Progetto PWM

Per il progetto di Programmazione Web e Mobile io (Salillari Tedi) e il mio collega (Kerkeni Ilyas) abbiamo deciso di implementare Social Network for Music.

Aprendo il sito web si puo notare subito una cosa, ovvero il design del sito. Ci siamo ispirati al design del sito di Threads di Meta, infatti si può notare come molto spesso, se si mettono in evidenza insieme, si possono trovare somiglianze in alcuni componenti e perfino colori.

Dopo aver visionato per bene come era strutturato, in termini di design, il sito di Threads, abbiamo iniziato ad implementare il nostro sito SNM.

Il sito è diviso in due parti fondamentali:

- Backend
- Frontend

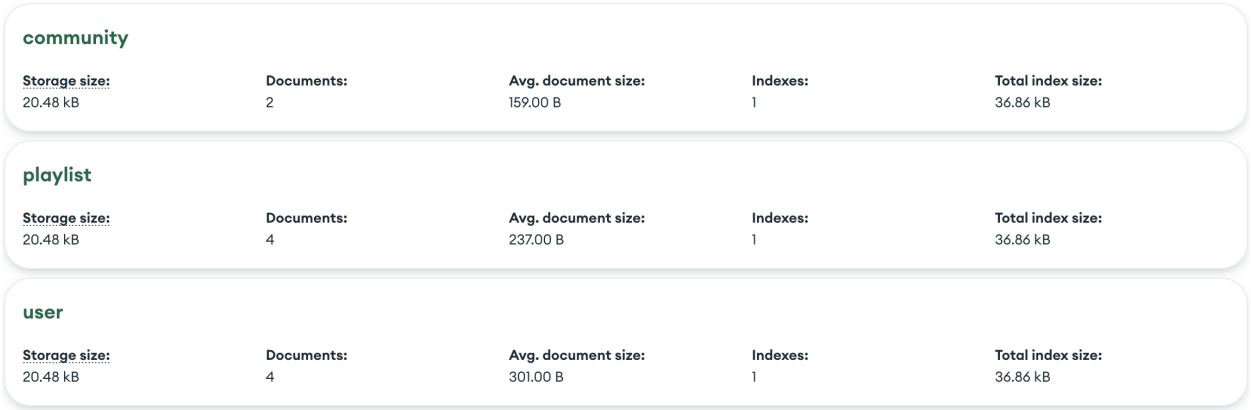
Partiamo parlando con il backend.

Backend

Il backend è stato implementato usando node.js, utilizzando la libreria Express per creare l'API, mentre mongoDB per creare il database dove immagazzinare i dati del nostro sito.

Il database è suddiviso in 3 sezioni (collezioni):

- user (collezione dove vengono inseriti i dati degli utenti)
- playlist (collezione dove vengono inseriti i dati delle playlist create dagli utenti)
- community (collezione dove vengono inseriti i dati delle playlist create dagli utenti)



User

La collezione user contiene i dati degli utenti che si registrano nel sito. Quando un utente si registra, vengono inserite queste informazioni nel database:

```
{
    "name": "",
    "surname": "",
    "username": "",
    "date_of_birth": "",
    "email": "",
    "password": "",
    "liked_artist": [],
    "liked_genres": [],
    "playlist": {
        "personal":[],
        "liked":[]
    },
    "community": []
}
```

Per ogni utente sarà presente un id, generato automaticamente da mongoDB. Lo username, come anche l'email, sono univoci per ogni profilo, ovvero non si possono avere per esempio due account con lo stesso username.

Playlist

La collezione playlist contiene i dati delle playlist create dagli utenti. Quando un utente crea una playlist, vengono inseriti questi dati nel database:

```
{
    "username": "", -> string
    "isPublic":   , -> boolean
    "name":     "", -> string
    "description":, -> string
    "tags": []    , -> array
    "songs": []   , -> array
}
```

Con il campo isPublic capiamo e riusciamo a getire se le playlist sono pubbliche o private. Anche qui l'id viene generato automaticamente da maongoDB.

Community

La collezione community contiene i dati delle community ceate dagli utenti. Quando un utente crea una community, vengono inseriti questi dati nel database:

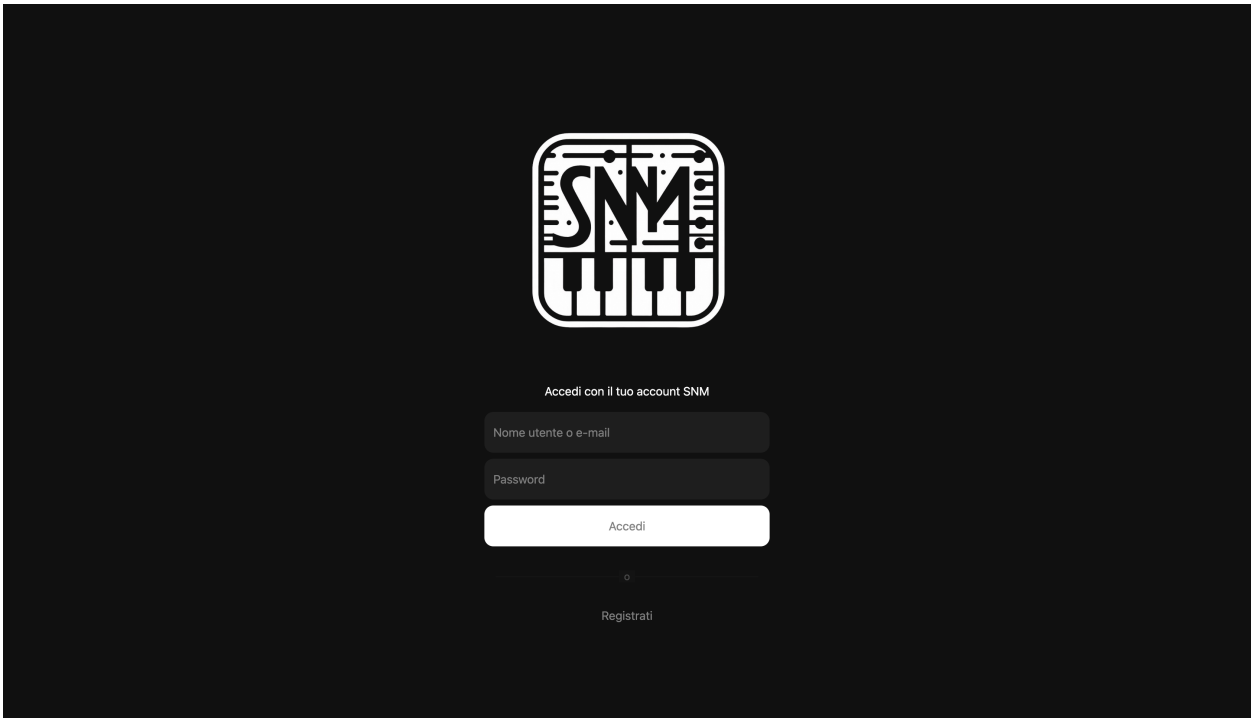
```
{
    "admin": "", -> string
    "name":  "", -> string
    "description": "", -> string
    "users": [], -> array
    "shared_playlist": [] -> array
}
```

