|  |  |  |
| --- | --- | --- |
| Utviklere | Studentnummer | E-post |
| Alexander Pettersen | 139953 | [alex1990pettersen@gmail.com](mailto:ALEX1990PETTERSEN@GMAIL.COM) |
| Erik Hellerslien | 139964 | [erik\_hellerslien@hotmail.com](mailto:ERIK_HELLERSLIEN@HOTMAIL.COM) |
| Erlend Andreas Hall | 139960 | [erlendahall@gmail.com](mailto:ERLENDAHALL@GMAIL.COM) |
| Sindre Østby | 139936 | [sindreoestby1@gmail.com](mailto:SINDREOESTBY1@GMAIL.COM) |



Innholdsfortegnelse

[Generering av HTML-sider 2](#_Toc481932667)

[Plass bruk 3](#_Toc481932668)

[Henvendelser 3](#_Toc481932669)

[Bilder 4](#_Toc481932670)

[Opplasting 4](#_Toc481932671)

[Søk 5](#_Toc481932672)

[Forhåndsvisning 5](#_Toc481932673)

[Linking av bilder 6](#_Toc481932674)

[Andre problemer/mangler 6](#_Toc481932675)

[Glemt passord 7](#_Toc481932676)

[Meny og artikler 7](#_Toc481932677)

[Nye artikler 7](#_Toc481932678)

[Sletting av filer 7](#_Toc481932679)

[Utfordringer 7](#_Toc481932680)

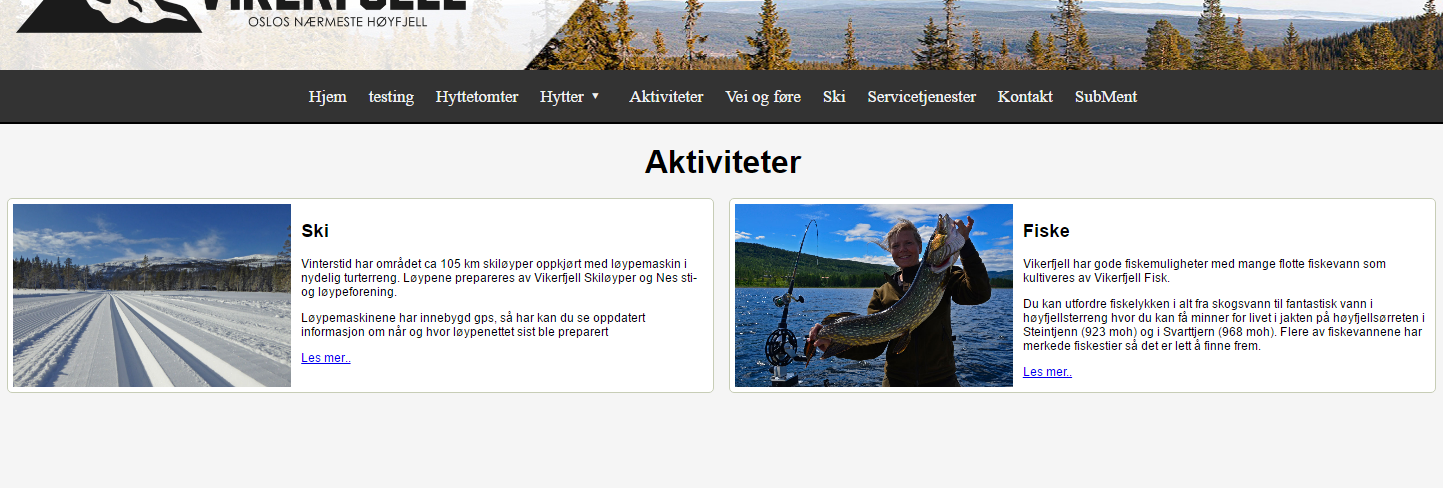
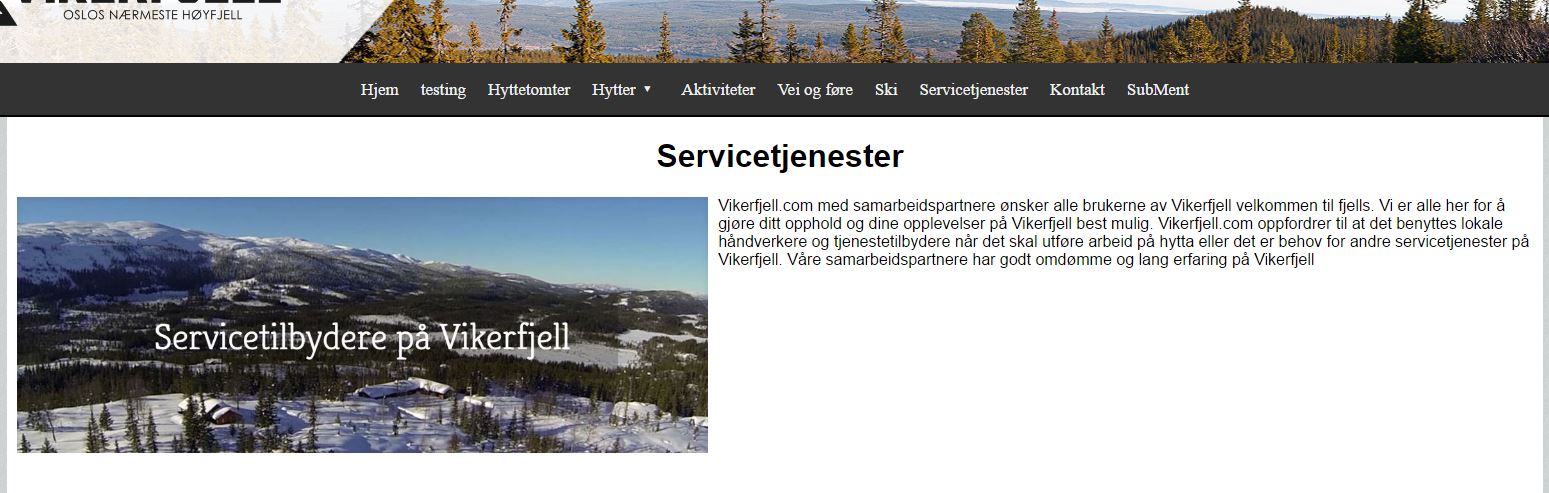
# Generering av HTML-sider

