

Semesteropgave Legekrogen Rapport

Uge 33 & 34

WebH124-1

Mark Wolf
Emil Hansen
Gabriel Hughes

Vurdering af indsats

Vi føler, at vi har lagt en god mængde indsats i dette projekt ved at udføre de påkrævet krav og derefter gå ud over det for at tilføje ekstra funktioner. Vi har også lavet næsten alle valgfri opgaver, så alle sider hentes med API, er responsiv, og har en kurv til shoppen. Vi brugte også ekstra tid på at gøre koden dynamisk ved at tilføje funktioner som en layout.jsx og en app context. Der blev lagt meget arbejde i at sikre, at vores kode er ren og læsbar, selv hvis du skulle slutte dig til projektet uden at vide noget om det på forhånd.

Opsætning og struktur

Som den allerførste beslutning fastslog vi, at React var et passende værktøj til at bygge hjemmesiden til dette projekt. React giver os muligheden for lettere at kommunikere med API'et og opdatere frontenden af vores projekt live. Vite blev også brugt til dette projekt, da det er standard miljøet for React.

Da vi startede projektet, gik vi straks i gang med at bruge API'et til at hente produkterne. Resten af projektet har været baseret på denne beslutning. Derudover har vi lavet siderne responsivt, og de har en shoppingkurv som kan benyttes fra alle sider.

Vi har besluttet at strukturere vores projekt ved hjælp af en Layout-fil og en Context-fil for at gøre hjemmesiden mere dynamisk. Hvis man beslutter sig for at ikke bruge en Context-fil, er window.addeventlistener også en løsning. Nedenfor gives en begrundelse for hver af disse beslutninger.

Context-fil

For det første lavede vi AppContext-filen for at minimere antallet af gange, det var nødvendigt at hente data fra API'et. Provider-komponenten undgår også, at vi skal omskrive den samme kode igen og igen, i stedet pakker vi det ind i en genanvendelig komponent.

Som en ekstra funktion har vi tilføjet en indkøbskurv-funktion, så brugeren kan tilføje og fjerne varer fra deres kurv, før de fortsætter til betaling. AppContext-filen har også en stor indflydelse på vores evne til at indlæse kurven fra hver side uden behov for at parse data igen.

```
1  const { cart, handleCart } = useContext(AppContext);
2  const [cartItem, setCartItem] = useState({});
3
4  useEffect(() => {
5    setCartItem(cart.find((item) => item._id === product._id));
6  }, [cart, product]);
7
```

Dette er et eksempel på vores context-fil, der bruges med useContext-hooken for at hente kurven, og derefter sætte kurv item til det tilsvarende produkt.

Event Listener

Denne useEffect hook initialiserer 'Liked' arrayet, når komponenten monteres, og opsætter en lytter til den brugerdefinerede hændelse 'favoritesUpdated'. Hver gang denne hændelse udløses, kalder lytteren updateLikedArray() for at opdatere arrayet. Når komponenten afmonteres, fjernes event listeneren for at rydde op i ressourcerne. Det er nemlig et godt alternativ til Context-fil, som vil sørge for at komponenterne bliver opdateret "live".

```
1  useEffect(() => {
2    updateLikedArray();
3
4    const handleCustomEvent = () => {
5      updateLikedArray();
6    };
7
8    //listens for event to update liked array
9    window.addEventListener("favoritesUpdated", handleCustomEvent);
10
11   //removes event listener if event becomes unmounted
12   return () => {
13     window.removeEventListener("favoritesUpdated", handleCustomEvent);
14   };
15 }, []);
```

Layout

Vores Layout-fil giver en dynamisk skabelon, hvori en side genereres. De komponenter, der bruges på hver side, nemlig DeliveryMessage, Header, Hero og Footer, er pakket ind i layout-filen for at undgå unødvendig gentagelse af kode.

Children-parameteren er det indhold, der er unikt for hver side, og som derefter håndteres på selve siden.

```
1  function Layout({ children }) {
2    return (
3      <div className={styles.container}>
4        <div className={styles.top}>
5          <DeliveryMessage />
6          <Header />
7          <Hero />
8        </div>
9        {children}
10       <NewsletterContainer />
11       <Footer />
12     </div>
13   );
14 }
```

```
1  const routes = useRoutes([
2    {
3      path: "/",
4      element: (
5        <Layout>
6          <Home />
7        </Layout>
8      ),
9    },
10  ]),
```

Ekstra Features / Valgfri

- Kurv
- DeliveryMessage
- Carousel
- Responsivity
- API (get og post)
- React
- Context-fil
- Layout-fil