



# **IT 309 SOFTWARE ENGINEERING**

## **PROJECT DOCUMENTATION**

Bookshelf

Prepared by:

**Amira Abdo**

**Rania Weiss**

Proposed to:

**Nermina Durmić, Assist. Prof. Dr.**

**Aldin Kovačević, Teaching Assistant**

Date of submission

**21st June, 2023**

<b>1. Introduction</b>	<b>3</b>
1.1. About the Project	3
1.2. Project Functionalities and Screenshots	3
<b>2. Project Structure</b>	<b>8</b>
2.1. Technologies	8
2.2. Database Entities	9
2.3. Design Patterns	9
2.4. Tests	10
<b>3. Conclusion</b>	<b>10</b>
3.1. Final thoughts	10
3.2. Future	10

# 1. Introduction

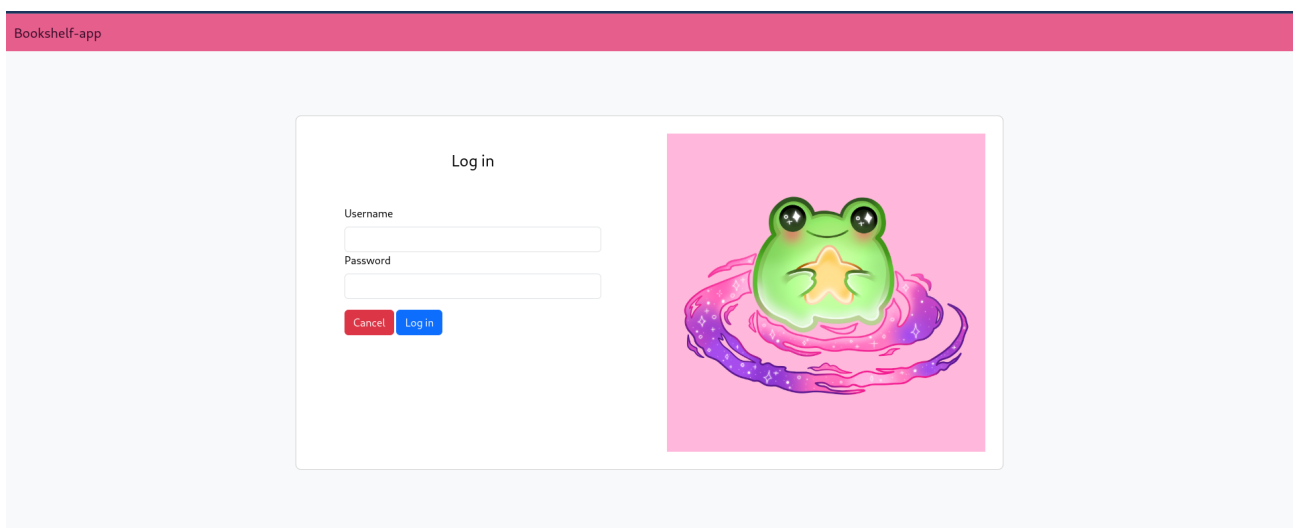
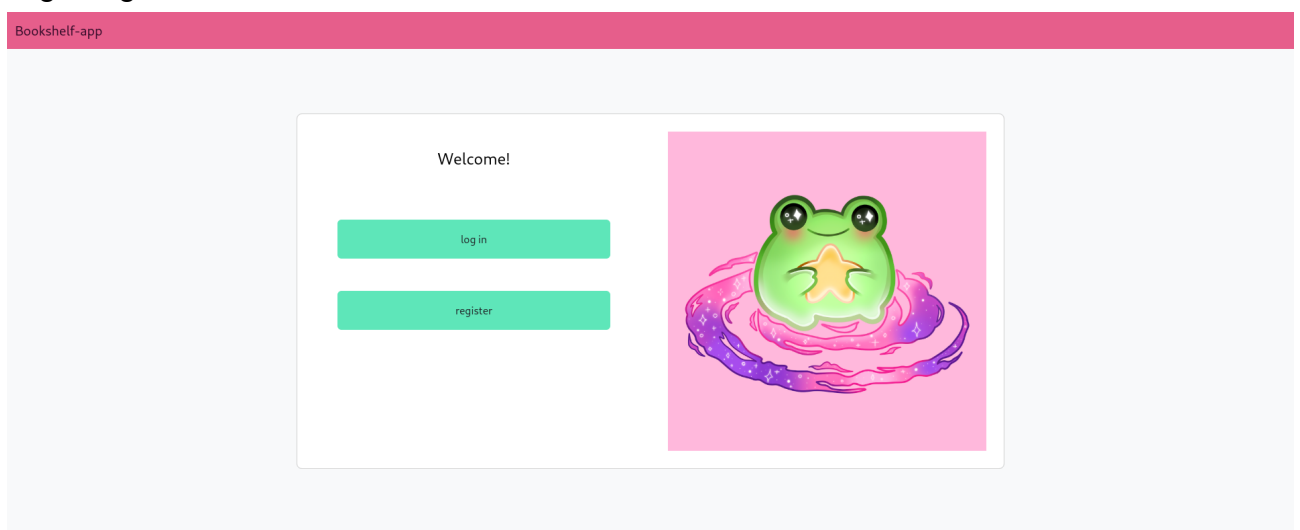
## 1.1. About the Project

