

RAPORT FINAL

Proiect III - Tehnologii și instrumente pentru dezvoltarea programelor



UNIVERSITATEA DIN CRAIOVA

FACULTATEA DE AUTOMATICĂ, CALCULATOARE ȘI ELECTRONICĂ

Student: Cocei Janina Constantina

Grupa: CR4.S1 A

Anul de studiu: IV

Specializarea: Calculatoare Română

Data: 15.01.2025

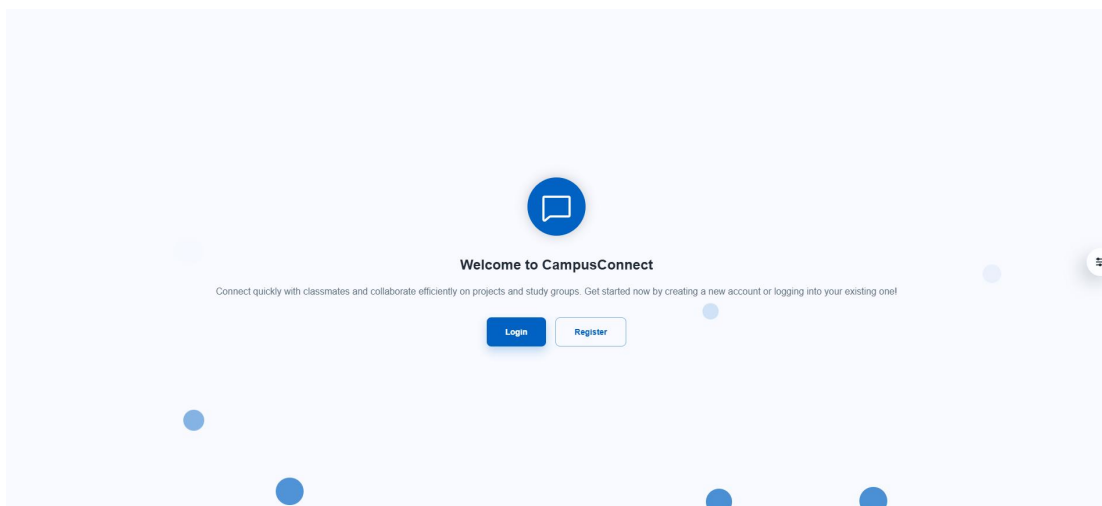
CUPRINS

I. Introducere -----	2
II. Funcționalitățile aplicației -----	2
III. Tehnologii folosite -----	9

I. Introducere

Într-un context educațional modern, accesul rapid și sigur la informații, suportul academic și colaborarea eficientă reprezintă piloni esențiali pentru succesul studenților. Aplicația CampusConnect a fost creată pentru a răspunde acestor nevoi, fiind un instrument digital dedicat instituțiilor de învățământ superior. Această platformă oferă funcționalități de comunicare în timp real, grupuri de discuții, precum și un sistem de administrare a comunității studențești. CampusConnect facilitează interacțiunea între studenți și tutori, oferind un mediu colaborativ și sigur.

II. Funcționalitățile aplicației



La pornirea aplicației, utilizatorul va fi întâmpinat de o pagină de tip Welcome, care prezintă două opțiuni principale:

Logare – Permite utilizatorului să se conecteze la aplicație prin furnizarea unui email și a unei parole.

Înregistrare – Permite utilizatorului să creeze un cont nou în aplicație.

La completarea fiecărei opțiuni (logare sau înregistrare), aplicația va verifica dacă informațiile introduse sunt corecte:

Logare: Verificarea existenței utilizatorului și a parolei corecte.


Înregistrare: Verificarea dacă adresa de email este validă și dacă nu există deja un cont asociat aceleiași adrese.

De asemenea, exista validari precum, email-ul si parola sunt required.

Parolele utilizatorilor sunt stocate criptate, folosind un algoritm de criptare puternic (bcrypt).

Toate datele introduse de utilizatori sunt validate atât pe client cât și pe server, pentru a preveni eventualele atacuri de tipul SQL Injection sau Cross-Site Scripting (XSS).

