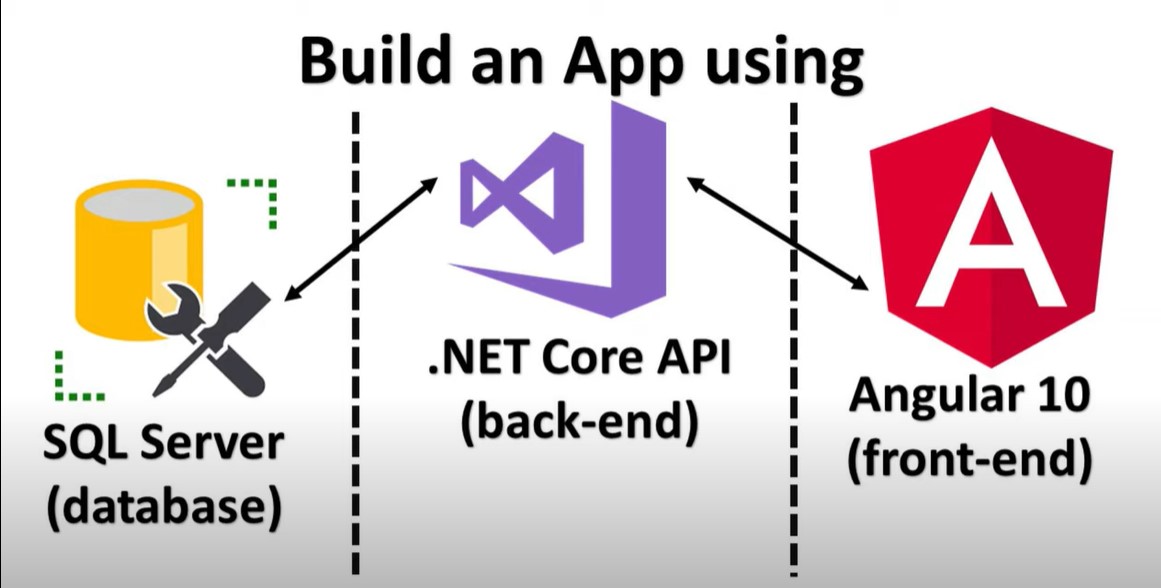
Aplicatie Fullstack Servicii IT

(ASP.NET - Angular)



Mocanu Gabriel

CUPRINS

[1. Introducere 3](#_Toc144827181)

[Figura 1 - Angajati 3](#_Toc144827182)

[Figura 2 – Auth0 4](#_Toc144827183)

[2. Configurare mediu dezvoltare 4](#_Toc144827184)

[Figura 3. CRUD in C# 5](#_Toc144827185)

[3. Dezvoltare Aplicatie 5](#_Toc144827186)

[Figura 4 – Swagger 6](#_Toc144827187)

[3.1 Configurare Angular 6](#_Toc144827188)

[Figura 5 – Functii CRUD 7](#_Toc144827189)

[Figura 6 – Adauga angajat 7](#_Toc144827190)

[Figura 7 – Tichete 8](#_Toc144827191)

[4 . Concluzii 8](#_Toc144827192)

# Introducere

Aplicatia Servicii IT si gestionare angajati este dezvoltata in C# cu Asp.Net Web API pe partea de Backend si cu ajutorul framework-ului Angular, pe partea de Frontend. Ca baza de date am ales sa utilizez SQL Server Management Studio. Aceasta aplicatie reprezinta un inceput pentru dezvoltarea unei aplicatii mai complexe de servicii IT - Call Center, pentru rezolvarea problemelor tehnice hardaware si software.

