

Problemstilling:

Etter vi leste oppgaven var første tanken at vi skulle lage et produkt lignende Ebay, bare Norsk og for antikviteter eller andre eldre gjenstander. Vi startet med å lage sketsjer for å få et overblikk på hva som måtte gjøres og systemer vi trengte.

Under dette punktet går vi inn i dybden på disse og har delt opp de større områdene

Frontend:

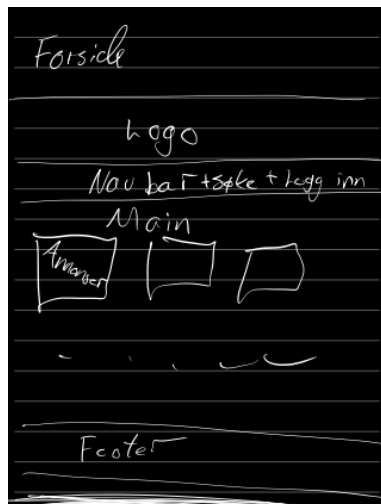
Vi endte opp med å bruke HTML og Next.js til å lage prosjektet. Basert på sketsjene som ble laget skapte vi et basic skjelett av hva vi trengte å ha tilgjengelig, HTML som skulle brukes og sider som vi trengte for å nå vårt mål. Hovedsiden viser alle tilgjengelige annonser, samt er det en navigasjons bar hvor en bruker har muligheten til å logge seg inn eller søke etter produkter.

Nettsiden vil også endre seg gradvis basert på om en bruker er logget inn eller ikke, om brukeren er logget inn vil de få et valg om å kunne lage en annonse.

CSS var ikke hovedfokuset vårt men vi endte opp med å ha et relativt enkelt men forståelig design og layout.

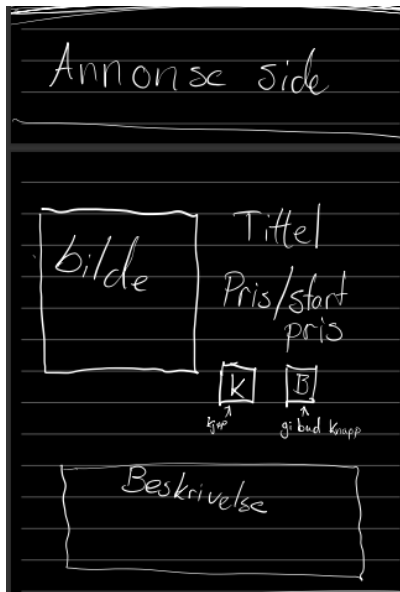
Sketsjer som vi lagde for å lage nettsiden

Forsiden



Her tenkte vi en simpel siden med Logo på toppen, navigasjonsbar under og en main som viser alle annonsene.

En singel annonse side

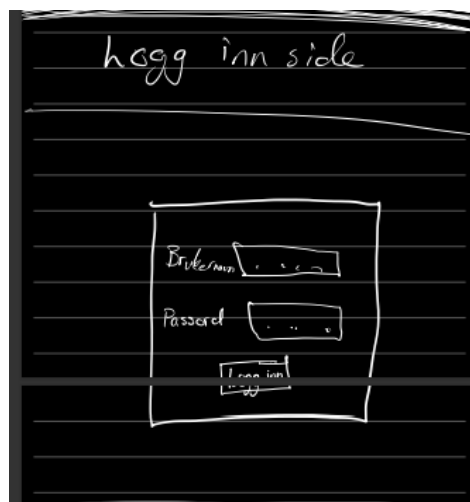


Denne siden tenkte vi når man har gått inn på en annonse som er på forsiden. Da skal vi ha mulighet til å kjøpe/ gi bud på varen. Så skal den ha en beskrivelse på bunnen. Ganske simplisert ebay side.

Resten er for brukeren/dashboard

1. Logg inn side

- Her har vi et felt for brukernavn og passord og en knapp for å logge inn til dashboardet.



b.

2. Min side

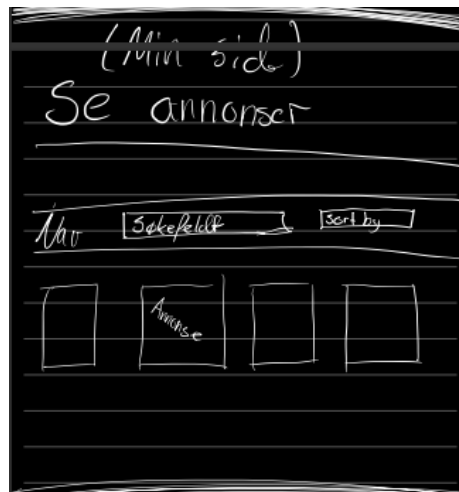
- Her tenkte vi bare å ha noe simpelt for prototypen. Her har vi to knapper, der ene er for å lage annonser og en er for å se alle brukerens annonser.



b.

