# Dokumentasjon fagprøve

## 13.02.2023-02.02.2023

## Arbeidsprosessen

Dag 1:

Første dagen skreiv jeg planleggingen for fagprøven og leverte den inn for sjekk. Den bel med en gang akseptert og jeg begynte å jobbe med den. Alt gikk som planlagt og det var faktisk mye lettere enn det jeg tenkte. Så klarte å gjøre alt jeg skrev ned for dagen og jeg klarte å gjøre at mann kunne logge inn

Dag 2:

Jeg forbedret innloggingen og gjorde at session var tilgjengelig på hele siden. Så laget jeg en navbar der mann har login knapp og gjestbok logoen. Så satt jeg opp hele databasen sånn at den er klar for at jeg skal sende inn data og hente data fra den. Også begynte jeg litt på å sende inn data og hente data. Hadde litt problemer med databasen siden jeg ikke fikk tilgang til å access den fra node. Dette fant jeg ut var fordi passordet var kryptert annerledes i den nye mysql versjon, og node bruker fortsatt den gamle, så måtte manuelt inn å endre passordet i mysql igjennom mysql workbench. Så funket det. Ellers gikk alt til plan og bra.

Dag 3 og 4

Det jeg gjorde i dag var å gjøre at mann kunne sende inn data til databasen, sånn at mann både kunne lage posts og kommentarer. Så lagde jeg pagination som viser 5 posts om gangen og alle kommentarene til de. Addet også at på venstre siden kan mann se sine egne posts. Her har mann også tilgang til å endre de og slette de, jeg jobbet til litt før 18 på torsdag og til 1630 på onsdag.

Slutten av torsdag skulle jeg egentlig jobbe med admin panel, men fikk ikke tid til dette da det oppsto noen problemer under håndtering av data og visning av data som fikk d til å ta lengre tid

Dag 5

Dag 5 gikk feil fri og fikk laget hele admin siden der mann kan slette posts til andre, kommentere på posts osv. så har jeg begynt å planlegge dokumentasjon til oppgaven som har blitt gjort

Dag 6 og 7

Jobbet med dokumentasjon og egenvurdering.

## API dokumentasjon

|  |  |
| --- | --- |
| Localhost:3000/api/createPost | Dette callet skal være post med variablene: postBody, postTitle, posterName. Dette blir inserted til databasen med sql query: Insert into posts (posterName. postTitle, postBody, postDate) |
| Localhost:3000/api/createComment | Dette callet skal være post med variablene: postBody, posterName. Dette blir inserted til databasen med sql query: Insert into postComments (posterName. postTitle, postBody, postDate) |
| Localhost:3000/api/changePost | Dette callet skal være med post og inkludere variablene: postBody, postTitle, postId. SQL query til dette api’et er UPDATE posts SET |
| Localhost:3000/api/deletePost | Dette callet skal være post med variablen: postId som kommer fra ID kolonnen til posts tabellen. SQL query til dette callet er: DELETE FROM posts WHERE id=»postID her» og enda en som er DELETE FROM postComments WHERE postId=»postId her» |
| Localhost:3000/api/getComments | Dette er en GET request. Denne kjør automatisk når siden lastes inn og henter kommentarer med sql query: SELECT \* FROM postComments. |
| Localhost:3000/api/getPosts | Dette er en GET request. Denne kjør automatisk når siden lastes inn og henter kommentarer med sql query: SELECT \* FROM posts |

## Database oppsett

### Generell oppsett

Databasen er satt opp med en schema som heter guestbook. I dette schemaet så har jeg 2 tabeller. Disse tabellene heter postComments og posts. Der post comments har alle kommentarer til posts i seg og posts har alle posts i seg.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

### postComments

Dette er tabellen med alle kommentarer i seg. Som mann ser på bildet så er det satt opp med 5 kolonner. idC er PK til tabellen som også har auto increment. posterName er navnet til den person som lager en kommentar. posterBody er innholdet til kommentaren. posterDate er datoen at kommentaren blir opplastet, og postId er Id’en til posten personen kommenterer unner

