



# **Fake Teams**

## **Proiect Transmisia Datelor**

Autori: Culcer Alexandru-Mihai

Josan Diana

Vieriu Constantin-Gabriel

Grupa: 30131

# Cuprins

1. Introducere.....	3
1.1 Scopul proiectului.....	3
1.2 Specificații.....	3
2. Descrierea aplicației.....	3
2.1 Arhitectura .....	3
2.2 Navigarea în aplicație.....	5
3. Implementare.....	8
4. Funcționalități.....	9

## **1. Introducere**

### **1.1 Scopul Proiectului**

Scopul proiectului este de a pune bazele cunoștințelor pentru realizarea unei aplicații Web, folosind tehnologii actuale și de interes, precum HTML, CSS și JavaScript. Având la dispoziție materialele de la laborator, respectiv de la cursul Transmisia Datelor, proiectul are ca fundamente noțiuni de o complexitate medie pentru a implementa o aplicație de tip chat.

### **1.2 Specificații**

Proiectul prezintă o aplicație Web de comunicare în timp real, denumită “Fake Teams”. Aceasta este construită pe modelul aplicației Microsoft Teams, din punctul de vedere al interfeței și al interacțiunii cu utilizatorul.

În proiect am folosit tehnologiile și tool-urile următoare: Node.JS, Vanilla JavaScript, HTML, CSS, NeDB, Socket.io framework, Moment, Express. Folosind Web Sockets, putem asigura comunicarea bidirecțională dintre client și server. Totodată, aplicația are în componență și o bază de date, cu ajutorul căreia am manipulat datele utilizatorilor. NeDB este un tip de bază de date scrisă integral în JavaScript și care implementează MongoDB API.

## **2. Descrierea aplicației**

### **2.1 Arhitectura**

Aplicația permite navigarea între trei pagini: pagina de logare, pagina de “manage users” și pagina de mesagerie.

Prima pagina are o structură relativ simplă, conținând un textbox, un câmp de unde utilizatorul poate alege o opțiune listă predefinită, un buton care va direcționa utilizatorul către cea de a doua pagină a aplicației, un al doilea buton care v-a deschide pagina de creare, ștergere și modificare a unui utilizator și logo-ul. (fig. 2.1.1)

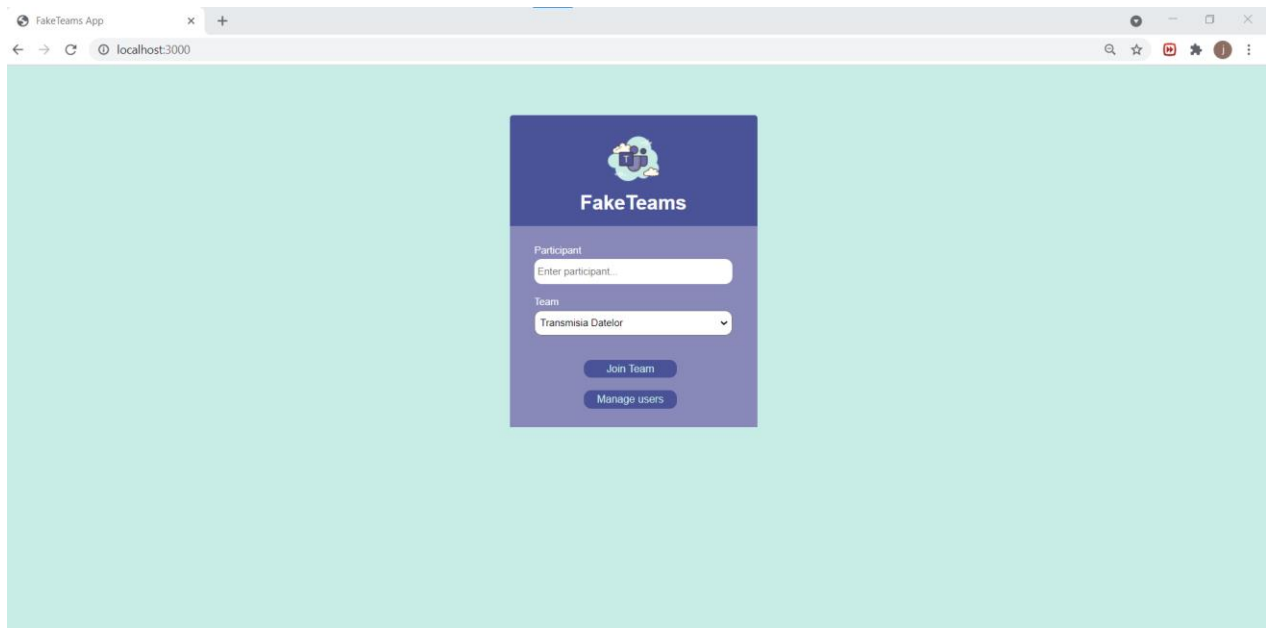


fig. 2.1.1

Cea de a doua pagină este împărțită în mai multe blocuri funcționale, unul pentru transmiterea textului dorit, unul pentru vizualizarea mesajelor trimise și primite pe acel chat și unul pentru a vizualiza cine este participant la chat și care este numele echipei la care s-a logat utilizatorul. Acestea vor fi detaliate la punctul următor. (fig.2.1.2)

