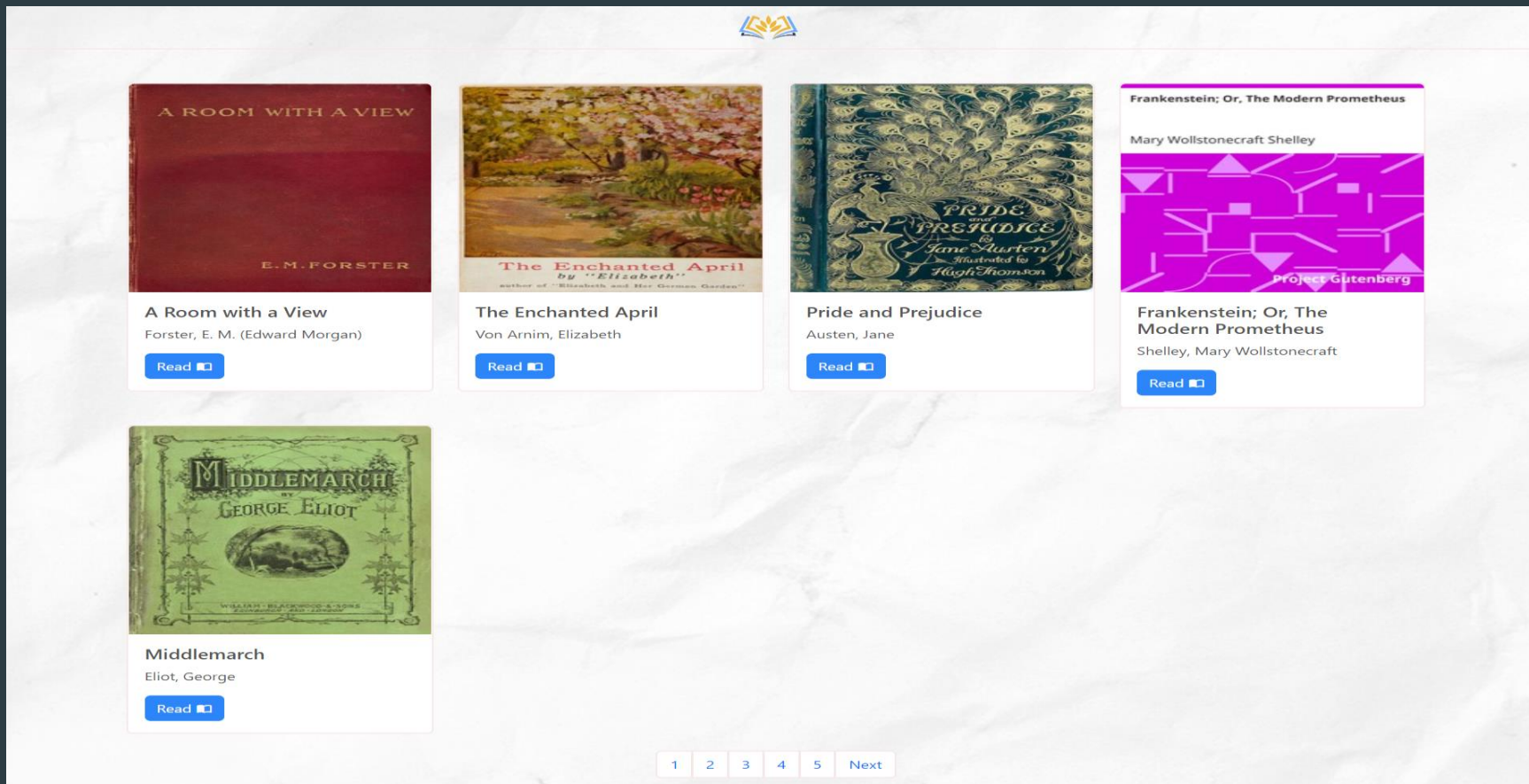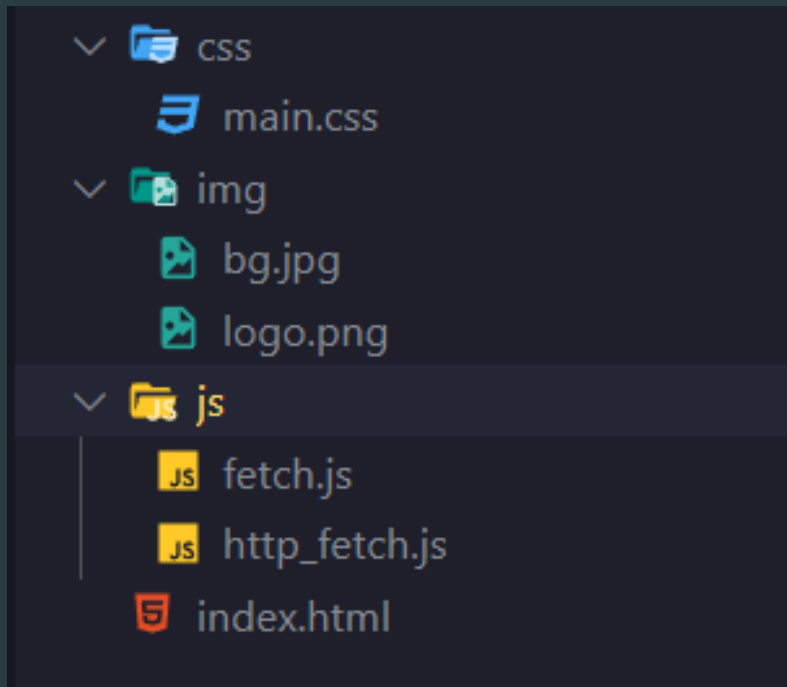# Bücherwurm Project

## Small project

# Project brief

▶ In this small project, I used the Gutenberg ebook API where I show the books by title, image, author, and the ability to read them.

# Programming language and Framework used in the project

# Projects files



- I made this project using two methods
- Method 1: http_fetch.js
- Method 2: fetch.js

# Method 1

```
const getBooks = (callBack) =>{
    var request = new XMLHttpRequest();

    request.addEventListener('readystatechange', () => {


        if (request.readyState === 4 && request.status === 200) {

            callBack(undefined, request);
        }
        else if (request.readyState === 4)
        {

            callBack("error getting Data", undefined);
        }
    });

    request.open("GET","https://gutendex.com/books/");
    request.send();


}
```

► In first method I used object XMLHttpRequest() that allow to make HTTP requests.

► then I used request.open() to specify the type of request and URL.

► Then I used request.readyState to check if the operation is done

► also I used request.statuts to check of request success

# Callback

```
getBooks(callBack = (error, data)=>{
    if(error)
    {
        console.log(error);
    }
    else
    {
        setupPagination(data.responseText);
    }
});
```

► In this example, I used with a callback function to handle the errors and data.

► A Callback function is effective to make async that waits until another function completes and returns with data. However, the dark side of the callback function can cause with what noun callback hell especially when we have consecutive asynchronous requests.

# Method 2

```
//this function return promise
const getBooks = async () =>{


//in this line we wait until we receive the
 response
    const response = await fetch("
https://gutendex.com/books");
    const data = await response.json();


    return data;
}
```

▶ **Fetch** is similar to **XMLHttpRequest** fetch we write less code than **XMLHttpRequest**

▶ **Fetch** return a **promise** which makes handling success and error better the than callback

▶ The **await** keyword is used in an **async** function to ensure that all promises returned.

# Thank you
# For you Attention