FIT 2095
Assignment 2 Document

Student Name: Aiken Lee
Student ID: 35423056
Semester: 2024 sem 2

Routing Table

Name	Endpoint	HTTP Method	Purpose	Handler Method
Home	/	GET	Home page	getHome
Add Driver	/35423056/Aiken/add-driver	GET	Renders the form to add a new driver	showAddDriverForm
New Driver	/35423056/Aiken/new-driver	POST	Processes the form data to create a new driver	createNewDriver
Delete Driver Page	/35423056/Aiken/delete-driver	GET	Renders the form to delete a driver	showDeleteDriverForm
Delete Driver	/35423056/Aiken/delete-driver	POST	Deletes a driver from the database	deleteDriver
List Drivers	/35423056/Aiken/list-driver	GET	Lists all drivers	listAllDrivers
List Drivers by Department	/35423056/Aiken/list-driver-department	GET	Renders the form to list drivers by department	showListDriversByDepartmentForm
View Drivers by Department	/35423056/Aiken/list-driver-department	POST	Lists drivers filtered by department	listDriversByDepartment
Add Package	/35423056/Aiken/add-package	GET	Renders the form to add a package	showAddPackageForm
New Package	/35423056/Aiken/new-package	POST	Processes the form data to create a new package	createNewPackage
Delete Package Page	/35423056/Aiken/delete-package	GET	Renders the form to delete a package	showDeletePackageForm
Delete Package	/35423056/Aiken/delete-package	POST	Deletes a package from the database	deletePackage
List Packages	/35423056/Aike	GET	Lists all	listAllPackages

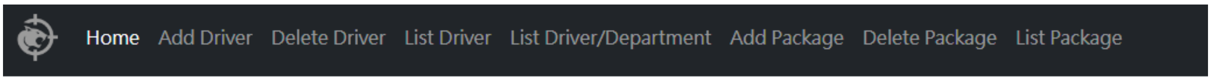
	n/list-package		packages	
404 Page	*	GET	Catches undefined routes and renders a 404 page	handle404

Routing Table for API request

Name	Endpoint	HTTP Method	Purpose	Handler Method
Insert New Driver	/35423056/Aiken/api/v1/drivers/add	POST	Inserts a new driver into the database	insertNewDriver
List All Drivers	/35423056/Aiken/api/v1/drivers	GET	Retrieves and lists all drivers	listAllDriver
Delete Driver	/35423056/Aiken/api/v1/drivers/delete	DELETE	Deletes a driver and their associated packages	deleteDriver
Update Driver	/35423056/Aiken/api/v1/drivers/update	PUT	Updates a driver's information in the database	updateDriver
Insert New Package	/35423056/Aiken/api/v1/packages/add	POST	Inserts a new package into the database	insertNewPackage
List All Packages	/35423056/Aiken/api/v1/packages	GET	Retrieves and lists all packages	listAllPackage
Delete Package	/35423056/Aiken/api/v1/packages/delete	DELETE	Deletes a package and removes it from its driver	deletePackage
Update Package	/35423056/Aiken/api/v1/packages/update	PUT	Updates a package's destination in the database	updatePackage

Instruction

The website is very simple to work with. On every page, there will be a navbar above. You can access different pages when you click on the above.



Home page:

Home page, you will see this page when you click on the back to home button in the list driver/list package page.

Add driver page:

This is the page for adding a driver to the database. There will be a form for you to enter the information. After you press the submit button. The backend will check if the data you entered fits the requirement, and will redirect to an invalid data page when the data does not fit the requirement. If the data is correct, the page will redirect you to a page that lists all the drivers.

Delete driver page:

A page to delete a driver. You have to enter a driver ID to do so. It will delete a driver if the driver ID is in the database, then it will redirect you to the list driver page. It will not delete any driver if you enter the wrong driver ID, it will instead redirect you to the invalid data page.

List driver page:

A page to list all drivers. There will be a red button on the right for each driver, you can delete a driver by pressing that button.

List driver/department page:

A page to list all drivers by department. It functions the same as the list driver page, notice that it will still show the driver from that department only even after you delete a driver through the red button.

Add package page:

A page to add packages to the database. It functions the same as an add driver. Also, notice that you need at least 1 driver in the database to use the drop-down menu for selecting a driver related to the package.

