Kode

Stack:

Jeg har valgt å bruke Adonis på toppen av Nuxt/Vue. Dette er fordi jeg har god erfaring ved denne stacken når det kommer til skalerbarhet, og videre utvikling. Jeg forsøker å kode i stor grad modulært og abstrahert. Med det mener jeg at jeg lager få komponenter som kan anvendes mange steder.

Dette gjelder også i CSS hvor jeg har en semi-utility-first tilnærming, som vil si at visse ting jeg trenger om og om igjen, f.eks. styre vertikal luft så har jeg egne klasser jeg kan benytte som bare legger til margin-top. Ved så korte prosjekter som dette er, er jeg nødt til å justere ambisjonene for abstrahering etter tiden som er tilgjengelig. Derfor blir det en del «hyperspesifikk» som kun brukes en gang/ ett sted. Det er slik jeg ser på det ikke idéelt om prosjektet blir større med tiden for da må man potensielt endre koden mange steder for å få likt utrykk/utseende, men av og til har man ikke tid, eller kunden råd til å prioritere det.

Plugins jeg bruker:

* Axios for å hente postkodedataen, og i et reelt prosjekt ville jeg brukt det for å hente eventuelt data fra et API
* Vuex for å lagre valgene brukeren foretar seg underveis sentralt, slik lett å hente/ endre i hele applikasjonen/ komponentene
* Lodash for å søke igjennom js objektene, og finne ut om postnummeret som er lagt inn er gyldig og eventuelt om det er i Vestfold og Telemark. Grunnen til at jeg bruker en pakke for dette er for å være sikker på at jeg støtter eldre nettlesere som ikke har de nyere JS-funksjonene.
* What-input, også beskrevet i Design/UX delen, men denne er brukt for å skille mellom klikk-events og keyboard-events

NPM audit:

* Det er en pakke som er klassifisert som «high risk» men dette er en pakke som brukes i devDependencies, med andre ord den er ikke der på sidensom er lansert og jeg velger å ikke gjøre noe med den i første omgang.

Nettleserstøtte:

* Med tiden, og ressursene jeg har hatt til rådighet har jeg valgt å ikke prioritere å fikse feil på Internett Explorer. Det er mulig å gjennomføre prosessen i IE11 men knappene og tittelen på første komponent er forskyvet til siden.   
  I en reel situasjon ville jeg hatt en dialog med kunden om hvorvidt IE er en prioritering, og foreslått å ikke prioritere den i førsteomgang, men heller observere via Google Analytics hvor stor andel besøkende det er som bruker IE over en periode på ~1mnd og ta en ny dialog da.

**Postnummer som påkrevd**

Jeg legger til grunn at de fleste som eier hytte og som har strøm på hytta vet postnummeret sitt. Derfor velger jeg å bruke postnummertabell fra posten for å finne ut om postnummeret er i Vestfold eller Telemark. Om ikke stoppes de fra å gjennomføre prosessen fordi HytteNett ikke tilbyr produktene sine der enda.

Postnummer tabellen leveres av Posten/Bring og kan finnes her: <https://www.bring.no/postnummerregister-ansi.txt>  
Jeg har gjort om formatet fra .txt til JSON og hoster den på nettsiden. Jeg valgte å gjøre det på denne måten for å ikke måtte parse/ tolke txtfilen hver gang de endrer postnummeret fordi det er en tung prosess, og kan potensielt føre til dårlig brukeropplevelse.

**Norsk telefonnummer**

Jeg har valgt å sjekke om nummeret som blir lagt inn er et gyldig norsk nummer. Dette er for at folk ikke skal gjøre små feil i innskjekkingen og får i minimere dårlige data. Basert på tall fra SSB er 97.5% av alle eiere av hytter i Norge fra Norge. I en reel situasjon ville jeg tatt en dialog med oppdragsgiver om denne problemstillingen, hvor vi kunne sett på vi skulle åpnet for eksempel svenske telefonnummere som har den nest størte eier andelen.

Kilder