# Final assessment: 📚 Book club

## A big assessment

In this assessment we will test you on everything we learned the past weeks. This is a large project and you need to make sure you are in your best shape!

- **You better check yo' self before you wreck yo' self**
    - 🛡️ Make a battleplan.
    - 🦆 Get a rubber duck (or a partner, or a plant, just something that doesn't mind when you talk to it). **This really works!**
    - 🛏️ Make sure to get enough sleep, and drink and eat enough to keep your energy stable.
    - 🌳 Take a walk by yourself if you are stuck.
- **Read the whole document before you start!**
    - This assessment has a different scoring system than our other assessments. A good strategy is to prioritise the features that are easiest and end with the ones that will be the hardest.
        - Some features are dependent on other features, those will be hard to skip.
        - You can find the exact amount of points you can score per feature at the end of this document.
    - This assessment requires you to commit often and include meaningfull commit messages.
    - There is also a required self-reflection you will have to write.
- **If you have problems that have nothing to do with coding, contact a teacher**
    - We can't answer every question, but there is a chance that an error is not your fault. At that point we might be able to help you.

## Assessment Rules

- You may refer to the Reader, previous exercises, and search the internet.
- If you copy and paste code from the internet, please include the source URL in a comment.
- You can not seek assistance from other individuals.
- This challenge can not be shared with anyone.
- The challenge must be submitted before the deadline, failing to do so will result in a failing grade.
- The results of this challenge will be shared as soon as the teachers have completed the grading.
- If you feel like your challenge has been graded incorrectly or unfairly, contact a teacher to talk about it.

## Assessment deadline

The assessment deadline will be shared with you by the teachers

## Assessment grading

The total amount of points you can score for this assessment is 100, **a score of 60 is considered a passing grade**.

## Submitting the assessment

You should create two repositories for this assessment and submit them both through this form

Do not forget to add @aidenbuis & @Swendude as collaborators to your repository

Good luck, we believe in you! 🚀

## What are we building?

We are building a web app where users can track their progress in reading books, and share that information with other users. It's called `Book Club`.

The site has multiple pages:

- `/register` to register as a user
- `/login` to log in as a registered user
- `/` a homepage that shows a list of all the books in the system
- `/book/[bookId]` where everyone can see all the stats of a single book
- `/progress` where the logged-in user can see their progress for all their books

## Database Schema

### User

This model represents a user of our application.

| field | type | notes |
|---|---|---|
| id | Int | primary key |
| username | String | unique |
| password | String | |

### Book

This model represents a book that can be tracked on our application.

| field | type | notes |
|---|---|---|
| id | Int | primary key |
| title | String | unique |
| coverImgUrl | String | |
| author | String | |
| pageCount | Int | |

### BookProgress

This model represents the progress a user has made on a book

| field | type | notes |
|---|---|---|
| id | Int | primary key |
| bookId | Int | relation to `Book` |
| userId | Int | relation to `User` |
| pageProgress | Int | |

## Features

### Part 1 - Database

#### ! 1.1: Create the models

Set up a new backend project and create the models in Prisma. Make sure to set up the relations the right way.

| Part | Goal | Points | Total |
|---|---|---|---|
| **1.1 Create the models** | | | 5 |
| 1.1 Create the models | Database models set up correctly | 5 | |

#### ! 1.2: Create a seed file

Create a file named `seed.ts`, running this file should create 5 users, 10 books and 5 bookProgresses in the database.

| Part | Goal | Points | Total |
|---|---|---|---|
| **1.2 Create a seed file** | | | 2 |
| 1.2 Create a seed file | Seeder with at least 5 users, 10 books and 5 bookProgresses | 2 | |

### Part 2 - Server Preparations

#### ! 2.1 Set up auth

Add a `server.ts` file and add an endpoint for `POST /register`(to create a user) and an endpoint for `POST /login`.

- The `/register` endpoint should validate the data and respond with an object describing the error (a Zod error object is fine)
- The `/login` endpoint should respond with a token if the user's credentials are correct

| Part | Goal | Points | Total |
|---|---|---|---|
| **2.1 Set up auth** | | | 5 |
| 2.1 Set up auth | Register endpoint for users | 2 | |
| 2.1 Set up auth | Login endpoint for users | 3 | |

## Part 3 - The app

### ! 3.1 Auth pages

**User story**: *As a user, I want to be able to register for an account and use that account to log in so that I can see and manage my progress on books*

In your front-end project, add two new pages.

