PLATFORMĂ DE STUDIU

Cuprins

1. Introducere

Proiectul presupune implementarea unui sistem informatic destinat gestiunii unei platforme de studiu.

Aplicația folosește un sistem de gestiune pentru baze de date MySQL, iar interacțiunea cu aceasta este realizată doar prin interfața grafică. Funcționalitățile pe care le oferă aplicația vizează operații ce țin de gestiunea studenților, profesorilor și administrarea operațiilor curente din cadrul unor programe de studiu.

Pentru dezvoltarea proiectului au fost folosite:

- **MySQL Workbench** - pentru crearea bazei de date, popularea inițială, dezvoltarea de view-uri, proceduri și triggere și pentru crearea diagramei UML a tabelelor.
- **IntelliJ** – mediu de dezvoltare Java.
- **MySQL Connector Java** - este un driver JDBC (Java Database Connectivity) oferit de Oracle, care permite aplicațiilor Java să se conecteze și să interacționeze cu baze de date MySQL. Acesta facilitează execuția de interogări SQL, manipularea datelor și gestionarea conexiunilor între aplicația Java și serverul MySQL.
- **JavaFX** - framework utilizat pentru dezvoltarea interfeței grafice (GUI) în Java.

2. Cerințe și decizii de implementare

a) Cerințele proiectului:

Aplicația va putea fi accesată, pe baza unui proces de autentificare, de către mai multe tipuri de utilizatori: studenți, profesori, administratori. Pentru fiecare tip de utilizator se vor reține informații precum CNP, nume, prenume, adresă, număr de telefon, email, cont IBAN, numărul de contract. Fiecare utilizator își va putea vizualiza datele personale imediat după ce va accesa sistemul informatic, fără a avea însă posibilitatea de a le modifica. Totodată, programul trebuie să ofere și o funcționalitate pentru deautentificare, prin care se revine la fereastra care solicită datele de acces, astfel încât și un alt utilizator să îl poată folosi ulterior, fără a fi necesară repornirea sa.

Utilizatorul de tip administrator poate adăuga, modifica și șterge informații în baza de date, informații legate de utilizatori. De asemenea, va exista și un rol de super-administrator care poate opera inclusiv asupra utilizatorilor de tip administrator.

Administratorii pot să caute utilizatorii după nume și îi pot filtra după tip, pot asigna profesorii la cursuri și pot face căutări după numele cursului. La căutarea unui curs se afișează și numele profesorilor de la acel curs și un buton care permite vizualizarea tuturor studenților înscriși la cursul respectiv.

Pentru un utilizator de tip profesor se vor reține și cursurile predate, numărul minim și numărul maxim de ore pe care le poate preda și departamentul din care face parte.

Pentru un utilizator de tip student se va reține și anul de studiu și numărul de ore pe care trebuie să le susțină.

Aplicația va permite gestiunea cu ușurință a activităților didactice și astfel a interacțiunilor dintre studenți și profesori. Cursurile sunt predate de mai mulți profesori și au una sau mai multe tipuri de activități (curs, seminar, laborator), o descriere și un număr maxim de studenți participanți. Studenții se pot înscrie la cursuri și sunt asignați profesorului cu cei mai puțini studenți la data înscrierii. Aceștia sunt evaluați cu note pentru fiecare tip de activitate și primesc o notă finală ca medie ponderată între tipurile de activități. Profesorul stabilește din interfața grafică împărțirea procentuală pe tipurile de activități (ex. 20% seminar, 35% laborator, 45% curs/examenul de la curs).

Fiecare activitate se desfășoară recursiv între două date, pe o anumită perioadă de timp. La asignarea unui profesor la un curs se vor alege tipurile de activități. De exemplu, profesorul X predă cursul Y cu activitățile: curs – săptămânal, laborator – săptămânal. Ulterior, profesorul poate programa activitățile (curs, seminar, laborator, colocviu, examen) într-un calendar, pe zile și ore, specificând și numărul maxim de participanți. Activitățile pot fi programate doar în viitor. Profesorii pot accesa un catalog, unde pot filtra studenții după cursuri și le pot adăuga note.

La logare, studenții și profesorii pot să își vadă activitățile din ziua curentă sau pot accesa o pagină cu toate activitățile la care sunt asignați/înscriși.

