

# Svømmeklubben Delfinen

Semesterprojekt - 1. semester

Datamatiker

**kea**  
KØBENHAVNS ERHVERVSAKADEMI



# Agenda

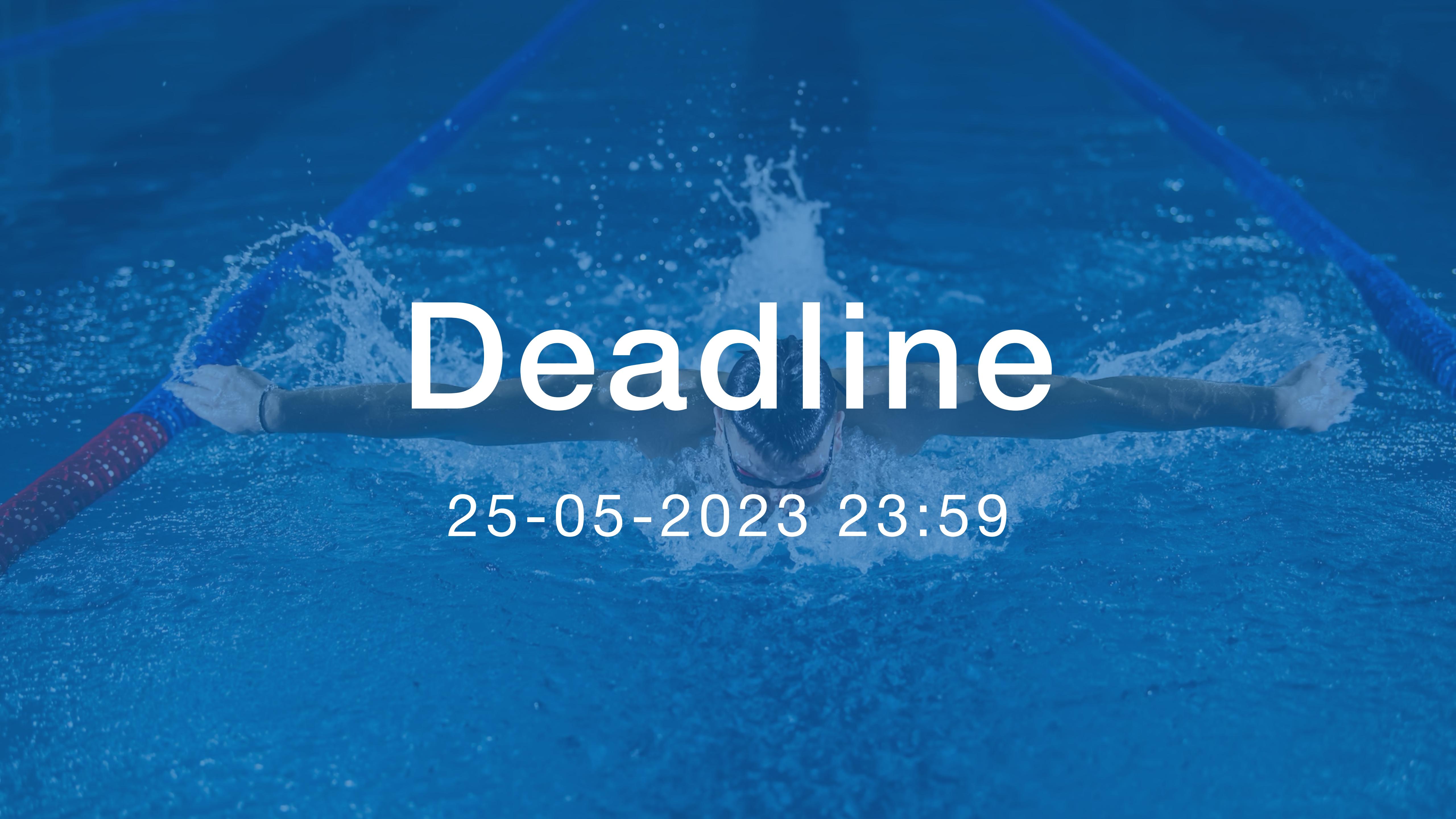
- Præsentation af semesterprojekt 🏊
- Grupper
- Views - MPA eller SPA?
- Firebase og kombiner data
- Begynd planlægning

# 1. SEMESTERS STRUKTUR

Semesteret består af 16 undervisningsuger (inkl. en masse helligdage)



Eksamens dato

A swimmer is performing the butterfly stroke in a swimming pool. The swimmer is in the middle of a pull, with arms extended forward and legs kicked powerfully. Water is splashing around them. The background shows the blue water of the pool.

# Deadline

25-05-2023 23:59

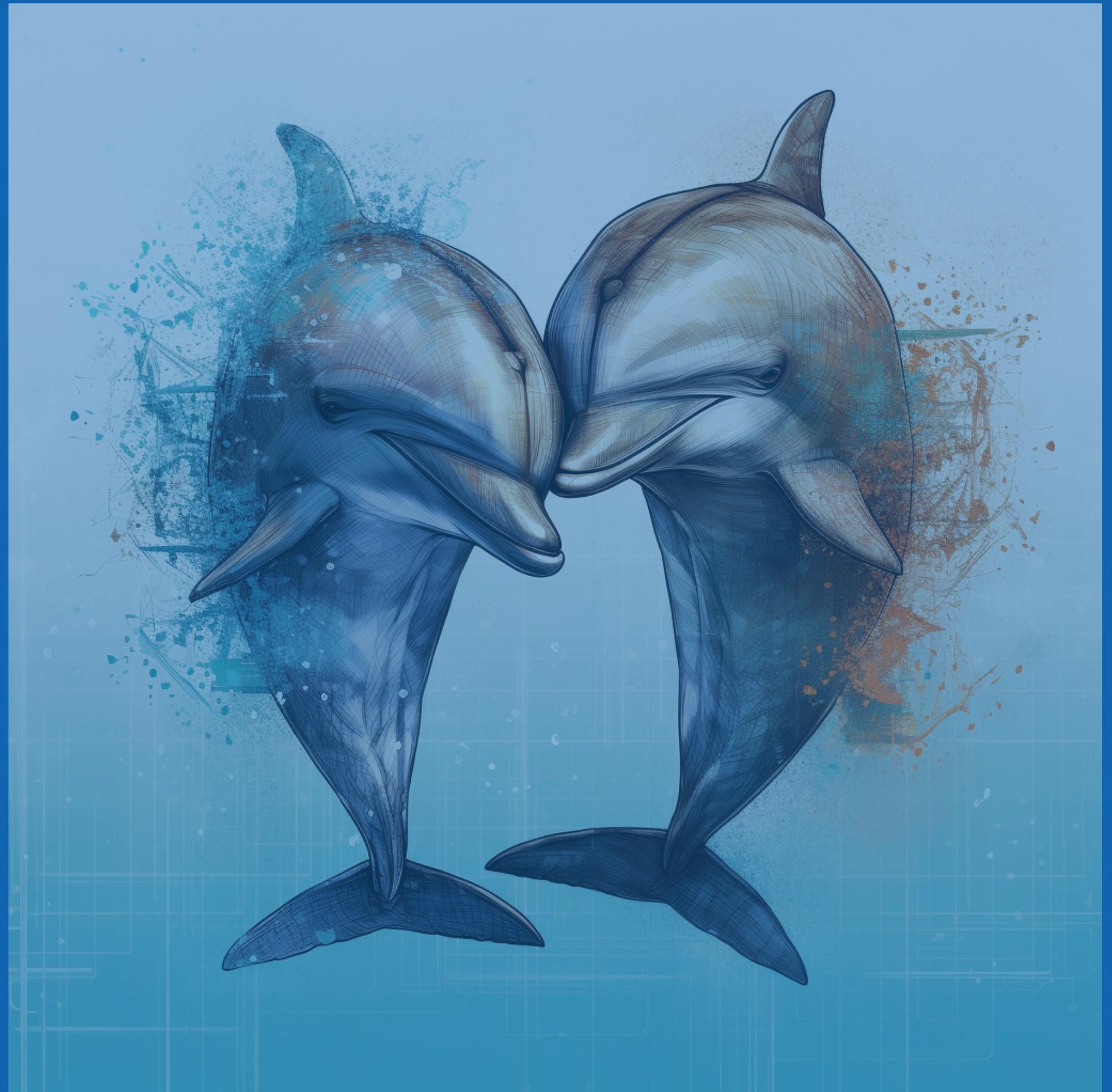
A photograph showing a group of five people in an office environment. Two women are standing on the left, one wearing glasses and a patterned scarf. Three men are seated at a long desk; one is looking towards the camera, another is looking down at a book he is holding, and the third is partially visible behind him. The background shows office cubicles and a whiteboard with some writing on it.

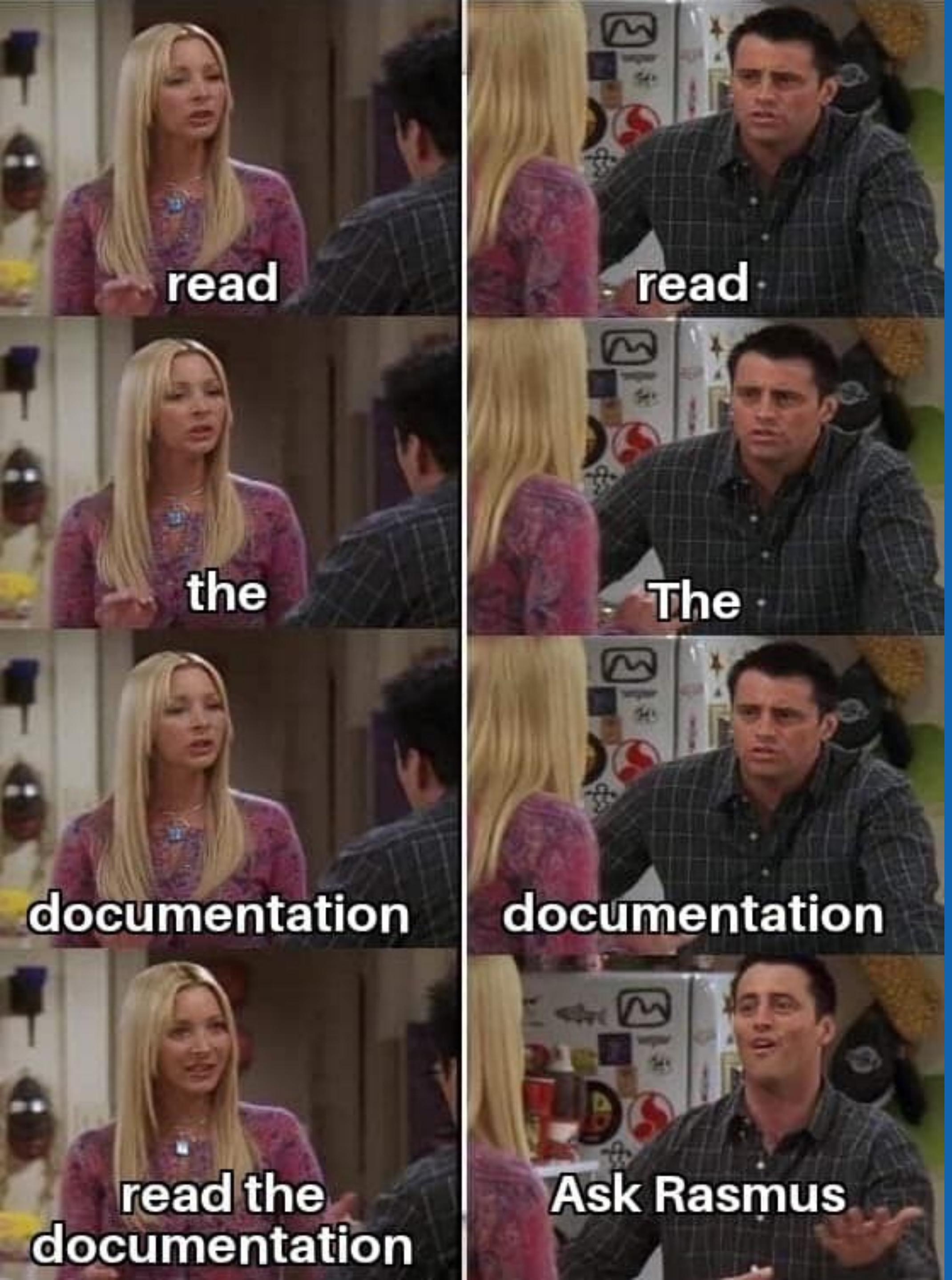
Og hvad med de der grupper der?