Come nelle altre due collezioni, anche nelle community l'id è creato da MongoDB.s

Frontend

Dopo aver scelto come volevamo che il sito fosse visto dagli utenti, abbiamo iniziato ad implementatare la prima pagina del sito: il ***login.***

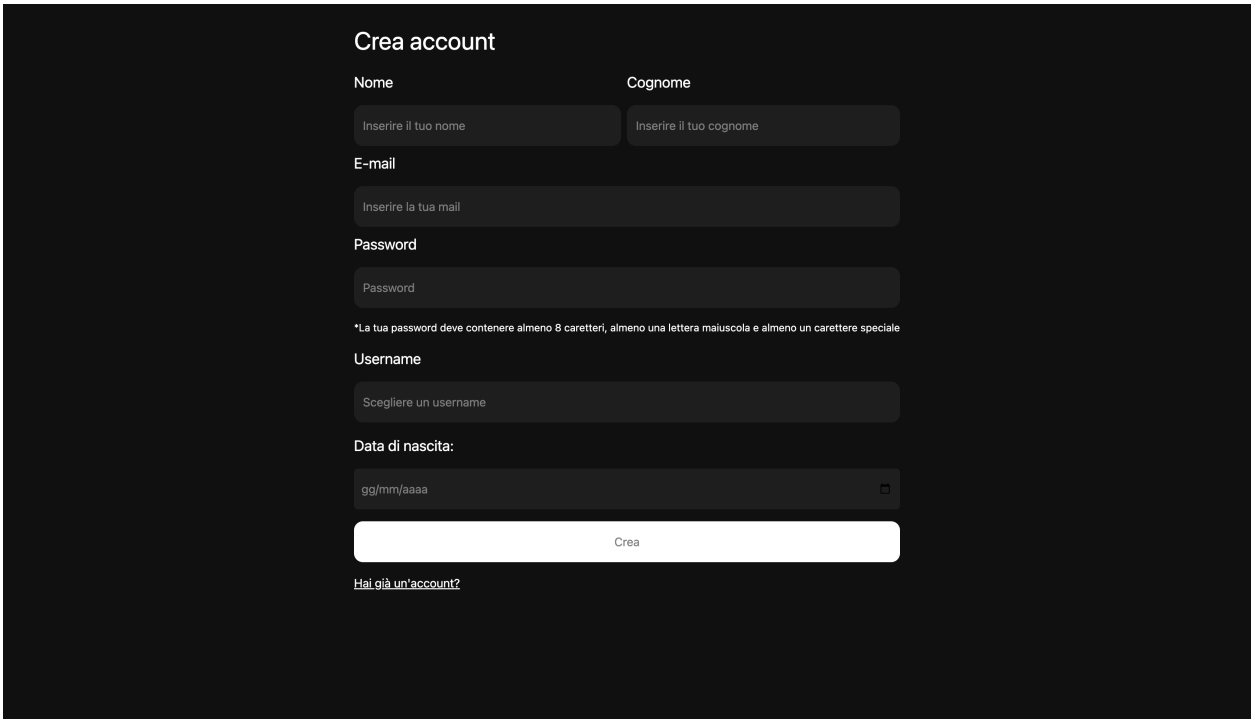
Login.html



La pagina di login è molto semplice ed è molto simile a quella di Threads sia nei colori che nel design vero e proprio. Classica pagina di login, dove troviamo due input e un tasto accedi per accedere se gia si ha un account, altrimenti ci si può registrare premendo il tasto in basso.

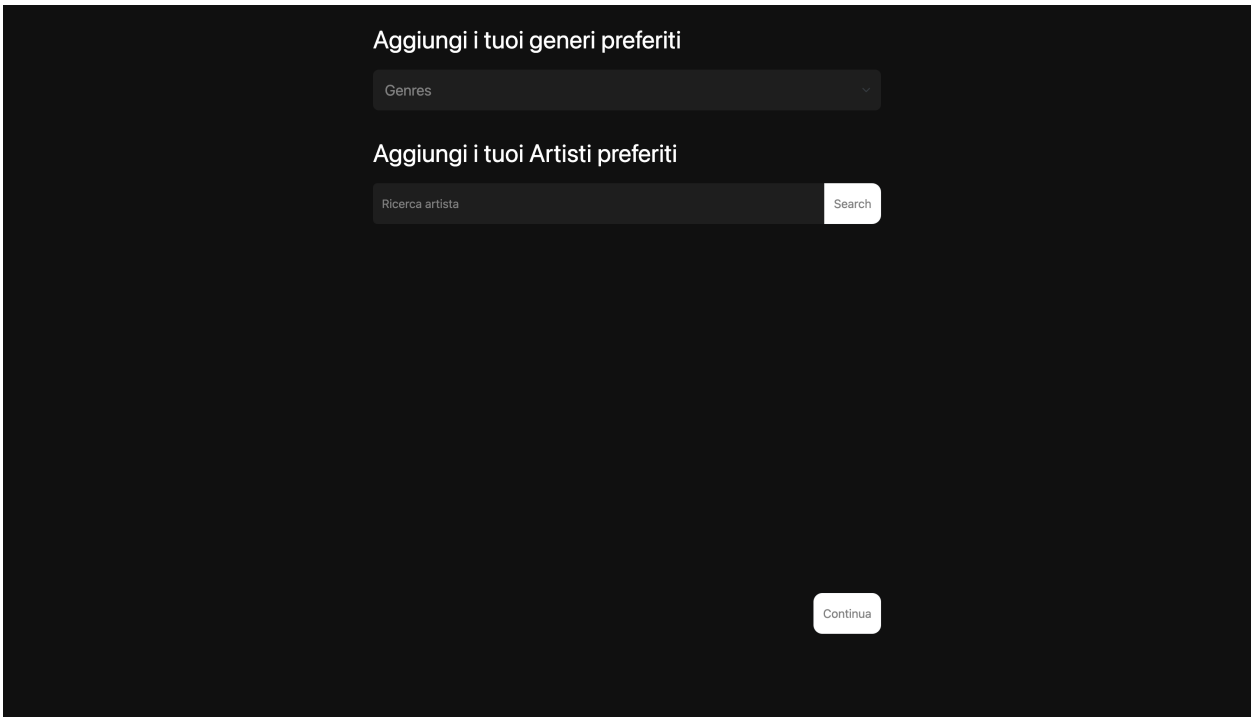
Il logo del sito è stato fatto utilizzando copilot.

