# 

# Afaceri electronice

# Proiect Modul 2

Vlad Oana-Andreea

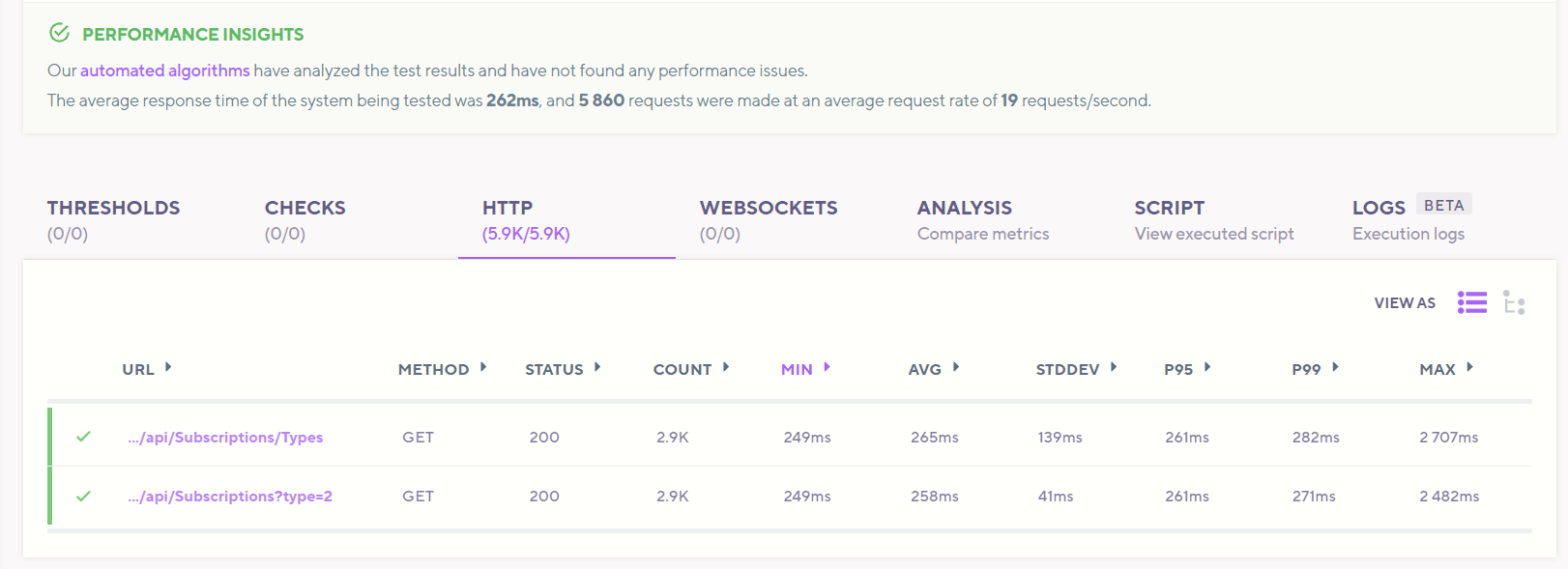
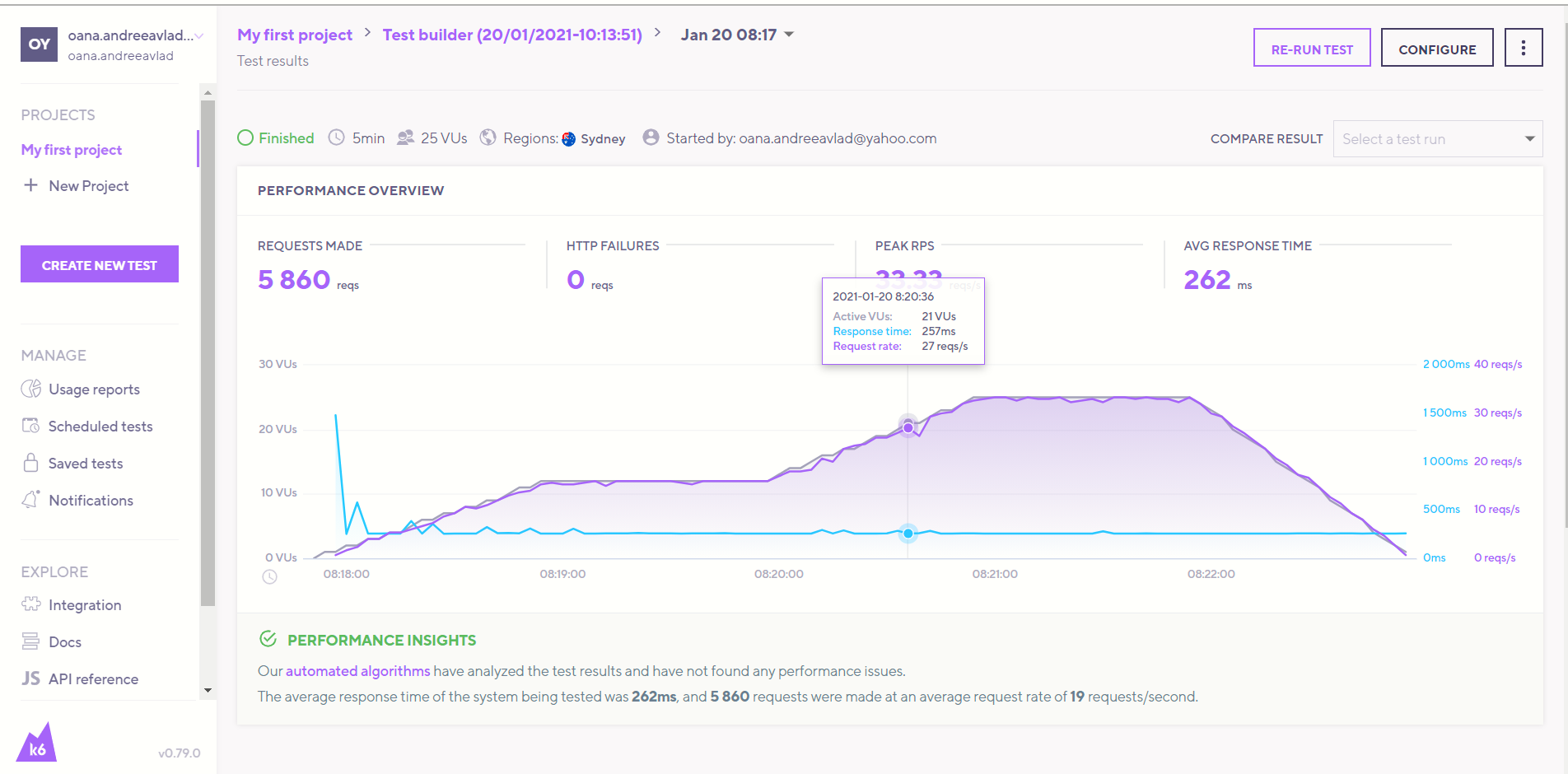
# Audit de performanta

