Functionality:

1. CRUD-operationer der læser og skriver til JSON-fil.
2. Brugeren skal kunne oprette, læse, opdatere og slette data (CRUD)
3. Filtrering og sortering på udvalgte parametere (props)
4. 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).
5. Det skal være muligt at kunne få både alle objekter og et objekt på baggrund af specificeret id.
6. Artist-objekterne i JSON-listen skal bestå af minumum følgende properties: name, birthdate, activeSince, genres, labels, website, image, shortDescription

JSON Datakilde


```
 {
    "id": "01",
    "name": "David Bowie",
    "birthdate": "1947-01-08",
    "activeSince": "1962-07-15",
    "genres": "Rock,Pop,Experimental",
    "labels": "RCA Records,EMI,Columbia Records",
    "website": "https://www.davidbowie.com/",
    "image":
"https://upload.wikimedia.org/wikipedia/commons/d/d0/David_Bowie_-
_TopPop_1974_03.png",
    "roles": "Singer,Songwriter,Actor",
    "shortDescription": "Iconic musician and actor",
    "favorites": false
  },
```


Usability:

1. User Interface implementeret med HTML, CSS og JavaScript
2. Anvendelse af CSS Grid, CSS Flex og/eller HTML Table samt relaterede HTML-elementer
3. 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).
4. Dokumentation af installation og hvordan man køre appen

Reliability:

1. Routes med endpoint for HTTP-metoderne GET, POST, PUT/PATCH, DELETE.
2. CRUD-operationer der læser og skriver til JSON-fil.
3. Det skal være muligt at kunne få både alle objekter og et objekt på baggrund af specificeret id.

Performance:

1. Separation of Concerns – så ting er adskilt så meget som muligt, I bruger forskellige funktioner/moduler til at manipulere data og vise data.
2. Loose Coupling – så funktioner er så uafhængige af hinanden som muligt, eller i det mindste kun har afhængigheder en vej.
3. High Cohesion – så funktioner der arbejder med det samme er samlet så tæt som muligt, enten i closures eller i modules.

Supportability:

1. Kode opdelt i modules
2. Separation of Concerns
3. Loose Coupling
4. High Cohesion
5. Skrive kommentare til funktioner og hvad de gør i applikationen