

- **Functionality:** CRUD-operationer, filtrering, sortering og præsentation af kunstnerdata **CHECK**: Har lavet API endpoints til POST, GET, PATCH og DELETE til Artists. Der er også filtrering efter genre og søgning efter Artist navn. Artists er vist i en liste i html i artistsList.html og på hver sin side dynamisk i artistPage: [http://localhost:3000/\(artistnavn\)](http://localhost:3000/(artistnavn))
- **Usability:** Brugergrænsefladen skal være brugervenlig og nem at navigere. Alle CRUD-operationer skal være let tilgængelige, og filtrerings/sorteringsfunktionerne skal være intuitive. **CHECK**: Brugergrænsefladen er brugervenlig og nem at navigere. Alle navne på kaldene giver mening og det er nemt at se hvad der er hvad på siden.
- **Reliability:** Applikationen skal være stabil og pålidelig. Dataintegritet og korrekt håndtering af CRUD-operationer er afgørende. **CHECK**: Det er en lille og meget simpel Applikationen så den stabil og pålidelig som den er.
- **Performance:** Applikationen skal have en acceptabel ydeevne, herunder hurtig datahentning og responsivitet i brugergrænsefladen. **CHECK**: Siden den ikke rigtig skal vente på svar da json ligger i localStorage så er det nemt at holde en acceptabel ydeevne under datahentning.
- **Supportability:** Koden skal være velstruktureret og veldokumenteret, så det er nemt for fremtidige udviklere at vedligeholde og udvide applikationen. **CHECK**: De vigtigste dele af koden bla. Endpoints er dokumenteret så en anden ville kunne tage over og forstå det.
- **+**: kodestruktur, modularitet og afhængighedsstyring, herunder Separation of Concerns, Loose Coupling og High Cohesion. **CHECK**: Der er ikke meget af koden som er afhængighed af andre dele i denne størrelse applikation som ville kunne gøre den vanskelig at vedligeholde.