Delete package page:

A page for deleting a package. This works the same as the delete driver page.

List package page:

A page for listing all packages. This works the same as the list driver page.

404 page:

This is a page you will see when you enter the wrong url.

Invalid data page:

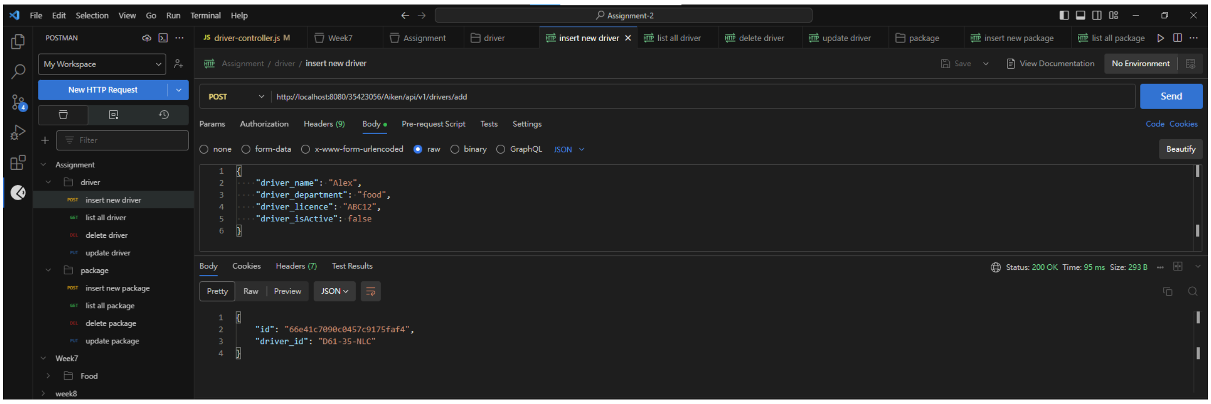
This is an invalid data page you will see when you enter the wrong data. The problem it shows will change based on what you did wrong.

Instruction for API

The API section is also very easy to work with. I will briefly talk about what it does.
It is worth to notice that every single API is wrap with a try-catch block to ensure that a fail API request will not just shut the website down.

Driver Insert:

The JSON format is the exact same as the requirement. Below is an example that I've done to showcase how to use this.



POST http://localhost:8080/35423056/Aiken/api/v1/drivers/add

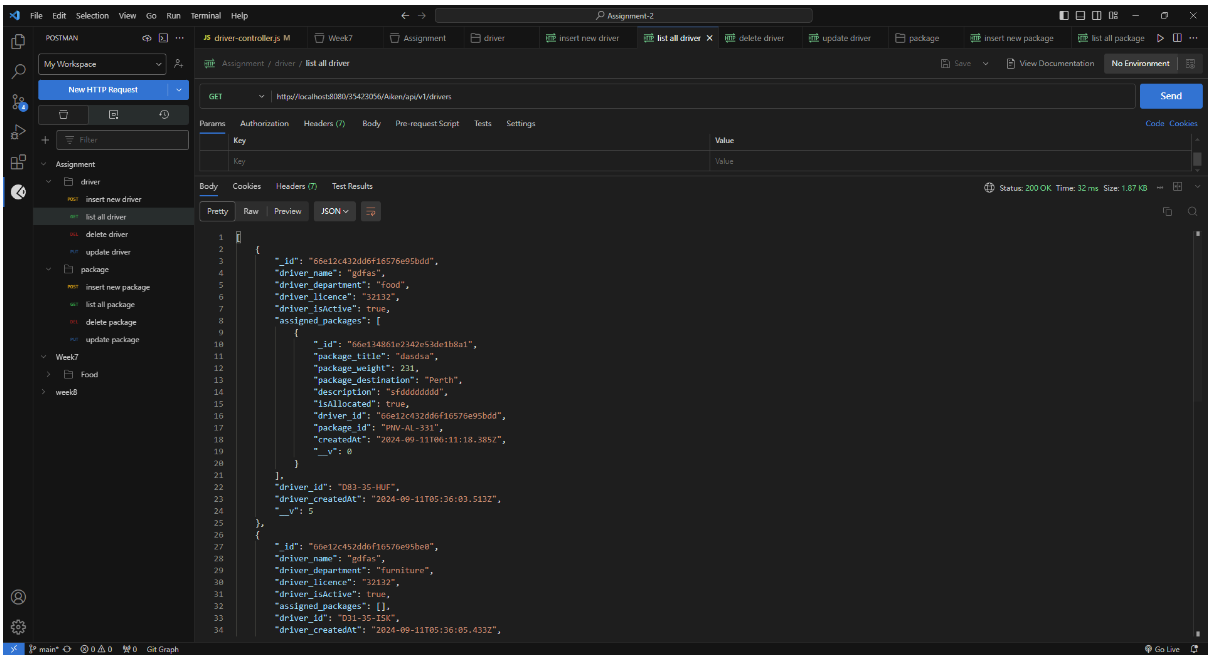
```
{
    "driver_name": "Alex",
    "driver_department": "food",
    "driver_licence": "ABC12",
    "driver_isActive": false
}
```