- `/register` should show a form that allows the user to register to the application. The data in the form should be validated and the user should receive feedback if any validation error happens.
- The validation for creating a user:
  - `username` should have a minimum length of 5 characters
  - `password` should have a minimum length of 10 characters
- `/login` should show a form that allows the user to log in, the resulting token should be stored in LocalStorage.

| Part | Goal | Points | Total |
|---|---|---|---|
| **3.1 Auth Pages** | | | 6 |
| 3.1 Auth Pages | `/register` page with form that creates a user by sending a `POST` request | 2 | |
| 3.1 Auth Pages | Register form validates the data | 2 | |
| 3.1 Auth Pages | `/login` page with form that retrieves a token by sending a `POST` request and stores it in LocalStorage | 2 | |

### ! 3.2 Navigation Bar

**User story**: *As a user, I want to see a navigation bar so I can quickly navigate between pages*

- Every page should have a navigation bar with a link to other pages
- The link to the `/progress` page should only appear if the user is logged in
- The links to `/register` and `/login` should disappear if the user is logged in and replaced with a button to log out.

| Part | Goal | Points | Total |
|---|---|---|---|
| **3.2 Navigation Bar** | | | 9 |

| Part | Goal | Points | Total |
|---|---|---|---|
| 3.2 Navigation Bar | There is a NavBar on every page and it has a button for every page | 2 | |
| 3.2 Navigation Bar | Only a logged in user sees the link to `/progress` | 3 | |
| 3.2 Navigation Bar | The links to `/register` and `/login` are replaced by a logout button | 2 | |
| 3.2 Navigation Bar | The logout button works | 2 | |

### ! 3.3 Homepage

**User story**: *As a user, I want to see a list of all the books available on the site so that I can decide what to read*

**User story**: *As a user, I want to be able to sort the list of books alphabetically so I can quickly find a book*

**User story**: *As a user, I want to see be able to sort the list of books by the number of users currently reading it so I can see which books are popular*

- The `/` (home)page should display a list of all available books
- The books should come from the API endpoint `GET /books`
- By default, Books should be ordered alphabetically by their name
- There are two buttons on the page: `Sort by name` and `Sort by popularity`.
  - When `Sort by name` is clicked, the books should be sorted by name
  - When `Sort by popularity` is clicked, the books should by sorted the amount of BookProgresses objects are connected to it. (It doesn't matter what the value of `pageProgress` is)
- Clicking on an book's name redirects to `/books/[bookId]`
- Books are displayed with the number of BookProgresses they have in the database and their name

**On the Backend**

There needs to be an endpoint `GET /books` that returns the necessary information.

| Part | Goal | Points | Total |
|---|---|---|---|
| **3.3 Homepage** | | | 16 |
| 3.3 Homepage | The homepage shows a list of available books that come from the API | 4 | |
| 3.3 Homepage | The books are ordered alphabetically on their name by default | 3 | |
| 3.3 Homepage | There are buttons to `Sort by name` and `Sort by popularity`. These buttons work | 3 | |

| Part | Goal | Points | Total |
|---|---|---|---|
| 3.3 Homepage | Every book is displayed with the number of bookProgresses they have and their name | 4 | |
| 3.3 Homepage | Clicking on an book's name redirects you to the detail page for that book | 2 | |

## ! 3.4 Book Detail Page

**User story**: *As a user, I want to be able to see the details of a specific book, I also want to see how many users are currently reading the book so that I can decide if the book is interesting*

- The `/books/[bookId]` dynamic page displays the details for a book. A book is displayed with:
    - title
    - The cover image of the book (get it from `coverImageUrl`)
    - The author
    - The pageCount
    - The amount of BookProgresses that are in the database and belong to this book
    - The average `pageProgress` for this book. To calculate this:
        - Get all the BookProgress objects associated with this book, sum their `pageProgress` values and divide by the number of BookProgress objects

**On the Backend**

There needs to be an endpoint `GET /books/:book_id` endpoint that returns the necessary information.

| Part | Goal | Points | Total |
|---|---|---|---|
| **3.4 Book Details Page** | | | 13 |
| 3.4 Book Details Page | The detail page shows the books information | 3 | |
| 3.4 Book Details Page | The data comes from the API | 2 | |
| 3.4 Book Details Page | The detail page shows the amount of bookProgresses belonging to this book | 3 | |
| 3.4 Book Details Page | The detail page shows the book's average `pageProgress` | 5 | |

