

Partypulse



GitHub URL

<https://github.com/mateicrainiceanu/partypulse>

Live DEMO

<https://partypulse-koha.vercel.app/register>

Povestea Partypulse

Despre idee

Partypulse este la baza o aplicație ce leagă DJ-ul și audiența sa într-un mod inedit, în care să existe comunicare. Practic, aplicația facilitează utilizatorului accesul la a comunica cu DJ-ul și de a cere melodia preferată într-un mod nou.

Cum a început?

Una din pasiunile mele, pe lângă programare este DJ-ing-ul. În timpul liber merg la majoratele prietenilor mei și pun muzică. Fiind o pasiune încă de mic, algoritmiile platformelor de socializare, m-au introdus în comunitatea online a DJ-ilor. Astfel am constatat că multă lume din breasla asta se confruntă cu problema cerințelor muzicale la evenimente.

De cele mai multe ori, lumea ne abordează pentru a pune o melodie. În mare parte melodiile cerute sunt pe lângă tema muzicală din acel moment, dar ne-ar prinde bine să le salvăm pentru mai târziu. Desigur, până tranziționăm dintr-un stil muzical în altul, uităm complet o recomandare de piesă care ar fi fost chiar utilă.

Așa că, am hotărât să construiesc o aplicație care să facă mult mai ușor procesul de a cere o melodie la un eveniment. De acolo, ideile s-au înlănțuit, și din acest simplu concept, am ajuns la o adevărată platformă de socializare, destinată participanților, organizatorilor și DJ-ilor la petreceri.

Inginerie web și programare

Aplicația este realizată folosind NextJS v14, integrând App Router-ul lor. Am ales această variantă, datorită sistemului de navigare și routare oferit de Next.

Pentru elemente de design ale aplicației am folosit MaterialUI, facilitând astfel un UI, cu care majoritatea utilizatorilor sunt deja obișnuiți.

Style-ing-ul general l-am realizat folosind TailwindCSS, cu care am lucrat pentru prima dată, dar am rămas surprins, de cât de ușor m-am obișnuit cu numele claselor și aplicarea acestora.

Pentru Version Management am folosit Git, făcând câte un commit în principal pe main, cu fiecare feature nou adăugat.

Pentru a menține performanța aplicației și a o dezvolta în continuare, eliminând în același timp din BUG-uri, pentru mare parte din etapa de development am folosit metoda celor 2 commit-uri: “Cu fiecare feature nou adăugat, în următorul commit va avea loc un BUG Fix al unui comportament observat în sesiunea de development aferentă acelui feature”.

Aplicația atinge loading-time-uri foarte bune, chiar < 1 sec în build-ul pentru production, ținând cont că se fac mai multe call-uri în API, respectiv multiple interogări ale bazei de date din multiple componente ale aplicației.

În prim plan a stat echilibrul între modularitate (reutilizarea componentelor) și performanța loading-ului.

Totodată, am folosit un Sistem de management al Spinner-ului într-un provider în layout-ul aplicației, care practic, adăuga multiple elemente de loading, și le scade în momentul când toate s-au terminat. Am simțit nevoia să implementez asta, deoarece în dev, Spinnerul, controlat de un context simplu se oprea prea repede - când primul element s-a încărcat sau existau efecte de flicker.

Funcționalități și utilitate

Platforma la bază oferă 3 tipuri de utilizatori: user, em (event manager), DJ.

- Userul poate aprecia alți utilizatori, salva locații, căuta evenimente, aprecia evenimente și poate marca participarea la un eveniment.
- EM-ul poate crea noi locații și adăuga useri care să poată modifica locațiile. Totodată are acces complet asupra tuturor evenimentelor din locația sa, chiar dacă sunt create de altcineva. Totodată poate crea la rândul lui evenimente, alocând unul sau mai mulți DJ.
- DJ-ul este o extensie a EM-ului. primește un ecran în plus în care poate vedea evenimentele la care el este DJ. Are acces la eveniment și la cererile muzicale pentru evenimentul respectiv, pe care le poate și edita.

Utilizatorul își poate crea facil cont, utilizând adresa de mail sau autentificarea prin Google sau Spotify.

Pentru asta am folosit un pachet special pentru NEXT, NextAuth. Din cauza unor incompatibilități cu NextAuth și AppRouter-ul Next, am folosit propriul token de management al utilizatorului în defavoarea sesiuni oferite de NextAuth.

Pentru sporirea securității și pentru a diminua spam-ul, utilizatorul va trebui, la crearea contului să își confirme adresa de mail printr-un cod, care va fi trimis pe email, alături de un link.

Management resurse client-server

Utilizatorul este salvat într-un token jwt în cookies, care este apoi verificat de fiecare dată când acesta încearcă să acceseze API-ul. În cazul în care tokenul nu este valid / este expirat, locația utilizatorului pe site (link) va fi salvată în cookies, iar mai apoi, după autentificare, acesta va fi redirecționat la aceea locație.

Totodată datele utilizatorului sunt stocate, de asemenea în cookies și utilizate la fiecare radare, pentru mai puține call-uri în backend, respectiv în baza de date.

De asemenea, acordul utilizatorului pentru folosirea cookies este stocată tot în cookies, pentru a nu mai notifica o dată utilizatorul, odată ce a luat la cunoștință.

Pentru autentificarea cu Google sau cu Spotify, utilizatorul și token-ul se preiau și se alcătuiesc separat din cauza unei incompatibilități între App Router, NextAuth și Typescript și se salvează în același token ca un user normal.

