# Briefly describe the artifact. What is it, and when was it created?

The artifact, Global Traveler, was created some time before the commencement of CS499. It is a travel website that I intend to turn into a real-life blog when I finish my degree.

# Justify the inclusion of the artifact in your ePortfolio. Why did you select it? What specific components of the artifact showcase your skills and abilities in algorithms and data structure? How was the artifact improved?

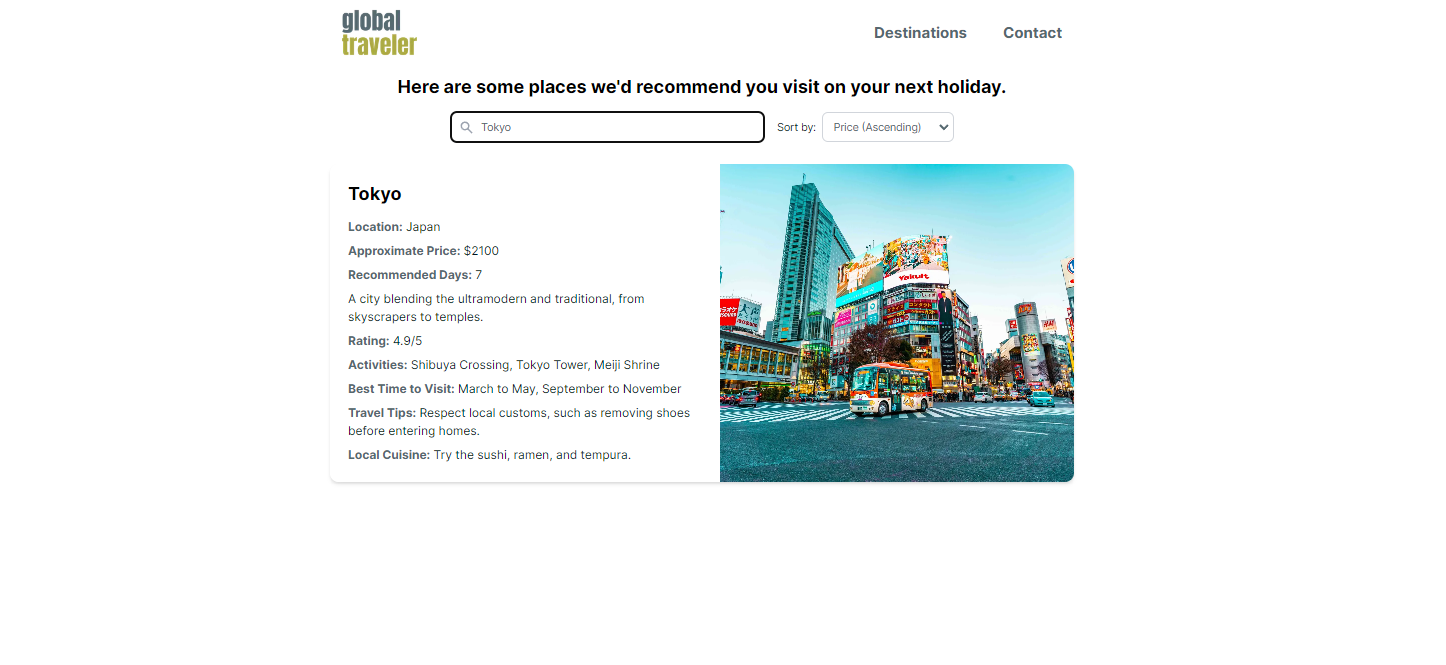
This project is a full-stack application above anything else. One of the functions that I wanted to incorporate into my blog was the destinations tab, where people could see information like where I visited, how much it cost (approximately) and what the best time to visit would be. The destinations page is what I enhanced for this category. Initially, it had a search bar that did not necessarily work to filter out according to what the user typed. This has since changed on my application.

1. Destinations page (https://global-traveler.vercel.app/destinations).

A screenshot of a travel website

Description automatically generated

1. Filter for ‘Tokyo’.



1. Code for quick sort.

A computer screen shot of code

Description automatically generated

Furthermore, I also incorporated the Quick Sort algorithm in place of the ordinary JavaScript sorting mechanism. This allows people to view the destinations by approximate price in ascending and descending order. Hopefully, this enhancement will make the app better in terms of its purpose in helping people choose destinations that fit their budget.

# Did you meet the course outcomes that you had planned in Module 1? Are there any updates to your outcome-coverage plans?

**Course Outcome 3**: Design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices.

I have met the outcome here given that I chose to use Quick Sort over the more common JavaScript sorting algorithm. Quick Sort gives me more control over how the sorting is performed, and it allows for custom optimization based on the data that I am trying to sort. On the other hand, the traditional JavaScript array.prototype.method depends heavily on the use case. Given that the list of countries I am looking to visit will undoubtedly grow, I am anticipating that I will have to eventually sort a greater number of items, therefore the Quick Sort.

No updates to my course outcome coverage, however, for enhancement three, a user authentication will be added to protect the API.

# Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?

The last time I really touched any Quick Sort or anything similar was in IT140 when I was working on a Python project. Therefore, it was quite challenging to remember how to apply it to my case. It took a number of tries to get it right, and I was afraid that at any given point I was going to render my app completely useless. The one useful lesson that I learned, however, was not to push to GitHub to deploy (Global Traveler is deployed on Vercel) too quickly - and to test first.