Registrati.html



Nella pagina di registrazione possiamo trovare tutti i campi necessari per creare un profilo, che poi andranno a finire nel database.

La pagina della registrazione è formata da due sezioni, una dove si inseriscono i dati personali di un utente, come l’email, password, username, data di nascita ecc..., l’altra dove si inseriscono le preferenze musicali dell’utente.



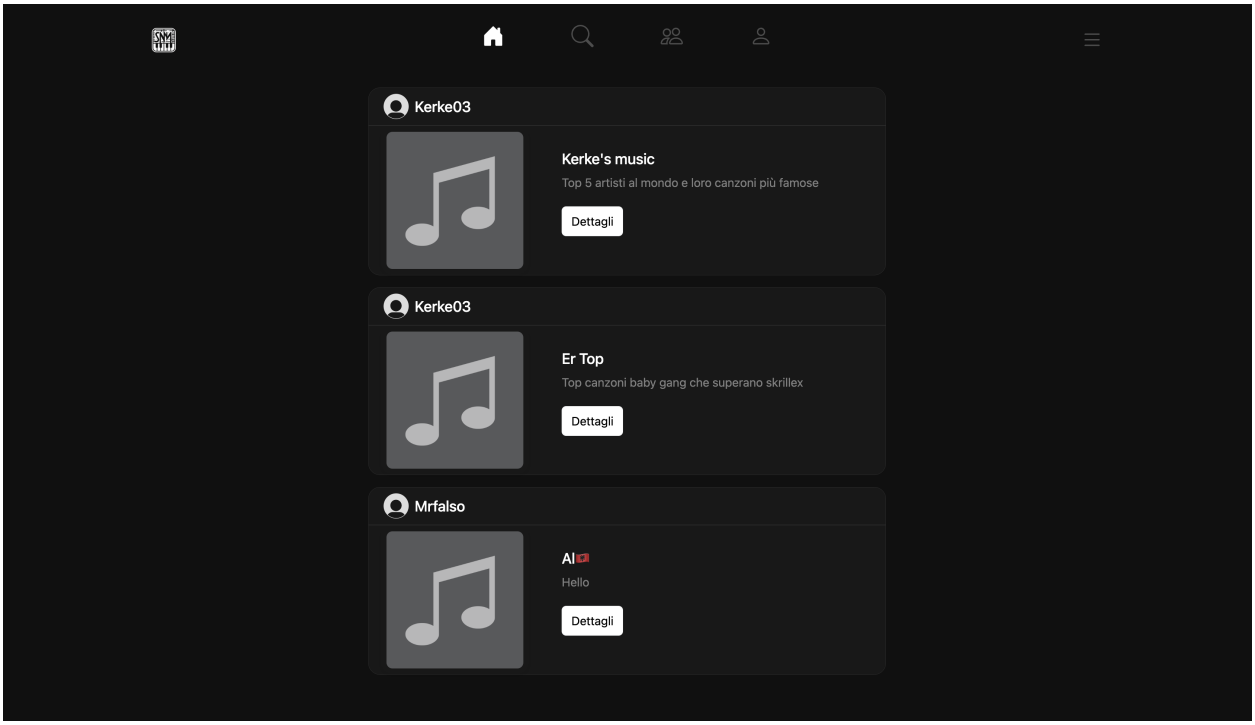
Nella seconda sezione della pagina registrati vengono chiesti all'utente di inserire i propri generi preferiti e i suoi artisti preferiti. Per tutti e due abbiamo usati l'API di Spotify, che ci permette di dare queste informazioni.

Per quanto riguarda la selezione dei generi, abbiamo creato un endpoint nella nostra API scaricando il JSON dall'API di Spotify, che ci ritorna tutti i generi disponibili. Abbiamo fatto questo perchè abbiamo notato che il rate limit dell'API è molto basso, quindi ci bloccava l'utilizzo dell'API per un pò di tempo. Facendo cosi non abbiamo piu questo problema.

Invece per la selezione degli artisti si chiama direttamente l'API di spotify che ci ritorna l'artista o artisti che digitiamo. Per semplicità abbiamo messo il limite della ricerca a 10 artisti.