# Svømmeklubben

## Delfinen

- Administrativt system til at styre:
  - Medlemsoplysninger
  - Kontingenter
  - Svømmeresultater





Og, ja. Det er  
projektbeskrivelsen på  
Fronter, jeg taler om!

Så lad os se på den 💪

# Administrativt system til medlemsoplysninger, kontingenter og svømmeresultater

- Brugere med forskellige behov.
- Registreringer, beregninger, filtreringer og sorteringer



Svømmeklubben Delfinen

Svømmeklubben Delfinen er en mindre klub, der er i vækst. Klubbens ledelse ønsker derfor udviklet et administrativt system til at styre medlemsoplysninger, kontingenter og svømmerresultater. Efter samtale med klubbens formand, tegner der sig følgende billede af arbejdsopgaverne, som systemet skal understøtte.

Det er klubbens formand, der tager sig af nye medlemmer. Ved indmeldelse i klubben registreres diverse stamoplysninger om personen, herunder alder.

Desuden registreres oplysninger om personens ønskede aktivitetsform, det vil sige aktivt eller passivt medlemskab, junior- eller seniorsvømmer, motionist eller konkurrencesvømmer.

Klubbens kasserer tager sig af alt vedrørende kontingentbetaling. Kontingentets størrelse er betinget af flere forhold:

For aktive medlemmer er kontingentet for ungdomssvømmere (under 18 år) 1000 kr. årligt, for seniorsvømmere (18 år og over) 1600 kr. årligt. For medlemmer over 60 år gives der 25 % rabat af seniortaksten. For passivt medlemskab er taksten 500 kr. årligt.

Kassereren vil gerne kunne danne sig et overblik over, hvor meget klubben kan forvente at få indbetalt i kontingent i alt.

Kassereren har desuden ønsket, at systemet kan vise en oversigt over medlemmer, der er i restance.

Konkurrencesvømmerne har tilknyttet en træner. Konkurrencesvømmerne er inddelt i 2 hold efter alder. Ungdomsholdet er for svømmere under 18 år. Seniorholdet er for svømmere på 18 og over. Hver konkurrencesvømmer er desuden registreret i forhold til hvilke svømmedisipliner, svømmeren er aktiv i.

Inden for hver svømmedisiplin registreres den enkelte svømmers bedste træningsresultat og dato løbende. For de svømmere, der har deltaget i konkurrencer, registreres stævne, placering og tid. Det er på baggrund af de enkelte svømmernes resultater, at træneren udtager svømmere til deltagelse i konkurrencer. Træneren ønsker derfor en oversigt, der kan vise klubbens top 5 svømmere inden for hver svømmedisiplin (butterfly, crawl, rygcrawl og brystsømning), fordelt på henholdsvis junior og seniorsvømmere.

Og resten må du  
læse, og derved  
udledede kravene til  
systemet.

# De Generelle Projektkrav

Semesterprojekt - 1. semester

Datamatiker

**kea**  
KØBENHAVNS ERHVERVSAKADEMI

# Virksomhed

- Med udgangspunkt i casen om Svømmeklubben Delfinen, skal der udarbejdes en **risikoanalyse** med tilhørende udvidet **risikotabel**.

# Udviklingsproces

- Scrum med:
  - Scrumboard til Product Backlog og Sprint Backlog
  - Scrum Master
  - Daily Scrum møder
  - Sprint Planning, Sprint Retro og Sprint Review møder
  - Fremlæggelse af projekt, præsenteret som Scrum-forløb (Oskar giver yderligere info)

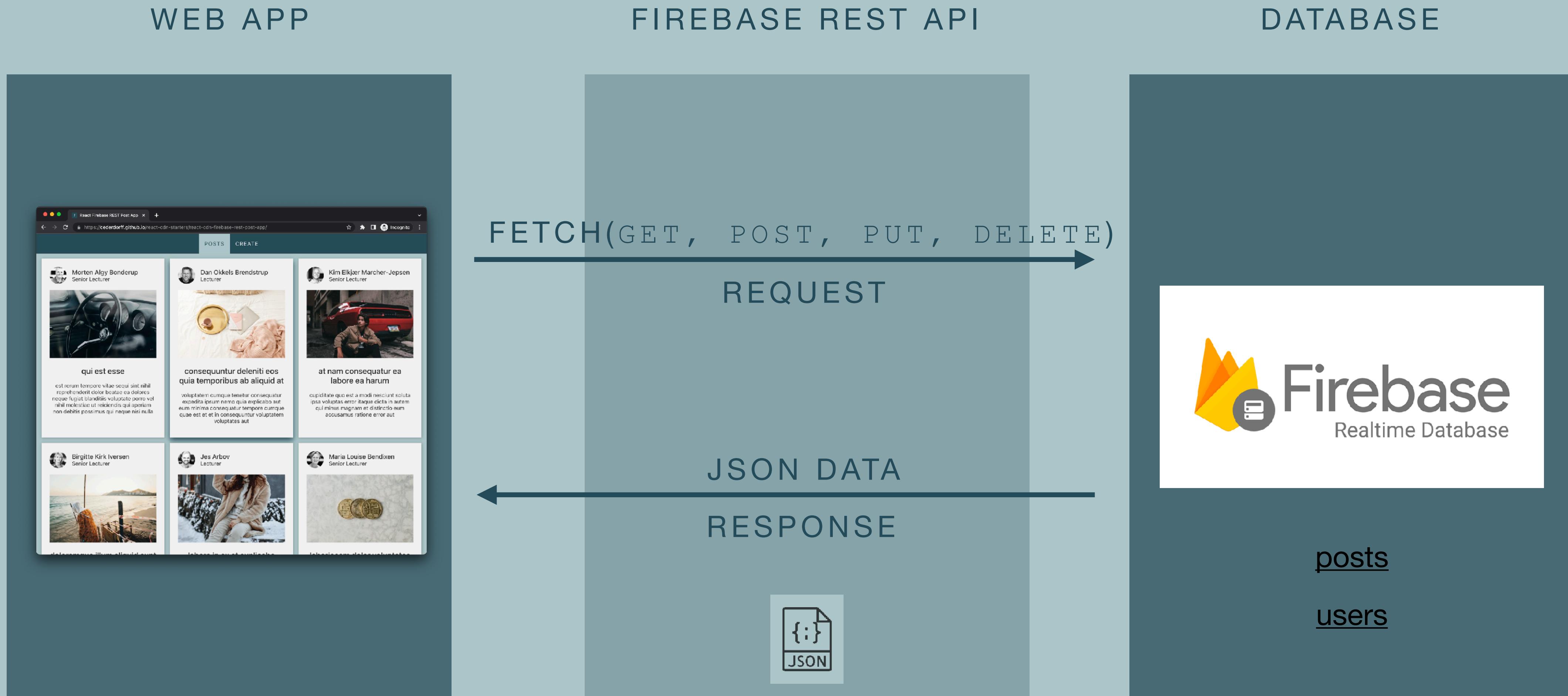
# Kravspecifikation

- Med udgangspunkt i casebeskrivelsen, skal I udarbejde en kravspecifikation bestående af:
  - User stories (inkl. acceptkriterier)
  - Løbende prioritering af user stories i samarbejde med Product Owner (kunde/kravstiller)

# Programmet

- Det vi har trænet på 1. semester:
  - Web-applikation med HTML, CSS og JavaScript
  - Interaktion med Database via REST API - GET, POST, PUT/PATCH og DELETE
  - Og anvend så mange af de andre termer, koncepter og principper, som giver mening.

