**GOTC – Gruppe 1**

**Endelig rapport, INF268**

**Applikasjonsutvikling for Internett**

|  |  |  |
| --- | --- | --- |
| Navn | Studentnr. | Epost |
| Ole Martin Helgesen | 139920 | [ohelgesen39@gmail.com](mailto:ohelgesen39@gmail.com) |
| Gabriel Aunan | 139948 | [Gabriel.aunan@hotmail.com](mailto:Gabriel.aunan@hotmail.com) |
| Christian Mol | 139927 | [Chris.mol@live.com](mailto:Chris.mol@live.com) |
| Tobias Pedersen | 139937 | [tobiasped@hotmail.com](mailto:tobiasped@hotmail.com) |

Denne applikasjonen for Vikerfjell har to hovedområder. Den første er front-end, som er utviklet for turister og andre interesserte i området. Den andre er back-end for administrator. I denne applikasjonen lages sidene dynamisk, hvor administrator har full kontroll over redigering av front-end fra back-end. Disse endringene skjer direkte i applikasjonen, slik at administrator ikke skal kunne trenge å skrive html, css, php og lignende.

Startfil for følgende

Frontend: localhost/VisitVikerfjell/Default.php

Backend: localhost/VisitVikerfjell/php/admin.php

Innhold

[**Innledning** 3](#_Toc484768730)

[**Hva har vi gjort** 4](#_Toc484768731)

[**Design og utforming** 4](#_Toc484768732)

[Krav til design 6](#_Toc484768733)

[Logo 6](#_Toc484768734)

[**Logg inn** 7](#_Toc484768735)

[**Back-end** 8](#_Toc484768736)

[Meny 8](#_Toc484768737)

[Sider 9](#_Toc484768738)

[Brukere 10](#_Toc484768739)

[Bilder 11](#_Toc484768740)

[Generering av html 12](#_Toc484768741)

[Brukervennlighet i backend 12](#_Toc484768742)

[**Front-end** 15](#_Toc484768743)

[Navigasjon 15](#_Toc484768744)

[**Autentisering I applikasjonen** 16](#_Toc484768745)

[**Konklusjon** 18](#_Toc484768746)

[Selve oppgaven 18](#_Toc484768747)

[Hva ville vi gjort annerledes 18](#_Toc484768748)

[Forbedringspotensialet 19](#_Toc484768749)

[Ferdigstilling 20](#_Toc484768750)

[**Kilder:** 21](#_Toc484768751)

## Innledning

Denne rapporten omhandler den endelige innleveringen i utviklingsprosjektet i Applikasjonsutvikling for Internett, INF268. Her har vårt oppdrag vært å utvikle et nettsted for Visit Vikerfjell, hvor formålet er å fremme området Vikerfjell, samt aktiviteter og lignende. Dette oppdraget fikk vi på bakgrunn av at de tidligere nettstedene til Visit Vikerfjell var utdatert. Nettstedet skulle bestå av en front-end, som er åpen for publikum, og en back-end, hvor eier har de funksjoner som er nødvendig for vedlikehold og oppdateringer i applikasjonen.

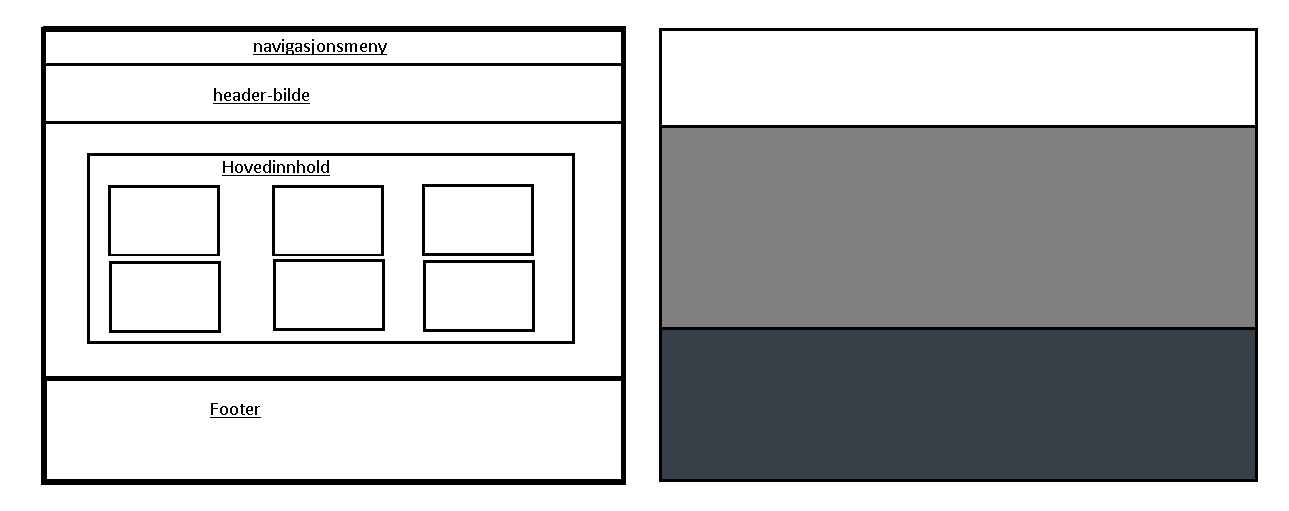
Tidsperspektivet til oppgaven har vært ett semester, hvor oppgaven var delt opp i fire deler med forskjellige innleveringsdatoer. Etter hver innlevering fikk vi tilbakemelding med eventuelle ting som burde forbedres.

## Hva har vi gjort

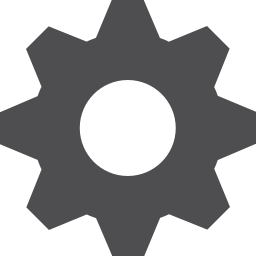
Utviklingsprosjektet bestod av fire deler, hvor sluttproduktet ble et ferdigstilt nettsted for Visit Vikerfjell. Under forarbeidet bestemte vi oss for å følge en utviklingsmetode som var smidig. Det mest naturlige var da en inkrementell utviklingsmetode, siden vi skulle utvikle 4 deloppgaver, samt tilleggsfunksjoner som vi mente kunne være nyttig å ha i en slik applikasjon som Visit Vikerfjell skal være.

## Design og utforming

Det vi startet med var å designe nettstedet, eller mer bestemt å sette opp en mal for sidene i front-end. Etter vi skisset ned hvordan siden skulle se ut, startet vi med html og css, hvor vi laget en menylinje, header-bilde, hovedinnhold og en footer.