3. Se annonser (Se brukerens sine annonse)

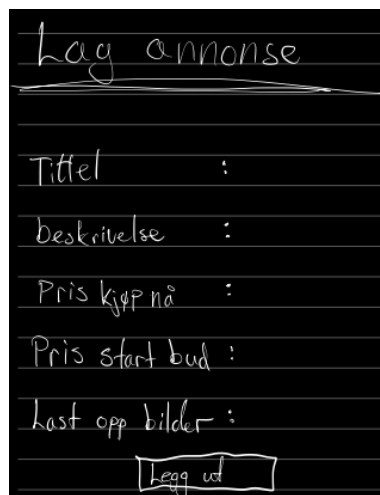
- a. Tenkte det var fint å ha en side som brukeren kunne se alle sine egne annonser.



b.

4. Lag annonse

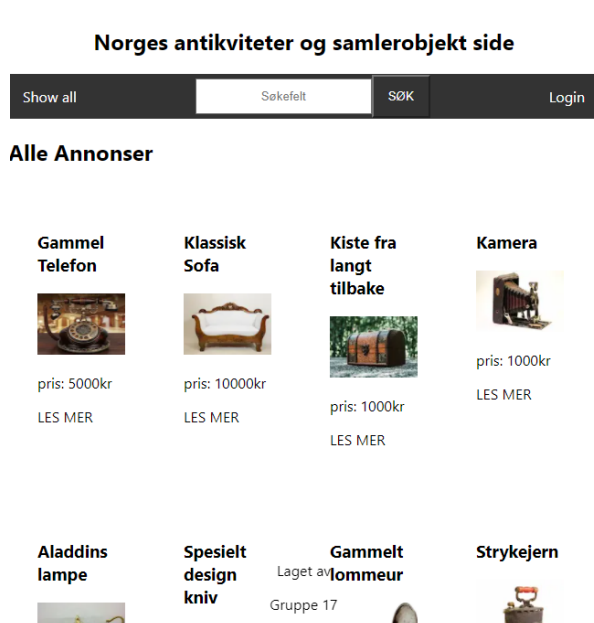
- a. Her tenkte vi at det var fint å ha en mulighet for å legge ut en annonse.



b.

Hvordan prototypen endte med å se ut. (Settes i samme rekkefølge som sketsj)

Forsiden



Forsiden ble veldig lik som vi tenkte i sketsjen vi hadde. Siden har en topp tittel som kunne blitt lett byttet til en logo om det firmaet har det. Har også en ganske simpel navigasjonsbar med Show all, søkefelt med knapp og en Logg inn knapp. Og tilslutt hovedinnholdet med at forsiden viser alle annonser som er i databasen på nettsiden.

En singel annonse

<input type="text" value="Søkefelt"/>	<input type="button" value="SØK"/>
---------------------------------------	------------------------------------

Gammel Telefon



Kjøp nå pris: 5000

Bud pris: 100

----- Beskrivelse -----

oasfnfoiasnfoiasnfasoind

Denne siden er nesten lik som sketsjen, Tittel er over og ikke på høyre side for bildet. Og priser er under istedenfor til høyre. Den har to knapper en for kjøp varen nå og en for å gi bud. En random bruker får ikke trykket på siden om ikke den brukeren er innlogget. Om brukeren er innlogget så om du trykker på gi bud så kommer det opp et input felt og en knapp som du bruker til å gi bud på varen.

Resten er for brukeren/dashboard

5. Logg inn side

- Da har vi fått laget en login side med brukernavn og passord. La også inn test info slik at sensor kan få logget seg inn på en av brukerne.

Norges antikviteter og samlerobjekt side

<input type="text" value="Søkefelt"/>	<input type="button" value="SØK"/>
---------------------------------------	------------------------------------

Brukernavn

Passord

Test verdier

Brukernavn: admin

Passord: admin

Eller

Brukernavn: gruppe17

Passord: gruppe17

-

6. Dashboard

- a. Siden har en knapp til å gå til mine annonser, lage annonse og logg ut om brukeren vil det.

