

## **Bookworm**

**This document explains why I selected this topic, purpose of it and the process of development.**

When I was younger, I used to read a lot and books were my passion. When I was pursuing a bachelor degree, I made several small webapps, like a library or a bookshop API. When thinking about what project to choose, the bookshop came to my mind. It would be a good practice for me to select something in the domain of the marketplace because writing a back-end code (please, let it be Java) would tackle my mind more.

For instance, the process of adding a book to the basket, purchasing it using Stripe API, updating the status in the database - this sounds like a prominent web application.

I added more than 8 HTML pages in my web app, because it felt quite empty without that quantity.

Without these, the website would feel too minimal. However, I chose not to create an individual page for every book at this stage to avoid unnecessary overhead. Instead, this will be handled dynamically when I develop the back-end.

My initial thought was to use Tailwind instead of a pure CSS, but I felt that would be overengineering.

I have a .git directory, where all my commits are stored. I started this application around 2 weeks ago, doing a basic header. In Rubric there was a statement for advanced HTML tags and CSS properties to be used. During development of my old projects, I used some animation. In the Bookworm project, I decided to add animation that I used before, for instance, flipping text or pulsating text or dancing shadow effects.

I decided to add some information about "our staff" - people working in Bookworm. While there is no one to be working there, adding information about people would seem more professional.

I used the Eason website as a reference, so our styles look alike. When you open Eason, there are blocks of New, Bestsellers, Books for Different Age groups. Eason also shows their benefits - a loyalty card and a discount system. This inspired me to implement a similar structure for my main page.

I adopted their approach to customer benefits, such as loyalty programs. In Bookworm, customers earn 25 points per book purchased, and once they accumulate 500 points, they can redeem a free book of their choice.

I designed a loyalty card in Figma and then exported it. I don't have much experience in Figma and other design tools, but I felt that Bookworm deserves their own loyalty card instead of just googling and downloading random pictures.

Footer is the same as footer on any website - Contact Us, FAQ, Delivery Information. Shipping & Delivery, Terms and Conditions and FAQ have the same layout, because I thought to make it minimalistic yet pretty.

I decided to add Events as well, so our customers would join online to enjoy a workshop or film screening or other types of events. Ennis Library website has a similar events system, which led me to the idea of doing the same.

Every e-commerce website should have an account system. When entering my website, they can Sign In. If they have no account, they can Sign up. After that, they'll see their personal data, change it and view the orders table.

When clicking 'Add to basket' on a book, it will be sent to the Basket page, which you can access just by clicking the Cart icon. In the near future, I'll add Stripe functionality for payment purposes.

Bookworm is an evolving project that blends my love for books with my passion for coding. By focusing on marketplace functionality and full-stack development, I am building a solid app that challenges my technical skills while creating a meaningful user experience.