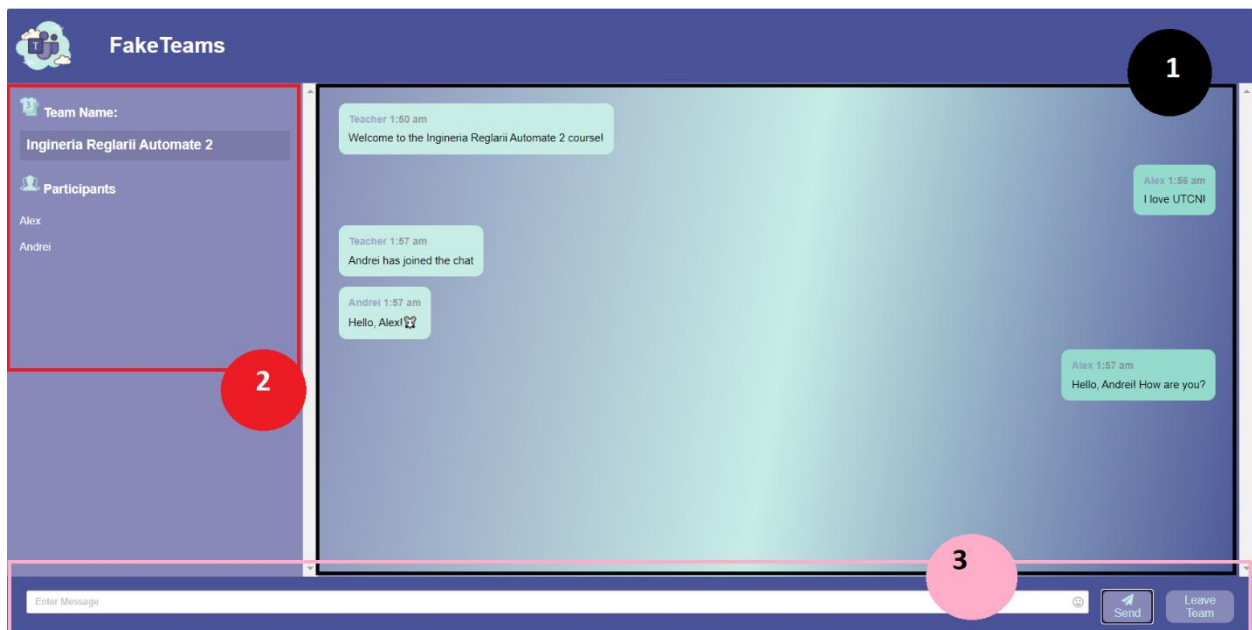
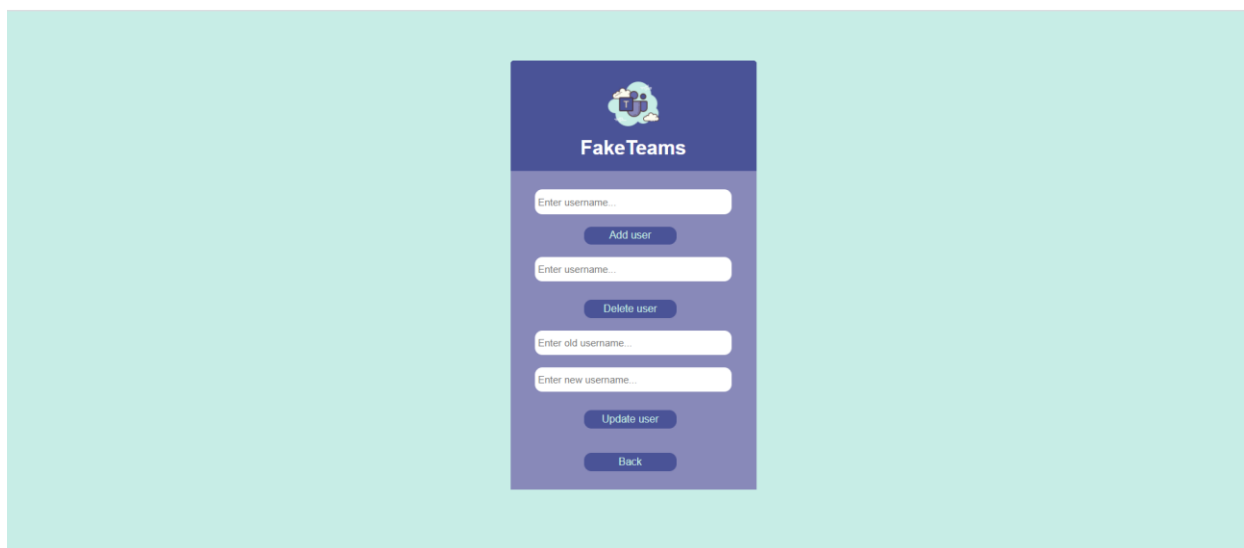


fig. 2.1.2

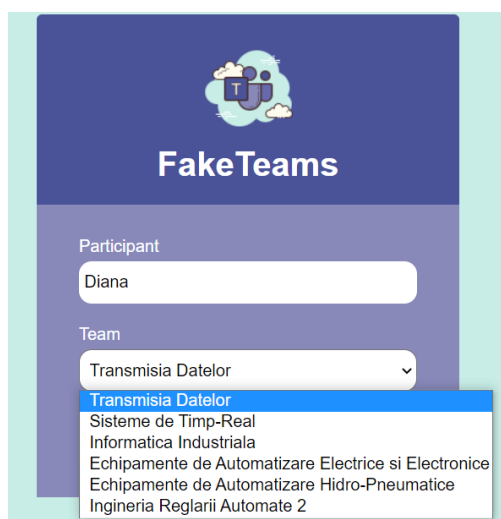
Cea de a treia pagină este cea în care se va face manipularea datelor din baza de date. Mai exact, se poate crea un utilizator nou, se poate șterge, respectiv sau se poate modifica numele unuiia deja existent. (fig. 2.1.3)