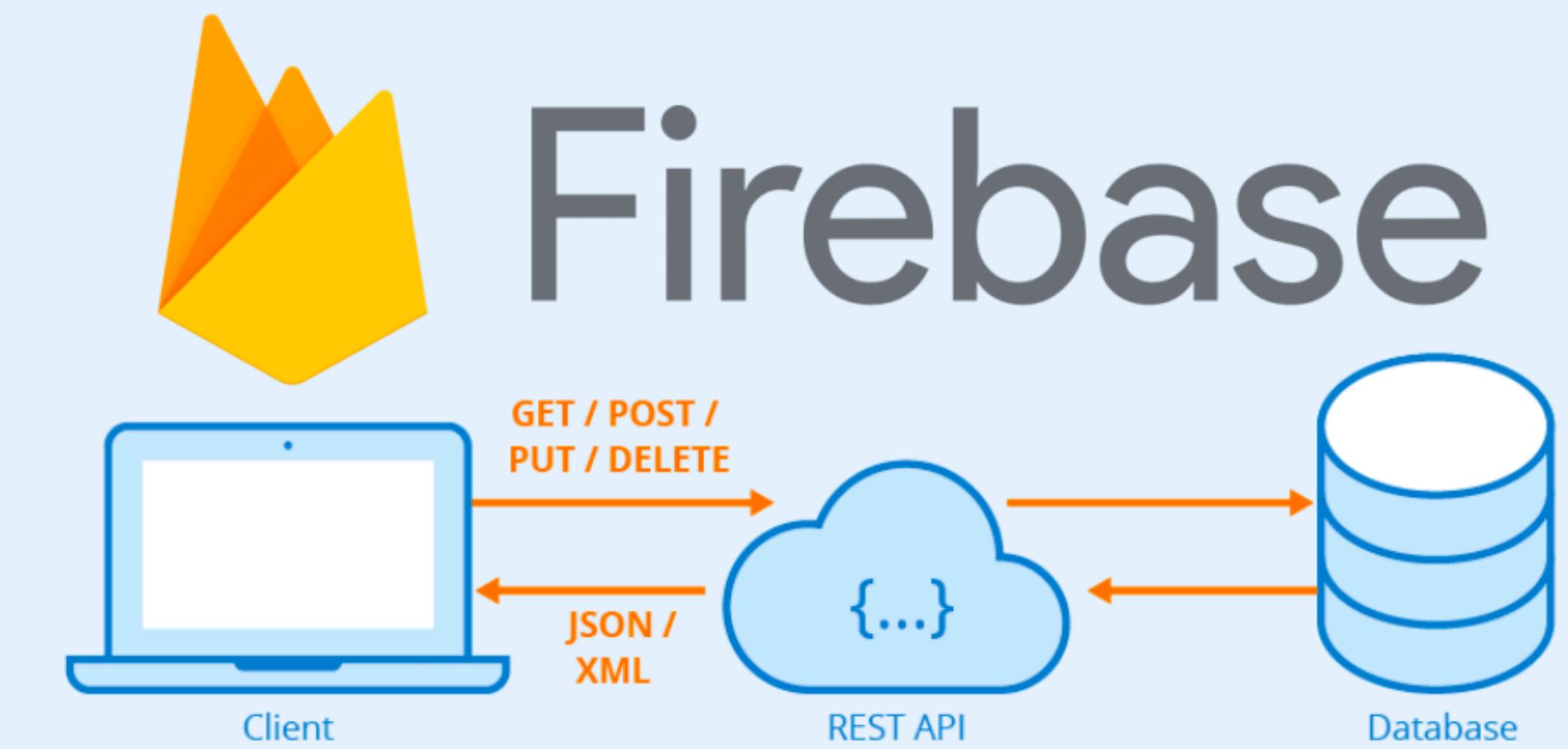
# Webudvikling - Client Server



# Programmet

- Se det som en mulighed for at øve jer og prøve “det hele” i en større sammenhæng.
- Dybere forståelse...

Ja, I SKAL  
bruge Firebase  
Realtime  
Database og  
REST API



[javascript.plainenglish.io/firebase-cloud-functions-tutorial-creating-a-rest-api-8cbc51479f80](https://javascript.plainenglish.io/firebase-cloud-functions-tutorial-creating-a-rest-api-8cbc51479f80)

# Husk programmeringsprincipper

- **Separation of Concerns** – så ting er adskilt så meget som muligt, i bruger forskellige funktioner/moduler til at manipulere data og vise data.
- **Loose Coupling** – så funktioner er så uafhængige af hinanden som muligt, eller i det mindste kun har afhængigheder en vej.
- **High Cohesion** – så funktioner der arbejder med det samme er samlet så tæt som muligt, enten i closures eller i modules.

# Desværre, ingen Libraries eller Frameworks

- Fx React, Vue, Bootstrap, Tailwind, jQuery, etc.
- Men bare rolig, det kommer! Dog ikke jQuery



Me: JQuery is cool library.  
I prefer this than JS

Rasmus:



# Brugergrænseflade

- Web-applikation - altså, det vi har lavet igennem semesteret.
- I skal designe/ prototype hele brugergrænsefladen.
- Golden Rules og Heuristikker
- Usability test af jeres prototype (wireframes eller implementering)

## 1 Visibility of System Status

Designs should keep users informed about what is going on, through appropriate, timely feedback.



Interactive mall maps have to show people where they currently are, to help them understand where to go next.

## 2 Match between System and the Real World

The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon.



Users can quickly understand which stovetop control maps to each heating element.

## 5 Error Prevention

Good error messages are important, but the best designs carefully prevent problems from occurring in the first place.



Guard rails on curvy mountain roads prevent drivers from falling off cliffs.

## 8 Aesthetic and Minimalist Design

Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface competes with the relevant units of information.



A minimalist three-legged stool is still a place to sit.

Nielsen Norman Group

## Jakob's Ten Usability Heuristics

## 3 User Control and Freedom

Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action.



Just like physical spaces, digital spaces need quick "emergency" exits too.

## 4 Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing.

*Follow platform conventions.*

Check-in counters are usually located at the front of hotels, which meets expectations.

## 6 Recognition Rather Than Recall

Minimize the user's memory load by making elements, actions, and options visible. Avoid making users remember information.



People are likely to correctly answer "Is Lisbon the capital of Portugal?"

## 7 Flexibility and Efficiency of Use

Shortcuts – hidden from novice users – may speed up the interaction for the expert user.

Regular routes are listed on maps, but locals with more knowledge of the area can take shortcuts.

## 9 Recognize, Diagnose, and Recover from Errors

Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.

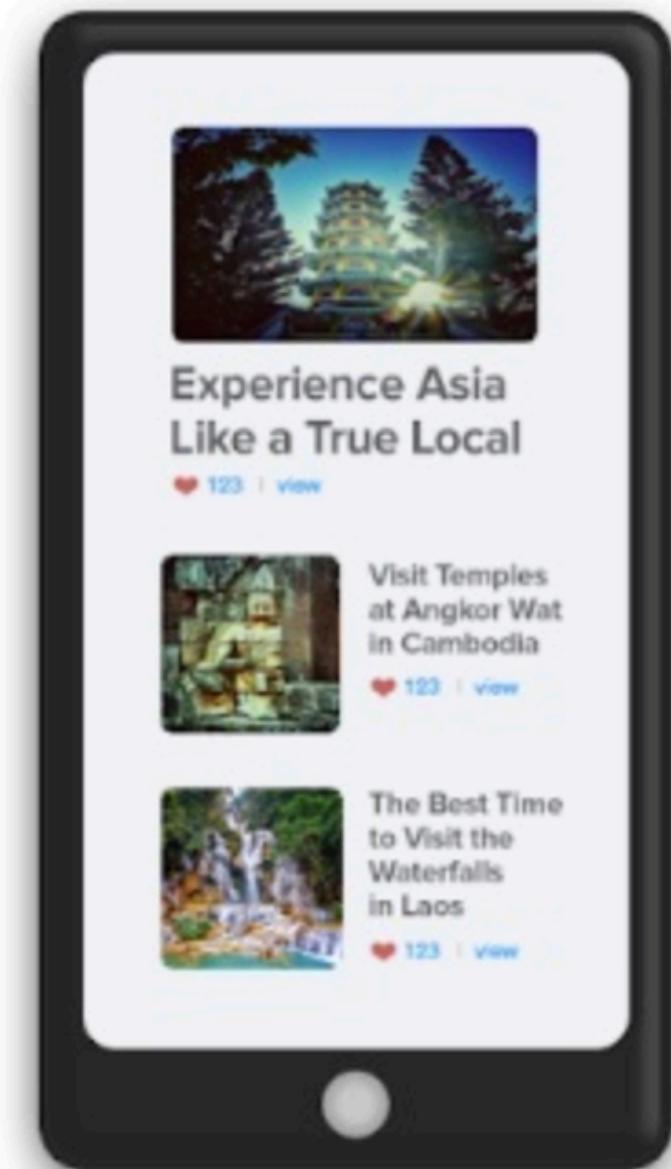
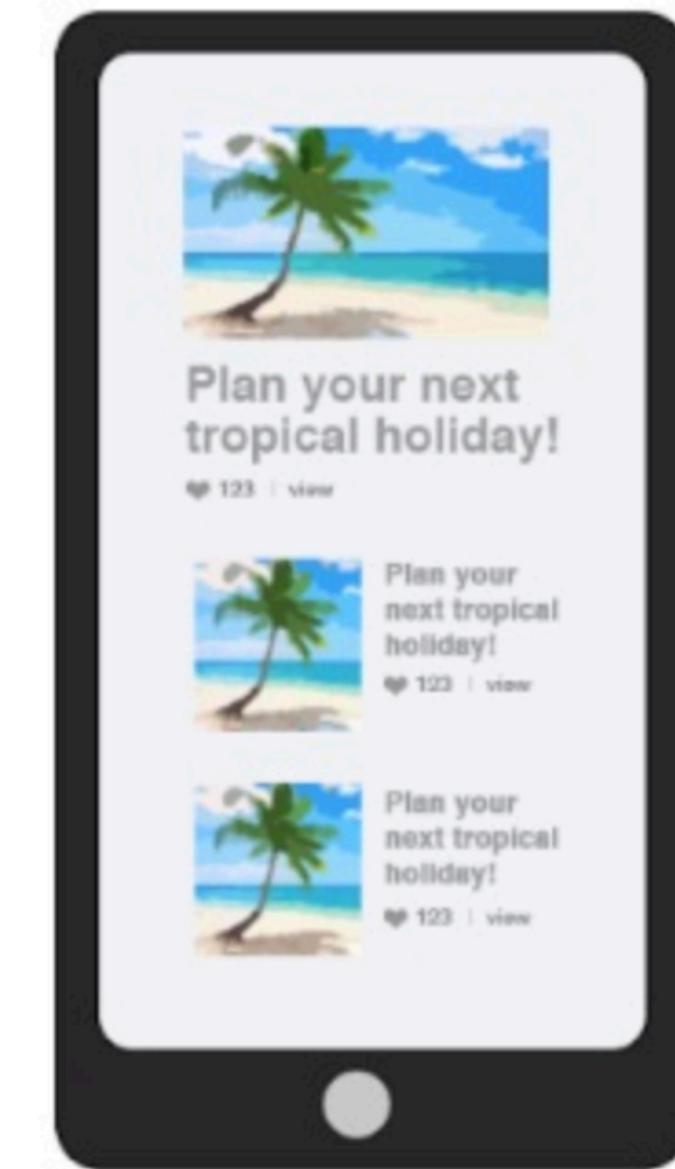
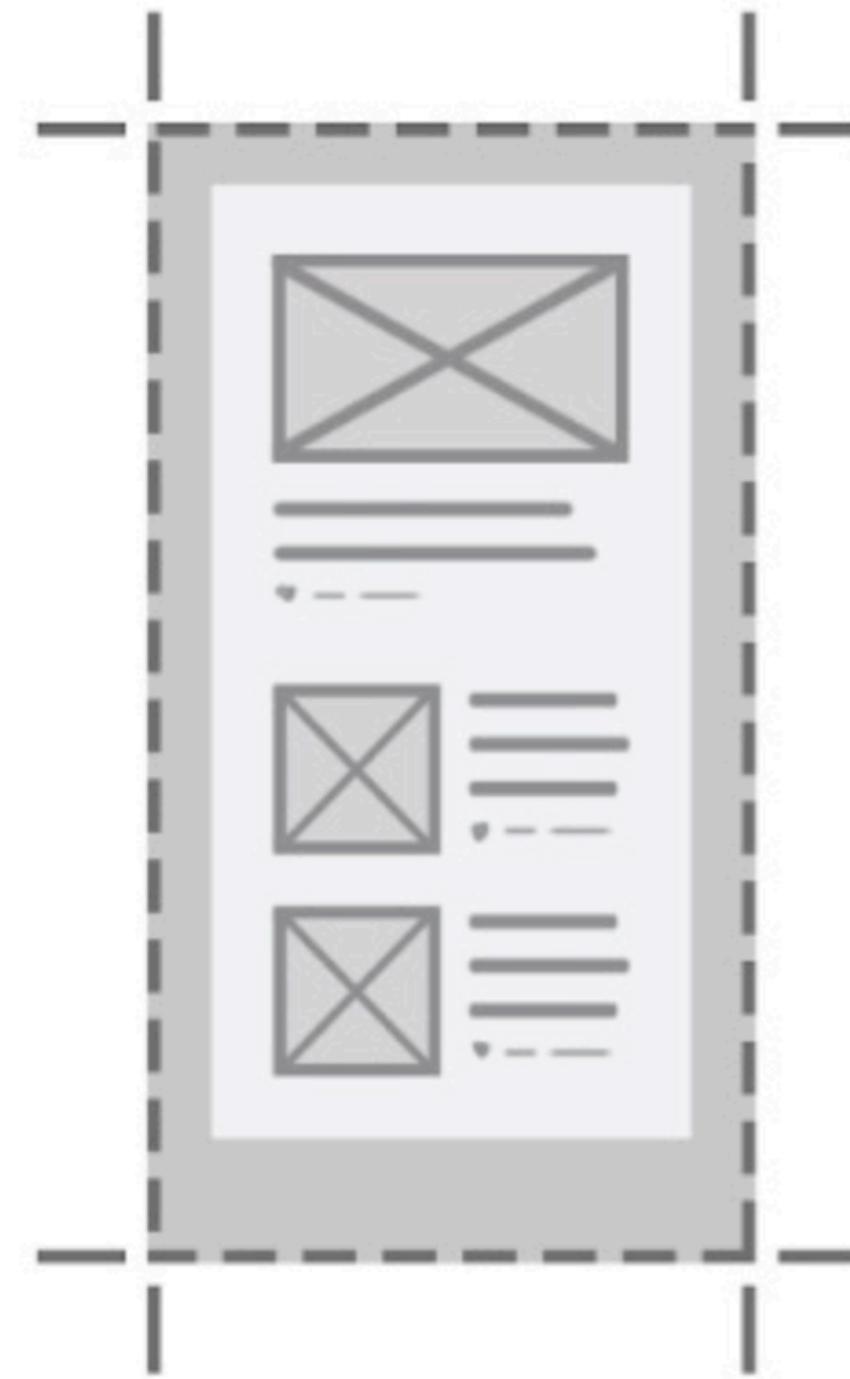
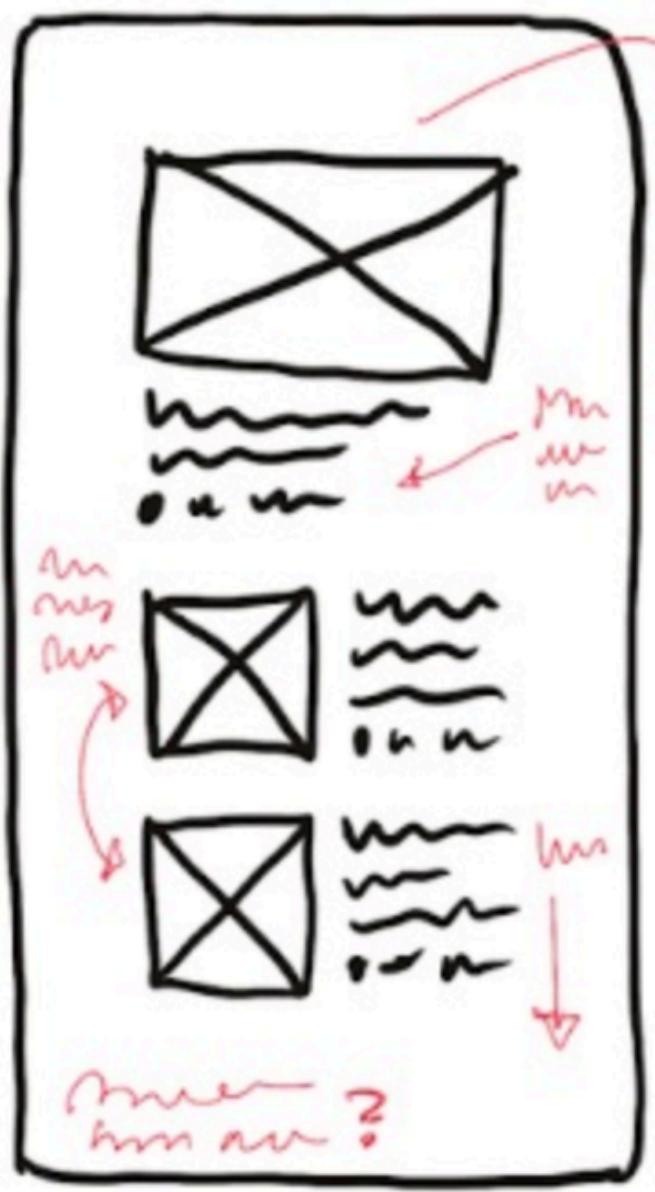