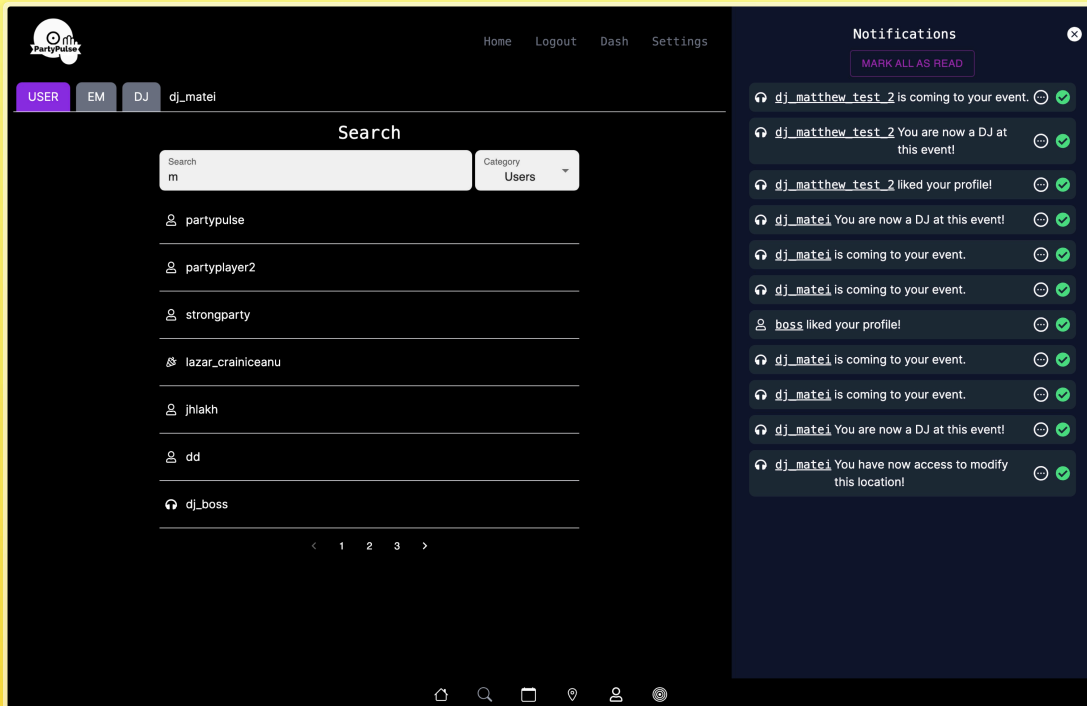
Interacțiunea cu utilizatorul (UX)

Notificările în aplicație sunt o parte importantă din aplicație. Pentru aceea am făcut un provider separat, care face fetch în backend la un interval regulat. Totodată, cu fiecare notificare, se trimite și un mail utilizatorului pentru a-l anunța pe acesta.

Pentru utilizatorii cu rol de management (em, dj) se adaugă în meniu o bară care are ca rol selectarea view-ului, potrivit pentru ceea ce vor să facă.

Interfața este testată și funcțională pe cele mai utilizate browsere: Chrome (și Arc), Safari, Edge, fără să apară probleme de afișare sau funcționalitate.

Totodată aplicația e responsive pentru toate dimensiunile de ecrane și oferă și suport dark/light mode. Light mode-ul adăugând doar o pată de culoare, fără să facă experiența neplăcută, mai ales fiind un produs conceput pentru a fi utilizat în spații întunecate.

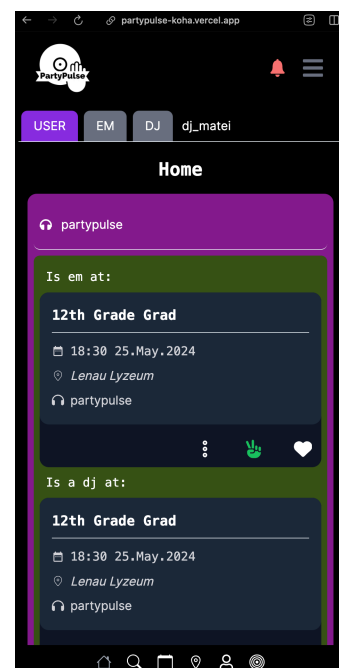
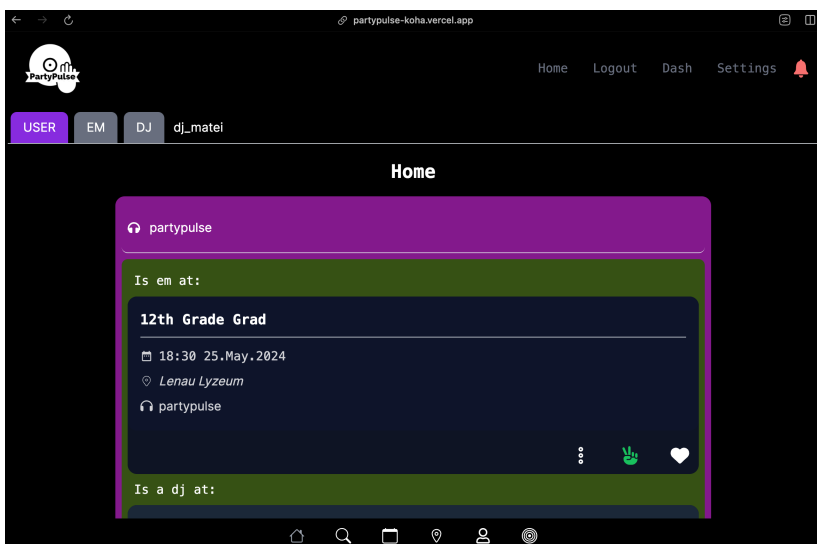


Securitate

Pentru a preveni posibile atacuri de tip SQL injection am folosit placehodere în statement-uri, urmate de array-ul cu valorile în mysql2.

