

# Relazione Progettuale

## Progetto di Programmazione Web e Mobile

### Social Network for Music (SNM)

#### Introduzione

Il presente documento descrive il progetto di sviluppo di un'applicazione web per la gestione di playlist musicali. Il progetto è stato realizzato utilizzando tecnologie web come **HTML**, **CSS**, **JavaScript** e **Bootstrap** per il **frontend**, e **Node.js**, **Express** e **MongoDB** per il **backend**. L'applicazione offre agli utenti la possibilità di creare, modificare, cancellare, visualizzare e importare playlist musicali, oltre a gestire brani musicali attraverso l'integrazione con la **Spotify API**.

#### Obiettivi del Progetto

Gli obiettivi del progetto sono stati:

- Sviluppare un'applicazione web per la gestione di playlist musicali che sia facile da utilizzare e intuitiva;
- Implementare funzionalità per la creazione, modifica, cancellazione, visualizzazione e ascolto delle playlist e dei brani musicali;
- Consentire agli utenti di importare nel proprio profilo le playlist pubbliche di altri utenti.

## Tecnologie Utilizzate

Le tecnologie utilizzate per lo sviluppo dell'applicazione sono state:

- **Frontend:** HTML, CSS, JavaScript e Bootstrap;
- **Backend:** Node.js, Express e MongoDB;
- **API:** Spotify API per l'integrazione con la piattaforma musicale.

## Struttura dell'Applicazione

L'applicazione è strutturata in diverse pagine web per gestire le operazioni relative alle playlist e agli utenti.

**login.html:** Consente agli utenti di effettuare l'accesso inserendo email e password.

**registrati.html:** Permette agli utenti di registrarsi all'applicazione inserendo username, email, password, sesso, fino a tre artisti preferiti e generi musicali.

**profile.html:** Mostra le informazioni dell'utente e consente di modificarle, effettuare il logout o cancellare l'account. L'utente può modificare username, email, password e preferenze su artisti e generi.

**playlist.html:** Visualizza le anteprime delle playlist dell'utente e consente di creare nuove playlist o modificare, cancellare e visualizzare quelle esistenti.

**home.html:** Contiene due campi di input: uno per inserire il titolo della traccia e uno per il nome dell'autore. Gli utenti possono inserire i termini di ricerca e cliccare su un pulsante di ricerca per visualizzare una lista di tracce corrispondenti recuperate dall'API di Spotify e consente di ascoltarle e aggiungerle a una playlist.

**playlistpub.html**: Questa pagina consente agli utenti di esplorare le playlist pubbliche attraverso due sezioni di ricerca flessibili che permettono di cercare playlist per tag, traccia selezionata o una combinazione di entrambi. Entrambi i criteri sono opzionali, consentendo una maggiore personalizzazione nella ricerca.

- **Selezione della Traccia da cercare:**

- La prima sezione fornisce due campi di input: uno per inserire il titolo della traccia e uno per il nome dell'autore.
- Gli utenti possono inserire i termini di ricerca e cliccare su un pulsante di ricerca per visualizzare una lista di tracce corrispondenti recuperate dall'API di Spotify.
- Da questa lista, è possibile selezionare una traccia specifica che si intende usare come criterio opzionale per la ricerca delle playlist nella seconda sezione.

- **Ricerca di Playlist per Tag e/o Traccia Selezionata:**

- La seconda sezione offre una barra di ricerca in cui gli utenti possono inserire un tag specifico per cercare playlist correlate.
- Sotto al campo di input per il tag, è presente una sezione che mostra il nome della traccia precedentemente selezionata nella prima sezione, se disponibile.
- Gli utenti possono scegliere di cercare le playlist usando solo il tag, solo la traccia selezionata, o una combinazione di entrambi.
- Una volta impostati i criteri desiderati, gli utenti possono cliccare su un pulsante per avviare la ricerca. I risultati mostreranno le anteprime delle playlist che soddisfano i criteri selezionati, consentendo agli utenti di esplorare e importare playlist di loro interesse.

**hoverplaylist.html**: Consente di visualizzare, modificare, cancellare e ascoltare una playlist selezionata, accessibile tramite un bottone "**Vedi Playlist**" dalle anteprime delle playlist.