Wrong-way signs on the road remind drivers that they are heading in the wrong direction.

## 10 Help and Documentation

It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users complete their tasks.

Information kiosks at airports are easily recognizable and solve customers' problems in context and immediately.



**SKETCHES** → **WIREFRAMES** → **MOCK-UPS** → **PROTOTYPE**

DRAWING  
PEN AND PAPER

MORE VISUAL  
PEN AND PAPER, XD,  
FIGMA

GRAPHIC DESIGN  
PS, XD, FIGMA

INTERACTIVE  
XD, FIGMA,  
CODE



Windows Media Player



The file wmploc.dll has a version number of 11.0.6001.7000 where  
11.0.6002.16489 was expected.

Windows Media Player is not installed properly and must be reinstalled.

Do you want to install the Player from the Microsoft Web site?

Yes

No

**Windows**

A fatal exception 0E has occurred at 0028:C0011E36 in UXD UMM(01) +  
00010E36. The current application will be terminated.

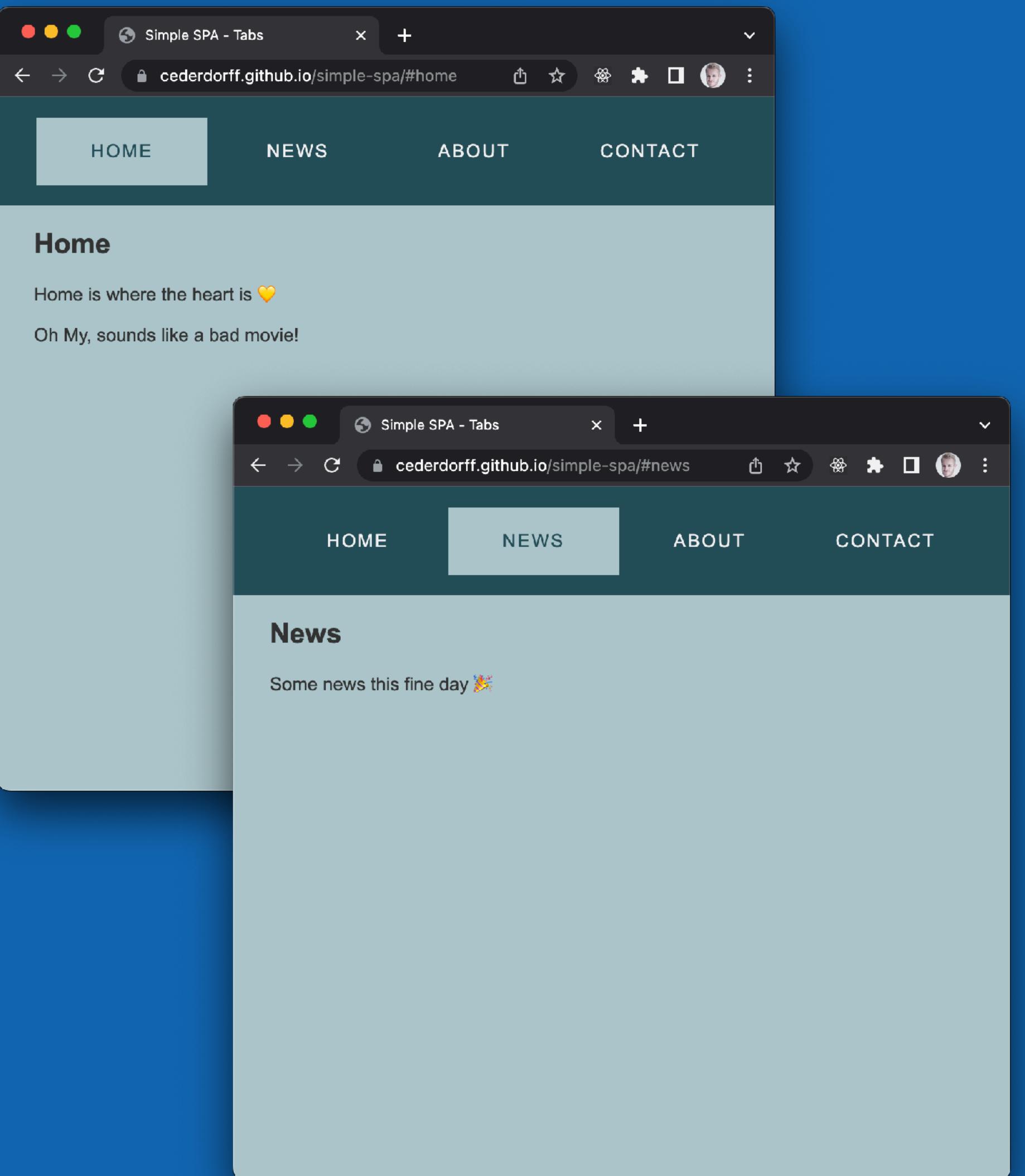
- \* Press any key to terminate the current application.
- \* Press CTRL+ALT+DEL again to restart your computer. You will lose any unsaved information in all applications.

