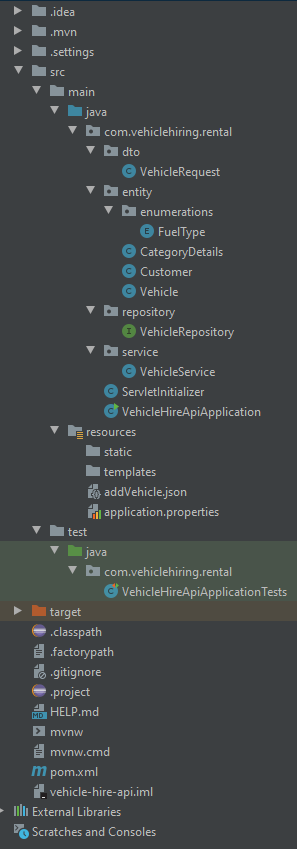
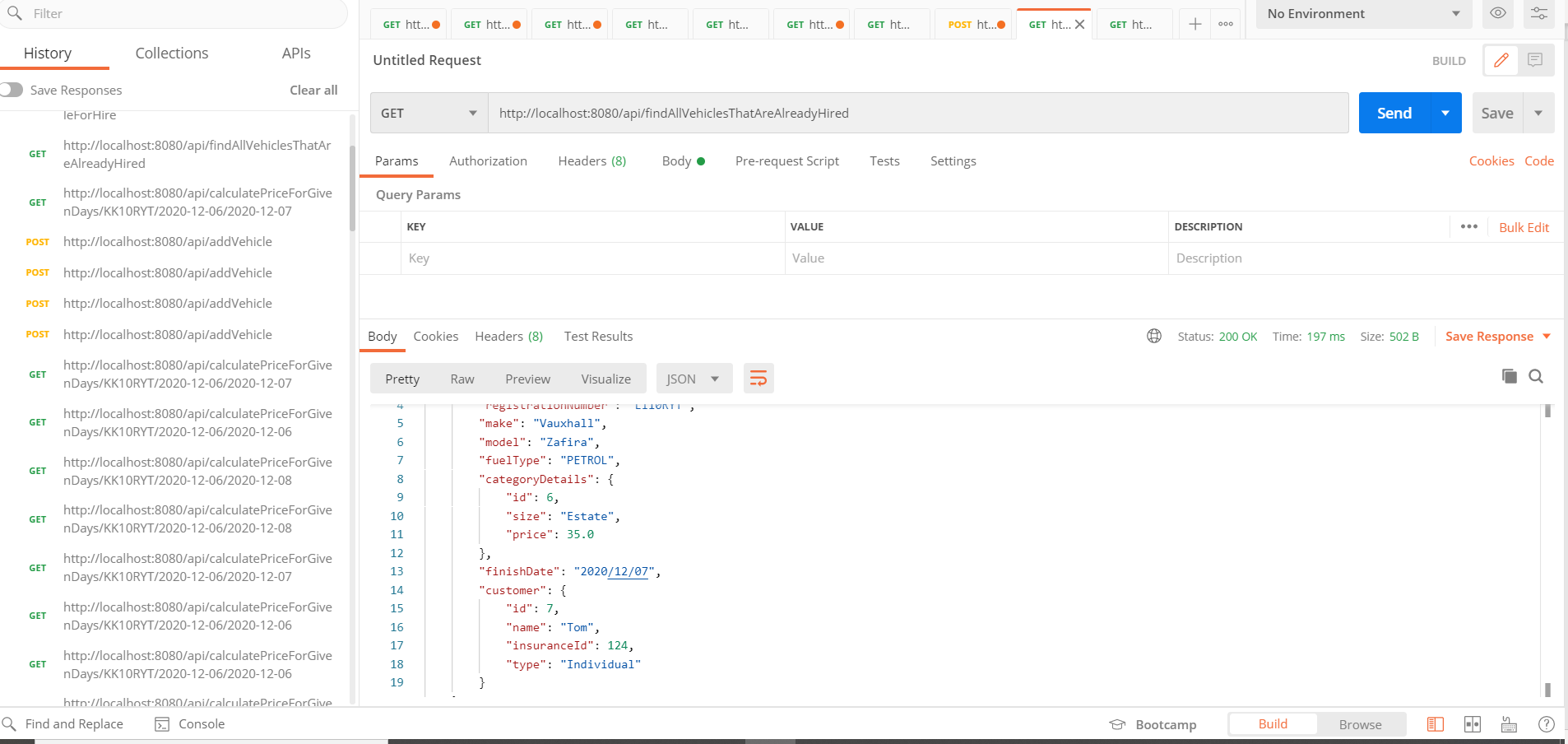
**Vehicle Hire Booking System**

