

## White Panda Backend Development Internship Task

This section tries to test your understanding of Node and NoSQL Database concepts. **You are strictly advised to use MongoDB for DB operations and Node for writing REST APIs.**

**You must also provide POSTMAN/Insomnia collection for testing your API. The API responses should be clear enough to depict the results of an action.**

**Your API must be hosted on a proper platform (Preferably on AWS - ec2 server with letsencrypt https configured on it OR Heroku)**

### Problem Statement:

Consider the following description\requirement given by a Car Rental agency-

“Our company owns several Cars in various cities available for rent on a daily basis. Every Car has a vehicle number, model, seating capacity, rent per day etc. People make reservations/booking for these cars. Booking status of these cars include the customer details (name, Ph No.), issue date (the date on which car is given on rent) and return date (the date on which customer returns the car) of the car.”

You have to make a basic system having the following features –

- Add new cars
- Book a specific car based on its availability. **A car can have multiple bookings**
- Show the cars that are available to book on a date, time, seating capacity or other filters
- Using Car Model (only) show the details of a particular car and its currently active booking
  - Update/Delete a Car from the system and ensure that the car is not already booked. **Explanation of this point:** Build endpoints to update and

delete a rental car object from the database while ensuring the condition that update or delete operations are not allowed in case the status of the car is 'booked'

You are expected to use the MVC architectural concepts -

- Define **Classes (Model)** for every Entity, mentioning the right data types, constructors, getters, setters, etc.
- Write the program logic in the **Controller** files using the above-defined Models.
- Use the right HTTP method for performing an action/operation
- Implementing an API Authentication **would be a bonus**

You need not work on the frontend.