

ITE5315 Assignment 1

Submission Deadline: Wednesday Oct 5, 2022@ 11:59pm

Assessment Weight: 5% of your final course Grade

Objective: To build simple Node/Express app which demonstrate the concept of handling route and forms.

Description: Using the examples of week1~3, develop a new Node/Express app program that:

- Load the sample JSON dataset
- Handle requested Route and Form
- Demonstrate best practices in ES6+ coding by using proper variable naming, commenting, error handling strategy wherever is required

Complete the following steps:

Step1: Create new Node/Express app, named “Asn1-yourname”.

Step2: Create a new server.js file, listen to the port 5500 and display your name/student ID when user opens the home page (<http://localhost:5500/>)

Step3: Modify server.js by handling wrong route (404). So incase user open a wrong route (<http://localhost:5500/wrongRoute>) , the program displays proper message and return proper status_error.

Step4: Modify server.js by loading the attached JSON file in the app when user access to the route : <http://localhost:5500/data/>

- Use the example of loading local files in NodeJS app, modify the server.js to load the given JSON file
- Add a new route, when user opens the route, show/send a message “JSON data is loaded and ready!” and display also the entire JSON data in the console.
- Since the dataset is big, what would be the best practice to improve the performance? Explain your ideas.

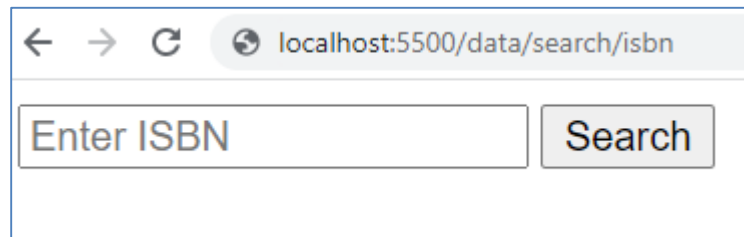
Step5: Modify server.js by handling a new route : <http://localhost:5500/data/isbn/{index}> in which when user enter the index_no, it displays the related book_isbn item in the list (in JSON data).

i.e:

- If user enter <http://localhost:5500/data/isbn/0> , it displays 044310073X
- If user enter <http://localhost:5500/data/isbn/1> , it displays 1438005687
- *Hint: use params!*

→ You also need to handle the errors in case user enter a wrong {index} value.

Step6: Modify server.js by handling a new route : <http://localhost:5500/data/search/isbn/> in which displays the following form:

A screenshot of a web browser window. The address bar shows 'localhost:5500/data/search/isbn'. Below the address bar, there is a search form with a text input field containing the placeholder text 'Enter ISBN' and a button labeled 'Search'.

When user enter the ISBN, the program searches the given JSON data (you need to loop through the JSON data that has been loaded in Step4 loaded JSON data) and displays the related book_info (from the given JSON data), if the given ISBN is existed. If the given ISBN is not existed, the program displays proper error.

Step7: Modify server.js by handling a new route : <http://localhost:5500/data/search/title/> . This is similar to Step6, however, you are asked to get book_title using the form and display related books in the output.

- Try to use JavaScript *String.include()* methods to implement “*partial search*”, so if user enter “Agile” the program displays all books which have “Agile” anywhere in their title. (Hint: study this: https://www.w3schools.com/jsref/jsref_includes.asp)
- If the program finds (a) book(s) with related book title, it displays the minimum of three fields about the book (like ISBN, Title,...)

Step8: Assume that you are in an interview and you are asked to explain your idea in using *middleware and next()* in this application (specially in implementing search-feature on different fields). You may explain this using your own word or code!

Assignment Submission:

- Add the following declaration at the top of .js files

```
/******  
***  
* ITE5315 – Assignment 1  
* I declare that this assignment is my own work in accordance with Humber Academic Policy.  
* No part of this assignment has been copied manually or electronically from any other source  
* (including web sites) or distributed to other students.  
*  
* Name: _____ Student ID: _____ Date: _____  
*  
*  
*****  
**/
```
- Compress (.zip) the files in your Visual Studio working directory (this is the folder that you opened in Visual Studio to create your client side code).

Important Note:

- Submitted assignments must run locally, ie: start up errors causing the assignment/app to fail on startup will result in a **grade of zero (0)** for the assignment.
- **LATE SUBMISSIONS for assignments.** There is a deduction of 10% for Late assignment submissions, and after three days it will grade of zero (0).
- Assignments should be submitted along with a video-recording which contains a detailed walkthrough of solution. Without recording, the assignment can get the maximum of 1/3 of the total.