Am întâmpinat probleme de genul conexiunilor de sql care nu se încheiau, așa că am extras tot db-ul într-o clasă, în care am închis conexiunea manual.

Intefață, UI și secțiuni



Utilizatorul avansat (em-ul și dj-ul) are acces la o bară de meniu, pentru a putea comuta între diferitele meniuri care oferă funcționalități corespunzătoare fiecărui rol.

În interfața user-ului simplu, aceasta bară dispăre. Acest dashboard are mai multe secțiuni:

Home

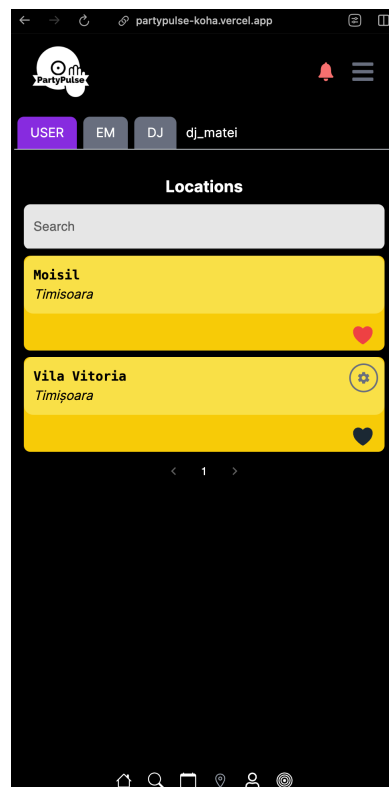
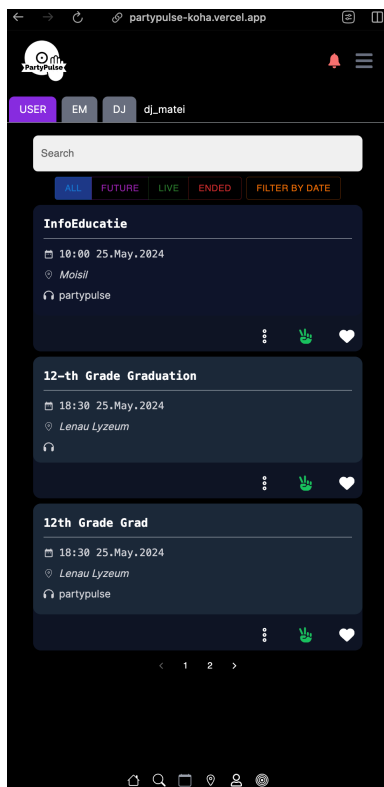
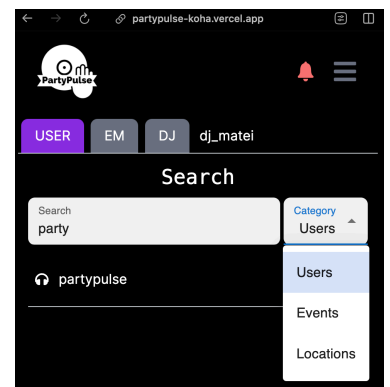
Exemplificată mai sus, aceasta arată utilizatorului evenimentele la care sunt dj sau em alți useri apreciați de acesta.

De asemenea, pe acest ecran apar și evenimente la locațiile favorite.

Search

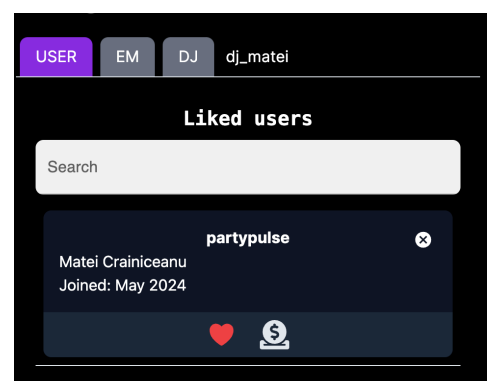
Utilizatorul poate căuta alți useri, alte evenimente sau alte locații din platformă. Ei apar toți, iar în cazul în care sunt mai mulți, va apărea un meniu de paginare în partea de jos a paginii.

