

Blogg webapplikasjon:

**backend:**

- 1-Flask som server
- 2-sqlite eller json ?
- 3-Flask-login plugin

**Frontend:**

- 1- bootstrap and CSS
- 2-vue.js
- 3-html
- 4- tilpasse til phone versjon

**funksjoner:**

- 1- login and registrere deg
- 2- legg til blogg
- 3-redigere blogg
- 4-slette blogg
- 5-kommentere
- 6-like
- 7-filter
- 8-søk
- 9-sorter
- 10-eksempel data
- 11-errors behandling
- 12-bilder behandlig og vising

hva applikasjonen gjør:

vår applikasjon er en type og en del av nettside for å publisere bloggposter den kan skrives av en enkelt person eller flere personer.

innleggene i bloggen sorteres i omvendt kronologisk slik at det siste innlegget kommer først i siden.

temer er:

- 1-teknologi
- 2-reise
- 3-nyheter

data skal lages:

- 1-blogg
- 2- bilder
- 3- kommentarer
- 4- forfatter
- 5- kilde

## Rapport om Blogg Webapplikasjon

### 1. Innledning

Denne rapporten beskriver designet og funksjonaliteten til en blogg webapplikasjon utviklet med Flask og Vue.js. Applikasjonen tillater brukere å registrere seg, logge inn, publisere blogginnlegg, redigere og slette innlegg, samt interagere med andre brukeres innlegg ved å kommentere og like dem.

### 2. Teknologier og Verktøy

#### - **Backend:**

- Flask som server.
- SQLite eller JSON for lagring av data.
- Flask-login-plugin for autentisering og autorisasjon.

#### - **Frontend:**

- Bootstrap og CSS for grunnleggende design.
- Vue.js for dynamisk interaksjon og oppdatering.
- HTML for strukturering av innhold.
- Responsivt design for tilpasning til ulike enheter, spesielt telefoner.

### 3. Funksjoner

- Registrering og innlogging.
- Opprettelse, redigering og sletting av blogginnlegg.
- Kommentering og liking av innlegg.
- Filtrering, søking og sortering av innlegg.
- Eksempeldata for testing og demonstrasjon.
- Behandling av feil og validering av brukerinput.
- Opplasting, lagring og visning av bilder i blogginnlegg.

### 4. Implementasjon

#### - **Backend:**

- Flask-ruter implementeres for å håndtere forespørsler knyttet til bruker- og bloggdata.
- SQLite eller JSON-database opprettes for lagring av blogginnlegg, kommentarer og brukerinformasjon.
- Flask-login-plugin brukes for å håndtere autentisering og tilgangskontroll.

#### - **Frontend:**

- Bruk av Vue.js for å opprette dynamiske komponenter for innlegg, kommentarer, osv.
- HTML og Bootstrap for å strukturere og style brukergrensesnittet.
- Responsivt design implementeres for å sikre brukervennlighet på ulike enheter.

### 5. Datastruktur

#### - **Tabeller/Entiteter:**

##### 1. Blogginnlegg

2. Bilder
3. Kommentarer
4. Forfattere
5. Kilder (valgfritt, hvis du trenger å spore kilden til informasjon i blogginnlegg)

## 6. Brukstilfeller

- Registrering av bruker og innlogging.
- Opprettelse, redigering og sletting av blogginnlegg.
- Interaksjon med andre brukeres innlegg gjennom kommentarer og liker.
- Filtrering, søking og sortering av blogginnlegg basert på ulike kriterier.

## 7. Fremtidige Forbedringer

- Implementere mer avansert bruker autentisering og autorisasjon.
- Legge til funksjonalitet for kategorisering av innlegg.
- Muliggjøre deling av innlegg på sosiale medier.
- Forbedre responsivt design for bedre tilpasning til forskjellige enheter.

## 8. Konklusjon

Denne webapplikasjonen gir en plattform for enkeltpersoner eller grupper å publisere og samhandle med blogginnlegg. Med Flask som server og Vue.js for frontend-dynamikk, tilbyr den et brukervennlig og responsivt grensesnitt for å utforske og dele innhold.

[Blogg](#) [Home](#) [blogg site 1](#) [blogg site 2](#) [blogg site 3](#) [blogg site 4](#) [blogg site 5](#)

# Blogg site

This is a simple hero unit, a simple jumbotron-style component for calling extra attention to featured content or information.

It uses utility classes for typography and spacing to space content out within the larger container.

The current link item

A second link item

A third link item

A fourth link item

A fourth link item

A fourth link item

## Blogs

