



CHECKPOINT-2

PROJECT MDS - MAI 2025

SUBIECTELE DE ASTĂZI

1. **Ideea noastră**
2. **Arhitectura proiectului**
3. **Aspecte interesante & dificile**
4. **Ce rămâne de implementat?**

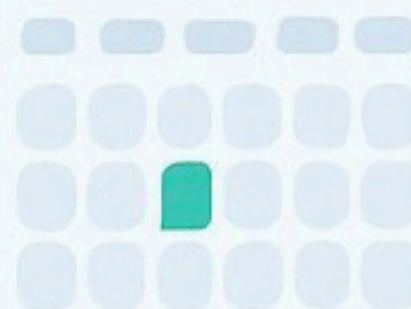


DriveFlow

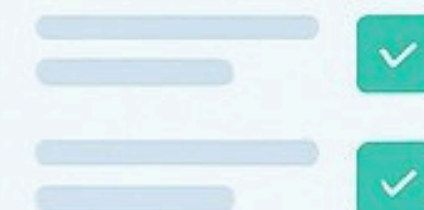
un singur CRM pentru toate școlile auto din rețeaua ta

DriveFlow este primul CRM creat special pentru școlile auto: înscrieri, programări la volan, dosare de examen și contabilitate – toate într-un singur tablou de bord.

Inscrieri



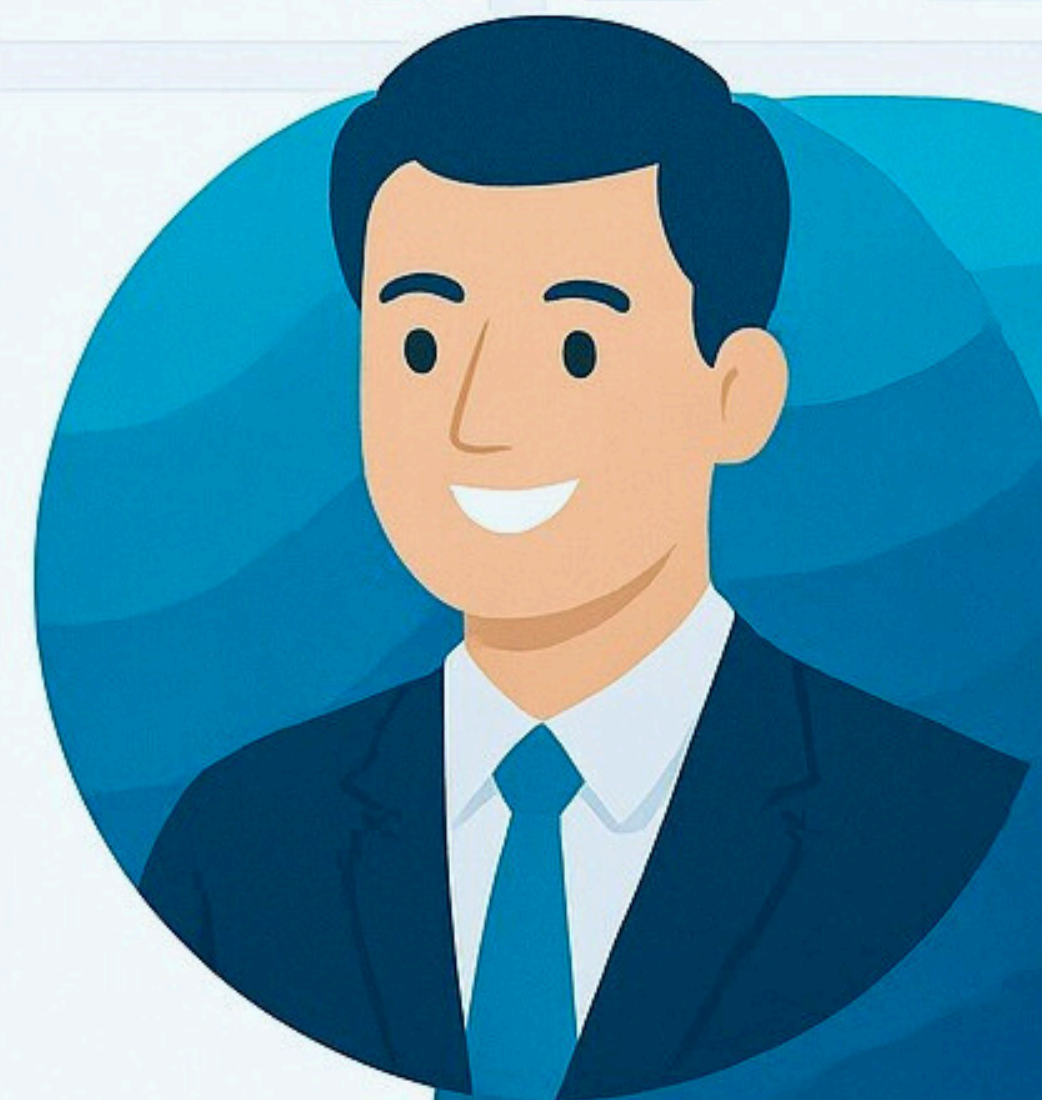
Dosare



Contabilitate

2.300 RON

Modulul Super-Admin ridică totul la puterea „overview”: acesta vede în timp real activitatea fiecărei școli și ia decizii pe loc, fără să mai deschidă alte platforme.



BACKLOG



01. Inscriere

În calitate de utilizator, doresc să mă pot autentifica pentru a accesa funcționalitățile aplicației; după autentificare, trebuie să fiu direcționat automat către un dashboard dedicat rolului meu (de ex. Instructor sau Administrator de Școală)



04. Contabilitate

Micro-serviciu Flask care, dintr-un POST JSON, emite factura PDF a cursantului și o livrează imediat platformei pentru vizualizare sau descărcare.