Studenții se pot înscrie la cursuri, pot renunța la cursuri și își pot vedea notele. Aceștia trebuie să aleagă activitățile la care vor să participe și pot participa la ele

doar dacă mai sunt locuri disponibile sau nu există o suprapunere cu o altă activitate (de exemplu, studentul dorește să participe la laboratorul de BD marți la ora 10. Se înscrie la acea activitate, iar înscrierea este validă doar dacă nu are deja o altă activitate marți la ora 10 sau dacă mai sunt locuri disponibile. În caz contrar, se afișează un mesaj de eroare).

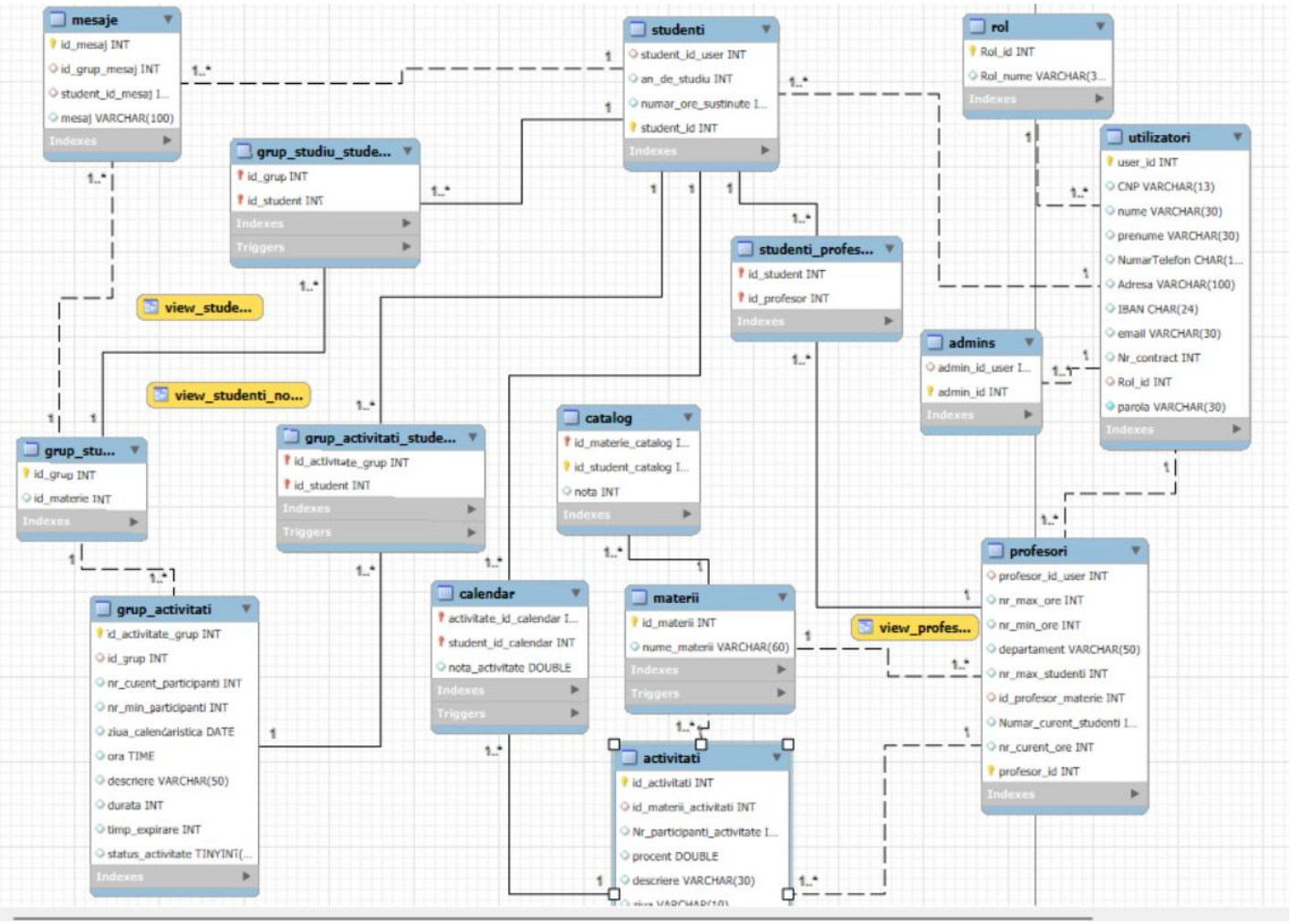
Totodată, studenții se pot înscrie în grupuri de studiu pentru o anumită materie, dacă sunt înscriși la materia respectivă. Aceștia pot să vadă toți membrii grupului și să lase mesaje. Pe grup, studenții pot adăuga activități și să definească un număr minim de participanți și o perioadă în care ceilalți studenți pot să anunțe participarea (de exemplu, un student adaugă o activitate de aprofundare a cursului pentru data de 12.12.2020, ora 16:00, 2 ore, cu număr minim de participanți 5 și timp de expirare 2 ore). Dacă numărul minim nu este atins, activitatea se anulează, iar studenții înscriși la ea primesc un mesaj de informare.