:



Get Started With CampusConnect

Already have an account? [Sign in](#)

Name
Demo



Email address
demo@yahoo.com

Password

Create Account

By signing up, I agree to [Terms of service](#) and [Privacy Policy](#).

OR

Login to CampusConnect

New User? [Create an account](#)


Email address
demo@yahoo.com

Password

[Forgot Password?](#)

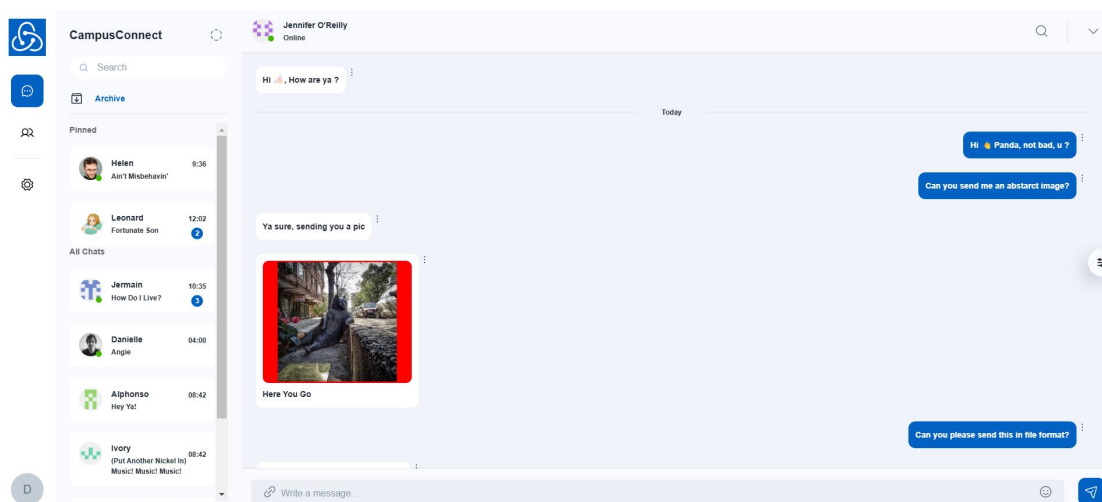
Login

OR



În cazul unei autentificări reușite, utilizatorul va fi redirecționat către pagina principală a aplicației.

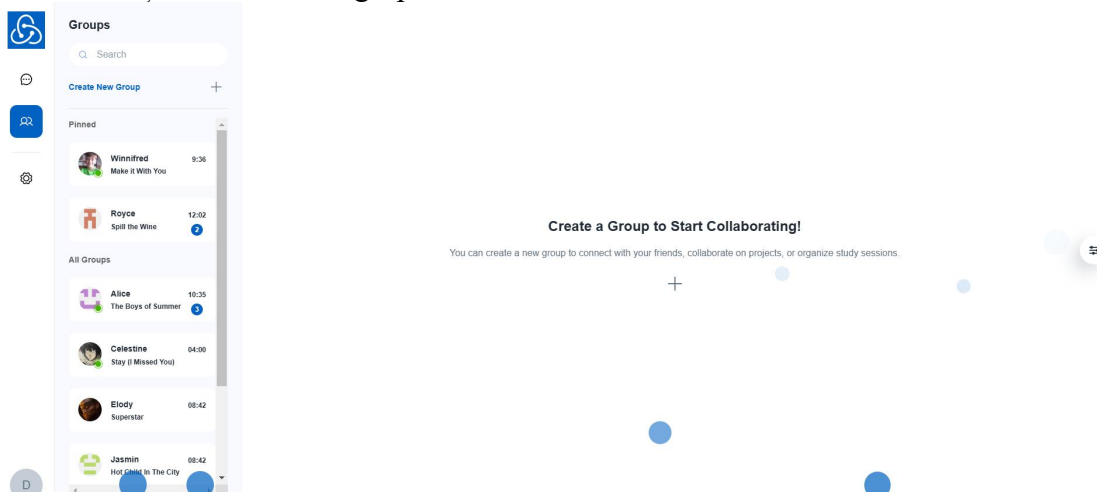
După o înregistrare cu succes, utilizatorul va fi redirecționat către pagina de logare.



