1. Provide screenshots to show the status of all packages loaded onto each truck at a time between 8:35 a.m. and 9:25 a.m.

A screenshot of a computer program

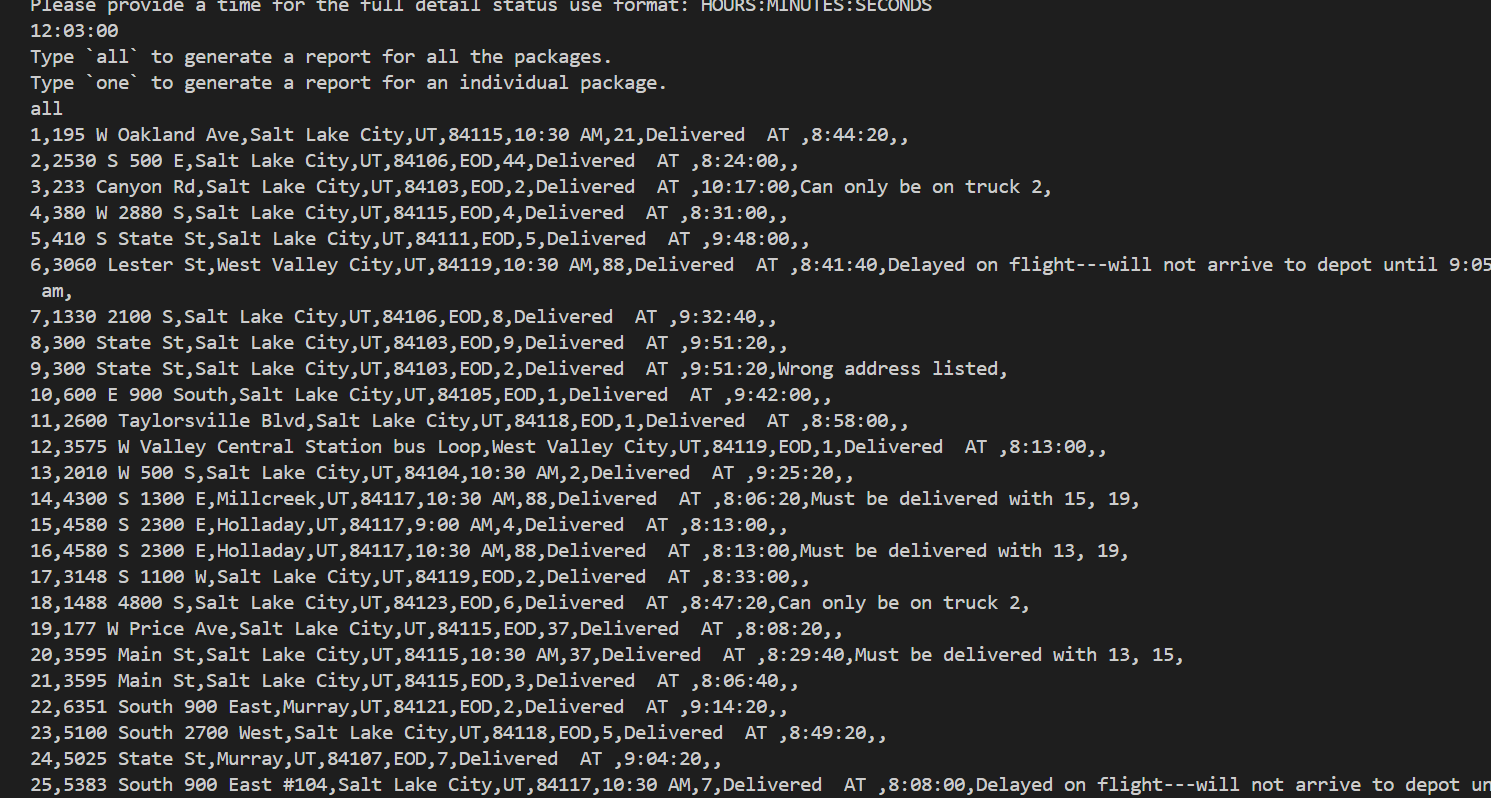
Description automatically generated

1. Provide screenshots to show the status of all packages loaded onto each truck at a time between 9:35 a.m. and 10:25 a.m.