  
*Bilde 1: Rutenett og fargespekter som er brukt i applikasjonen*

Med tanke på design av applikasjonen har vi tatt i betraktning de 7 prinsippene for universell utforming som ligger til grunn for at en plattform skal være brukervennlig. Der har vi tenkt mye på skjermstørrelser, fargevalg, skriftstørrelser, skrifttyper og andre ting som er relevant for brukervennligheten.   
For å gjøre det enkelt og intuitivt for alle brukere har vi benyttet oss av flere verktøy som lar oss teste størrelser og lignende opp mot landskap og portrett til ulike android telefoner, tablets, ipader og iphoner, samt pc. Dette for å tilpasse bruken ut ifra størrelsen på enhetene som blir brukt. Vi har også valgt å holde oss til konvensjonen om kun vertikal rulling, og ikke både horisontal og vertikal rulling, som bryter med reglene.  
  
Når det gjelder fargevalg på nettstedet har vi benyttet oss flere nyanser av grå, der spekteret går fra hvitt i toppen til mørkegrå i bunnen. Når det gjelder temafarge så har vi valgt blå med flere varianter med ulik lyshet og metning. Fargevalg på tekst i nettstedet er valgt ut fra bakgrunnen teksten står på. Ved lys bakgrunn bruker vi sort tekst, og på mørk bakgrunn bruker vi lys tekst, og som en HTML5 standard bruker vi blå på hyperlinker.   
  
Med tanke på fonter har vi valgt å benytte oss av sans-seriffonter uten dekorasjoner, for å gjøre leseligheten bedre for brukerne av nettstedet. Dette har vi gjort fordi mange brukere kan ha vanskeligheter med å lese seriffonter. I tillegg har vi kontrast med bakgrunnen og skriften for at man enklere kan lese skriften. Vi har også valgt å bruke H1 til overskrifter, H2 til underordnede overskrifter, og Paragraf-taggen til vanlig plaintekst. Ingressen på de forskjellige sidene har vi valgt å utheve med bruk av <strong> som sier noe om viktigheten.  
  
Applikasjonen har flere ikoner for å optimalisere brukervennligheten for brukerne. Disse ikonene er for eksempel det britiske og norske flagget helt til høyre i menylinjen. Disse vil da illustrere språk på nettstedet, siden det er en sannsynlighet for at turister som ikke kan lese norsk kommer til å bruke nettstedet. Denne funksjonen er pr. Dags dato ikke fungerende, men vi mener dette er noe som burde implemteres i applikasjonen. Vi har også valgt å bruke ikoner på knapper i front-end og back-end for å illustrere hva oppgaven til knappene er. Noen av de ikonene vi har valgt å bruke er hentet fra Font Awesome.   
Disse ikonene er vektorbaserte ikoner, som lar oss endre størrelse, farge, skygge og mye mer for å optimalisere hvordan vi selv vil ikonene skal se ut, uten å bruke .png filer eller tilsvarende.   
Dersom vi skulle fortsette å bygge på nettstedet ville vi prøvd å lage egne vektorbaserte ikoner i illustrator, eller bare brukt .png-filer som ikoner, da dette ikke er hentet fra et eksternt css-ark.  
Vi har også valgt å bruke ikoner på tab-linjen i nettleseren.



*Bilde 2: Ikoner på tab-linjen*

Disse ikonene er .png-filer, og baserer seg på om man er i frontend, der vi har valgt å bruke et fjell, som illustrerer at nettstedet omhandler et fjell og På back-end som kun administrator har tilgang til har vi valgt å bruke et tannhjul/cog som fav-ikon som illustrerer at bruker er på admin-sidene. Dersom man vil legge til nettstedet på hjem-skjerm hos Iphone eller Ipad så vil ikonet som er brukt i tab-baren på frontend bli lagt til på skjermen. Dette skjer fordi vi har lagt inn apple-touch-icon i head som genereres i *LagSide.php.*

### Krav til design

Under utvikling av nettsted har responsivt design vært en viktig faktor for brukervennligheten. Her er det da snakk om å tilpasse skjermstørrelsen og elementene på skjermen, slik at det blir mest mulig oversiktlig for brukeren. Nettstedet kommer til å bli brukt på flere plattformer og nettlesere, og da er det viktig at vi har lagt til rette for dette i applikasjonen.

Grunnen for dette er at førsteinntrykk har mye å si når det gjelder å få brukere tilbake til siden ved en senere anledning. Derfor er det viktig at siden ser pen og oversiktlig ut, ut i fra hvilken enhet brukeren benytter seg av. I tillegg kan elementer bli borte eller overlappe hverandre, og derfor er det viktig at vi har testet applikasjonen og gjort den brukervennlig på så mange enheter som mulig. Måten vi har gjort dette på er å bruke media-queries som lar oss fortelle hvordan siden skal se ut og fungere ut ifra hvor mange piksler siden benytter seg av i bredden. De valgene som er gjort i forbindelse med det responsive designet er at kolonnene går fra 3 til 2 og videre til en i bredden etter hvert som skjermstørrelsen minsker. Den vanlige menylinjen blir også erstattet med en burgermeny når skjermen blir liten nok. Grunnen for vi har valgt å gjøre det på denne måten er fordi vi vil unngå å ha horisontal scrolling på nettstedet, og at en burgermeny vil være en mye bedre løsning for å vise meny-elementene, enn å beholde meny-linjen slik den var.

### Logo

Siden ingen av oss har noen erfaring med å lage logoer fra tidligere valgte vi å benytte oss av et gratis program (<https://logomakr.com/>) for å generere en enkel logo til nettstedet.



*Bilde 3: Logo til applikasjonen*

Da utvikling av logo ikke var definert som et krav valgte vi å ikke legge større vekt på det senere i prosjektet, og valgte dermed å holde oss til denne logoen. Dersom vi skulle fortsatt videre å utvikle nettstedet er dette noe vi ville endret på. Da hadde vi laget en logo som viser Visit Vikerfjell på en mer genuin måte, da logoen per dags dato illustrerer et høyt fjell med skarpe topper, og ikke et fjell med en mer jevn topp, slik Vikerfjell har.