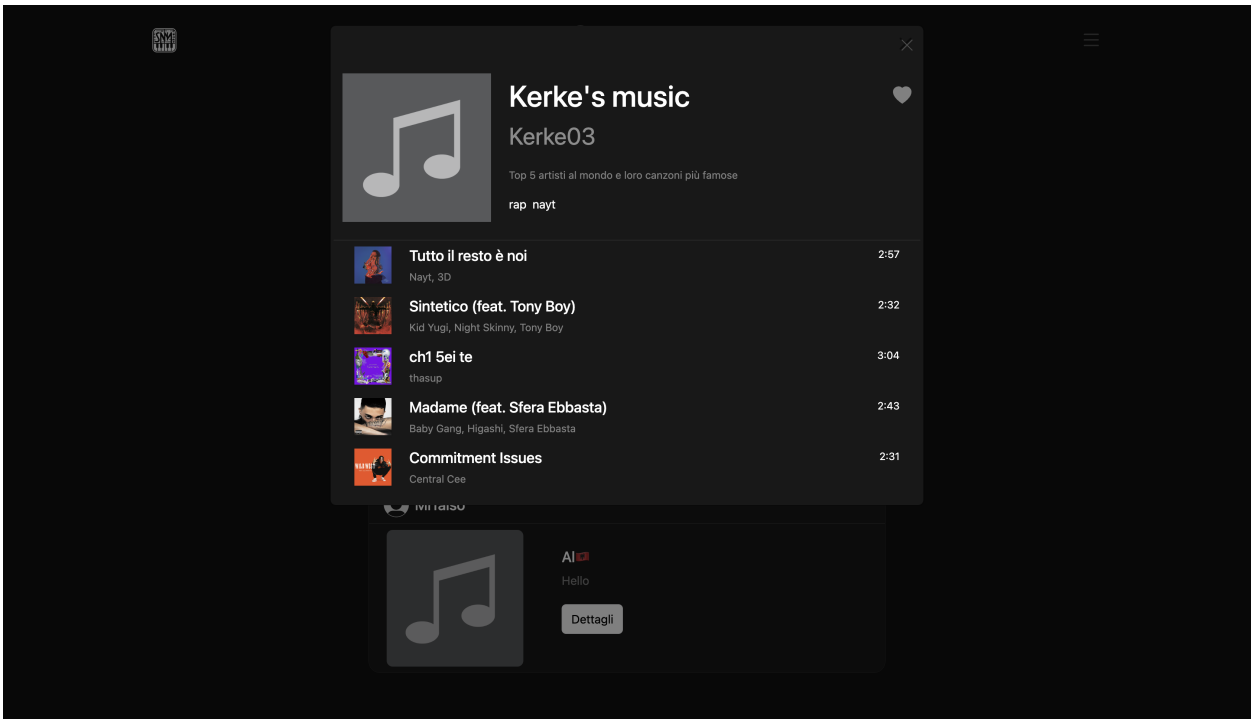
Quando si clicca invia si viene spostati nella pagina principale del sito: index.html.

Index.html

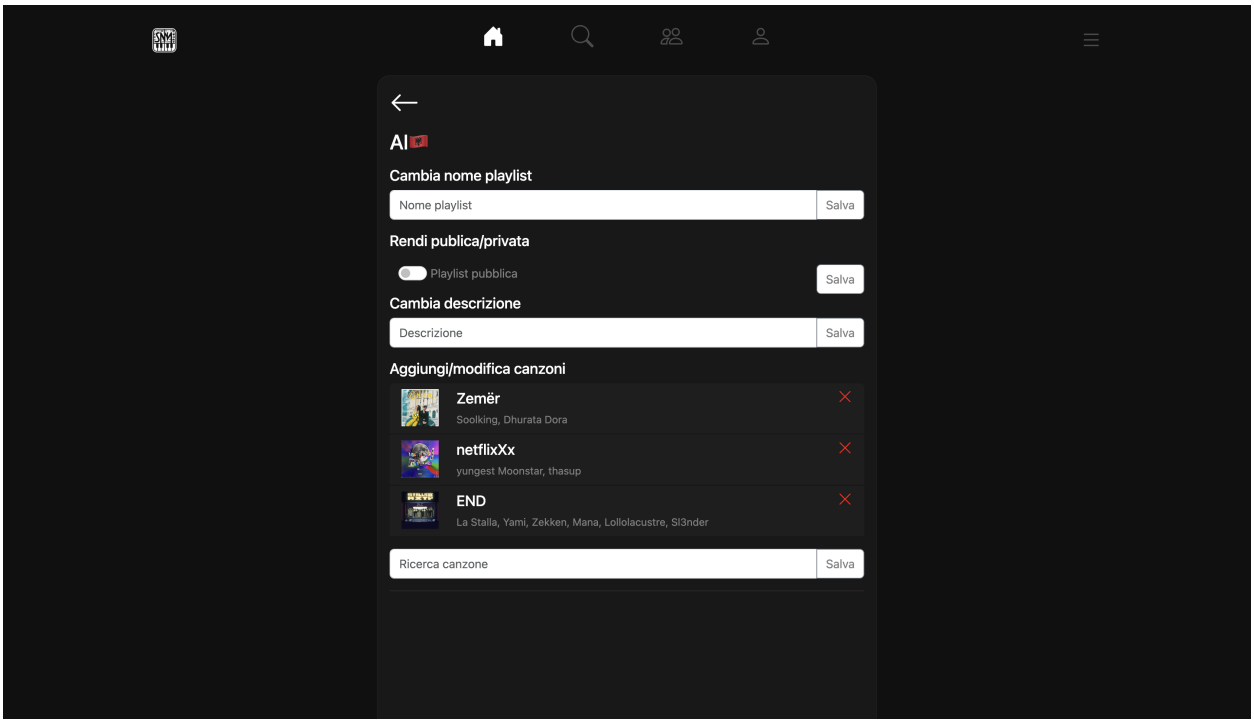


Nella pagina index.html, ovvero la pagina principale del sito, possiamo trovare tutte le playlist pubbliche che gli utenti hanno creato. Delle playlist si possono vedere il nome, la descrizione e un tasto per far vedere una visione dettaglia della playlist.

Il modo con il quale facciamo vedere le informazioni delle playlist è la stessa per tutto il sito, ovvero abbiamo usato un modal che si apre quando si clicca nel bottone dettagli.