*fig. 2.1.3*

## **2.2 Navigarea în aplicație**

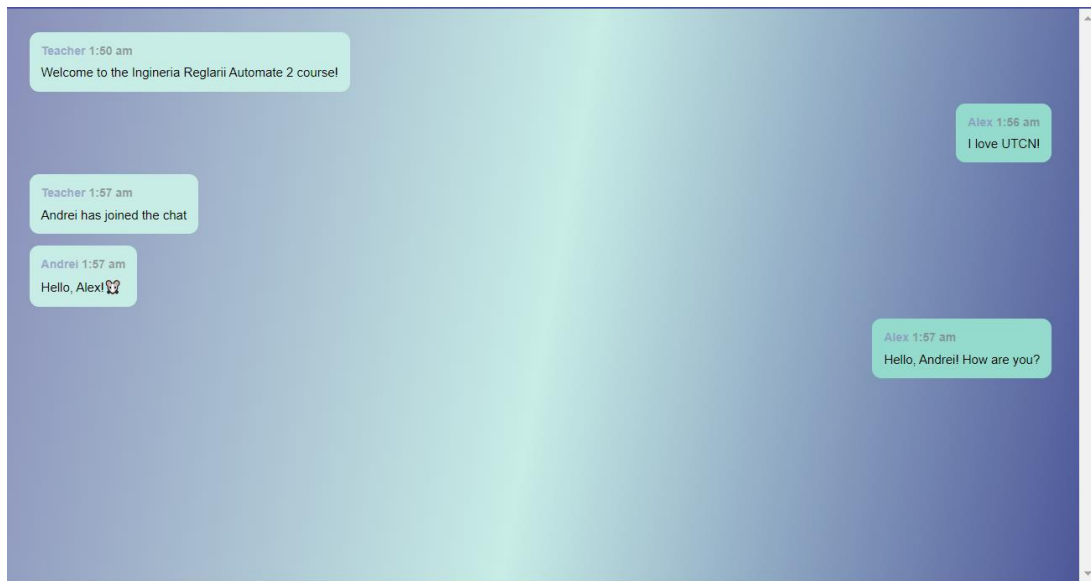
Aplicația se deschide cu o pagină principală care va permite conectarea la o echipă (team). Pentru acest lucru este nevoie ca utilizatorul să introducă un username și să aleagă echipa în care dorește să intre, dintr-o listă predefinită. (fig. 2.2.1). De menționat este faptul că numele utilizatorului trebuie să existe deja în baza de date, deci pentru a permite logarea, prima dată trebuie ca acesta să își creeze un cont.



*fig. 2.2.1*

Odată apăsând butonul de “Join Team”, se deschide cea de a doua pagină, care reprezintă interfața de mesagerie a echipei în care utilizatorul a ales să se logheze. După cum a fost menționat și la punctul anterior, această a doua pagină este formată din mai multe blocuri funcționale.

Partea centrală este reprezentată de blocul de chat, unde utilizatorul poate vizualiza mesajele primite sau trimise. Acesta este prevăzut și cu un scroll bar, pentru a putea naviga (sus/jos) între mesajele anterioare și a le putea revedea oricând pe toate. (fig. 2.2.2)



*fig. 2.2.2*

În partea stângă a paginii este blocul unde se poate observa numele chat-ului (Team Name) la care este conectat utilizatorul și, de asemenea, participanții la conversație, adică alți utilizatori care s-au înscris în același chat. Dacă deschidem aplicația într-un alt tab și ne logăm cu un alt username la aceeași echipă unde deja este o persoană logată, în partea stângă a paginii de mesagerie vor apărea ambele nume în secțiunea cu participanți. (fig. 2.2.3) Acest proces se păstrează pentru orice număr de participanți care se adaugă în chat.



*fig. 2.2.3*

În partea de jos a paginii este blocul unde se poate scrie mesajul pe care utilizatorul dorește să îl trimită în chat, precum și două butoane, unul cu care utilizatorul poate părăsi chat-ul, denumit „Leave Team” și unul denumit „Send”, la apăsarea căruia, odată introdus mesajul text, acesta se va trimite în blocul de mesagerie. (fig. 2.2.4)



fig. 2.2.4

Totodată, în dreapta locului predestinat redactării textului este prezent și un buton care deschide un meniu cu emoticoane, un feature foarte des utilizat în toate aplicațiile actuale de transmitere de mesaje între utilizatori. (fig. 2.2.5)