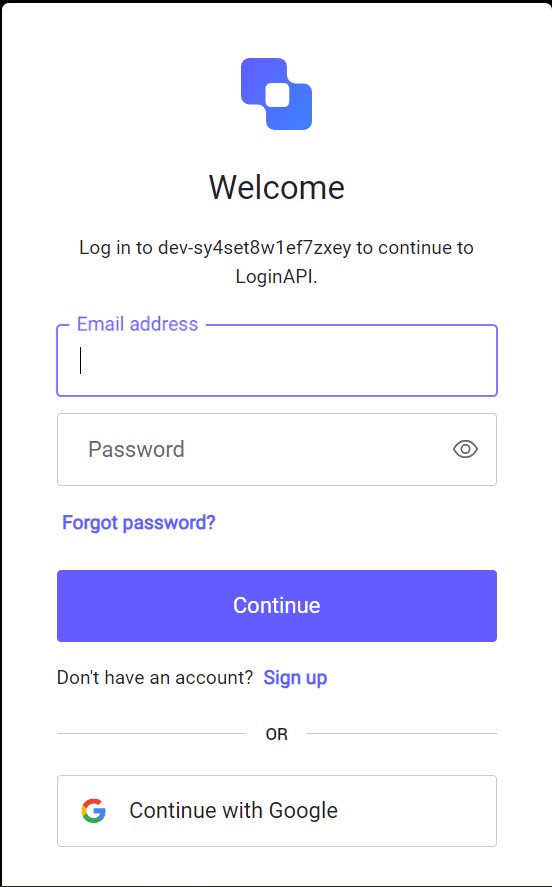
## Figura 1 - Angajati

A screenshot of a computer

Description automatically generated

In *figura 1*, este reprezentata interfata principala de unde se poate observa un navbar pe fundal albastru cu pricipalele functii. Datele despre angajati sau tichete deschise ale clientilor este disponibila doar daca un angajat este autentificat. Pentru aceasta am ales sa utilizez un API de autentificare numit **Auth0.** Acesta ofera posibilitatea de autentificare cu un cont privat sau un cont de google. Mai jos in Figura 2 este reprezentata interfata de Log in.

