

Implementation Assignment

Assignment 3

Class

Client-Server Web Development

Tutor

Doug MacKenzie

Date Last Revised

Sunday, 18 June 2023

Table of Contents

1. Setup.....	3
A. MongoDB Database.....	3
a. User Credentials.....	3
b. Database Details.....	3
c. Application Details.....	3
B. Global Modules and Visual Studio Extensions	3
a. Global Modules	3
b. Visual Studio Extensions.....	3
c. Extension to Recover Missing Local Packages.....	3
C. Commands to Start Project	4
a. Client.....	4
b. Server.....	4
2. API Testing	5
A. Headings.....	5
a. Fetch by Page	5
B. Locations	5
a. Fetch All	5
b. Fetch All Store Names.....	6
c. Fetch By Store Name.....	6
d. Fetch Head Office.....	7
C. Vehicle.....	7
a. Fetch By ID.....	7
b. Fetch By Category	8
c. Not Implemented in Website Due to Client Promise and Time Constraint.....	9
D. Book.....	12
a. Get All Vehicle Makes.....	12
b. Get All Distinct Vehicle Models by a Specific Make.....	12
c. Get All Colours for a Specific Vehicle.....	13
3. Database Testing	14
A. Headings.....	14
a. Find One By Page.....	14
B. Locations	14
a. Find All	14
b. Find All Store Names.....	14
c. Find Head Office.....	15
C. Vehicles.....	16
a. Find By ID.....	16
b. Find By Category	16
c. Update By ID.....	17
d. Delete By ID.....	17
D. Book.....	17
a. Get All Vehicle Makes.....	17
b. Get All Distinct Vehide Models by a Specific Make.....	17
c. Get All Colours for a Specific Vehicle.....	17
References	18

1. Setup

A. MongoDB Database

a. User Credentials

i. Username

Admin

ii. Password

Toiohomai1234

iii. Authentication Database

admin

iv. Role

root

b. Database Details

i. Database Name

phoenix

ii. Collections

headings

locations

vehicles

c. Application Details

i. Database URI

mongodb://admin:Toiohomai1234@127.0.0.1:27017:27017/phoenix?authMechanism=DEFAULT&authSource=admin

