

Sort

Men også UI for Create, Update, Delete og søgning

Datamatiker

kea
KØBENHAVNS ERHVERVSAKADEMI

Dagens Formål

- Fuld fungerende CRUD Post App
- Fungerende UI med formularer.
- Sortering og filtrering.
- Kobling mellem JS-funktionalitet og UI.

Agenda

- Overblik, semesterflow og grupper
- Post App
 - Create (POST) og UI
 - Delete (DELETE) og UI
 - Update (PUT) og UI
 - Søgning, .filter og UI
 - .sort() og UI

```
main.js
1 "use strict";
2
3 fetch("http://headlesscms.cederdorff.com/wp-json/wp/v2/posts?_embed")
4   .then(function(response) {
5     return response.json();
6   })
7   .then(function(json) {
8     appendPosts(json);
9   });
10
11 function appendPosts(posts) {
12   for (let post of posts) {
13     console.log(post);
14     document.querySelector("#grid-posts").innerHTML += `
15       <article>
16         <h3>${post.title.rendered}</h3>
17         <p>Email: <a href="mailto:${post.acf.email}">${post.acf.email}</a></p>
18         <p>Phone: ${post.acf.phone}</p>
19       </article>
20     `;
21   }
22 }
```

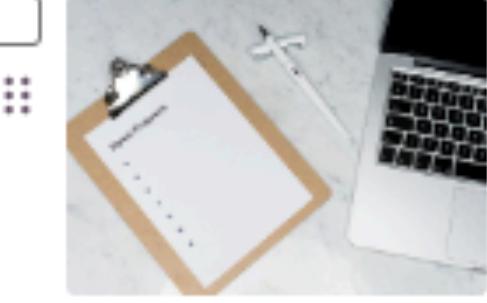
1. SEMESTERS STRUKTUR

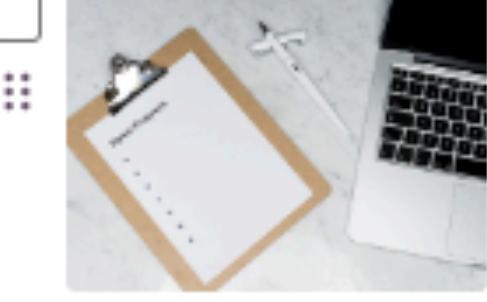
Semesteret består af 16 undervisningsuger (inkl. en masse helligdage)

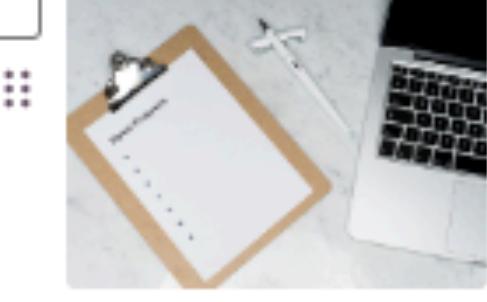


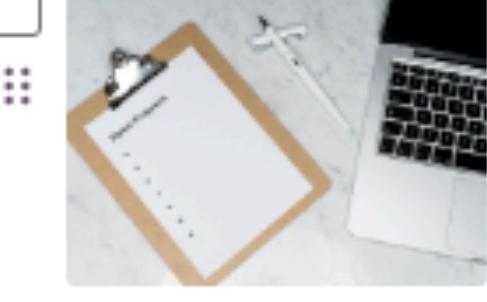
Eksamens dato: 12.-14. juni

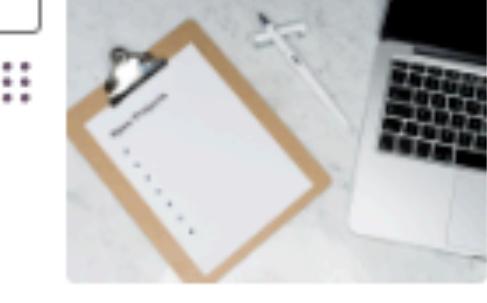
Overblik

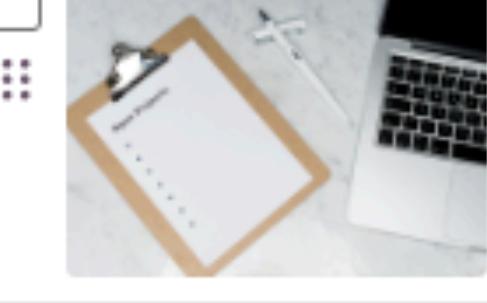
- 

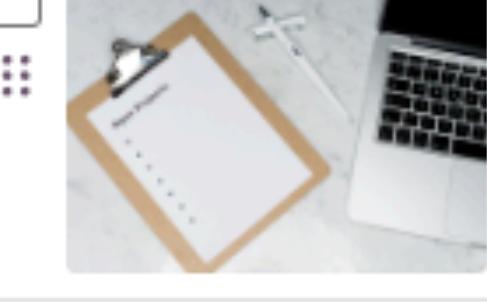
Sort FORMS, REST OG CRUD
Emner Update og UI (form) Delete og UI .sort() og UI Evt. opsamling på Create Form Evt. opsamling på callback-...
25. apr. 1 ressource
- 

Eksamensform og -øvelser FORMS, REST OG CRUD
Vi bruger dagen på at tale om eksamensformen på 1. semester. I vil også få udleveret eksempler på...
26. apr.
- 

Arrow-funktioner FORMS, REST OG CRUD
27. apr.
- 

Git og teamsamarbejde FORMS, REST OG CRUD
1. maj
- 

Moduler, validering og fejlhåndte... FORMS, REST OG ...
2. maj
- 

Mini CRUD projekt FORMS, REST OG CRUD
3. maj
- 

Mini CRUD projekt FORMS, REST OG CRUD
4. maj

Ja, en masse om eksamen 🌟

() => 🎉

27. og 28. april: semestersnak og feedback på Dataprojekt

Mere om Git og samarbejde

Moduler, validering og fejl - og UI

Vejledning - Arbejd på projekt

Vejledning - Arbejd på projekt
Hand In: 04-05-2023 kl. 23:59

Semestersnak og Feedback

27. og 28. april

“Book” en tid [her](#)

A photograph showing a group of five people in an office environment. Four individuals are seated around a light-colored wooden conference table, while one person stands to the left. They appear to be engaged in a collaborative discussion or meeting. The background features office cubicles and a whiteboard with some writing on it.

