Ali Mohammad 06/06/2023

FURPS+ - Ali Mohammad

Functionality (F):

Backend:

- 1. Implementering af REST API med Node.js og Express.js.
- 2. Oprettelse af ruter med endpoints for HTTP-metoderne GET, POST, PUT/PATCH og DELETE.
- 3. Implementering af CRUD-operationer (Create, Read, Update, Delete) for at læse og skrive til JSON-fil.
- 4. Mulighed for at hente alle objekter og et objekt baseret på et specificeret id.
- 5. Definering af JSON-fil som datakilde.
- 6. Oprettelse af Artist-objekter med disse properties som minimum: name, birthdate, activeSince, genres, labels, website, image, shortDescription.

Frontend:

- 1. Implementering af brugergrænseflade med HTML, CSS og JavaScript.
- 2. Mulighed for at oprette, læse, opdatere og slette data (CRUD-operationer).
- 3. Implementering af filtrering og sortering baseret på udvalgte parametre.
- 4. Mulighed for at markere en kunstner som favorit og vise en liste over favoritkunstnere. Samtidig mulighed for at fjerne en kunstner som favorit.

Usability (U):

Backend:

- 1. Implementering af pålidelige CRUD-operationer, der læser og skriver til JSON-filen.
- 2. Sikre fejlhåndtering i API-routes for at minimere nedetid og sikre, at brugerne får pålidelige svar fra serveren.

Frontend:

1. En brugervenlig grænseflade med HTML, CSS og JavaScript for nem navigation og interaktion med data.

Ali Mohammad 06/06/2023

2. Implementering af filtrering og sortering for at forbedre brugeroplevelsen.

Reliability (R):

Backend og Frontend:

- 1. Applikationen skal være stabil og pålidelig.
- 2. Dataintegritet og korrekt håndtering af CRUD-operationer er afgørende.

Performance (P):

Backend og Frontend:

- 1. Optimering af API-anmodninger og responstider for at opnå en god ydeevne, især når der arbejdes med store datamængder.
- 2. Brug af CSS Grid, CSS Flex og/eller HTML Table og relaterede HTML-elementer for at opnå effektiv præsentation af data.

Supportability (S) og +:

Backend og Frontend:

- 1. Anvendelse af generelle principper som "Separation of Concerns," "Loose Coupling," og "High Cohesion" for at gøre koden mere vedligeholdelsesvenlig og forståelig.
- 2. Opdeling af kode i moduler for bedre organisering og vedligeholdelse.