MAKKA ISMAILOVA *HOLD: H1WE010124* *WEBUDVIKLER - MARTS 2025*

Den Grønne Avis

URL-adresser og adgangsoplysninger

https://github.com/ maylova93/den-groenne-avis

Brugernavn: info@webudvikler.dk

Adgangskode: password

1) ***Vurdering af egen indsats:*** *Projektet er lavet i HTML, Javascript, React, SCSS,*
2) Dette projekt startede med en grundig planlægning, hvor jeg organiserede opgavens struktur og fastlagde en arbejdsproces. Jeg har arbejdet systematisk for at sikre, at alle funktioner blev implementeret korrekt, og at projektet overholdt de tekniske krav.
3) **Mandag:** Jeg startede med at opsætte projektet og tilslutte backend. Jeg planlagde mappestrukturen for at sikre en logisk opdeling og begyndte at udvikle navigationen **Navigation** og footeren **Footer**. Derudover sikrede jeg, at grundlæggende routing fungerede korrekt i App.
4) **Tirsdag:** Jeg arbejdede med produktvisningen og kategorierne. **ProductList** blev implementeret for at hente og vise produkter fra API'et, og **KategoriListe** blev udviklet for at gøre navigationen mere brugervenlig. Jeg tilføjede også **HeroSection** for at styrke forsiden visuelt. Samtidig begyndte jeg at skrive den første del af rapporten.
5) **Onsdag:** Fokus var på at oprette og håndtere annoncer. Jeg udviklede **AnnoncePage**, hvor brugere kan oprette annoncer, og sikrede validering samt datahåndtering via API-kald. Jeg implementerede også **Donation**, som viser, hvor meget der er blevet doneret til klimavenlige formål gennem platformen. Herefter begyndte jeg at udvikle **LoginPage**, hvor jeg arbejdede med validering og loginfunktionalitet. Jeg testede API'et og sikrede, at login-data blev håndteret korrekt.
6) **Torsdag:** Jeg implementerede **ProductDetail**, hvor brugere kan se detaljer om et produkt. Derudover udviklede jeg **ContactSeller**, så brugere kunne kontakte sælgere direkte gennem platformen. Jeg havde nogle udfordringer med at sikre, at beskederne blev sendt korrekt og opbevaret i databasen, men jeg fik det løst gennem debugging og test. Jeg lavede også **GreenLine**, en genbrugelig komponent, der skaber en visuel adskillelse mellem sektioner på hjemmesiden.
7) **Fredag:** Jeg arbejdede på **MinSide**, hvor brugere kan se og redigere deres profil. Derudover udviklede jeg **MineAnnoncer**, som viser de annoncer, en bruger har oprettet. Jeg havde en del fejl i visningen af annoncer, men jeg fik det løst ved at tilpasse API-kaldet og sikre korrekt datahåndtering. På **SignUpPage** havde jeg udfordringer med validering og automatisk login efter oprettelse, men jeg løste det ved at tilføje en funktion, der sender brugeren direkte til **MinSide** efter en succesfuld oprettelse. Jeg testede funktionaliteten, foretog designforbedringer og optimerede koden. Samtidig skrev jeg videre på rapporten og færdiggjorde den til aflevering.

Argumentation for de valg du har truffet under løsningen af opgaven

Jeg valgte **React** som frontend-framework, da det giver en struktureret og fleksibel måde at håndtere komponenter og state på. **React Router** blev brugt til at navigere mellem siderne, hvilket giver en mere dynamisk brugeroplevelse.

Stylingen er lavet med **SCSS**, fordi det giver bedre organisering af designet og mulighed for genbrug af variabler, hvilket sikrer et ensartet layout. Jeg har også fokuseret på at gøre hjemmesiden responsiv, så den fungerer både på desktop og mobil.

For at håndtere API-kald har jeg brugt **fetch** sammen med custom hooks som useGet() og usePost(). Dette gjorde koden mere overskuelig og genanvendelig, især ved indhentning og oprettelse af data.