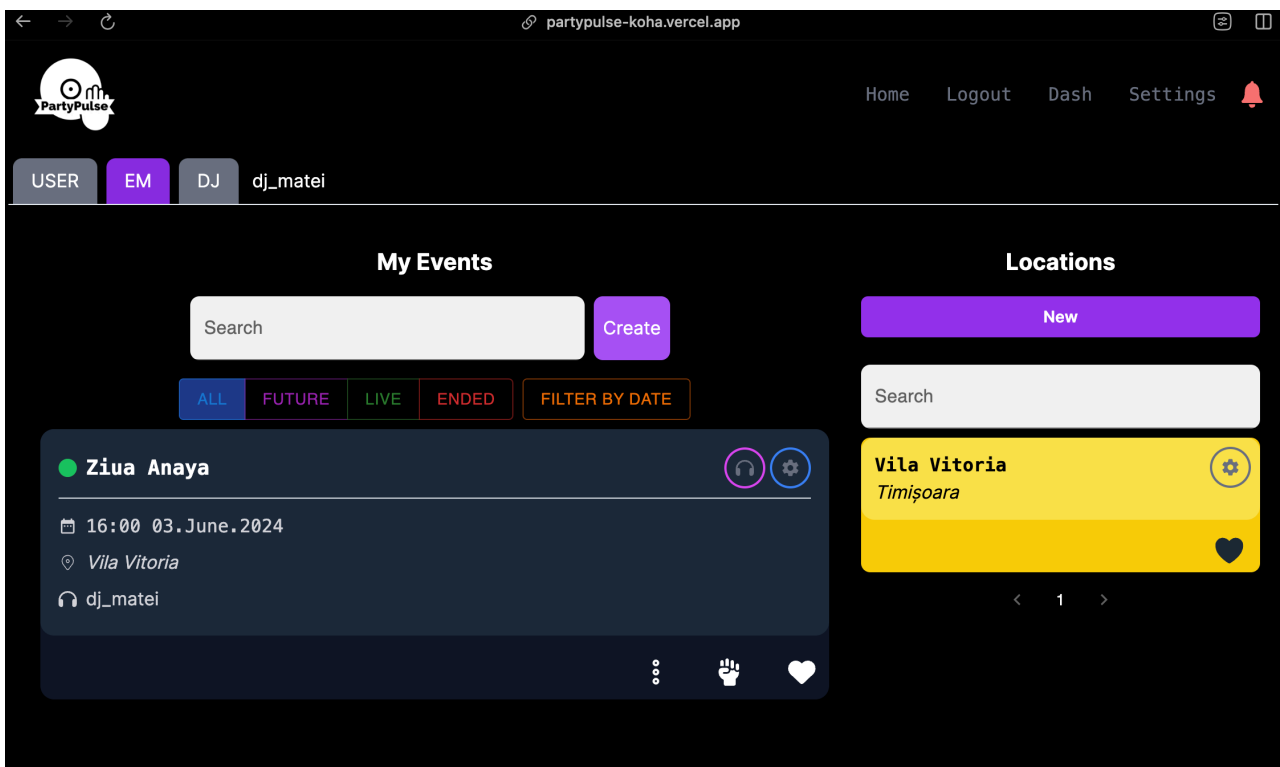
De asemenea, în cazul evenimentelor, acestea pot fi filtrate după dată sau status.



Events, users & users

În fiecare dintre aceste secțiuni, utilizatorul vede elementele apreciate ale fiecărei categorii.

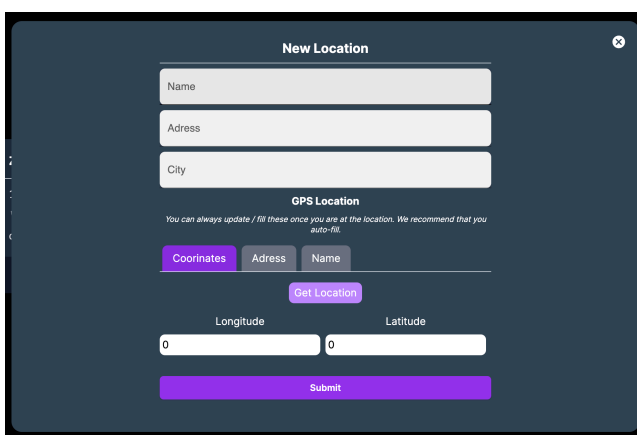
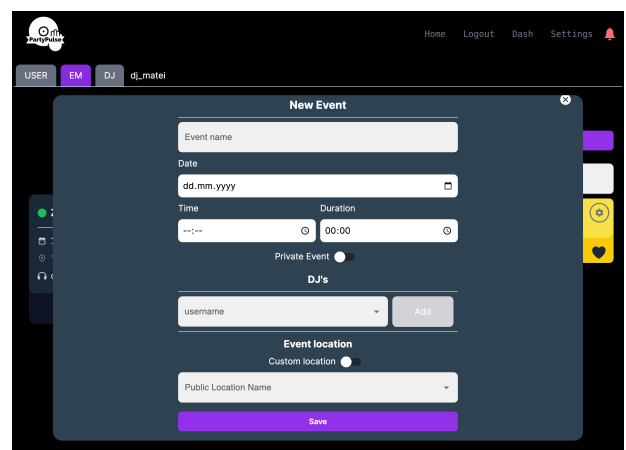




În secțiunea de EM, a dispărut bottom-bar-ul. Sunt în schimb pe ecran două secțiuni, una pentru a crea și edita evenimente, și cea de-a doua pentru a crea, și edita locații.

Evenimentele se crează din meniul exemplificat alături, iar acestea Pot avea unul, niciunul sau mai muți DJ.

De asemenea ele pot fi private, sau publice, la locații private sau publice.



Locațiile se pot crea tot din meniul EM-ului.

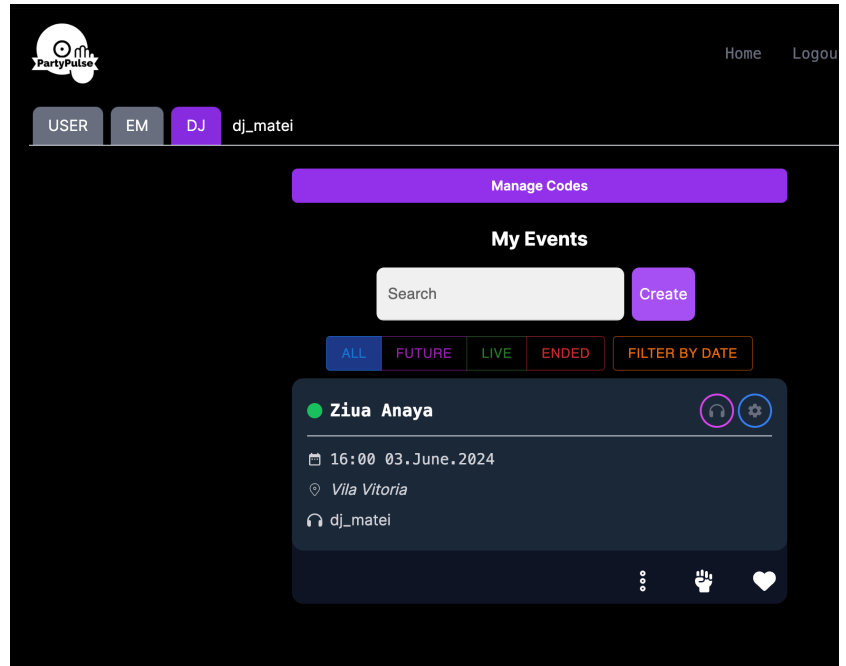
Pentru locația lor, se pot folosi fie coordonatele preluate automat, fie o adresă introdusă de către utilizator, sau, în cazul în care există o locație de pe google maps. Utilizatorul va vedea un preview de îndată ce o metodă este selectată.

