**Implement pagination in Node JS Backend**

Currently we are showing all the posts in React frontend home page and also in the admin page. Let's implement pagination so that we can load certain number of posts by default. Then as the user hits Next or Previous button, We will fetch new posts based on page.

So the default page will be 1 and as the user hits Next or Previous buttons, It will update. So the updated page will be sent to backend as a query param.

Let's handle it in the backend now. Then in the next lecture you will see how we make a GET request to backend with page number as a route param.

**/nodeapi/controllers/post**

exports.getPosts = async (req, res) => {

// get current page from req.query or use default value of 1

const currentPage = req.query.page || 1;

// return 3 posts per page

const perPage = 3;

let totalItems;

const posts = await Post.find()

// countDocuments() gives you total count of posts

.countDocuments()

.then(count => {

totalItems = count;

return Post.find()

.skip((currentPage - 1) \* perPage)

.populate("comments", "text created")

.populate("comments.postedBy", "\_id name")

.populate("postedBy", "\_id name")

.sort({ date: -1 })

.limit(perPage)

.select("\_id title body likes");

})

.then(posts => {

res.status(200).json(posts);

})

.catch(err => console.log(err));

};

Now back to frontend. Visit '/'

You will see only 3 posts on the page, we need a way to load more posts.

So let's move to react frontend in the next lecture where we will implement pagination in the frontend.