## ! 3.5 Create a BookProgress

**User story**: *As a user, I want to be able to start tracking my progress on a book*

- There's a button on the `/books/[bookdId] page that says 'Start reading book'.
- This button is only shown if the user is logged in

- Clicking the button should create a new BookProgress in the database, with the correct `userId` and `bookId`.
- The new BookProgress should be created with `pageProgress` equal to `0`

**On the Backend**

There needs to be an endpoint `POST /bookprogress` endpoint that creates a new BookProgress object in the database and uses authentication to determine the `userId`

| Part | Goal | Points | Total |
|---|---|---|---|
| **3.5 Create a BookProgress** | | | 9 |
| 3.5 Create a BookProgress | There is a button that creates a bookProgress on the detail page | 2 | |
| 3.5 Create a BookProgress | Clicking the button creates a BookProgress in the database with the correct `userId` and `bookId` and `pageProgress` | 4 | |
| 3.5 Create a BookProgress | The button is only shown if the user is logged in | 3 | |

## ! 3.6 The Progress Page

**User story**: *As a user I want to see an overview of all the books I'm reading so I can get an idea of my progress*

- The `/progress` page shows a list of all the books that have a BookProgress by the logged in user
- Every book shows the amount of pages the user has read and the total amount of pages in the book (something like: `50/230 pages read`)
- The page can only be seen if the user is logged in
    - If a user is not logged in and navigates to this page, one of the following should happen:
        - The user is redirected to `/login`
        - The user sees an error message, something like: `Login to see your progress` or `You are not authorized`, for example.

**On the Backend**

There needs to be an endpoint `GET /my-progress` that uses authentication to get all the books the user has a BookProgress for.

| Part | Goal | Points | Total |
|---|---|---|---|
| **3.6 The Progress Page** | | | 10 |
| 3.6 The Progress Page | The progress page shows a list of all the books that have a BookProgress for the current User | 2 | |

| Part | Goal | Points | Total |
|---|---|---|---|
| 3.6 The Progress Page | Every book shows the amount of pages read and the total amount of pages | 4 | |
| 3.6 The Progress Page | The page can only be seen when a user is logged in | 4 | |

### ! 3.7 Modify Progress Form

**User story**: *As a user I want to be able to modify the progress for a specific book so that I can track my progress*

- Every book on the `/progress` page should have a form that allows the user to modify the `pageProgress` for that specific book.
- There is no requirement for how the form looks like, it can be a textbox, slider, etc..
- There should be a button to submit the new `pageProgress`.

**On the Backend**

There needs to be an endpoint `PATCH /bookprogress/:id` that uses authentication to modify the `pageProgress` of that BookProgress object. This endpoint will need additional validation described in the next feature.

| Part | Goal | Points | Total |
|---|---|---|---|
| **3.7 Modify Progress Forms** | | | 9 |
| 3.7 Modify Progress Form | The progress page displays a a form for every book | 5 | |
| 3.7 Modify Progress Form | Submitting the form with a button changes the `pageProgress` for the associated BookProgress | 4 | |

### ! 3.8 Backend BookProgress validation

**User story**: *As a user I want the application to check if my progress is valid so that there is no chance I create an invalid progress*

- The `PATCH /bookprogress/:id` endpoint should check if the new `pageProgress` value is a valid one.
    - The pageProgress can't be higher than the associated book's `pageCount` value.

| Part | Goal | Points | Total |
|---|---|---|---|
| **3.8 Backend BookProgress validation** | | | 6 |

| Part | Goal | Points | Total |
|---|---|---|---|
| 3.8 Backend BookProgress validation | The `PATCH /bookprogress/:id` endpoint does not allow to set the `pageProgress` value higher than the associated Book's `pageCount` value | 6 | |

### ! 4.1 Code Organization

- You should make multiple commits during the project, using descriptive commit messages
- Any code you copy-pasted from somewhere else should have a reference to where you found it
- The README.md should contain a short message explaining the project

| Part | Goal | Points | Total |
|---|---|---|---|
| **4.1 Code Organization** | | | 5 |
| 4.1 Code Organization | The commit messages are meaningfully, there is not only one single commit | 3 | |
| 4.1 Code Organization | The `README.md` contains a description of the project | 2 | |

### ! 5.1 Reflection

- Add a file named `REFLECTION.md` To your **Frontend** repository, in this file you should reflect on your work, specifically answering these questions:
  - What parts did you consider easy?
  - What parts did you consider hard?
  - Is there anything you could have done better?
  - Are you aware of any problems with your code?

