CS 839 Project Stage 2 Report

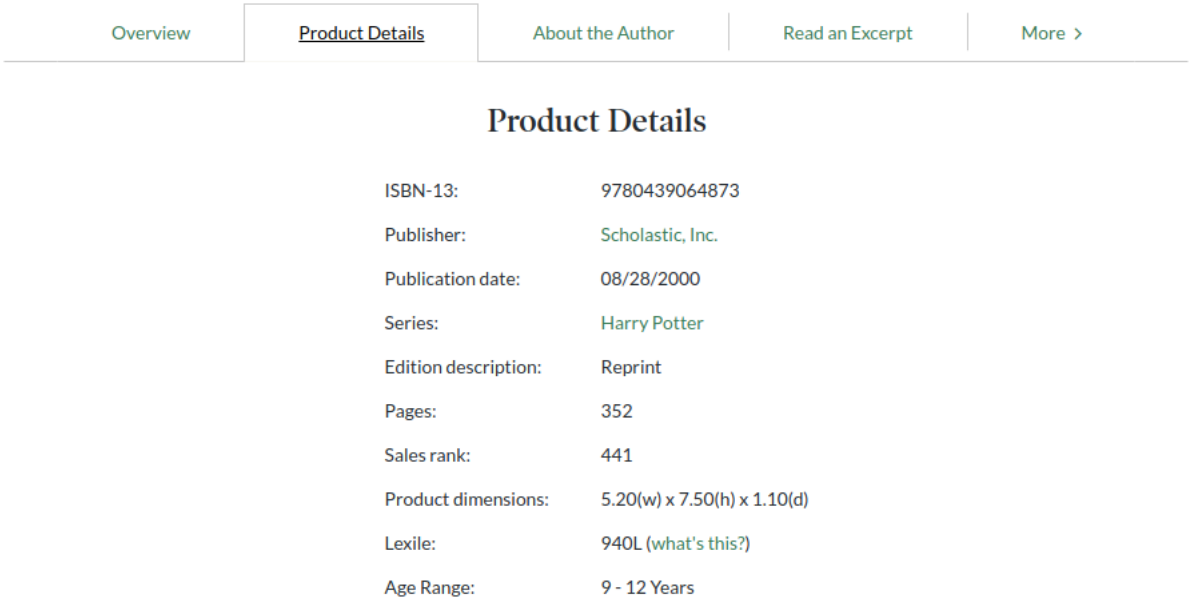
Zeping Ren, Stephen Sheen, Shiqi Yang

Data Sources

We extracted paper-based book selling information from two online book-selling websites:

- Barnes & Noble: `https://www.barnesandnoble.com`
- Better World Books: `https://www.betterworldbooks.com`

Both websites contain the information about the books. For example, if we look up "Harry Potter and the Chamber of Secrets", Barnes & Noble Website will show the following information on its web page:



And Better World Books shows the following information:

Data Extraction Techniques

As mentioned above, both websites display the information with a structured HTML element. Specifically, they use tables to display the details. So we can make use of this to extract the information we need.

Given a web page (as an url), our web crawler converts the web page into an HTML string. If the web page is that of a book, it then extracts the title, the author(s), and the table containing the details of the book. It also extracts any links to other book web pages and adds those into a list of links to explore. Our web crawler repeatedly does this in a depth-first search fashion.

Entity Type

We collect book details information from both websites and obtained the two tables with the following schema:

```
A(title,author,isbn,publisher,publication date,series,edition,pages,
  sales rank,dimensions,lexile,age range)
B(title,author,form,pages,language,publisher,edition,isbn13,dimension,
  isbn10,weight,category)
```

Then we take the intersection of the schema and have the common schema of the following form:

```
A, B(title,author,isbn,publisher,edition,pages,dimension)
```

Our data contains 3600 tuples per table.

Open Source Tools

We used the Python built-in libraries (`urllib`, `html.parser`) to write our web crawler. These two libraries basically provide essential HTML source parsing functionalities.