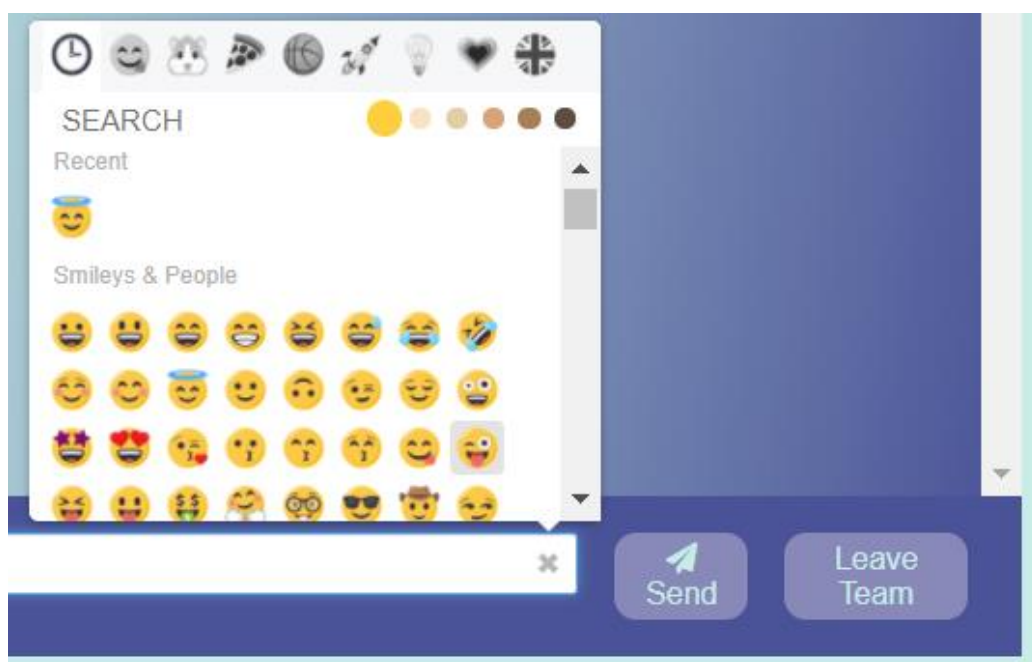
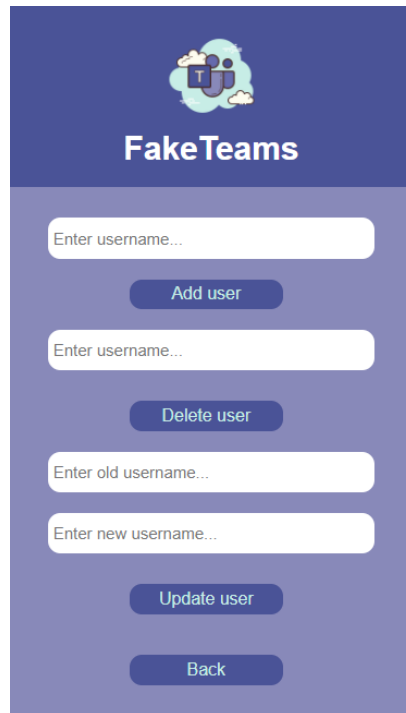


fig. 2.2.5

Pentru ca pașii anterior menționați să poată fi realizați, fiecare utilizator va trebui să existe în baza de date. Pentru a realiza acest aspect, acesta trebuie să acceseze cea de a treia pagină a aplicației, prin apăsarea butonului “Manage users” din ecranul principal. Astfel se deschide pagina de manage users, în care utilizatorul are opțiunile de add, delete și update. Cele trei funcționalități vor fi detaliate mai jos. De asemenea, această pagină are și un buton de back, prin apăsarea căruia utilizatorul este trimis înapoi în pagina principală de login. (fig. 2.2.6)



*Fig. 2.2.6*

### **3. Implementare**

Implementare propriu zisă a aplicației a început cu proiectarea părții de front-end, folosindu-ne de HTML și CSS pentru a crea design-ul celor două pagini care urmau a fi populate.

În folderul „public” avem 3 fișiere de tip html și anume: index.html, conceput pentru crearea interfeței primei pagini care reprezintă pagina de logare; chat.html, pentru crearea interfeței celei de a doua pagini, cea de mesagerie, și register.html pentru editarea contului.

După ce ne-am asigurat că avem instalate toate tool-urile necesare și că am făcut setările pentru a realiza conexiunea client-server, am proiectat partea de back-end, motorul aplicației, adică serverul. Codul aferent se poate găsi în fișierul „server.js”. Primul pas care a trebuit efectuat pentru a putea deschide prima pagina a aplicației a fost să setăm în acesta fișierul denumit „public” ca fiind de tip static și să setăm portul de comunicare cu browser-ul pentru a putea deschide aplicația. Astfel, dacă în browser scriem <http://localhost:3000>, se va deschide pagina de join în echipă (index.html). Apoi am realizat conexiunea cu web socket-ul și cu baza de date NeDB, pentru gestiunea utilizatorilor.

După ce toate aceste detalii tehnice au fost puse la punct și toate conexiunile necesare au fost îndeplinite, a urmat partea de implementare a funcționalităților și a interfeței grafice.



## 4. Funcționalități

Primele funcționalități care trebuie menționate sunt cele legate de logarea utilizatorilor în aplicație. În pagina deschisă prin apăsarea butonului “Manage users” din meniul principal, orice persoană își poate crea contul, prin introducerea în primul textbox a unui username și prin apăsarea butonului “Add user”. Utilizatorul va primi apoi un mesaj că a fost adăugat cu succes (fig. 4.1). Odată ce acesta a fost adăugat în baza de date, se va reîntoarce în pagina principală, de unde își va putea alege chat-ul la care dorește să se alăture (fig. 4.2).