## Logg inn

For å komme til logg inn må man bruke linken i footeren som heter "logg inn", evt. Skrive inn stien til logg inn, som er: *localhost/VisitVikerfjell/php/logginn.php* i nettleseren. Logg inn fører til back-end, som bare administrator skal ha tilgang til. Grunnen for plasseringen av linken er nettopp på grunn av at det bare er administrator som skal ha tilgang til denne siden, og derfor er den litt gjemt bort fra resten av nettstedet. Alle brukere av nettstedet har tilgang til logg inn, men uten brukernavn og passord kommer man ikke inn på back-end.

Logg inn består av to funksjoner; selve logg inn, og glemt passord. Logg inn fungerer som en ordinær logg inn, hvor brukernavn og passord sjekkes opp mot data fra databasen. Hvis bruker skriver inn feil brukernavn og passord flere enn 5 ganger på rad, blir bruker automatisk utestengt fra å kunne logge inn på admin-sidene i 15 minutter. Dersom brukernavn og passord stemmer får brukeren logget seg inn. Her vil også brukernavn og passord lagres i session-variabler, som brukes videre inne i back-end.

Glemt passord er en funksjon for å få tilsendt nytt passord. Her har vi valgt å bruke php-mailer, som er en funksjon for å sende e-post. (<https://github.com/PHPmailer/PHPmailer>) Denne mailfunksjonen fungerer slik at brukeren skriver inn sin registrerte e-post, og deretter sjekkes e-posten opp mot data i databasen. Dersom mailen finnes vil brukernavn og en nytt generert passord sendes til e-postadressen. Dette passordet sendes først til e-postadressen, før det saltes og legges inn i databasen. Her valgte vi å bruke e-postadressen som knutepunkt for brukeren istedenfor brukernavn. Grunnen til at vi valgte e-postadressen er fordi at den er unik, og som regel har en bruker forskjellige brukernavn på forskjellige nettsteder, men en e-postadresse er bestandig unik. Ved å bruke e-postadresse vil vi da forhindre problemer dersom brukeren skulle glemme sitt brukernavn.

## Back-end

Som nevnt tidligere er back-end sidene for administrator av nettstedet. Disse sidene er laget for at administrator kan gjøre endringer på nettstedet uten å stille krav til forkunnskaper innenfor html, css, php og lignende. Dette har vi gjort mulig ved å lage funksjoner i php, slik at administrator utfører endringer på nettstedet direkte på siden.

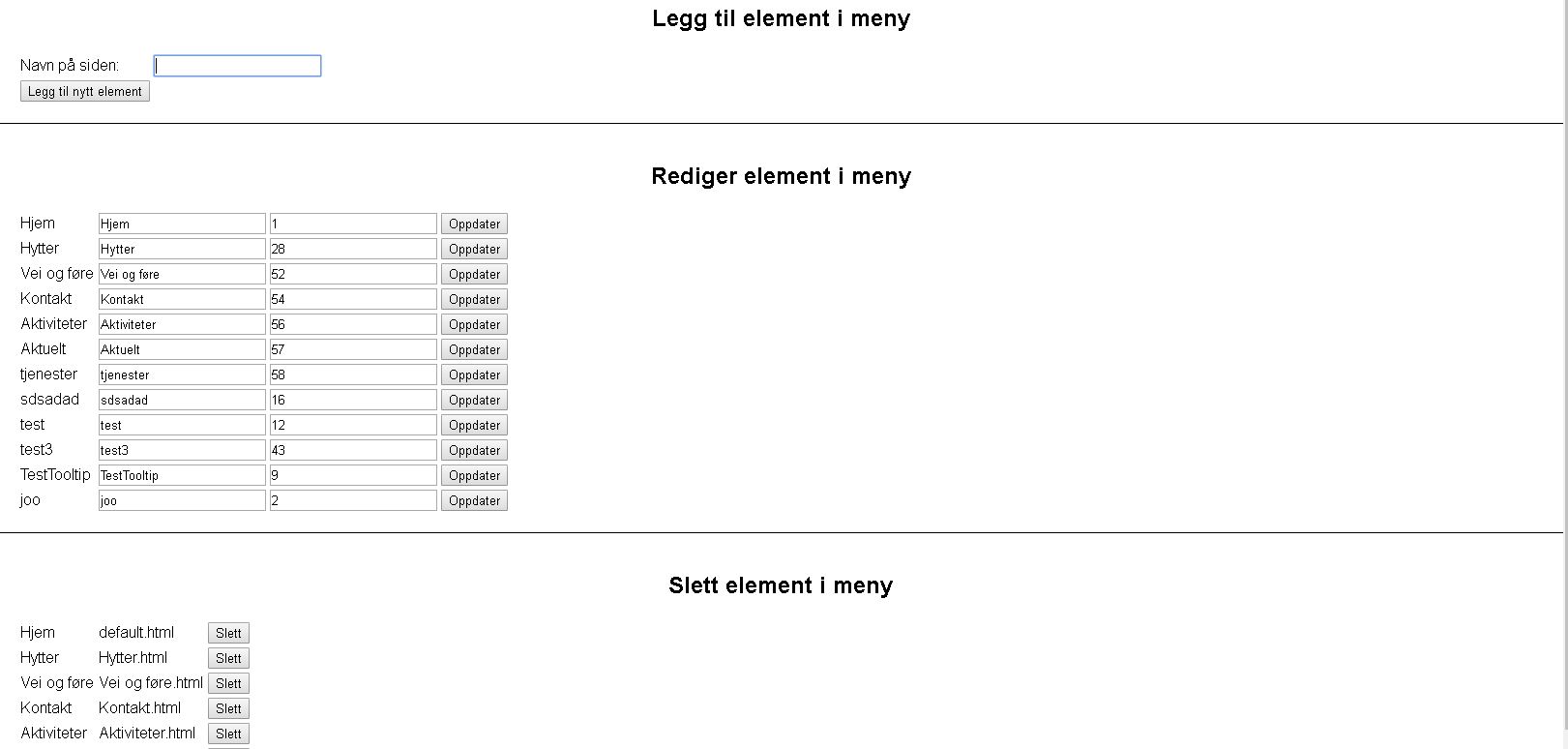
Som nevnt i logg inn må man logge seg inn for å komme til backend. Her har vi brukt session-variabler for å forsikre oss om at en bruker faktisk er logget inn. Disse session-variablene blir blant annet brukt for å sjekke om brukeren har vært inaktiv i for lang tid. Dersom brukeren er inaktiv i mer enn 10 minutter blir man da bli logget ut fra sidene. Variablene blir også brukt for at ikke hvem som helst skal kunne komme seg inn på back-end sidene. Dersom man skriver inn en url til en backend-side vil man ikke få tilgang til denne siden fordi brukeren ikke er definert.