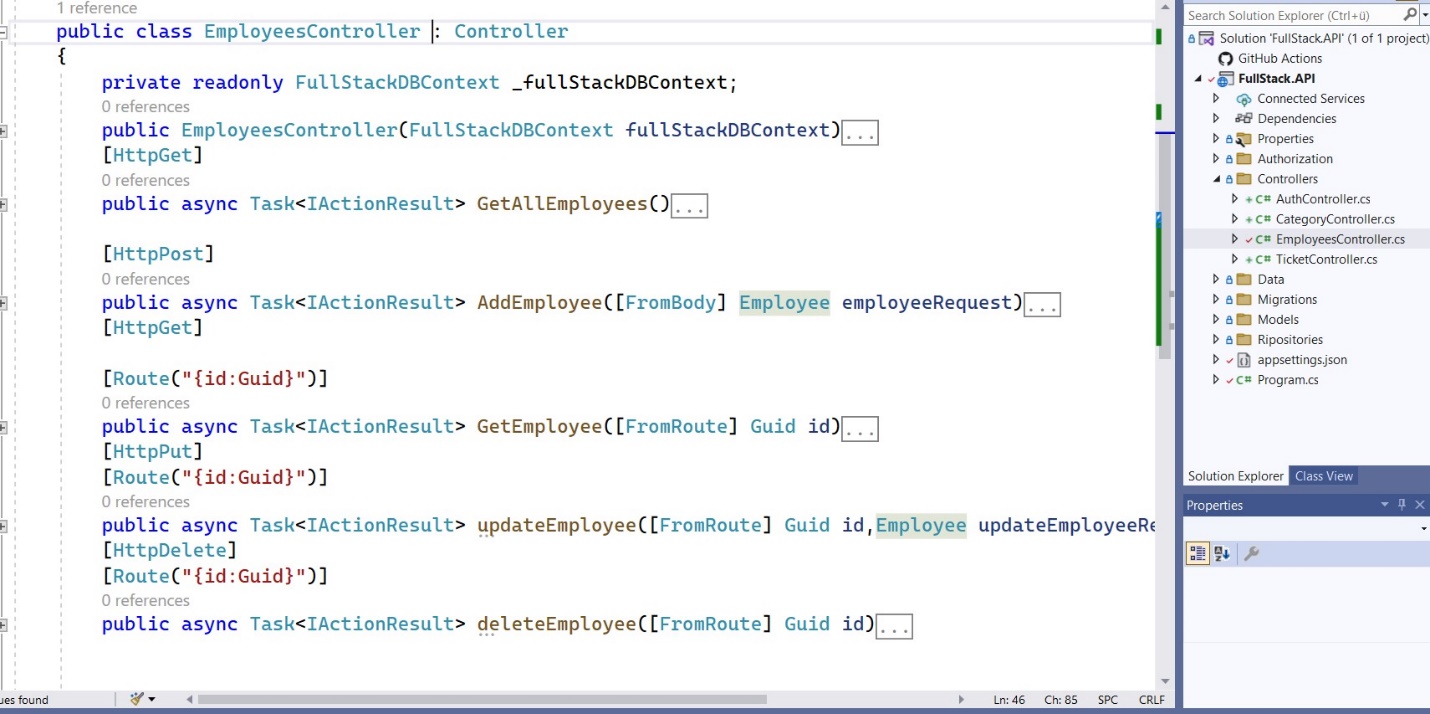
## Figura 2 – Auth0



# Configurare mediu dezvoltare

Pentru a putea dezvolta o aplicatie fullstack, e nevoie de instalarea unui IDE, cum ar fi Visual Studio si SQL Server pentru baza de date. Angular foloseste Node.js si npm (Node Package Manager) pentru a gestiona dependentele si pentru a construi pachetul. Angular CLI (Command Line Interface) este o unealta care ajuta pentru gestionarea proeictelor Angular. Comanda in Visual Studio Code este de forma: npm install -g @angular/cli. Pentru inceput se creeaza un proiect nou in Visual Studio de tip Asp.Net Web Api pe partea de server. Aici se vor defini operatiile de tip CRUD (Create, Read, Update, Delete). Aplicatia pe partea client in Angular se va conecta la server prin efectuarea de cerer HTTP, pentru a obtine si trimite date, utilizand HttpClient si ApiController in C#.