Når vi genererer sider har vi valgt at filnavnet blir det brukeren skriver inn som overskrift, dette er for å gjøre det enklere for brukeren slik at det ikke er nødvendig å tenke på filnavn. Ved generering av filer blir det kjørt en sjekk, som sjekker om filnavnet finnes fra før. Hvis navnet finnes fra før vil det bli lagt til et «1»-tall etter det som blir sendt til databasen(overskriften), og denne kjører helt til den ikke finner noe. Sjekken blir utført for å slippe at filer overskriver hverandre under generering.

I følge oppgaven skulle nettstedet bestå av statiske html-sider, det skal ikke være nødvendig for nettstedet å hente informasjon direkte fra databasen i front-end. For å løse dette har vi valgt å generere html-sidene ved hjelp av PHP, hvor det blir generert nye sider ved endring i back-end. For å minimalisere antall genererte sider ved en oppdatering, satte vi opp de ulike situasjonene som kunne oppstå i back-end og så på hvilke sider som måtte oppdateres ved endring. Gruppen kom frem til to mulige situasjoner:

* **Ved en oppdatering av innhold:** Oppdatere «oversikten» til det meny-elementet som innholdet tilhører og skrive/endre/slette fil.
* **Ved en endring i meny:** Alle sider må oppdateres for å legge in det nye meny-elementet.