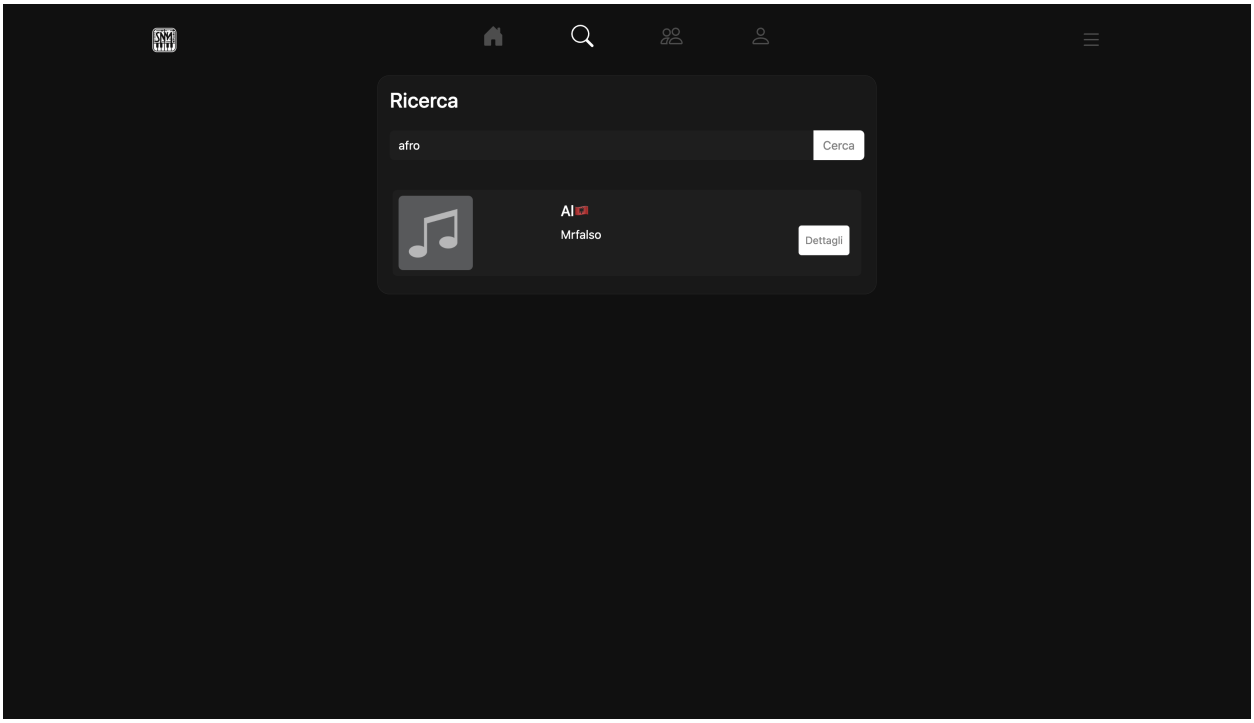
Nel modal si possno vedere tutte le informazioni della playlist, ovvero il nome, propretario della playlist, descrizione, tags, e le canzoni che ne fanno parte. Il cuore alla destra serve per mettere la playlist nei tuoi mi piace. Nelle playlist è anche presente un bottone a tre puntini, con il quale è possibile modificare i dati della playlist, o perfino cancellarla.



Nella modifica della playlist è possibile modificare i dati principali, come si vedono nello screen.

Nella sezione Aggiungi/modifica canzoni si possono vedere prima le canzoni che ne fanno parte, poi sotto è possibile inserirne altre cercandole.