Back-end består av fire sider, meny, sider, brukere og bilder.

### Meny

Siden "meny" har funksjoner for å legge til, redigere og slette menyelementer. Denne siden er delt opp i tre deler. Den første delen er laget for å legge til et nytt menyelement. Her skal administrator bare skrive inn navnet til siden. Dette er da navnet som vil vises på knappen i menylinjen. I tillegg bruker vi elementnavnet for å lage navnet på siden. De andre radene i databasen, som for eksempel rekke, legges til automatisk uten at administrator trenger å legge det til selv. I feltet for rekke har vi i tillegg lagt til at det bare er mulig å skrive tall.

Den andre delen er redigering av menyelementene. Her vil en liste av alle menyelementene vises, og man kan endre navnet til siden, samt rekke til elementet. Den siste delen er sletting av menyelementene. Her vises en liste med alle menyelemente, hvor man trykker "slett" for å slette elementet.



*Bilde 4: bilde av adminmeny.php*

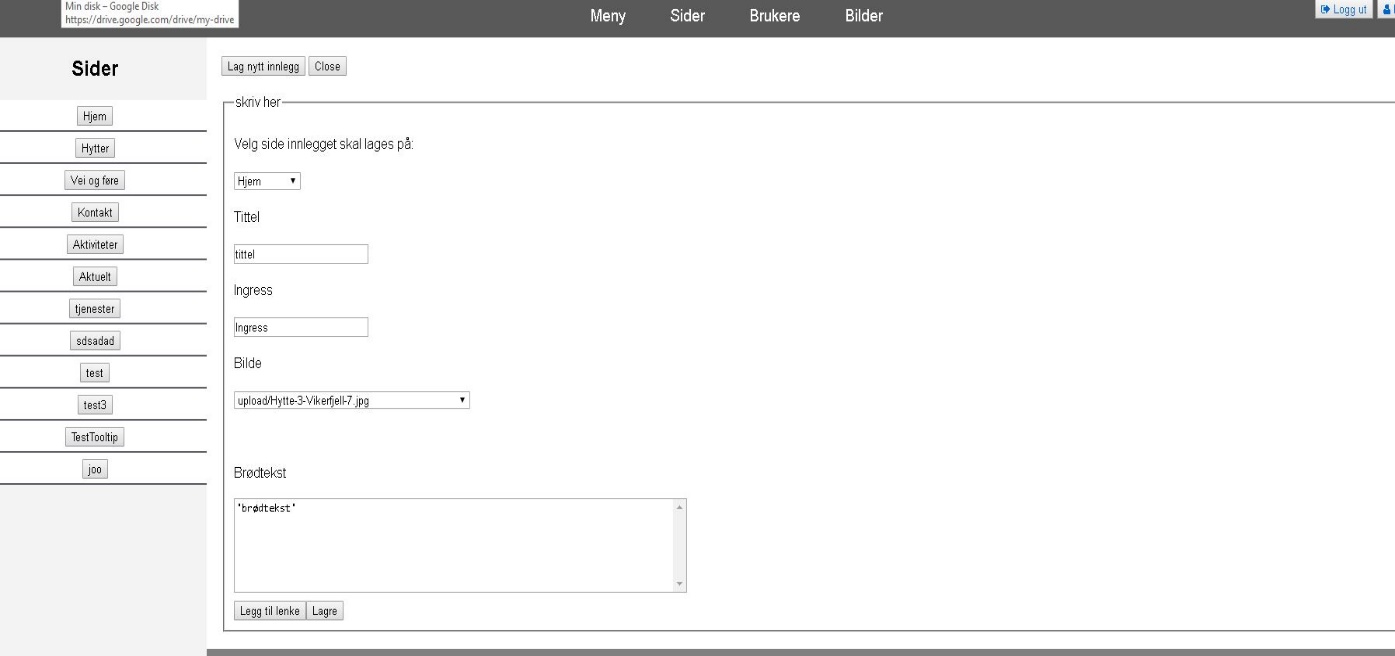
### Sider

Siden "Sider" er startsiden for back-end. Denne siden brukes hovedsakelig for å håndtere innholdet på sidene. Denne siden har to deler. Til venstre er det en sidemeny, med menyelementene på front-end. Når man trykker på disse knappene vil innholdet til den valgte siden dukke opp i feltet til høyre. Her vil man kunne oppdatere og slette innholdet.

Til høyre er også et felt for å legge til et nytt innlegg. Her må administrator velge side innlegget skal legges til under, samt annen innholdsinformasjon som tittel, bilde og lignende. Det er også mulighet for å legge til en link i brødteksten til innlegget.

Med tanke på bilder vil administrator få opp et drop-down element med bildene som er lagret i databasen. Dersom man skal ha et nytt bilde i et innlegg må man først legge til bilde på siden adminbilder.php.

Databasen vi bruker har en tabell for å koble innhold med bilder. Dette skjer i bakgrunnen slik at administrator slipper å tenke på dette.