Proiectul urmareste realizarea unei aplicatii web cu arhitectura SPA care sa reprezinte o pagina de achizitie client (landing page). Entitatea comerciala se ofera bunuri si servicii in domeniul plantelor si remediilor naturiste, punand la dispozitie abonamente ce pot fi achizitionate online.

In realizarea analizei performantei aplicatiei web, se va utiliza platforma K6.io si tool-ul Google Lighthouse cu scopul de a evalua eficienta API-ului aplicatiei, timpilor de incarcare a site-ului si comportamentul acestuia pe device desktop versus device mobil.

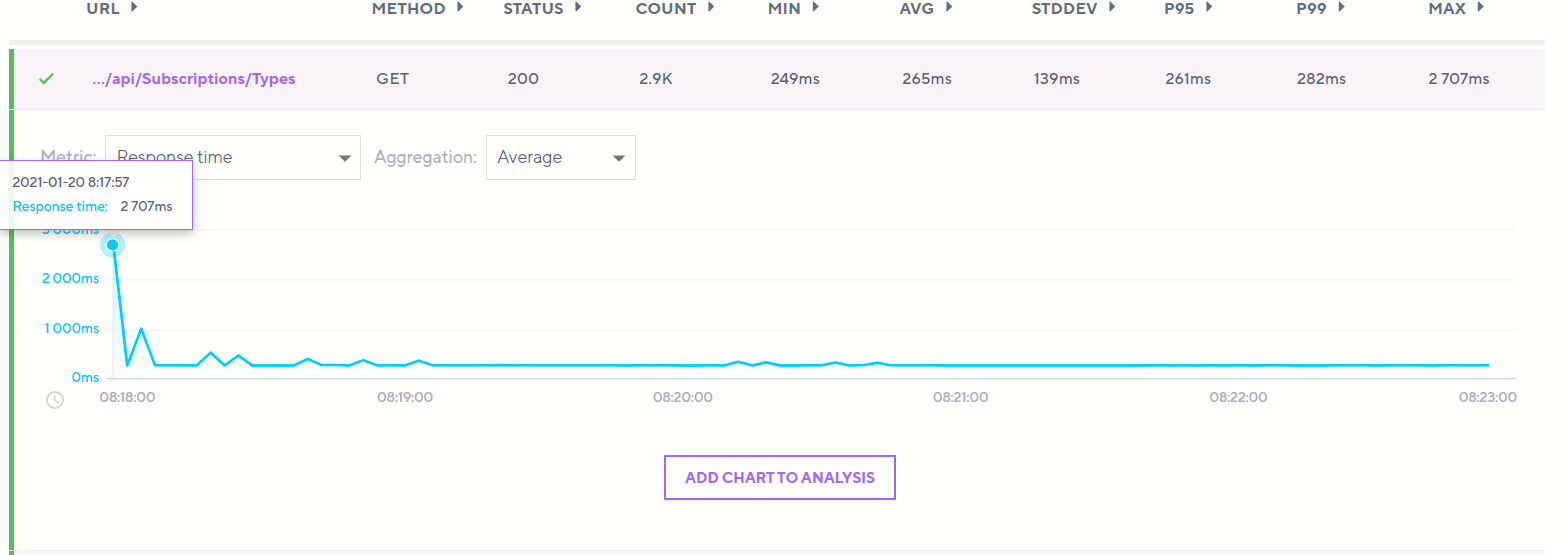
## K6.io

1. Testarea timpilor de raspuns pentru interogarea datelor din pagina principala.



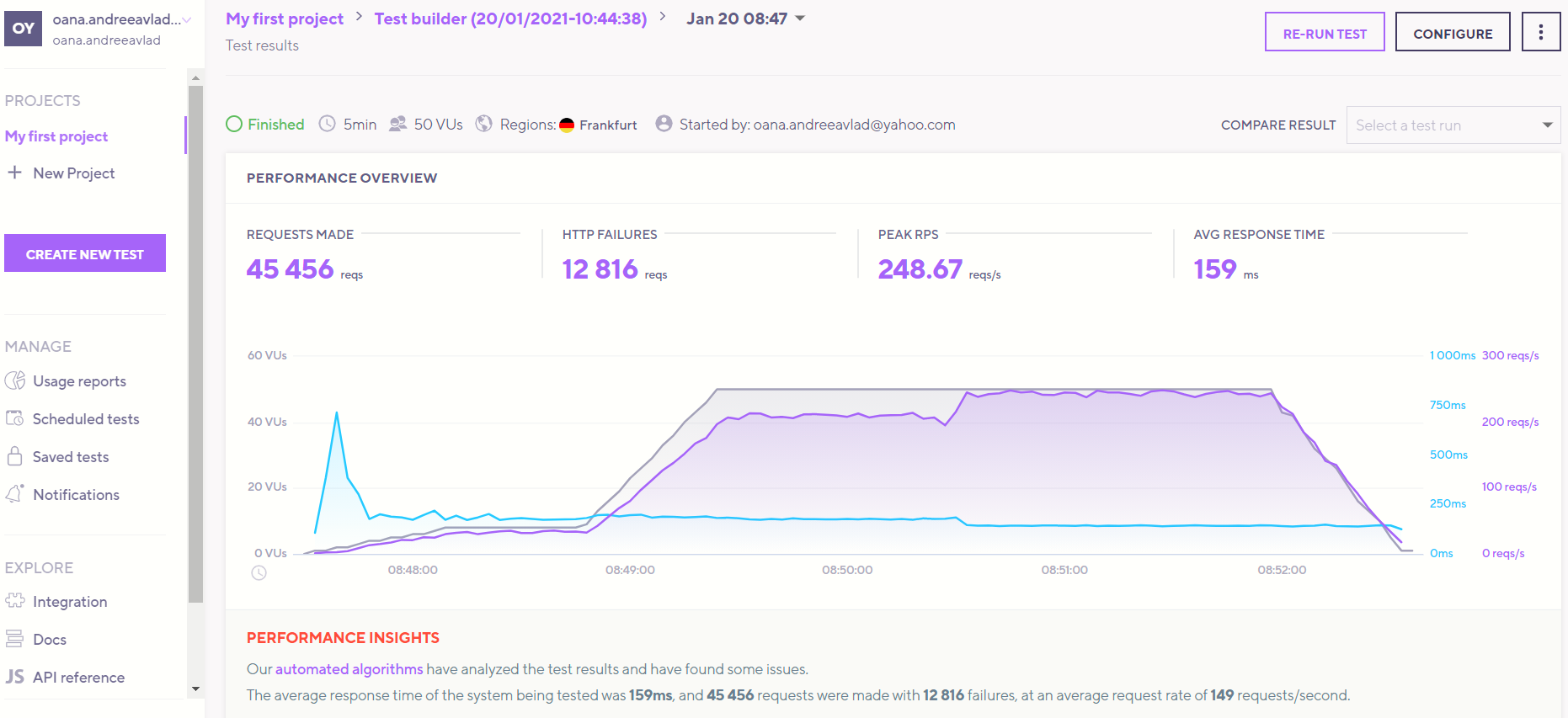
Pentru a analiza de raspuns a interogarii datelor se va face cate un request spre 2 endpoint-uri ale API-ului aplicatiei, returnand categoriile de abonamente disponibile cat si abonamentele existente in categoria Home.