Status: 200 OK Time: 95 ms Size: 293 B

```
{
    "id": "66e41c7090c0457c9175faf4",
    "driver_id": "D61-35-NLC"
}
```

Driver List:

The JSON format will reply all the drivers and all of their packages.



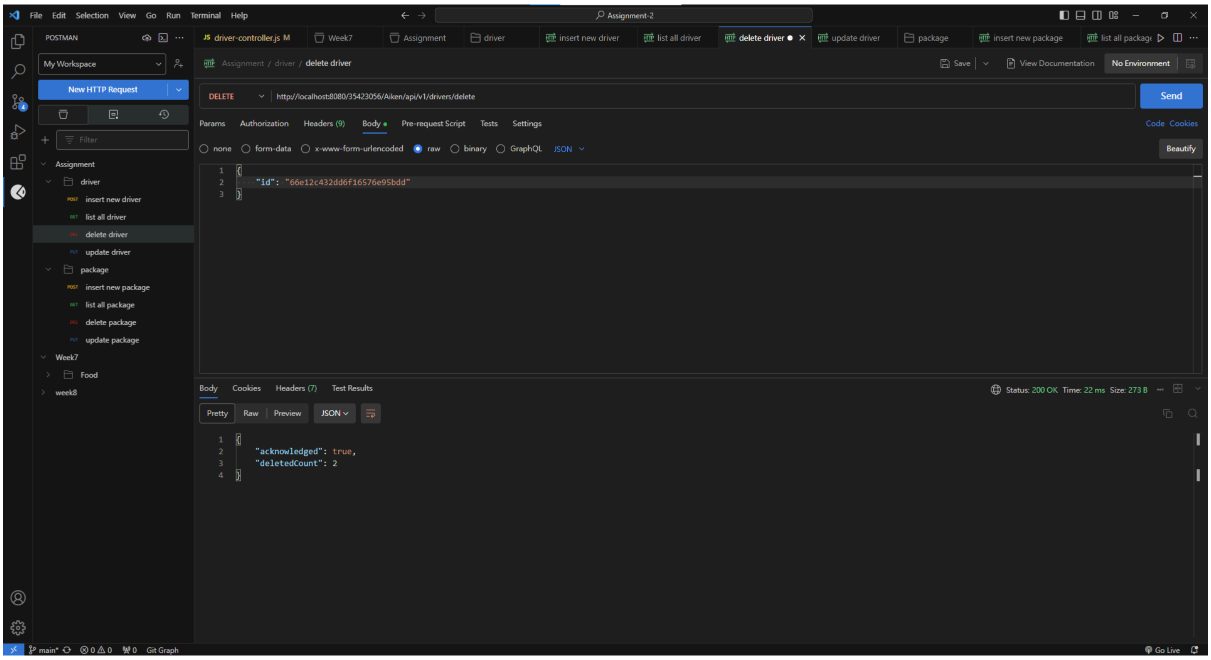
GET http://localhost:8080/35423056/Aiken/api/v1/drivers