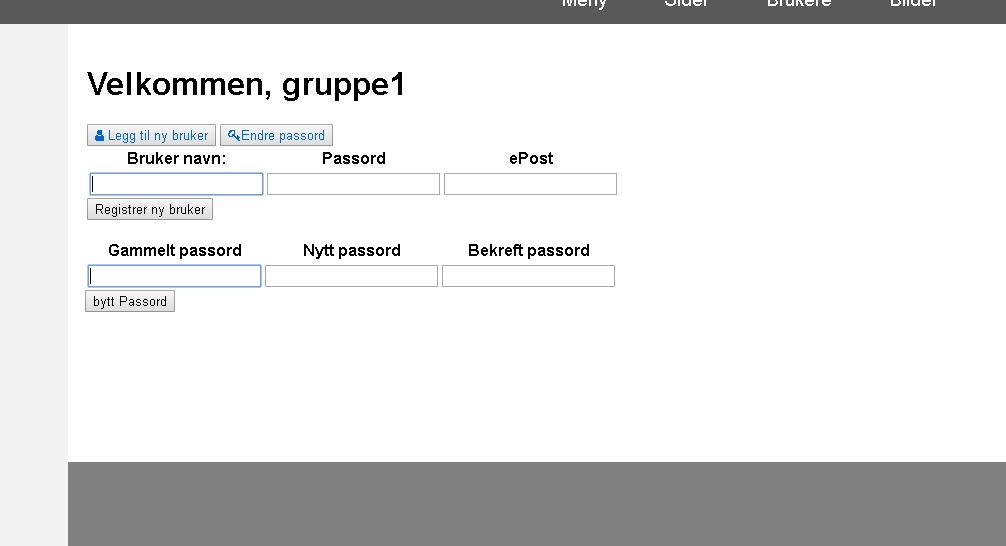


*Bilde 5: bilde av admin.php*

### Brukere

Siden "Brukere" er siden for å legge til nye brukere, samt å endre passord på innlogget bruker. Administrator fyller inn tekstfeltene, og da legges informasjonen inn i databasen. Funksjonen for å endre passord dukker opp når administrator trykker på knappen "endre passord". Her blir da session-variabel passord brukt for å hente passordet til innlogget bruker. Dersom session-passordet ikke er likt det gamle passordet vil det ikke foretas en endring for passordet. I tillegg må det nye passordet skrives inn to ganger for å være sikker på at man ikke skriver feil. Dette er tiltak vi har gjort for å gjøre applikasjonen så brukervennlig som mulig.

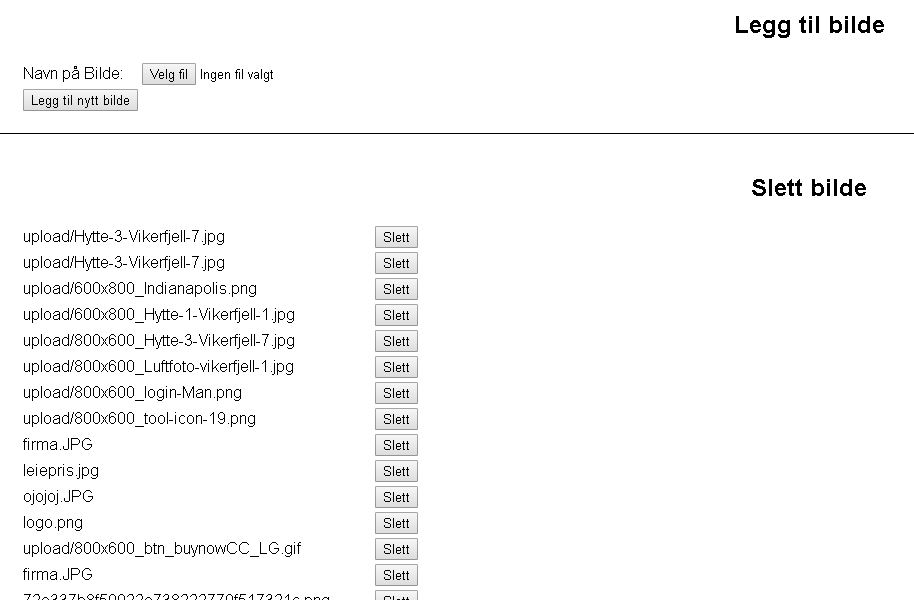
Ved endring av passord og legge til ny bruker blir informasjonen sendt til databasen, hvor passordet alltid saltes.



*Bilde 6: bilde av adminbruker.php*

### Bilder

Den siste siden på back-end er "bilder". På denne siden vil administrator kunne legge til nye bilder, samt slette allerede eksisterende bilder som ligger i databasen. Den første delen er legge til bilde. Her har vi implementert en "type=file" slik at man direkte kan hente navnet på bildet. Ved å bruke denne metoden blir prosessen mer brukervennlig, med tanke på at det ikke burde være mulig å legge til et bilde som ikke eksisterer i bildemappen. Den andre delen er sletting av bildene. Her vil da en liste med registrerte bilder dukke opp, hvor man trykker "slett" for å slette definert bilde.



*Bilde 7: bilde av adminbilder.php*

### Generering av html

Denne applikasjonen genererer html-sidene til front-end fra back-end. Funksjonen som lager sidene kjører hver gang administrator endrer eller oppdaterer data fra databasen som har påvirkning til en eller flere sider. Her vil da alle sidene som allerede eksisterer bli overskrevet med de nye sidene, og dersom en side slettes helt vil siden bli slettet fra katalogen. Med tanke på genereringen har vi valgt å oppdatere alle sidene på en og samme gang. I denne typen applikasjon vil ikke oppdateringer forgå hyppig, i tillegg til at endringer på en side kan føre til endringer på en annen side. Her unngår vi da komplikasjoner ved å generere alle sidene på nytt. Ulempen med å generere alle sidene hver gang er at det vil ta lengre tid å utføre endringer, men som sagt vil ikke oppdateringer skje så hyppig i en slik applikasjon. Vi har også optimalisert spørringene så mye som mulig for at genereringen skal ta kortest mulig tid.

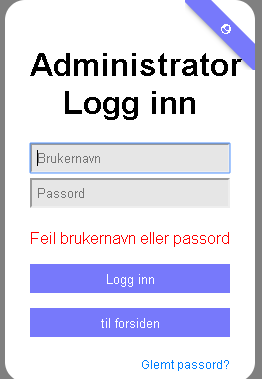
### Brukervennlighet i backend

I backend har administrator store muligheter for å gjøre endringer på nettstedet. Etter kravet at administrator ikke skal trenge å skrive kode av noe slag, har vi satt brukervennlighet som et viktig punkt her. Dette omhandler hovedsakelig dataflyten til og fra databasen. Et tiltak vi har gjort her er tilbakemelding når endringer skjer. Dette trer i kraft allerede fra logginn.php. Her har vi laget session-variabler med meldingstekst. Meldingen endres ut fra funksjonen som kjøres, for eksempel på logginn.php trer den første tilbakemeldingen i kraft når man prøver å logge inn med feil brukernavn eller passord. I tillegg har vi benyttet oss av fargene grønn og rød for å formidle betydningen av tilbakemeldingen. For eksempel dersom man prøver å logge inn med feil brukernavn eller passord vil tilbakemeldingen bli rød, og dersom det for eksempel foretas en oppdatering av menyelement vil tilbakemeldingen bli grønn. I denne applikasjonen vil det ikke være mulig å slette sider som har et innhold koblet til seg, i tillegg til at det ikke er mulig å slette et bilde som er koblet til et innlegg. Her kommer da en informativ tilbakemelding som forteller hva som må gjøres for å utføre slettingen.

