Documentatie Proiect

Energy utility platform

Student: Buzila-Garda Andra-Maria

Profesor indrumator: Liana Toderean

1. **Descriere**

Acest proiect a fost creat in 3 etape:

* Etapa 1 (Tema 1): Am creat o aplicatie in care utilizatorul este de 2 tipuri: utilizator simplu sau administrator. In functie de tipul acesta, user-ul poate executa diferite operatii. Ca si user poti vedea device-urile, respectiv un chart cu energia consumata de respectivul device. Ca si administrator ai posibilitatea de a efectua anumite operatii asupra userilor: vizualizare useri, adaugare useri, modificare useri, dar si stergere. Aceste operatii le poate face administratorul si asupra device-urilor. Andministratorul isi poate vedea device-urile, respectiv energia consumata intr-un chart.
* Etapa 2 (Tema 2): In partea a doua a proiectului am implementat, pe aceeasi aplicatie, o comunicare dintre client si server cu ajutorul tehnologiei RabbitMQ. La un interval de timp de 10 sec. serverul trimite clientului energia consumata de catre un device anume la momentul respectiv. Am creat o pagina web unde apar diferite notificari in functie de energia consumata. Am facut o comparatie intre energia consumata si energia maxima. Coada a fost salvata in cloud si astfel s-a putut realiza deploy-ul.
* Etapa 3 (Tema 3): Aceasta ultima etapa am integrat-o in acelasi proiect cu celelalte 2 parti. In etapa 3 am creat un chat cu ajutorul webSockets. Am creat o conexiune si am trimis mesaje pentru toti utilizatorii care se afla pe un canal, sau am reusit sa trimit chiar mesaje private. Am implementat de asemenea si functionalitatea de typing, in momentul in care utilizatorul scrie altui utilizator, cel de-al doilea are posibilitatea de a vedea acest lucru.

Utilizatorul nu poate naviga in contul administratorului prin ruta si nici invers. O alta

parte ce tine tot de Securitate este faptul ca parola este criptata cu ajutorul unui salt. Am folosit BCrypt pentru a realiza acest lucru.

Graphical user interface, application, Word

Description automatically generated

Fig. 1. Pagina web pentru chat

A screenshot of a computer

Description automatically generated with medium confidence

Fig. 2. Pagina de administrare a userilor si a device-urilor

1. **Arhitectura conceptuala**

Pentru Tema 3 am initializat un webSocket in backend cu ajutorul caruia am putut sa creez topicuri sau canale in functie de ce am avut nevoie. In aceasta aplicatie am creat un canal general unde s-a putut conecta toata lumea care are cont in aplicatie si cate un canal privat intre utilizatori. Cee ace se intampla in spate: Utilizatorul trimite mesaj (request) catre server pe un topic, iar serverul trimite clientului inapoi mesajul.

WebSockets

Backend app

Response

Request

Frontend

Client

1. **Deploy**

Deploy-ul la Tema 3 a fost facut in acelasi mod ca si la celelalte teme si anume local in docker. Am creat Dockerfile si docker-compose si am scris instructiunile pentru crearea imaginilor respectiv a containerelor.

Docker

localhost:3000

Frontend

Backend port:8083

Postgres port:5432

Diagrama de deploy