După autentificare, utilizatorul va fi redirecționat către pagina principală a aplicației, care include un sidebar pentru navigare și un meniu pe user pentru gestionarea setărilor și profilului.

Aplicația oferă următoarele opțiuni:

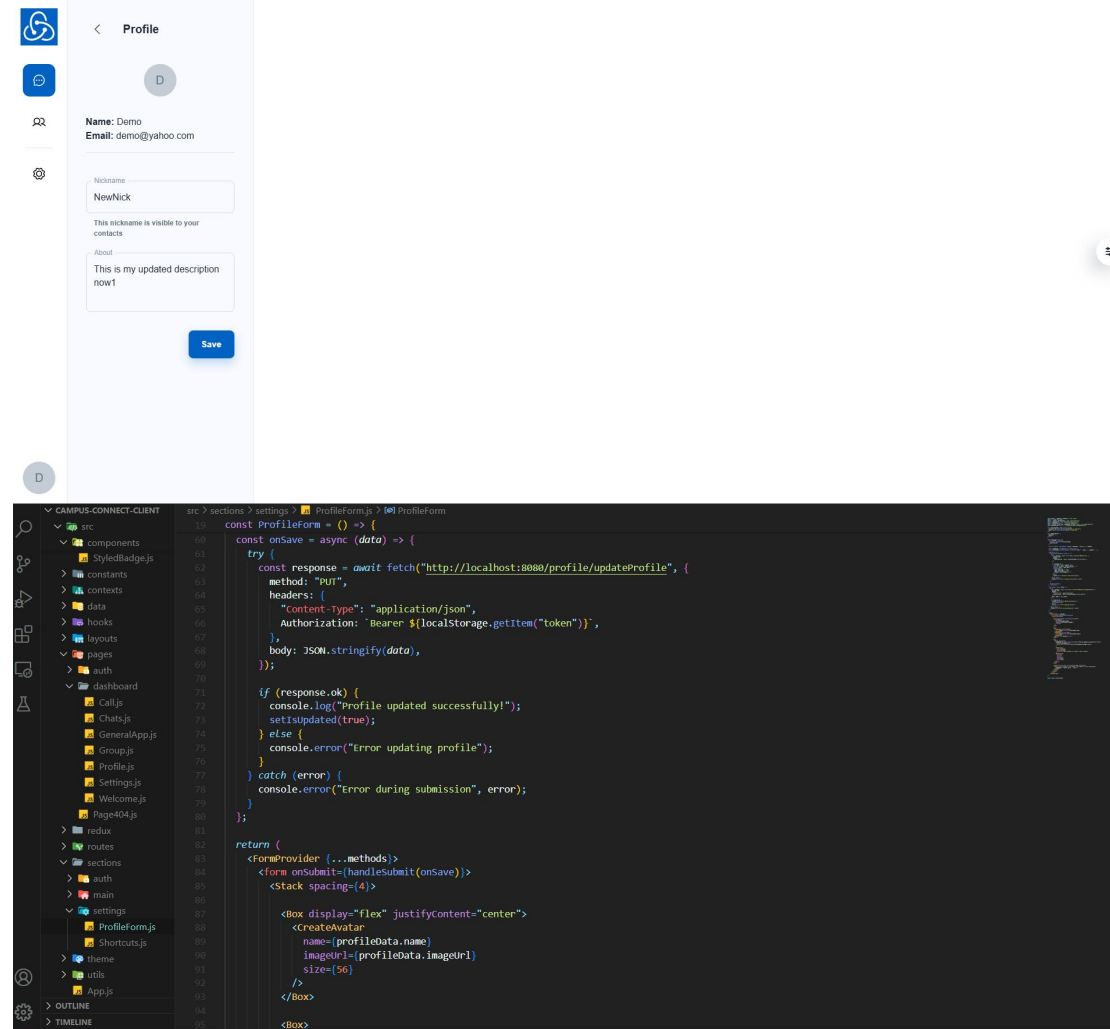
- Logo-ul Aplicației: La apăsarea logo-ului aplicației, utilizatorul va fi redirecționat către pagina de Chat, unde poate vizualiza și participa la conversațiile existente.
- Sidebar: Pe partea laterală a ecranului, utilizatorii vor găsi un sidebar care conține trei secțiuni principale:
 1. Chat: Secțiune care permite utilizatorilor să vizualizeze și să acceseze conversațiile lor anterioare.
 2. Grupuri: Secțiune dedicată grupurilor între studenți, unde utilizatorii pot interacționa cu diferite grupuri.