Norges antikviteter og samlerobjekt side

[Show all](#)

[SØK](#)

[admin](#)

Dashboard siden

[Mine Annonser](#)

[Lag annonse](#)

[Logg ut](#)

b.





7. Mine annonser (Se brukerens sine annonse)

- a. Her kan man se alle annonsene til brukeren.

[Show all](#)

[SØK](#)

[admin](#)

<p>Tittel: Gammel Telefon</p> <p>Beskrivelse: oasfnfoasfnfoasfnfoasind</p> <p>Kjepspris: 5000kr</p> <p>StartBud: 100kr</p> <p>Bilde</p> 	<p>Tittel: Klassisk Sofa</p> <p>Beskrivelse: oasfnfoasfnfoasfnfoasind</p> <p>Kjepspris: 10000kr</p> <p>StartBud: 100kr</p> <p>Bilde</p> 	<p>Tittel: Kiste fra langt tilbake</p> <p>Beskrivelse: oasfnfoasfnfoasfnfoasind</p> <p>Kjepspris: 10000kr</p> <p>StartBud: 100kr</p> <p>Bilde</p> 	<p>Tittel: Kamera</p> <p>Beskrivelse: oasfnfoasfnfoasfnfoasind</p> <p>Kjepspris: 10000kr</p> <p>StartBud: 100kr</p> <p>Bilde</p> 
---	---	---	--

b.

8. Lag annonse

- a. Her lager vi en annonse og så deretter gå den til mine annonser siden

[SØK](#)

Tittel:

Beskrivelse:

Pris, kjøp nå:

Pris, start bud:

[Legg ut](#)

b.

Backend:

Basert på hva vi trengte endte vi opp med å bruke Node.js libraries for å oppnå ønsket funksjoner, dette gjør det lettere å skape en dynamisk side når en bruker trykker inn på en annonse.

Prosjektet bruker en js fil med JSON format til å holde på test informasjon og programmet tillater midlertidig lagring av nye annonser, men disse blir fjernet når local serveren stopper eller at blir refreshet. Dette er gjort fordi det skal vise hvordan annonse systemet funker, samt hvordan det å lage en annonse skal fungere, men vi har ikke direkte skrijving til js filen med data siden det er bedre å knytte opp en ekte database til å holde på informasjonen. Med andre ord, dette er en tredjeparts jobb for å sette opp og eventuelt koble til en database.

Et annet punkt som også bør håndteres av et tredjeparts element er login/registrering av en bruker, vi har eksempel på hvordan dette skal se ut men er ikke et ekte autorisering system, men tillater en bruker å lage annonser så lenge de har logget inn på nettsiden. Disse blir midlertidig lagret som nevnt over.

Det er delvis satt opp for håndtering av kjøp eller bud på et produkt, men er igjen en tredjeparts oppgave å linke opp riktig med betalings metoder

Arbeidsmetode:

Originalt skulle vi bruke Agile som utviklings model, men vi endte opp med å møte ofte på skolen og arbeide med prosjektet sammen, slik at vi kunne ha en semi agile utvikling.

Testing:

Produktet har relativt få funksjoner, men hver funksjon er ganske utdypet og/eller henger sammen med andre kritiske funksjoner(For eks. Å lage en annonse, må en bruker være logget inn). Vi laget derfor tester for de fleste funksjoner og dems funksjonaliteter

Testene er delt opp i flere filer, men mest av testene ligger i filen "navigationTest".

Testene går i hovedsak ut på om navigasjonen på sidene fungerer som forventet, enkelte går ut på å bytte mellom et par sider, mens andre tester om en bruker som er logget inn fortsatt får tilgang til det som er forventet, blant annet sine annonser.

Det testes også at en spesifikk annonse blir lastet inn.

"buyAndBidTest" går ut på å sjekke kjøp funksjonen vår, samt ulike måter en kan oppnå dette, det testes både kjøp og bud funksjoner. Det testes når brukeren er logget inn,

siden det er allerede elementer i prosjektet som hindrer at en bruker får tilgang til å kjøp eller gi bud med mindre de er logget inn. Og den tester vis en bruker som ikke er logget inn prøver å trykke på kjøp eller gi bud knappen.

“createAnnonseTest” inneholder kun en test, men en relativt lang test. Dette tester om en bruker som er logget inn har mulighet til å logge seg inn og navigere så brukeren kan lage sin egen annonse, så fyller den inn test data og deretter klikker på legg ut knappen. Det skal også sjekkes at annonsen som er laget faktisk blir laget.

