



Thank you for taking the time to complete this take-home coding challenge as a part of the interview process for joining Ovarc. The coding challenge is a standard real-life example of how we can build a front-end application using REST APIs. We will assess your code carefully and run the code to make sure the requirements are met. Please pay close attention to the requirements and your deliverables.

Introduction

We are building the front-end of an API for a bookstore. The purpose is to expose the bookstore for external public use where other bookstores or print houses can use it to perform CRUD operations on the books. The main entities are the store, book & author. The intention is that this tool is to be used by an admin who is managing multiple stores and can list books in different stores with different prices.

Requirements

- Build a frontend application using React
- Use the sample JSON objects as stubs for the API endpoints.
- The design for that app is available through this Figma link <https://figma.fun/mVEdR8>
- The design consists of the below pages and their features
 - **Shop** page which has a list of cards containing the book cover page, title & author, and which stores this book is available in. The sell button should mark this as sold but keep the card on the page.
 - **Authors** page which has a simple list of authors and two CTAs to edit the name (in-line edit) or delete the author entirely. There is a CTA & a modal too for adding a new author.
 - **Books** page which has a list of books, the number of pages, and who the author is. The edit CTA should also be an in-line edit for the book title.
 - **Stores** page. Same as the above two. The entire row should be a CTA for the next page.
 - **Store inventory** page. This is where the admin adds more books to the store's inventory. The books are viewable either in a list view or grouped by the author via the tab selection. The add to inventory CTA pops up a modal as shown in the Figma.
- The frontend should be containerized using docker (nice to have)
- It should also use docker-compose to start it (nice to have)



Deliverables

You are required to submit this exercise within 8 hours of receiving it. We request that you report the actual time spent on delivering these requirements. The coding challenge should be submitted back as a public repository on any well-known platform such as GitHub, bitbucket, or popular alternatives. Once the code is pushed to the repo and it is ready, please reply to the email with the repository link. A **Readme must** be included in the repository explaining how to get the containers up and running, or how to start the app if it is not containerized.

If you have any questions, please feel free to reply to the email with the coding challenge.

Good luck and hope to see you onboard soon.