Og hvad med de der grupper der?

Hvad kunne vi sidst?

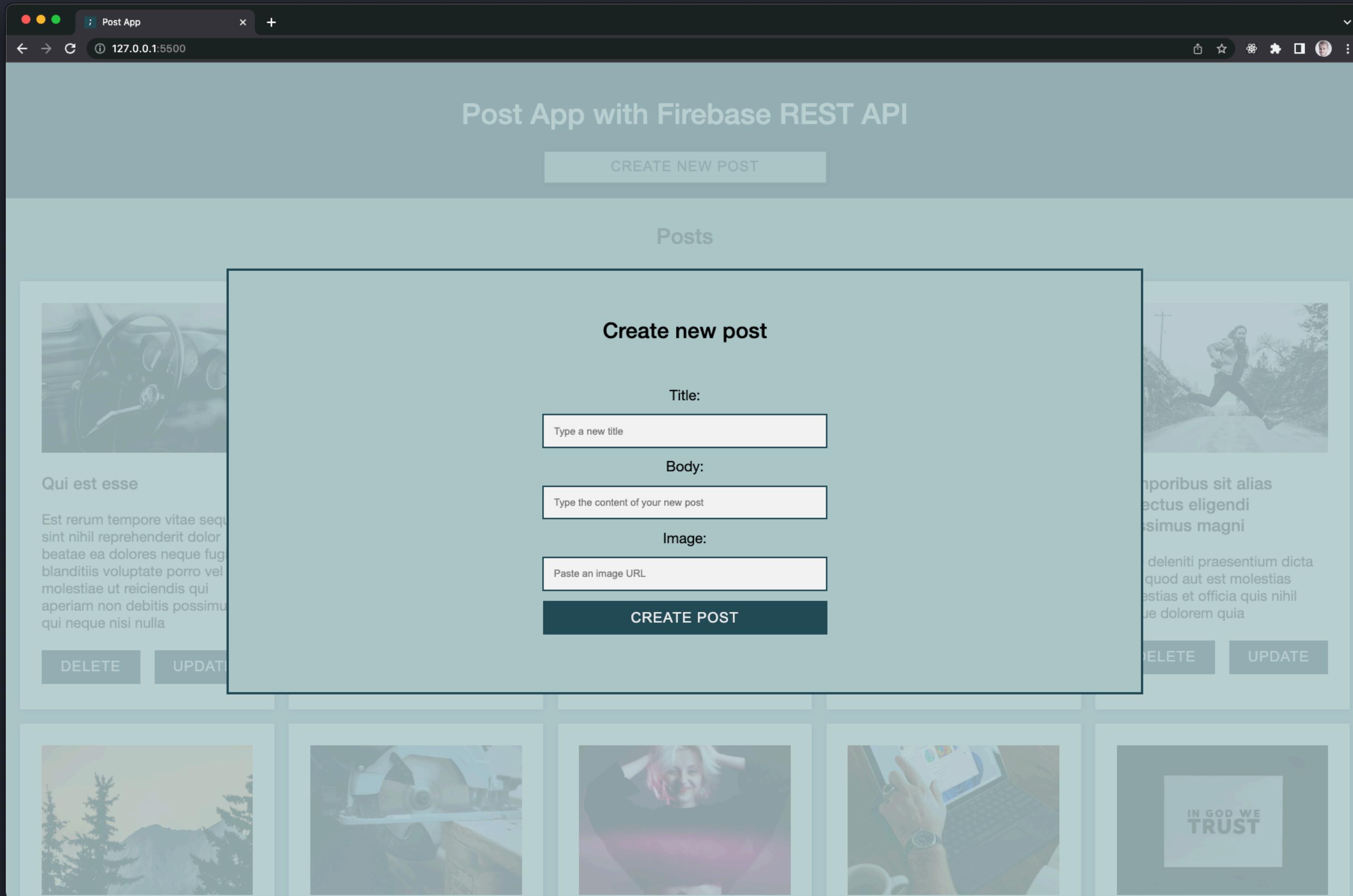
Post App with Firebase REST API