ii. Location

Project > Server > .env > DB_URI

B. Global Modules and Visual Studio Extensions

a. Global Modules

@vue/cli

b. Visual Studio Extensions

Vue 3 Snippets

Vetur

c. Extension to Recover Missing Local Packages

npm install missing

C. Commands to Start Project

a. Client

npm run serve

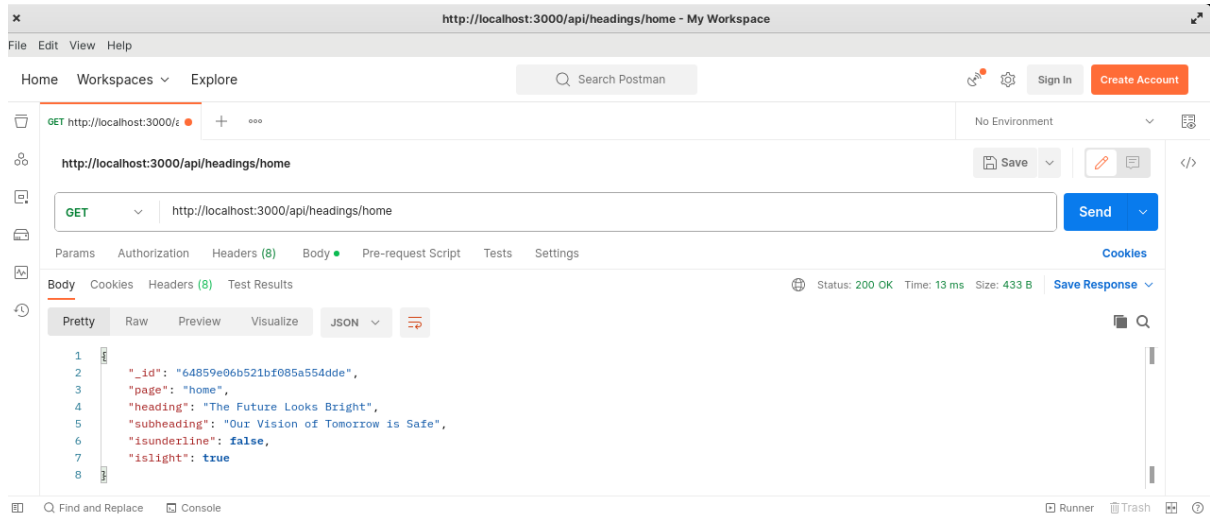
b. Server

npm start

2. API Testing

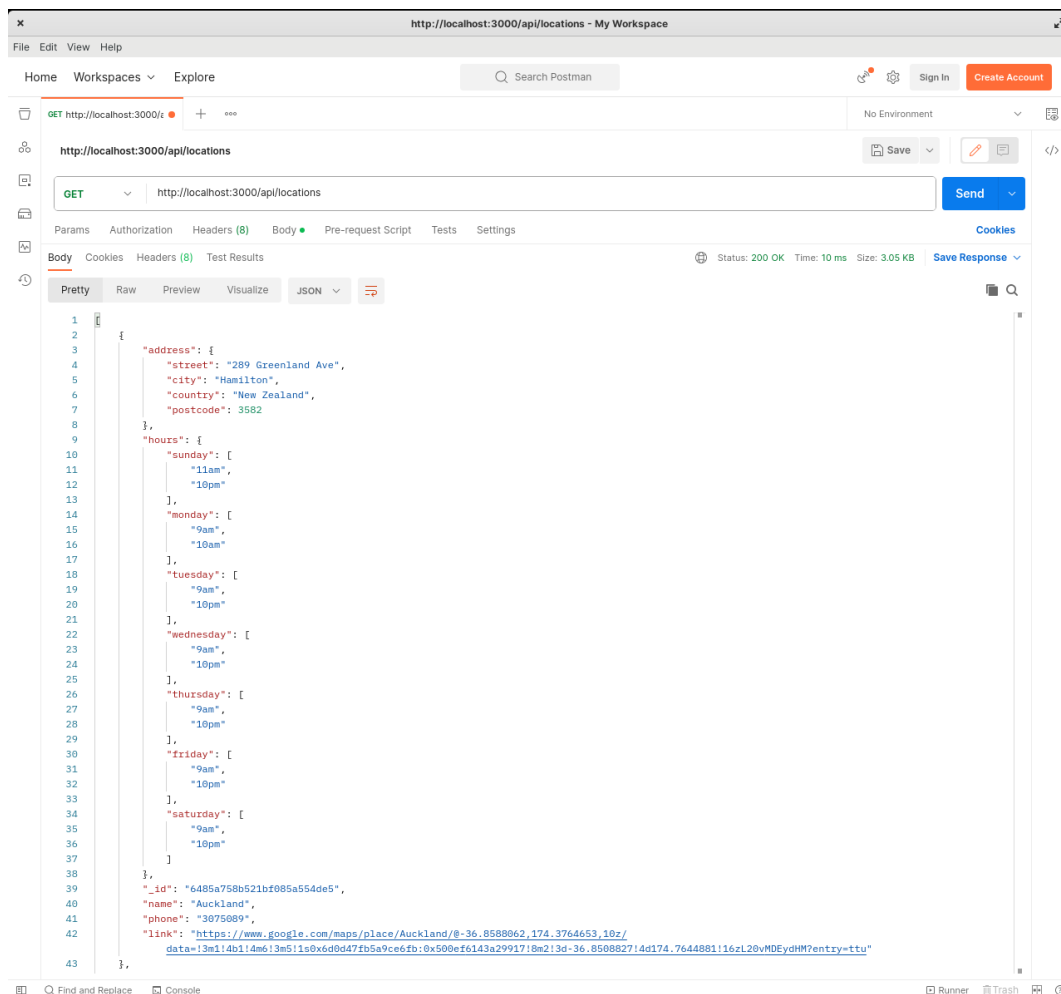
A. Headings

a. Fetch by Page



B. Locations

a. Fetch All



b. Fetch All Store Names

Postman interface showing a GET request to `http://localhost:3000/api/locations/names/any`. The response is a JSON array of 5 store objects, each with an `_id` and a `name`.

```
1 {
2   {
3     "_id": "6485a758b521bf085a554de5",
4     "name": "Auckland"
5   },
6   {
7     "_id": "6485a60eb521bf485a554de4",
8     "name": "Head Office"
9   },
10  {
11    "_id": "648dbcaf01b4bd8da1b10a5f",
12    "name": "Tauranga"
13  },
14  {
15    "_id": "648dbcf901b4bd8da1b10a61",
16    "name": "Wellington"
17  },
18  {
19    "_id": "648dbd8a01b4bd8da1b10a62",
20    "name": "New Plymouth"
21  }
22 }
```