Ved å se på de ulike situasjonene som kunne oppstå, kom vi frem til at det ikke var nødvendig å oppdatere alle filene ved en endring av innhold. Det gjør at bruker slipper å vente på generering av unødvendige filer. Ved å gjennomføre denne typen av oppdatering for de ulike situasjonene, visste vi at det ville oppstå flere utfordringer. Vi kunne løst dette på en enklere måte, ved å generere alle filer på nytt i begge situasjoner. For å optimalisere siden ønsket vi å utføre oppdateringen som krever minst generering.

Videre hadde vi også noen kriterier fra oppdragsgiver når det kom til fremvisning av innhold. Det skulle vises en oversikt over innholdet som var under det valgte meny-elementet. Hvis det kun finnes ett innhold koblet til ett meny-elementet, skal vi frem det ene innholdet med en gang.

Figur 2: Visning av kun ett innhold. (Antall innhold = 1)

Figur 1: Visning av en oversikt (Antall innhold > 1)

# Plass bruk

Ved bruk av plass på siden så har vi tenkt å ta hensyn til brukeren av siden istedenfor å gjøre det lett å programmere. Når vi nevner lett å programmere så mener vi at brukeren ikke har noe valg av hvor mye av siden som er i bruk til enhver tid, så f.eks når en side er laget til tablett så vil vi ofte se at siden har mye ubrukt plass. Vi har løst dette ved å tilpasse plassbruken på forskjellige skjermstørrelser, dette fører til at brukeren kan bestemme selv hvor stort vinduet kan være men til enhver tid se hele bredden på siden uten at det skal være mye ubrukt plass.

# Henvendelser

Oppdragsgiver skal kunne motta henvendelser på e-post gjennom en side på nettstedet. Her har vi valgt å ha en link på «Kontakt» som blir funnet nede i footeren, som videresender brukeren til en ny side kalt «henvendelser.php». Foreløpig har vi valgt å gjøre det enkelt, så istedenfor å ha alle meny-elementene vist på denne siden har vi bare en tilbake knapp som fører brukeren tilbake til «default.php». Dette er ikke endelig, og vi vurderer at denne også skal bli generert hver gang noe blir endret i menyen slik at den blir lik resten av sidene. Dette vil også gjøre denne siden til «.html» istedenfor «.php». Her vil brukeren få kunne skrive inn sin epost adresse, emne og kommentar som vil bli sendt til vikerfjell sin epost. Vi har laget en epost for testing så du kan se at epostene blir sendt.  
E-Post: [Vikerfjell2017@gmail.com](mailto:Vikerfjell2017@gmail.com)  
Passord: VikerfjellGR52017

# Bilder

Backend har nå funksjonalitet for å både laste opp bilder og linke dem til et innhold på frontend.

## Opplasting

*Bruker angir hvilket bilde som skal lastes opp sammen med en tittel/beskrivende tekst.* Dagens løsning støtter bilder i formatet png, gif og jpg.   
*Vi støtter foreløpig ikke jpeg.* PHP sine metoder skiller sømløst mellom .jpg og .jpeg. For å få støtte for .jpeg, må vi ha en større omskrivning av koden som lagrer filene på webserveren. Vår kode gjør at .jpeg-filer blir lagret som .jpg på serveren, mens databasen fremdeles får .jpeg som extension.

Når bruker trykker knappen «Last opp» skjer følgende:

* Bildefilen blir overført til «tmp» mappen på serveren.
* En thumbnail/minitatyrutgave generes og overføres til «Bilder/thumbs»
* Metadata (data om bildet) skrives til databasen
* Til slutt flyttes det opplastede bildet til «Bilder» mappen.