În secțiunea de EM, dar și în cea de DJ (următoarea), atât evenimentele, cât și locațiile sunt împărțite prin paginație.

DJ

Dash-ul pentru DJ este diferit față de cel de EM. În timp ce, în ecranul anterior apărau toate evenimentele la care userul este dj sau em, aici apar doar evenimentele la care utilizatorul are dreptul de DJ.

Dar, în partea de sus a ecranului, se observă butonul de manage codes.

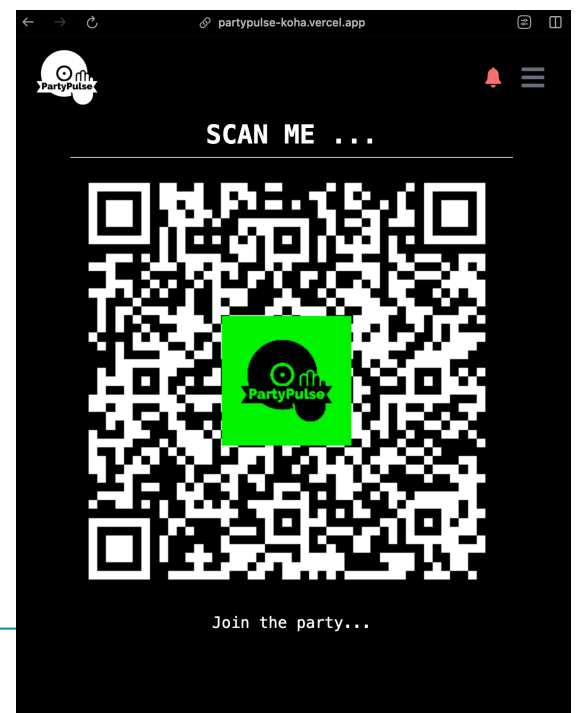
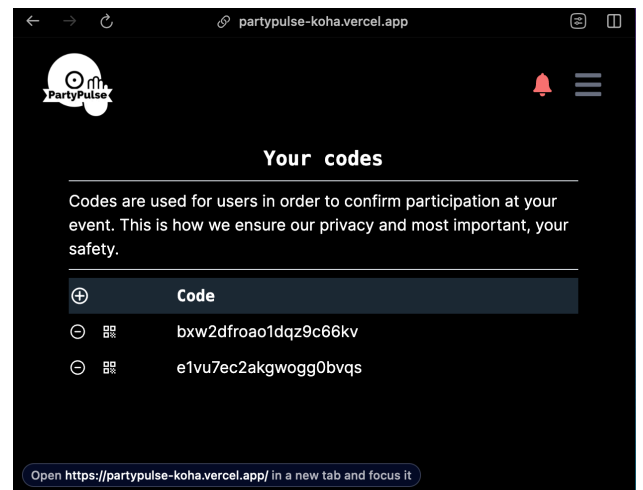


Practic, pentru a asigura buna folosire a platformei, în încercarea de a opri spam-ul, pentru a putea participa la un eveniment, trebuie să se confirme prezența prin scanarea unui cod qr de la locație, funcționalitate prezentată în continuare.

QR - Aplicare și reducere de spam

Printre primele provocări ale acestei aplicații a fost, faptul că utilizatorii pot vota la orice petrecere, chiar dacă ei nu sunt prezenți.

Astfel, un mod de a preveni această situație, a fost utilizarea codurilor. Nu am vrut ca acestea să fie de unică folosință, din motive evidente pentru risipa de hârtie sau pentru alte probleme tehnice, în utilizarea și încurcarea lor.



Astfel că am venit cu ideea de a avea coduri qr pentru locații și DJ. Practic: Utilizatorul scanează codul qr, aparținând locației, sau DJ-ului, iar acesta este asociat cu evenimentul la care DJ-ul este prezent în timp real, sau care se desfășoară live la locație. Astfel, un DJ sau o locație poate avea un singur sau mai multe coduri qr, amplasate pe echipamente sau în locație permanent.