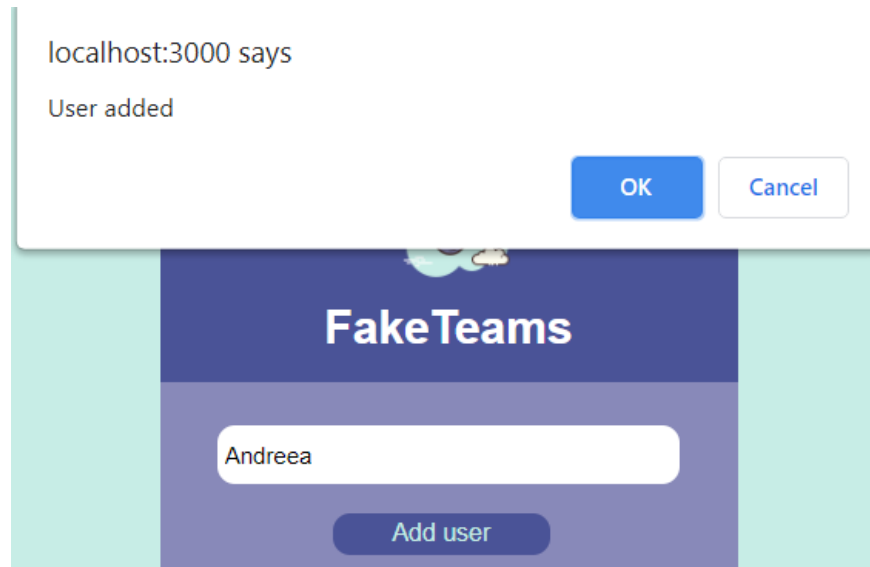


fig. 4.1

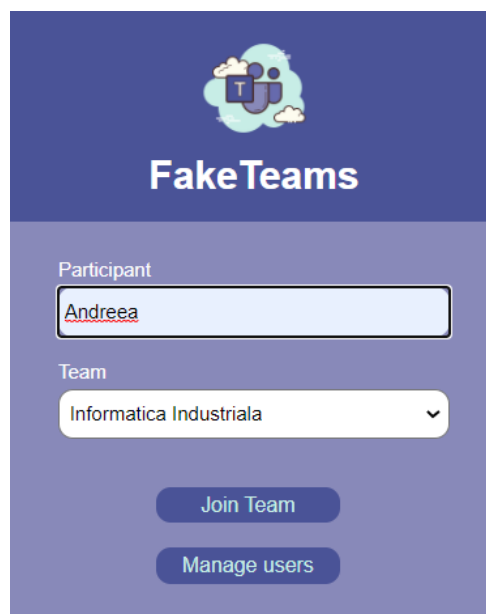
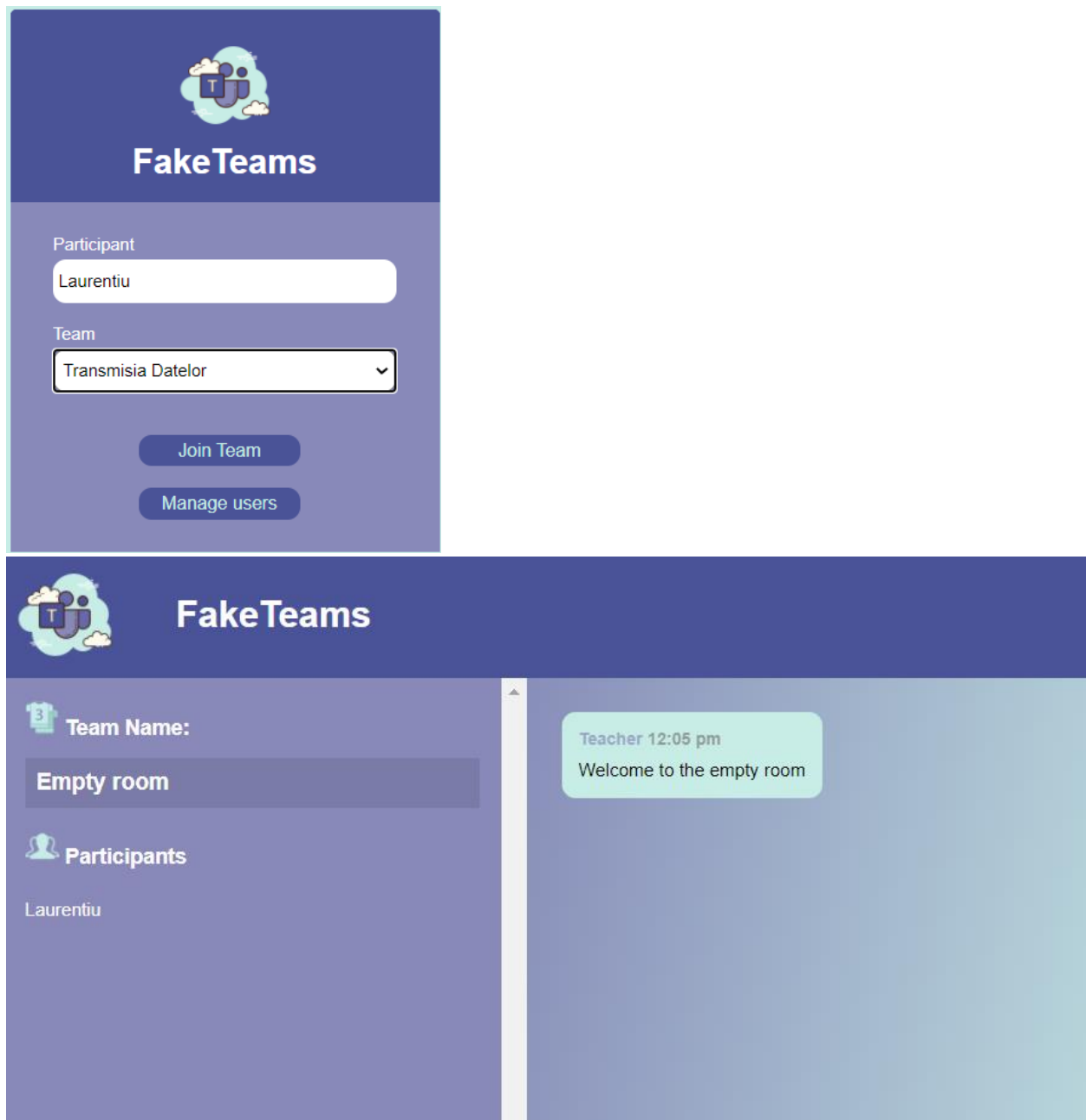


fig. 4.2 Am fost adăugat utilizatorul Andreea la baza de date. Se va putea conecta la chat-ul “Informatica Industrială”

Cu toate acestea, dacă utilizatorul nu a fost adăugat și încearcă să se alăture unui chat, acesta va fi direcționat către un chat denumit “Empty Room”, indiferent de chat-ul pe care l-a ales din lista predefinită a paginii de logare (fig. 4.3).