CREATE NEW POST

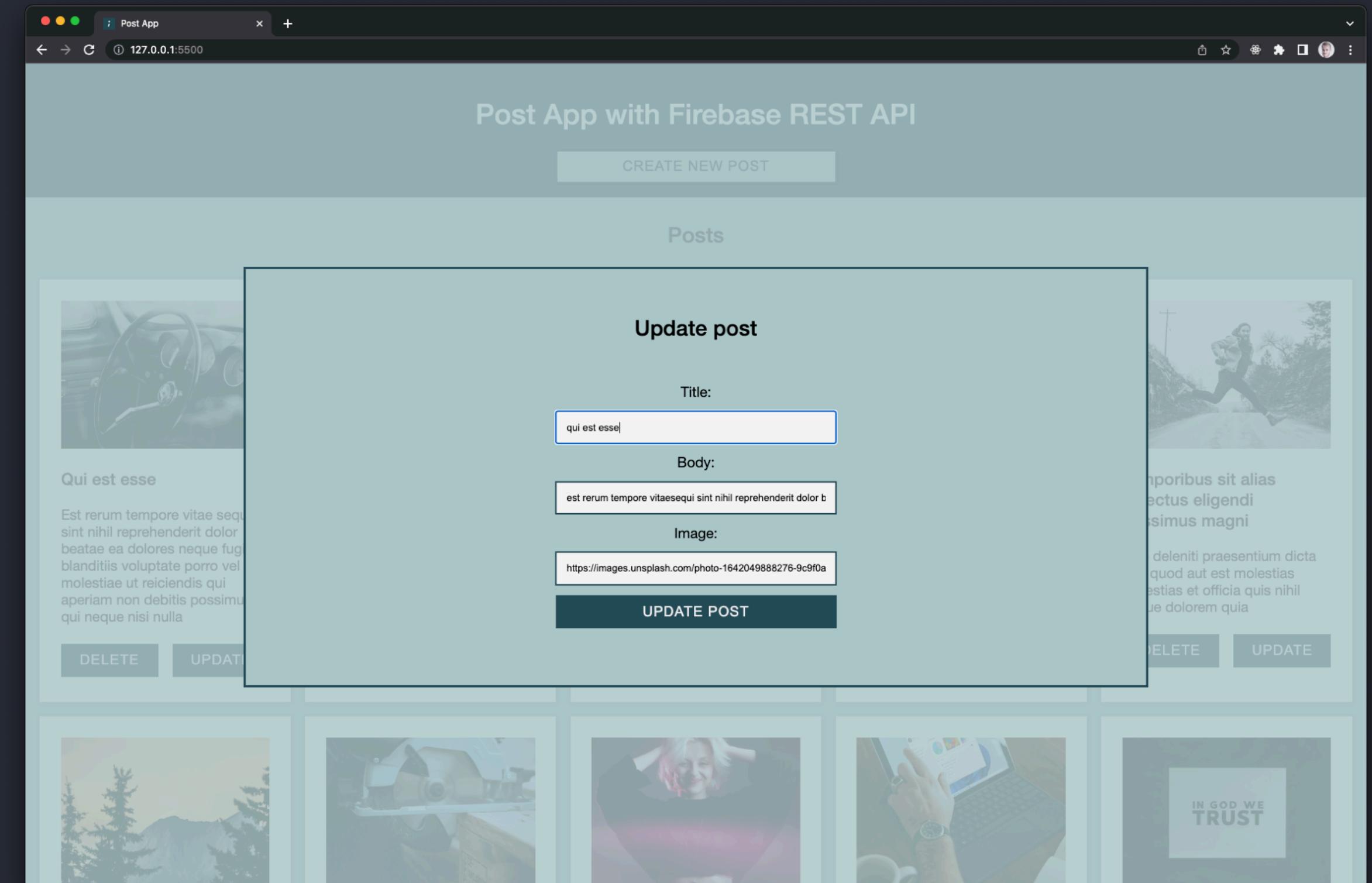
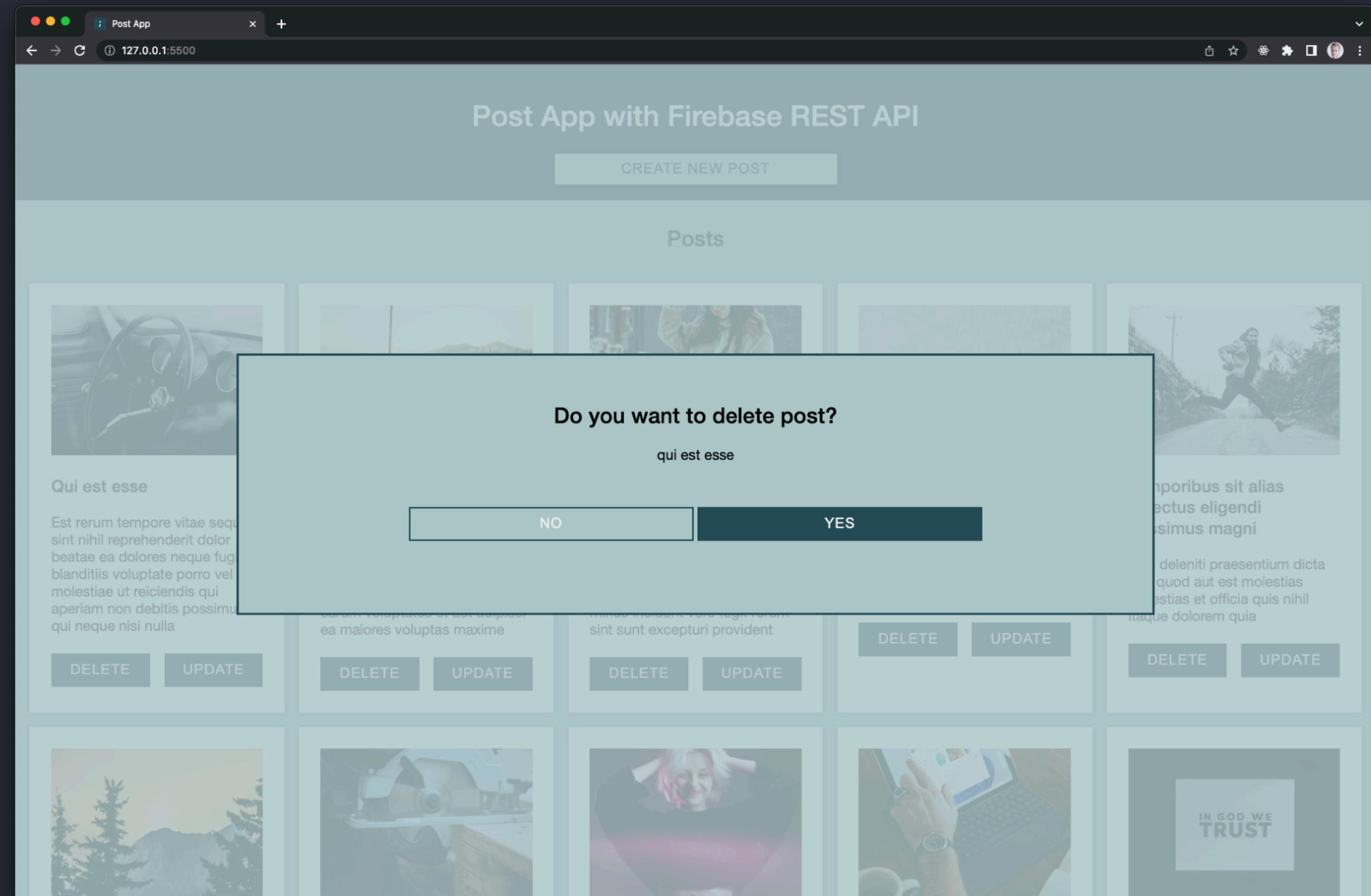
Posts

