

Functionality:

- **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:

Jeg har med tanke på 10 golden rules og gestalt lovene i tankerne, prøvet at gøre mit UI så simpelt og konsekvent som muligt. Knapper der opretter, opdaterer og konfirmerer, ikke slettende funktioner, er farvet grønne. Der er muliggjort reversal for brugeren med Knapper der går tilbage(close). Close og delete knapper er røde. Delete er mere rød end close for at indikere at her går hele artisten tabt. Knapper med samme funktion er også samme sted dvs. close til venstre og confirm til højre.

Hvad angår feedback til brugeren er favoritknappen implementeret så den ændrer farve når den trykkes for at indikere at artisten er blevet tilføjet. Knappen ændrer også navn til Unfavorite så brugeren er klar over at et nyt klik vil fjerne favorit statusen. Ligeledes var tanken at der skulle være et feedback når brugeren sletter, der popper op, men siden reloader, således at modalen står der meget kortvarigt og jeg kan ikke gennemskue hvorfor.

- Reliability:

Eftersom data er gemt via en API skulle data være beskyttet i tilfælde af netværksforbindelsen går.

- Performance
- Supportability

Der er gjort forsøg på at inddеле backend koden i så små funktioner som muligt således der opstår loose coupling og funktionerne er nemmere at læse og derfor opdatere.

- Forbedringer til næste gang

Jeg vil til næste projekt ikke blot overveje FUPRPS inden jeg programmerer appen, men systematisk udfylde hver eneste punkt i 10 golden rules og lave en skitse til en prototype inden jeg går i gang.

Ligeledes vil jeg præsentere min app til andre fx medstuderende for at lave en usability test, og fastsætte krav til hvad intuitivt er.

Det er også tanken at tage krav og lave dem til user stories, og præsentere det for en af underviserne med henblik på om jeg har omskrevet kravene adækvat. Tanken med dette er at træne denne omskrivning af krav.