*fig. 4.3 Laurentiu nu s-a înregistrat, deci nu va putea intra în chat-ul “Transmisia Datelor”, ci a fost redirecționat către un chat nepopulat.*

O altă funcționalitate este aceea de modificare a username-ului unui utilizator. Această acțiune este posibilă tot din pagina Manage users. Este necesar a fi introdus vechiul username și

username-ul cu care va fi schimbat. Prin apăsarea butonului “Update User”, în baza de date se va modifica username-ul alocat utilizatorului respectiv și acesta se va putea loga cu noul nume ales.

Cea de a treia funcționalitate pentru manipularea datelor utilizatorilor este ștergerea acestuia. Dacă un utilizator a fost deja adăugat în baza de date, pentru ștergere, acesta trebuie să introducă username-ul și să apese butonul “Delete User”. Apoi va primi un mesaj în care i se spune că utilizatorul a fost șters (fig. 4.4)

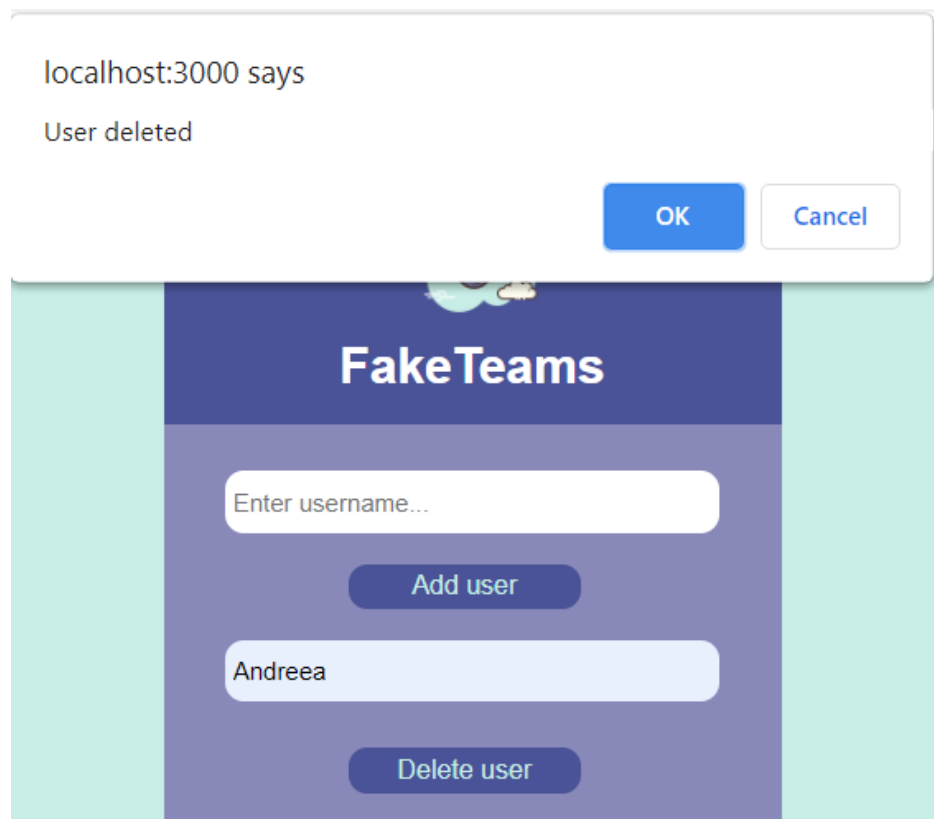


fig. 4.4

După ce utilizatorul Andreea a fost șters, ca și în cazul anterior, dacă dorește să se alăture unui chat, va fi redirecționat către “Empty Room” (fig. 4.5).

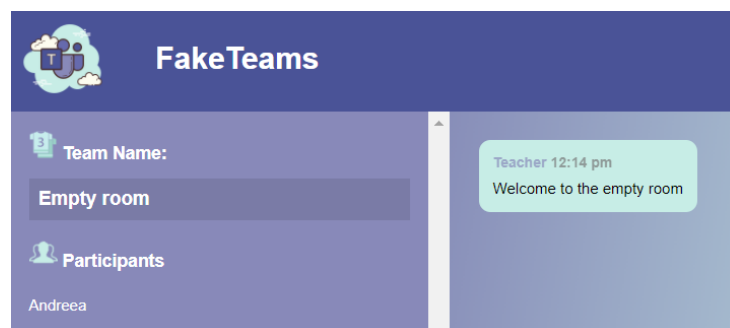
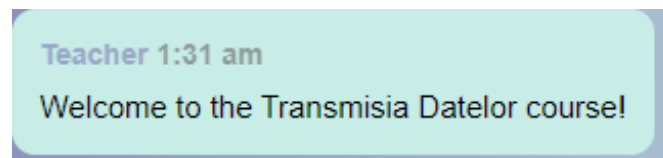


fig. 4.5

După ce utilizatorul s-a logat și a fost redirecționat la cea de a doua pagină a aplicației, acesta va primi, în mod automat de la server un mesaj de tipul “Welcome to the X course!”, unde X reprezintă numele echipei la care s-a logat. (fig. 4.6)