Press any key to continue \_



# Brugergrænseflade

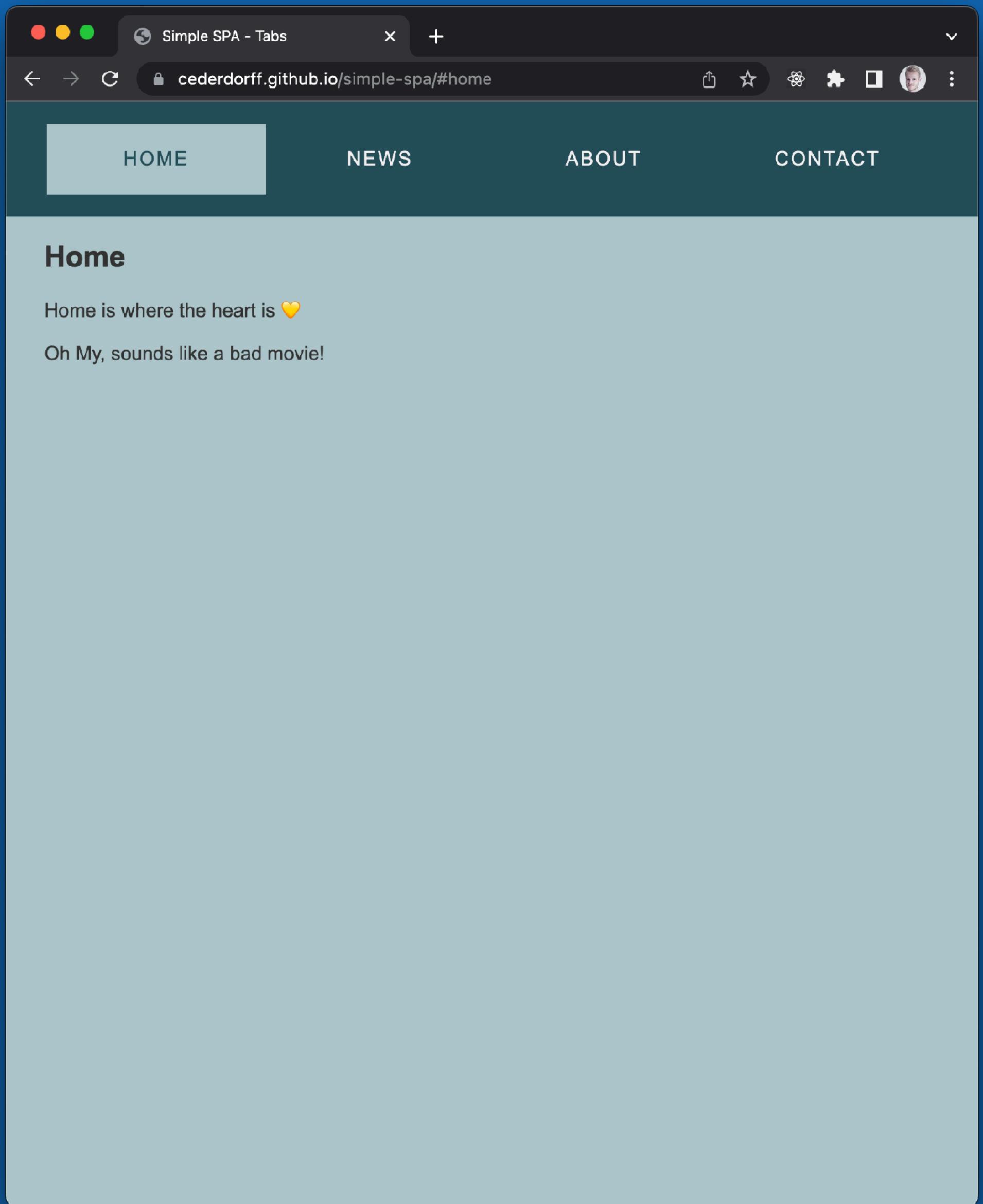
- Flere skærbilleder (“views”)
- Dialogs og modaler
  - muligvis 🤔
- Flere HTML-filer eller tabs (“views”) i samme .html-fil



# Simpel SPA

## Simple Page Application

- En index.html.
- Flere “views” i samme HTML-fil.
- JavaScript der styrer hvilket “view”, der er aktivt.
- Demo af [cederdorff.github.io/simple-spa](https://cederdorff.github.io/simple-spa)



[cederdorff.github.io/simple-spa/](https://cederdorff.github.io/simple-spa/)

# Simpel SPA

## Simple Page Application

- En index.html som holder flere “views” - sections. Kun et er synligt ad gangen.
- De forskellige views (sections) har et id.
- Links har href, der matcher id på det view, der skal vises.

```
<body>
  <!-- Nav Bar with custom links -->
  <!-- The href must match with an id of a view (section) -->
  <nav>
    <a href="#home" class="view-link">Home</a>
    <a href="#news" class="view-link">News</a>
    <a href="#about" class="view-link">About</a>
    <a href="#contact" class="view-link">Contact</a>
  </nav>

  <!-- Views -->
  <!-- All views (sections) must have an id -->
  <main>
    <!-- Home View -->
    <!-- Route: #home -->
    <section id="home" class="view-content">
      <h2>Home</h2>
      <p>Home is where the heart is 🌹</p>
      <p>Oh My, sounds like a bad movie!</p>
    </section>

    <!-- News View -->
    <!-- Route: #news -->
    <section id="news" class="view-content">
      <h2>News</h2>
      <p>Some news this fine day 🎉</p>
    </section>

    <!-- About View -->
    <!-- Route: #about -->
    <section id="about" class="view-content">
      <h2>About</h2>
      <p>Who we are and what we do 🤝</p>
    </section>

    <!-- Contact View -->
    <!-- Route: #contact -->
    <section id="contact" class="view-content">
      <h2>Contact</h2>
      <p>Get in touch, or swing by for a cup of coffee ☕</p>
      <p>I'm just one arm away 🤝</p>
    </section>
```

# Simpel SPA

## Simple Page Application

- JavaScript afgører på baggrund af “hash URL”:
  - hvilket view, der skal vises
  - Hvilket link, der skal være aktivt.

```
function initViews() {
  window.addEventListener("hashchange", viewChange); // whenever the
  viewChange(); // by default, call viewChange to display the first v
}

function viewChange() {
  let hashLink = "#home"; // default view

  if (location.hash) {
    // if there's a hash value, use as link
    hashLink = location.hash;
  }

  hideAllViews(); // hide all views

  document.querySelector(hashLink).classList.add("active"); // add .acti
  setActiveLink(hashLink); // set active link in nav bar
}

function setActiveLink(view) {
  const link = document.querySelector(`a.view-link[href="${view}"]`);
  if (link) {
    link.classList.add("active"); // add .active to active link in
  }
}

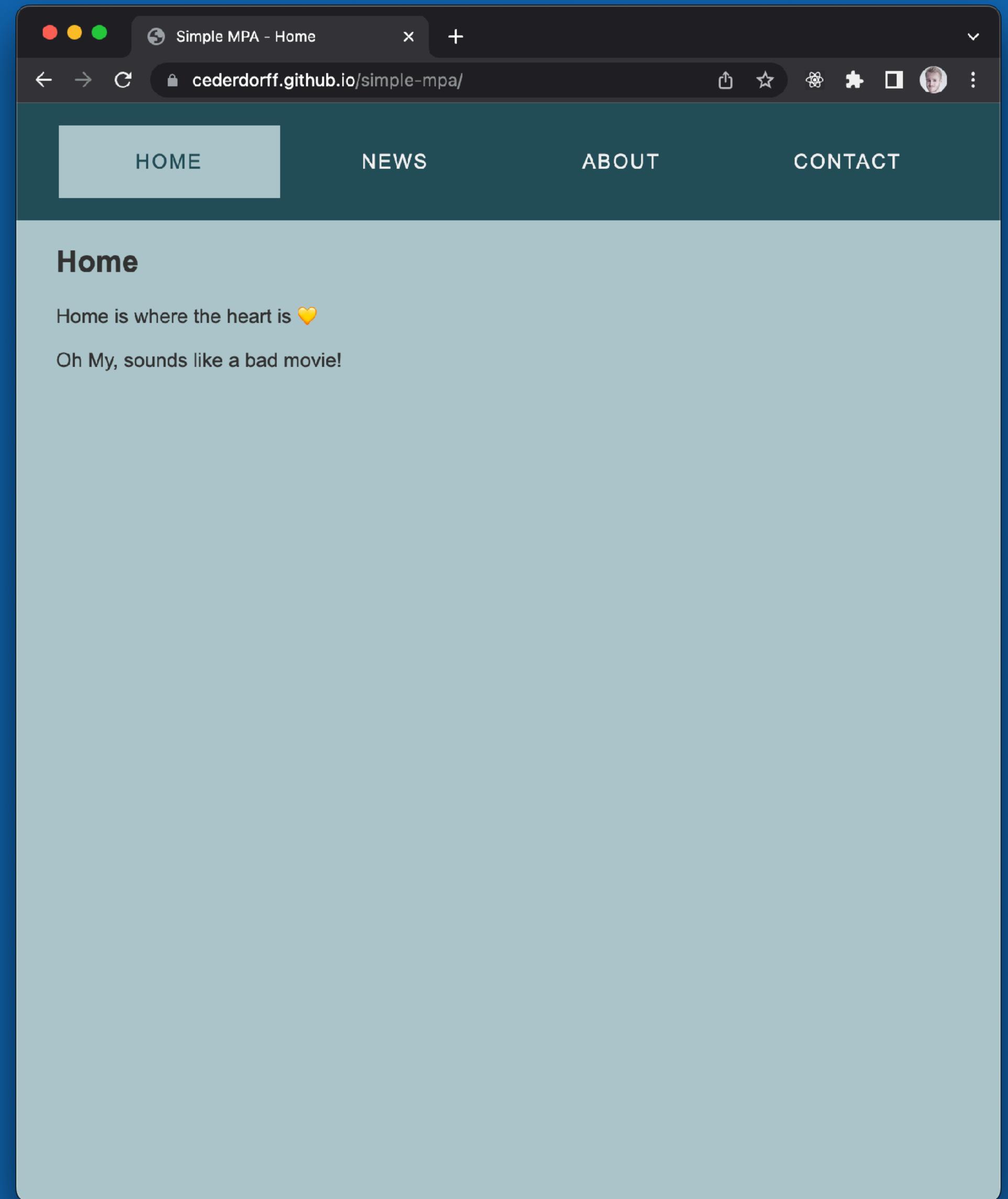
function hideAllViews() {
  // remove .active for all .view-content elements (all views) and .v
  document.querySelectorAll(".view-content").forEach(link => link.clas
  document.querySelectorAll(".view-link").forEach(link => link.classList.r
}

export { initViews };
```

# Simpel MPA

## Multi Page Application

- Flere .html-filer - og gerne en index.html
- Et “view” i én HTML-fil.
- Du kan loade samme .js-fil i alle .html-filer. Du kan også loade en særskilt .js-fil.
- Demo af [cederdorff.github.io/simple-mpa/](https://cederdorff.github.io/simple-mpa/)



# Simpel MPA

## Multi Page Application

- index.html, news.html, about.html og contact.html
- Indhold defineres i hver .html-fil.
- Det aktive link sættes i hver .html-fil.