Status: 200 OK Time: 32 ms Size: 1.87 KB

```
[
    {
        "_id": "66e12c432dd6f16576e95bdd",
        "driver_name": "gdfas",
        "driver_department": "food",
        "driver_licence": "32132",
        "driver_isActive": true,
        "assigned_packages": [
            {
                "_id": "66e134861e2342e53de1b8a1",
                "package_title": "dasdsa",
                "package_weight": 231,
                "package_destination": "Perth",
                "description": "sfddddddddd",
                "isAllocated": true,
                "driver_id": "66e12c432dd6f16576e95bdd",
                "package_id": "PNV-AL-331",
                "createdAt": "2024-09-11T06:11:18.385Z",
                "__v": 0
            }
        ],
        "driver_id": "D83-35-HUF",
        "driver_createdAt": "2024-09-11T05:36:03.513Z",
        "__v": 5
    },
    {
        "_id": "66e12c452dd6f16576e95be0",
        "driver_name": "gdfas",
        "driver_department": "furniture",
        "driver_licence": "32132",
        "driver_isActive": true,
        "assigned_packages": [],
        "driver_id": "D31-35-ISK",
        "driver_createdAt": "2024-09-11T05:36:05.433Z",
```

Driver Delete:

This API deletes a driver and all of its packages. It will reply true if driver is deleted and the number of item deleted. For example, for this one, I've deleted 1 driver and 1 package. So that will give the deleted count of 2.



DELETE http://localhost:8080/35423056/Aiken/api/v1/drivers/delete

Body: raw JSON

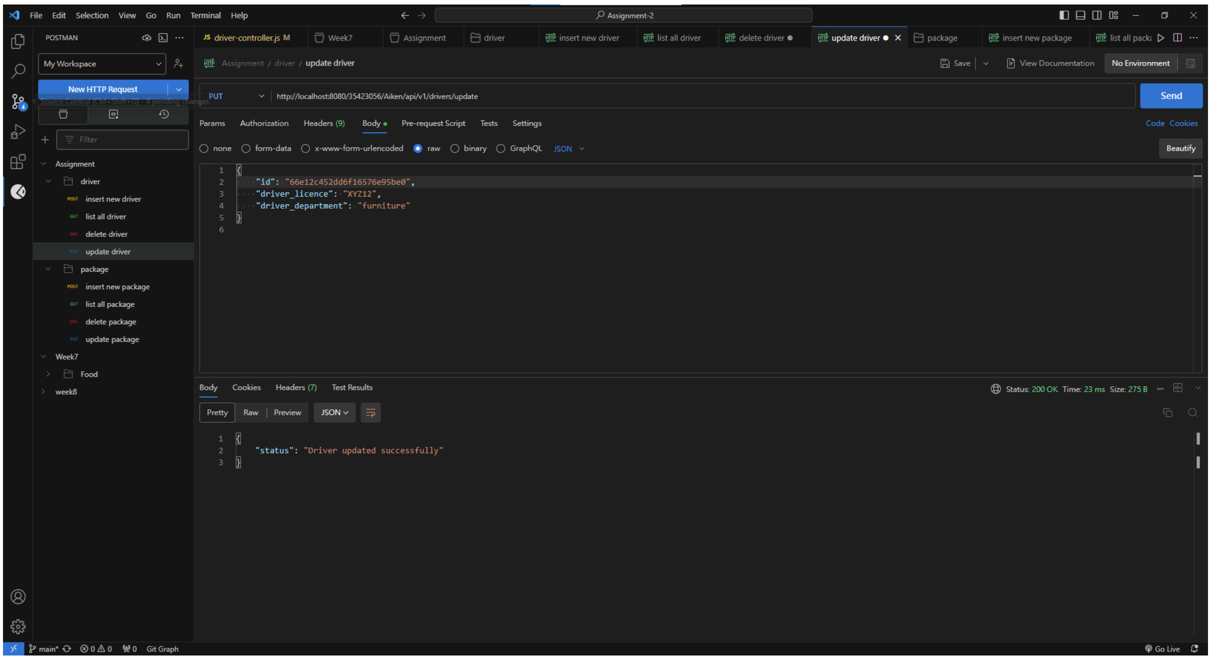
```
{
    "id": "66e12c432dd6f16576e95bdd"
}
```

Status: 200 OK Time: 22 ms Size: 273 B

```
{
    "acknowledged": true,
    "deletedCount": 2
}
```

Driver Update:

This will simply update a driver's licence and department.