Status: 200 OK Time: 15 ms Size: 544 B

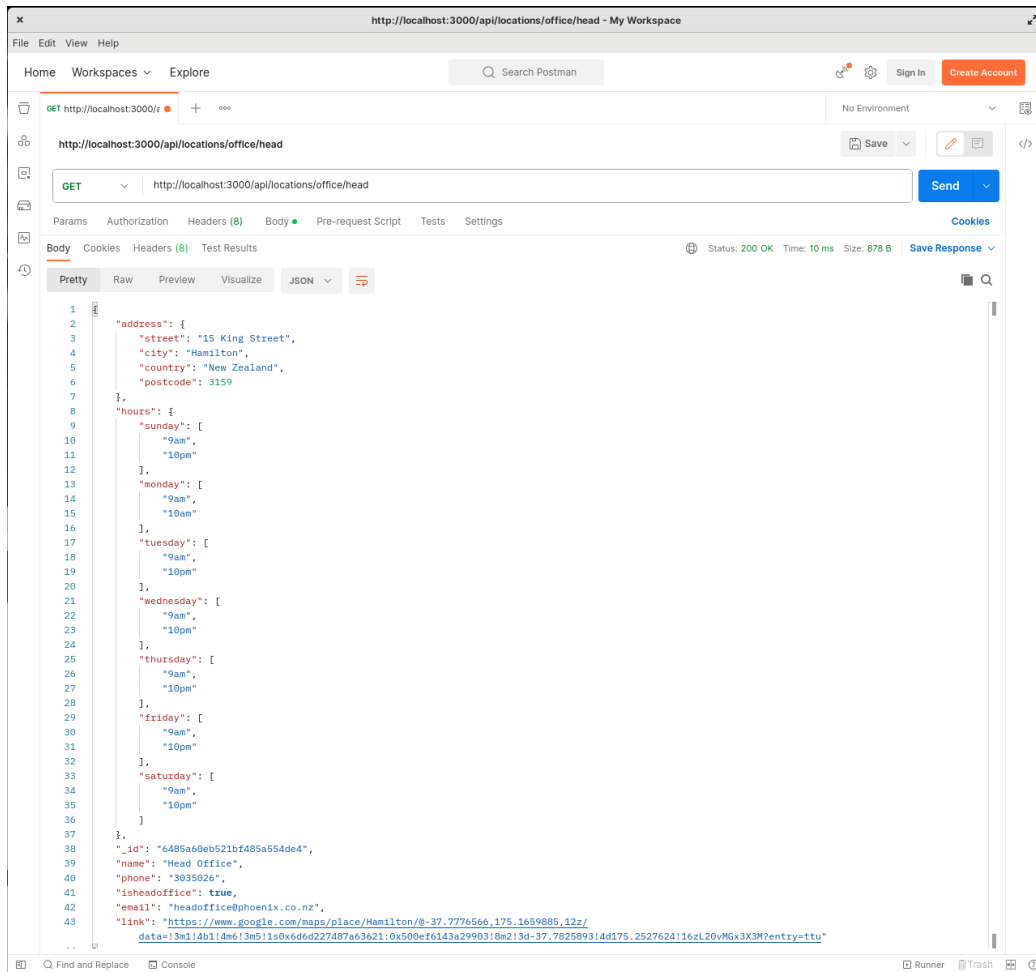
c. Fetch By Store Name

Postman interface showing a GET request to `http://localhost:3000/api/locations/Tauranga`. The response is a JSON object containing detailed information for the Tauranga store.

```
1 {
2   "address": {
3     "street": "50 Windermere Dr",
4     "city": "Tauranga",
5     "country": "New Zealand",
6     "postcode": "3159"
7   },
8   "hours": {
9     "sunday": [
10      "1pm",
11      "5pm"
12    ],
13     "monday": [
14      "9am",
15      "12pm"
16    ],
17     "tuesday": [
18      "9.30am",
19      "12pm"
20    ],
21     "wednesday": [
22      "9.30am",
23      "12pm"
24    ],
25     "thursday": [
26      "9.30am",
27      "12pm"
28    ],
29     "friday": [
30      "9.30am",
31      "12pm"
32    ],
33     "saturday": [
34      "1pm",
35      "7pm"
36    ]
37  },
38   "_id": "648dbcaf01b4bd8da1b10a5f",
39   "name": "Tauranga",
40   "phone": "3258276",
41   "link": "https://www.google.com/maps/place/Tauranga/8-37.7536341,175.9866782,11z/data=!3m1!4m1!4m3!5!1s0x6d6e76e77bd464e5:0x508ef6143a39926!8m2!3d-37.6869653!4d176.1654272!16zL28vYDFuNmX6?entry=ttu"
42 }
```