1. **Created a spring boot application from https://start.spring.io/ with the required dependencies like web, sql server, jpa to begin with.**
2. **Coded entities for the application and the relevant associations and mappings between the entities.**
3. **Wrote the repository layer interface and the required methods for rest service.**
4. **Generated the rest endpoints to add vehicle, find all available vehicles, find all hired vehicles, calculation of total cost for the selected date ranges in the application.**
5. **Defined the right datasources in application.properties.**
6. **Ran as spring boot application.**
7. **Tested the endpoints using postman**
8. **I have placed the sample json in the resources section of the vehicle-hire-api project**
9. **Following is the screenshot of the project I have created.**
10. 
11. **Following is a snapshot of the endpoint trying to fetch the already hired vehicles.**
12. 
13. [**http://localhost:8080/api/findAllVehiclesThatAreAlreadyHired**](http://localhost:8080/api/findAllVehiclesThatAreAlreadyHired)
14. [**http://localhost:8080/api/calculatePriceForGivenDays/KK10RYT/2020-12-06/2020-12-07**](http://localhost:8080/api/calculatePriceForGivenDays/KK10RYT/2020-12-06/2020-12-07)
15. [**http://localhost:8080/api/addVehicle**](http://localhost:8080/api/addVehicle) **This is a post request with the following json raw body request**

