

# GrooveMuze (Artists directory)

## Funktionalitet:

- **CRUD-operationer:** Applikationen leverer komplette CRUD-operationer (opret, læs, opdater og slet) for kunstnerhåndtering.

## Backend:

- **REST API:** Implementeret med Node.js og Express.js med routes for HTTP-metoderne GET, POST, PUT/PATCH og DELETE.
- **Datakilde:** JSON-fil bruges som datakilde. Artist-objekterne i JSON-listen indeholder mindst følgende properties: name, birthdate, activeSince, genres, labels, website, image, shortDescription.
- **Datahåndtering:** CRUD-operationer der læser og skriver til JSON-fil. Mulighed for at hente alle objekter samt et enkelt objekt baseret på et specificeret id.

## Frontend:

- **User Interface:** Implementeret med HTML, CSS og JavaScript.
- **Funktionalitet:** Brugere kan udføre CRUD-operationer, filtrere og sortere efter bestemte parametre, markere en kunstner som favorit og vise en liste over alle favoritkunstnere.
- **Design:** Anvender CSS Grid, CSS Flex og/eller HTML Table samt relaterede HTML-elementer.
- **Modulopbygning:** Kode er opdelt i moduler for bedre struktur og vedligeholdelse.

## Adgangskontrol og Rettighedsstyring:

- Admin Adgang:
  - Der er implementeret en admin login-funktionalitet.
  - Brugerrettigheder: Når en bruger er logget ind som admin, får de specifikke rettigheder. Dette inkluderer muligheden for at slette og redigere kunstnerprofiler.
- **Loginoplysninger:**

Brugernavn: admin

Adgangskode: password123
- Brugere uden admin rettigheder kan fortsat udføre basis CRUD-operationer som defineret, men vil ikke have adgang til at slette eller redigere eksisterende kunstnerprofiler.

## Brugervenlighed:

- **Navigation:** Brugergrænsefladen er designet til at være nem at navigere med klare vejledninger for sortering, filtrering og tilføjelse af favoritkunstnere.

## Pålidelighed:

- **Data Validering:** Systemet validerer JSON-data for korrekthed ved CREATE og UPDATE funktioner.

- **Fejlhåndtering:** Backend leverer fejlbeskeder, hvis brugere forsøger at bruge GET, PUT eller DELETE metoder på ikkeeksisterende kunstner-id'er i JSON-filen.

#### **Performance:**

- Selvom der ikke er fastlagte ydelseskrav, forventes det, at applikationen leverer hurtig datahentning og responsivitet i brugergrænsefladen.

#### **Supportabilitet:**

- **Kode Struktur:** Ved at opdele koden i moduler forbedres struktur og vedligeholdelsespotentialer.
- **Videreudviklingspotentialer:** Koden kan nemt udvides, for eksempel ved at tilslutte gemte data til en database eller ved at hoste siden med et domæne.
- **Dokumentation:** I GitHub-repositoryet inkluderer en README-fil, der giver en guide til opsætning og installation af applikationen.