The screenshot shows a code editor window titled "about.html — simple-mpa". The left sidebar displays the project structure under "SIMPLE-MPA", including files like .gitattributes, about.html, app.css, app.js, contact.html, index.html, news.html, and README.md. The right pane shows the source code for "about.html". The code includes an HTML header with meta tags for charset, http-equiv, and viewport, followed by a title "Simple MPA - About". The body contains a navigation bar (`<nav>`) with links to index.html, news.html, about.html (which is highlighted with the class "active"), and contact.html. Below the nav bar is a main content area (`<main>`) containing an h2 tag and a paragraph. A script tag (`<script src="app.js" type="module"></script>`) is also present. The status bar at the bottom indicates the code is 15 lines long, 23 columns wide, using spaces for indentation, and is saved in UTF-8 format with LF line endings. It also shows the port is 5500 and Prettier is active.

```
about.html — simple-mpa
EXPLORER ... index.html news.html contact.html about.html
SIMPLE-MPA .gitattributes about.html app.css app.js contact.html index.html news.html README.md
about.html — simple-mpa
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=>
7   <meta name="viewport" content="width=device-wid=
8   <link rel="stylesheet" href="app.css">
9   <title>Simple MPA - About</title>
10 </head>
11
12 <body>
13   <!-- Nav Bar with links -->
14   <nav>
15     <a href="index.html">Home</a>
16     <a href="news.html">News</a>
17     <a href="about.html" class="active">About</a>
18     <a href="contact.html">Contact</a>
19   </nav>
20
21   <main>
22     <h2>About</h2>
23     <p>Who we are and what we do 🚀</p>
24   </main>
25   <script src="app.js" type="module"></script>
26 </body>
27
28 </html>

Ln 15, Col 23  Spaces: 4  UTF-8  LF  HTML  Port : 5500  ✓ Prettier
```

# Simpel MPA

## Multi Page Application

- Du kan loade samme JavaScript i alle .html-filer. Derved køres samme JavaScript på alle sider.
- Du kan også loade en særskilt .js-fil i .html-filer. Derved kan du have forskelligt JavaScript på dine sider.
- Og typisk vil du have brug for begge dele.

The screenshot shows a code editor window titled "app.js — simple-mpa". The left sidebar displays a file tree for a project named "SIMPLE-MPA" containing files like ".gitattributes", "about.html", "# app.css", "app.js" (selected), "contact.html", "index.html", "news.html", and "README.md". The main editor area shows the contents of "app.js":

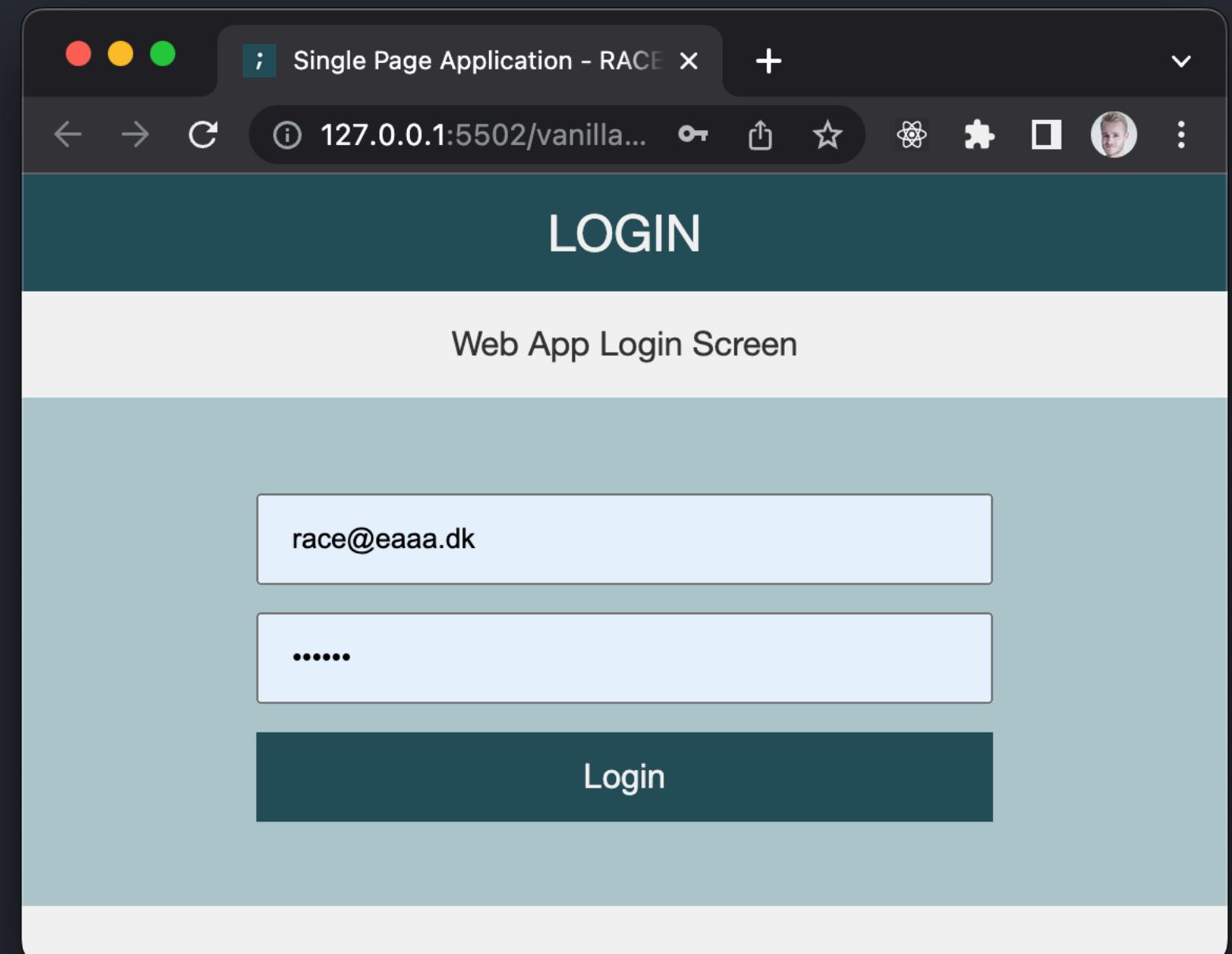
```
1 "use strict";
2
3 window.addEventListener("load", initApp);
4
5 function initApp() {
6     console.log("app.js is running 🎉");
7 }
8
```

The status bar at the bottom indicates the file is "main", has 0 changes, and is using "JavaScript". It also shows the current line (Ln 8, Col 1), spaces (Spaces: 4), encoding (UTF-8 LF), and port (Port : 5500). The Prettier icon is also visible.

# Simple Login

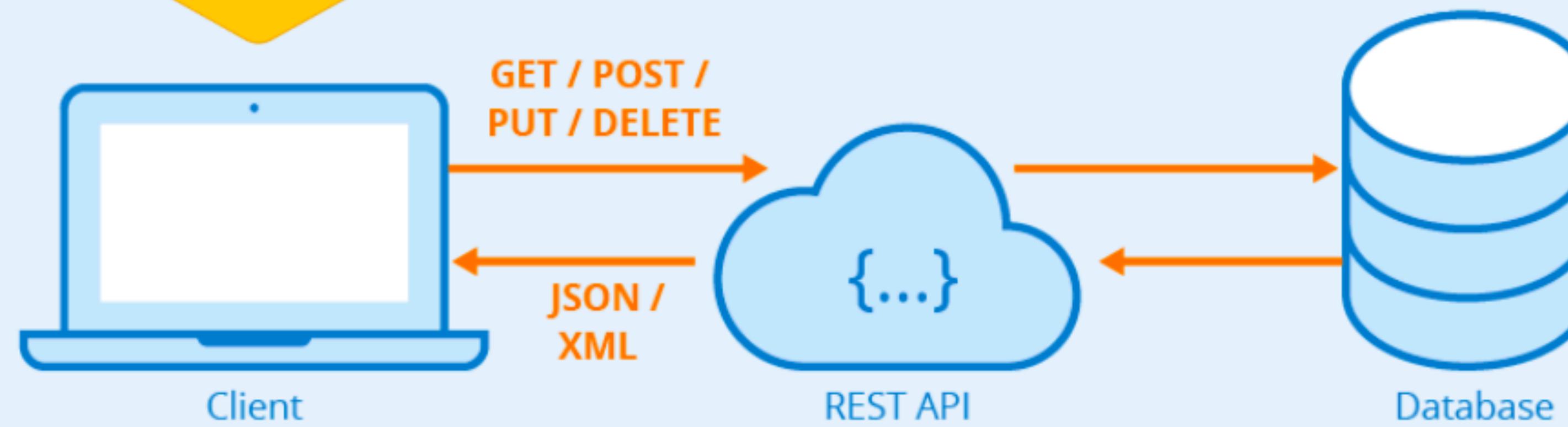
Gør det “dumt” og simpelt

- Ja, simpelt.
- Fx en if sætning, der kontrollerer om brugeroplysningerne er korrekte.
- Evt flere hard-coded brugere.





# Firebase



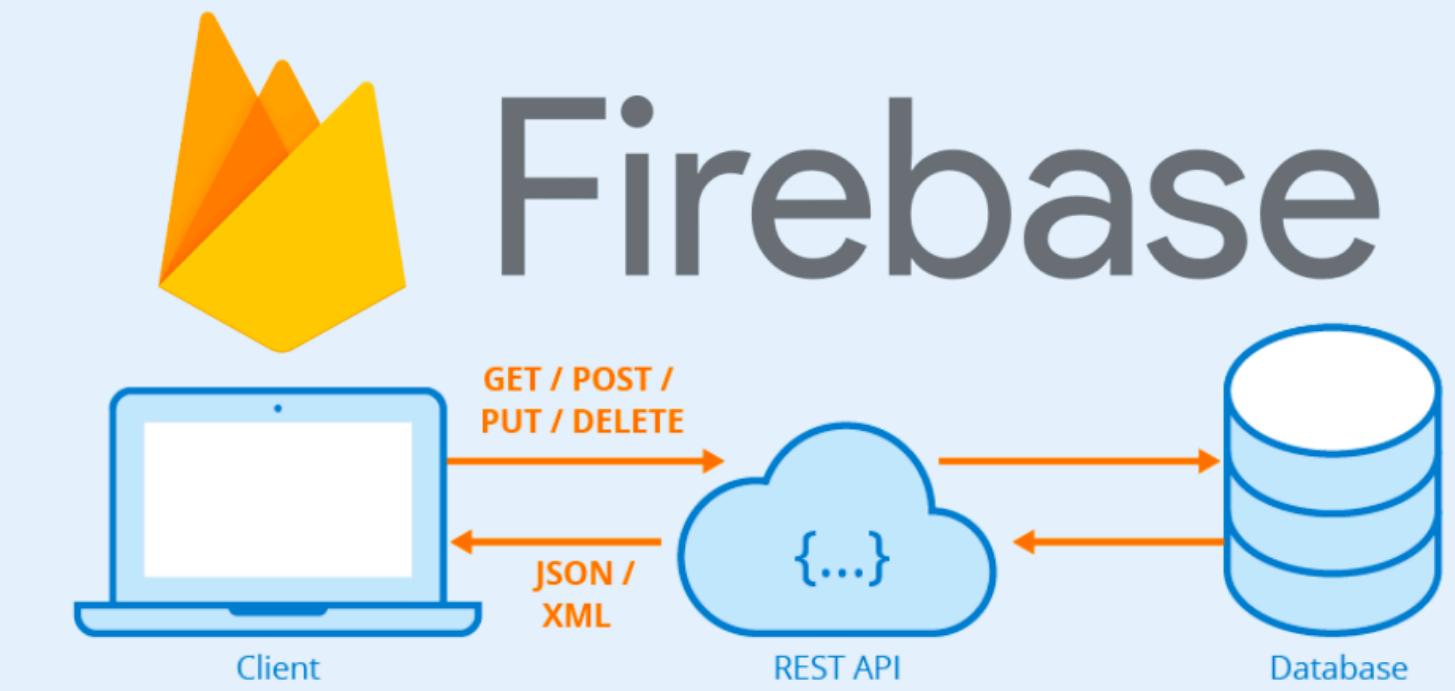
“

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client.