Bookshelf is a manager application which keeps track of what the user has read, what they are still reading, and what they have finished. The feature for adding books to bookshelves allows the user full control over how they would like to organize their books. Books can be created and accessed by anyone, but each user has their own shelves which they manage.

[Link](#)

## 1.2. Project Functionalities and Screenshots

- Login/registration




### Register

Email

Username

Password

[Cancel](#) [Register](#)



Album example is © Bootstrap, but please download and customize it for yourself!  
New to Bootstrap? [Visit the homepage](#) or read our [getting started guide](#).

[Back to top](#)

- list of books

Bookshelf-app

My shelves

My books

Search

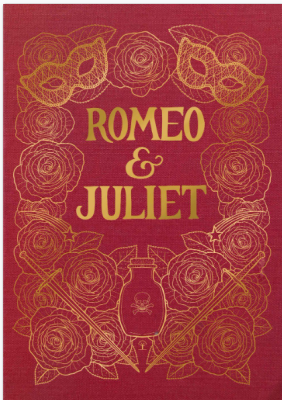
Search

Log out

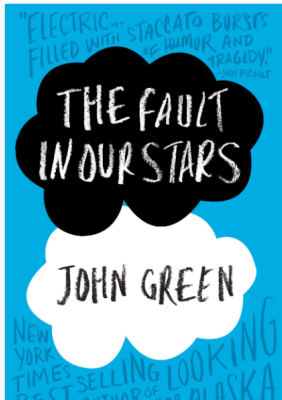
Welcome!