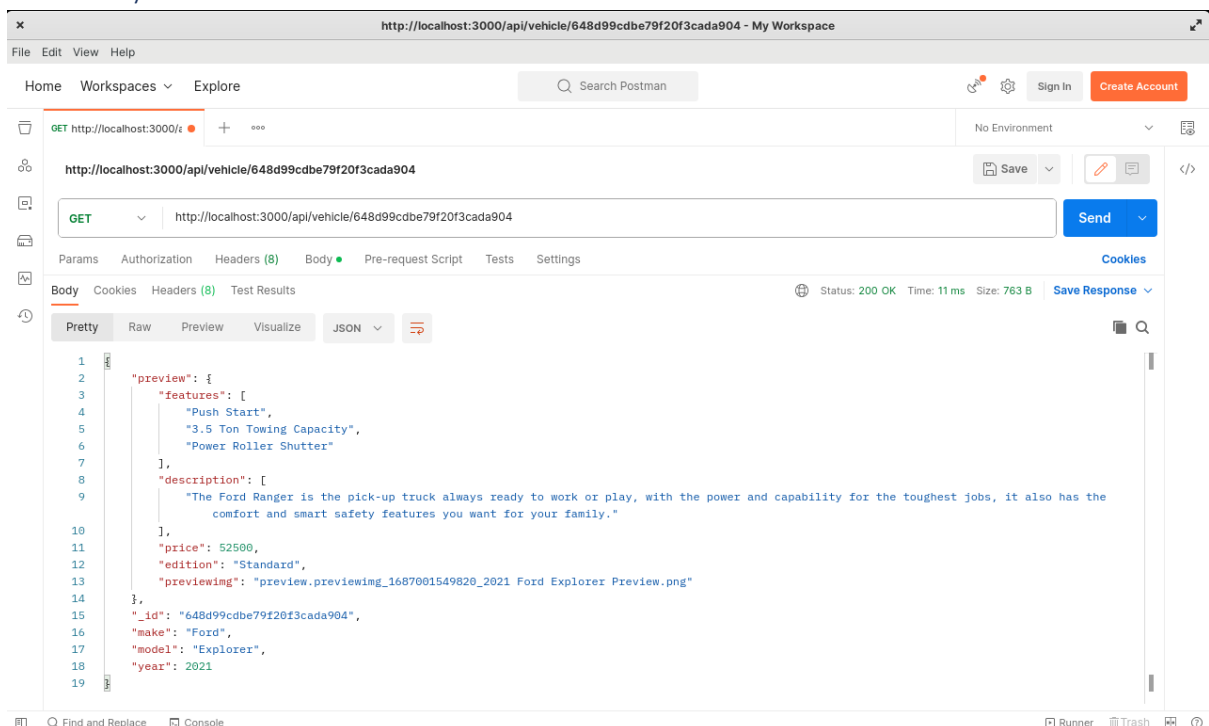
Status: 200 OK Time: 7 ms Size: 833 B

d. Fetch Head Office

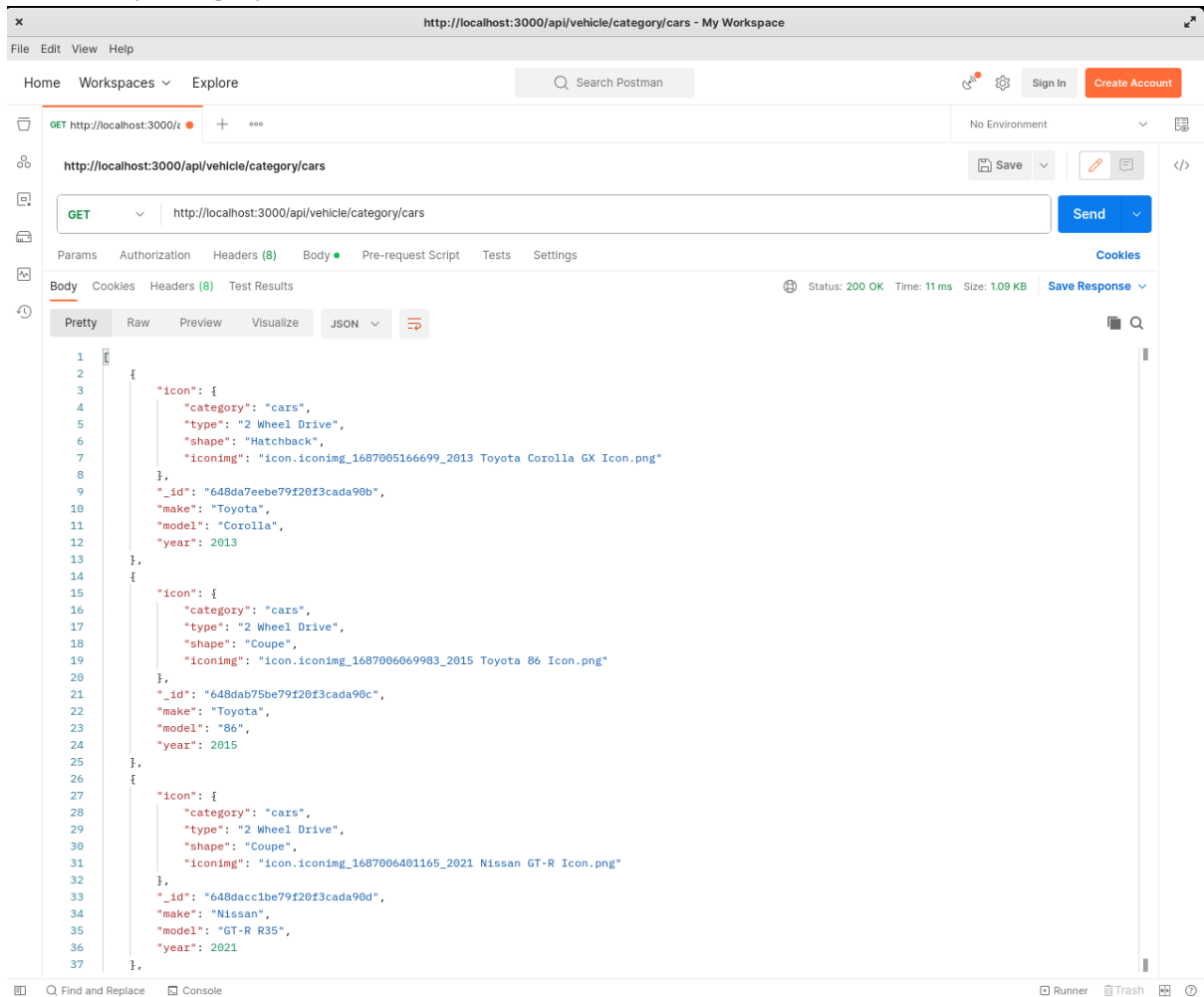


C. Vehicle

a. Fetch By ID



b. Fetch By Category



The screenshot shows the Postman interface with a GET request to `http://localhost:3000/api/vehicle/category/cars` successfully executed. The response status is 200 OK, with a time of 11 ms and a size of 1.09 KB. The response body is displayed in the 'Body' tab, showing a JSON array of three car objects. Each object contains an 'icon' field with a URL to a specific car icon.

```
1 {
2   {
3     "icon": {
4       "category": "cars",
5       "type": "2 Wheel Drive",
6       "shape": "Hatchback",
7       "iconimg": "icon.iconimg_1687005166699_2013 Toyota Corolla GX Icon.png"
8     },
9     "_id": "648da7eebe79f20f3cada90b",
10    "make": "Toyota",
11    "model": "Corolla",
12    "year": 2013
13  },
14  {
15    "icon": {
16      "category": "cars",
17      "type": "2 Wheel Drive",
18      "shape": "Coupe",
19      "iconimg": "icon.iconimg_1687006069983_2015 Toyota 86 Icon.png"
20    },
21    "_id": "648dab75be79f20f3cada90c",
22    "make": "Toyota",
23    "model": "86",
24    "year": 2015
25  },
26  {
27    "icon": {
28      "category": "cars",
29      "type": "2 Wheel Drive",
30      "shape": "Coupe",
31      "iconimg": "icon.iconimg_1687006401165_2021 Nissan GT-R Icon.png"