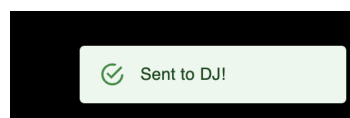
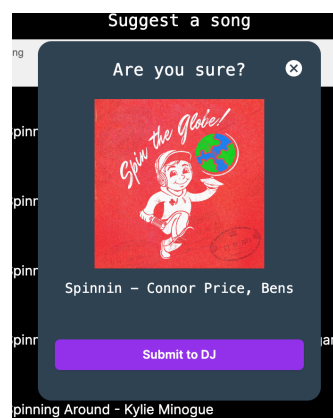
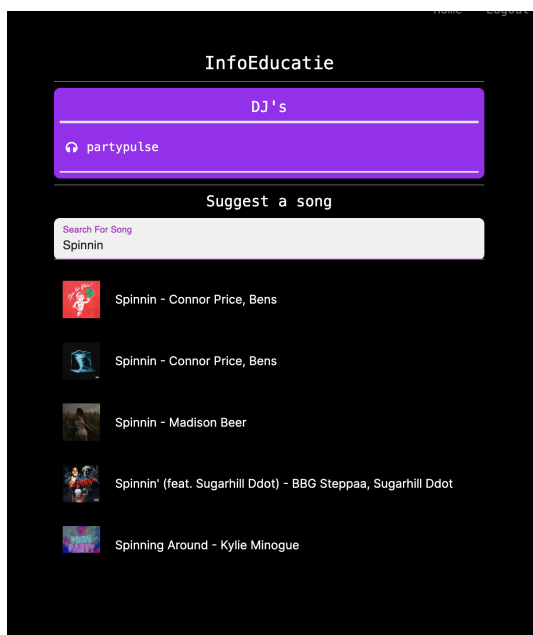
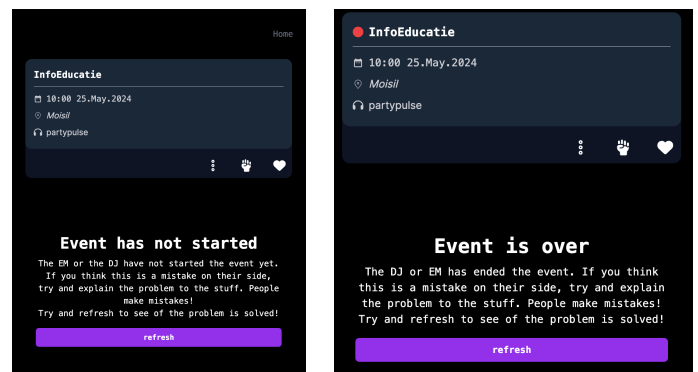
Desigur, codurile qr pot fi șterse sau se pot crea altele noi pentru un utilizator anume sau pentru o locație.

Settings

Din meniul de settings, utilizatorul își poate schimba parola, poate deveni DJ sau EM și își mai poate schimba datele personale.

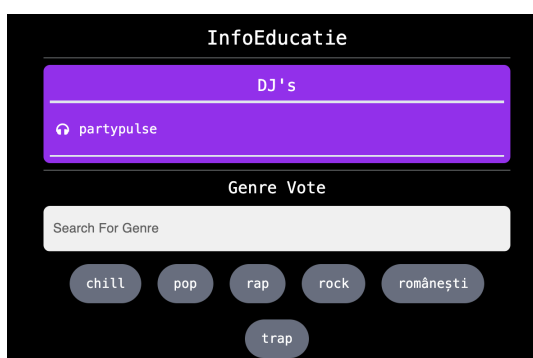
Live View - User

Principala funcție a aplicației este faptul că utilizatori pot vota genul muzical, sau sugera melodii către DJ. De aceea, următorul component este cel mai important.



Utilizatorul poate căuta o melodie. Căutarea se face prin API-ul de la Spotify, deci practic, utilizatorii aleg din catalogul lor de muzică.

Odată selectată o melodie, utilizatorul este rugat să confirme, urmând ca după trimiterea acesteia să primească el confirmarea de trimitere sub forma unei alerte subtile.



O altă funcționalitate este genre-vote-ul. DJ-ul poate începe un vot, iar audiența va vota ce gen de muzică se va asculta în următoarea perioadă.

Am văzut această tentație le petreceri, de a folosi

Google Forms pentru această funcționalitate.

DJ-Interface

DJ-ul are o interfață compactă în care am înglobat cât mai multă funcționalitate.

Acesta are în partea stângă controalele evenimentului. Îi poate modifica statusul, poate activa/dezactiva sugestiile muzicale sau poate porni sau opri un genre-vote. De asemenea își poate vedea live și modifica pagina pentru donații.

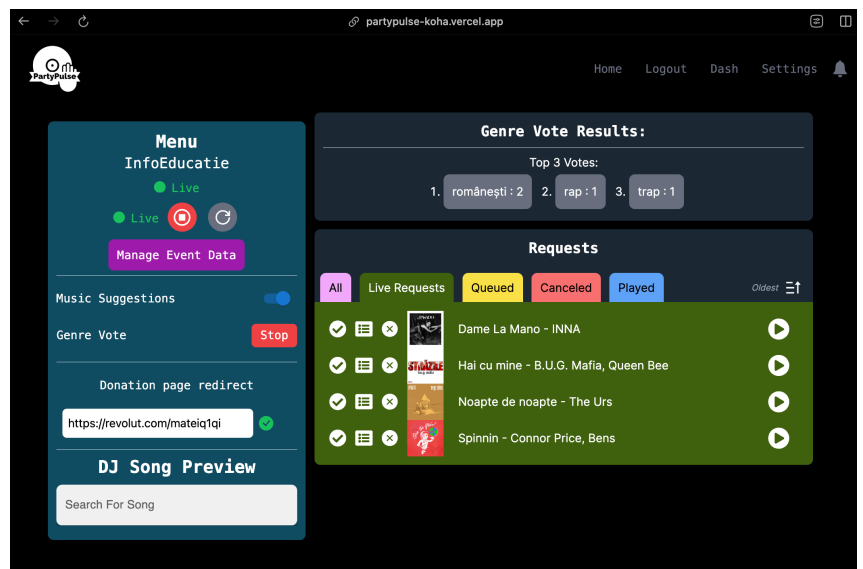
În cazul în care genre-vote-ul este pornit, în partea de sus, dreapta, se pot vedea rezultatele la genere vote, ordonate crescător, cu tot cu numărul de voturi. pentru a păstra totul neaglomerat și aerisit, dj-ii văd doar top 3 genuri votate.

