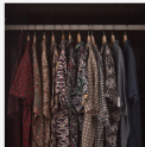
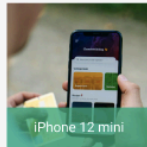


Udvalgte Produkter



Den Grønne Avis

Vi går forest i kampen om klimaet ved at give 2 kr. til klima-venlige formål, hver gang du handler brugt på Den Grønne Avis

Den Grønne Avis

H1WE010124

14. marts 2025

Michael Poulsen - H1WE010124

<https://github.com/michmich16/Den-Groenne-Avis---Svendeproeve>

login til side: info@webudvikler.dk & password

database: DBNAME = DGA DBUSER = root DBPASSWD = test1234

Dokumentation for HomeLands.....	2
Evaluering af egen indsats.....	2
Redegørelse for kodeelementer.....	2
Fremhævelse af punkter til bedømmelse.....	3
Beskrivelse:.....	3
Forsiden:.....	3
CategoriesPage:.....	3
ProductPage.....	3
LoginPage & SignUpPage.....	4
AccountPage.....	4
NewProductPage.....	4
Bilag.....	4

Dokumentation for Den Grønne AVIS

Udvikling:

- ReactJS framework
- HTML, SCSS og JavaScript

Database:

- Database er opsat på local MySQL
- API Data er udleveret fra <https://github.com/TCAA-Web/DGA-API>

Kilder:

- Font: Roboto
- <https://documenter.getpostman.com/view/31531123/2sAYdbND6z>
- <https://www.figma.com/design/8KALqehT0mXp5ld3i5ZcFR/Den-Gr%C3%B8nne-Avis?node-id=0-1&t=vy1PIC9JoH4nngRQ-1>

Evaluering af egen indsats

Jeg synes, det er gået fint, men der er stadig små ting, som jeg ikke har fået lavet færdige, f.eks. styling til kommentarer, MyListing, der fetcher alle annoncer (de skal kun fetches, hvis de har samme user_id som den bruger, der er logget ind), timestampet, som ikke er, som det ser ud i designet, nyhedsbrevsfunktionen i footeren og et "vælg kategori"-input i -

NewProductPage (jeg skulle have brugt en select, men lige nu skal man indtaste kategori-ID for at sende en post request).

Det skyldes blandt andet, at jeg var syg i en halv dag i mandags og var nødt til at gå tidligt hjem. Men ellers er jeg meget tilfreds og stolt over det, jeg har nået. Jeg har fokuseret meget på at få alle funktionerne på plads, og derfor er designet ikke helt 100 % som i Figma. Alt i alt synes jeg selv, at jeg har gjort det godt.

Redegørelse for kode elementer

Jeg har anvendt Hooks (useGet) til at lave en fetch, hvor den sender en GET-request til en URL (API-link) for at hente data derefter render dem ned i en komponent i DOM.

Fremhævelse af punkter til bedømmelse

Til mundtlige eksamen vil jeg gerne fortælle om React Hooks, specifikt useGet fordi jeg synes, de er smarte og kan genbruges hver gang, man skal fetch data. Man skal bare skifte URL'en, så fungerer det med det samme. Derudover så vil jeg også gerne snakke om Login funktion hvor man sender en post request osv. og hvis der er tid tilbage så vil jeg snakke om Delete User funktionen.

Beskrivelse:

Sitet er opbygget af HTML, CSS og Javascript med ReactJS framework. Alle data og information i sitet er fra en API som jeg har forket fra github og kører via min local MySQL.

Forsiden:

- Header med Logo og Dropdown (select med options som fetchet fra API), en user ikon som redirect til LoginPage og en opret annonce knap som redirecter til NewProductPage.
- Card components som hente 6 tilfældige produkter fra API'et
- Statisk banner ligesom i designet
- Card components som hente 6 tilfældige kategorier fra API'et
- 2 Statisk banners ligesom i designet
- Footer med et link som redirecter til verdensmål hjemmeside

CategoriesPage:

- Cards component som henter tilhørende produkter fra API'et.
- Side-kategorier (CategoryFilter) som redirect til CategoriesPage igen men med ny value i slug.
- Pagination.

ProductPage

- ProductDetails component som henter en Data med ID fra API'et og render i en dom.
- Comments sections som er hentet fra API.
- Comment input som sender en post request til API (kræver at man er logget ind eller have access_token)

LoginPage & SignUpPage

- En loginform som sender en post request til API og tjekker hvis email og password er der, hvis ja så returnere den en access_token tilbage.
- En signup-form som sender en post-request med body til API.

AccountPage

- En ProfileDashboard med forms som man kan bruge til at sende en patch request til API hvis man skal redigerer user data, en delete knap som sender delete request til API for at slette user.
- MyListings component som henter produkter fra API. Der er også en "slet annonce" knapp som sender en delete request til api (kræver header med access_token)

NewProductPage

- input component som sender post request til API for at opret en produkt (kræver at man er logget ind og have access_token)

Bilag.

Tidsplan.

	Mandag	Tirsdag	Onsdag	Torsdag	Fredag
08:00	Opret github repo.	Forsiden og card components	Comments	AccountPage	Styling
09:00	læs opgaven og lav planlægning	Forsiden og card components	LoginPage	AccountPage	Styling
10:00	Header, Navigation og Banner	Forsiden og card components	LoginPage, SignUpPage	AccountPage	Styling
11:00	Footer	CategoriesPage	SignUpPage	MyListing	Styling til mobile
12:00	syg	CategoriesPage	SignUpPage, Validation	MyListing	Styling til mobile
13:00	syg	ProductPage	NewProductPage	ProfileDashboard	Rapport
14:00	syg	ProductPage	NewProductPage	ProfileDashboard	Rapport
15:00	syg	Comments	NewProductPage	ProfileDashboard	
16:00		Comments	NewProductPage	Styling	