Image	Title	Content	Actions
	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
	Doloremque illum aliquid sunt	Deserunt eos nobis asperiores et hic est debitis repellat molestiae optio nihil ratione ut eos beatae quibusdam distinctio maiores earum voluptates et aut adipisci ea maiores voluptas maxime	DELETE UPDATE
	Labore in ex et explicabo corporis aut quas	Ex quod dolorem ea eum iure qui provident amet quia qui facere excepturi et repudiandae asperiores molestias provident minus incident vero fugit rerum sint sunt excepturi provident	DELETE UPDATE
	Laboriosam dolor voluptates	Doloremque ex facilis sit sint culpasoluta assumenda eligendi non ut eiussequi ducimus vel quasiveritatis est dolores	DELETE UPDATE
	Temporibus sit alias delectus eligendi possimus magni	Quo deleniti praesentium dicta non quod aut est molestias molestias et officia quis nihil itaque dolorem quia	DELETE UPDATE

Hvad kan du nu?



Hvad mangler du?



Sortering, filtrering, søgning ...

Harry Potter Characters

Filter by house: All Search by name

Image	Name	House	Gender	Year of birth
	Harry Potter	Gryffindor	male	1980
	Hermione Granger	Gryffindor	female	1979
	Ron Weasley	Gryffindor	male	1980
	Draco Malfoy	Slytherin	male	1980

ALL USERS

Order by: Choose here Filter by: All Search

	Diana Riis Myjak Andersen eaadimy@students.eaaa.dk Enrollment: StudentEnrollment UPDATE DELETE
	German Arias Rodriguez eaagar@students.eaaa.dk Enrollment: StudentEnrollment UPDATE DELETE
	Haya Barakat eaahbar@students.eaaa.dk Enrollment: StudentEnrollment UPDATE DELETE

USERS CREATE RANDOM

Canvas Users

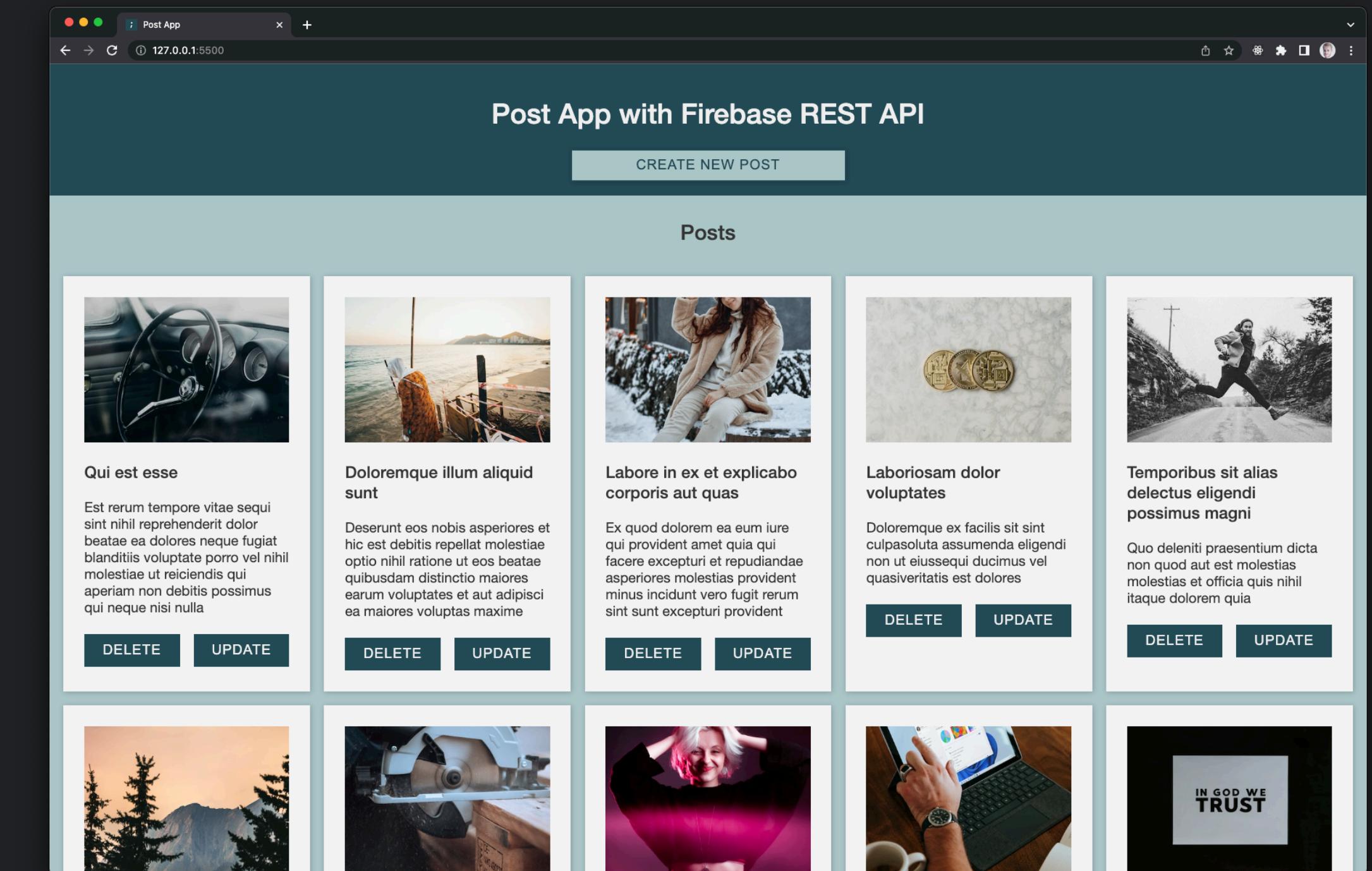
Show Students Show Teachers Search Search by name Sort by First name