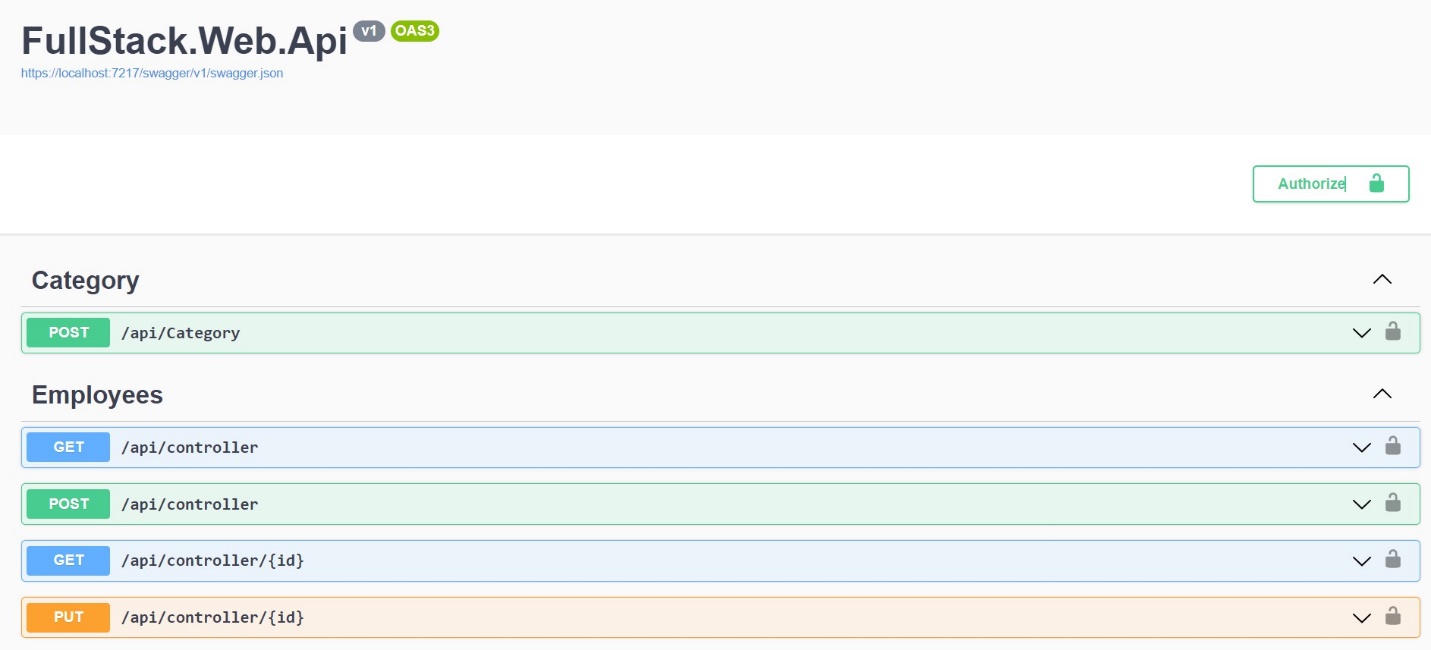
## Figura 3. CRUD in C#



# Dezvoltare Aplicatie

In cadrul dezvoltarii aplicatie pe partea de server in C#, am creeat un Model, care reprezinta o clasa, in cazul de fata **Employee,** in care vor fi definite toate proprietatile. Cu ajutorul pachetului *MicrosoftEntityFramework,* se vor genera automat tabelele in SQL Server, cat si constrangerile acestora. Implementarea functionalitatilor CRUD se definesc la sectiunea Controllers dupa cum se poate observa si in figura 3. Pentru a testa si gestiona partea de backend, am utilizat Swagger, care este un limbaj de descriere a interfetei API folosing un sistem JSON. De aici se pot vedea datele din baza de date, adauga un angajat nou sau chiar sterge orice informatie.

## Figura 4 – Swagger



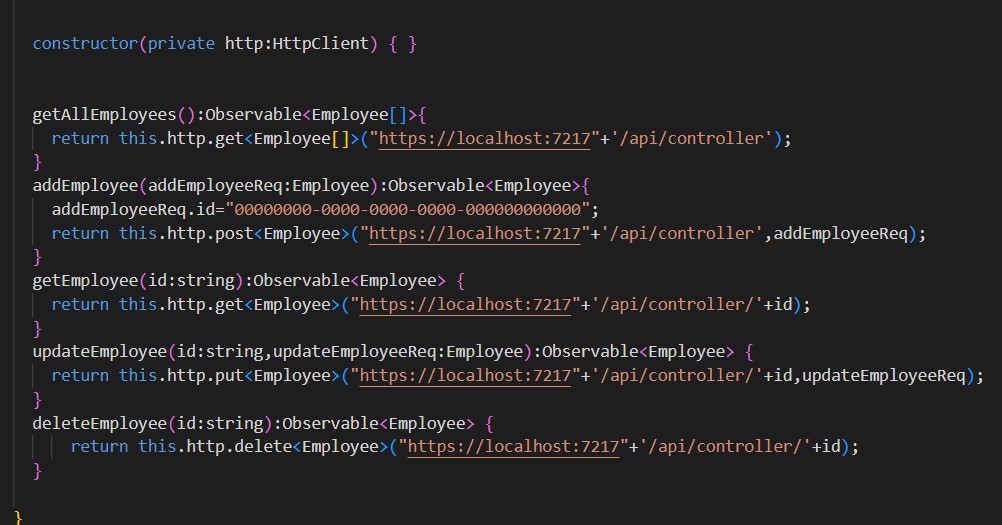
## Configurare Angular

In ceea ce priveste dezvolatarea aplicatiei pe partea de frontend, s-a utilizat Angular care este o platforma de dezvoltare web bazata pe limbajul Typesript. In Visual Studio Code, pentru a genera un nou proiect Angular, trebuie executat in Terminal comanda: ng new nume\_proiect. Se vor genera automat 3 fisiere dupa cum urmeaza in exemplul urmator:

1. *App.component.ts* (Acest fisier include de obicei logica componentei respective, se importa anumite dependinte si noi componente. De asemenea se pot defini proprietati si metode noi ).
2. *App.component.html* ( Acest fisier contine sablonul HTML al componentei si defineste cum arata componenta in interfata utilizatorului).
3. *App.component.css* ( Acest fisier contine stilurile pentru o anumita componenta, care vor fi aplicatei fisierului .html).

