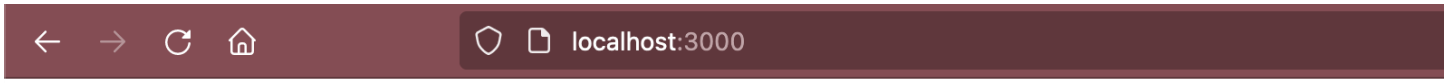
Name: Andrew Parsons
TMU ID: 500992021
Due Date: Apr 1, 2023
Github Repo Link: https://github.com/Parsonswlu/parsonswlu.github.io/tree/main/CCPS530_Lab6

CCPS 530 - Web Systems Development - Lab 6

Website Screenshots

- localhost:3000/



Andrew Parsons Book Inventory

- Current Book List
- Add a Book

Home Page

- localhost:3000/bookinventory/list

Andrew Parsons Book Inventory

- Home Page
- Add a Book

Current Book List:

Title: Deep Learning for Coders with fastai & PyTorch: AI Applications Without a PhD
Author: Jeremy Howard & Sylvain Gugger
Publisher: O'Reilly Media, Inc.
Date: 2021-11-05
Website: https://course.fast.ai/Resources/book.html

Title: Quantum Computation and Quantum Information: 10th Anniversity Edition
Author: Michael A. Nielsen and Isaac L. Chuang
Publisher: Cambridge University Press
Date: 2011-01-30
Website: https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE#overview

Title: Spatial Computing
Author: Shashi Shekhar and Pamela Vold
Publisher: The MIT Press
Date: 2020-02-18
Website: https://mitpress.mit.edu/9780262538046/spatial-computing/

Title: Machine Learning Engineering
Author: Andriy Burkov
Publisher: True Positive Inc.
Date: 2020-09-08
Website: http://www.mlebook.com/wiki/doku.php

Title: Principles of Computer Architecture
Author: Miles J. Murdocca, Vincent P. Heuring
Publisher: Prentice-Hall
Date: 1999-11-29
Website: https://academicos.azc.uam.mx/oan/lac/Murdocca_en.pdf

- localhost:3000/bookinventory/add

Andrew Parsons Book Inventory

- Home Page
- Current Book List

Insert a book:

Title: title1
Author: author1
Publisher: publisher1
Date: date1
Website: website1
Submit

- localhost:3000/bookinventory/addbook

Andrew Parsons Book Inventory

- Home Page
- Current Book List
- Add Another Book

Title: title1 is added!

- localhost:3000/bookinventory/list (again)

Andrew Parsons Book Inventory

- Home Page
- Add a Book

Current Book List:

Title: Deep Learning for Coders with fastai & PyTorch: AI Applications Without a PhD
Author: Jeremy Howard & Sylvain Gugger
Publisher: O'Reilly Media, Inc.
Date: 2021-11-05
Website: https://course.fast.ai/Resources/book.html

Title: Quantum Computation and Quantum Information: 10th Anniversity Edition
Author: Michael A. Nielsen and Isaac L. Chuang
Publisher: Cambridge University Press
Date: 2011-01-30
Website: https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE#overview

Title: Spatial Computing
Author: Shashi Shekhar and Pamela Vold
Publisher: The MIT Press
Date: 2020-02-18
Website: https://mitpress.mit.edu/9780262538046/spatial-computing/

Title: Machine Learning Engineering
Author: Andriy Burkov
Publisher: True Positive Inc.
Date: 2020-09-08
Website: http://www.mlebook.com/wiki/doku.php

Title: Principles of Computer Architecture
Author: Miles J. Murdocca, Vincent P. Heuring
Publisher: Prentice-Hall
Date: 1999-11-29
Website: https://academicos.azc.uam.mx/oan/lac/Murdocca_en.pdf

Title: title1
Author: author1
Publisher: publisher1
Date: date1
Website: website1

Technical Report

1. Explain your design choices and what you used to test the RESTful endpoints from numbers 4 and 5.

- Design-wise, I wanted to ensure that I had my name on each of the web pages and an unordered list of links to the other web pages to navigate back and forth
- This involved prepending a HTML string at the beginning of each res.send() response, like the following for the Home Page:

```
app.get('/', function (req, res) {
 res.send('<h1>Andrew Parsons Book Inventory</h1><ul><li><a href="/bookinventory/list">Current Book
List</a></li><li><a href="/bookinventory/add">Add a Book</a></li></ul><h2>Home Page</h2>');
})
```