	Anders Husted eaaahu@students.eaaa.dk Role: Student
	Anders Lassen Skrydstrup eaaanls@students.eaaa.dk Role: Student
	Aron Ingi Svansson eaaarsv@students.eaaa.dk Role: Student

Post App

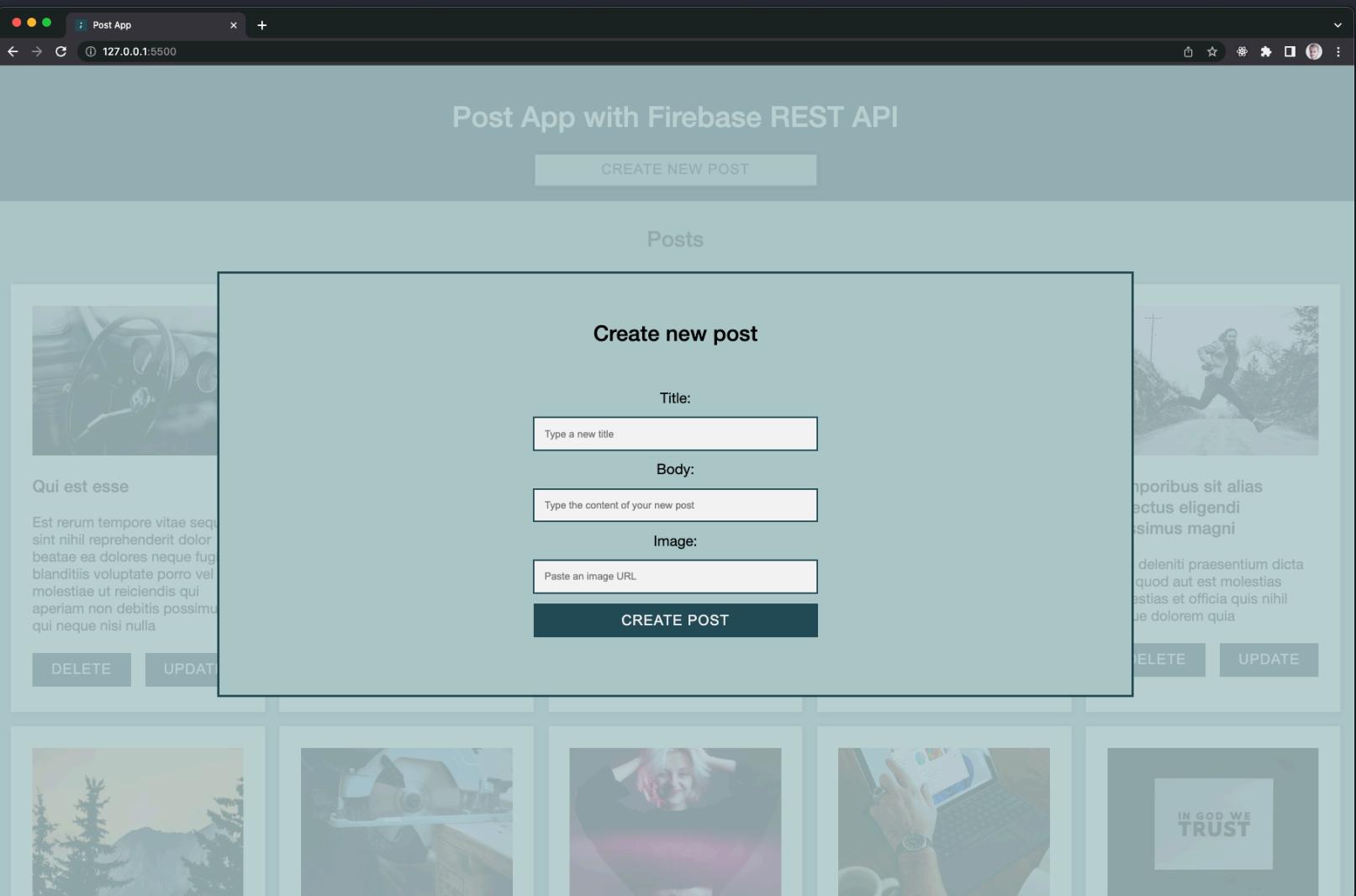
With Firebase REST API

<https://race.notion.site/Post-App-With-Firebase-REST-API-94b14a24519c467fad0299e1f8952e01>