b) Decizii de implementare:

- un profesor predă doar o materie
- fiecare profesor are în orar toate tipurile de activități (curs, seminar, laborator)
- există un singur grup de studiu per materie
- fiecare student poate participa la câte un curs, seminar, laborator per materie
- dacă un student a contractat o materie, acesta va avea un loc asigurat la fiecare tip de activitate din cadrul acelei materii (curs, seminar, laborator)

3. Modelul de date

a) Diagrama UML pentru modelul de date complet



b) Entități și atributele lor

Tabele:

- **Utilizatori** – informațiile personale ale utilizatorilor: CNP, nume, prenume, număr de telefon etc.
- **Rol** – fiecare rol este identificat printr-un număr (Rol_id) și numele rolului (Rol_nume).

- **Profesori** – conține informațiile utilizatorilor de tipul “profesor”: numărul maxim și minim de ore predate, departamentul, id-ul materiei predate etc.
- **Studenți** – conține informațiile utilizatorilor de tipul “student”: anul de studiu și numărul de ore pe care trebuie să le susțină
- **Admins** – conține informațiile utilizatorilor de tipul “administrator” și “super-administrator”.
- **Studenți_profesori** – perechile student-profesor (fiecare student este înscris la câte un profesor pentru fiecare materie pe care a contractat-o)
- **Catalogul** – acesta conține notele studenților la toate materiile pe care le-au contractat
- **Materii** - conține materiile care sunt predate în cadrul facultății. Acestea sunt identificate printr-un număr unic (id_materii) și prin nume (nume_materii).
- **Activități** – conține toate activitățile (cursuri, seminarii, laboratoare) programate de către profesori, cu detaliile necesare: id_ul materiei de care aparține activitatea, id-ul profesorului care predă activitatea, numărul curent și numărul maxim de participanți, ziua și ora în care este susținută, precum și procentul notei de la activitatea respectivă (ponderea în calculul notei finale). Studenții trebuie să se înscrie la aceste activități.
- **Calendar** – fiecărui student îi este asociat un calendar (orarul studentului) în care sunt reținute id_urile activităților la care participă, precum și nota la aceste activități (aceasta va fi setată de către profesor). Notele corespunzătoare activităților vor fi folosite pentru calcularea notei finale din catalog.
- **Grup_studiu** – pentru fiecare materie predată va exista un grup de studiu la care studenții se pot înscrie dacă doresc să-și aprofundeze cunoștințele.
- **Grup_studiu_studenți** – aici sunt salvate id-urile studenților și grupurile de studiu la care s-au înscris (perechi student – grup_studiu).
- **Mesaje** – în cadrul unui grup de studiu, studenții pot lăsa mesaje. Acestea se vor salva în tabelul “mesaje” și vor fi vizibile doar pentru studenții înscriși la grupul de studiu corespunzător.
- **Grup_activități** – conține activitățile adăugate de studenți în cadrul unui grup de studiu, cu detaliile necesare. După trecerea timpului de expirare, dacă numărul minim de studenți înscriși nu este atins, activitatea se anulează și nu mai este valabilă (status_activitate devine 0).
- **Grup_activități_studenți** – aici sunt salvate id-urile studenților și activitățile din cadrul grupurilor de studiu la care s-au înscris.

c) Vederi:

- view_profesor
- view_studenti
- view_studenti_note

d) Proceduri:

- **'AdaugaActivitateGrup'** – adaugă o activitate în cadrul unui grup de studiu
- **'AdaugaAdmin'** – adaugă un utilizator de tip “administrator”
- **'AdaugaProfesor'** - adaugă un utilizator de tip “profesor”
- **'AdaugaStudent'** - adaugă un utilizator de tip “student”
- **'AfisareGrupuriDeStudiu'** – afișează grupurile de studiu la care este înscris un student
- **'GetActivitatiStudent'** – afișează toate activitățile la care participă studentul într-o zi a săptămânii
- **'GetAdminData'** – afișează datele unui utilizator de tip “administrator”
- **'GetMateriiProfesor'** – procedură auxiliară pentru funcționalitatea interfeței grafice
- **'GetProfesorData'** - afișează datele unui utilizator de tip “profesor”
- **'GetStudentData'** - afișează datele unui utilizator de tip “student”
- **'GetStudentNote'** – afișează notele unui student
- **'InscriereActivitate'** – înscrierea unui student la o activitate programată de profesor
- **'InscriereActivitateGrup'** - înscrierea unui student la o activitate din cadrul unui grup de studiu
- **'InscriereGrupStudiu'** - înscrierea unui student la un grup de studiu
- **'ProfesorProgramareActivitate'** – programarea unei activități de către un profesor
- **'SelectActivitatiProfesor'** – afișează activitățile predate de către un profesor

- **'SelecteazaMesajeGrup'** – afișează mesajele trimise în cadrul unui grup de studiu (asemănător unui chat)
- **'SelecteazaProfesor'** – afișează profesorul de la o anumită materie la care studentul este înscris
- **'SelecteazaStudentiProfesor'** – afișează toți studenții care sunt înscriși la un anumit profesor
- **'SetareNoteStudent'** – profesorul adaugă notele unui student pentru fiecare tip de activitate (curs, seminar, laborator)
- **'SeteazaProcenteActivitatiProfesor'** – profesorul setează procentul fiecărui tip de activitate în cadrul notei finale
- **'StudentInscriereMaterie'** – înscrierea unui student la o materie
- **'StudentRenuntareMaterie'** – studentul renunță la o materie pe care a contractat-o anterior (sunt șterse toate detaliile studentului din câmpurile legate de materia respectivă)
- **'TrimiteMesaj'** – studentul trimite un mesaj în cadrul unui grup de studiu la care este înscris
- **'VeziActivitatiGrup'** – afișează toate activitățile din cadrul unui grup de studiu
- **'VeziMembriGrup'** – afișează toți studenții înscriși la un grup de studiu
- **'VeziProfesoriMaterie'** – afișează toți profesorii care predau o anumită materie
- **'VeziStudentiMaterie'** – afișează toți studenții înscriși la o anumită materie.

e) Triggere:

- **'trg_before_insert_activitati'** – verifică dacă profesorul mai are suficiente ore disponibile înainte de programarea unei activități
- **'check_activity_enrollment'** – verifică dacă studentul se poate înscrie la o anumită activitate (verifică dacă există locuri disponibile pentru activitatea respectivă și verifică dacă studentul are alte activități care se suprapun)
- **'trg_insert_activitate_student'** – verifică dacă studentul face parte din grupul de studiu asociat activității la care dorește să se înscrie
- **'before_grup_studiu_student_insert'** – verifică dacă studentul a contractat materia asociată grupului de studiu la care dorește să se înscrie

- **'after_insert_materii'** – după ce s-a adăugat o nouă materie în tabelul "materii", se creează automat un grup de studiu asociat acesteia
- **'TriggerRecalculeazaNoteFinala'** – dacă profesorul schimbă procentul activităților pe care le predă, se recalculează notele finale din catalog ale tuturor studenților înscriși la activitățile respective

f) Event (scris în interfața grafică):

```
"CREATE EVENT IF NOT EXISTS " + eventName +
      " ON SCHEDULE AT TIMESTAMPADD(MINUTE, ?,
CURRENT_TIMESTAMP) " +
      " DO BEGIN " +
      "   IF (SELECT nr_curent_participanti FROM grup_activitati WHERE
id_activitate_grup = ?) < " +
      "   (SELECT nr_min_participanti FROM grup_activitati WHERE
id_activitate_grup = ?) THEN " +
      "     UPDATE grup_activitati SET status_activitate = 0 WHERE
id_activitate_grup = ?; " +
      "   ELSE " +
      "     UPDATE grup_activitati SET status_activitate = 2 WHERE
id_activitate_grup = ?; " +
      "   END IF; " +
      " END;";
```

Acest event calculează numărul de participanți după ce a trecut timpul necesar înscrierii studenților la activitatea din cadrul grupului de studiu. Dacă numărul minim de participanți nu este atins, această activitate se anulează (status_activitate = 0). În caz contrar, activitatea rămâne valabilă și se va organiza (status_activitate=2), dar nu se vor mai putea înscrie alți studenți, deoarece timpul necesar înscrierii a expirat.