Add shelf


Add book



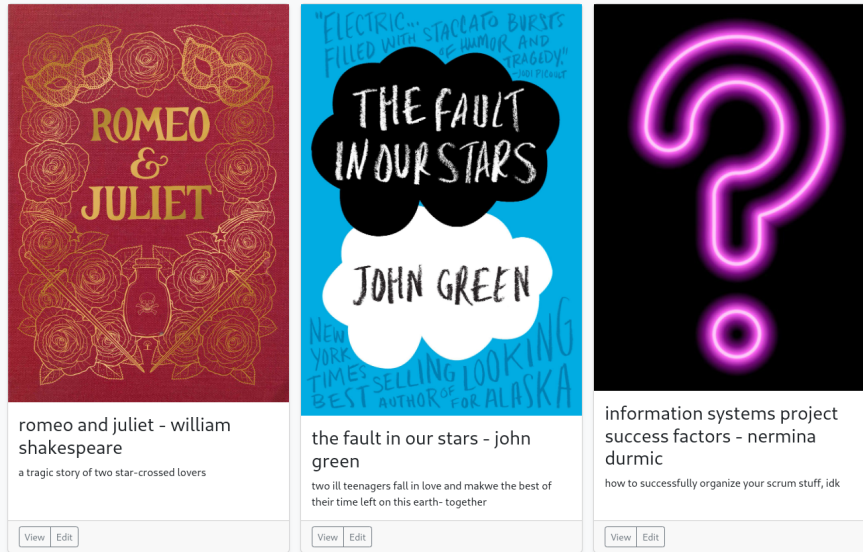
romeo and juliet - william shakespeare



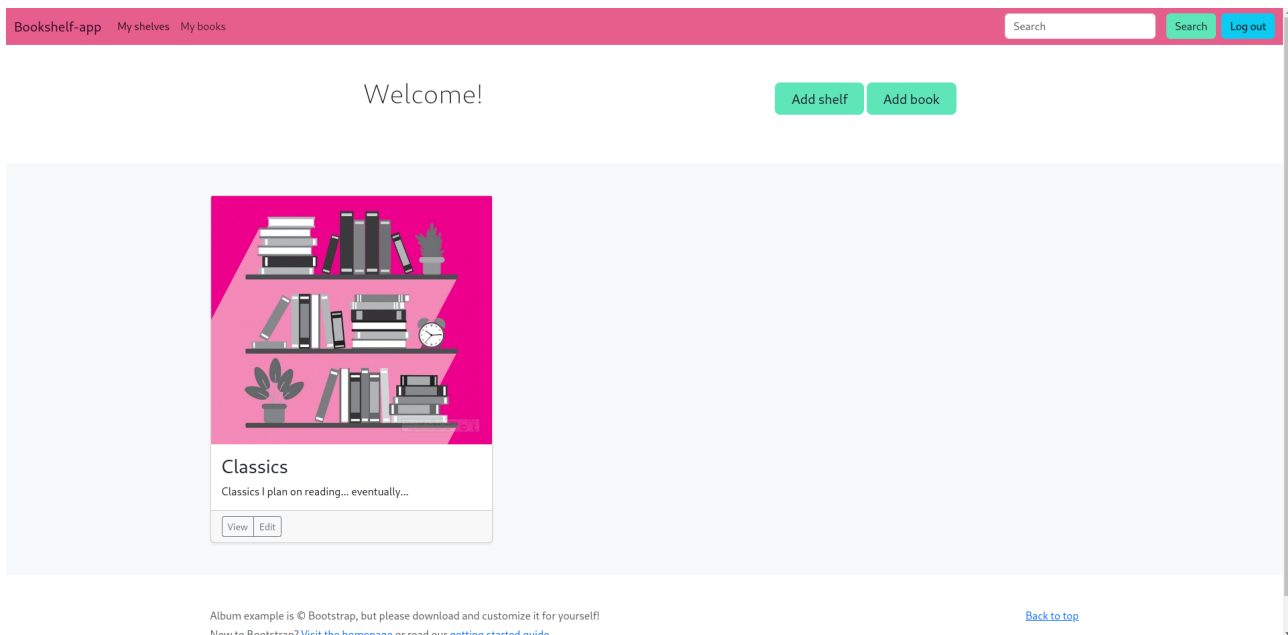
the fault in our stars - john green

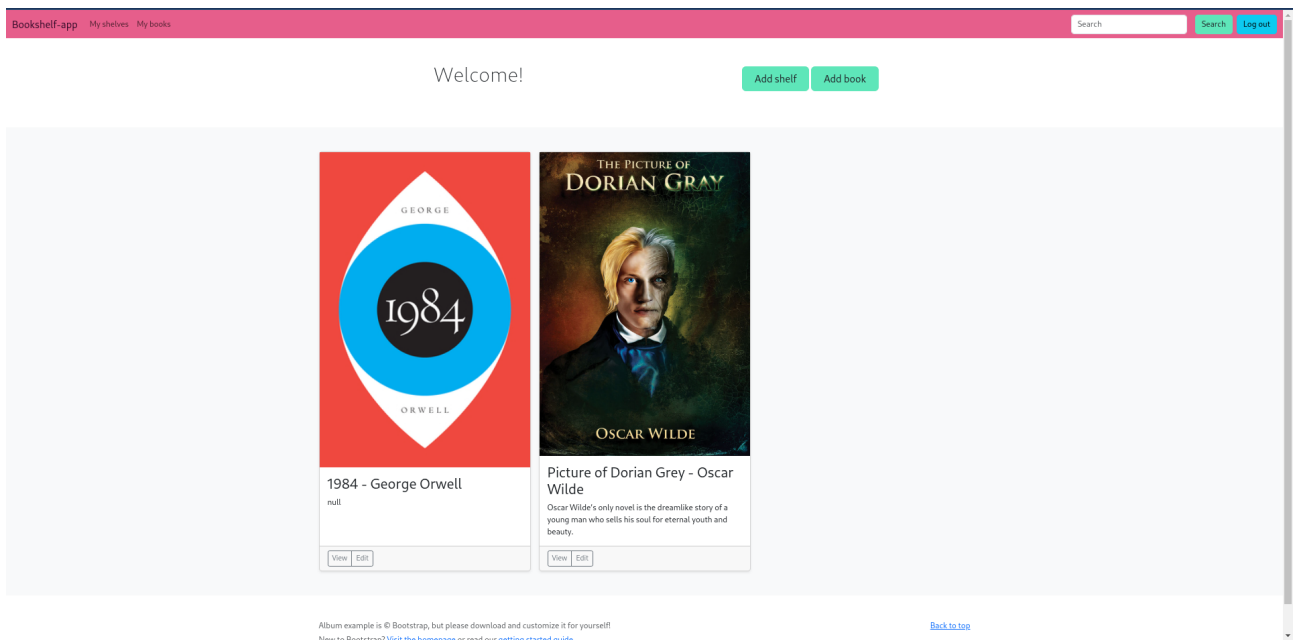


information systems project success factors - nermina

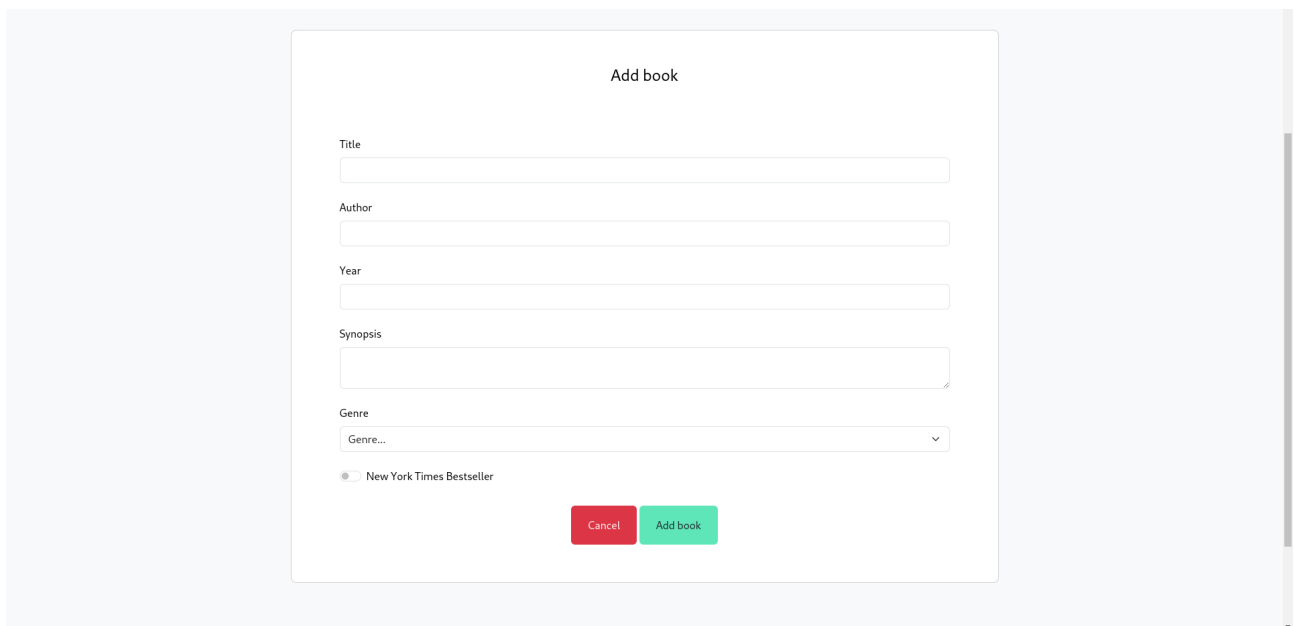


- list of shelves per user
  - books in the shelf





- creating book



- creating shelf

Welcome!

Add shelf

Add book

#### Add shelf

Name

Description

Cancel

Add shelf

Album example is © Bootstrap, but please download and customize it for yourself!  
New to Bootstrap? [Visit the homepage](#) or read our [getting started guide](#).

[Back to top](#)