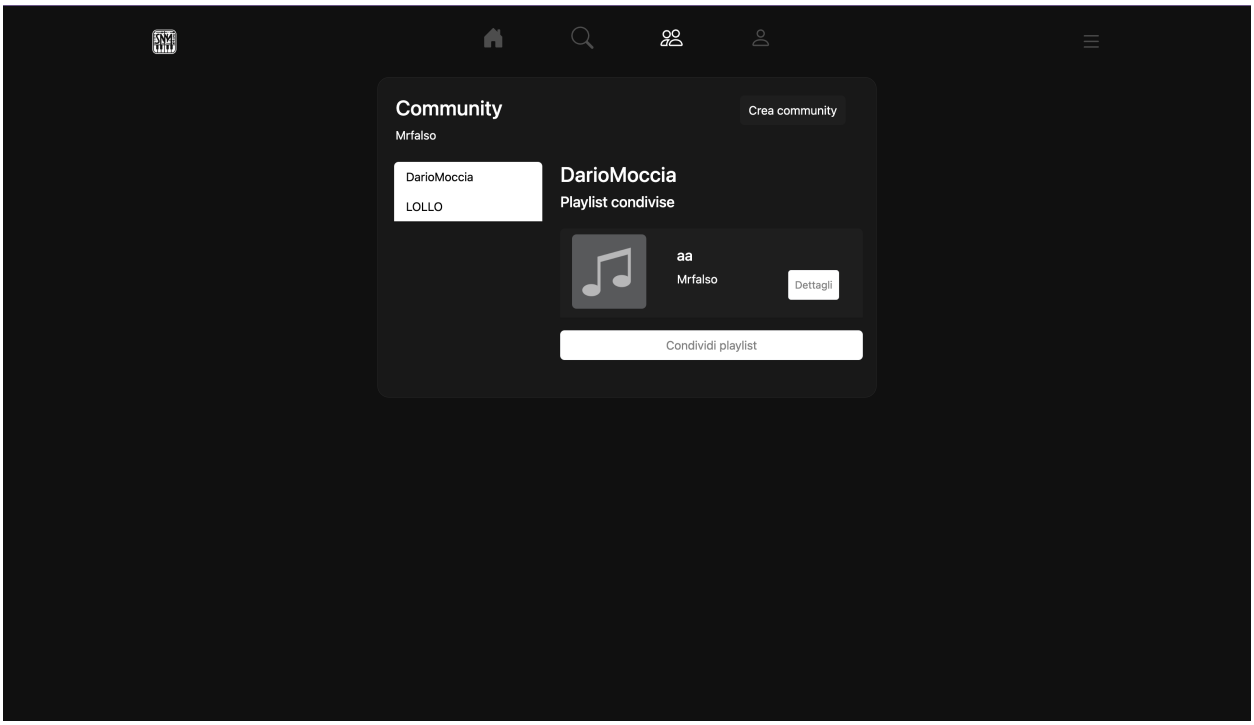
Search.html



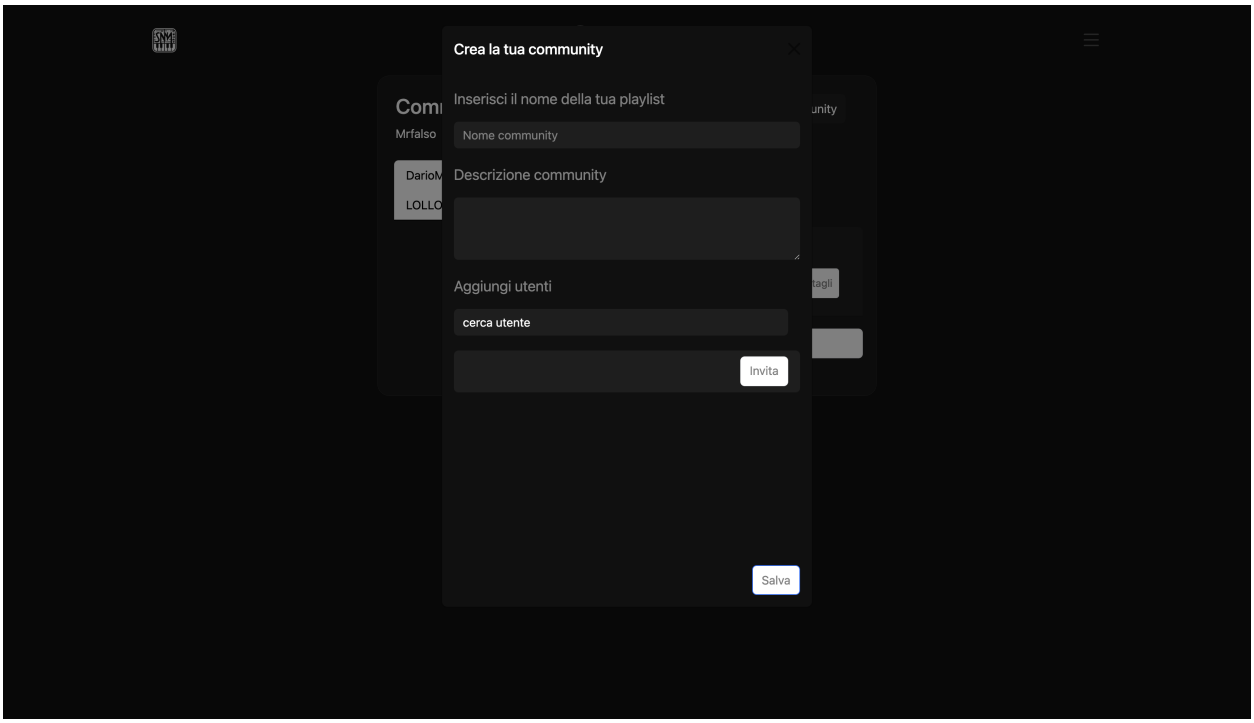
La pagina di ricerca è molto semplice. In questa pagina è possibile cercare una playlist pubblica tramite il suo nome o il suo tag. Non abbiamo inserito la possibiltà di cercare anche le comunità perchè è possibile vedere dalla pagina delle comminuty tutte le community a cui si fanno parte.

Quando si ricerca una playlist il ritorna le playlist con i criteri di ricerca che abbiamo scelto (nel caso sopra abbiamo cercato una playlist che aveva o il nome della playlist afro o il tag afro). Se si clicca ne bottone dettagli si vedra di nuovo la schermata del modal vista sopra dove è possibile vedere tutte le informazioni della playlist. Non possono essere viste le playlist private.

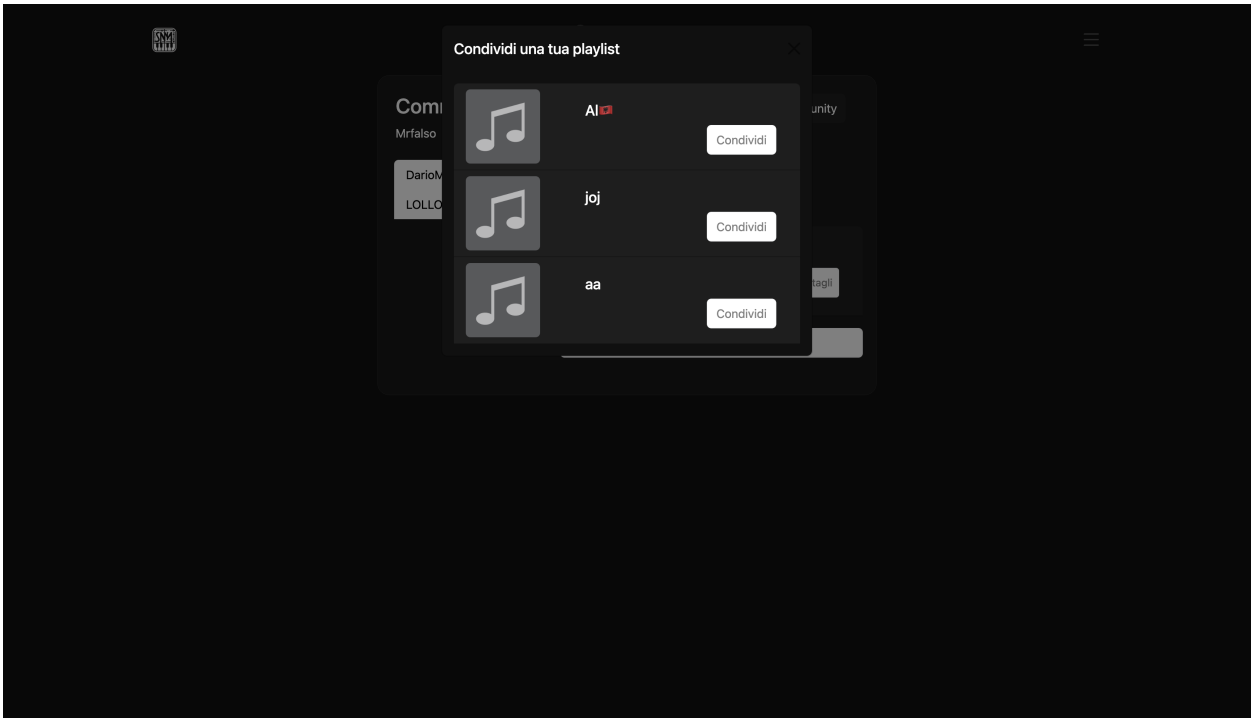
Community.html



Nella pagina delle community si trovano appunto tutte le community di cui si fa parte. All’interno della community si possono vedere tutte le playlist condivise dagli utenti, ed è possibile condividerne anche delle proprie. E anche possibile creare delle community, cliccando sul tasto in alto a destra, che aprirà un modal dove sarà possibile inserire i dati della community.