3. Setări: Secțiune care permite accesul la setările aplicației pentru personalizarea experienței utilizatorului.

4. Meniu pe User: În partea de sus a aplicației, utilizatorul va găsi un meniu care conține opțiunile:

➤ Profil: Permite utilizatorului să vizualizeze și să editeze informațiile sale personale, inclusiv numele, poza de profil și alte detalii relevante.

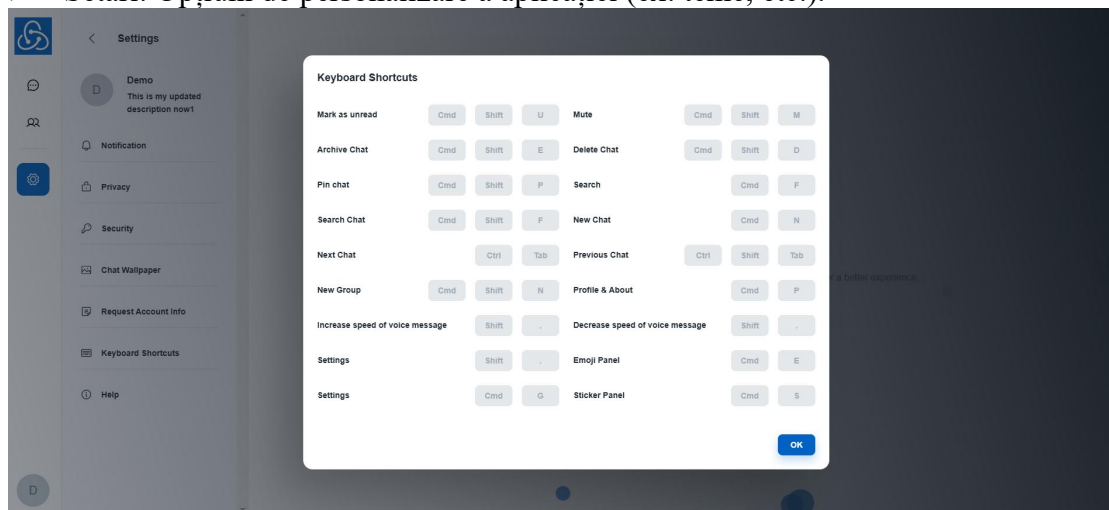


```

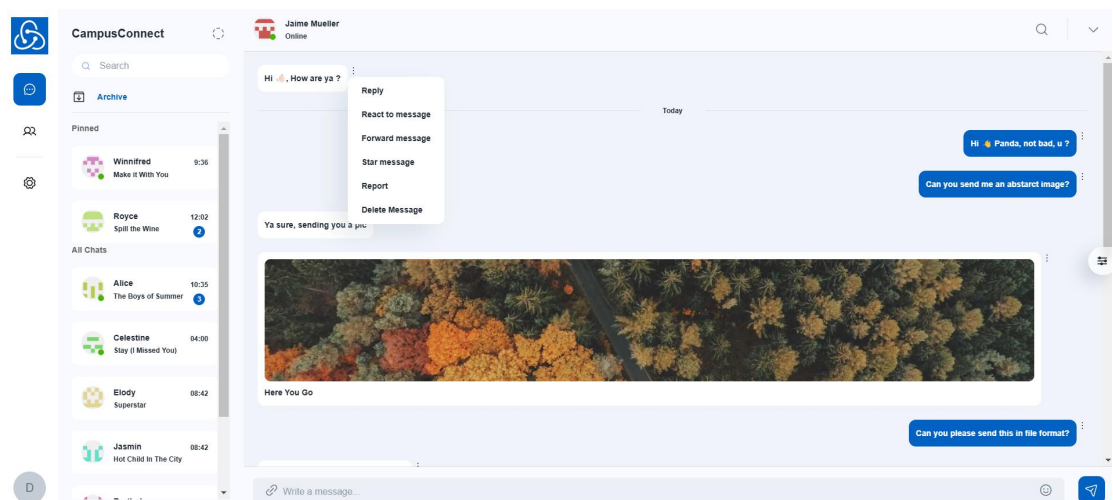
19 const ProfileForm = () => {
81
82   return (
83     <FormProvider {...methods}>
84       <form onSubmit={handleSubmit(onSave)}>
85         <Stack spacing={4}>
86
87           <Box display="flex" justifyContent="center">
88             <CreateAvatar
89               name={profileData.name}
90               imageUrl={profileData.imageUrl}
91               size={56}
92             />
93           </Box>
94
95           <Box>
96             <Typography variant="body1">
97               <strong>Name:</strong> {profileData.name}
98             </Typography>
99             <Typography variant="body1">
100               <strong>Email:</strong> {profileData.email}
101             </Typography>
102             <Divider sx={{ marginY: 2 }} />
103           </Box>
104
105           <Box>
106             <Stack spacing={3}>
107               {isUpdated && <Alert severity="success">Profile updated successfully!</Alert>}
108               {!!errors.afterSubmit && (
109                 <Alert severity="error">{errors.afterSubmit.message}</Alert>
110               )}
111
112               <RHFField
113                 name="nickname"
114                 label="Nickname"
115                 helperText="This nickname is visible to your contacts"
116               />

