paginabile. Paginarea este aplicată și pe rezultate sortate.

- **Căutări complexe cu cursori (find + while(cursor.hasNext()))**

S-au implementat scripturi care procesează rezultatele cu ajutorul cursorilor pentru afișări controlate și formate. De exemplu, afișarea tuturor pacienților cu adresă și dată înscrierii, afișarea terapeuților cu toate specializările sau listarea sesiunilor cu scor emoțional peste o anumită valoare.

- **Agregări complexe (aggregate)**

Scripturile folosesc operatori precum \$group, \$project, \$unwind, \$round, \$limit pentru a agrega date semnificative. De exemplu: calculul mediei mood_score în funcție de durata sesiunii, gruparea feedbackurilor după rating cu comentarii sau distribuția topicurilor discutate în sesiuni.

- **Relații între colecții (\$lookup)**

Un punct forte este integrarea \$lookup pentru a corela date din colecții diferite. De exemplu, afișarea scorului emoțional al unei sesiuni pe baza session_id din feedbacks, asocierea sesiunilor cu pacienți și terapeuți prin appointments sau afișarea numelui terapeutului și a ratingului mediu obținut de la pacienți.

Toate operațiile sunt susținute de scripturi separate și testate, care pot fi rulate secvențial. Structura modulară permite scalarea și extinderea aplicației cu noi funcționalități.