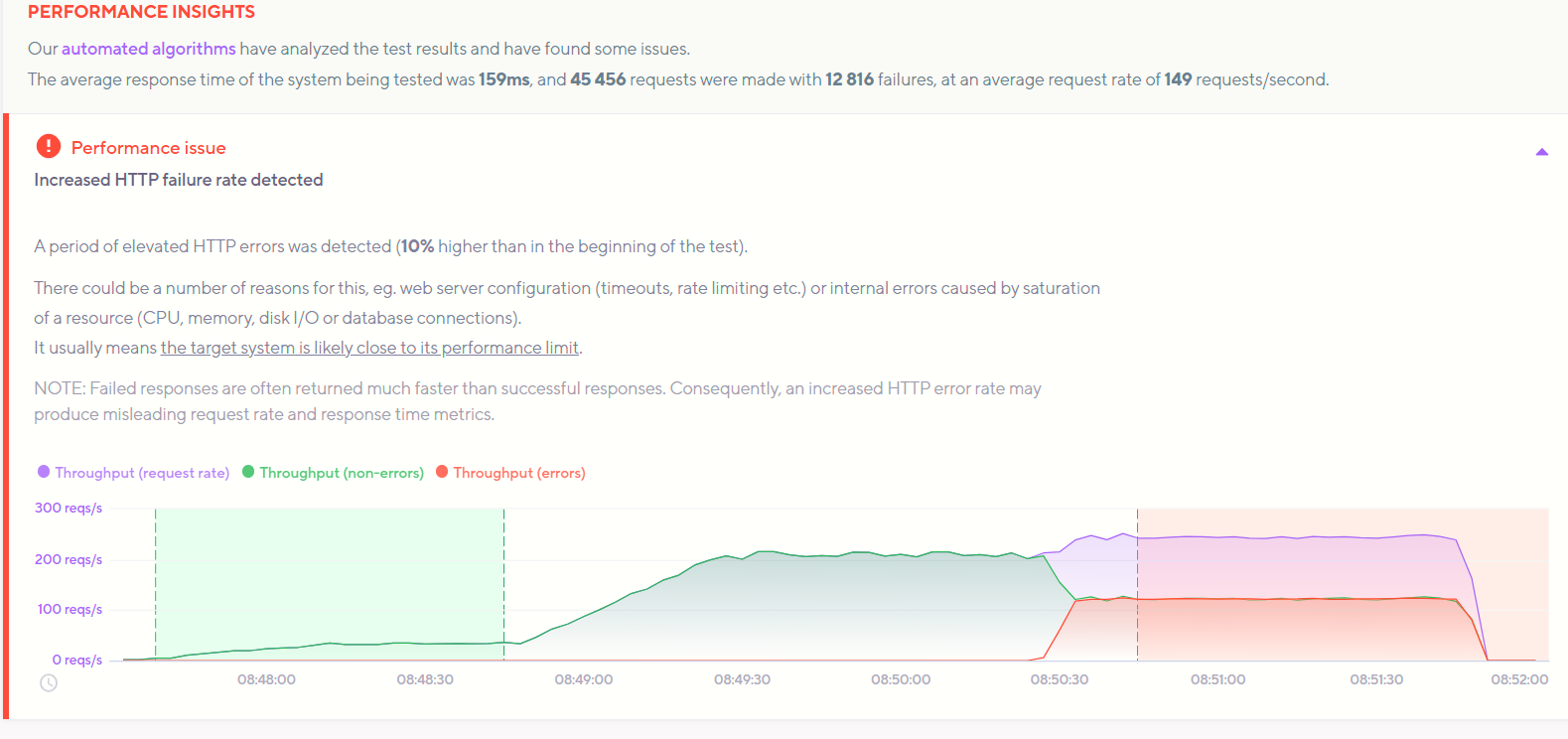
Considerand un test de tip stress testing executat timp de 5 minute cu maximum 25 de utilizatori concomitenti la un moment dat se poate observa o rata de raspuns preponderent constanta. In special un ultimele 4 minute de executie, 99% din utilizatori pot obtine datele despre categorii in aproximativ 271 milisecunde, iar datele despre abonamente in 282 milisecunde.