PUT http://localhost:8080/35423056/Aiken/api/v1/drivers/update

Body: raw JSON

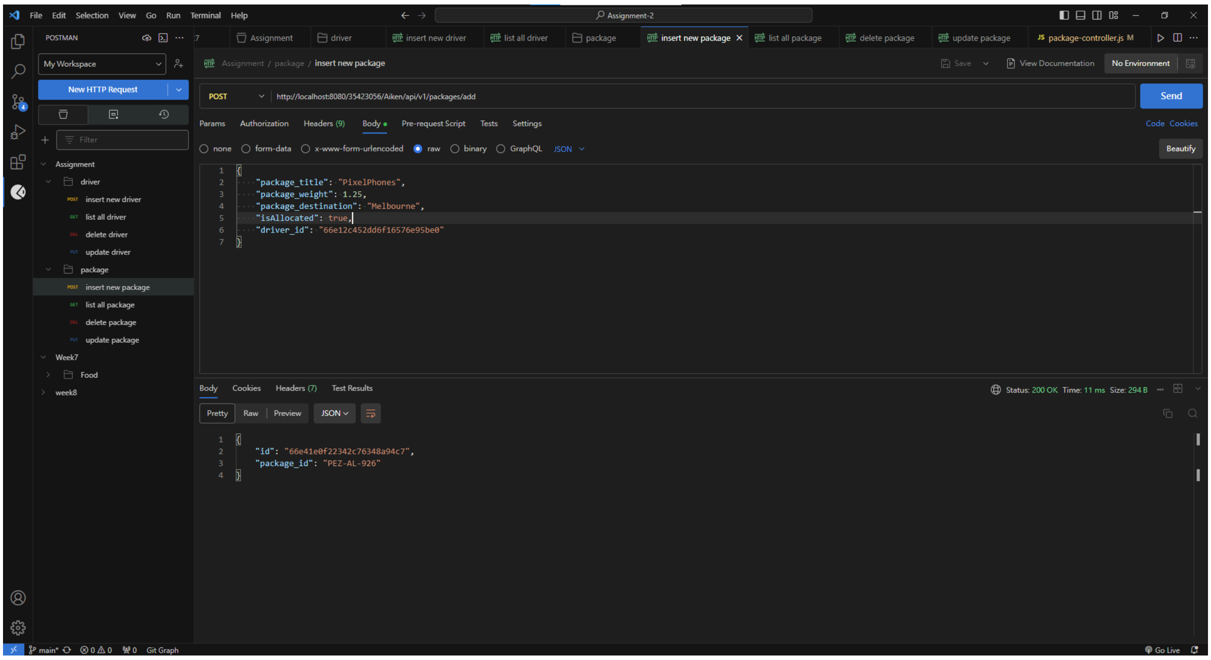
```
{
    "id": "66e12c452dd6f16576e95be0",
    "driver_licence": "XYZ12",
    "driver_department": "furniture"
}
```

Status: 200 OK Time: 23 ms Size: 275 B

```
{
    "status": "Driver updated successfully"
}
```

Package Insert:

This is different from the requirement. I've made it more flexible by allowing descriptions to be not in there. You can call the API with or without a description. Below are 2 examples. This is easily done by changing the required in the package class.



