Why should we use a pagination for users, products or any items?

Pagination is a common feature in many [web applications](https://www.turing.com/development-services/web-apps). It allows users to navigate through large sets of data by displaying only a subset of that data at a time. Implementing JavaScript pagination is relatively straightforward and in this article, we’ll explore how to do so as well as the best practices to follow.

We'll implement a simple pagination functionality that allows users to navigate through a list of items with a fixed number of items displayed per page. Let's get started!

Why use JavaScript pagination?

JavaScript pagination can provide several benefits to a website or application:

* Firstly, it allows you to break up large amounts of content into manageable chunks, making it easier for users to navigate and find the information they need. This can improve user experience and engagement, as users are less likely to become overwhelmed or frustrated.
* Additionally, it can reduce page load times by only loading a subset of content at a time, which can improve overall site performance and speed. This is especially important for mobile users who may have limited bandwidth or slower connections.
* Finally, it allows you to dynamically update content without requiring a full page reload. T This can create a smoother and more seamless user experience. Overall, JavaScript pagination is a powerful tool for improving usability, performance, and engagement on a website or application.

Examples of pagination

JavaScript provides several options for implementing pagination, including client-side and server-side approaches.

* Client-side pagination involves rendering a limited number of items on the page and providing controls to move between pages. This approach is fast and efficient for small datasets but can be slow for larger ones.
* Server-side pagination, on the other hand, involves fetching a specific subset of data from the server based on the current page number and the number of items per page. This approach is more suitable for larger datasets but requires more server resources.

Some popular pagination libraries in JavaScript include DataTables, React-Paginate, and Bootstrap Pagination.