I applikasjonen har vi benyttet oss av noen GET-metoder. GET-metoder kan medføre stor sikkerhetsrisiko hvis brukt feil, vi har dog kun brukt det for å sjekke om visse knapper er blitt trykket på, og ikke for å hente data. Dette vil altså ikke la en bruker manipulere data.

Back-end er utformet slik at sidene skal være enkel og konkret. Sidene er selvforklarende og man trenger ikke mye opplæring for å bruke sidene. Ved å unngå bruken av mange elementer gjør vi det lettere for administrator å bruke back-end delen.

Med tanke på responsivt design på back-end er dette noe vi ikke har tatt mye stilling til. Vi mener at det ikke er nødvendig med repsonsivt design på disse sidene, så lenge den er tilrettelagt ordinære datamaskiner. Grunnen for at vi har valgt å ikke legge mye vekt på dette er fordi det meste av bruksområdet til back-end kommer til å skje på en ordinær datamaskin. I den siste versjonen av nettstedet vil det likevel være mulig å foreta endringer ved hjelp av andre verktøy enn en datamaskin, men her er ikke back-end like optimalisert som front-end. Dette er også noe vi ville fokusert på ved videre utvikling av applikasjonen.



*Bilde 8: eksempel på tilbakemelding/feilmelding i applikasjonen*

## Front-end

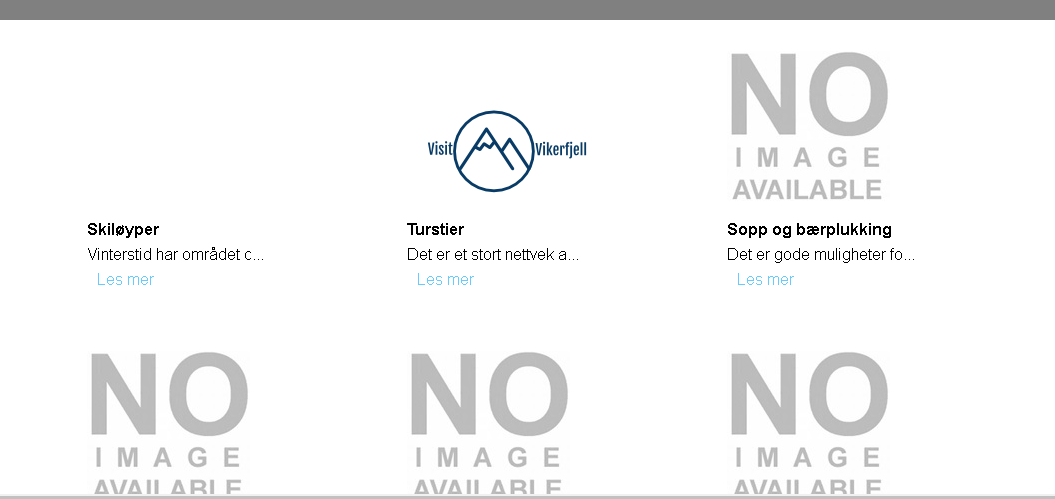
Front-end er selve brukergrensesnittet til applikasjonen. Det er disse sidene som blir generert i back-end av applikasjonen. Sidene i front-end er laget ved bruk av samme mal, som inneholder en navigasjonsmeny, header-bilde, hovedinnhold og en footer.

### Navigasjon

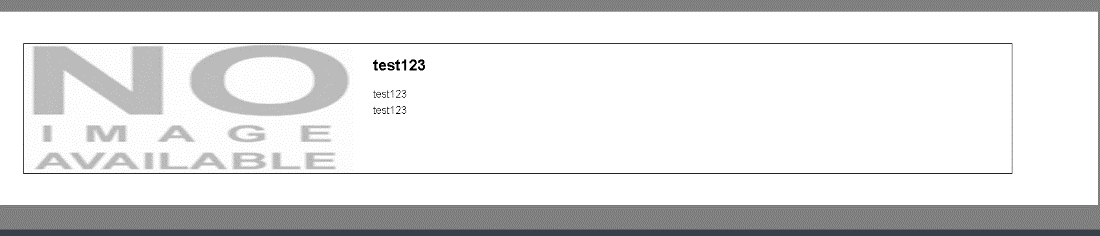
Navigasjonsmenyen er det øverste elementet i applikasjonen. Det er her man navigerer seg gjennom front-end delen av applikasjonen. Navigasjonsmenyen inneholder en logo, sidene i applikasjonen, en søkebar og knapper for muligheten for å endre språk. Søkebaren dukker opp når man trykker på søkeikonet, ved siden av knappene for å velge språk. Når man trykker på knappen vil søkefeltet dukke opp. Dette var heller ikke et krav fra arbeidsgiver, men dette er noe vi mener burde være med i nettstedet. Denne søkebaren fungerer ikke, men er laget for å vise hvordan den kunne sett ut.

Knappene for å navigere seg rundt på siden tar hovedområdet av navigasjonsmenyen. Her kommer alle sidenavnene opp i skrift, og for å navigere seg frem til de forskjellige sidene må man trykke på knappene. Her vil det bli markert med tykk skrift på den siden man er på. Menyelementene hentes dynamisk fra databasen når sidene genereres.

Hovedinnholdet blir hentet fra databasen, som genereres fra back-end. Her vil antall innhold på en side bestemme hvordan hovedinnholdet skal se ut. Dersom en side bare har ett innhold koblet til seg, vil innholdet få hele området for seg selv med bilde på venstre side, og tekst på høyre side. Dersom en side har flere enn ett innhold vil hovedinnholdet deles opp etter antall innhold. Hver linje kan maksimalt inneholde tre innhold, og dersom det er flere enn tre innhold vil det fortsette på neste linje. Dette gjentas for antall innhold til siden. Disse feltene består av innholdsbildet, tittel, kort ingress og en link til innlegget. Når man trykker på innholdslinken blir man ført til innholdet, som vises likt som de sider som bare har ett innhold.