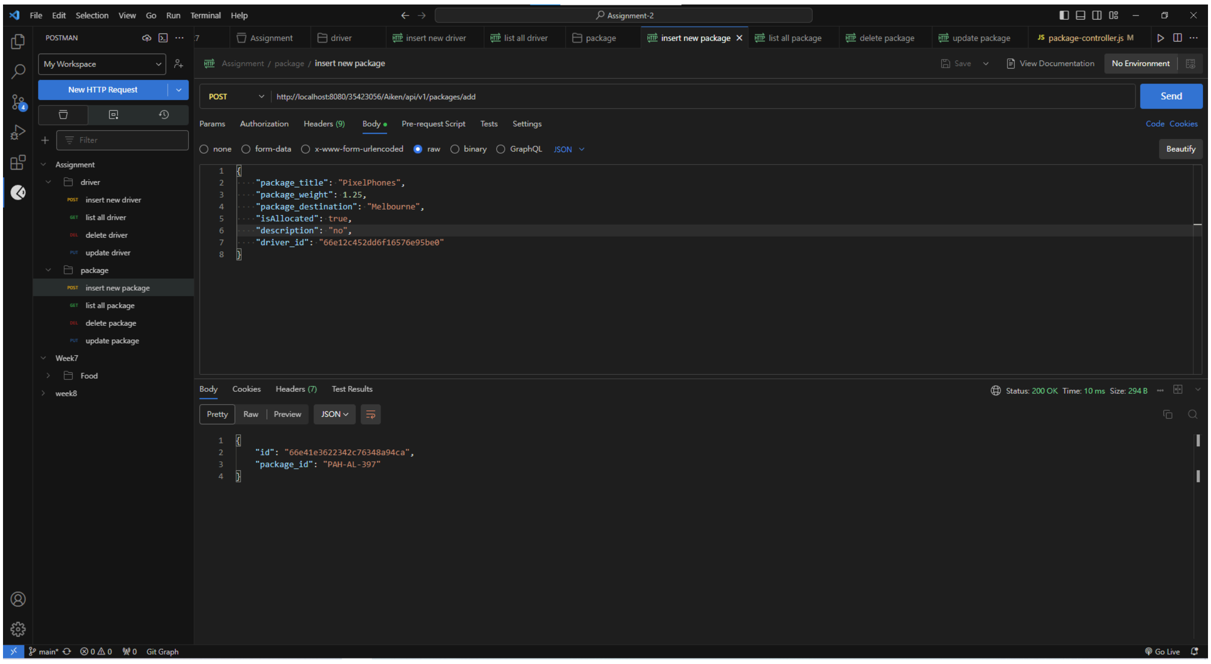
POST http://localhost:8080/35423056/Aiken/api/v1/packages/add

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "package_title": "PixelPhones",
    "package_weight": 1.25,
    "package_destination": "Melbourne",
    "isAllocated": true,
    "driver_id": "66e12c452dd6f16576e95be0"
}
```

Body Cookies Headers (7) Test Results — Status: 200 OK Time: 11 ms Size: 294 B

```
{
    "id": "66e41e0f22342c76348a94c7",
    "package_id": "PEZ-AL-926"
}
```



POST http://localhost:8080/35423056/Aiken/api/v1/packages/add

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

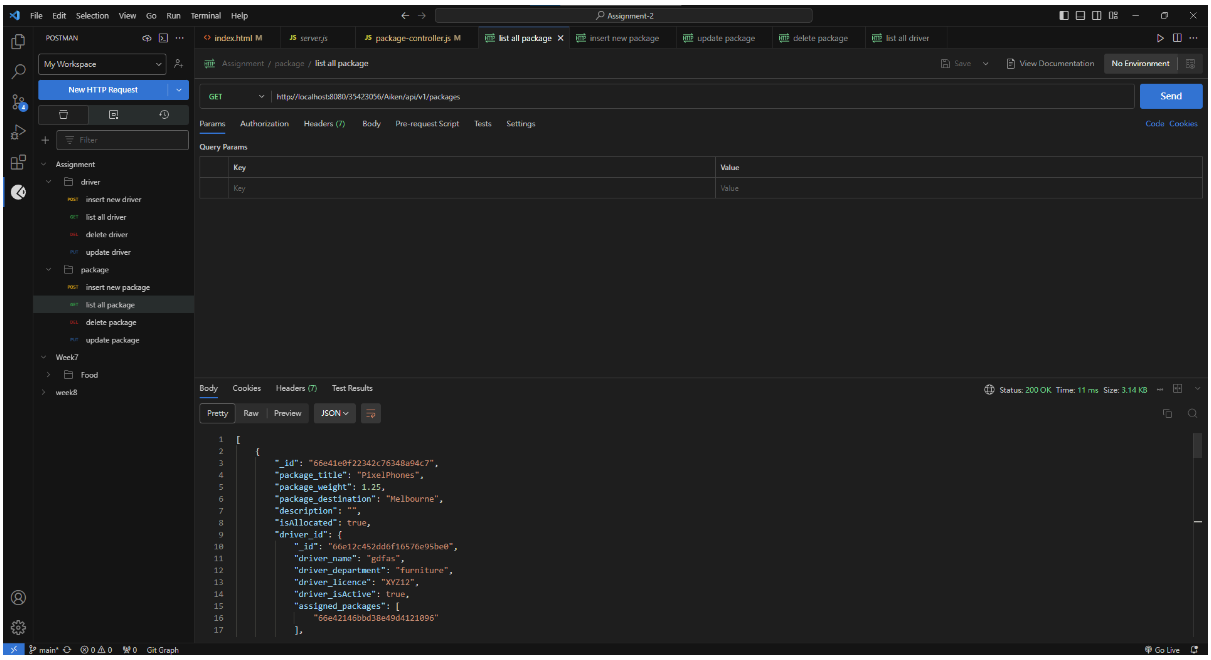
```
{
    "package_title": "PixelPhones",
    "package_weight": 1.25,
    "package_destination": "Melbourne",
    "isAllocated": true,
    "description": "no",
    "driver_id": "66e12c452dd6f16576e95be0"
}
```

Body Cookies Headers (7) Test Results — Status: 200 OK Time: 10 ms Size: 294 B