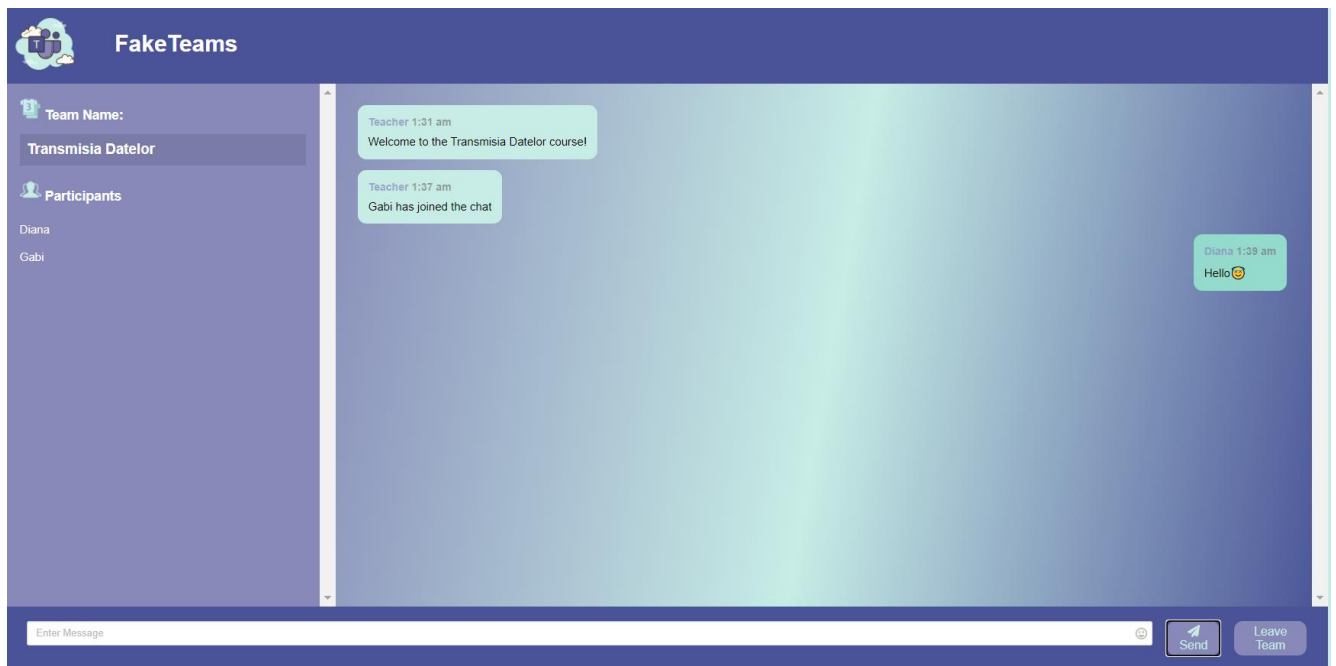


*fig. 4.6*

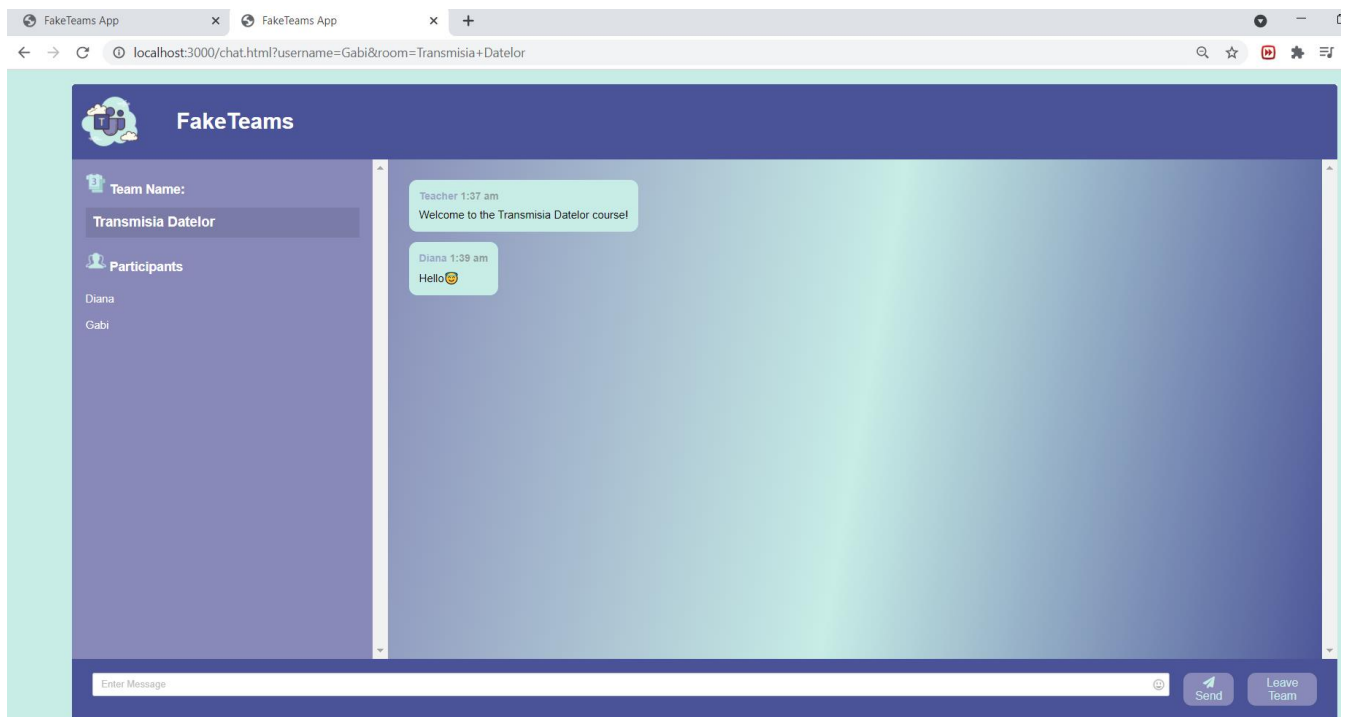
În fișierul denumit “main.js” am implementat funcționalitatea principală și anume trimiterea mesajelor de către utilizator, la apăsarea butonului “Send”, după introducerea textului. Dacă nu este niciun text scris în spațiul alocat pentru redactarea mesajului, dar totuși se apasă butonul “Send”, nu se va trimite niciun mesaj.