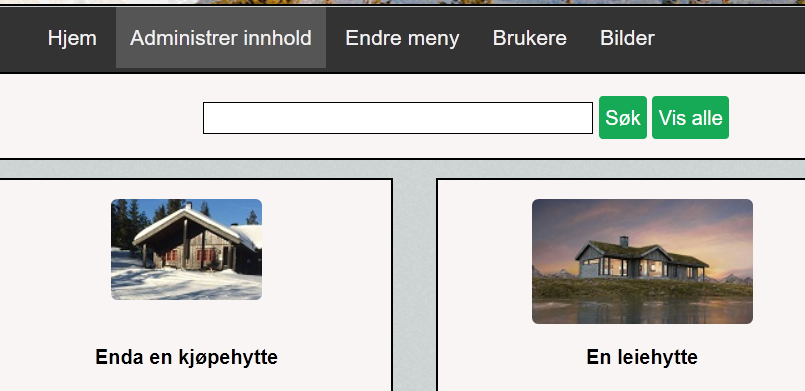
Vi har designet bildesiden slik at brukeren ikke trenger å bry seg om selve filnavnet. Dvs at brukeren kan fore serveren med duplikate filnavn. På grunn av en begrenset kolonnelengde (45) på «hvor»-kolonnen, tar vi en md5-hash av filnavnet for å sikre at filnavnet får for mange tegn. Om hashen allerede finnes på webserveren, tar applikasjonen en foranstilt salt på filnavnet og kjører ny hash på filnavnet.

En utfordring er å finne en passende dimensjon for thumbnail. I dagens løsning skiller vi mellom bilder som er i portrett, landskap eller kvadaratisk. Vi bruker følgende dimensjoner:  
  
Portrett: *100x150*  
Landskap: *150x100*Kvadratisk*: 100x100*

Med disse dimensjonene har vi en grei dekning på ulike bildedimensjoner. Vi skiller bare mellom portrett, landskap og kvadratisk, noe som fører til at bildene mister sitt sideforhold. Dette er ikke et veldig stort problem for miniatyrbilder, men f.eks et landskapsbilde med 21:4 forholdet vil bli veldig skviset, mens et 16:9 bilde ser helt OK ut.

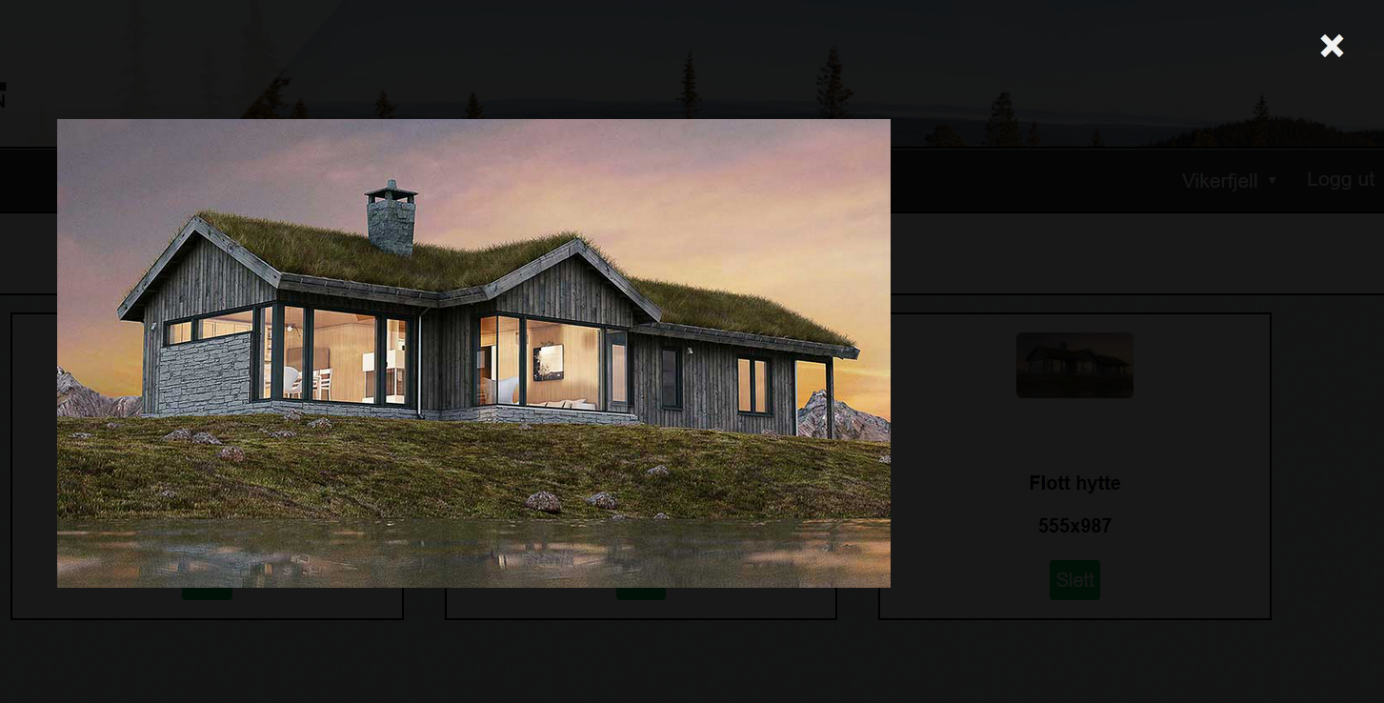
Vi skal gjøre et forsøk på å skalere bilder med sideforholdet som utgangspunkt til neste levering.