```
{
    "id": "66e41e3622342c76348a94ca",
    "package_id": "PAH-AL-397"
}
```

Package List:

List all packages, and list all of the drivers related to all of the packages.



Assignment / package / list all package

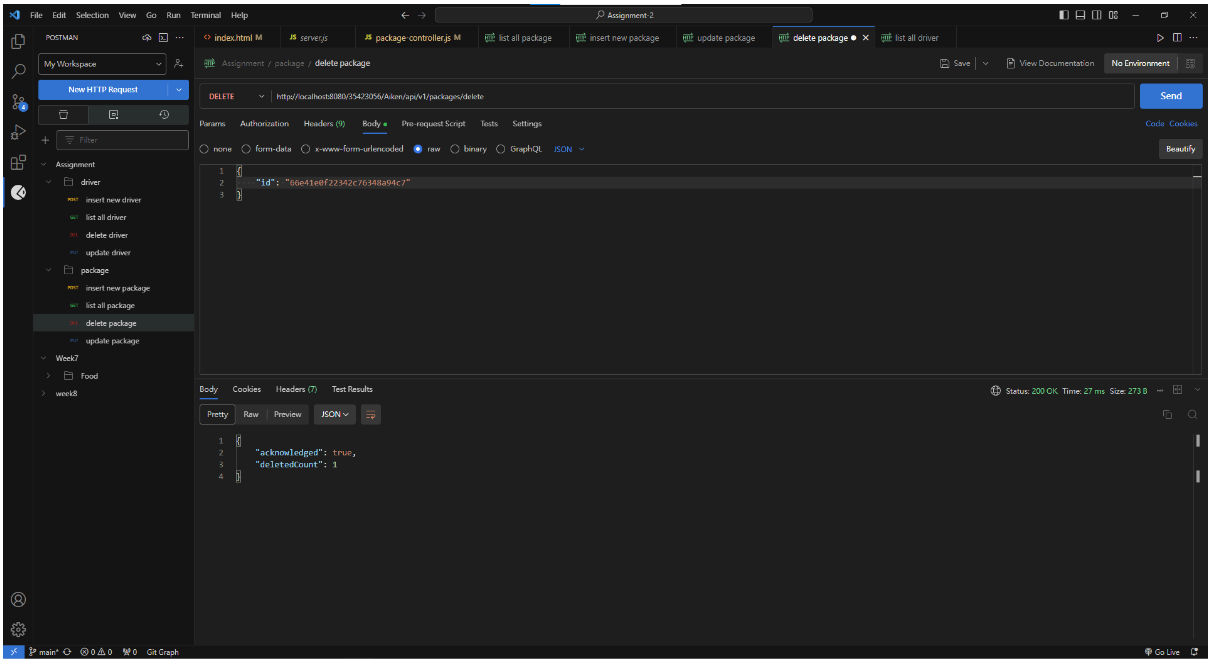
GET http://localhost:8080/35423056/Aiken/api/v1/packages

Status: 200 OK Time: 11 ms Size: 3.14 KB