Create

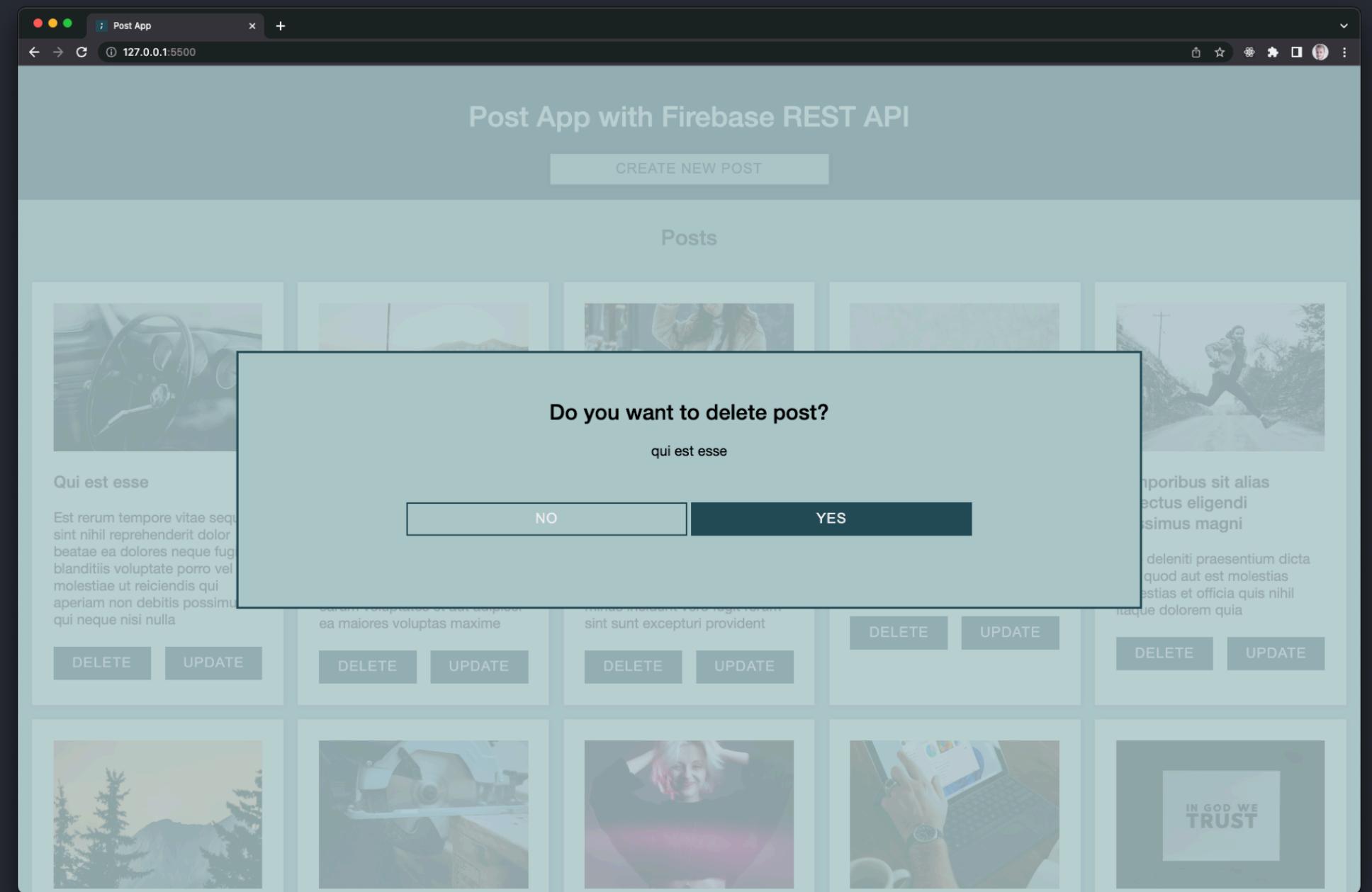
- Form og UI
- Function på submit, der tager informationer (værdier) ud af form og kalder:
- En funktion der foretager et POST request og opretter ny post i Firebase



```
function createPostClicked(event) {  
  event.preventDefault();  
  // ...  
  createPost(title, body, image);  
  // ...  
}  
  
async function createPost(title, body, image) {  
  // ...  
  const response = await fetch(`$endpoint}/posts.json`, {  
    method: "POST",  
    body: json  
  });  
  if (response.ok) {  
    //...  
  }  
}
```

Delete

- “Form” og UI
- Delete-knap åbner Dialog (confirm delete).
- “Yes” kalder deletePostClicked, som tager id fra data-id attribut og bruger det argument til at kalde...
- En funktion der foretager et DELETE request og sletter post i Firebase

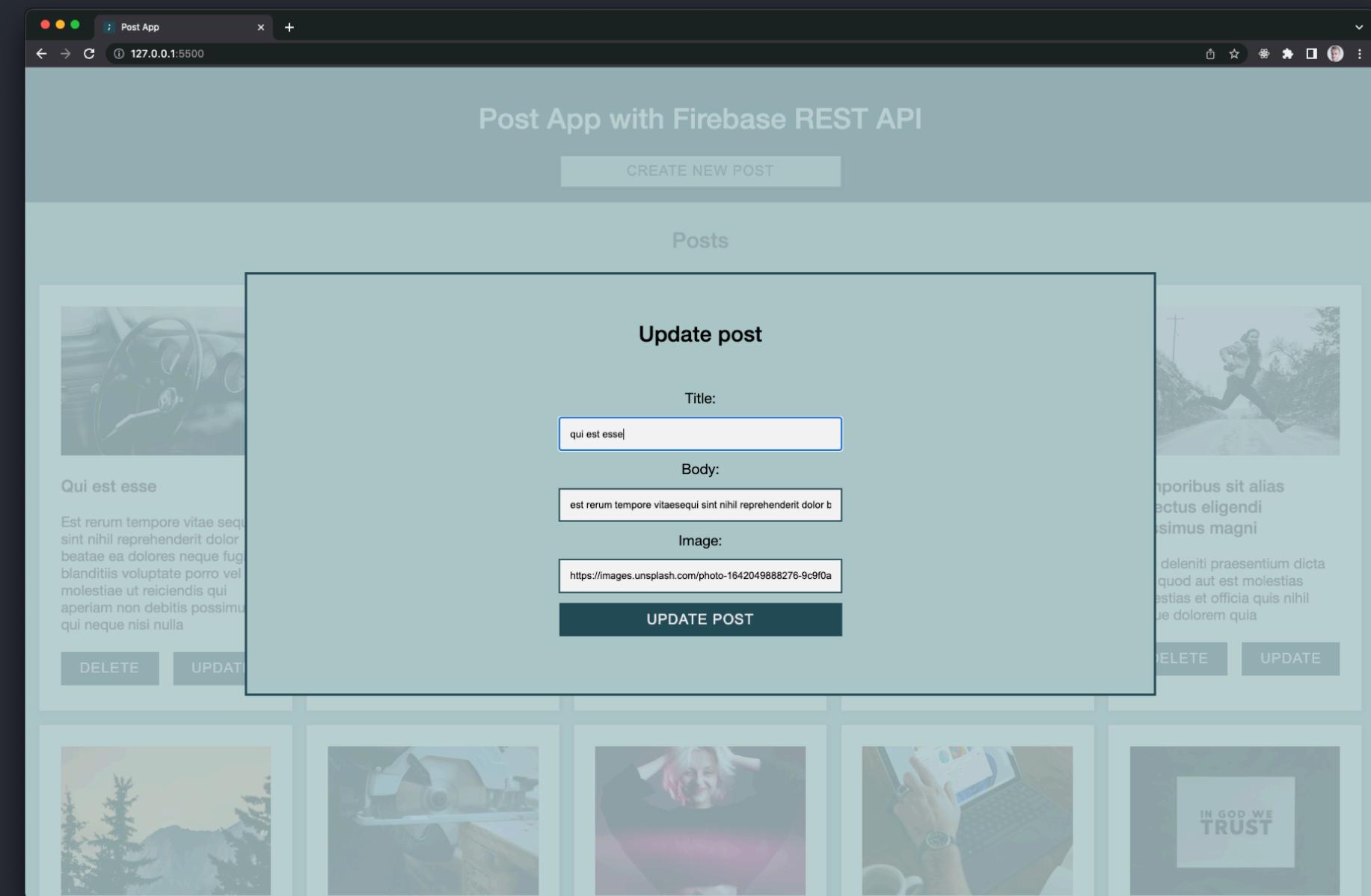


```
function deletePostClicked(event) {
  const id = event.target.getAttribute("data-id");
  deletePost(id); // call deletePost with id
}

async function deletePost(id) {
  const response = await fetch(` ${endpoint}/posts/${id}.json` , {
    method: "DELETE"
  );
  if (response.ok) {
    // ...
  }
}
```

