

Artist-App

Applikationens krav specificeret ud fra FURPS+ principperne

- **Functionality:**

Applikationen skal med sine create, read, update og delete funktioner opfylde kravene for CRUD-operationer. De nævnte funktioner skal spille sammen med get, post, put og delete metoderne i backenden

- **Backend**

- REST API implementeret med Node.js og Express.js
 - Routes med endpoint for HTTP-metoderne GET, POST, PUT/PATCH, DELETE.
 - CRUD-operationer der læser og skriver til JSON-fil.
 - Det skal være muligt at kunne få både alle objekter og et objekt på baggrund af specificeret id.
- JSON-fil som datakilde
 - Artist-objekterne i JSON-listen skal bestå af minimum følgende properties: name, birthdate, activeSince, genres, labels, website, image, shortDescription

- **Frontend**

- User Interface implementeret med HTML, CSS og JavaScript
- Brugeren skal kunne oprette, læse, opdatere og slette data (CRUD)
- Filtrering og sortering på udvalgte parametre (props)
- Du skal kunne markere en kunstner som favorit og vise en liste over alle favoritkunstnere. Du bestemmer selv, hvordan denne liste gemmes (backend, localStorage, variabel eller lignende).
- Anvendelse af CSS Grid, CSS Flex og/eller HTML Table samt relaterede HTML-elementer
- Kode opdelt i modules

- **Usability:**

Brugergrænsefladen skal være brugervenlig og nem at navigere.

Mulighederne for sortering, filtrering samt tilføjelse af kunstnere til en oversigt af valgte favoritter, skal være intuitive og tydeliggjorte for brugeren.

- **Reliability:**

Koden skal kunne forholde sig til, hvorvidt den JSON-data som bliver sendt i CREATE og UPDATE funktionerne er i korrekt format inden det bliver indsat i JSON-filen. Derudover skal backenden fejlmelde i tilfælde af at brugeren prøver at gøre brug af GET, PUT og DELETE metoderne på kunstnere gennem ikkeeksisterende ID's i JSON-filen.

- **Performance:**

Kravene til applikationens ydeevne er ikke blevet eksplicit defineret i opgavebeskrivelsen. Der bliver dog nævnt at applikationen skal have en acceptabel ydeevne, herunder hurtig datahentning og responsivitet i brugergrænsefladen.

- **Supportability:**

Koden skal være opdelt i modules for at bygge struktur og simplificere projektet. Koden skal også kunne videreudvikles på simpel vis f.eks. ved at hooke den gemte data op på en database og evt. hoste siden med et domæne. I github-repoet skal der også ligge en readme-fil med en guide til opsætning og installation af programmet.