**Final Project: booksmart**

**Zeke Van Dehy**

**Homepage / index**

**Graphical user interface, application, website

Description automatically generated**

**About**

**Graphical user interface, text, application, website

Description automatically generated**

**Contact**

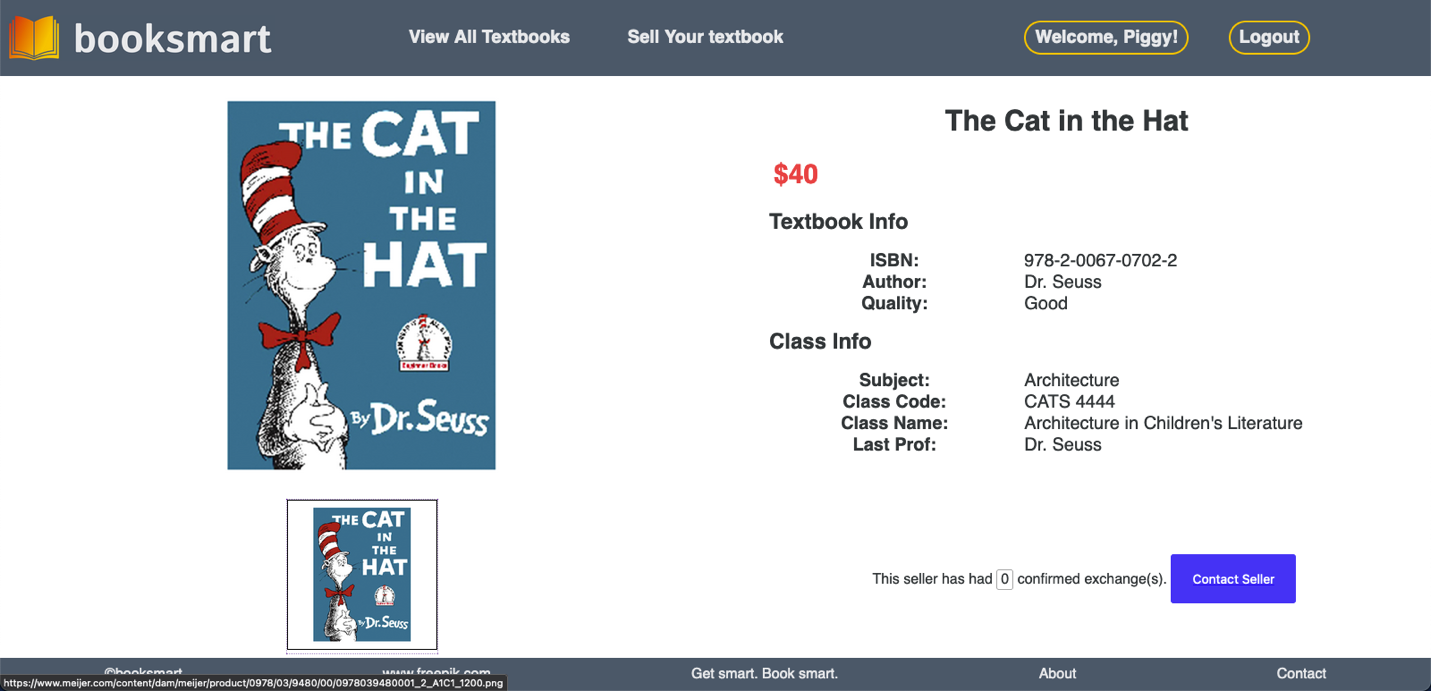
**Graphical user interface, application, website

Description automatically generated**

**All textbooks (connections)**

**Graphical user interface, Teams

Description automatically generated**

**Show textbook**

**Graphical user interface, application

Description automatically generated**

**Edit textbook**

**Graphical user interface, application

Description automatically generated**

**Create textbook**

**Graphical user interface

Description automatically generated**

**RSVP/Inquire about textbook**

**Graphical user interface, application, Teams

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Confirmation from both parties**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application, Teams

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Profile displays textbooks, inquiries, and sales**

**Graphical user interface, application, Teams

Description automatically generated**

**Explanation of additional features**

Instead of RSVP system, I implemented a messaging system that allows members to inquire about a textbook that was listed on the site. This process involves creating an inquiry object for the textbook (which is basically equivalent to RSVP) and message objects for the inquiry. A user cannot create an inquiry for their own textbook. Users can only create one inquiry for each textbook listing, but multiple users can create one inquiry (each) for a single textbook. Once both the inquirer and the owner of the textbook have confirmed the sale/exchange, then the textbook is listed as “sold” on their profiles and it is no longer displayed on the /textbooks page. When a textbook is sold all of the inquiries for that textbook are deleted. When an inquiry is deleted (including when it is rejected by the owner), all messages are also deleted. When a user is viewing a textbook that isn’t theirs, they can see how many successful exchanges (buy or sell) the owner of the textbook has completed. Owners with more exchanges are more reliable because they have a confirmed history of successful exchanges.

**Explanation of status**

All of the required components are present and working. Uploading a photo, or even assigning two images currently does not work.

**Discuss challenges**

The most difficult part of this project was deleting objects that are associated with other objects. For example, it was difficult to delete all of the message objects when a textbook was sold or deleted. This was difficult because I had to think critically about how to avoid race conditions or for loops to delete all of the objects. More specifically, I couldn’t use “post” middleware to detect when an inquiry was deleted because the “deleteMany” middleware doesn’t return the ids of the deleted objects. I had to use multiple “.then()” clauses to delete all of the messages and all of the inquiries when a textbook was sold or deleted.