{

  "vehicle": {

    "registrationNumber":"SH18RYT",

    "make":"Citroen",

    "model":"Q1",

    "fuelType":"PETROL",

    "categoryDetails":{

      "size":"Small",

      "price":25

    },

    "customer":{

      "name": "Bob",

      "insuranceId":411,

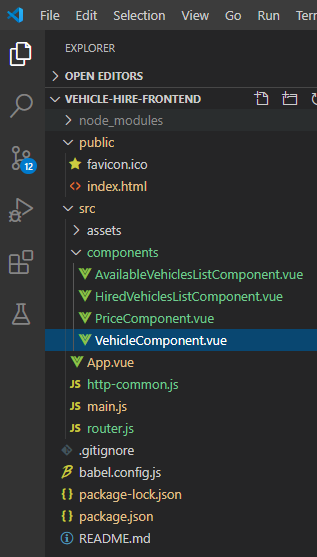
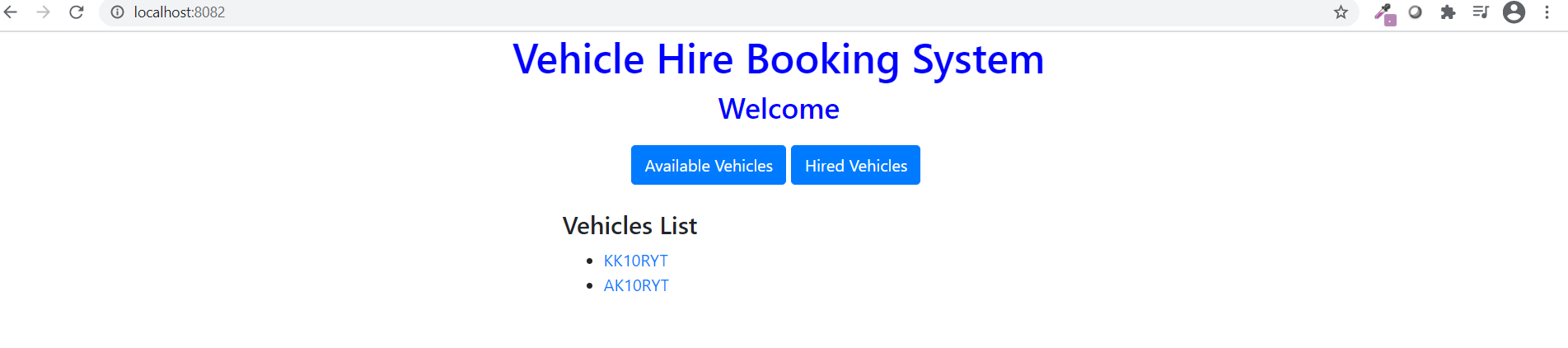
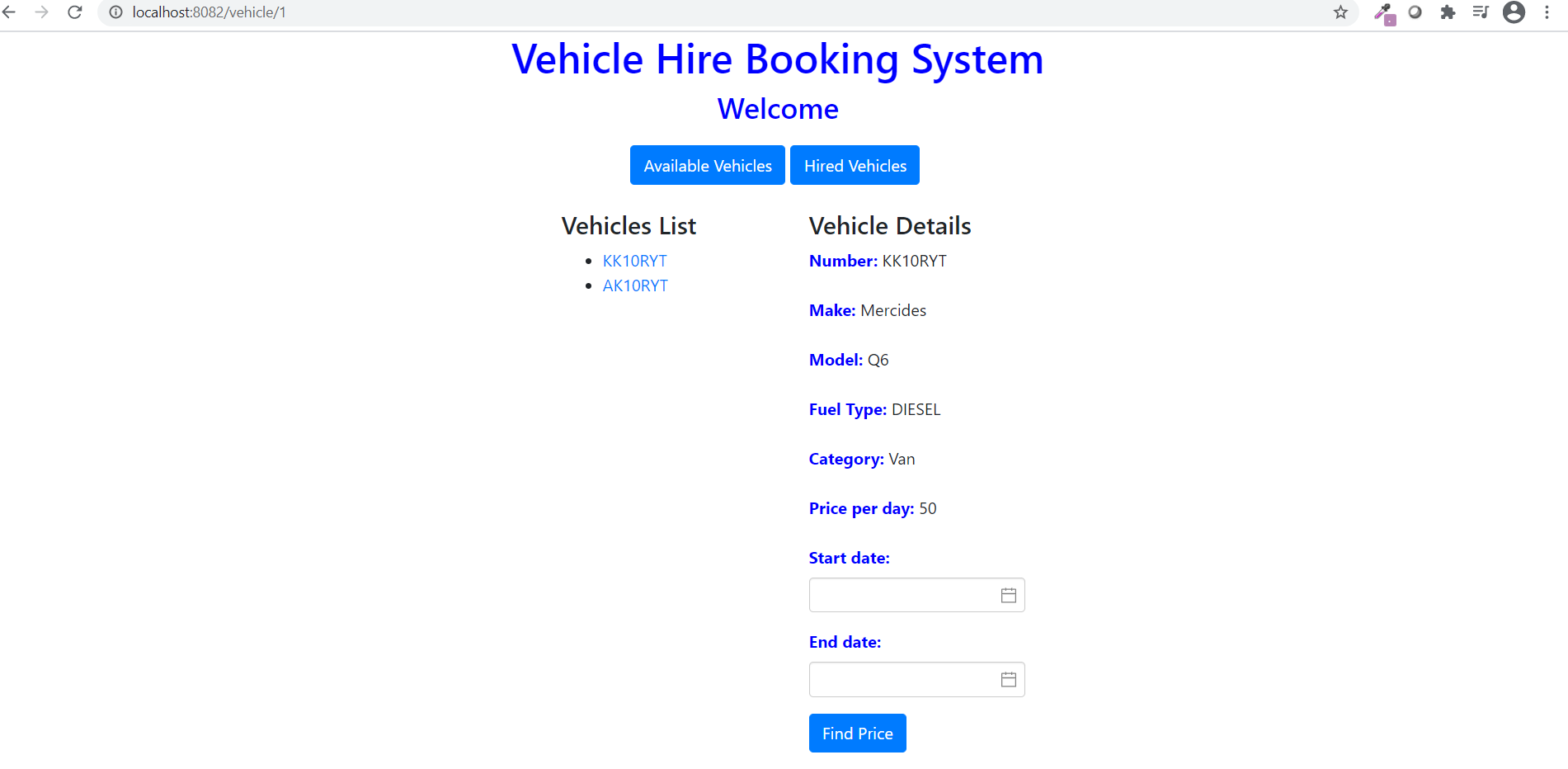
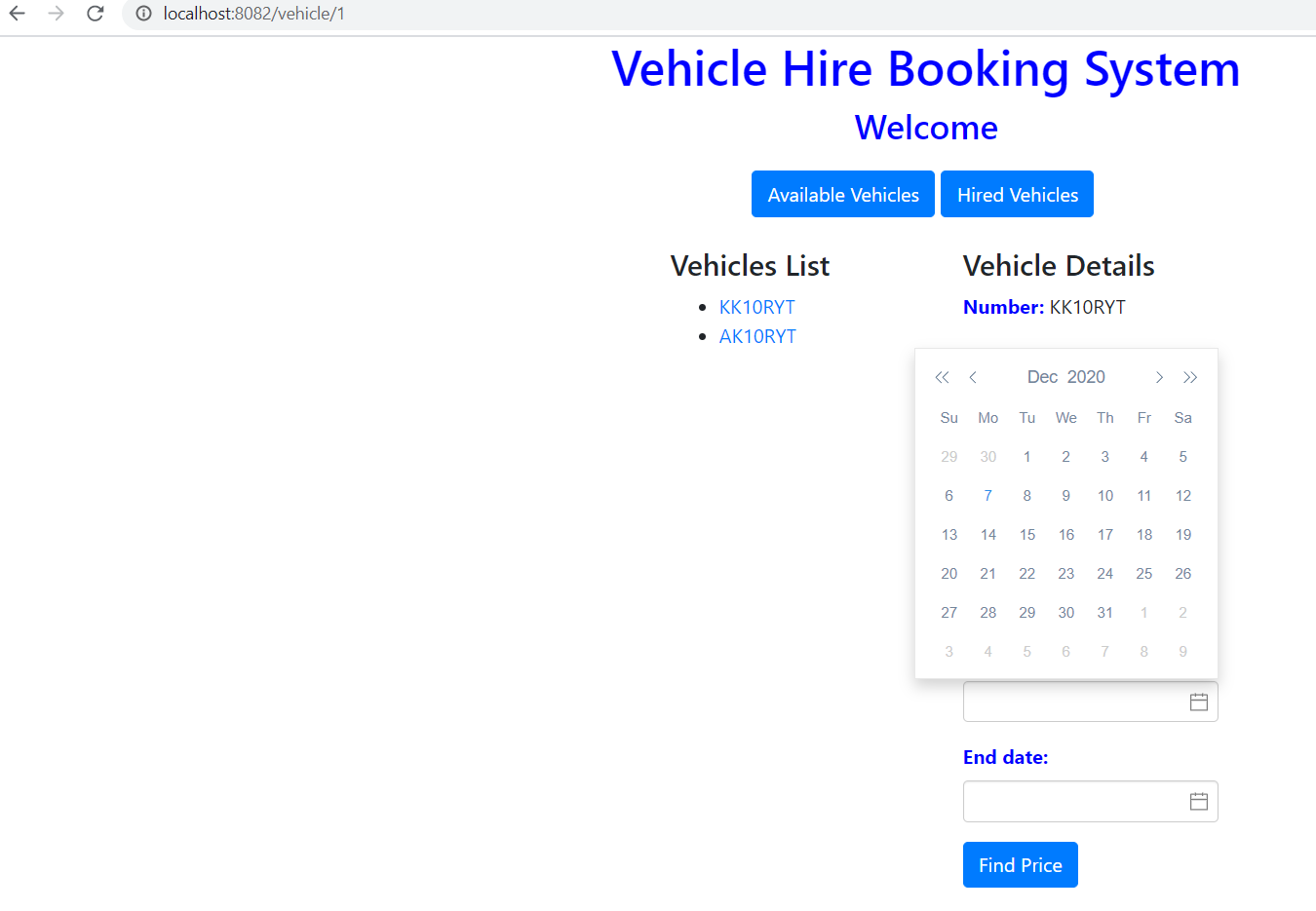
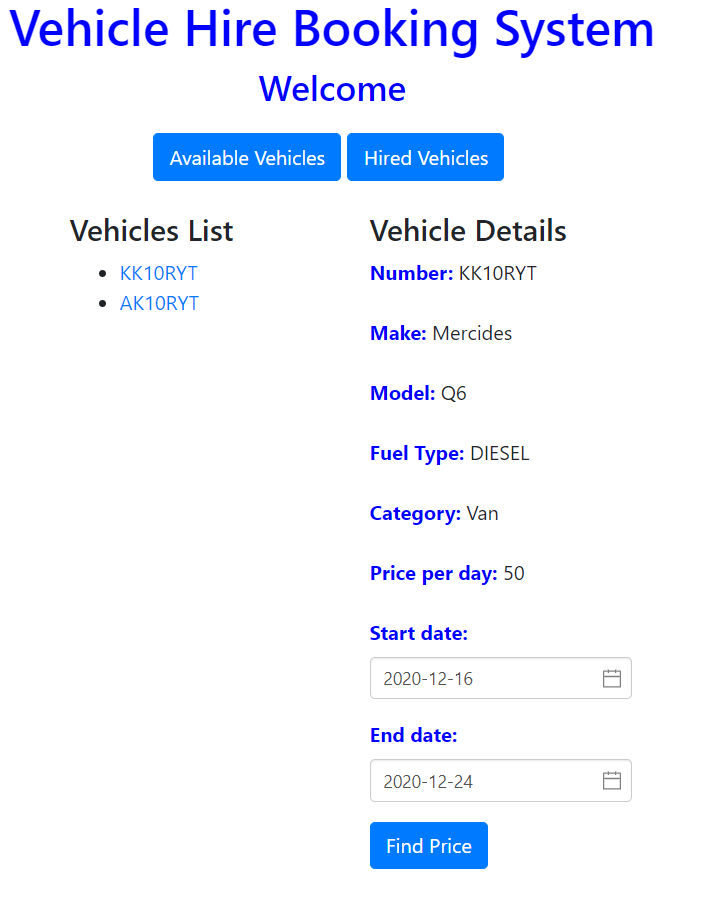
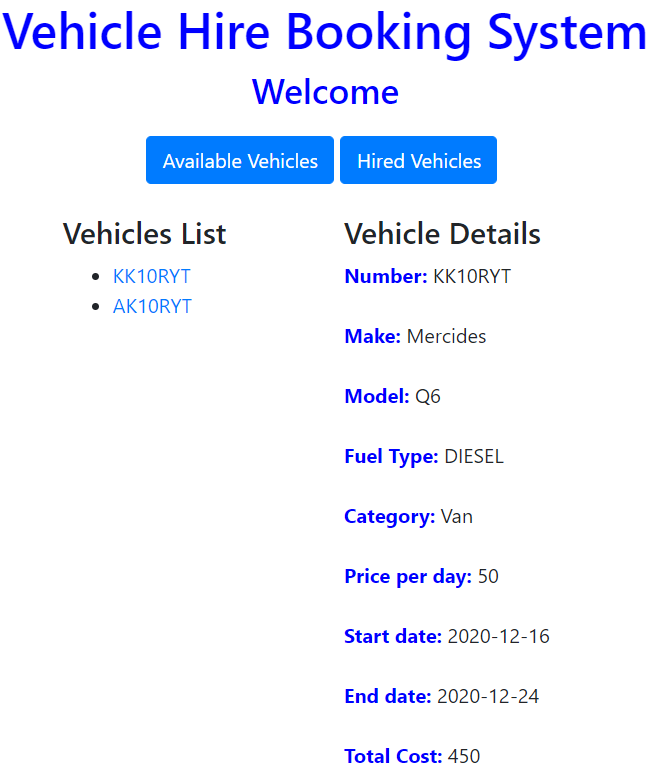
      "type":"Company"