- For testing the 'Add' functionality at endpoint **/bookinventory/add**, I created a form that would allow me to add each of the values of the book's JSON keys in a text box and then submit those values to a second endpoint, **/bookinventory/addbook** :

```
app.get('/bookinventory/add', function (req, res) {

 var html = '<br><form action="/bookinventory/addbook" method="post"><label
for="title">Title:</label><input type="text" id="title" name="title"><br><label
for="author">Author:</label><input type="text" id="author" name="author"><br><label
for="publisher">Publisher:</label><input type="text" id="publisher" name="publisher"><br><label
for="date">Date:</label><input type="text" id="date" name="date"><br><label
for="website">Website:</label><input type="text" id="website" name="website"><br><input
type="submit" value="Submit"><br></form>'

 res.send('<h1>Andrew Parsons Book Inventory</h1><ul><li><a href="/">Home Page</a></li><li><a
href="/bookinventory/list">Current Book List</a></li></ul><h2>Insert a book: </h2>' + html);
});
```

- One the form data was passed to the endpoint **/bookinventory/addbook**, it would extract each form component and populate a new json object, which would be pushed onto the master list of book JSON objects.

```
app.post('/bookinventory/addbook', function (req, res) {
 var new_json = {
   'title': req.body.title,
   'author': req.body.author,
   'publisher': req.body.publisher,
   'date': req.body.date,
   'website': req.body.website
 };
 bookinventory.push(new_json);
 res.send('<h1>Andrew Parsons Book Inventory</h1><ul><li><a href="/">Home Page</a></li><li><a
```

```
href="/bookinventory/list">Current Book List</a></li><li><a href="/bookinventory/add">Add Another
Book</a></li></ul><h2>Title: ' + req.body.title + ' is added!</h2>');
}

);
```

- For testing the 'List' functionality at endpoint **/bookinventory/list**, I created a for loop that iterated over each book object in the master list of book JSON objects, and then output each component on a separate line next to the name of that component (i.e. for book JSON object i, **'Title: ' + bookinventory[i].title** would output the title of that book):

```
app.get('/bookinventory/list', function (req, res) {

 var html = '<p>'
 for (var i = 0; i < bookinventory.length; i++) {
   html = html + 'Title: ' + bookinventory[i].title + '<br>';
   html = html + 'Author: ' + bookinventory[i].author + '<br>';
   html = html + 'Publisher: ' + bookinventory[i].publisher + '<br>';
   html = html + 'Date: ' + bookinventory[i].date + '<br>';
   html = html + 'Website: ' + bookinventory[i].website + '<br>';
   html = html + '<br>';
 }
 html += '</p>'

 res.send('<h1>Andrew Parsons Book Inventory</h1><ul><li><a href="/">Home Page</a></li><li><a
href="/bookinventory/add">Add a Book</a></li></ul><h2>Current Book List: </h2>' + html);
});
```

2. How did you consume and parse JSON?

- First, the command **app.use(express.json());** was input at the top of the javascript file in order to support JSON-encoded bodies
- After that, it was as simple as creating a consistent JSON structure for each book object, in terms of having identical and required set of object keys (title, author, publisher, date, website)
- Each value in the book JSON object could then be accessed directly via the notation **object.key**, where **key** represents the key component of the key:value pair within the object
- Each JSON object was placed into an array of JSON objects entitled **bookinventory**
- When a new book JSON object needed to be added, it was created using a form submission where each text field populated would form one of the values for the necessary key:value pairs
- Once a new JSON object was created, it was pushed onto the array variable **bookinventory** via the command **bookinventory.push(new_json);**

3. How long did you spend on this lab? The length of time includes readings and research and code experimentation. State time involved in readings and research as well as code experimentation sessions.

- I spent approximately 1.5 hours working on this lab.

- It took about 1 hours to update the code from the starter file **app.js** we used in class, extending the functionality to reference multiple aspects of a JSON object rather than just a **user** key, as well as add the necessary ornamentation of including my name and cross-linking the respective endpoints.
- It then took about another 30 minutes to do the validation, take screenshots, write up the technical report, etc.