

HA(IT) - Introduktion til programmering og udvikling af små systemer samt databaser

Ole Torp Lassen - CBS - Institut for Digitalisering - Februar 2025

Godkendelsesopgave 5 (G5)

I denne opgave skal I lave en simpel webapplikation ved hjælp af Express.js til at håndtere registreringen af dyr i en zoo. Applikationen skal give brugerne mulighed for at tilføje nye dyr, og se listen over registrerede dyr, som en bonus kan man også lave en funktion hvor man kan søge efter dyr, men dette er dog ikke et krav til opgaven.

I opfordres til at løse opgaven i jeres studiegrupper og bruge bitlab og canvas-forum til at diskutere alternative løsninger og søge inspiration og vejledning.

Aflevering

Opgaven er med **individuel aflevering** senest:

Fredag 7. Marts kl. 23:59

Afleveringen skal indeholde:

- En zip-fil med en kodefiler (html, js, css).
- En pdf på max 5 sider med en velstruktureret beskrivelse af:
 1. Hvem, I har arbejdet sammen med.
 2. Problemanalyse:
 1. hvordan har I forstået opgaven.
 2. eventuelle uklarheder i opgaveteksten og hvordan, I har valgt at fortolke dem.
 3. uformel beskrivelse af jeres løsning (delopgaver, centrale komponenter, overordnet algoritme, etc.).
 3. Kort beskrivelse af jeres implementation.
 1. centrale funktioner og begreber og
 4. Afrundning med refleksion og diskussion, f.eks.:
 1. er der noget, som I ikke har kunne få til at virke? Hvad kan grunden være? Hvordan har I testet?
 2. er der noget, der kunne/burde være løst bedre/anderledes?
 3. ideer til videreudvikling, andet, ...

NB: Man afleverer **individuel** på Canvas.

- Hver studerende **skal** skrive og aflevere sin egen rapport.
- I opfordres til at samarbejde om og diskutere opgavens løsning og I må gerne aflevere identiske programmer (skriv hvem I har arbejdet sammen med).
- I praksis vil beskrivelserne af problemanalysen selvfølgelig også ligne hinanden noget.

NB: Godkendelse af opgaven er **ikke** betinget af at I har et kørende program, der løser problemet fuldstændigt, men at I har arbejdet seriøst med problemet og dokumenteret arbejdet.

NB: Brug af AI

Det betragtes som eksamenssnyd at anvende chatGPT eller lignende til at løse G-opgaverne. Mistænkelige opgaver bliver ikke godkendt og eksamenssnyd kan medføre udelukkelse fra eksamen eller bortvisning fra studiet!

Krav til løsning af opgaven

- Web-appen skal opbygges af en side som er lavet med Express.js i Javascript.
- Der skal være minimum 2 endpoints, et som er tilgængeligt på "/" som eksponeres af express som implementere en side med en brugergrænseflade hvor det er muligt at se en liste af dyr.
- Det andet endpoint skal være tilgængeligt på "/add-animal" som tager imod HTTP post-beskeder, hvor det skal være muligt at tilføje dyr.
- På hovedsiden, skal der være et script, som kører på klient-siden og gør det muligt at tilføje et dyr, forespørgslen skal sendes til det ovenstående API.
- Hvert dyr skal have et fortløbende id, navn, art, tilføjelsesdato, fødselsdato, dødsdato, alder.
- Alder skal udregnes efter fødselsdato, hvis dyret stadig er i live.
- Funktionaliteten, der skal udregne dyrets alder, skal implementeres i et js-modul og importeres.

Metode:

Den følgende metode er vejledende og kan følges, men der kan også findes andre acceptable metoder og løsninger.

Opsætning af Express.js Server:

- Initialiser et nyt Node.js-projekt.
- Installér Express.js og andre nødvendige pakker.
- Opret en grundlæggende Express.js-server, der lytter på en specifik port.

Dataopbevaring:

- Brug en in-memory datastruktur (f.eks. et array eller et json-objekt) til at gemme dyredata.

Router:

- Opsæt ruter til at håndtere forskellige endpoints:
- /: roden, der viser en velkomstbesked og en liste over registrerede dyr.
- /add-animal: Rute til at tilføje et nyt dyr til registret.
- /search: Rute til at søge efter dyr efter navn.

Logik og Moduler:

- Opret et eller flere moduler til at håndtere logikken for tilføjelse, udregning og søgning af dyr.
- Det skal sikres at applikationen er modulær og følger bedste praksis.

Brugergrænseflade:

- Opret en simpel HTML-grænseflade til at interagere med applikationen.
- Brug EJS (Embedded JavaScript templates) til at render dyredata dynamisk til HTML.

Hint til Initialisering af projekt:

```
mkdir zoo-registration
cd zoo-registration
npm init -y
npm install express ejs
```

Hint til filstruktur:

```
zoo-registration/
├── views/
│   ├── index.ejs
│   └── add-animal.ejs
├── public/
│   └── style.css
├── app.js
├── animalRegistry.js
└── package.json
```

Hint til søgefunktion:

```
function searchAnimals(name) {
  return animals.filter(animal =>
    animal.name.toLowerCase().includes(name.toLowerCase())
  );
}
```