## Backend

Il backend dell'applicazione è stato realizzato utilizzando **Node.js**, **Express** e **MongoDB**. Le API sono state create per gestire le operazioni di creazione, modifica e cancellazione di **playlist**, **brani** e **utenti**, nonché per l'autenticazione degli **utenti**.

## Database

Il database è stato realizzato con **MongoDB** e include due collezioni principali:

- **users**: Contiene i documenti relativi agli utenti.

```

_id: ObjectId('66d8cd186eee90ae05e52afb')
username: "Marco"
password: "*****"
email: "marco.castucci@studenti.unimi.it"
sex: "Uomo"
▼ artists: Array (3)
  ▼ 0: Object
    id: "19i93sA0D7yS9dYoVNBqAA"
    name: "thasup"
    ▶ images: Array (3)
  ▼ 1: Object
    id: "3l900qBvHs4SM91bWqtaQb"
    name: "yungest Moonstar"
    ▶ images: Array (3)
  ▼ 2: Object
    id: "3fMbdgg4jU18AjLCKBhRSm"
    name: "Michael Jackson"
    ▶ images: Array (3)
▼ genere: Array (3)
  ▼ 0: Object
    name: "hip-hop"
  ▼ 1: Object
    name: "pop"
  ▼ 2: Object
    name: "k-pop"

```

- **playlists**: Contiene i documenti relativi alle playlist.

```

_id: ObjectId('66d3ae45f0e8afa5cce54f25')
title: "PlaylistPorva"
userId: "66d8cd186eee90ae05e52afb"
creatorName: "Marco"
description: "Porva"
numberOfTracks: 3
duration: 475251
▼ tracks: Array (3)
  0: "0EimIRiiRd76tqlo9LPLUJ"
  1: "0NUzLcrkkXeztSVLJYH0ug"
  2: "5lWJY6ZvKEcza921CaWL5I"
▼ tags: Array (2)
  0: "pop"
  1: "trap"
isPublic: true

```

## Endpoint e Operazioni Disponibili

Le richieste HTTP per interagire con le operazioni definite nel file `index.js` possono essere effettuate utilizzando i seguenti endpoint:

- **GET:**

- `/`: Reindirizza alla pagina di login.
- `/profile/:id`: Recupera le informazioni di un utente tramite il suo ID.
- `/playlists/:userId`: Recupera tutte le playlist associate a un utente tramite il suo ID.
- `/playlist/public/:userId`: Recupera tutte le playlist pubbliche e le playlist dell'utente tramite il suo ID.
- `/playlists/:id`: Recupera le informazioni di una singola playlist tramite il suo ID.
- `/playlist/:title/:userId`: Verifica se esiste una playlist pubblica o dell'utente con il titolo fornito.

- **POST:**

- `/register`: Registra un nuovo utente.
- `/login`: Autentica un utente.
- `/createplaylist`: Crea una nuova playlist.
- `/playlist/:playlistId/track/:trackId/:trackDuration`: Aggiunge una traccia a una playlist tramite il suo ID e la durata.
- `/playlist/:userId`: Restituisce una lista di playlist che soddisfano i criteri di ricerca forniti tramite il body e l'ID dell'utente.

- **PUT:**

- `/modificauser/:id`: Aggiorna le informazioni di un utente tramite il suo ID.
- `/modificaplaylist`: Aggiorna le informazioni di una playlist.

- **DELETE:**

- `/delete/:id`: Elimina un utente tramite il suo ID.
- `/playlist/:id/:userId`: Elimina una playlist tramite il suo ID verificando se l'utente è autorizzato a eliminarla.
- `/playlist/:playlistId/track/:trackId/:trackDuration`: Rimuove una traccia da una playlist tramite il suo ID.

## Conclusione

L'applicazione realizzata consente agli utenti di creare, modificare, cancellare e ascoltare playlist musicali, gestire brani, e importare nel proprio profilo le playlist pubbliche di altri utenti. Grazie all'integrazione con la Spotify API, gli utenti possono cercare, aggiungere e ascoltare brani direttamente nell'applicazione, migliorando l'esperienza d'uso.