

Prosjektrapport - Content Management System

Martin Riksheim

Gruppe 30.

<https://github.com/Datateknikk/prosjektoppgave-varen--22-prosjekt-30>

Fildeling nettsted kalt 'Mini' for visning og deling av bilder, video, dokumenter og filer.

https://kark.uit.no/~tri032/flask_prosjekt/

Vanlig bruker:

User: Martin

Pw: pepsi

Administrator:

User: admin

Pw: admin

Nettstedet har all ønsket funksjonalitet beskrevet i CMS oppgave teksten pluss mer.

Dokumentet «*funksjonalitet.pdf*» inneholder en liste over nettstedets funksjonalitet og en kort demonstrasjon inkludert bilder.

Websidens frontend design:

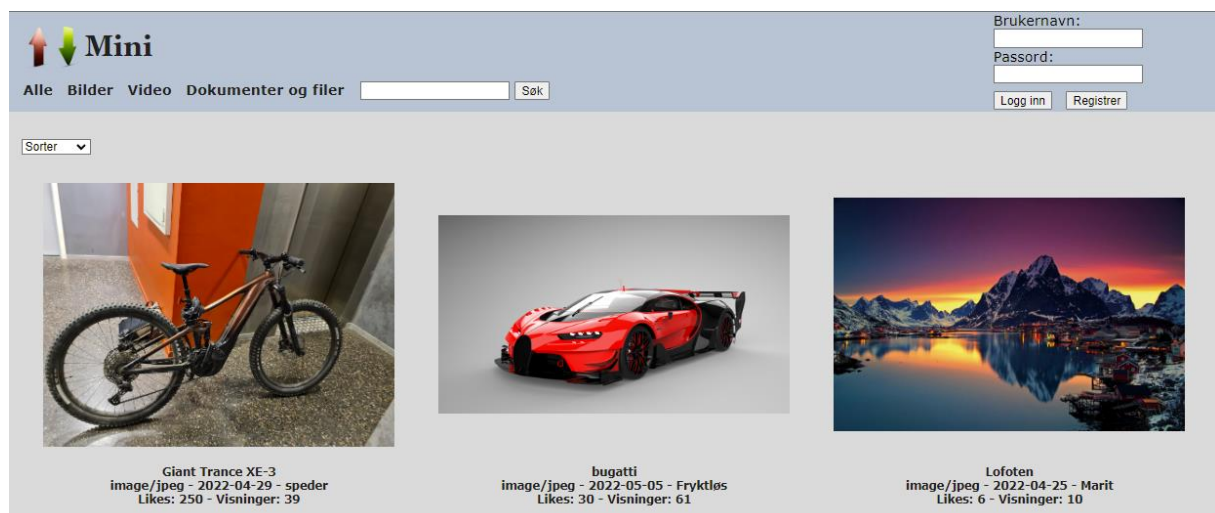
Valgte å gå for et design med 'actionbar' på toppen av siden som alltid er tilgjengelig, hvor operasjoner som kategorivalg, innhold søk og bruker pålogging utføres.

Innholds oversikt:

Oversikt over innhold vises i grid view på 450x400px per innhold med bilder på opptil 400x300px sammen med utvalgt relevant informasjon som tittel, filtype, etc.

For at visning av blandet innhold skal se bra ut viser filer som ikke inneholder bilder eller video et relevant bilde tatt fra en egen 'asset' db tabell basert på innholdets filtype.

Ved å trykke på innhold her sendes bruker videre til en utvidet visning av innholdet.




Utvidet visning:

Viser bildet/video/filen og all informasjon om innholdet.


Bildet/filen eller filnavnet kan trykkes på i for å åpnes i fullskjerm eller lastes ned.

Kommentarfelt plassert i nederste del av utvidet visning.

 **Mini**

Alle Bilder Video Dokumenter og filer Søk


Innlogget som:
Martin
[Logg ut](#)
[Last opp innhold](#)



Giant Trance XE-3

Fil:20210831_215803.jpg - Type:image/jpeg - Size:457391 bytes

Beskrivelse:
Fulldemptet Giant Trance XE-3
elsykkel

 Likes: 250
Dato: 2022-04-29
Lastet opp av: speder
Synlighet: Alle
Visninger: 41
Tags: e-bike sykkel fulldempt

Kommenter

Kommentarer

hissig sykkel

Martin - 2022-05-04 16:07:00 [Slett]

Konklusjon av frontend design løsning:

Meget fornøyd med hvordan det visuelle endte opp med å se ut. Synes det ble oversiktlig og ryddig med en minimalistisk stil, og føler ikke det er noe vesentlig jeg ville forandret på i ettertid.

Nettsiden passerer validering hos <https://validator.w3.org/> uten feilmeldinger.

Inkludert HTML og CSS kode:

HTML templates

base.html
content.html
edit.html
register.html
upload.html

CSS stilark

cms.css

Backend løsning:

Webserver er laget med hjelp av python flask og bruker flere av modulens funksjoner.

Alle nødvendige sikkerhetsfunksjoner som oppgave teksten krever er tatt i bruk.

XSS protection er ivaretatt av flask templates innebygde auto escaping.

Bruker login, logout og login-status handles av flask LoginManager og er beskyttet med CSRF protection. Passord lagres som hash i databasen.