02. Gestionarea dosarelor

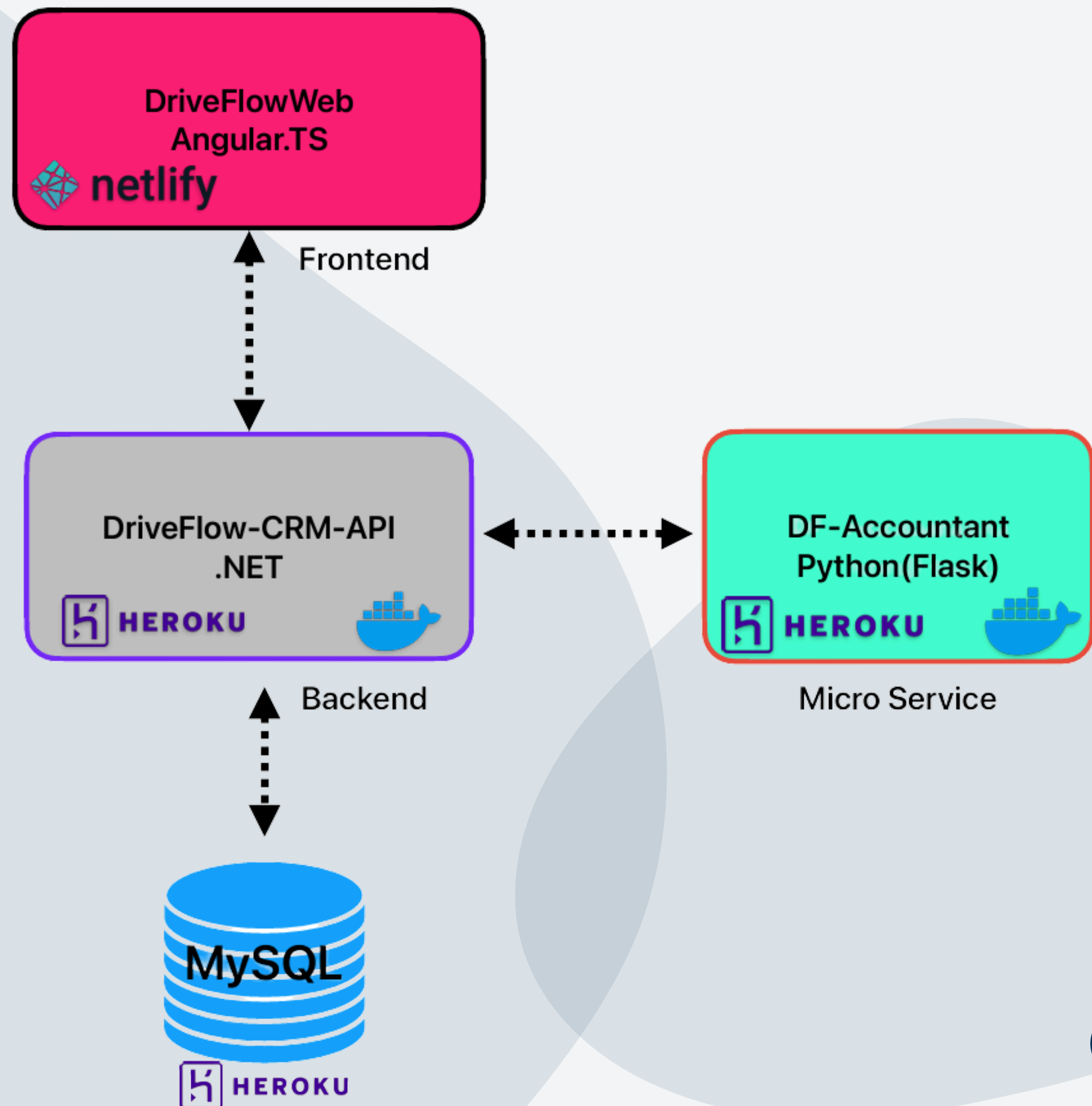
În calitate de Instructor, vreau să pot accesa progresul detaliat al unui cursant – datele personale, stadiul dosarului, plățile, cerințele pe categorie și istoricul orelor efectuate – pentru a monitoriza în timp real evoluția respectivului elev



03. Evidenta cursantilor

DriveFlow calculează lunar veniturile, cheltuielile și profitul fiecărei școli auto într-un singur raport PDF.

Arhitectura Proiectului



Aspecte dificile și interesante

Integrarea frontend-ului cu API-ul .NET configurând CORS, rute REST, serializare JSON și autentificare JWT, aplicând pattern-urile Factory, Builder, Strategy și Chain of Responsibility pentru un cod extensibil și curat.

Am containerizat aplicația și am migrat pe Docker + Heroku/Fly, completate de un pipeline CI/CD simplu și health-checks, optimizând costurile.

Structurarea proiectului în MVP, epics și pull-request-uri pe Git Flow, cu stand-up-uri zilnice și code-reviews, asigurând coerența și calitatea livrabilelor.

Ce rămâne de implementat?

What's next? BIG DATA

📖 Problema principală: Pentru a verifica disponibilitatea, trebuie realizate join-uri complexe între:

Appointment ↔ File ↔ Vehicle ↔ Appointment ↔ File ↔ Instructor ↔ InstructorAvailability ↔ Appointment ↔ File ↔ Vehicle

Implicații numerice:

Într-un an:

~10,000 programări

~20,000 înregistrări de disponibilitate

Verificarea disponibilității pentru o programare ar însemna join-uri cu mii de înregistrări

Soluția propusă

Adoptarea unei strategii de data warehousing