

GitHub-repositoriet: https://github.com/mrflexxi/eksamensdokument_gruppe20

Vanskelighetsgrad:

Gruppen har siktet mot C-kravet. All styling er ifølge skissene til C-kravet, men fikk ikke tid til Sanity.

GitHub-alias:

Arien Djazirehie: mrflexxi

Djuneid Gubadag: TheJkmega

Mina Melissa Martin: Minmar03

Oscar Kanstad Frisell: ApolloDesktop

Forutsetninger og redegjørelser:

Vi startet Pokedex-oppgaven sent på grunn av t gruppemedlemmene hadde forskjellige eksamensleveringer. Dette forsinket felles start av UIN-eksamensarbeidet.

Vi oppdaget at lenker til skisser var ødelagt, noe som førte til forvirring og tap av ytterligere arbeidstid.

Vi måtte kontakte Tore Marius for korrekt PDF, noe som resulterte i omarbeid av all styling. Oppgaven krevde at alt arbeid skulle følge skissene, som begrenset vår kreative frihet innen applikasjonsdesign og styling.

Vi forsto ikke tydelig hva "PokeCard" og "TypeCard" refererte til, og måtte kontakte veileder for avklaring.

Arbeidsmetodikk:

Vi hadde planleggingsmøter hvor vi fordelte oppgaver og satte tidsfrister. Deretter jobbet vi både individuelt og i gruppen før vi møttes igjen fysisk for å laste opp kodene i repositoriet.

Uførte oppgaver:

Main Pokemons: I tillegg til item.nr, bilde, og navn, har vi også lagt inn Type(s). Det går an å legge til enda flere informasjoner på Pokemonskortet. Kodingen blir det samme, og vi valgte å bruke kun disse fire elementene.

SearchResults.jsx: her sier oppgaven at brukeren skal taste hele navnet til en pokemon og da skal appen vise kortet til den søkte pokemon. Sensoren kan finne denne funksjonaliteten i filen Searchresults_org.jsx.

Vi mener søkefunksjonen blir mer brukervennlig, hvis den søker etter f.eks. de fire første bokstavene. For eksempel, hvis brukeren har glemt et navn og bare husker at pokemon-navnet starter med a, eller ab eller abo, da får han alle pokemon som starter med disse bokstavene. Lettere å finne fram pokemon.

Videre fant vi at hvis et pokemonsnavn skrives med blokkbokstaver eller starter med en blokkbokstav, kan ikke søkefunksjonen finne den. Her har vi gjort det slik at søke-ordet etter tasting, blir gjort om til «lowercase» for å sikre at søke er vellykket.

Pokemon.jsx: Det var ingen krav til klikkbarhet i denne siden, men vi syntes det er lurt å ha en direkte lenke til siden som viser pokemon med samme type som den valgte pokemon. Vi ser på skissen at det er et bra oppløsningsbilde som er blitt brukt, men vi fant ikke slike bilder i PokeAPI (Det kan hende at vi ikke lettet godt etter), isteden brukte vi det samme som vi har brukt ellers, og prøvde «width» of «Height» i css og økte størrelsen. Kvaliteten er ikke blitt bra på bildet.

Type.jsx: Denne komponenten skal vise alle pokemon med samme type. Her blir colors.css brukt til å bestemme bakgrunnsfargen. Vi tenkte hva med pokemon som har mer enn en type? Hvilken farge skal vi velge? I Type.jsx, valgte vi at pokemon som har flere typer, får med seg fargen til siste typen. Dette får å vise på en godt visuell måte at pokemonen har flere typer. I Pokemon.jsx som viser informasjon om en valgt pokemon, brukte vi fargen til den første typen selv om pokemonen hadde flere typer.

Sanity:

Vi ble hindret av usikkerhet rundt oppgaven og tap av tid på grunn av helligdager like før innlevering. Sanity var den eneste oppgaven som ble påbegynt, men vi rakk ikke å fullføre.

Kilder:

Generelt:

<https://lms.webtricks.blog/kurs/uin>

<https://react.dev/>

<https://fonts.googleapis.com/>

PropTypes:

React-pakken "prop-types" lar oss definere de forventede typene for props (egenskaper) som sendes til en komponent. Dette hjelper til med å oppdage feil og gi en klarere forståelse av hvilke data som forventes å bli mottatt av komponenten.

<https://www.npmjs.com/package/prop-types>

<https://www.freecodecamp.org/news/how-to-use-proptypes-in-react>