O modificare pe care am realizat-o a fost ca după apăsarea butonului “Send”, locul de introducere a textului să se golească, iar utilizatorul să poată introduce automat un alt mesaj, fără ca textul anterior trimis să rămână acolo. De asemenea, pentru o interfață cât mai plăcută visual pentru utilizatori, am proiectat și funcționalitatea ca fiecare mesaj trimis să aibe deasupra textului propriu zis și numele utilizatorului care l-a trimis și ora la care a fost trimis.

O funcționalitate relevantă din cadrul proiectului este că atunci când un utilizator nou se alătură grupului, persoanele deja logate primesc un mesaj de tipul “X has joined the chat”, din partea unui bot predefinit (denumit “Teacher”, în cazul nostru), unde X reprezintă username-ul utilizatorului nou logat. Același lucru se întâmplă și în cazul în care un utilizator părăsește chat-ul. După apăsarea butonului de “Leave Team”, ceilalți utilizatori logați pe acel chat vor primi un mesaj de tipul “X has left the chat”. (Aceste funcționalități sunt exemplificate în figurile fig. 4.7.1, fig. 4.7.2 și fig. 4.9)

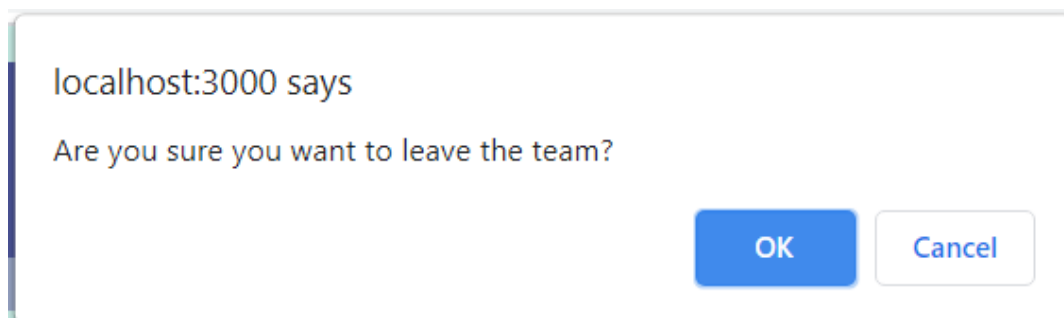


*fig. 4.7.1 Chat-ul Dianei arată că Gabi s-a alăturat chat-ului, iar ea a trimis mesajul cu “Hello”*

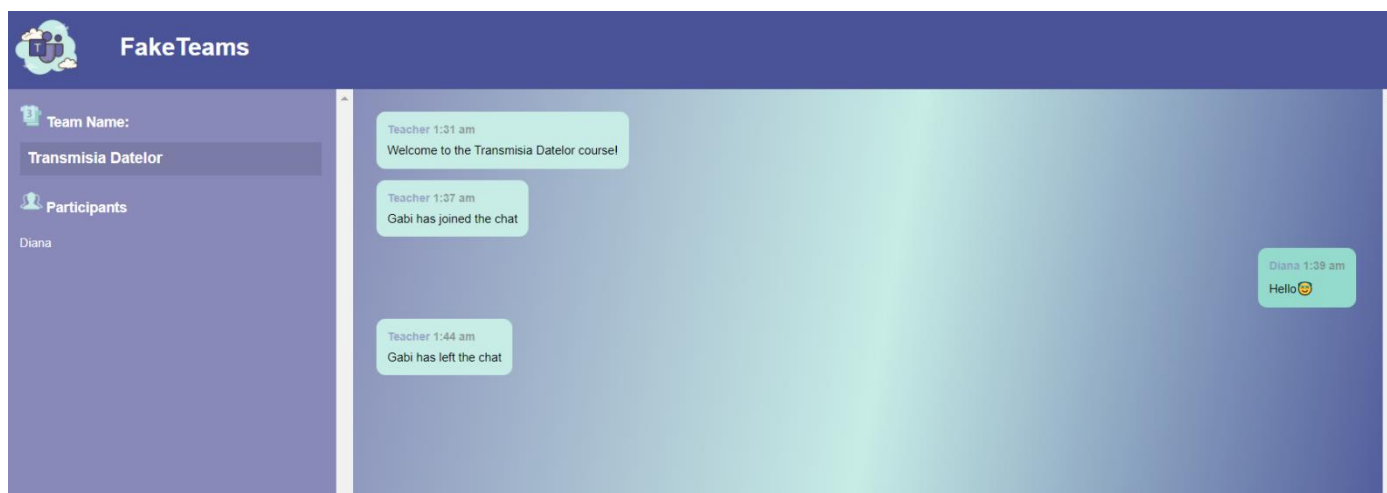


*fig. 4.7.2 Chatul lui Gabi arată doar mesajul Dianei cu “Hello”*

Totodată, dacă utilizatorul apasă pe butonul de “Leave Team”, îi va apărea acest mesaj. Doar în cazul în care acesta apasă pe “OK”, utilizatorul va fi scos din chat-ul respectiv și se va reîntoarce la prima pagină a aplicației. (fig. 4.8)



*fig. 4.8*



*fig. 4.9 Dacă Gabi a părăsit chat-ul, Dianeii îi va apărea acest mesaj*