Oprettelse af annoncer i AnnoncePage krævede særlig opmærksomhed på inputvalidering og datahåndtering, så brugeren får en god oplevelse uden fejl. **ProductDetail** blev implementeret for at vise detaljer om hvert produkt, og jeg tilføjede **ContactSeller**, så brugerne kan kommunikere direkte med sælgere.

Jeg havde udfordringer med **validering i SignUpPage**, især i forhold til at sikre automatisk login efter oprettelse. Jeg løste dette ved at tilføje en funktion, der sender brugeren direkte til **MinSide** efter succesfuld oprettelse.

Til sidst arbejdede jeg på **MineAnnoncer**, hvor brugere kan se og administrere deres egne annoncer. Her opstod der fejl i visningen af data, men jeg fik det løst ved at tilpasse API-kaldet og sikre korrekt håndtering af brugerens adgangstoken.

Redegørelse for oprindelsen af de forskellige kodeelementer i prøven

Størstedelen af koden i projektet er udviklet fra bunden med inspiration fra dokumentation og tidligere projekter.

- **Navigation og footer** er bygget med React Router og SCSS for en responsiv brugeroplevelse.
- **Produktvisning** (ProductList.jsx, KategoriListe.jsx) henter data fra API'et ved hjælp af en custom hook (useGet()).
- **Annoncehåndtering** (AnnoncePage.jsx) inkluderer validering og datahåndtering via API-kald.
- **Brugerhåndtering** (LoginPage.jsx, SignUpPage.jsx) blev kodet med validering og autentificering for at sikre en sikker login-proces.
- **Produktdetaljer og kontakt** (ProductDetail.jsx, ContactSeller.jsx) gør det muligt at se produktinfo og kontakte sælgere.
- **Brugerprofil** (MinSide.jsx, MineAnnoncer.jsx) giver brugere adgang til deres annoncer og profiloplysninger.

Projektet er testet og optimeret for at sikre en stabil funktionalitet.

Beskrivelse af særlige punkter til bedømmelse

Jeg har valgt at fremhæve ProductList-komponenten, da den viser min forståelse af React-hooks, API-kald og dynamisk rendering. Denne komponent er central for brugeroplevelsen og sikrer en interaktiv og brugervenlig visning af produkter.

Brug af useEffect og fetch

- **Data hentes dynamisk fra API'et, så produktlisten altid er opdateret.**
- **Produkterne vises i en ny rækkefølge hver gang, så brugeren ser et varieret udvalg ved hver visning.**

Dynamisk produktvisning

- **Produkterne vises i et responsivt grid, der tilpasser sig forskellige skærmstørrelser.**
- **Hvert produkt linker til en detaljeret visning via React Router (NavLink), hvilket skaber en intuitiv navigation.**

Fejlhåndtering og brugeroplevelse

- **Hvis der opstår en fejl ved datahentning, vises en fejlbesked direkte på skærmen.**
- **Fejlhåndtering sikrer, at layoutet forbliver intakt, selv hvis API'et ikke svarer.**

Hvorfor har jeg valgt denne komponent til fremlæggelsen?

- **Den viser mine færdigheder i React-hooks (useState, useEffect).**
- **Den demonstrerer, hvordan jeg håndterer API-kald og fejl.**
- **Den har et rent og struktureret UI, stylet med SCSS.**
- **Jeg kan forklare funktionaliteten og dens betydning for brugeroplevelsen.**

Her er planen for ugen:

Mandag	Tirsdag	Onsdag	Torsdag	Fredag
Opsætning af projekt, tilslutning af backend, planlægning af mappestruktur, udvikling af navigation og footer, grundlæggende routing.	Implementering af produktvisning i **ProductList.jsx**, udvikling af **KategoriListe** og **HeroSection**, start på rapportskrivning.	Udvikling af **AnnoncePage** med validering og API-håndtering, implementering af **Donation**, udvikling af **LoginPage** og test af login-systemet. Rapport skrivning	Implementering af **ProductDetail** og **ContactSeller**, debugging af beskedsystem og API-håndtering. Rapport skrivning	Udvikling af **MinSide.jsx** og **MineAnnoncer**, fejlfinding, optimering af **SignUpPage**, og Test, rettelser, afslutning af rapport