Alle forms med data validert med WTForms før eventuell skriving til databasen.

Restriksjoner for visning av innhold kun for påloggede brukere er alltid ivaretatt.

Funksjoner som krever pålogging eller en bestemt bruker valideres alltid før utførelse.

Alle SQL spørringer mot databasen er beskyttet mot sql-injection med bruk av prepared statements.

Liste over alle implementerte funksjoner er lagt ved i dokumentet «*funksjonalitet.pdf*».

Konklusjon backend løsning:

Er fornøyd med resultatet av backend løsningen også og føler ikke det noe jeg ville endret.

Nettsiden har fått de funksjonene som er naturlig å ha i forhold hvilken type nettside det er og alle av oppgavens krav til sikkerhet og funksjoner er implementert.

Inkluderte python filer og klasser:

Flask applikasjon main

CMS.py

Db setup, cursor og sql spørringer

CMSDB.py

- class MyDb

Klasser for objekter

content.py

- class Content

asset.py

- class Asset

comment.py

- class Comment

user.py

- class User

Klasser for WTForms objekter

Forms.py

- class LoginForm

- class UserForm

- class ContentForm

- class CommentForm

ER-Databasemodell:

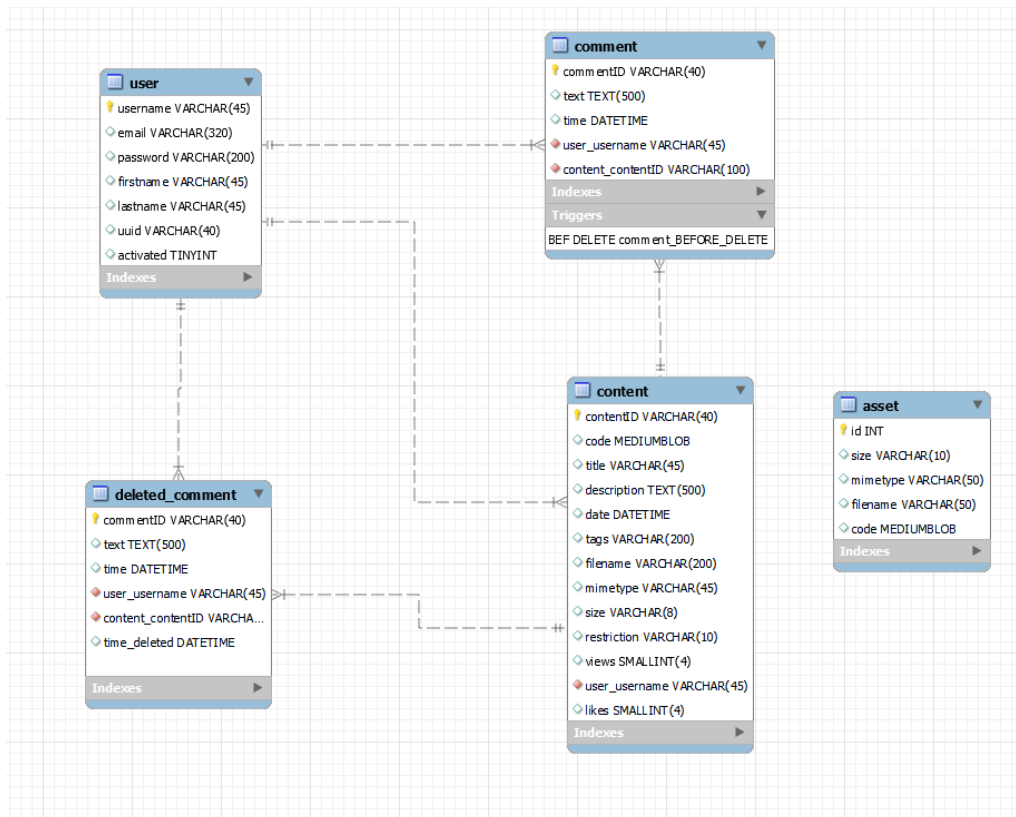
Gjort noen små endringer i forhold til ER-modell første utkast.

- Fjernet flertall 's' ending i tabell navn.
- Lagt til 'likes' i content tabellen.
- Endret noen datatyper.
- Lagt til tabellen 'asset'.

Databasemodellen har fortsatt rom for forbedring og 'content' tabellen kunne for eksempel vært splittet opp i flere tabeller. Egen tabell for fildata med unike filnavn/id som fremmednøkkel, egen tabell for tags etc...

Endte ikke opp med å inkludere forbedringene ettersom utvikling prosessen var kommet såpass langt at veldig mye ferdig skrevet kode måtte vært gjennomgått og endret på for å implementere dette, og følte ikke det var verdt å bruke den ekstra tiden til det.

Databasemodellen fungerte ellers bra og støttet ikke på noen begrensinger under utvikling av nettsiden.



Databasemodell og SQL scripts:

Skjermutklipp av databasemodell

- CMS_ER_diagram.png

MySQL Workbench modellfil

- cms_model.mwb

SQL script for opprettelse av databasemodell

- cms_setup.sql

SQL trigger script for backup av slettende kommentarer

- comment_backup_trigger.sql

Andre vedlegg / prosjektdokumenter:

- funksjonalitet.pdf
- prosjektdagbok.xlsx
- CMS_prosjekt_presentasjon_riksheim.mp4