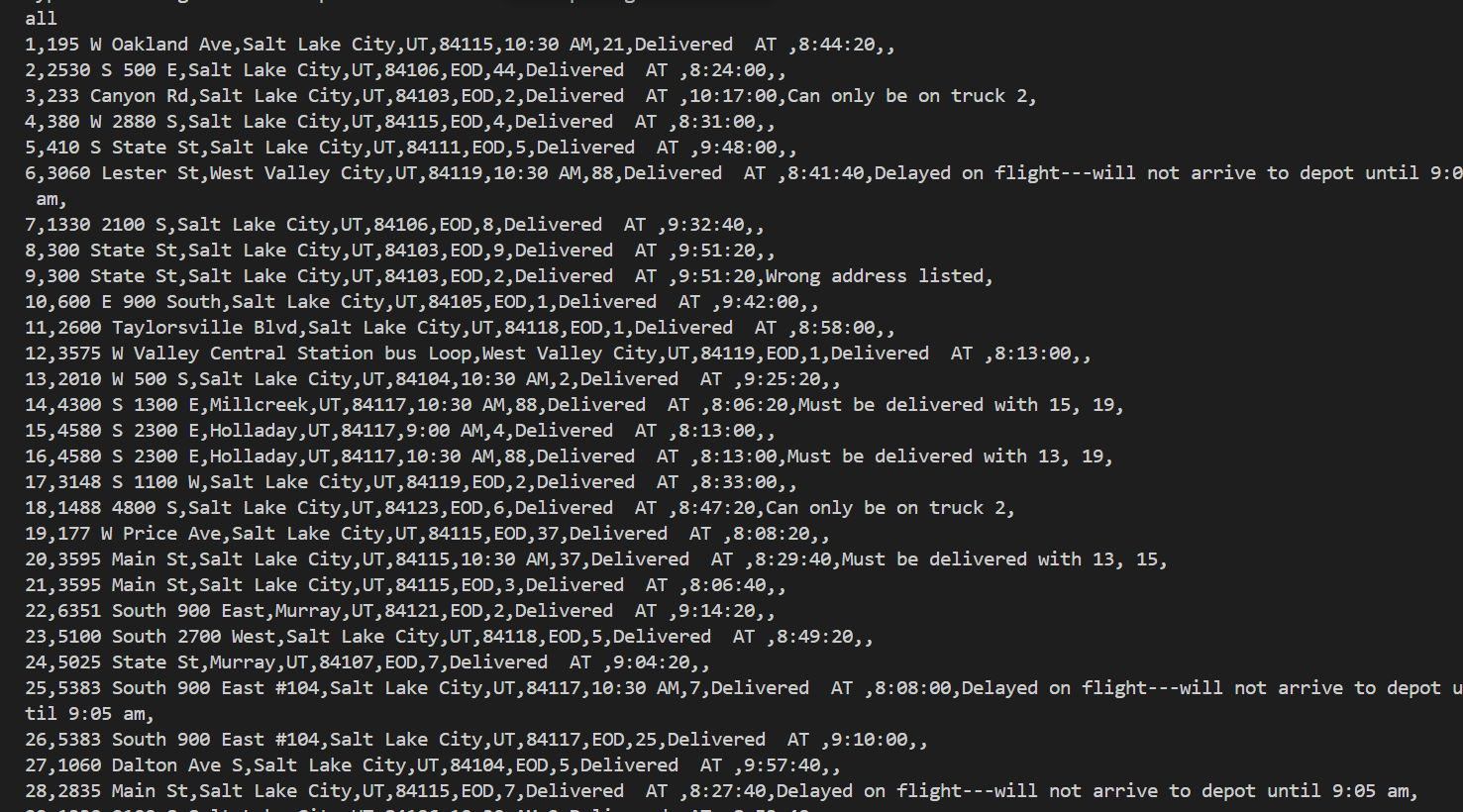
A screen shot of a computer screen

Description automatically generated

3.  Provide screenshots to show the status of all packages loaded onto each truck at a time between 12:03 p.m. and 1:12 p.m



E.  Provide screenshots showing successful completion of the code that includes the total mileage traveled by all trucks.



A screen shot of a computer

Description automatically generated

F.  Justify the package delivery algorithm used in the solution as written in the original program by doing the following:

1.  Describe **two or more** strengths of the algorithm used in the solution.

First strengths is that Nearest neighbor algorithm is less sensitive to noisy data compared to other algorithms. It can still provide reasonable results even if the dataset contains outliers or errors. Second, unlike many other machine learning algorithms that require a training phase, the nearest neighbor algorithm does not have a training phase. This makes it quick and easy to implement, especially for small to medium-sized datasets.

1. Verify that the algorithm used in the solution meets all requirements in the scenario.

The Nearest neighbor algorithm meets all of the requirements for this scenario because it gets packages by selecting the shortest path but introducing greedy algorithm method and most importantly it was able to accomplish the total miles under 140 miles.

1. Identify **two** other named algorithms that are different from the algorithm implemented in the solution and would meet all requirements in the scenario.

The two that stands out is Dijkstra’s algorithm and prims problem these are greedy as well and would work with the scenario.

1. Describe how both algorithms identified in part F3 are different from the algorithm used in the solution.

Dijkstra algorithm outputs the shortest path form the source node to all other nodes in the graph and Nearest Neighbor algorithm outputs the tour that visits each city exactly once and returns to the starting city, but it may not be the best optimal solution.

Now comparing NN to Prims algorism Prim’s algorithm outputs the minimum spanning tree of the input graph while nearest neighbor could be primarily used for solving the traveling salesman problem like initial project it finds the shortest path possible.

G.  Describe what you would do differently, other than the two algorithms identified in part F3, if you did this project again, including details of the modifications that would be made.

If I would do it differently, I would have all the csv objects and variables into another python file with another class to organize it a little better I would also use linked list with a hash table.

H.  Verify that the data structure used in the solution meets all requirements in the scenario.

The Hash table was the data structure used to meet all the requirements for the scenario. In order to complete each requirement we would have needed a key value pair, for example package ID along with the state,zip,city and status for the values that matches with the key.

1. Identify **two** other data structures that could meet the same requirements in the scenario.

Priority queue and a double linked list would also be a similar data structure to complete each scenario in the project.

1. Describe how each data structure identified in H1 is different from the data structure used in the solution.

A double linked list can use both forward and backward traversal and it stores each key with a pointer so allowing a easier way to retrieve data back this will help organize the each data set for state,zip,city and package ID similar to hash table it also can insert and delete.

For a Priority Queue: Elements are typically accessed based on their priority, with the highest (or lowest) priority element being accessed first similar to hash table priority queue also searches for a element based on its value or key helping with a faster search response and deletion using FIFO and it can also be implemented with arrays,linked list,heap, binary search tree giving you an efficient optimal solution.

1. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized.

GfG. “What Is Priority Queue: Introduction to Priority Queue.” *GeeksforGeeks*, GeeksforGeeks, 11 Jan. 2023, www.geeksforgeeks.org/priority-queue-set-1-introduction/.

GfG. “What Is Priority Queue: Introduction to Priority Queue.” *GeeksforGeeks*, GeeksforGeeks, 11 Jan. 2023, www.geeksforgeeks.org/priority-queue-set-1-introduction/.