Update

- Form og UI
- Dialog med update form.
- På “Update Post” kaldes updatePostClicked, der trækker informationer (værdier) fra form samt id fra data-id attribut og kalder...
- updatePost som foretager PUT request og opdaterer post i Firebase.



```
function updatePostClicked(event) {
  event.preventDefault(); // prevent default behaviour of submit
  ...
  const id = form.getAttribute("data-id");
  updatePost(id, title, body, image); // call updatePost with arguments
  document.querySelector("#dialog-update-post").close(); // close dialog
}

async function updatePost(id, title, body, image) {
  ...
  const response = await fetch(` ${endpoint}/posts/${id}.json`, {
    method: "PUT",
    body: json
  });
  ...
}
```

Array Reference

Array Methods

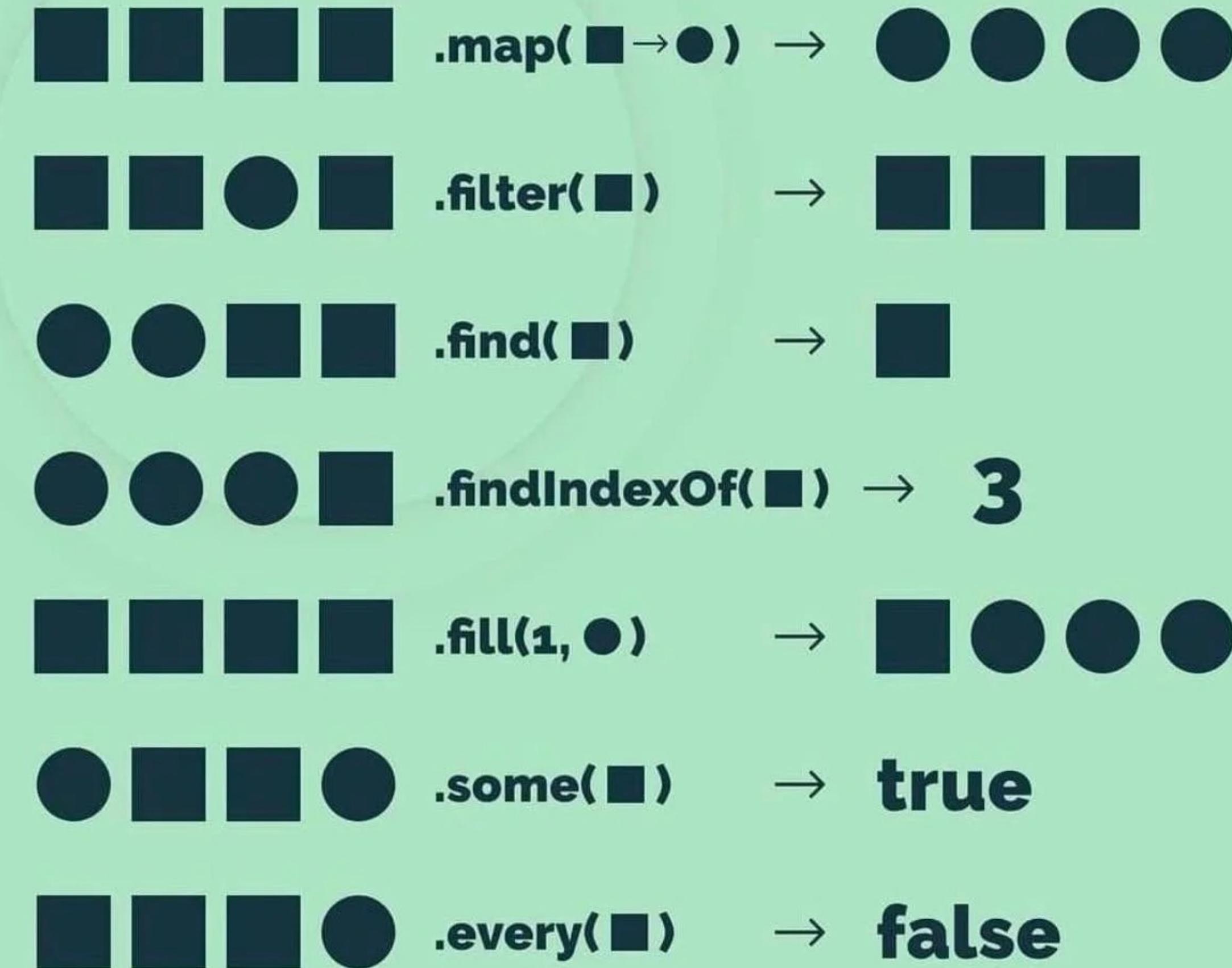
Der er mange, og vi skal nok vende tilbage til dem igen og igen.