32    },
33    "_id": "648dacc1be79f20f3cada90d",
34    "make": "Nissan",
35    "model": "GT-R R35",
36    "year": 2021
37  },
38 }
```


c. Not Implemented in Website Due to Client Promise and Time Constraint

i. Create

The screenshot displays two applications used for testing and database management. The top application is Postman, showing a POST request to `http://localhost:3000/api/vehicle/create` with a body containing vehicle details. The bottom application is MongoDB Compass, showing the `vehicles` collection in the `phoenix.vehicles` database with 16 documents and 1 index.

Postman Request Details:

- Method: POST
- URL: `http://localhost:3000/api/vehicle/create`
- Body (form-data):

Key	Value	Description
make	Toyota	
model	Maxima	
year	2007	
- Status: 201 Created, Time: 36 ms, Size: 299 B
- Response (JSON):

```
{  "message": "Vehicle Added"}
```

MongoDB Compass Details:

- Database: `phoenix.vehicles`
- Collection: `vehicles`
- Documents: 16, Indexes: 1
- Filter: `{ field: 'value' }`
- Export Data: 1 - 17 of 17
- Document Structure (JSON):

```
{  "_id": ObjectId("648ec86132607fa54f186207"),  "icon": Object,  "preview": Object,  "colours": Array,  "make": "Toyota",  "model": "Maxima",  "year": 2007,  "__v": 0}
```

Website Header:

PHOENIX Home Store Contact

Vehicle Categories: CARS, SUVs, VANS, UTES

Vehicle Listings:

- TOYOTA COROLLA 2013
2 WHEEL DRIVE | HATCHBACK