Mai jos este secțiunea de “dedicații” sau sugestii muzicale. De aici, dj-ul preia sugestiile din secțiunea “live requests”. În partea stângă a fiecărei melodii se află acțiunile aferente: “played” - “am pus-o”, “que” - “o pun mai târziu” sau “cancel” - “nu știu ce e asta, nu o pun”.

Dar totuși în caz că DJ-ul nu cunoaște o melodie, poate asculta preview-ul cu butonul de play, din partea dreaptă.

Cu toate astea, revenind la secțiunea de control, am adăugat un feature pentru DJ. Aceștia pot căuta orice melodie rapid și pot să îi asculte preview-ul.

Aceste preview-uri vin de la API-ul spotify, de unde vin și melodiile.



Componente individuale - UI in detail



Evenimentele au secțiunile pentru informații (nume, locație, dată, oră, dj), dar au și un Status Indicator, care este absent dacă evenimentul nu a început, verde, dacă este live sau roșu dacă s-a terminat.

În partea dreaptă sus,

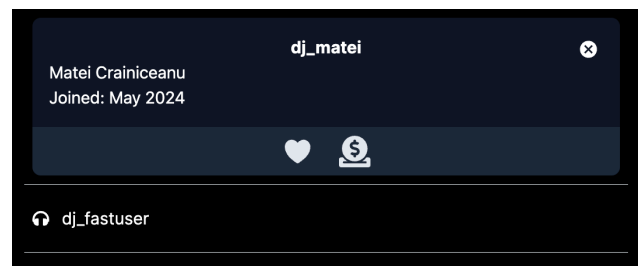
apar două opțiuni: una de DJ, pentru a trece la interfața de DJ și a doua pentru a schimba datele evenimentului. Acesta apar DOAR DACĂ userul are permisiuni de DJ, sau (pentru a doua), evenimentul se desfășoară la o locație din cele la care acesta are drept de admin.

În partea de jos, evenimentul are opțiuni pentru a reacționa. Prima este pentru a vedea detalii, conduce utilizatorul către pagina evenimentului. A doua este pentru ca utilizatorul să poată spune “going”, adică acesta va fi prezent la eveniment, iar a 3-a, inima, este pentru ca utilizatorul să poată aprecia evenimentul, astfel având acces la el mai târziu, fără ca să bifeze “going”.

Locațiile au, la fel, secțiunea pentru informații, urmate de iconița de setări care apare doar dacă utilizatorul are drept de edit asupra acesteia. Inima este de asemenea pentru ca utilizatorul să poată aprecia locația.



Utilizatorii apar și ei în două moduri. unul prescurtat, astfel încât să poată fi vizualizați mai mulți pe o pagină, și altul extins pentru a putea interacționa, prin donații sau aprecieri cu aceștia.

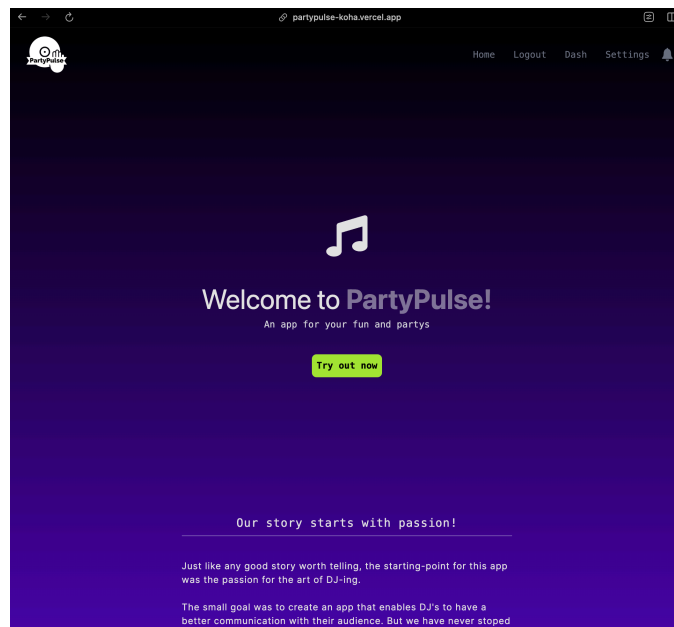
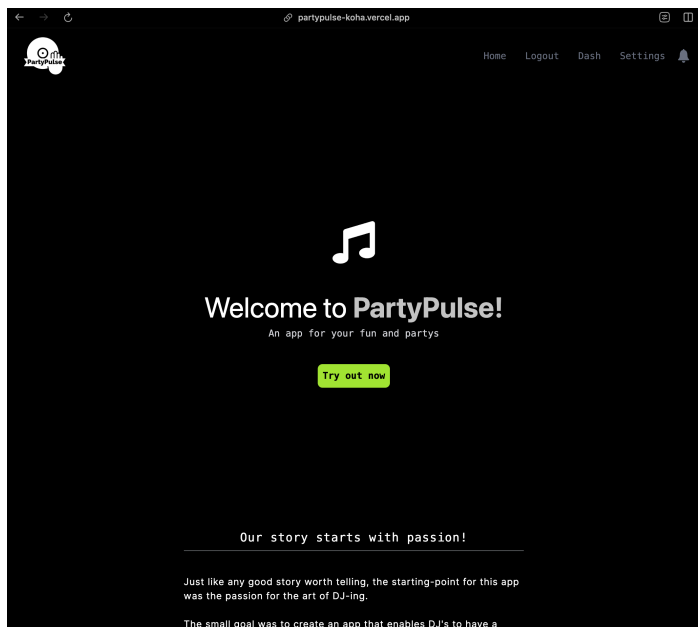


De asemenea sunt disponibile anumite informații precum numele și data alăturării platformei.

