

Dancing Docs – Svendepøve WU06 10/11 oktober 2022

/*****

LANDRUP DANS

stack:

reactjs, react-router-dom, react-DOM, useLocation, useParams
tailwindcss, framer-motion
axios

github

react-bibliotekerne er komponent-baserede, 'augmenteret' js, der ser til, at html, css og js 'flyder sammen'.

tailwindcss er et library af 'utility classes', shorthand snippets for css, implementeret 'inline'.

framer-motion er et 2D (og 2.5 D) animationsbibliotek, også meget shorthand-orienteret.

axios er en på overfladen letvægts-version af fetch (de 'lever' begge af at udstede requests til API'er, kort sagt). axios udmærker sig bl.a. ved selv at finde ud af at parse/stringify JSON-objekter under hjelmen.

github er naturligvis 'go-to' for versions- og backup-styring.

proces:

afvikling:

github 'push til dev branch' (og senere pull request og merge ovf. main).

Jeg har ikke brugt KanBan denne gang; min styring har været en blyant og en a4-blok.

udvikling/stack perspektivering:

Denne del af processen vil jeg lade springe direkte til forholdet mellem trekløveret 'stack perspektivering' vs. 'udvikling af/i koden' vs 'hvad er der af tid';

Når tiden er knap, betyder det, lidt skarpt sat op, at koderne bliver stressede, at de ikke når at udvikle eller implementere den til opgaven, eller koderens behov, bedst egnede 'struktur' eller 'metode', og/eller at der bliver uforholdsmæssigt stort fokus på de kvikke løsninger, ikke fordi de er gode, men fordi de er kvikke.

Jeg vil ikke skrive mere her, men hellere forsøge at uddybe teksten her ved min fremlægning, ved blandt andet at beskrive min egen foretrukne arbejdsproces.

projekt-perspektivering:

i forhold til min aflevering kunne man forestille sig en ekstra 'refactoring', der kunne fjerne min 'inline-css', der er tydeligt apokryf, til fordel for ren tailwind eller ren sass, eller noget andet, så længe det er 'rent'.

Derudover har jeg en fornemmelse af 'stivhed' i side-skiftene. Man kunne måske kigge på at animere en lille smule på alle transitions :)
