

Organizare

Intrebari:

Care sunt task-urile pe care le-am putea avea pe partea de frontend + backend?

Generarea de rapoarte

Cum ne impartim in mod cat mai egal task-urile (fiecare trebuie sa munceasca atat pe frontend cat si pe backend)?

In ce va consta proiectul?

- Scopul este de a oferi gestionarea cu usurinta a datelor despre gunoi. Acestea vor fi structurate pe categorii precum menajer, hartie, plastic si vor face referire la cetateni, persoanele din personalul autorizat si factorii de decizie (Instituti precum Primaria).
- Modul in care datele vor fi organizate se va baza pe generarea de rapoarte numerice sau grafice despre situatia actuala la nivel de cartier/localitate cu scopul de a atrage atentia asupra locurilor cele mai curate/mizere. Aceste rapoarte vor acoperi diferite unitati de timp (zile, saptamani, luni), fiind disponibile in formatele HTML, CSV si PDF.
- Obiectivul secundar este de a oferi suport pentru initierea campaniilor de atentionare a locuitorilor in ceea ce priveste **colectarea selectiva a gunoiului** si raportarea incidentelor ce tin de depozitarea gresita prin intermediul aplicatiei.
- Bonus: includerea unui simulator al desfasurarii proceselor de reciclare.

Cum se va realiza implementarea?

- Un serviciu GraphQL realizat in maniera personala
- Niciun framework nu va fi folosit, inclusiv Express. Partea de management a sesiunilor clientilor va fi facuta prin intermediul cookie-urilor. Similar, niciun SDK (de exemplu [googleapis - npm](#)) nu va fi folosit, totul facandu-se prin intermediul unui serviciu GraphQL sau REST cu ajutorul unei biblioteci precum libcurl.
- Modul in care aplicatia va fi structurata se va baza pe (micro)servicii. In plus, va trebui sa oferim un nivel de securitate care sa protejeze aplicatia de atacuri precum *Cross Site Scripting* sau *SQL injection*
- Aplicatia va avea un design responsive pentru a permite clientilor sa o foloseasca si de pe telefon
- Vom oferi posibilitatea de a efectua import/export pentru datele aplicatiei (minim csv)
- Va trebui sa avem un modul propriu de administrare a aplicatiei Web.
- Prioritati:
 1. Server
 2. Client
- Tehnologii folosite:

- Vom structura aplicatia pe baza design pattern-ului [MVC](#), care ofera modularitate si posibilitatea de a intretine codul cu usurinta
- Pentru a ne simplifica munca, vom folosi Scss, o extensie a limbajului css ce ofera o organizare optima a codului si un timp mai scurt de lucru, si Babel pentru a avea acces la functionalitatile moderne ale limbajului Javascript.
- Vom folosi suita de tehnologii [Ajax](#) pentru a realiza comunicarea intre server si client.
- Sqlite va fi utilizat pentru stocarea informatiilor.
- Pe partea de Frontend vom apela la limbajul de programare Javascript deoarece permite interactiunea cu clientul in mod dinamic. In ceea ce priveste partea de Backend vom folosi Node.js deoarece ofera viteza de raspuns pentru cererile clientilor.
- Solutia va contine un modul de administrare a aplicatiei ce va fi realizat prin fisiere de configurare
- O licenta libera ce va respecta termenii mentionati de [Creative Commons](#) va fi folosita pentru a expune intreg codul din care va fi formata implementarea acestui proiect.

De facut:

- Validare cod [html5](#) si [css](#) al clientului
-

Functionalitati esentiale !!!:

Cetateanul

Formular de depus plangere

Vizualizarea

rapoarte

Care sunt cele mai murdare locuri?

Cod qr pentru fiecare cos de gunoi

Personalul autorizat

Cod pt fiecare (numar de telefon ce va fi stocat in baza de date fara nici un fel de criptare)

Identificare

Harta

Angajati: unde trebuie sa mearga

Autoritati: care este situatia la nivel de regiune
sqlite