Responsiveness

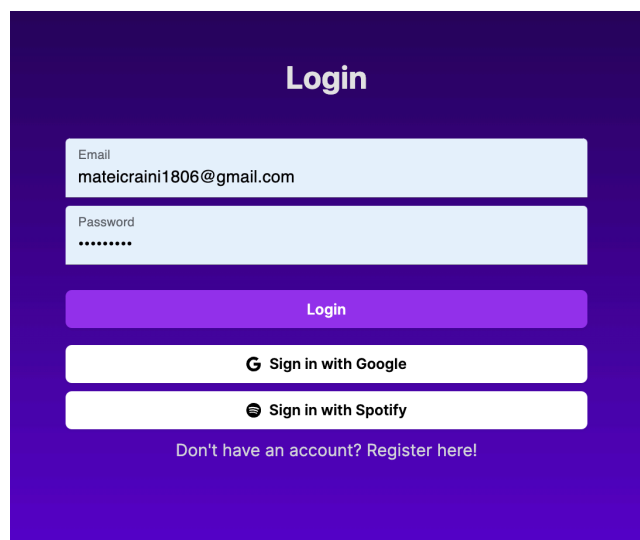
Fiecare ecran prezentat este utilizabil și pe mobil. Acest fapt este testabil și pe varianta demo web a aplicației.

De asemenea, am creat și un light-dark mode, având grija ca light-mode-ul să nu fie prea light, ci să aducă o pată de culoare.



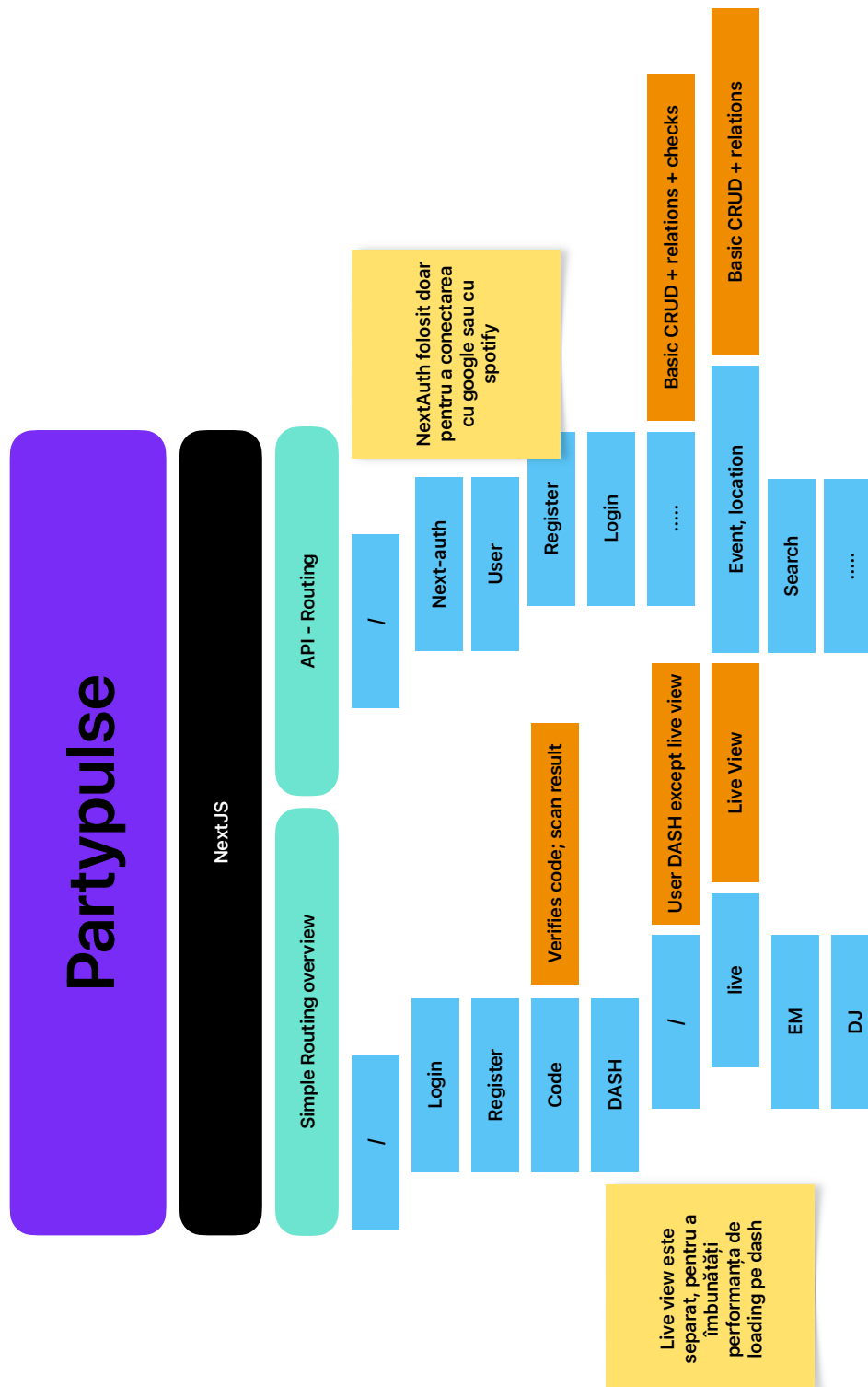
Ease of use

Pentru a facilita accesul în aplicație, utilizatorii se pot conecta cu Google sau cu Spotify, însă, pentru sign-up-ul clasic cu email și parolă, utilizatorul va primi un cod pe mail pentru a putea fi confirmat.



Arhitectura

Ca și framework pentru development am folosit NextJS 14 cu AppRouter și Typescript.



Arhitectură DB

DB PARTYPULSE (SQL)