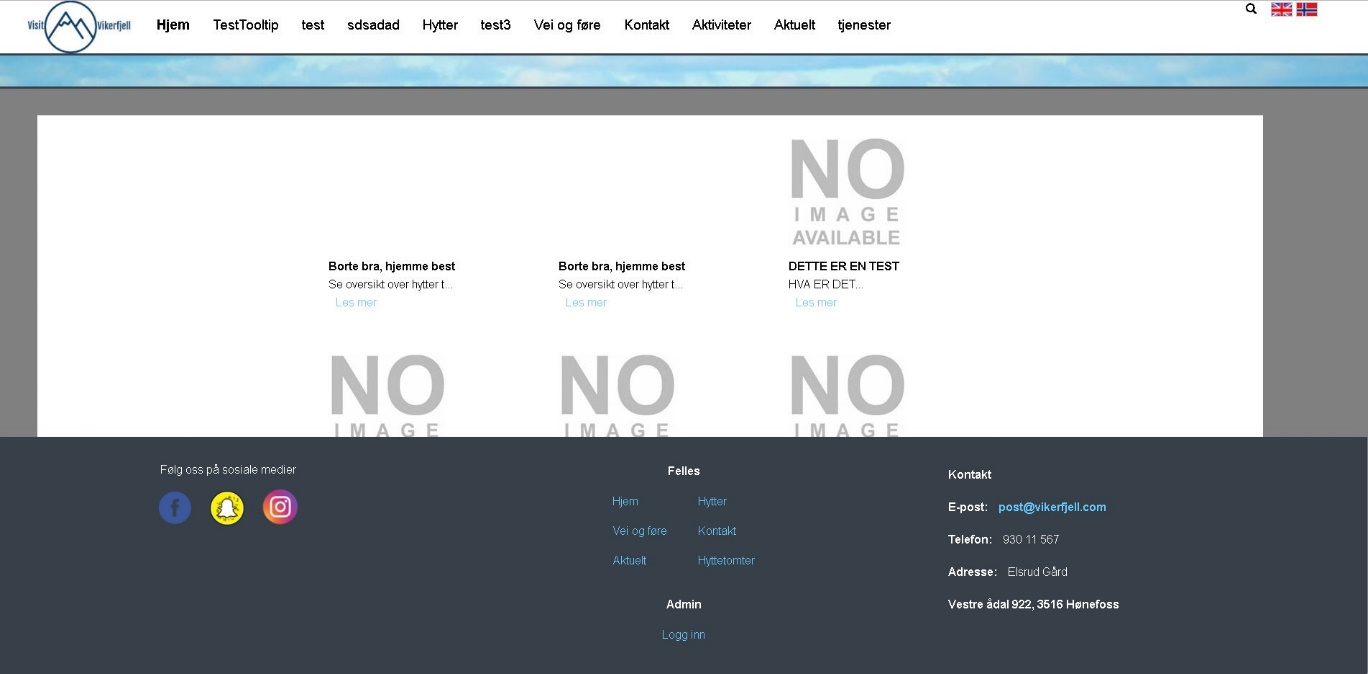
*Bilde 9: eksempel på en side med flere innhold*



*Bilde 10: eksempel på side med bare ett innhold.*

Innholdet på hver enkelt html-side er plassert på en hvit bakgrunn. Bakgrunnen symboliserer et ark, og gir inntrykk av at brukeren leser innholdet fra et ark.

Footeren sitter nederst på siden, og inneholder grunnleggende kontaktinformasjon, et nettverkskart og linker til sosiale medier. Kontaktinformasjonen er informasjon som er hentet fra de tidligere nettstedene. Nettverkskartet er ikke helt funksjonelt, men brukes hovedsakelig for å føre til logginn.php. De resterende sidene under felles er statiske sider fra en tidligere versjon. Dette er noe som kan endres, men siden dette ikke var et krav fra oppdragsgiver har vi valgt å ha det med som en idé på hva som kan være med. Linkene til sosiale medier funker rett frem som bildelinker, som fører til de forskjellige plattformene.



*Bilde 11: Bilde av front-end*

## Autentisering I applikasjonen

På dette nettstedet har vi lagt inn krav til autentisering. Dette er hovedsakelig knyttet til back-end sidene, som er ment for administratorbrukere. Brukere må da logge inn med brukernavn og passord, der passordet er både kryptert og saltet, for å unngå at personer som har tilgang til databasen skal kunne lese passordet i klartekst. Passordet vil i databasen bli lagret som en hashstring. Det er også opprettet en Session når bruker logger inn, dette kreves for å komme til back-end. Dersom bruker ikke er innloggget og prøver å skrive inn url til back-end blir brukeren da referert tilbake til innloggingssiden. Dersom bruker er inaktiv i mer enn 10 minutter blir bruker da automatisk utlogget og session avsluttet, og henvist tilbake til innloggingssiden. Ved utlogging vil session bli avsluttet.

## Konklusjon

### Selve oppgaven

Etter ett semester med utviklingsoppgaven sitter vi nå igjen med en fungerende og en funksjonell applikasjon. Vi har laget en applikasjon for Vikerfjell, hvor kravene fra oppdragsgiver er oppfylt. Dette prosjektet har gitt oss en god innsikt og forståelse for hvordan web-utvikling i prosjektarbeid kan foregå, i tillegg til god kompetanse innen html, css, js og php.

Vi mener at vi har løst oppgaven på en god måte, selv om vi har støtt på enkelte utfordringer. Her var den største utfordringen det å jobbe mot en felles database. Dette er, og var, et problem fordi alle gruppene løser oppgavene og problemene sine på forskjellige måter. For eksempel hadde noen lagt til html-kode i felter som ikke burde inneholde html. Dette gjorde at vi brukte mye tid på å løse problemer som var relatert til data fra databasen. Etter at vi ble observant på dette var det ikke et like stort problem, fordi vi da visste hva som mest sannsynlig var årsaken til problemene.

Under dette prosjektet har vi benyttet oss av inkrementell utvikling som har bestått av 4 deler og den endelige ferdigstillingen av nettstedet. Kravene og oppgavene til de forskjellige delinnleveringene har blitt sammensatt til en fullstendig applikasjon, og kravene har blitt forbedret underveis. Under delinnlevering 1 har designet vårt til en viss grad fungert som en prototype til de videre delinnleveringene. Fordelene med denne utviklingsmetoden er at det er mindre risiko for at prosjektet skal feile og at funksjonalitet med høyest prioritet blir best testet. Vi har også benyttet oss av parprogrammering for inkrementer innenfor de forskjellige delinnleveringene. Deretter har vi byttet om på rollene der vi har kontrollert og testet de forskjellige delene som er utviklet. På den endelige ferdigstillingen har vi gjort iterasjoner på hele prosjektet for å sjekke at alle kravene og funksjonalitet er oppfylt.

med tanke på oppgaveteksten er det noen elementer som ikke er implementert. Dette er blant annet submeny, som vi valgte å ikke ta med siden vi ikke fant et behov for en undermeny på dette nettstedet. I tillegg valgte vi å ikke ta med prefiks i katalogen sidene. Her brukte vi tittelen som sidenavn. Grunnen for at vi gjorde det slik er at det er lettere for brukeren å se hvor vedkommende er på siden bare ved å se på url-en.