We can use any Firebase Realtime Database URL as a REST endpoint. All we need to do is append .json to the end of the URL and send a request from our favorite HTTPS client.

”

<https://firebase.google.com/docs/database/rest/start>



# How data is structured

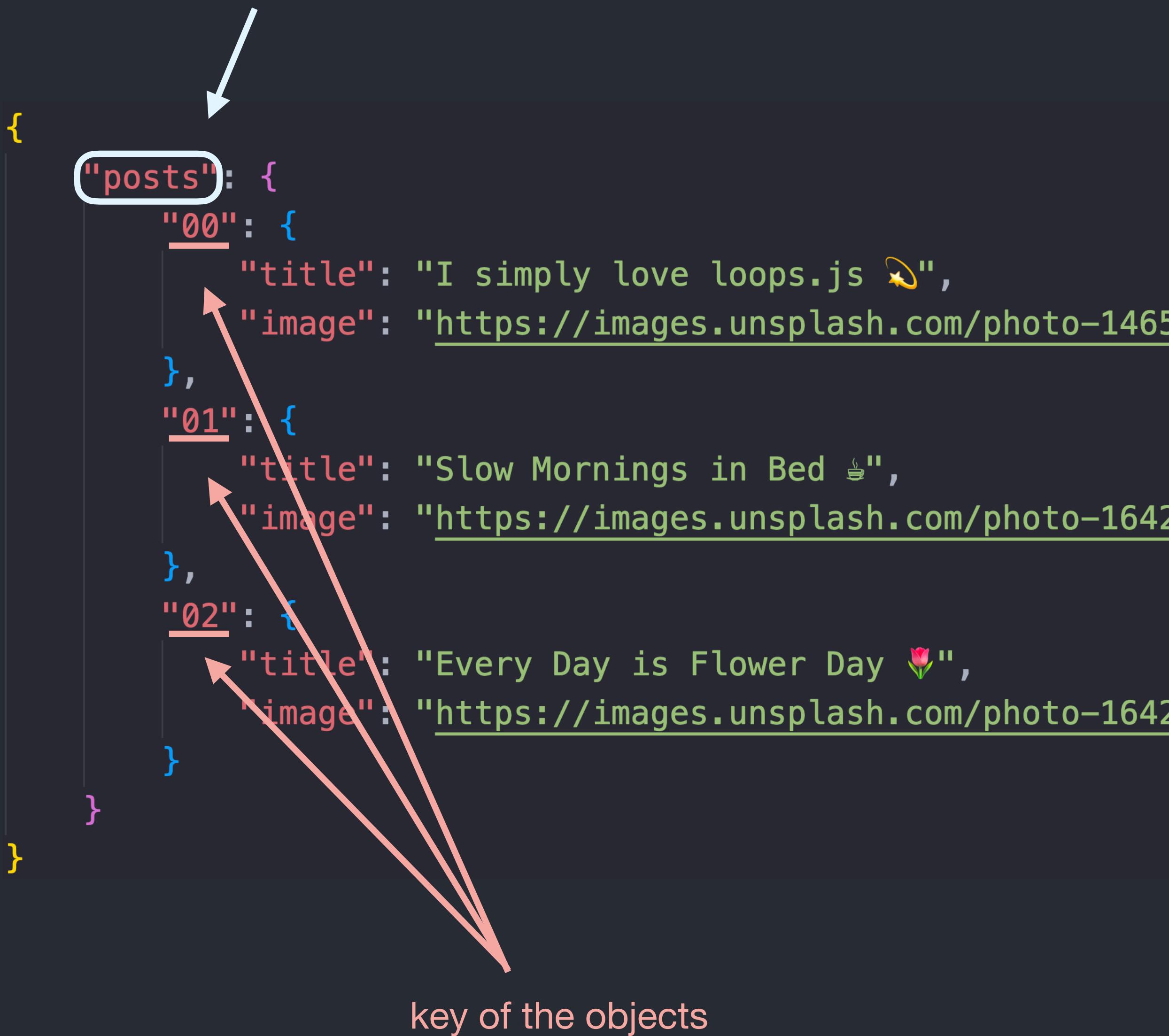
```
{  
  "posts": {  
    "00": {  
      "title": "I simply love loops.js 🌟",  
      "image": "https://images.unsplash.com/photo-1465...  
    },  
    "01": {  
      "title": "Slow Mornings in Bed ☕",  
      "image": "https://images.unsplash.com/photo-1642...  
    },  
    "02": {  
      "title": "Every Day is Flower Day 🌸",  
      "image": "https://images.unsplash.com/photo-1642...  
    }  
  }  
}
```

- One big object with objects.
- All objects have a unique key
- “posts”, “00”, “01” and “02” are all keys.
- “posts” is the key of the collection of post objects (list of posts).
- “00”, “01” and “02” are keys of a post object.
- “title” and “image” are keys inside of every post object. “title” and “image” contains values.
- Remember a property contains a key and a value.

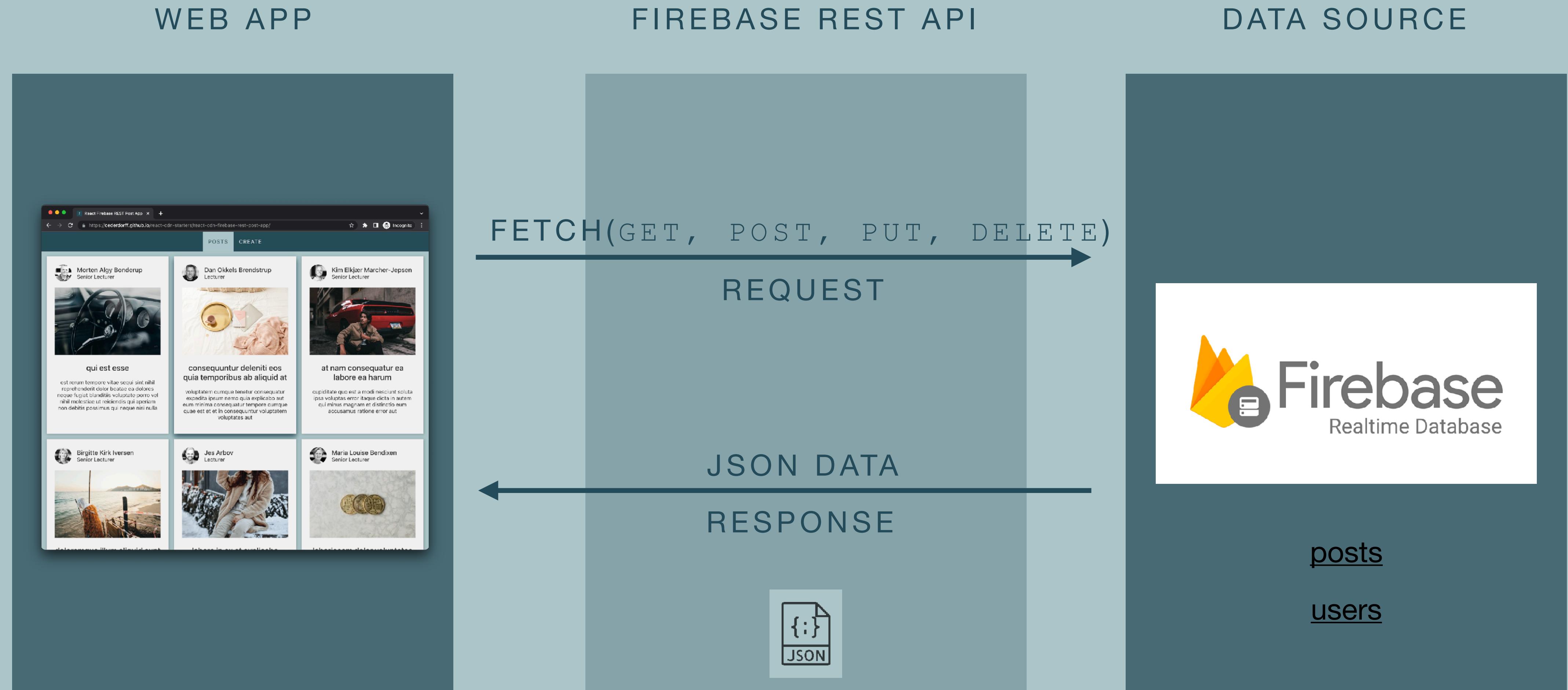
```
{  
  "posts": {  
    "00": {  
      "title": "I simply love loops.js 🌟",  
      "image": "https://images.unsplash.com/photo-1465...  
    },  
    "01": {  
      "title": "Slow Mornings in Bed ☕",  
      "image": "https://images.unsplash.com/photo-1642...  
    },  
    "02": {  
      "title": "Every Day is Flower Day 🌸",  
      "image": "https://images.unsplash.com/photo-1642...  
    }  
  }  
}
```

- Just think of an array of objects (posts array with post objects).
- We are just using an object to hold all the objects instead of an array.
- It's kind of a dictionary. The posts object contains unique prop names (keys) which easily can be "looked up".
- Then every post object (with title and image props) can be returned:
  - `["posts"]["02"]`
  - `["posts"]["00"]`
  - `["posts"]["01"]`
  - `["posts"]["01"]["title"]`

key and name of the collections



# Fetch, HTTP Request & Response



# How to combine data collections (lists)

Post App

127.0.0.1:5501/index.html

## Post App with Firebase REST API

CREATE NEW POST

Sort by: Not selected

Search: Search by title

Morten Algy Bonderup  
Senior Lecturer

Qui est esse  
Est rerum tempore vitae sequi sint nihil  
reprehenderit dolor beatae ea dolores  
neque fugiat blanditiis voluptate porro  
vel nihil molestiae ut reiciendis qui  
aperiam non debitis possimus qui neque  
nisi nulla