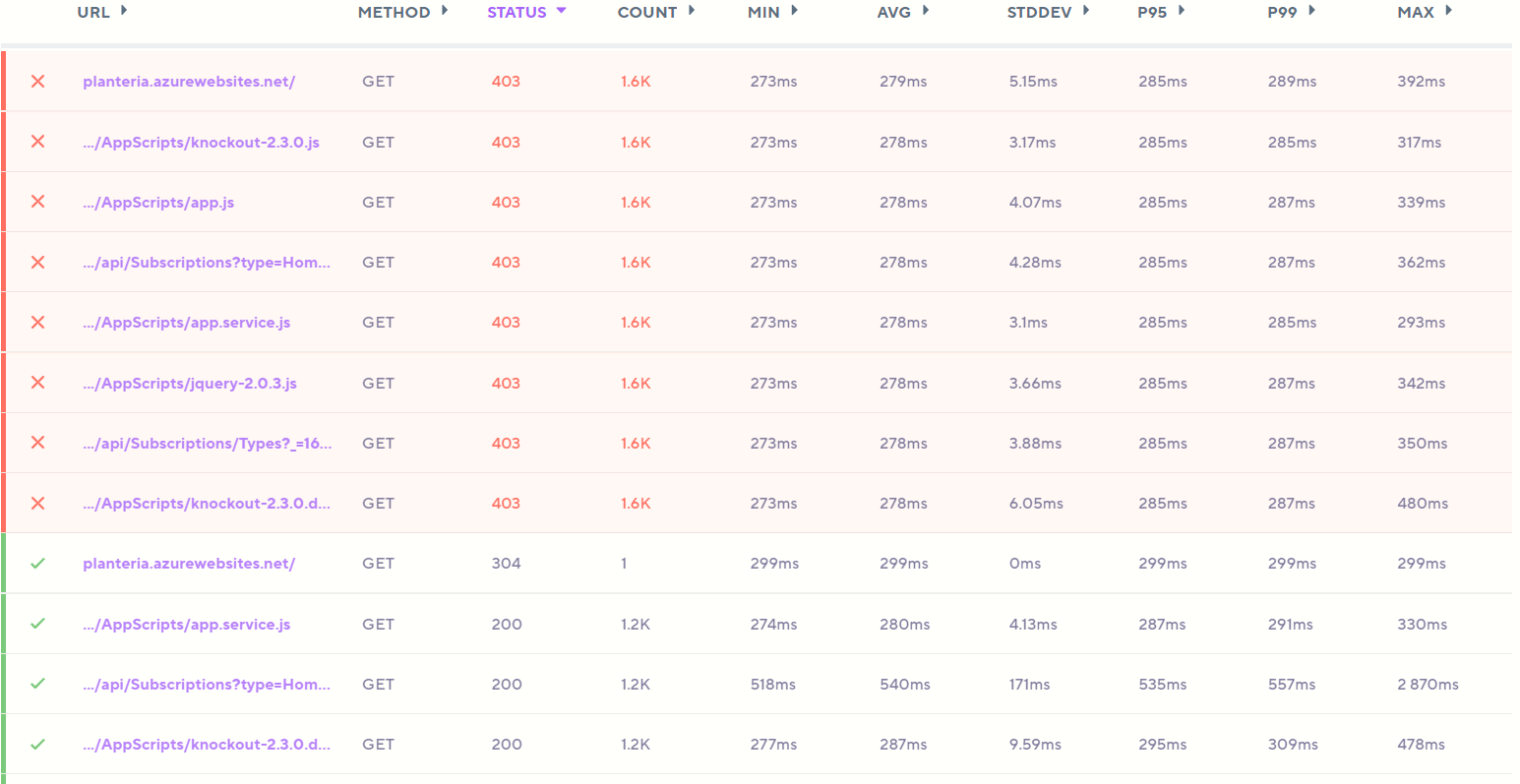


Se poate observa ca valorile maxime inregistrate (MAX) au loc la inceputul analizei, moment in care numarul de utilizatori incepe sa creasca progresiv. O posibila cauza este utilizarea de catre server-ul Azure a unui thread pool care are capacitatea de a se auto-regla in functie de gradul de utilizare al CPU sau al altor resurse prin adaugarea sau eliminarea unor thread-uri. Cand numarul de request-uri creste si nu exista suficiente thread-uri pentru a evita procesarea acestora in coada, acest thread pool va analiza starea resurselor si poate adauga noi thread-uri pentru o gestiune mai eficienta. Desigur, acest lucru implica un cost al initializarii, fapt care determina izolat un timp mai mare de asteptare urmat de o revenire in viteza procesarii request-urilor.

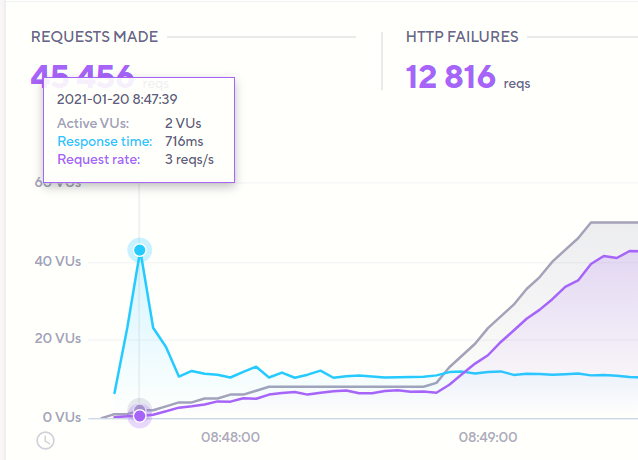
1. Testarea performantei incarcarii site-ului prin arhiva .HAR (Spike testing)







