**Testing**

Run both backend and frontend for testing by entering their respective directories and running:

$ yarn run dev

In the root directory of the repository there is a file called DansCarBarnpostman\_collection.json that can be imported into postman. There are various endpoints that are useful for testing the backend and can be used to test the data fetching and response capabilities of the backend. We have used these endpoints periodically to test the functionality of the Car Barn and have found them extremely useful. They are pasted below for usage on a case by case basis.

***Tests for backend to be run through postman:***

{

"info": {

"\_postman\_id": "2211635a-4362-42e9-bbe9-9743d7666f07",

"name": "DansCarBarn",

"schema": "https://schema.getpostman.com/json/collection/v2.1.0/collection.json",

"\_exporter\_id": "25728539"

},

"item": [

{

"name": "Reservations",

"item": [

{

"name": "Add Reservation",

"request": {

"method": "POST",

"header": [],

"body": {

"mode": "raw",

"raw": "{\n \"userId\": 1,\n \"vehicleId\": 1,\n \"start\": \"02/21/2023\",\n \"end\": \"02/22/2023\",\n \"status\": \"IN\_TRANSIT\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "http://localhost:3001/reservation",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"reservation"

]

}

},

"response": []

},

{

"name": "Get Reservations",

"request": {

"method": "GET",

"header": [],

"url": {

"raw": "http://localhost:3001/reservation",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"reservation"

]

}

},

"response": []

},

{

"name": "Get Reservation",

"request": {

"method": "GET",

"header": [],

"url": {

"raw": "http://localhost:3001/reservation/1",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"reservation",

"1"

]

}

},

"response": []

},

{

"name": "Update Status",

"request": {

"method": "PUT",

"header": [],

"body": {

"mode": "raw",

"raw": "{\n \"status\": \"RETURNED\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "http://localhost:3001/reservation/status/1",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"reservation",

"status",

"1"

]

}

},

"response": []

}

]

},

{

"name": "Vehicles",

"item": [

{

"name": "Create Vehicle",

"request": {

"method": "POST",

"header": [],

"body": {

"mode": "raw",

"raw": "{\n \"make\": \"GMC\",\n \"model\": \"Canyon\",\n \"year\":2007,\n \"price\_class\":\"GOLD\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "http://localhost:3001/vehicle",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"vehicle"

]

}

},

"response": []

},

{

"name": "Get All Vehicles",

"request": {

"method": "GET",

"header": [],

"url": {

"raw": "http://localhost:3001/vehicle/all",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"vehicle",

"all"

]

}

},

"response": []

},

{

"name": "Browse Vehicles",

"request": {

"method": "GET",

"header": [],

"url": {

"raw": "http://localhost:3001/vehicle/browse?start=06/06/2023&end=07/07/2023",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"vehicle",

"browse"

],

"query": [

{

"key": "start",

"value": "06/06/2023"

},

{

"key": "end",

"value": "07/07/2023"

}

]

}

},

"response": []

},

{

"name": "Update Vehicle",

"request": {

"method": "PUT",

"header": [],

"body": {

"mode": "raw",

"raw": "{\n \"id\": 1,\n \"year\": 2006\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "http://localhost:3001/vehicle",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"vehicle"

]

}

},

"response": []

}

]

},

{

"name": "Users",

"item": [

{

"name": "Register User",

"request": {

"method": "POST",

"header": [],

"body": {

"mode": "raw",

"raw": "{\n \"email\": \"madramsdell@gmail.com\",\n \"password\": \"gamer1000\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "http://localhost:3001/user",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"user"

]

}

},

"response": []

},

{

"name": "Login User",

"request": {

"method": "POST",

"header": [],

"body": {

"mode": "raw",

"raw": "{\n \"email\": \"madramsdell@gmail.com\",\n \"password\": \"gamer1000\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "http://localhost:3001/user/login",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"user",

"login"

]

}

},

"response": []

},

{

"name": "Get User Info",

"request": {

"method": "GET",

"header": [],

"url": {

"raw": "http://localhost:3001/user",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"user"

]

}

},

"response": []

},

{

"name": "Update Permissions",

"request": {

"method": "PUT",

"header": [],

"body": {

"mode": "raw",

"raw": "{\n \"id\": 1,\n \"role\": \"MANAGER\"\n}",

"options": {

"raw": {

"language": "json"

}

}

},

"url": {

"raw": "http://localhost:3001/user/permissions",

"protocol": "http",

"host": [

"localhost"

],

"port": "3001",

"path": [

"user",

"permissions"

]

}

},

"response": []

}

]

}

],

"auth": {

"type": "bearer",

"bearer": [

{

"key": "token",

"value": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwiaWF0IjoxNjc2NTcxNjY4fQ.ik5fo\_fqpDBbaOZpD5oDQDWBtk2Vm3NF7xDmU1RKr7k",

"type": "string"

}

]

},

"event": [

{

"listen": "prerequest",

"script": {

"type": "text/javascript",

"exec": [

""

]

}

},

{

"listen": "test",

"script": {

"type": "text/javascript",

"exec": [

""

]

}

}

]

}

Frontend Tests:

1. **User rents a vehicle:**
   1. navigate to the login page, enter a username and password, and click ‘Login’ to redirect to the browse page
   2. open the account tab and add sufficient funds to rent a vehicle
   3. put in the date range for when the vehicle is needed, and several available ones come up on the screen
   4. select one, view its details, go back, and select another.
   5. click the reserve button found on the page. You are then brought to a checkout page with all of the fees associated with reserving the rental.
   6. click the checkout button, and the rental is marked as reserved in this user’s name.
   7. set the date to the appointed day, log in as the manager, and mark the reservation as active.
   8. set the date to the end of the period, and mark the reservation as fulfilled.
2. **Customer renting a vehicle but doesn’t have enough money in account:**
   1. The customer logs into the application by inputting their username and password. They are then redirected to the browse rentals page. They click on a vehicle which pops up a modal with information on the vehicle and buttons to rent the vehicle. The user clicks the “rent vehicle” button and is redirected to a checkout page where they can select to add insurance, view their total, and a button that says checkout. Upon clicking checkout the user is blocked and notified that they have insufficient funds. There is a button to redirect the user back to the home page. The user clicks it and is redirected to the home page. They then navigate to their user profile where they select to manage account balance. This takes them to a page where they can choose to add or remove money from their account. They select to add money and input the amount they want to put in. Upon completion they navigate back to the page where they select the vehicle they want again and are directed down the same pages. From there when they select checkout they are greeted with a success page and details about their reservation. Details for their reservation are added and can be viewed by them from their “My reservations” page. The user proceeds to logout as they are satisfied with their experience.
3. **Manager Adds Car:** 
   1. The business has a new vehicle that they want to display on Dan’s car barn available for rental. They open up the admin panel and select the new vehicle button. A page opens up in which the manager then inputs the information and attributes associated with the new vehicle, and submits the vehicle. The vehicle now displays on the users’ end as available for rental.