My major take-ways from this project are the following:

* **Pointers**

I learnt through experience that pointers can be tricky to work with. I learnt that a pointer can be of any type, and when we create our own classes, we can create pointers of type said class.

I learnt that an asterik can either be used to create a pointer or dereference a pointer. I learnt that the ampersand symbol is used to get the memory address of an object. I learnt that putting an asterik before a variable during assignment, makes it store a reference to the memory address of the object it is being assigned to.

* **Classes, Inheritance, Polymorphism**

I understood that an object can not have a member variable which stores an object of the same class. We need to create a pointer to that object in order to compile successfully. For example, when we have a node supposed to store a copy of it’s parent or it’s child, we must instead store a pointer to those objects.

* **Vectors**

Vectors store objects in contigous memory, while lists store objects in non-contigous memory. Lists are like linked lists and do not support random access of elements in the list. Vectors can be accessed using indexes, they are dynamic, synchronized and capable of resizing themselves.

* **Character format in c++**

Reading from a csv in c++ returns each word by identifying the commas. So if there is no comma separating two words in a csv, it is returned as one word.

* **Fairness and business optimality**

I used a depth-first search to ensure that different airlines appeared in the search process, not just one airline.