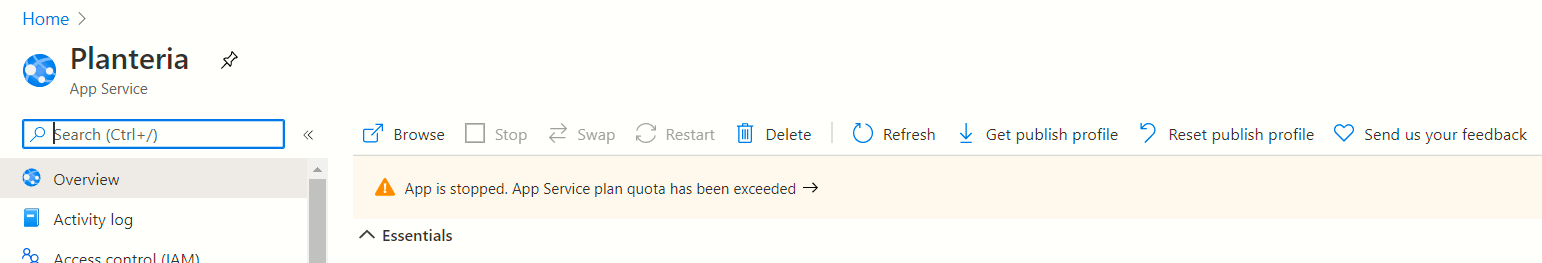
Testul de evaluare a performantei intregului site a fost executat in stil Spike Testing cu un maxim de 50 de utilizatori, pe o durata de 5 minute, implicand toate request-urile efectuate de aplicatie atat pentru accesarea resurselor (statice sau externe), cat si pentru a returna date prin intermediul API-ului.

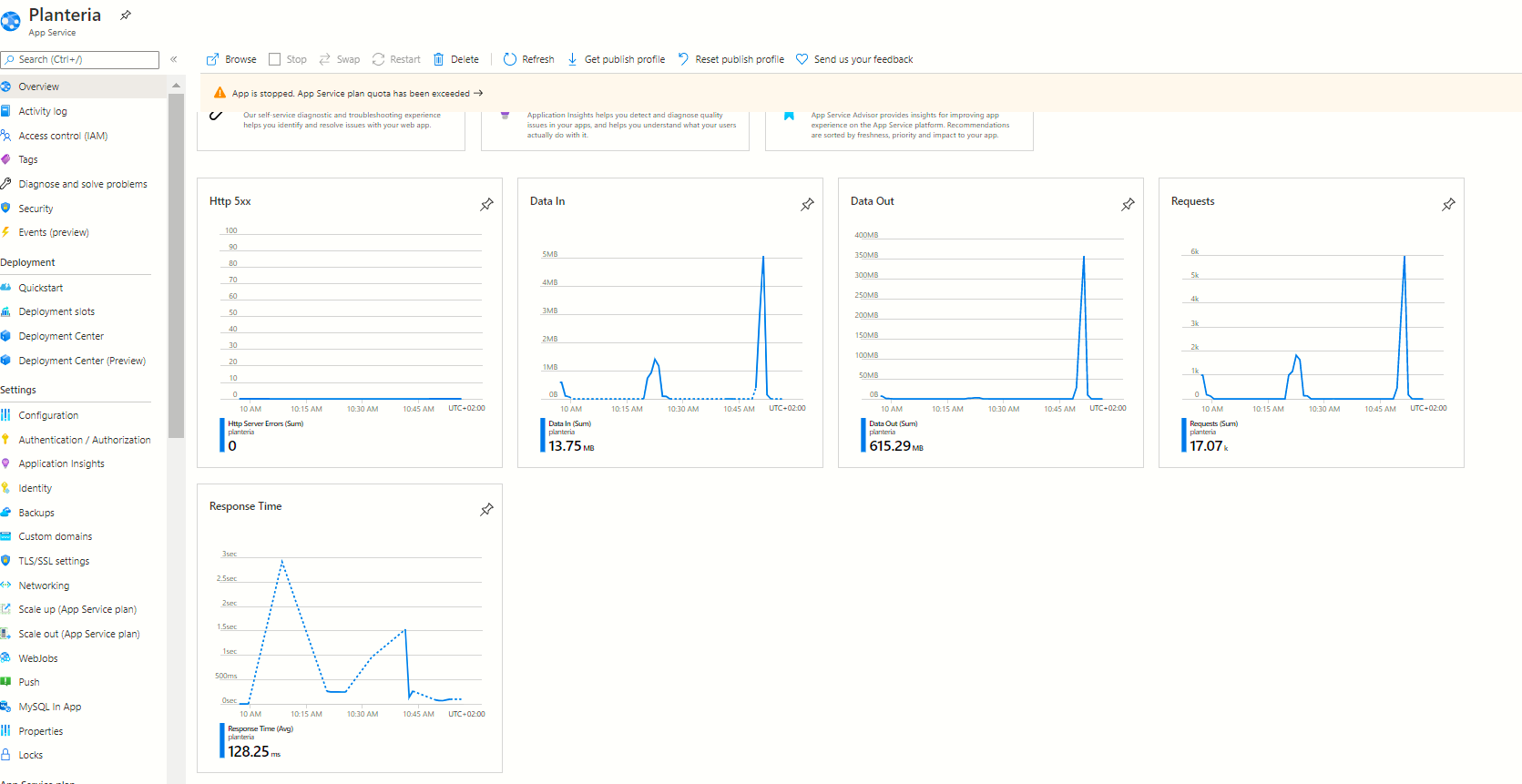


Ca si in testul anterior, se observa un varf al timpului de asteptare in inceputul executiei datorat aceluiasi cost al optimizarii server-ului, punct din care numarul de cereri creste iar viteza de raspuns se imbunatateste considerabil.