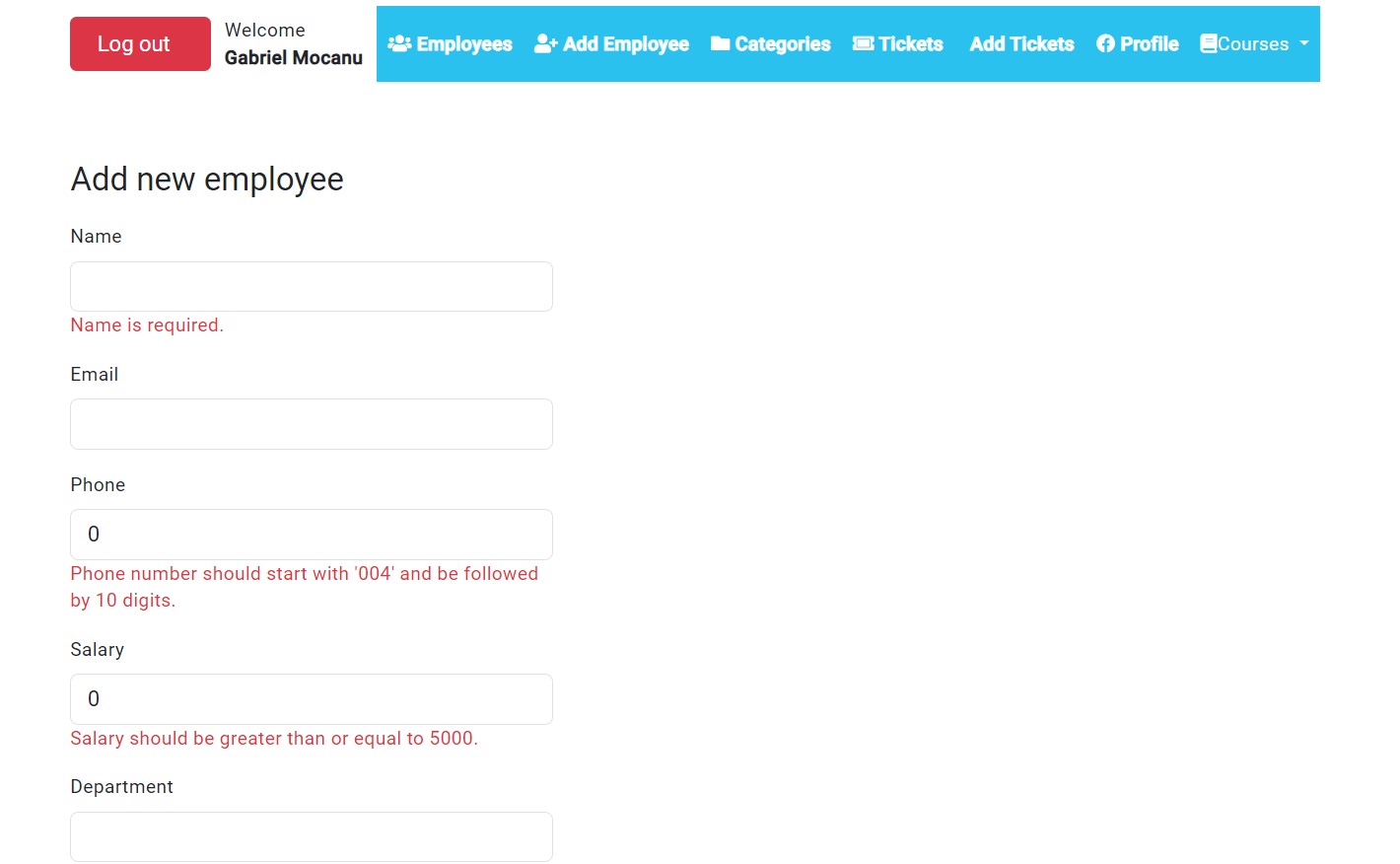
Se vor utiliza servicii pentru a se grupa mai multe functii comune pentru diferite componente prin comanda ng generate service nume\_serviciu. In figura 5 este reprezentat fisierul employees.service.ts in care sunt definite functiile de baza (citire,updatare,editare si stergere) cu ajutorul modulului HttpClient, prin care vom avea acces la partea de server prin adresa https:// localhost:7217/api/controller .

