# Project Checklist

## Programming / Functionality

* Couriers.txt would have destination, weight and valuation. The format of data stored in file would be destination, weight, value.
* Allocate memory for destination dynamically.
* Create a Hash table of size 127, each index of the HashTable will hold the root node of the Tree structure.
* use the “djb2 function” to generate the unique Hash Value
* Each node in BST represents a parcel.
* Each node in the BST is placed using parcel’s weight.
* menu:
  + 1. Enter country name and display all the parcels details: when user enters 1, application allow user to enter County name. Then display each parcel’s details.
  + 2. Enter country and weight pair: This menu offers user to display all the parcel for given country whose weight is higher/lower than weight entered.
  + 3. Display the total parcel load and valuation for the country: When user enters country name, display the cumulative parcel load and total valuation of all the parcels.
  + 4. Enter the country name and display cheapest and most expensive parcel’s details.
  + 5. Enter the country name and display lightest and heaviest parcel for the country.
  + 6. Exit the application.
* Clean up all allocated memory before exiting.
* The field containing the names must be dynamically allocated to an appropriate size.

## Submitting

* project.cpp
* checklist.pdf
* couriers.txt