De asemenea, informatiile oferite de analiza executiei indica spre finalul testului faptul ca request-urile au avut o rata cu 10% mai mare de esec decat intr-un stadiu incipient. Cand numarul de utilizatori si request-uri concomitente tind spre maximum, se constata saturarea utilizarii de resurse (CPU, memorie, conexiunea si citirea din baza de date) cee ace a condus la o crestere in raspunsuri cu status de eroare.

Asa cum am intuit, portalul Azure indica faptul ca pentru serviciul de hosting Web App Service s-au atins limitele de utilizare si ca este necesara scalarea serviciului pe un plan de utilizare cu mai multe resurse alocate (plan quota).

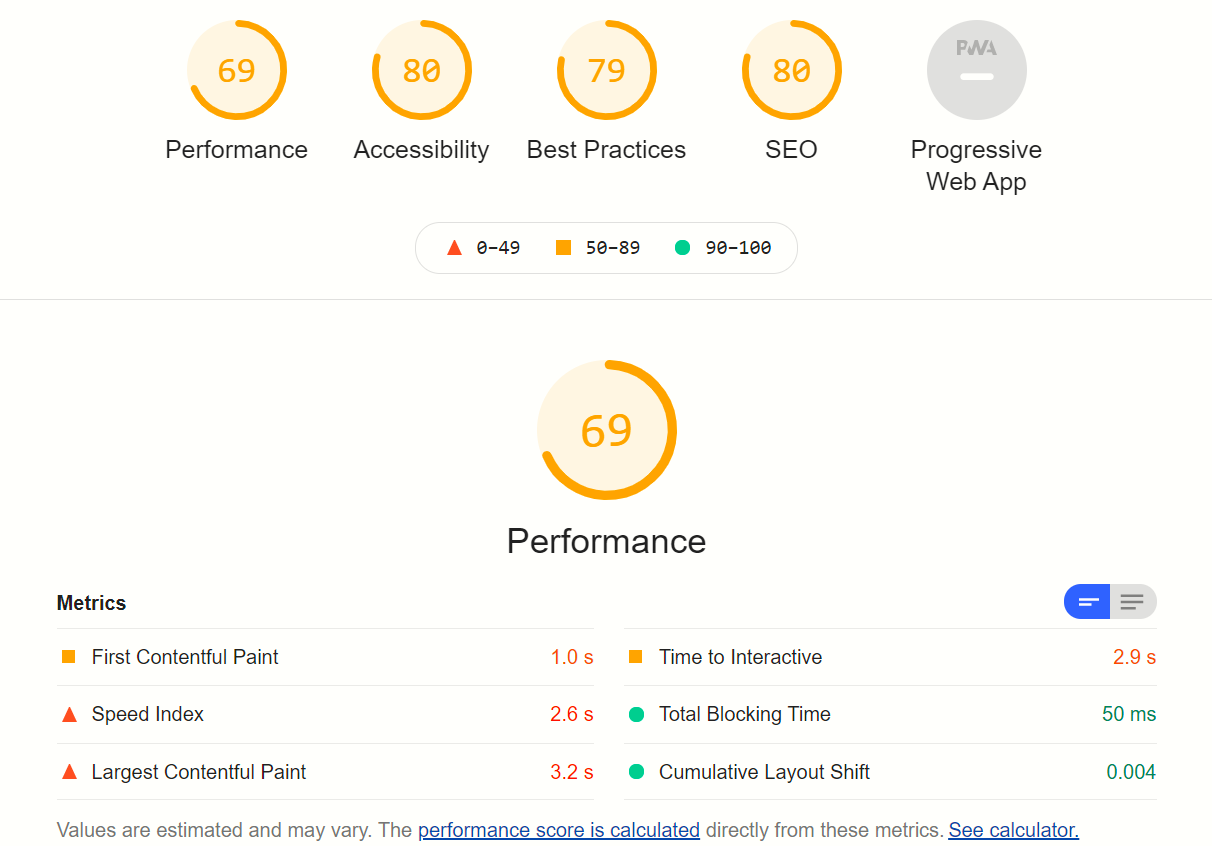
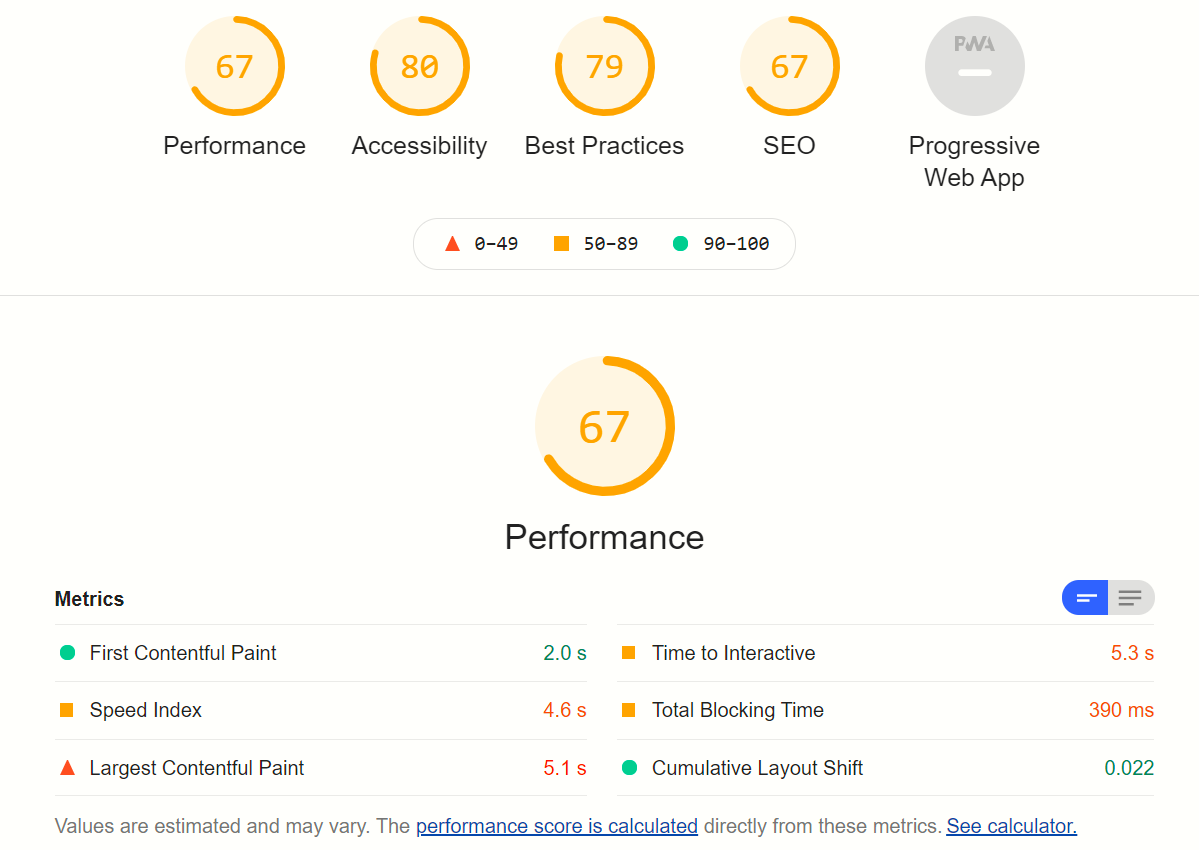




