Assignment 1

Instructions:

- Code will be executed using NodeJS. Prior to submitting, ensure your program works as expected by running the command: node index.js in a console window.

- Grading is based on implementing the required functionality and coding style, specifically: clearly organized code, appropriate variable naming, code readability, coding conventions demo'd in class, etc.

- Javascript syntax rules:

  - Variables must be declared using let/const, not var
  - Functions must be declared using arrow function syntax, not function() syntax
  - When checking equality, use strict equality (triple equals ===), **not** double equals (==)
  - Do **NOT** use higher order array functions: forEach, map, reduce, filter, closest, etc.

Submission Checklist:

1. Create NodeJS project

- ❏ Create a folder called A1**FirstName**. Replace **FirstName** with your name, example: A1**David**
- ❏ Inside the folder, create a new NodeJS project.
- ❏ Within your project's **package.json** file, update the **author** field with your name, and the **description** with a description of your project.
- ❏ Within the project, create a Javascript file called **index.js.** Put your solution code in the index.js file.

2. When you are ready to submit:

- ❏ Create a zip file containing your project folder.
- ❏ Rename your zip file A1FirstName.zip. Replace FirstName with your name, example: A1David.zip.
  - ❏ Ensure you use a zip file. Rar and 7zip files are not accepted.

Academic Integrity

- You are responsible for familiarizing yourself with the college's Academic Integrity Policy.

- This is an individual assessment

- Situations which often cause academic integrity issues:

  - Reposting any part of the assessment to online forums or homework help websites
  - Contract plagiarism: Purchasing a solution, or completing a solution for financial compensation
  - Sharing or receiving source code, references, or assistance from others

Problem Description:

Using NodeJS, create a standalone console-based Javascript program that lets the user reserve a car rental. The user must be able to enter the type of vehicle, number of days of the rental, and whether they want a child seat in the car. Based on this user input, the program should search for an available vehicle. If a vehicle is available, then calculate and output a receipt for the rental.

Specifically:

1. Create a variable that stores a list of cars. This list represents the cars owned by the car rental agency.
   - The list must contain **3 cars**.
   - Each car in your list must be represented as an **object literal**.
   - Each car in the list must have a **type** (string)**, license plate** (string), and **availability** (boolean). You should use different data for each car.

Here is an example of car data. You may customize this data to suit your program:

Type	**License Plate**	**Availability**
SUV	ABC 124	false
Sedan	BXL 009	true
SUV	KML 155	true

2. Create these functions:

   - findVehicle(*carType*): The function accepts the type of car as a **string** (example: "SUV", "Truck", "Sedan"). Using the car type, loop through the list of cars to search for a car that is **available** and also has the **same type**. If a matching vehicle is found, the function returns the **position of the vehicle in the array**. Otherwise, return **-1**.
     - Your search logic must be written in a way that would work for *any number of cars* in the list of cars.

   - createReservation(*reservationDetails*): The function accepts a **reservation details object literal**. This object contains information about the user's requested reservation. Using the information in the object literal:
     - Search the list cars for a matching vehicle (you must use the findVehicle() helper function)
     - If no matching vehicle is found, output an error message to the console
     - Otherwise, calculate the cost of the car rental and output a receipt. The receipt must contain a **randomly generated reservation id**, matching car type, license plate, reservation subtotal, tax amount, and final total to pay.
       - The reservation id must be generated using the **uuid** library (https://github.com/uuidjs/uuid#quickstart). Use the **CommonJS** syntax.

   The total cost of the rental car is calculated as: **(rental fee + car seat fee) + sales tax**

     - Rental fee: # of days requested * daily rate. Sedans are $10 per day, SUVS are $15 per day, and all other vehicle types are $20 per day.

- Car seat fee: If the user requests a car seat, then the fee is $5 per day. Rentals that are 3 or more days are charged at a rate of $2 per day.
- Sales tax is 13%.

3. After creating the list of cars and the helper functions, write a program that performs the following operations.

   a. Display a welcome message to the user
   b. Using the **prompt-sync** library, ask the user to enter the type of car they want to rent, the number of days, and whether they want a child seat (yes/no). You do **NOT** need to perform validation on the user input.
   c. Using the provided information, create a *reservation details object*.
   d. Using the reservation details object, call the createReservation() function to book a car for the user.

Format of JS file

NOTE: The order of code in your index.js file should be:

// imports from any required packages (prompt-sync, etc)
// list of cars declaration
// function declarations
// main program code

Sample output

Example 1: Successful Reservation

```
-------------------------------
Welcome to David's Car Rentals
-------------------------------
What type of car do you want to rent? SUV
How many days? (min 1): 3
Do you need a car seat? (y/n): y

-------------------
RECEIPT
-------------------
Reservation Number: 042ddaa1-5ff5-439f-a153-eb82c0ad7ad8
Car Type: SUV
License Plate: KML 155
Subtotal: $51.00
Tax: $6.63
Total: $57.63
```

Example 2: Unsuccessful Reservation

```
-------------------------------
Welcome to David's Car Rentals
-------------------------------
What type of car do you want to rent? Truck
How many days? (min 1): 2
Do you need a car seat? (y/n): n

A matching vehicle cannot be found
```