| Part | Goal | Points | Total |
|---|---|---|---|
| **5.1 Self Reflection** | | | 5 |
| 5.1 Self Reflection | The `REFLECTION.md` contains a self reflection by the student | 5 | |

## Points overview

| Part | Goal | Points | Total |
|---|---|---|---|
| **1.1 Create the models** | | | 5 |
| 1.1 Create the models | Database models set up correctly | 5 | |
| **1.2 Create a seed file** | | | 2 |

| Part | Goal | Points | Total |
|---|---|---|---|
| 1.2 Create a seed file | Seeder with at least 5 users, 10 books and 5 bookProgresses | 2 | |
| **2.1 Set up auth** | | | 5 |
| 2.1 Set up auth | Register endpoint for users | 2 | |
| 2.1 Set up auth | Login endpoint for users | 3 | |
| **3.1 Auth Pages** | | | 6 |
| 3.1 Auth Pages | `/register` page with form that creates a user by sending a `POST` request | 2 | |
| 3.1 Auth Pages | Register form validates the data | 2 | |
| 3.1 Auth Pages | `/login` page with form that retrieves a token by sending a `POST` request and stores it in LocalStorage | 2 | |
| **3.2 Navigation Bar** | | | 9 |
| 3.2 Navigation Bar | There is a NavBar on every page and it has a button for every page | 2 | |
| 3.2 Navigation Bar | Only a logged in user sees the link to `/progress` | 3 | |
| 3.2 Navigation Bar | The links to `/register` and `/login` are replaced by a logout button | 2 | |
| 3.2 Navigation Bar | The logout button works | 2 | |
| **3.3 Homepage** | | | 16 |
| 3.3 Homepage | The homepage shows a list of available books that come from the API | 4 | |
| 3.3 Homepage | The books are ordered alphabetically on their name by default | 3 | |
| 3.3 Homepage | There are buttons to `Sort by name` and `Sort by popularity`. These buttons work | 3 | |
| 3.3 Homepage | Every book is displayed with the number of bookProgresses they have and their name | 4 | |
| 3.3 Homepage | Clicking on an book's name redirects you to the detail page for that book | 2 | |
| **3.4 Book Details Page** | | | 13 |

| Part | Goal | Points | Total |
|---|---|---|---|
| 3.4 Book Details Page | The detail page shows the books information | 3 | |
| 3.4 Book Details Page | The data comes from the API | 2 | |
| 3.4 Book Details Page | The detail page shows the amount of bookProgresses belonging to this book | 3 | |
| 3.4 Book Details Page | The detail page shows the book's average `pageProgress` | 5 | |
| **3.5 Create a BookProgress** | | | 9 |
| 3.5 Create a BookProgress | There is a button that creates a bookProgress on the detail page | 2 | |
| 3.5 Create a BookProgress | Clicking the button creates a BookProgress in the database with the correct `userId` and `bookId` and `pageProgress` | 4 | |
| 3.5 Create a BookProgress | The button is only shown if the user is logged in | 3 | |
| **3.6 The Progress Page** | | | 10 |
| 3.6 The Progress Page | The progress page shows a list of all the books that have a BookProgress for the current User | 2 | |
| 3.6 The Progress Page | Every book shows the amount of pages read and the total amount of pages | 4 | |
| 3.6 The Progress Page | The page can only be seen when a user is logged in | 4 | |
| **3.7 Modify Progress Forms** | | | 9 |
| 3.7 Modify Progress Form | The progress page displays a a form for every book | 5 | |
| 3.7 Modify Progress Form | Submitting the form with a button changes the `pageProgress` for the associated BookProgress | 4 | |
| **3.8 Backend BookProgress validation** | | | 6 |
| 3.8 Backend BookProgress validation | The `PATCH /bookprogress/:id` endpoint does not allow to set the `pageProgress` value higher than the associated Book's `pageCount` value | 6 | |

| Part | Goal | Points | Total |
|---|---|---|---|
| **4.1 Code Organization** | | | 5 |
| 4.1 Code Organization | The commit messages are meaningfully, there is not only one single commit | 3 | |
| 4.1 Code Organization | The `README.md` contains a description of the project | 2 | |
| **5.1 Self Reflection** | | | 5 |
| 5.1 Self Reflection | The `REFLECTION.md` contains a self reflection by the student | 5 | |