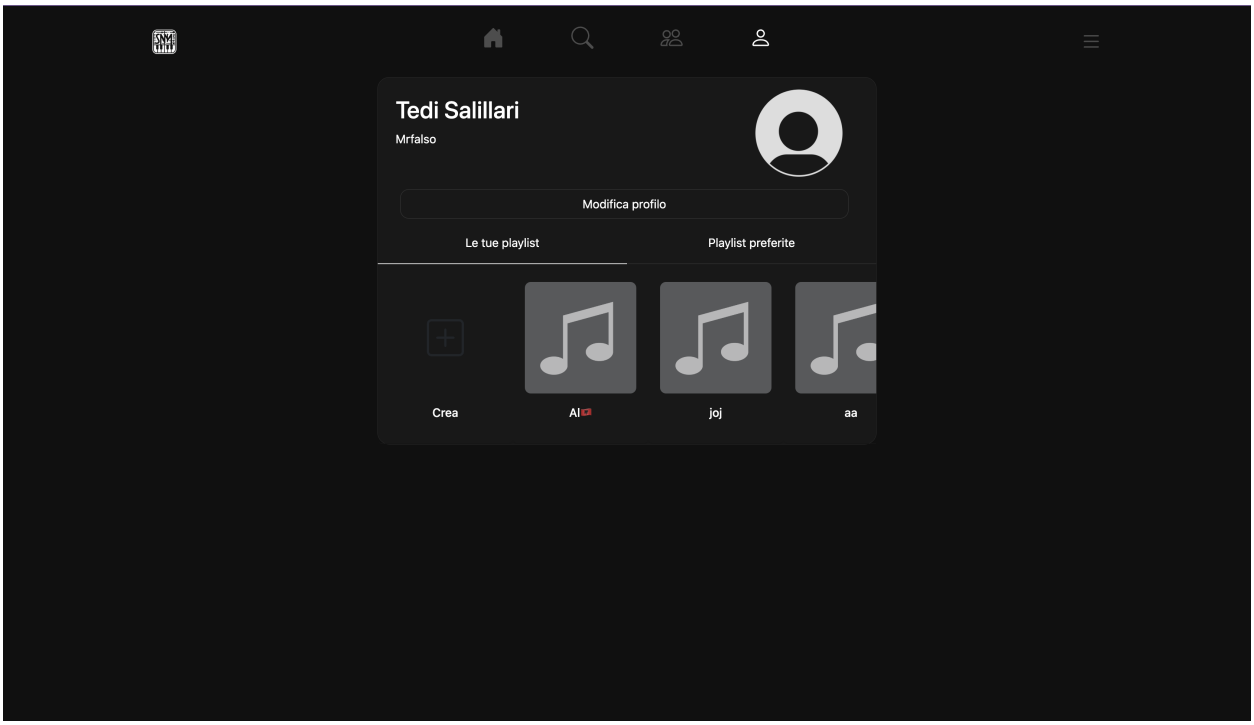


Quando si clicca sul tasto crea community si apre un modal dove è possibile inserire tutte le informazioni della community, quali nome, descrizione ed utenti.

