

# Projekt 2 rapport

Jon Bjarke Sørensen

## Indhold

Indledning med indhold og kort resume.....	3
Problemformulering.....	3
Metodeovervejelser .....	3
Sikkerhed.....	3
Sessions .....	3
JSON web tokens .....	3
Ændelige valg .....	3
Funktioner og databaser.....	3
Express og pug.....	4
MongoDB .....	4
API .....	4
Research og Analyse .....	4
Opsætning af quiz .....	4
Delet funktion .....	4
Konstruktion .....	5
MongoDB .....	5
API .....	5
Evaluerings af proces.....	5
Konklusion.....	5
Reflection .....	6
Referencer .....	6

## Indledning med indhold og kort resume

Projektet givet bygger på at designe en sikker side hvor brugere kan registrere en konto, login på en konto og slette en konto.

Siden skal inkludere en quiz som skal have en score der gemmes i en database og skal opsættes i express med en api til sendelsen af quiz til andre ikke relaterede web apps

## Problemformulering.

Hvordan kan man ved hjælp af forskellige sikkerheds metoder opbygge en sikker web app som er forbundet til en database som kan gemme brugere samt også slette dem med quizzes som der kan sendes som en api

## Metodeovervejelser

Her vil der blive beskrevet hvorfor siden gør brug af specifikke metoder

### Sikkerhed

Her vil der blive kommet ind på de metoder der kunne implementeres for at styrke sidens sikkerhed

### Sessions

Sessions vil blive brugt på siden for at forbedre sikkerheden ved at sikre for at bruge forbliver logget ind i en givet timeframe og til at authenticate dem ved opbevaringen af informationer som skal bruges for at kunne navigere rundt på siden uden at blive smidt af siden

### JSON web tokens

JSON web tokens ligesom session hjælper til at forhindre uautoriseret adgang til hjemmesiden. Med JSON web tokens er siden sat op til kun at give brugeren tilladelse til at logge in på siden hvis de har en token.

Token laves efter validering af password og username

Hvis du prøver at logge in på siden uden en token vil der komme en fejl samtidig hvis du kommer ud på forsiden mens du har en token vil du automatisk blive sendt tilbage til menuen efter login

### Ændelige valg

Til sidst blev det valgt at arbejde med JSON web tokens siden det også blev brugt i sidste projekt og det derved ville være nemmere at arbejde ud fra.

Der blev tænkt over at inkludere begge dele men det blev til sidst valgt ikke at gå videre med da det ville blive for ruget med koden samt en smule overkill

## Funktioner og databaser

Her vil der blive kommet ind på hvilket metoder der blev brugt i forhold til funktioner og databaser

## Express og pug

Til opsætning af siden var brugen af Express påkrævet dette ledte så til mange muligheder for brug af

## MongoDB

mongoDB er en database som der kan bruges til at sende, gemme og kalde informationer fra til forskellige dele af projekter.

I dette projekt blev der gjort brug af mongoDB sammen med mongoose

## API

En del af projektet var også at lave en api til spørgsmålene sådan at tredjeparts sider ville være i stand til at gøre brug af dem igennem en række kommandoer, til brug i deres egne projekter

## Research og Analyse

For at lave et ordentligt projekt ville en del research og analyse være nødvendigt

## Opsætning af quiz

Siden skulle inde holde en eller flere quizzes hvis resultat skulle kunne sendes til databasen så det var vigtigt at finde en måde hvorpå det resultat brugeren ville få ville kunne nemt blive sendt over i databasen her blev der kigget på en række forskellige quiz opsættninger før jeg ændte med at gøre brug af den som nævnes i kapitlet(Konstruktion)

Den blev valgt grundet dens simplicitet sådan at man nemt ville have noget at arbejde udefra til når der vil laves forbedringer senere

## Delet funktion

Siden det skal være muligt for at slette deres brugere her, blev der kigget på mongodbs indbygget slet funktioner (MongoDB n.d.)

Dette blev så opbygget til

```
85  })
86
87  route.delete('/delete', async (req, res)=>{
88    console.log('test1')
89    const deleteUser = req.body.email
90    try{
91      const findEmail = await User.findOne({email:deleteUser.email})
92      if(findEmail !== null){
93        const validateInput = await compare(deleteUser.email, findEmail.email)
94        if(validateInput){
95          db.collection("users").deleteOne(deleteUser, function(err, obj){
96            if (err) throw err;
97            console.log("1 document deleted");
98          })
99        }
100      }
101    }
102    }catch(err) {
103    }
104  }
105  })
```

(den virker ikke lige nu)

## Konstruktion

Meget af det der blev lavet først, blev bygget op af tidligere projekter for at spare tid da det var noget som der ikke var så meget af tilgængeligt i løbet af projektet. (kunuutkaali 2023)

Quizzen var så sat op baseret på et tutorial fundet på internettet med ændringer som der gør koden mere overskuelig blandt andet ved at køre spørgsmålene i JSON sådan at de ikke fylder i JavaScript hvis der skulle tilføres flere spørgsmål. (Simple Steps Code 2023)

User og forsidens routes var lavet med kode fra andet semesters første projekt med til førelsen af nye routes for at kunne udføre de ønskede funktioner som der var sat af opgaven.

Samt brug af flere webkilder til at opbygge en bedre forståelse for hvordan at de nødvendige funktioner kan sættes op

## MongoDB

Projektet gjorde brug af mongodb til at opbevare brugerinfo samt info om deres score på quizen. For at gøre processen nemmer blev der brugt mongoose skemaer hvilket bruges til opsætningen af collections i mongodb " *Everything in Mongoose starts with a Schema. Each schema maps to a MongoDB collection and defines the shape of the documents within that collection.*" (Mongoose 2023)

Skemaerne kan så importeres ind i routes hvor de så blev brugt til at sende info til databasen

## API

Api blev opsat i et dokument for sig selv væk fra alt det andet.

API blev sat op til at den kan printe enten alle quizzes i konsollen eller en specifik eller en række af quizzes.

## Evaluering af proces

Generelt var der for lidt tid men ellers okay interessant

Det kan nemt siges at der mange steder ville kunne laves forbedringer til projektet samt at grundet

## Konklusion

Det kan konkluderes at det endelige produkt lever op til problemformulering ved det absolutte minimum.

Grundet tids pres blev der skåret mange hjørner i produktets konstruktion. Så hvis jeg kunne gå tilbage og få en uge til så ville jeg med glæde gøre det.

## Reflection

Hvis der havde været mere tid var der en lang række af ting som der kunne have været gjort bedre

- Færdig implementering af delet funktion
  - Delet funktion gennem sendt email
- Kodet quiz fra bunden
- Flere quizzes
  - Kode som der kunne tjekke siden og derved implementere den korrekte quiz
- Eksperimentering med andre authentication systemer
- Eksperimentering med andre sikkerhedssystemer

## Referencer

kunuutkaali (2023) *Todoapp*. Pug. available from <<https://github.com/kunuutkaali/todoapp>> [14 March 2023]

MongoDB (n.d.) *Db.Collection.DeleteOne()* — *MongoDB Manual* [online] available from <<https://www.mongodb.com/docs/manual/reference/method/db.collection.deleteOne/#mongodb-method-db.collection.deleteOne>> [15 March 2023]

Mongoose (2023) *Mongoose v7.0.1: Schemas* [online] available from <<https://mongoosejs.com/docs/guide.html>> [14 March 2023]

Simple Steps Code (2023) *How to Make a Simple JavaScript Quiz | Simple Steps Code* [online] available from <<https://simplestepscode.com/javascript-quiz-tutorial/>> [13 March 2023]