```

➤ Setări: Opțiuni de personalizare a aplicației (ex: teme, etc.).



➤ Logout: Opțiune care permite utilizatorului să se deconecteze din aplicație.



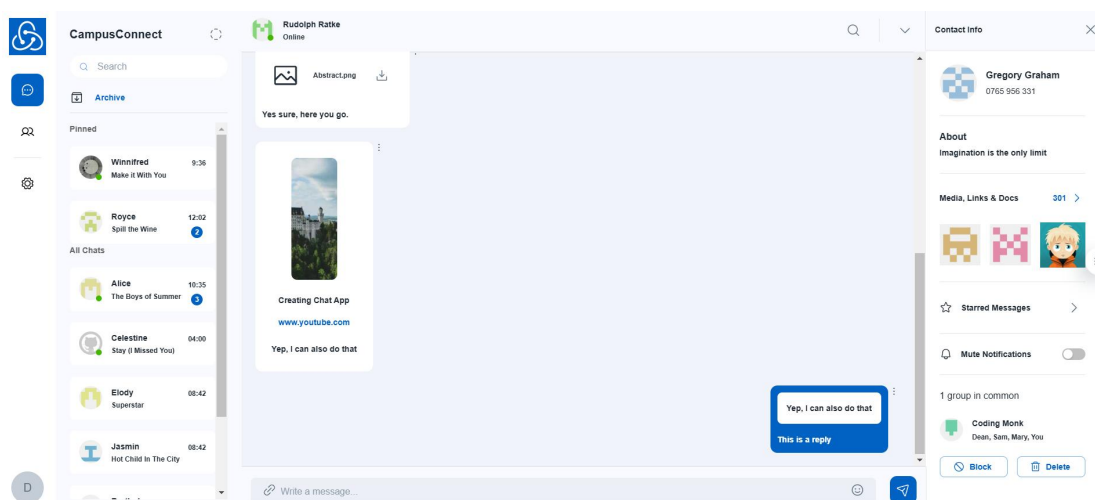
În cadrul aplicației, mesajele dintre utilizatori sunt gestionate într-un mod interactiv și flexibil, oferind opțiuni multiple pentru fiecare mesaj în parte. Aceste opțiuni sunt disponibile pentru a îmbunătăți experiența de utilizare și pentru a oferi mai multe funcționalități utilizatorilor. Opțiunile disponibile pentru fiecare mesaj sunt:

Replay – Permite utilizatorului să răspundă unui mesaj anume din cadrul conversației.

Raport – Permite utilizatorului să raporteze un mesaj ca fiind nepotrivit sau încălcând regulile aplicației.

Starred – Permite utilizatorului să salveze un mesaj important pentru acces rapid ulterior, marcându-l cu o stea.

Forwarded – Permite utilizatorului să redirecționeze mesajul către un alt utilizator sau grup.