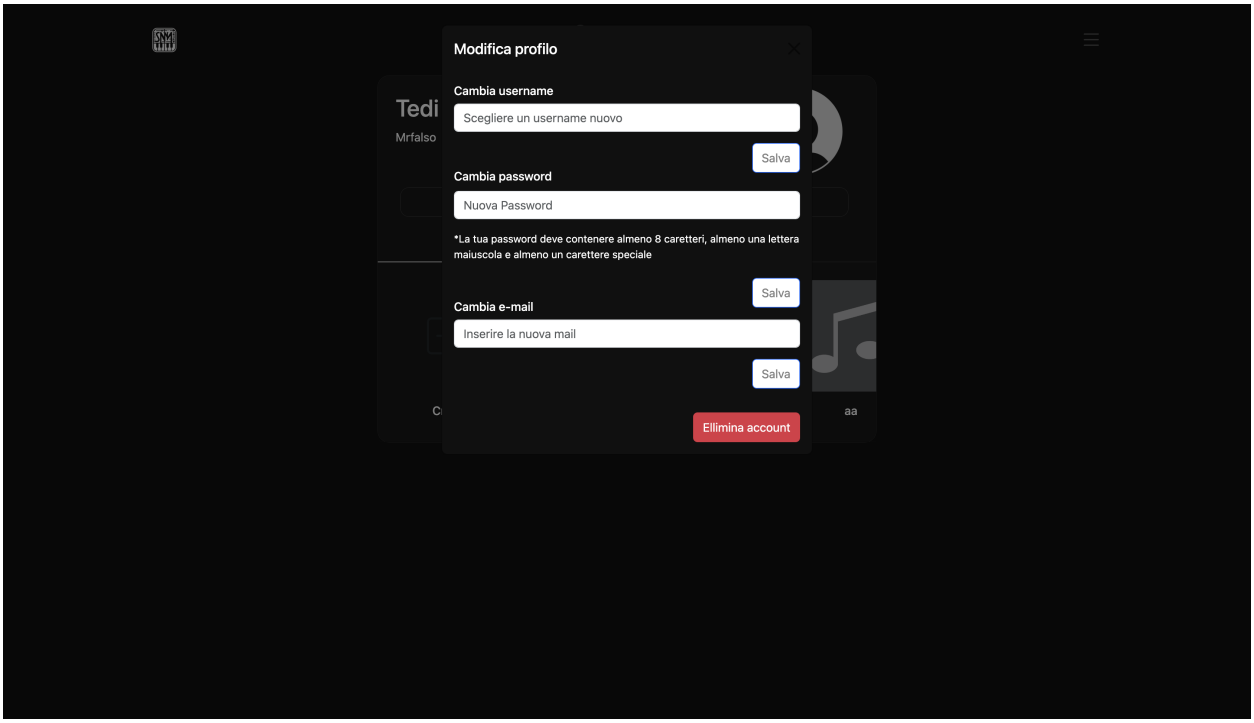


Quando si clicca su tasto condividi playlist si apre un’altro modal dove è possibile scegliere quali delle playlist si vuole condividere. Nelle comunità è possibile condividere anche playlist private. Abbiamo scelto cosi perchè siccome un utente sceglie, quando crea la comunità, quali utenti ne faranno parte, può condividere anche le playlist private senza paura che altri al di fuori la vedano.

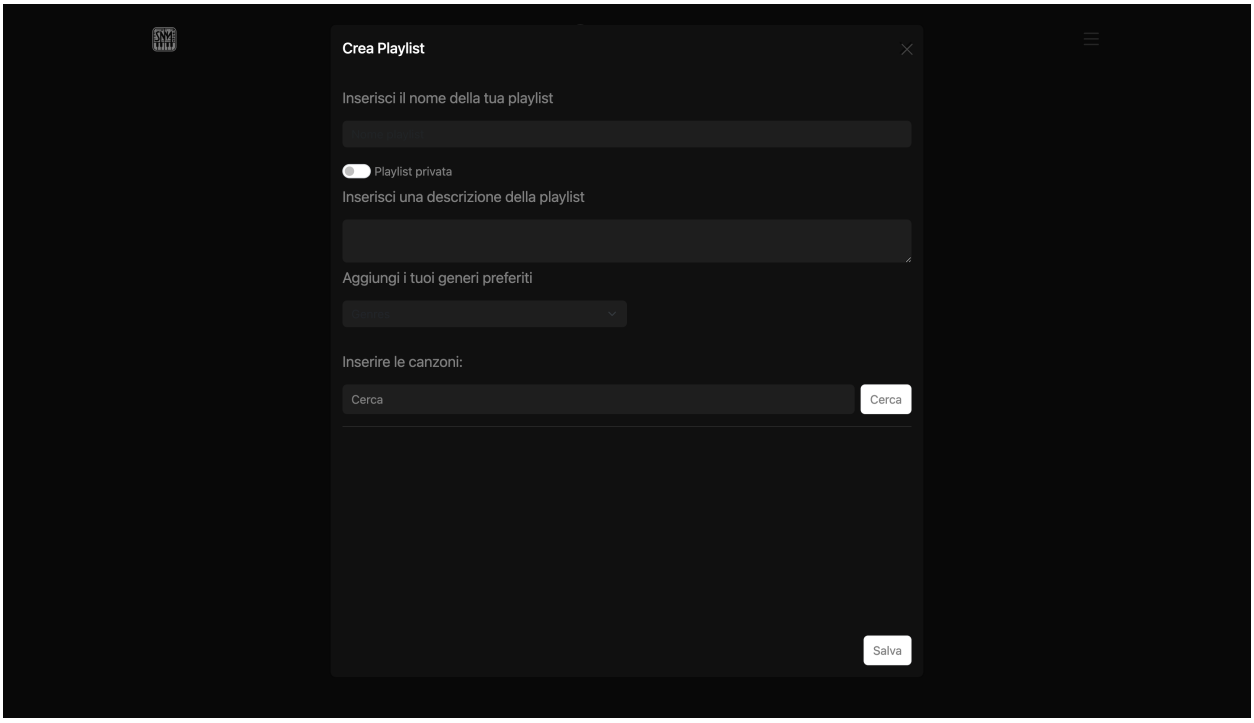
Profilo.html



Nella pagina del profilo è possibile visualizzare le informazioni principali dell’utente, quali nome, cognome, username e le playlist. Da qui è possibile creare le playlist cliccando sul tasto crea in basso. Se invece si clicca sul tasto modifica profilo è possibile modificare le informazioni del profilo utente.



Cliccando sul tasto modifica profilo si aprirà un modal che mostrerà tutte le possibili modifiche che si possono fare al proprio profilo. Abbiamo scelto di poter modifcare solo l'username, l'email e la password.



Quando si clicca nel tasto crea si apre un modal che ci permetterà di creare una playlist. All'interno è possibile inserire il nome, dire se sarà una playlist pubblica o privata, descrizione, tag e canzoni.

All'interno del sito è anche possibile vedere il profilo di un utente cliccando sul nome dalla playlist, e le informazioni principali di una canzone cliccando sul nome della canzone.

Nella pagina dedicata alle canzoni è anche possibile, cliccando su un link, andare nella pagina spotify dove si potra ascoltarla, ovviamente loggandosi a spotify.

Durante la creazione del progetto abbiamo scelto di usare un solo file JS dove inserire tutte le funzioni principali che vengono usate per rendere dinamico il sito, ma molte funzioni piccole sono presenti in tutti i file html.

Per quanto riguarda il token dell'API di spotify, ogni volta che usiamo un endpoint della sua API viene controllato se il token è ancora valido. Se si, si esegue la funzione, se no, si richiede un nuovo token, salvandolo nel localStorage del browser, e si richiama la funzione. Per tutti gli altri errori di Spotify abbiamo segnalato tali errori quando avvengono. Nel localStorage non viene salvato solo il token, ma anche l'id_user dell'utente.

E' possibile, se si vuole, uscire dal proprio account per entrare con un'altro.

Per quanto riguarda la 'responsività' del sito, abbiamo dovuto usare delle regole css per renderlo responsive a causa di componenti aggiuntivi creati da noi. Le regole usate sono poche e si applicano su pochissimo componenti.