```
[
    {
        "_id": "66e41e0f22342c76348a94c7",
        "package_title": "PixelPhones",
        "package_weight": 1.25,
        "package_destination": "Melbourne",
        "description": "",
        "isAllocated": true,
        "driver_id": {
            "_id": "66e12c452dd6f16576e95be0",
            "driver_name": "gdfas",
            "driver_department": "furniture",
            "driver_licence": "XYZ12",
            "driver_isActive": true,
            "assigned_packages": [
                "66e42146bbd38e49d4121096"
            ],
```

Package Delete:

Delete a package. Similar to driver delete, it will also reply to the deleted count. But in this case, the max is 2 because a package can not have more than 1 driver.



Assignment / package / delete package

DELETE http://localhost:8080/35423056/Aiken/api/v1/packages/delete

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

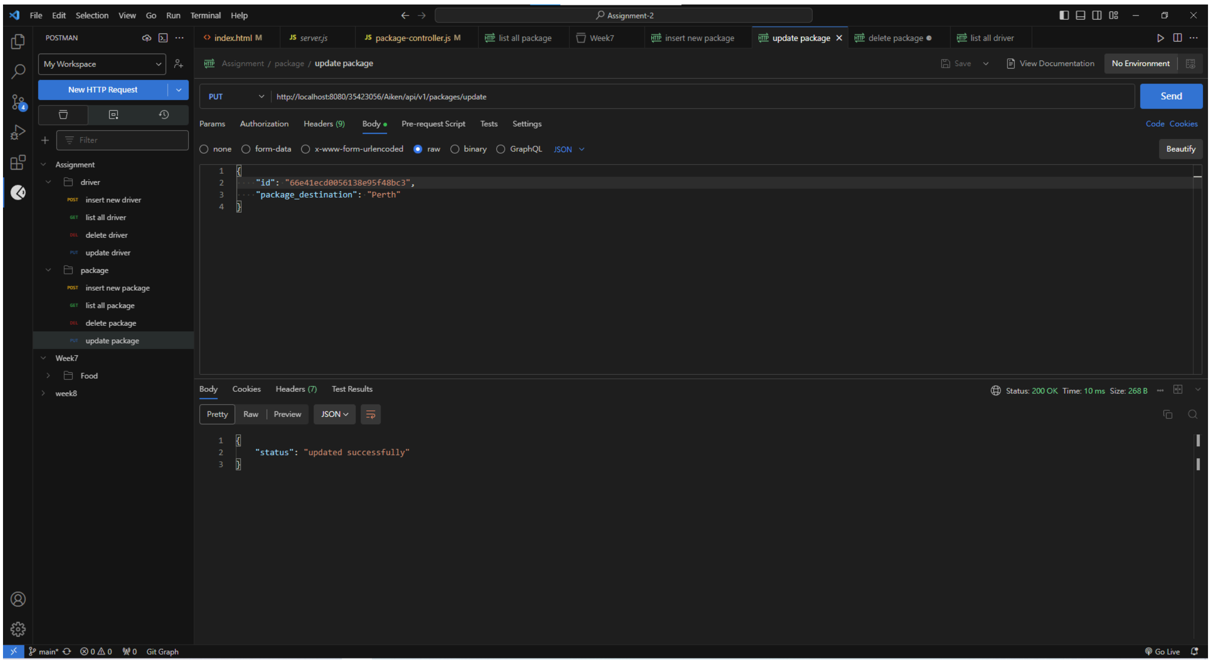
```
{
    "id": "66e41e0f22342c76348a94c7"
}
```

Status: 200 OK Time: 27 ms Size: 273 B

```
{
    "acknowledged": true,
    "deletedCount": 1
}
```

Package Update:

change the package destination.



PUT http://localhost:8080/35423056/Aiken/api/v1/packages/update

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "id": "66e41ecd0056138e95f48bc3",
    "package_destination": "Perth"
}
```

Status: 200 OK Time: 10 ms Size: 268 B

Body: JSON

```
{
    "status": "updated successfully"
}
```

VM deployment

Current IP: http://34.129.71.104:8080/
VM is deployed. However, I can not guarantee that the IP will be the same by the time the assignment is graded. Therefore I am putting a instruction on how to restart the instance.

1. cd Assignment-2
2. node server.js

Key Point to Notice

There are a few key points to notice when navigating the website.

1. In order to achieve a navbar that is usable in all aspect ratios, I've chosen to use the default Bootstrap JS and dependencies for interactive nav bar. The navbar will collapse into a burger menu when you are on an aspect ratio that makes it unable to showcase everything.
2. The footer for every page is achieved by referencing the same footer.ejs file in views. It is good for making it not redundant(because the footer stays the same on all pages.).
3. The port number I've chosen is 8080. However, you can always modify it by changing the PORT_NUMBER constant.
4. "node server.js" is the command to setup the server.

Key Point to Notice API

1. Some of the pushes I've made are in assignment 1 git. I've accidentally pushed to assignment 1.