status_activitate = 0 => activitatea s-a anulat și nu mai este valabilă

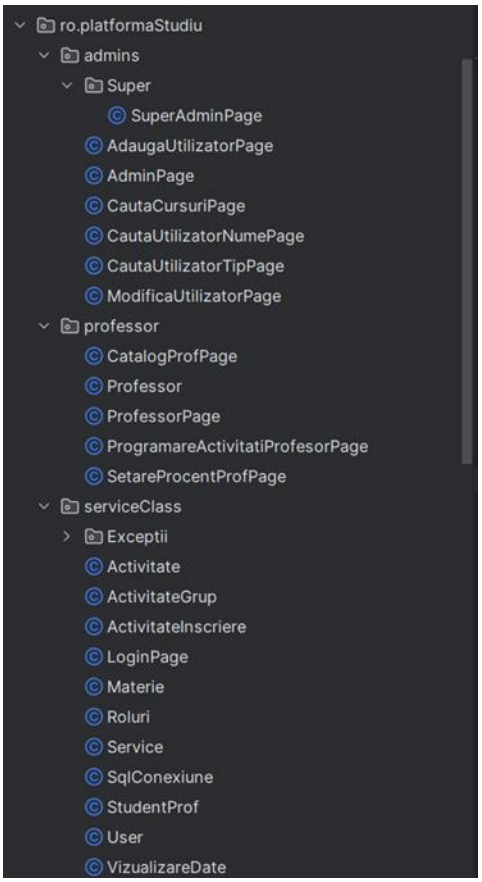
status_activitate = 1 => activitatea este activă și studenții se pot înscrie la aceasta

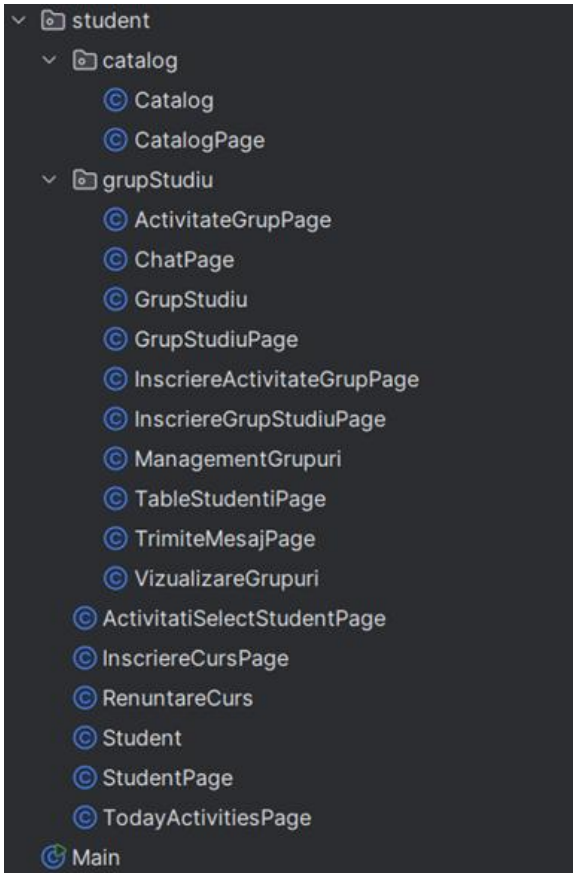
status_activitate = 2 => activitatea se va organiza, dar timpul necesar înscrierii a expirat.

4. Normalizarea datelor

Baza de date respectă forma normală Boyce-Codd (BCNF). Fiecare tabel are o singură cheie primară după care sunt identificate înregistrările și este suficientă pentru a identifica în mod unic orice înregistrare din baza de date. De exemplu, pentru tabela “utilizatori” avem cheia primară “user_id” și toate dependențele au în stânga această supercheie.

5. Descrierea structurală a proiectului Java (pachete):





Clasele folosite pentru implementarea interfeței grafice sunt împărțite în pachete, conform funcționalității acestora.

În pachetul “admins”, sunt grupate clasele care implementează metodele pentru cerințele administratorilor. Exemplu: căutarea utilizatorilor după tip sau căutarea studenților după cursuri.

Pachetul “professor” este asemănător și conține clasele necesare pentru cerințele profesorilor: vizualizare catalog, programare activități etc.

În pachetul “serviceClass” sunt prezente clasele necesare pentru funcționare, ex: conexiunea cu baza de date, login etc. Unele tabele din baza de date sunt transformate în clase pentru a implementa strategia de programare orientată pe obiecte.

