

Proiect Licenta

Mustafa Taner, 341C1

Aplicatie WEB de masurare si monitorizare performanta aplicatii mobile iOS. Metricile masurate: timpul de pornire al aplicatiei, memoria RAM maxima utilizata si marimea aplicatiei pe disc.

Functionalitate similara cu [Jenkins](#) si [Looker](#) si cu baza de date proprie.

Tehnologii folosite in implementare

React, NodeJS, HTML, CSS, Javascript, Python, Flask, baza de date SQL (TBD), Docker, Github/Gitlab, Xcode, iOS.

Frontend - Structura Aplicatie Web

Paginile si continutul lor

- **SignUp**

- Email;
- Username;
- Parola;
- Confirmare parola;
- Tip de utilizator:
 - Developer;
 - Guest - poate fi creat doar cu un token provenit de la un developer;
 - Admin - poate fi creat doar cu un token provenit de la ownerii aplicatiei.

- **LogIn**

- Email / Username;
- Parola.

- **About**

- Informatii despre aplicatia web;
- Date de contact.

- **Home**

- **Buton de creare workspace nou** (doar pentru developer/admin), care duce catre o pagina ce contine:
 - **Formular** despre aplicatia care va fi testata, cu urmatoarele campuri de completat:
 - **Denumiri:**
 - Nume aplicatie;
 - Nume arhiva pentru instalare;
 - Bundle id;
 - **Descriere succinta aplicatie;**

- **Link catre repository**-ul unde este codul sursa al aplicatiei;
 - **Formatul logurilor de start & exit** pentru a masura timpul de pornire.
- **Lista aplicatii monitorizate.** Pentru fiecare aplicatie, exista o pagina / un workspace care contine:
 - Sectiune “**About**” - informatii despre aplicatie;
 - Buton de **Stergere Workspace**;
 - Sectiune **Rulare Job**. Un formular cu urmatoarele campuri:
 - PR id-ul pe care ruleaza testele (0 daca e master branch);
 - Linkul catre aplicatia compilata care va fi testata;
 - Buton de trigger job.
 - Sectiune cu **Joburi Rulate** - lista cu toate testele de performanta rulate pe aceasta aplicatie. Fiecare job duce catre o pagina care contine urmatoarele informatii:
 - Rezultatele rularii jobului (*metricile obtinute*);
 - Console output-ul rularii;
 - Buton de stergere job;
 - Sectiune cu **Grafice de Performanta**, unde se vad trendurile metricilor de performanta, rezultatele in urma rularii testelor:
 - Grafice separate pentru:
 - Timpul de pornire;
 - Consum de memorie RAM;
 - Marime aplicatie pe disc;
 - Filtre aplicabile pe grafice:
 - Dispozitivul pe care au rulat testele;
 - Scenariile de testare;
 - Tipuri de pornire ale aplicatiei;
 - Branch-ul pe care au rulat testele (*master sau PR*).

● Administrare

(doar pentru admini)

- **Lista** cu **toti utilizatorii** si optiunea de **stergere utilizatori**;
- **Grafice** pe baza activitatii utilizatorilor:
 - Ore la care e cea mai mare activitate pe site;
 - Timpul mediu petrecut de utilizator intr-o sesiune;
 - Numarul de rulari de teste de performanta per user.

● Setari

- Buton **Logout**;
- Buton **Stergere Profil**;
- Meniu **Schimbare Limba**;
- Meniu **Schimbare Tema**.

Backend - servicii utilizate

- **Autorizare**
 - Serviciu separat care se ocupa de autorizare utilizatori si comunica cu serviciul de frontend logic prin tokeni;
 - **Implementare:** Javascript, Docker.
- **Backend Logic**
 - Serviciu care se ocupa cu actiunile de UI realizate de utilizator, afisare pagini etc.;
 - Comunica cu Business Logic pentru a ii cere sa ruleze teste de performanta si pentru a primi rezultatele sub forma de JSON;
 - **Implementare:** Javascript, Docker.
- **Business Logic**
 - Serviciu care primeste request-uri de rulare teste de performanta de la Backend Logic, ruleaza testele si trimite rezultatele inapoi sub forma de JSON;
 - **Implementare:** Python, Flask, Xcode Command Line Tools, Docker;
 - Hostat neaparat pe Apple hardware.
- **Baza de date**

Stocheaza:

 - Informatii despre utilizatori:
 - Date profil;
 - Activitate pe aplicatia web;
 - Metricile colectate in urma rularii testelor de performanta ;
 - **Implementare:** TBD, Docker.
- **UI Baza de date**
 - Utilitar de gestiune a bazei de date;
 - **Implementare:** TBD, Docker.

3rd party-uri folosite cu REST API

- **Google Drive**
 - Business Logic comunica cu Google Drive pentru a stoca logurile testelor de performanta;
 - Backend Logic comunica cu Google Drive pentru a descarca logurile (primind link-ul catre ele in raspunsul de la Business Logic) si a le afisa in job.

- Github/Gitlab
 - Folosit pentru deployment automat al aplicatiei web;
 - Backend Logic posteaza pe PR-ul completat in formularul de rulare al jobului un comentariu cu rezultatele testelor de performanta.

User flows (cateva exemple):

Varianta 1 (Developer):

1. Signup;
2. Login;
3. Creare workspace;
4. Rulare test pe aplicatie;
5. Vizualizare loguri;
6. Click pe link catre PR-ul unde s-au postat rezultatele;
7. Logout.

Varianta 2 (Guest):

1. Login;
2. Cautare aplicatie in lista celor existente;
3. Vizualizare grafice de performanta;
4. Logout.

Varianta 3 (Admin):

1. Login;
2. Stergere useri inactiv;
3. Logout.

Diagrama Servicii