- updating book

#### Edit book

Title

Author

Year

Synopsis

Genre

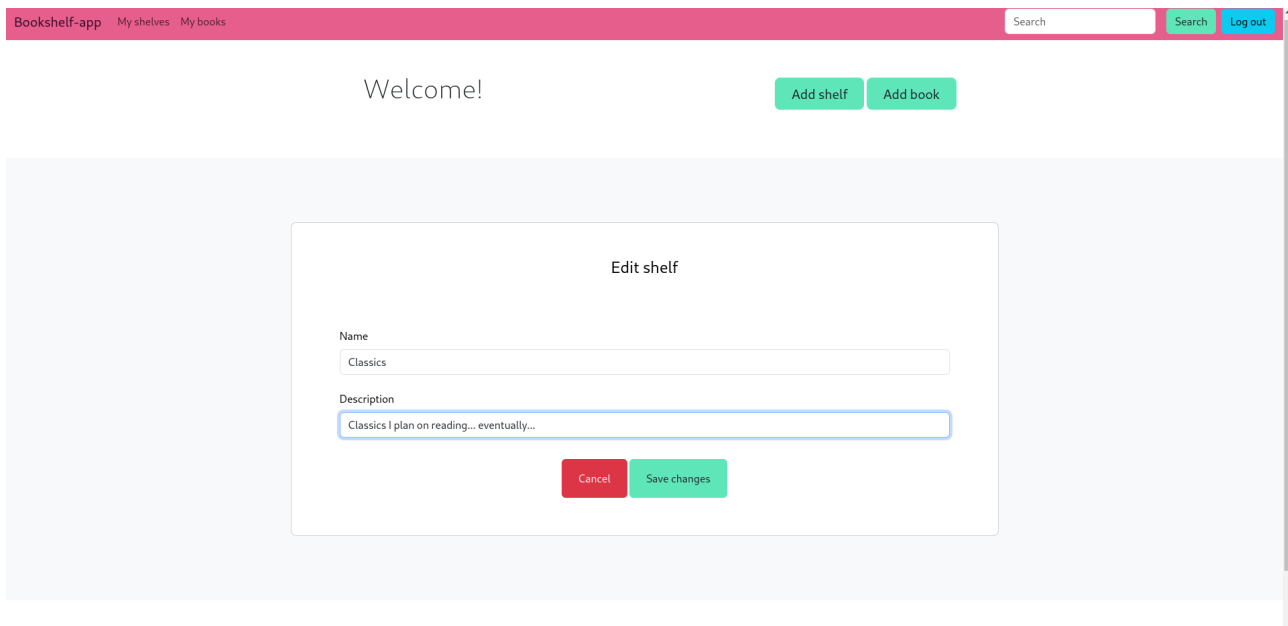
☐ New York Times Bestseller

☐ Read  
☒ Reading  
☐ Unread

Cancel

Update book

- updating shelf



## 2. Project Structure

### 2.1. Technologies

For this project, we intended to use Flutter, but due to complications with it, we hastily switched to our familiar LAMP stack. We used FlightPHP on the backend, and JQuery and Bootstrap on the frontend.

The coding standards we used on the backend is PSR-12 for PHP, and for frontend we wanted to use StandardJS, but unfortunately implementing it was not as simple. We were hoping the process of making our frontend adhere to StandardJS would be as simple as using a linter, and creating a script to run it before committing our code, but since we implemented it too late, it was almost impossible to make the code adhere to the standards. In our pre-commit script, we have commented out the code which would check if our frontend adheres to the standards. The script successfully checks if the php files adhere to the standards, and corrects them if they do not.

### 2.2. Database Entities

- user
- book
- bookshelf
- book\_bookshelf → since the relationship between book and bookshelf was many-to-many, we had created an intermediary table called book\_bookshelf

### 2.3. Design Patterns

- builder → used in the backend, src/Book.php
  - only ever used when a book needs to be updated, since there are a lot of fields and there is a good chance that the user does not want to update all fields, rather only one or two of them
- singleton → used in the backend, src/Database.php



- since we want to always have only one connection to the database, we used this pattern

## 2.4. Tests

There are three tests which run before committing changes to the main branch:

- LoginTest.php
  - intended to test whether functionalities of the user entity have been compromised
  - this test uses specific credentials to check whether logging in is still possible by assuring that there is a JWT being returned and checking the value of that JWT
- GetBookByIdTest.php
  - tests whether functionalities of Book entity have been compromised
  - this test checks whether it is possible to retrieve the Book object with ID = 1, and if its values are accurate
- AddBookshelfTest.php
  - tests whether functionalities of Bookshelf entity have been compromised
  - this test tries to add a new bookshelf and retrieves its data to verify its accuracy

These are our most basic functionalities which we believe should be tested to assure there are no breaking changes in the new code.

## 3. Conclusion

### 3.1. Final thoughts

The application itself is not completed, since we prioritized fulfilling the requirements of the project rather than providing all promised features. On the other hand, we believe we managed to include design patterns, and did our best to make sure our code follows its specific standards, as well as general best practices.

### 3.2. Future

If we were to continue with the project, we would finish the intended app features, and would try to make our frontend code be compliant with StandardJS, even if it would mean rewriting a lot of the code.

Some features that we would like to add:

- allowing only the user who created a book to be able to edit it
- make it possible on the frontend to add a book to a shelf
  - this feature was a priority, but it was difficult to implement it in on the frontend without making it unnecessarily complicated
- delete operations for all entities
  - we did not consider this to be a priority, so there is only implementation on the backend