Pachetul “student” conține toate clasele necesare cerințelor unui utilizator de acest tip. Ex: înscrierea la materii și la activități etc.

6. Manual de utilizare

La lansarea în execuție a aplicației, utilizatorul este întâmpinat de fereastra de login. Pentru a continua, acesta trebuie să introducă datele de logare: adresa de email și parola.

Ulterior, se va deschide pagina corespunzătoare rolului utilizatorului (student, profesor, administrator sau super-administrator). Fereastra permite efectuarea diverselor operații în cadrul platformei de studiu. Studenții se pot înscrie la materii și la activități. Profesorii pot să programeze diferite activități (curs, seminar, laborator) și pot nota studenții. Administratorii și super-administratorul pot adăuga, modifica și șterge informații în baza de date, pot căuta utilizatorii după nume și tip.

Programul oferă și o funcționalitate pentru deautentificare, prin care se revine la fereastra care solicită datele de acces, astfel încât și un alt utilizator să îl poată folosi ulterior, fără a fi necesară repornirea sa.

Interfața grafică permite realizarea tuturor acestor operațiuni cu ușurință. De asemenea, design-ul este unul prietenos, care faciliteaza gestiunea datelor.

7. Concluzii, limitări și dezvoltări ulterioare

Toate cerințele au fost respectate și îndeplinite, acestea ducând la un mediu plăcut prin care utilizatorii pot interacționa cu platforma de studiu.

a) Limitări:

- **Scalabilitate**: Deși aplicația este funcțională pentru un număr limitat de utilizatori și cursuri, performanța ar putea fi afectată pe măsură ce numărul de studenți și activități crește semnificativ. Ar fi necesar un plan de optimizare a performanței pentru a face față unor volume mari de date.
- **Autentificare și securitate**: În prezent, procesul de autentificare se bazează pe email și parolă, iar pentru securitate suplimentară ar putea fi implementată autentificarea multi-factor (MFA) pentru protejarea conturilor utilizatorilor.
- **Gestionarea resurselor**: Platforma nu include o gestionare avansată a resurselor, cum ar fi sălile de curs sau echipamentele, care ar putea fi necesare pentru o planificare detaliată a activităților didactice.
- **Integrare cu alte platforme**: Deși aplicația permite gestionarea activităților și cursurilor, nu există integrare cu alte platforme educaționale existente (de exemplu, platforme pentru cursuri online, platforme de videoconferință, etc.).

b) Dezvoltări ulterioare:

- **Optimizarea performanței și scalabilității**: Pentru a îmbunătăți performanța aplicației în condiții de trafic mare, vor fi realizate optimizări ale interogărilor SQL și vor fi implementate tehnici de indexare mai eficiente.
- **Implementarea autentificării multi-factor (MFA)**: În scopul de a îmbunătăți securitatea aplicației, ar trebui implementată o autentificare multi-factor, pentru a preveni accesul neautorizat.
- **Managementul resurselor**: În viitor, ar putea fi implementată o funcționalitate de gestionare a resurselor (săli de curs, echipamente), care să ajute la organizarea și planificarea activităților didactice într-un mod mai eficient.

- **Integrare cu platforme externe**: Pentru a spori funcționalitatea și accesibilitatea platformei, ar fi utilă integrarea acesteia cu platforme educaționale externe (ex. Moodle, Zoom, Google Classroom), oferind astfel un mediu mai complet pentru învățare și interacțiune.
- **Extinderea funcționalităților de raportare**: Ar putea fi adăugate funcționalități de raportare avansată pentru profesori și administratori, astfel încât aceștia să poată analiza mai ușor performanțele studenților și activitățile din cadrul platformei.
- **Sprijin pentru învățarea online**: În contextul educației moderne, se impune integrarea de funcționalități pentru susținerea învățării online (ex. forumuri, sesiuni live de Q&A, integrarea cu instrumente de videoconferință).

De asemenea, ar putea fi adăugate următoarele:

- elaborarea unui algoritm de repartizare a studenților la toate tipurile de activități ale cursurilor la care sunt înscriși, fără să existe suprapuneri de orar și sugerarea unor ferestre potrivite pentru activitățile din grupurile de studiu;
- afișarea unor sugestii de participanți la grupurile de studiu;
- posibilitatea adăugării unui cadru didactic la o activitate din grupul de studiu;