## Søk



Bildesiden har en enkel søkebar for å søke etter nøkkelord i bildeteksten. Knappen «Vis alle» viser alle bildene, men rent designmessig er det nok bedre å ha denne knappen et annet sted.

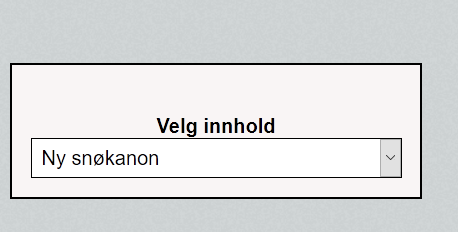
## Forhåndsvisning



Om ønsket kan bruker klikke på en miniatyrutgave for å se den tilsvarende utgaven i full størrelse.

## Linking av bilder





For å inkludere et bilde i innhold, trenger brukeren å angi innholdet fra nedtrekksmenyen og trykke på knappen «Inkluder i innhold»  
For å fjerne et bilde fra innhold, trenger bruker å kun trykke på knappen «Fjern fra innhold». Dette er fordi et spesifikt bilde kun kan opptre i ett innhold, og ikke flere.

## 

## Andre problemer/mangler

**Tooltips**  
Dagens løsning har ikke mulighet for å angi en tooltip (tekst som dukker opp når muser holdes over bildet). En tooltip kan bestå av opp til 100 tegn, noe som gjør at vi trenger en multilinjet tekstboks i grensesnittet. Før vi kan støtte tooltips, må vi ha en større rejustering av grensesnittet for å få plass til et såpass stort element.

**Endring av bildetittel**I dagens løsning har ikke bruker mulighet å endre bildetittelen på grunn av tidsbegrensninger. Vi vil gjerne ha dette på plass til siste delinnlevering.  
  
**Første opplastede bilde**Første opplastede bilde har en miniatyrutgave, men det kommer en udefinert referanse til fullskala-versjonen. Vi er usikre på hvorfor dette kun skjer med det første opplastede bildet, men vi skal kunne få dette fikset senere siden vi enkelt kan reprodusere problemet.  
  
**«Linkmodus»-knappen**Når bruker klikker på knappen «Linkmodus», er det ikke mulig å linke bilder før brukeren har valgt et innhold fra nedtrekksmenyen. Ved å klikke på knappen «Linkmodus» er det meningen at brukeren skal kunne begynne å linke til det første innholdet i databasen, men av en eller annen grunn må brukeren gå via nedtrekksmenyen.

# Glemt passord

Når vi skulle løse problemet med sending av glemt passord så er det mye vi må tenke på, som måten vi sender passordet på og sikkerheten rundt sending av passordet.