- TOYOTA 86 2015
2 WHEEL DRIVE | COUPE
- NISSAN GT-R R35 2021
2 WHEEL DRIVE | COUPE
- FORD MUSTANG 2023
2 WHEEL DRIVE | COUPE
- TOYOTA MAXIMA 2007
2 WHEEL DRIVE | DOUBLE CAB PICKUP

ii. Update

http://localhost:3000/api/vehicle/update/648ec86132607fa54f186207 - My Workspace

File Edit View Help

Home Workspaces Explore Search Postman Sign In Create Account

PATCH http://localhost:3000/api/vehicle/update/648ec86132607fa54f186207

Save Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/> icon.category	utes		
Key	Value	Description	

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 11 ms Size: 296 B Save Response

Pretty Raw Preview Visualize JSON

```
1
2
{"message": "Vehicle Updated"}
```

MongoDB Compass - Assignment 3 - Offline DB/phoenix.vehicles

Connect Edit View Collection Help

Assignment 3 - ... Documents phoenix.vehicles

My Queries Databases Search

admin config local phoenix

headings locations vehicles

16 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options


ADD DATA EXPORT DATA

1 - 17 of 17


```
{
  "_id": ObjectId('648ec86132607fa54f186207'),
  "icon": {
    "category": "utes",
    "type": "2 Wheel Drive",
    "shape": "Double Cab Pickup",
    "iconimg": "icon.iconimg_1687079009723_2022 Nissan Navara Icon.png"
  },
  "preview": {
    "colours": Array,
    "make": "Toyota",
    "model": "Maxima",
    "year": 2007,
    "v": 0
  }
}
```