### Hva ville vi gjort annerledes

Som nevnt tidligere i rapporten har dette prosjektet bestått av 4 delinnleveringer, hvor vi utførte delinnleveringene til oppgitt innleveringsfrist. Med tanke på hva vi ville gjort annerledes dersom systemet skulle lages på nytt er blant annet oppsettet av hele oppgaven. Vi startet med og gjøre delinnlevering for delinnlevering. Etter de første delinnleveringene var ferdig fant vi ut at sidene skulle genereres fra back-end. Da hadde vi allerede laget statiske sider til applikasjonen. Hadde vi vist det fra starten hadde vi laget applikasjonen med mer objektorientert PHP. I tillegg var det mange på gruppen som hadde lite eller ingen erfaring med css, html og php. Dette førte til at det ble mye testing og lignende. Problemet her var at det var mye unødvendig kode som ble liggende igjen. Dette kunne vært løst med å være flinkere til å kommentere koden.

Med tanke på selve arbeidsprosessen er det også noen områder vi ville ha endret på. Her er det da hovedsakelig snakk om kommunikasjon. Med tanke på lagring og oppbevaring av filer valgte vi å bruke ordinær skylagring over OneDrive for at alle på gruppen enkelt fikk tilgang til filene. Dette var greit i starten, men etter hvert viste det seg at det skapte problemer med å slå sammen filene dersom flere på gruppa hadde gjort flere endringer. Noen filer ble skrevet over som i utgangpunktet ikke skulle endres, og noen filer ble ikke oppdatert. Dette problemet hadde ikke oppstått dersom vi hadde benyttet oss av git-tjenester som GitHub, GitBucket eller GitKraken, for å redigere og dele filer.

Dette prosjektet har vært en prosess mellom oss, som utviklere, og foreleser, som arbeidsgiver og fagperson. Noe vi ikke har vært flinke til er å ha regelmessige møter med arbeidsgiver for å diskutere utviklingen av nettstedet. I starten av utviklingsprosessen fikk vi et dokument med oppgavetekst og kravspesifikasjoner. Det førte til at vi gjorde det som var definert der, og det som ikke var definert bestemte vi selv. Her burde vi diskutert utviklingen med arbeidsgiver fremfor å bestemme dette selv. Vi tror at grunnen til at det ble slik er at vi alle er ganske nye innenfor html, css, php og lignende, og vi vil ved eventuelle senere prosjekter forsøke å ha bedre kommunikasjon mellom oss og arbeidsgiver.

### Forbedringspotensialet

Applikasjonen har noe forbedringspotensial. Her har vi valgt å ikke gå inn på de små forbedringene som kunne vært gjort, men heller se på det store bildet.

En ting vi har tenkt på er å implementere noen funksjoner og sider på front-end delen. Her har vi blant annet ikke med en egen kontakt-side. Dette er noe vi ikke har laget, og dersom vi skulle utvikle applikasjonen videre er dette noe vi ville implementert. Her tenkte vi å ha en grunnleggende side for kontaktinformasjon, samt funksjoner for å sende mail til administrator, samt gjestebok eller lignende. Vi ville også implementert eksterne tjenester, slik som værmelding fra yr. Dette er detaljer ved nettstedet som er med på å skape en stor forskjell.

I back-end er det noen områder vi også ville endret på dersom applikasjonen skulle videreutvikles. Her er det blant annet generering av innhold. I den ferdige versjonen av applikasjonen blir alt innholdet satt opp på samme måte. Vi kunne tenkt oss å ha muligheten til å velge mal til innholdet. Da tenker vi at det kunne lages noen maler som administrator kan velge mellom når innholdet lages. I tillegg er det en del som kan optimaliserer med tanke på responsivt design til forskjellige enheter i back-end. Vi har heller ikke bekreftelse når man skal oppdatere eller slette elementer i databasen, noe som burde vært med. Dette vil hindre endring som ikke er meningen. Dette er noe som kunne vært forbedret til videreutvikling av applikasjonen.

### Ferdigstilling

Gjennom prosjektet har applikasjonen blitt testet av alle medlemmene på gruppen. Her har vi da hovedsakelig testet funksjonene i back-end, i tillegg til hvordan de genererte sidene ser ut og hvordan de blir vist i applikasjonen. Etter grundig testing fungerer nå alt i applikasjonen som det skal mot forskjellige nettlesere og plattformer.

## Kilder:

**Mailfunksjon:**

GitHub *”PHPMailer”,*Hentet  1. Mai 2017, fra:

<https://github.com/PHPmailer/PHPmailer>

**Ikoner:**

Font Awesome "The icons", hentet 28. April 2017, fra:

<http://fontawesome.io/icons/>

Og referanse I koden til stilarket:

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

**Favicon:**

Schwarz, D. "Here's Everything You Need to Know About Favicons in 2017" hentet 06.06.2017, fra:

<https://sympli.io/blog/2017/02/15/heres-everything-you-need-to-know-about-favicons-in-2017/>

Favikoner, fra:

<http://findicons.com/>

**Ribon på admin login:**

Rene Spronk, "How to Create a Corner Ribbon", hentet 02.05.2017, fra:

<http://www.cssportal.com/blog/create-corner-ribbon/>

**Responsivt kart i footer:**

Kartverket, "Lage kart på nett", hentet 23.03.2017, fra:

<http://www.kartverket.no/data/lage-kart-pa-nett/>     
css-link: https://unpkg.com/leaflet@1.0.1/dist/leaflet.js

**Testing:**

"Screenfly" Hentet 02.02.2017, fra:

<http://quirktools.com/screenfly/>

**Universell utforming**

Sandnes, F.A. "Universell utforming av IKT-systemer."

Responsive Web Designer Tester, hentet 13.02.2017, fra:

<http://responsivewebdesigntester.com/>

**Logo:**

Free LogoMaker, hentet 23.01.2017, fra:

<https://logomakr.com/>

**Burgermeny:**

Terwan, E. "Pure CSS Hamburger fold-out menu", hentet 19.05.2017, fra:

<https://codepen.io/erikterwan/pen/EVzeRP>

**PDF converter**

Hentet 02.08.2017

<https://www.freepdfconvert.com/no>

**Programmer som er brukt:**

WinMerge

Microsoft Word

Notepad++

Xampp