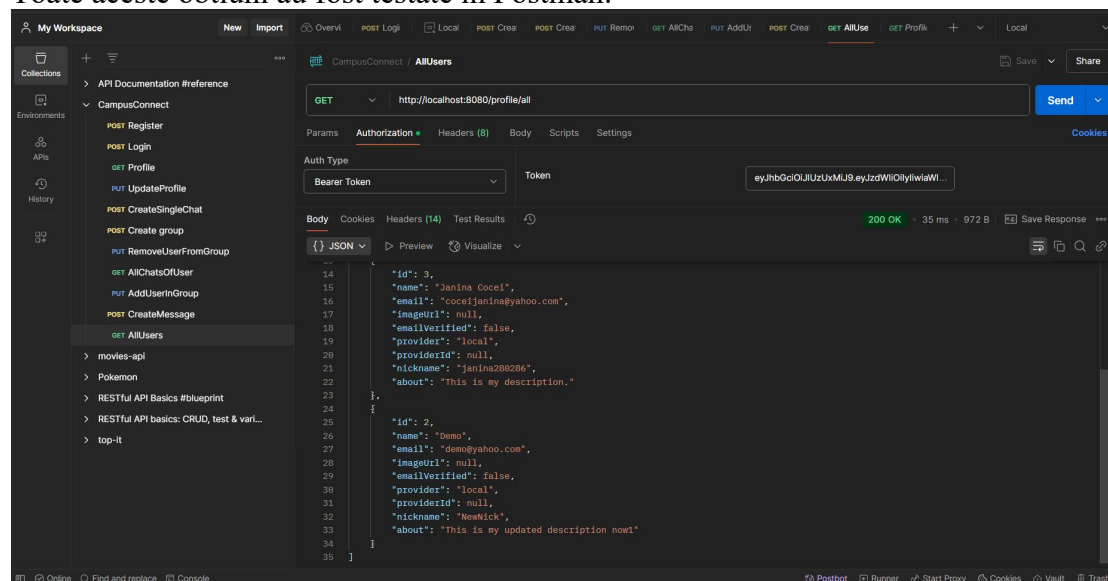
Aplicația permite utilizatorului să interacționeze într-un mod mai detaliat și personalizat cu ceilalți utilizatori prin oferirea unor funcționalități suplimentare. Aceste funcționalități sunt destinate să îmbunătățească experiența de utilizare, permițând utilizatorului să acceseze informații suplimentare despre conversațiile sale și să își personalizeze notificările sau să gestioneze interacțiunile.

În aplicație, Redux este folosit pentru gestionarea stării aplicației, în special pentru componentele care implică interacțiuni dinamică, cum ar fi deschiderea detaliilor despre utilizator, vizualizarea imaginilor și fișierelor din conversații. Redux ajută la centralizarea stării aplicației și la sincronizarea acesteia între diferitele componente, facilitând astfel o experiență de utilizare mai fluidă și ușor de întreținut.

Opțiunile disponibile pentru fiecare conversație sunt:

- Vizualizarea Detaliilor Utilizatorului – Permite utilizatorului să vizualizeze informațiile detaliate despre persoana cu care discută.
- Vizualizarea Imaginilor și Documentelor Trimise – Permite utilizatorului să vizualizeze fișierele media și documentele trimise în cadrul conversației.
- Mute la Notificări – Permite utilizatorului să oprească notificările pentru o anumită conversație.
- Blocarea Utilizatorului – Permite utilizatorului să blocheze complet persoana cu care discută, interzicându-i orice interacțiune ulterioară.
- Ștergerea Conversației – Permite utilizatorului să șteargă întreaga conversație, eliminând-o din istoricul său.

Toate aceste obțiuni au fost testate în Postman:



În cadrul aplicației, testele unitare pentru componenta UserRepository sunt realizate folosind framework-ul JUnit. Aceste teste verifică funcționarea corectă a interacțiunilor cu baza de date, asigurându-se că operațiunile de citire și scriere a datelor pentru entitatea User sunt corect implementate. Testele se bazează pe un context Spring Boot și utilizează EntityManager pentru accesarea directă a bazei de date într-un mod controlat.

Testul testUserQuery:

Scop: Verifică dacă un utilizator cu un ID specificat poate fi recuperat din baza de date.