Et bilde som inneholder bord

Automatisk generert beskrivelse

### Posts

Dette er tabellen som holder dataen til alle posts som er på siden. Som man ser på bildet har denne tabellen også 5 kolonner. Den første er id, id er PK til tabellen som også har auto increment. posterName er navnet til den person som lager posten. postTitle er tittelen til posten personen lager. postBody er innholdet til posten. posterDate er datoen at kommentaren blir opplastet

Et bilde som inneholder tekst

Automatisk generert beskrivelse

### Brukere

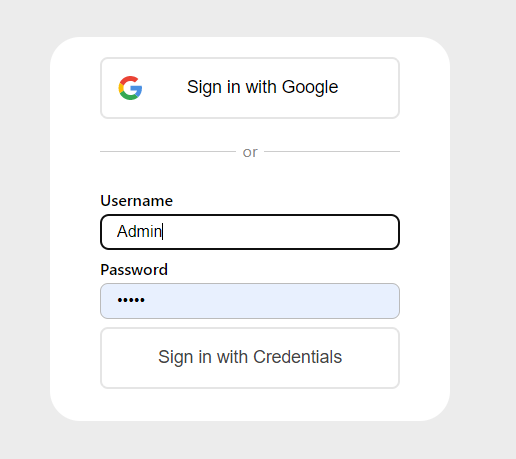
For brukere og innlogging til siden så brukes modulen next-auth. For denne modulen har jeg satt opp en auth fil som heter […nextauth].ts som ser sånn her ut:

Et bilde som inneholder tekst

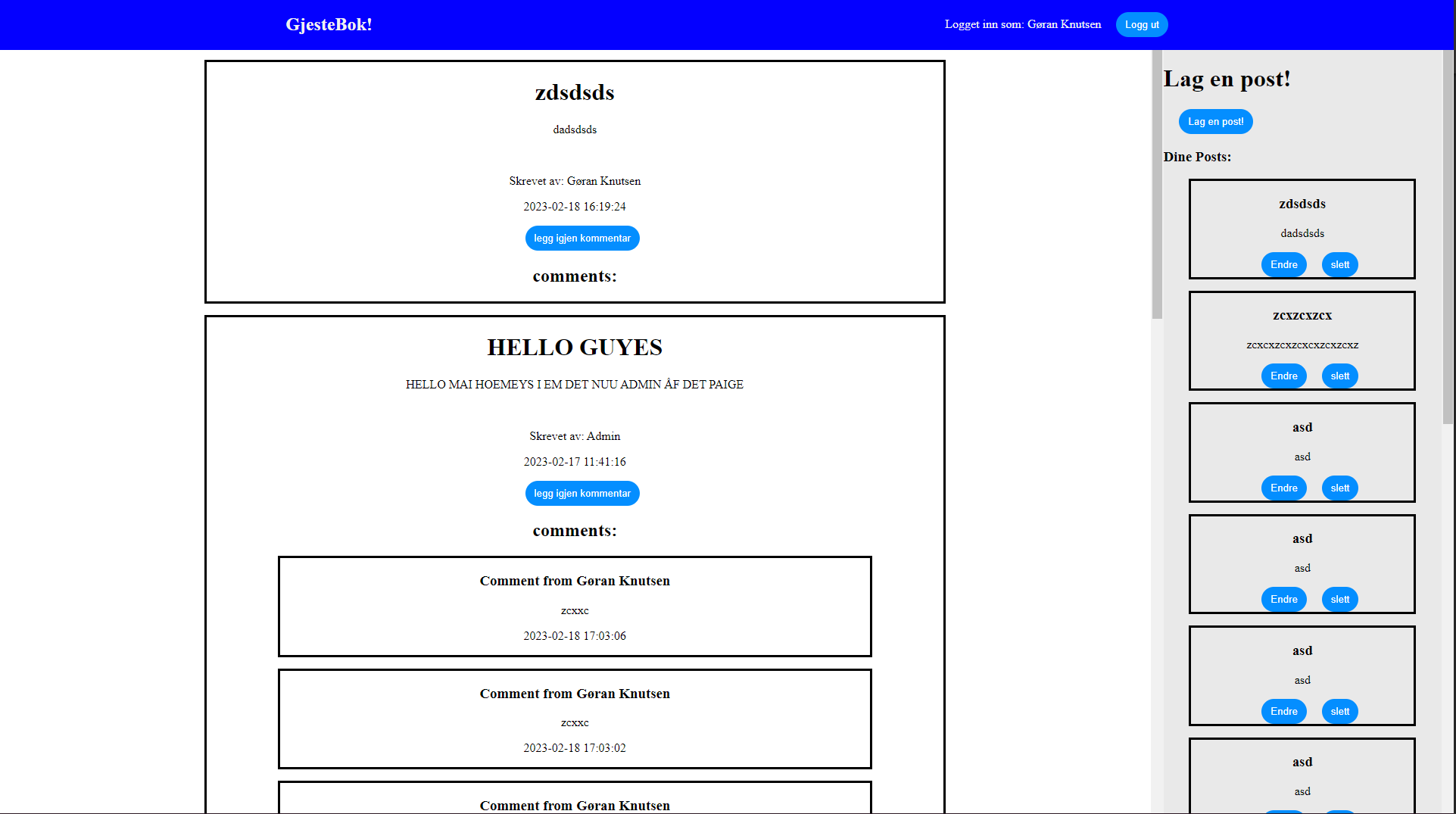
Automatisk generert beskrivelse

Den er satt opp til å kunne ta innlogging med credentials og igjennom google. Credentials innloggingen brukes bare for den person som skal logge inn med admin brukeren. Her må de bruke brukernavnet Admin. Alle brukere til siden logger bare inn igjennom google også henter siden all dataen den trenger og putter det i en session automatisk. Her får mann både navnet og mailen til brukeren. Dette gjør at jeg ikke trenger en table i databasen min der jeg lagrer brukere lokalt.

Dette er hvordan innloggings siden ser ut med bruk av nextAuth



### Sideoppsett



Bildet over vise oppsettet til en normal bruker på siden. Toppen av siden ser mann logoen til siden og på høyre siden av navbaren så står det hvem du er logget inn som og en logg ut knapp. Er man ikke logget inn her så er det en knapp der det står logg inn på. På hoved delen av siden er det satt opp at den største delen som er til venstre med hvit bakgrunn viser fram posts som er laget. Her har man en container for hver post der tittelen er, så body av posten, så hvem som har skrevet den, under det kommer tiden den ble postet, så kommer det en knapp der mann kan legge inn kommentarer. Etter denne kommer da alle kommentarene som er skrevet til posten, med navnet til person, body til kommentaren og datoen. Dette er modalen som kommer opp når mann vil legge igjen en kommentar:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Høyre siden med grå bakgrunn kan mann finne knappen for å lage posts, trykker mann på denne kommer det en modal der mann putter inn dataen.



Under denne knappen så finner mann en liste over alle posts man selv har lagd. Her har mann mulighet til å slette og endre postene sine. Endrer mann en post så kommer denne modalen opp der mann kan putte inn det nye mann vil det skal stå:

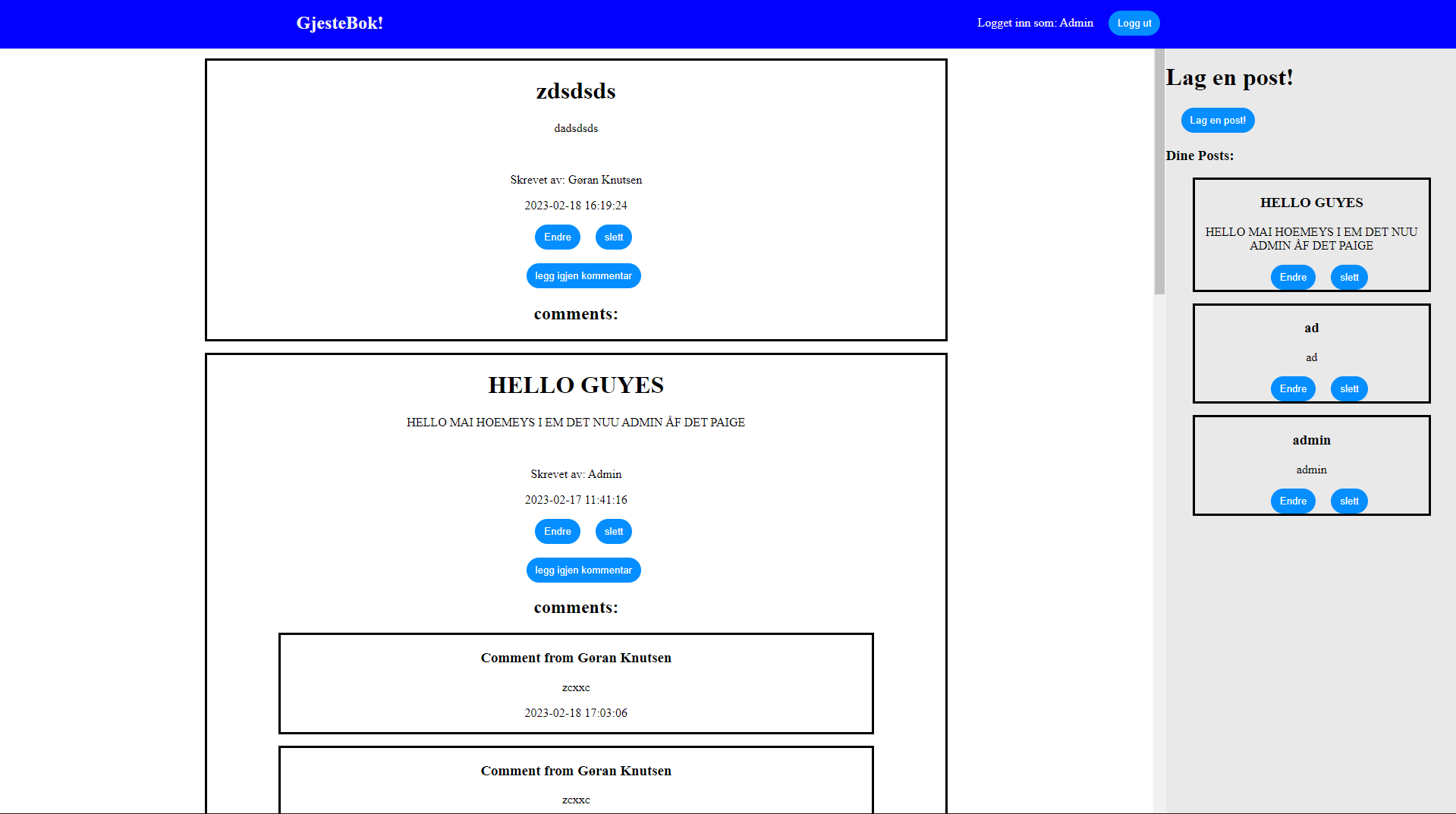


Når mann ikke er logget inn så ser mann bare dette:

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Når mann logger inn som admin på siden så vises siden fram på lik måte som for brukeren, den eneste forskjellen er at mann kan trykke slett og endre på hver eneste post som vises på venstre delen av skjermen.



### .env fil

Et bilde som inneholder tekst

Automatisk generert beskrivelse