The screenshot shows a web browser window with the title "JavaScript Array Reference" from w3schools.com. The page lists various array methods and properties with their descriptions. On the left, there's a sidebar titled "JS Array" with a list of methods. The main content area has a table with columns for "Name" and "Description". A sidebar on the right promotes "Start DEVELOPING Without Installations" and "BACKEND Support Included".

Name	Description
concat()	Joins arrays and returns an array with the joined arrays
constructor	Returns the function that created the Array object's prototype
copyWithin()	Copies array elements within the array, to and from specified positions
entries()	Returns a key/value pair Array Iteration Object
every()	Checks if every element in an array pass a test
fill()	Fill the elements in an array with a static value
filter()	Creates a new array with every element in an array that pass a test
find()	Returns the value of the first element in an array that pass a test
findIndex()	Returns the index of the first element in an array that pass a test
flat()	Creates an array from an object
flatMap()	Check if an array contains the specified element
forEach()	Calls a function for each array element
from()	Search the array for an element and returns its position
includes()	Checks whether an object is an array
indexOf()	Joins all elements of an array into a string
join()	
keys()	
lastIndexOf()	
length	
map()	
pop()	
push()	
shift()	
slice()	
sort()	
splice()	
unshift()	

https://www.w3schools.com/jsref/jsref_obj_array.asp

Array Methods



<https://javascript.info/array-methods>

<https://medium.com/@mandeepkaur1/a-list-of-javascript-array-methods-145d09dd19a0>

Loops, iteration & Array-metoder

Jeg har samlet lidt
lækkert til jer 

Loops, iteration & Array-metoder

↳ 1 backlink

0. Projektskabelon

1. Loops & iteration

1.1. forEach

Gennemløb names

forEach og index

Gentag med years og teachers

DOM-Manipulation (ekstra)

1.2. For Loop

Gennemløb years

Numbers

Mere for træning (ekstra)

1.3. For Of Loop

Log teachers

Hvis age er over 40

Søgefunktion med for of

Mere for of træning (ekstra)

1.4. For In Loop

Gennemløb alle props i teacher

Hvorfor for in?

1.5. While Loop

Sålænge index er mindre end teachers.length

Hvis teacher er "Lecturer"

Søgefunktion med while

Mere while træning (ekstra)

2. Array-metoder

.filter

.find

.sort

0. Projektskabelon

- Clone eller fork følgende projekt fra GitHub:  [loops-filter-find-and-sort](#) 
- Undersøg projekt. Hvad har du? Hvad er der i JavaScript-filen?

Loops, iteration & Array-metoder

.filter()

Array Method

```
const users = [  
    { name: "John", age: 35 },  
    { name: "Pete", age: 40 },  
    { name: "Mary", age: 44 }  
];  
  
// returns an array with users over the age 39  
const results = users.filter(checkAge);  
  
function checkAge(user) {  
    return user.age > 39;  
}  
  
console.log(results);
```

.filter() creates a new array with elements that pass a test provided by a function.

The filter() method does not execute the function for empty elements.

The filter() method does not change the original array.

.filter()

Og Post App.

includes() method returns true if a string contains a specified string. Otherwise, it returns false.

The includes() method is case-sensitive. Therefore, convert the strings you “compare” to lowercase.

```
const results = posts.filter(checkTitle);

function checkTitle(post) {
  const title = post.title.toLowerCase();
  return title.includes(searchValue);
}
```

https://www.w3schools.com/jsref/jsref_includes.asp
https://www.w3schools.com/jsref/jsref_tolowercase.asp

.find()

Array Method

```
const users = [  
    { name: "John", age: 35 },  
    { name: "Pete", age: 40 },  
    { name: "Mary", age: 44 }  
];  
  
// returns an object with the  
// first matching user of the age 40  
const result = users.find(matchAge);  
  
function matchAge(user) {  
    return user.age === 40;  
}  
  
console.log(result);
```

find() returns the value of the first element that passes a test.

find() executes a function for each array element.

find() returns undefined if no elements are found.

find() does not execute the function for empty elements.

find() does not change the original array.

.find()

Og Post App.

Vi bruger den ikke, men
det kunne vi have gjort 🤷

Fx til at finde én given post
på baggrund af dets id.

```
function findPostById(id) {  
  const result = posts.find(matchId);  
  
  function matchId(post) {  
    return post.id === id;  
  }  
  
  return result;  
}  
  
findPostById("-NTmr_tuQo2Fla_8gWEz");  
findPostById("LWPSKpRvx7Htyjl8JqV");
```

.sort()

Array Method

```
const names = ["Peter", "Magdalena", "Frederikke", "Anders"];
names.sort(); // sorts the original array with strings
console.log(names);
```

```
const years = [2003, 2032, 1990, 1989, 1975, 2023, 1933];
years.sort(); // sorts the original array of numbers
console.log(years);
```

.sort() sorts the elements of an array.

.sort() overwrites the original array.

.sort() sorts the elements as strings in alphabetical and ascending order.

.sort()

Array Method

```
const users = [
    { name: "John", age: 35 },
    { name: "Pete", age: 40 },
    { name: "Mary", age: 44 }
];

// compare and sort by the age prop
users.sort(compareAge);

function compareAge(user1, user2) {
    return user1.age - user2.age;
}

// compare and sort by the name prop
users.sort(compareName);

function compareName(user1, user2) {
    return user1.name.localeCompare(user2.name);
}
```

If you need to sort an array of objects, you must provide a compare function.

The compare function defines how to compare the objects.

.sort() uses the compare function to sort the original array.

.sort()

Og Post App.

Vi kan bruge sort til at sortere visningen af posts. Fx efter title eller body værdierne.

.sort(...) har brug for, at vi fortæller hvordan post-objekterne skal “sammenlignes” vha en funktion som argument.

.localeCompare sammenligner to strings.

```
posts.sort(compareTitle);

function compareTitle(post1, post2) {
  return post1.title.localeCompare(post2.title);
}

posts.sort(compareBody);

function compareBody(post1, post2) {
  return post1.body.localeCompare(post2.body);
}
```



Code Every Day

- Loops, iteration & Array-metoder
- Post App With Firebase REST API