

Den Grønne Avis

Rolf Mousten

Mit holdnummer: H1WE010124

GitHub Repository: <https://github.com/Rolfie1977/Den-Gr-nne-Avis>

Indholdsfortegnelse:

1. Indledning
 2. Vurdering af min egen indsats
 3. Valg, argumentation og begrundelse
 - 3.1 Teknologi valg
 - 3.2 Tidsforbrug
 - 3.3 Opgave rækkefølge
 - 3.4 Problemer undervejs
 4. Redegørelse af kodeelementer
 5. Fremhævelse af særlige punkter til bedømmelse
 6. Afsluttende bemærkninger
-

1. Indledning:

I denne rapport beskriver jeg arbejdet med Javascript React projektet, svendeprøven - Den Grønne Avis. Den gennemgår indsats, valg af teknologier, nogle kodeelementer og nogle af mine mange tanker gennem disse 5 dage.

2. Vurdering af egen indsats:

Jeg startede hver morgen med at arbejde på projektet. Typisk kiggede jeg det igennem, som jeg havde lavet dagen før for at lave en reminder om det hele. Dernæst gik jeg så i gang med dagens kodning. Jeg fandt ud af, at jeg begyndte at være meget træt i hovedet hver dag ved en kl. 15 tiden, så jeg forsøgte at nå mest muligt af selve koden fra om formiddagen og frem til min frokost, og så et par timer efter frokost.

3. Valg, argumentation og begrundelse:

3.1 Teknologi valg:

Til udarbejdelse af Den Grønne Avis, der har jeg af tech stack:

- **Styling:** SCSS
- **Frameworks:** React.js
- **Database:** MySQL
- **Værktøjer:**
 - Vite til opbygning af React projekt
 - Axios (enklere og mere elegant til API-kald)
 - VS Code

- Github
- React Router (til håndtering af routes, altså håndtere navigation mellem sider)
- React Hooks (useState, useEffect samt useParams)
- Postman (til at teste min API kald)
- MySQL Workbench (database administration - oprettelse af database)
- **NPM Pakker:**
 - SCSS-embedded(forbedret SCSS understøttelse i Vite)
 - Axios (Håndtering af API kald)
 - React-router-dom (Håndtering af routes)
- **Kodeteniker og implementerede funktioner:**

Kan kort nævne, at jeg til login har brugt Async/ Await til fetch som er brugt til loginsystemet til at håndtere fetch-kald. Jeg har brugt det til at vise, at jeg også forstår moderne Javascript og asynkron programmering.

Jeg har endvidere også valgt at bruge Axios til at fetch, da det er effektivt til API kald, og jeg synes, at det er overskueligt til fejl og error handling. Og koden er lidt kortere. Til login systemet har jeg som nævnt brugt await fetch, da det er det, som vi har haft undervisning i der.

3.2 Tidsforbrug: Tidsplan.pdf

3.3 Opgave rækkefølge:

1. Forside
2. Kategori side
3. Gøre mine kategori cards på forsiden clickable
4. Andre små forbedringer (Link på forside logo)
5. Login side
6. Responsivitet på kategori siden

3.4 Problemer undervejs:

1. Da jeg startede med at sætte det hele op i React med Vite. Lavede jeg det om første gang, da jeg kom til at opsætte det i React Vue. Slettede derfor alle filer – discard all changes i github desktop. Dernæst rensede jeg ud i filerne, som jeg ikke skulle bruge, altså css, filer etc. Som vi altid skal gøre med Vite i React.
2. Tænkte faktisk meget over valg af **Axios**, da vi mere har arbejdet med Async/await, men jeg synes, at Axios er lidt kortere og mere enkel.
3. **Footer input placeholder** - footer placeholder har ingen placeholder tekst på designet, men jeg skønnede at dette var mere intuitivt at have.

4. Redegørelse af kodeelementer:

1. Axios fetch: (fetcher data fra APIet)

- Kilde:
<https://www.geeksforgeeks.org/difference-between-fetch-and-axios-js-for-making-http-requests/>
- Frontpage.jsx
 - i. Fra linje 13 til linje 20
 - ii. Fra linje 24 til linje 33
- Kategoripage.jsx
 - i. Fra linje 16 til 26

2. Media Queries

- Kilde: https://www.w3schools.com/css/css3_mediaqueries.asp
- Footer.module.scss
 - i. Linje 38
- Loginform.module.scss
 - i. Linje 33
- Frontpage.module.scss
 - i. Linje 68
- Kategoripage.module.scss
 - i. linje 12
- header.module.scss
 - i. Fra linje 31

5. Særlige punkter til bedømmelse:

1. Axios fetch på kategori siden

- fra linje 16 til 26

Jeg vil gerne forklare, hvad jeg godt kan lide ved det. Synes det er ret elegant, og gør API kaldet dynamisk

6. Afsluttende bemærkninger:

Projektet er gået bedre end, jeg troede og selvom jeg ikke nåede, hvad jeg håbede på, så er jeg stadig stolt og tilfreds med hvad jeg har nået at lave.. Jeg havde hoppet at nå at lave

produktsiden med de enkelte produkter, men jeg valgte at fokusere på noget, der var ordentligt i stedet for nogle flere sider, der var halvfærdige. Men det har været hårdt, og har været meget udkørt hver eftermiddag.

—

