

1. Functionality

1.1 Krav

Frontend Funktionalitet

Implementer en brugervenlig brugergrænseflade (UI) med JavaScript-moduler.

Indlæs data fra backenden via API-kald og vis dem i frontend.

Mulighed for at oprette, læse, opdatere og slette ressourcer (f.eks. brugere eller poster) gennem brugergrænsefladen.

Implementer autentificering og autorisation for brugeradgang.

Backend Funktionalitet:

Udvikl API-endepunkter til at håndtere frontend-anmodninger.

Implementer datahåndtering, inklusive CRUD-operationer (Create, Read, Update, Delete).

Sikkerhedskontroller forhindrer uautoriseret adgang.

1.2 Status

Frontend og backend funktionalitet er blevet implementeret i overensstemmelse med de specificerede krav.

Brugergrænsefladen er responsiv funktionel.

Autentificering og autorisation er implementeret ikke implementeret, da vi ikke er blevet introduceret til dette.

2. Usability

2.1 Krav

Frontend skal have en intuitiv og brugervenlig grænseflade.

Brugere skal kunne navigere nemt og forstå systemets funktioner uden problemer.

2.2 Status

UI-design og layout er designet simpelt for at gøre brugervenligheden intuitiv og overskuelig.

Min kæreste udførte en "brugertest", og kunne selv tænke sig frem til al funktionalitet!

3. Reliabilitet (Reliability)

3.1 Krav

Systemet skal være stabilt og pålideligt uden hyppige fejl.

Dataintegritet og -sikkerhed skal opretholdes.

3.2 Status

Projektet er blevet testet for at identificere og rette fejl.

Systemet har været stabilt under testperioden.

Jeg har lavet en "backup.json" med original data, for at sikre at jeg kan genoprette en funktionel datakilde.

4. Performance

4.1 Krav

Frontend og backend skal have en acceptabel svartid og ydeevne.

Systemet skal skalere effektivt, hvis antallet af brugere øges.

4.2 Status

Systemet har vist god ydeevne inden for testperioden.

Der er taget brug af node.js som kan håndtere multithreading, hvilket sikrer mulighed for skalering

5. Supportability

5.1 Krav

Koden skal være velstruktureret og dokumenteret.

Der skal være en supportplan for vedligeholdelse og fejlrettelse.

5.2 Status

Kildekoden er organiseret i moduler og dokumenteret med kommentarer i koden.

Der er ikke en supportplan for vedligeholdelse af systemet, da det ikke skal tages i brug.

6. +

6.1 Krav

Koden skal overholde de relevante standarder og best practices for JavaScript og Node.js-udvikling.

6.2 Status

Koden er opdelt med tanken om separation of concern, både via moduler, og express.js funktionalitet.