    },

    "finishDate" : "2019/12/07"

  }

}

1. [**http://localhost:8080/api/findAllVehiclesAvailableForHire**](http://localhost:8080/api/findAllVehiclesAvailableForHire)
2. **For front end, vue js framework has been used**
3. **Node.js was already installed**
4. **I have launched up comman prompt and ran the following command.**
5. **npm install -g @vue/cli**
6. **I have then created my vue application using the following command.**
7. **Vue create vehicle-hire-frontend**
8. **Cd vehicle-hire-frontend**
9. **Npm run serve**
10. **I ran the vue application with port 8082 as that was the port assigned**
11. **I have mentioned this port 8082 in the crossorigins attribute in my spring boot application.**
12. **Npm add axios**
13. **Npm add vue-router (This one is for the vue application to communicate the rest end points.**
14. **Generated the required components as can be seen in the screenshot below:**
15. **Following is the structure of the vue project created**
16. 
17. **Following are the screenshots of the working vue application**
18. 
19. 
20. 
21. **I have used the vue2 datepicker component in the application**
22. 
23. 
24. **With the url** [**http://localhost:8082**](http://localhost:8082)**, displays the list of available vehicle for today, if there are any available.**
25. **Displays a list of cars with registration number.**
26. **On clicking on a registration number, displays the details of the vehicles in a different component.**
27. **When dates are selected with the datepicker component, then on clicking on the “Find price” button, displays the total cost for the number of days the booking was requested, for the specific car.**
28. **There is also an option to see the already hired vehicles. On clicking the “Hired Vehicles” button, we can see the list of already hired vehicles as seen below.**
29. 
30. **This is the demo of the application I have created using Spring Boot, JPA with backend as sqlserver database and front end as vue.js framework. The runtime environment used was Java 11.**
31. **If more time was given, I would have worked on creating Spring Auto Rest documentation for the application by creating integration tests and including the necessary dependencies in the pom.xml file.**