Valgene vi sto ved var å sende et engangs passord eller å sende en lenke som fører deg til en side hvor du kan endre passordet ditt. Vi valgte etter å ha diskutert hva vi tror vil være mest brukervennlig og samtidig ha best sikkerhet der valget ble å sende en lenke for å endre passordet. Ved å velge dette så var tanken at brukeren vil kun trenge å trykke på lenken og skrive inn passord h\*n ønsker istedenfor å skrive inn et engangspassord som kan være vanskelig å huske. Med tanke på sikkerhet så har vi tenkt at lenken må ha en tid som sier når lenken ikke kan brukes mer, i tillegg til dette har vi tenkt at vi må kunne verifisere brukeren på en måte her har vi tenkt at vi enten kan sende salt og kryptering av mailen til brukeren med i linken så vi igjen kan sammenligne mot mailen som ligger i databasen. En annen løsning på verifisering kan være å sende en sms til brukeren med en pinkode så brukeren skriver inn denne ved endring av passordet. Når det kommer til å implementere dette i løsningen vår så har vi funnet ut at vi ikke vil ha nok tid. Så i løsningen vår så har vi valgt å se bort ifra sikkerheten. Ved å gjøre det så vil «glemt passord» funksjonen fungere men ikke være en sikker løsning. For å kunne teste dette så må prosjektet lastes opp til f.eks filezilla eller lignende, og man må bytte ut linken som blir sendt. Dette er linken som blir sendt nå og det som trenger å bli endet er det som er markert i rødt:  
«$message = "https://home.hbv.no/139953/PHP/EnderPassordframail.php?id=$idbruker";» denne ligger i filen glemtpassordsend.php som ligger i katalogen include.

# Meny og artikler

## Nye artikler

Når vi lager innhold blir disse koblet opp mot det menyelementet brukeren velger. Her vil artiklene bli fylt opp med innholdet fra databasen. Bredden er satt til 80% på større skjermer, som endrer seg til 100% når skjermen blir litt mindre. Grunnen til at vi har gjort det slik er fordi det ikke ser naturlig ut å ha full bredde på store skjermer. Så om brukeren vil ha full bredde kan de gjøre nettleseren mindre. Det vi ikke har helt fått til enda er å ha artikler på flere submenyer. Grunnen til dette er at datamodellen ikke var helt optimal for bruk av submenyer når vi jobbet med dette. For denne innleveringen har vi da satt det resterende av menyer og submenyer på vent, for å så se hva vi velger å gjøre når den endelige innleveringen skal gjøres. Det som nå fungerer er å legge til, endre og slette artikler på hovedmenyer.

## Sletting av filer

Når brukeren sletter en artikkel eller meny, vil disse bli slettet både fra databasen og mappen utifra valgt element. Sånn som denne slett-funksjonen er satt opp nå, vil applikasjonen ikke kunne bli lastet opp til hvilken som helst katalog og virke uten endringer. Her er «unlink» satt opp til en spesifikk lokasjon.

# Utfordringer

Når vi startet prosjektet så valgte vi å dele alle filer via dropbox noe som fungerte greit men føltes som veldig tungvint siden alle måtte hente endringer for å kunne test og jobbe mot en lokal versjon. Derfor så valge til å gå over til Github som skulle være en lett måte å dele endringer. Dette fant vi ut at var en større utfordring enn hva vi hadde sett for oss. Vi måtte først bruke tid på å installere og siden vi ikke brukte samme «text-editor» (Atom, Visual studio og Sublime text3) så var ikke Github grensesnittet likt. Dette førte til at vi ikke kunne hjelpe hverandre og måtte bruke mye tid på å finne feil. det største problemet vi fikk var at vi mistet endringer som vi hadde gjort når andre oppdaterte hva de hadde gjort, dette førte til mange tapte arbeidstimer og at atmosfæren i gruppa ble dårligere da vi måtte løse problemer som egentlig var løst. Det vi lærte av dette var å ha backups på prosjektet til enhver tid, at vi må planlegge bedre hvilken programmer vi vil bruke i prosjektet og for å bruke Github så kreves det mer erfaring enn hva vi så for oss.