Dependencies:

Vi bruker Node.js til å lage nettsiden, vi har valgt dette fordi vi har brukt React og Next.js i tidligere prosjekter. React og Next.js er også utrolig fine verktøy å bruke for større applikasjoner, og vil gjøre det enklere for videreutvikling av de som kunden ønsker skal utvikle.

Vi bruker også Playwright, dette er et library innenfor Node.js som lar oss kjøre tester på prosjektet, samt tester den at funksjoner vi lager kan brukes på andre nettlesere

Guide for å starte nettsiden.

1. **Node package manager(npm)** trenger å bli lastet ned og installert. Siden dette er et NEXT.js prosjekt.
 - a. <https://nodejs.org/en/download/>
 - b. Brukte node versjon : v16.7.0 og node package manager 7.20.3
2. Trenger eks **Visual Studio Code** eller et annet **IDE** bare det har et terminal som man kan skrive kommandoer i. (Viser her for VS Code)
3. Åpne mappen **webpage** i terminalen slik at du kan skrive kommandoer i den mappen. Lignende slik :

```
\webpage> |
```

4. Siden npm er installert så skal du kunne skrive: **npm install** i terminalen så skal den installere alt av dependencies som trengs til Next.js.

- a.

```
\webpage> npm install
```

5. Etter at alt er installert så kan du starte nettsiden ved å kjøre kommandoen: **npm run dev.**
6. Når siden er oppe å går så skal det stå slik

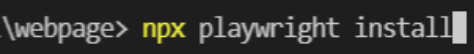

```
ready - started server on 0.0.0.0:3000, url: http://localhost:3000
```

Kan være ctrl + click på din datamaskin.

Eller gå inn på linken <http://localhost:3000> når siden har begynt å kjøre.

Testene

For å kjøre testene så må **Guide for å starte nettsiden** være oppe å går.

1. Disse testene bruker playwright til å teste funksjoner på nettsiden. Så i terminalen i VS Code pass på at den er samme nivå som nettsiden.
 - a. Her skriv i terminalen: **npx playwright install** så installerer den det som trengs for at det skal være mulig å kjøre testene.
 - b. 
2. Etter den har installert alt som trengs for å kjøre testene.
 - a. Start opp nettsiden igjen med **npm run dev**.
 - b. I et nytt terminal vindu åpent på samme nivå. Kjør kommandoen i terminalen: **npm run test:e2e**
 - c. 

Da skal testene være oppe å går. Må kjøre **npm run test:e2e** hver gang om testen skal kjøres flere ganger

System krav:

API og Hooks (L): Et ganske stort område, siden dette vil håndtere samarbeidet mellom databasen og hvordan selve prosjektet vil få inn informasjon. Blant annet påvirker dette alle de andre delene av prosjektet.

Forside(M): Et område hvor brukeren kan se alle tilgjengelige annonser og få tilgang til å logge seg inn, eventuelt kunne se på individuelle annonser.

Login system(L): Like kritisk som API/Hooks, en bruker vil ikke kunne kjøpe eller gi bud på produkter som er en av hovedfunksjonene til prosjektet.

Navigerings system(M): Dette vil ikke ta mye tid å sette opp, men er en viktig del som tillater brukere å logge inn, søke etter produkter og navigere rundt på prosjektet

Kjøp av produkt(S): Dette er en viktig funksjon som nevnt tidligere, men er ikke et system vi direkte skal håndtere, men bare legge inn enkle funksjoner som kan utvides ved videreutvikling.

Utfordringer/Refleksjon:

Vår første utfordring var å finne et sted å starte prosjektet, det var lite input fra resten av gruppen før folk forlot gruppen. Vi hadde planlagt en mer Agile fokusert utvikling av programmet, men sent inn i prosjektet fikk vi vite at en av vår gruppe medlemmer hadde blitt “kastet ut av faget”, og kunne derfor ikke fortsette med prosjektet. Vi mistet også kommunikasjon med et annet gruppemedlem ikke lenge etter dette.

Vi støtte på et lite problem med memory leak. Det skapte problemer med navigasjon og innlasting av sider. Men det var ikke en stor problem å fikse siden det måtte bare fjernes et ord i “useAllAnnonser”-hooken.

Nærmere slutten av prosjektet forsvant enda et gruppemedlem og vi fikk ikke tak i denne personen før en dag før leveringsfrist av prosjektet, medlemmet meldte seg også ut av gruppen etter vi fikk snakket med dem