## Figura 5 – Functii CRUD



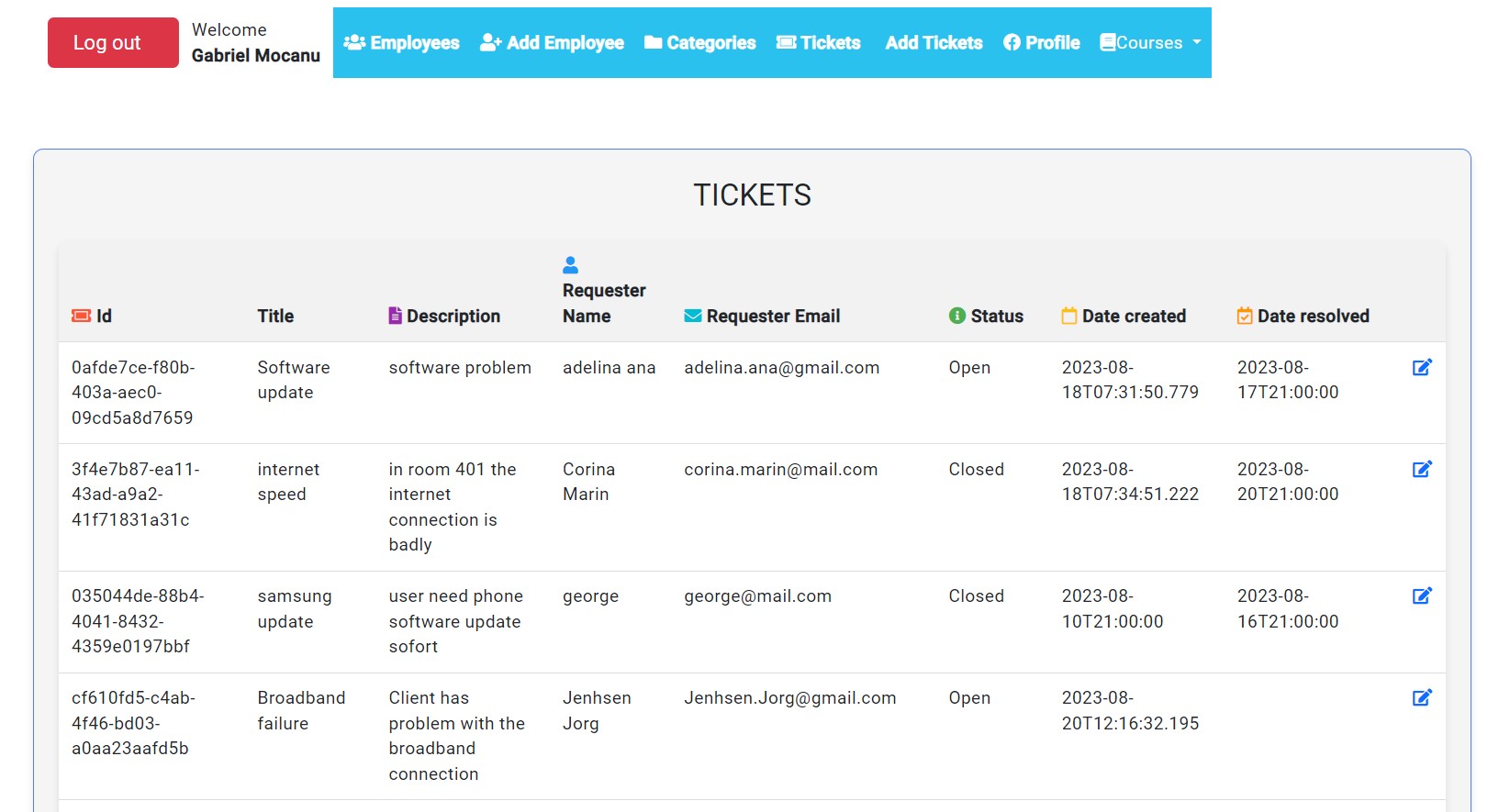
In *figura 6* este descrisa componenta **add.employees,** in care utilizatorul aplicatiei poate adauga un nou angajat. La introducerea noilor date, campul ID se va genera automat iar celelalte campuri detin restrictii pentru validarea datelor.

## Figura 6 – Adauga angajat



O alta componenta de baza utilizata in aplicatie este **Tickets** , in care angajatii companiei preiau date de la clienti despre natura problemei tehnice. Aceasta poate fi rezolvata pe loc sau ulterior pentru o problema mai delicata, astfel status-ul tichetului va fi deschis pana ce va fi rezolvat.

## Figura 7 – Tichete



# 4 . Concluzii

Aceasta aplicatie reprezinta un inceput si o parte introductiva pentru o dezvoltare ulterioara mai complexa. In decursul dezvoltarii aplicatiei m-am confruntat cu diferite probleme si bug-uri, care m-au ajutat sa ma dezvolt, sa invat din greselile altor programatori incepatori. Obiectivele proiectului au fost atinse in mare parte, exceptand parte de autentificare in care rolurile de admin si user normal, nu au fost dezvoltate corespunzator pentru restrictionarea accesului la anumite date.