DELETE UPDATE

Dan Okkels Brendstrup  
Lecturer

Consequuntur deleniti eos quia  
temporibus ab aliquid at  
Voluptatem cumque tenetur consequatur  
expedita ipsum nemo quia explicaboaut  
eum minima consequaturtempore  
cumque quae est etet in consequuntur  
voluptatem voluptates aut

DELETE UPDATE

posts and  
users are stored  
in two different  
collections (lists)

Post App with Firebase REST API

CREATE NEW POST

Sort by: Not selected

Search: Search by title

Morten Algy Bonderup  
Senior Lecturer

Qui est esse  
Est rerum tempore vitae sequi sint nihil  
reprehenderit dolor beatae ea dolores  
neque fugiat blanditiis voluptate porro  
vel nihil molestiae ut reiciendis qui  
aperiam non debitis possimus qui neque  
nisi nulla

DELETE UPDATE

Dan Okkels Brendstrup  
Lecturer

Consequuntur deleniti eos quia  
temporibus ab aliquid at  
Voluptatem cumque tenetur consequatur  
expedita ipsum nemo quia explicaboaut  
eum minima consequaturtempore  
cumque quae est etet in consequuntur  
voluptatem voluptates aut

DELETE UPDATE

All posts have  
a uid prop with  
the value of  
one user's id.

```
{
  "posts": {
    "1mNVXNbZWjKsiaTFsDw": {
      "title": "qui est esse",
      "uid": "GFCogUyrdizw6dD221Qp",
      "image": "https://images.unsplash.com/photo-1642049888276-9c9f0a1a8758?crop=er",
      "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae"
    },
    "5tl4jHHSRaKEB0UW9nQd": {
      "title": "consequuntur deleniti eos quia temporibus ab aliquid at",
      "uid": "HlvRHR58C05gu0Ll64k5",
      "body": "voluptatem cumque tenetur consequatur expedita ipsum nemo quia explic"
      "image": "https://images.unsplash.com/photo-1642013126529-f674c5b87dbe?crop=er"
    },
    "B3xo03oPgy0fpr3i3x9D": {
      "title": "laboriosam dolor voluptates",
      "image": "https://images.unsplash.com/photo-1638913658179-18c9a9c943f7?crop=er",
      "uid": "GFCogUyrdizw6dD221Qp",
      "body": "doloremque ex facilis sit sint culpa\\nsoluta assumenda eligendi non u"
    }
  },
  "users": {
    "GFCogUyrdizw6dD221Qp": {
      "name": "Morten Algy Bonderup",
      "title": "Senior Lecturer",
      "phone": "72286339",
      "image": "https://www.baaa.dk/media/hi4lv5hw/morten-bonderup.jpg?anchor=center"
      "mail": "moab@aaaa.dk"
    },
    "HlvRHR58C05gu0Ll64k5": {
      "image": "https://www.eaaa.dk/media/bdojel41/dan-okkels-brendstrup.jpg?anchor=",
      "title": "Lecturer",
      "mail": "dob@aaaa.dk",
      "name": "Dan Okkels Brendstrup",
      "phone": "72286327"
    }
  }
}
```

One user can  
create  
multiple posts

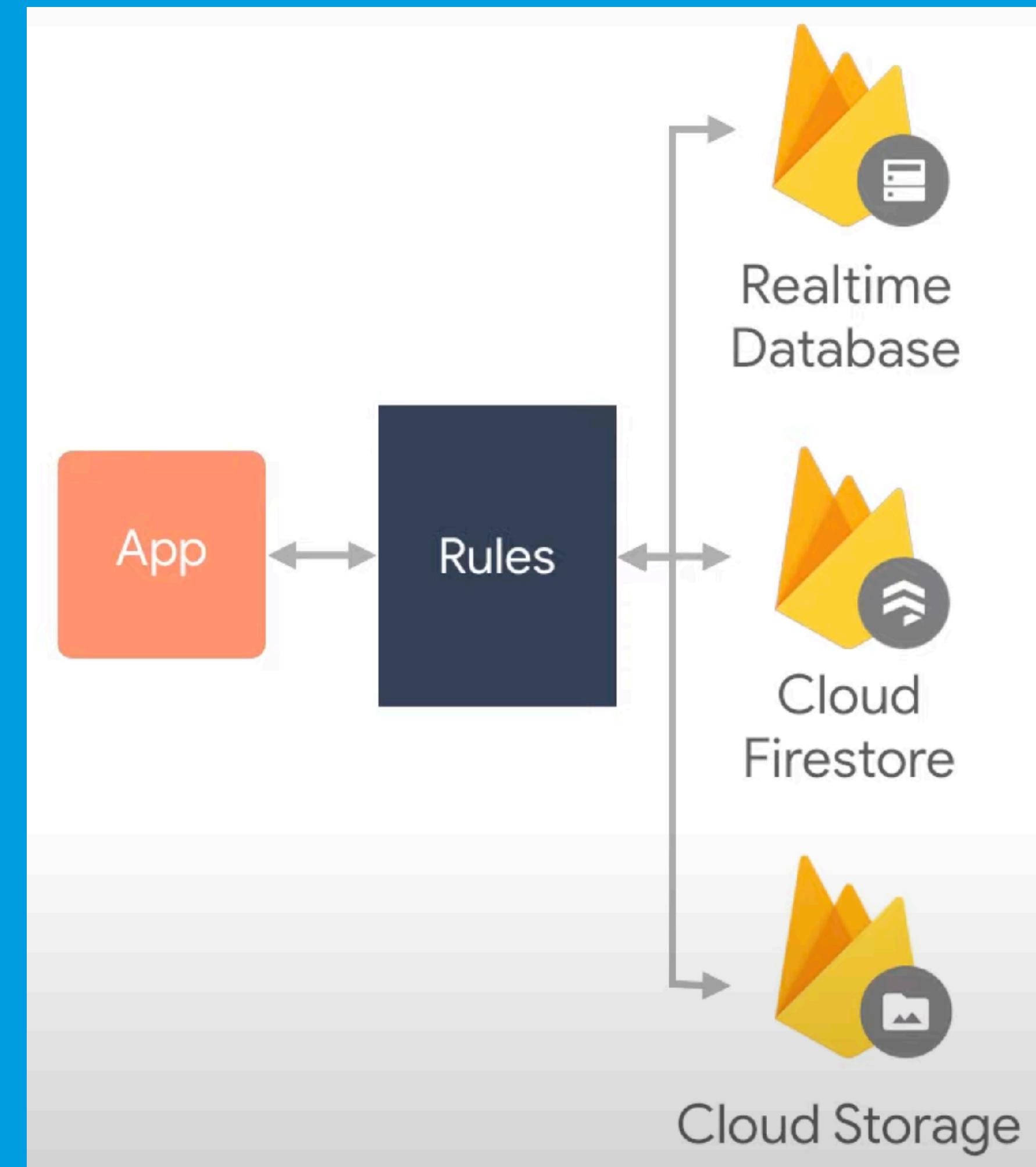
```
{  
  "posts": {  
    "1mNVXMNbZWjKsiaTFsDw": {  
      "title": "qui est esse",  
      "uid": "GFCogUyrdizw6dD221Qp",  
      "image": "https://images.unsplash.com/photo-1642049888276-9c9f0a1a8758?crop=er",  
      "body": "est rerum tempore vitae\\nsequi sint nihil reprehenderit dolor beatae  
    },  
    "5tl4jHHSRaKEB0UW9nQd": {  
      "title": "consequuntur deleniti eos quia temporibus ab aliquid at",  
      "uid": "HlvRHR58C05gu0Ll64k5",  
      "body": "voluptatem cumque tenetur consequatur expedita ipsum nemo quia explic",  
      "image": "https://images.unsplash.com/photo-1642013126529-f674c5b87dbe?crop=er",  
    },  
    "B3xo03oPgy0fpr3i3x9D": {  
      "title": "laboriosam dolor voluptates",  
      "image": "https://images.unsplash.com/photo-1638913658179-18c9a9c943f7?crop=er",  
      "uid": "GFCogUyrdizw6dD221Qp",  
      "body": "doloremque ex facilis sit sint culpa\\nsoluta assumenda eligendi non u",  
    },  
  },  
  "users": {  
    "GFCogUyrdizw6dD221Qp": {  
      "name": "Morten Algy Bonderup",  
      "title": "Senior Lecturer",  
      "phone": "72286339",  
      "image": "https://www.baaa.dk/media/hi4lv5hw/morten-bonderup.jpg?anchor=center",  
      "mail": "moab@eaaa.dk"  
    },  
    "HlvRHR58C05gu0Ll64k5": {  
      "image": "https://www.eaaa.dk/media/bdojel41/dan-okkels-brendstrup.jpg?anchor=",  
      "title": "Lecturer",  
      "mail": "dob@eaaa.dk",  
      "name": "Dan Okkels Brendstrup",  
      "phone": "72286327"  
    },  
  }  
}
```

# Consider your data structure!

```
{
  "posts": {
    "1mNVXMNbZWjKsiaTFsDw": {
      "title": "qui est esse",
      "uid": "GFCogUyrdizw6dD221Qp",
      "image": "https://images.unsplash.com/photo-1642049888276-9c9f0a1a8758?crop=er",
      "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae"
    },
    "5tl4jHHSRaKEB0UW9nQd": {
      "title": "consequuntur deleniti eos quia temporibus ab aliquid at",
      "uid": "HlvRHR58C05gu0Ll64k5",
      "body": "voluptatem cumque tenetur consequatur expedita ipsum nemo quia explic",
      "image": "https://images.unsplash.com/photo-1642013126529-f674c5b87dbe?crop=er"
    },
    "B3xo03oPgy0fpr3i3x9D": {
      "title": "laboriosam dolor voluptates",
      "image": "https://images.unsplash.com/photo-1638913658179-18c9a9c943f7?crop=er",
      "uid": "GFCogUyrdizw6dD221Qp",
      "body": "doloremque ex facilis sit sint culpa\\nsoluta assumenda eligendi non u"
    }
  },
  "users": {
    "GFCogUyrdizw6dD221Qp": {
      "name": "Morten Algy Bonderup",
      "title": "Senior Lecturer",
      "phone": "72286339",
      "image": "https://www.baaa.dk/media/hi4lv5hw/morten-bonderup.jpg?anchor=center",
      "mail": "moab@eaaa.dk"
    },
    "HlvRHR58C05gu0Ll64k5": {
      "image": "https://www.eaaa.dk/media/bdojel41/dan-okkels-brendstrup.jpg?anchor=",
      "title": "Lecturer",
      "mail": "dob@eaaa.dk",
      "name": "Dan Okkels Brendstrup",
      "phone": "72286327"
    }
  }
}
```



# Firebase Security Rules



# Code Every Day



- Altså, det er bare den eneste måde at lære det på 🤷‍♂️