## Google Lighthouse

Pentru a testa performanta site-ului atat pe desktop cat si pe dispozitive mobile s-a folosit instrumentul de audit Google Lighthouse. Acesta analizeaza performanta, accesibilitatea, aplicabilitatea bunelor practice in dezvoltarea website-ului si gradul de optimizare pentru motoarele de cautare (SEO).

Scorurile obtinute de aplicatie in executie desktop versus mobile sunt foarte apropiate (69, respectiv 67) si incadreaza metricele obtinute in zona de performanta medie.

Figură - Audit desktop Figură - Audit mobile

Principalele elemente cu impact negativ asupra performantei sunt legate de viteza de randare a componentelor HTML in pagina. Accesarea resurselor pentru styling (Bootstrap, font-awesome) se face inlantuit si cauzeaza blocarea incarcarii paginii pana ce acestea nu sunt primite. In acelasi timp, perioada de incarcare al continutului media (imaginile categoriilor de abonamente) este marita din cauza ca acesta este extern si necesita prelucrari aditionale inaintea aplicarii regulilor de styling precum ajustarea dimensiunii. (Largest Contentful Paint: 3.2-5.1 secunde)

Acest lucru contribuie la un timp cuprins intre 2.9-5.3 secunde pana ca utilizatorul sa poata interactiona cu continutul paginii.

Din punct de vedere al bunelor practici de dezvoltare a aplicatiilor web, auditul evidentiaza faptul ca site-ul nu este protejat prin protocolul HTTPS si ca sunt folosite librarii front-end de JavaScript cunoscute pentru vulnerabilitati de securitate.

## Propuneri de imbunatatire

Pe baza rezultatelor testelor si auditului de performanta, se pot considera urmatoarele masuri imbunatatire:

* Protejarea traficului de date prin protocolul HTTPS
* Eliminarea resurselor blocante pentru randarea in pagina prin utilizarea unei librarii optimizate de front-end (Vue.js, React, Angular)
* Implementarea unei politici de caching pentru resursele statice
* Adaugarea tag-urilor de metadate pentru optimizare SEO
* Redimensionarea explicita a imaginilor si servirea acestora din resurse statice

## Concluzii

In urma auditului de performanta, site-ul de ecommerce propus prezinta deficiente in ceea ce priveste timpul de incarcare al elementelor in pagina, iar in situatia unor request-uri multiple in conditii de stres exista imposibilitatea de a raspunde cu succes cererilor.

Aceste probleme impun o refactorizare a interfetei printr-un framework specializat de front-end si considerarea unor modalitati diferite de accesare a resurselor (static versus dinamic).