Se execută o interogare JPQL pentru a găsi un utilizator pe baza ID-ului său (id = 1L).

Testul se asigură că utilizatorul găsit nu este null.

Testul va returna un utilizator existent în baza de date cu ID-ul specificat. Dacă utilizatorul există, se va confirma că acesta nu este null.

Dacă utilizatorul nu există, testul va eșua, iar procesul va indica absența utilizatorului respectiv.

Test Assert:

assertNotNull(user); – Asigură că rezultatul interogării nu este nul.

Testul testUserQueryCreate:

Verifică dacă un utilizator poate fi creat și salvat corect în baza de date, iar apoi poate fi recuperat cu succes.

Un obiect User nou este creat, setându-i valorile pentru nume, email, nickname și descriere.

Utilizatorul este salvat în baza de date prin entityManager.persist() și confirmat cu un apel de entityManager.flush().

După salvare, testul interoghează baza de date pentru a recupera utilizatorul prin ID-ul său.

Se verifică dacă detaliile utilizatorului salvate corespund celor introduse la crearea acestuia.

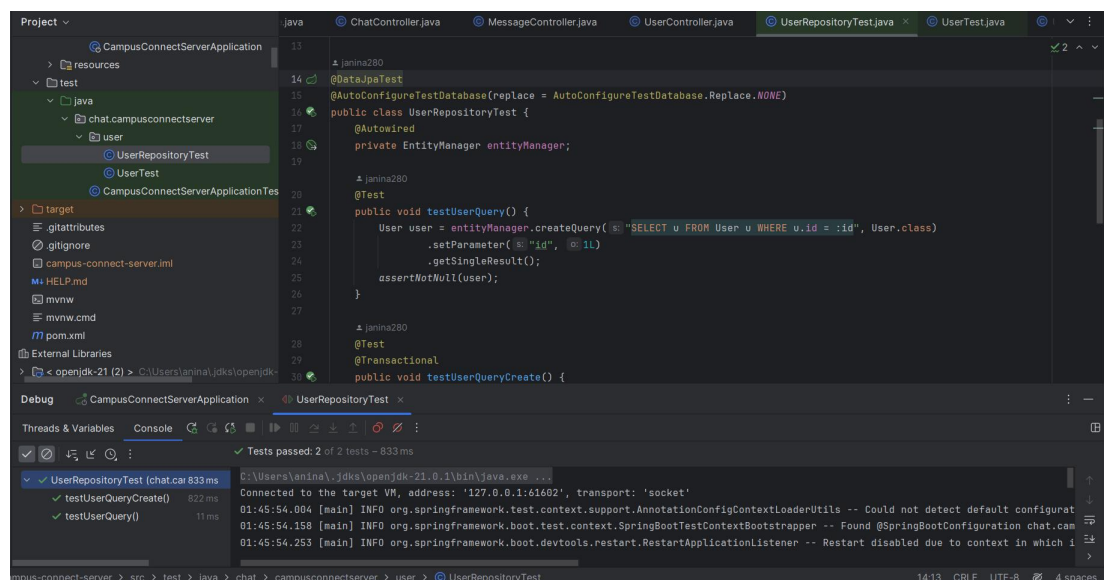
Un utilizator nou este introdus în baza de date. Apoi, acesta este recuperat din baza de date folosind ID-ul generat de persistenta în entitate.

Testul verifică faptul că datele utilizatorului, cum ar fi numele, email-ul, nickname-ul și descrierea, sunt corect stocate și returnate.

Test Assert:

`assertNotNull(user);` – Asigură că utilizatorul a fost salvat și că nu este null.

`assertEquals` pentru verificarea valorilor introduse la crearea utilizatorului, comparându-le cu cele salvate în baza de date.



III. Tehnologii folosite

Frontend: React/HTML/CSS pentru implementarea interfeței și a opțiunilor de interacțiune cu mesajele.

Backend: Java Spring Boot pentru gestionarea mesajelor și a operațiunilor de redirectionare, raportare, salvare și răspuns.

Baza de date: PostgreSQL pentru stocarea mesajelor și a acțiunilor utilizatorilor (starred, forwarded, etc.).

Testare: JUnit