Welcome to XYZ Books

XYZ Books is planning to develop an application that can get information from students, college textbooks, and other course material. It has functionalities like displaying applicable product data (e.g., authors, publishers, etc) or converting an ISBN (i.e., International Standard Book Number) to and from different formats.

As a full-stack engineer, you would be responsible for contributing to the front and back end of any system.

Front End. You would be responsible for implementing features that deliver this functionality through tasks such as UI design and styling, API integrations, and more. You are expected to accomplish the following:

1. **Display.** Create a UI that displays information about a particular book given its ISBN-13 in the URL path. As part of this task, your code should handle the following:
    - The styling should match the provided mockup as closely as possible.
    - Discrete sections should be structured as separate components.
    - The page should be responsive (i.e., elements are scaled properly based on variable width).

2. **Form Validation.** Add an input field (e.g., search bar) to item #1 for navigating to other book pages.

    - This element should accept both ISBN-13 and ISBN-10 values.
    - For invalid ISBN values, you should display an error message on the page (e.g., “Invalid ISBN”). In that scenario, you should not call the aforementioned product API. Therefore,
    - you must implement your logic to determine if a provided value is an invalid ISBN (Hint: You will likely need to utilize some of the code from item #3 below).

3. **Logic Formulation.** Implement a function that converts an ISBN-13 to an ISBN-10 (and vice versa).
    - For this logic, you will need to have a basic understanding of how ISBN check digits are computed and used.
    - We recommend you reference the Wikipedia page for “International Standard Book Number.” In particular, please see the “Check digits” section (which can be found here).
    - You are free to handle error cases in several different ways. However, we expect your code will not trigger error logging that would be visible to the client in their browser console.

Back-End. You would be responsible for implementing features that deliver this functionality through tasks such as code design using ORMs, API creation, and more. You are expected to accomplish the following:

1. **Schema Development.** Create both a schema and corresponding classes to represent an author, book, and publisher with the following constraints:

TABLE	ATTRIBUTES
Author	Should always have a first name and last name.
	Can have a middle name/initial, but is not required.
	Can have many books
Book	Should always have a title, ISBN-13, list price, publication year, and publisher.
	Can have an image URL and edition.
	Must have at least one author but can have many authors.
Publisher	Should always have a name
	Can have many books

For demo and testing purposes, we will provide the underlying data for 5 books (and their authors/publishers). However, you are expected to provide the necessary SQL (or equivalent) database statement for adding this data to your data store.

Title	Author	ISBN 13	ISBN 10	Publication year	Publisher	Edition	Price
American Elf	Joel Hartse, Hannah P. Templer, Marguerite Z. Duras	978-1-891830-85-3	1-891-83085-6	2004	Paste Magazine	Book 2	1000
Cosmoknights	Kingsley Amis	978-1-60309-454-2	1-603-09454-7	2019	Publishers Weekly	Book 1	2000
Essex County	Kingsley Amis	978-1-60309-038-4	1-603-09038-X	1990	Graywolf Press		500
Hey, Mister (Vol 1)	Hannah P. Templer, Fannie Peters Flagg, Camille Byron Paglia	978-1-891830-02-0	1-891-83002-3	2000	Graywolf Press	After School Special	1200
The Underwater Welder	Rainer Steel Rilke	978-1-60309-398-9	1-603-09398-2	2022	McSweeney's		3000

2. **Implementation**. Provide an API endpoint to display information about a particular book given its ISBN-13 (i.e., via a GET call to an API where the ISBN-13 is provided in the URL path as the id). The response body should be in JSON format for all successful (i.e., status code 200) responses.

   For this task, we expect the following:

   - All authors of the book should be combined and provided as a single, comma-separated string (e.g., “author 1, author 2, author 3”).
   - The publisher’s name should be provided as an unnested object.
   - The response code for a book that does not exist should be 404.
   - The response code for a request with an invalid ISBN-13 should be 400. For this condition, you will likely need to utilize some of the code from item #3 below.

3. **Logic Formulation**. Implement an endpoint that converts an ISBN-13 to an ISBN-10 (and vice versa), with the converted value returned in the response body.
    - For this logic, you will need to have a basic understanding of how ISBN check digits are computed and used. We recommend you reference the Wikipedia page for “International Standard Book Number.”
    - In particular, please see the “Check digits” section (which can be found here).
    - You are free to handle error cases in several different ways. However, we expect that your code will not throw exceptions that lead to 5xx responses.

Deliverables

Outside of these requirements, you are free to implement your solution however you see fit. However:

- You must, however, provide instructions on how to run and execute your code.
- Additionally, unit or functional tests are not required but are **recommended.**
- Your code should be delivered via a repository on GitHub.
- It must be runnable from the clone without modification and only by following the instructions you provide in your documentation.

Testing

The code will be tested on either macOS Monterrey or Ubuntu 22.04 LTS. Please minimize external dependencies that require additional installation beyond setting up your project. An in-memory or file-based database is preferred (although MySQL or PostgreSQL are also acceptable)

Performance Task Rubrics

Criteria	4-Advance/Excellent	3-Proficient	2- Developing	1-Beginning	0 - Incomplete
Display (15%)	The UI displays information about a particular book given its ISBN-13 in the URL path. The styling and responsiveness are beyond satisfactory level.	The UI displays information about a particular book given its ISBN-13 in the URL path. The styling does match the provided mockup and the page is responsive.	The UI displays information about a particular book given its ISBN-13 in the URL path but the styling does **not** match the provided mockup **OR** the page is **not** responsive.	The UI displays information about a particular book given its ISBN-13 in the URL path but the styling does **not** match the provided mockup **AND** the page is **not** responsive.	The UI does not display information
Code Construction (15%)	The code is well organized and demonstrates an understanding of the software development life cycle beyond a satisfactory level.	The code is well organized and demonstrates an understanding of the software development life cycle.	The code demonstrates an understanding of basic software principles but does not follow all best practices.	The code organization and structure follow basic language rules (e.g. using objects in an object-oriented language)	Code is poorly constructed.
Form Validation (10%)	The input field is added and works beyond a satisfactory level.	Input field is added and able to show corresponding form response.	The input field is added but the response is not correct.	Input field is added but not able to show corresponding form response.	The input field is added but does not show any response.
Logic Formulation (10%)	The conversion function is working beyond a satisfactory level.	The conversion function is working.	Some part of the conversion function is correct	The conversion function is incorrect	The conversion function is incomplete
Explanation / Justification (50%)	Explanation demonstrates a strong understanding of the problem.	Explanation demonstrates a good understanding of the problem	Explanation demonstrates some understanding of the problem	Explanation demonstrates limited understanding of the problem	The candidate was unable to offer any explanation.