PHOENIX Home Store Contact


CARS SUVs VANS UTES




FORD RANGER 2022
4 X 4 | PICKUP




ISUZU D-MAX 2022
AWD | DOUBLE CAB PICKUP



MAZDA BT-50 2022
AWD | DUAL CAB PICKUP



NISSAN NAVARA 2022
4 X 4 | DOUBLE CAB PICKUP



TOYOTA MAXIMA 2007
2 WHEEL DRIVE | DOUBLE CAB PICKUP

iii. Delete

Postman interface showing a DELETE request to `http://localhost:3000/api/vehicle/delete/648ec86132607fa54f186207`. The response is a JSON object: `{ "message": "Vehicle Deleted" }`.

MongoDB Compass interface showing the `phoenix.vehicles` collection. The collection contains two documents:

```

{
  "make": "Mazda",
  "model": "BT-50",
  "year": 2022,
  "__v": 0
}

{
  "_id": ObjectId("648dba2ebe79f20f3cada918"),
  "icon": Object,
  "preview": Object,
  "colours": Array,
  "make": "Nissan",
  "model": "Navara",
  "year": 2022,
  "__v": 0
}

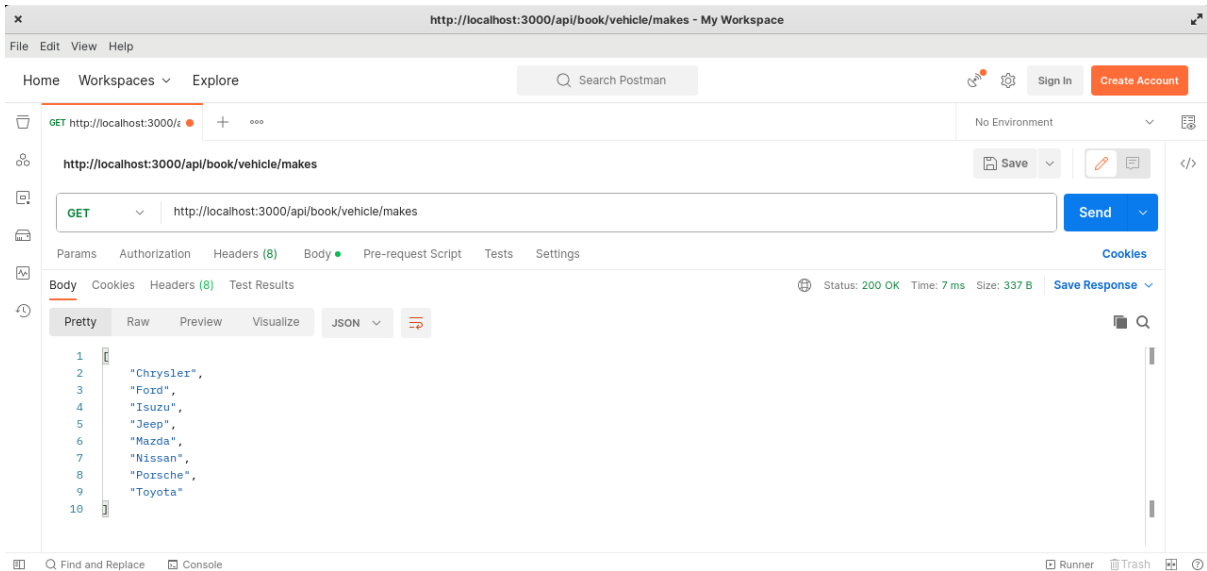
```

Phoenix website showing a grid of four pickup trucks:

- FORD RANGER 2022 4 X 4 | PICKUP
- ISUZU D-MAX 2022 4WD | DOUBLE CAB PICKUP
- MAZDA BT-50 2022 4WD | DUAL CAB PICKUP
- NISSAN NAVARA 2022 4 X 4 | DOUBLE CAB PICKUP

D. Book

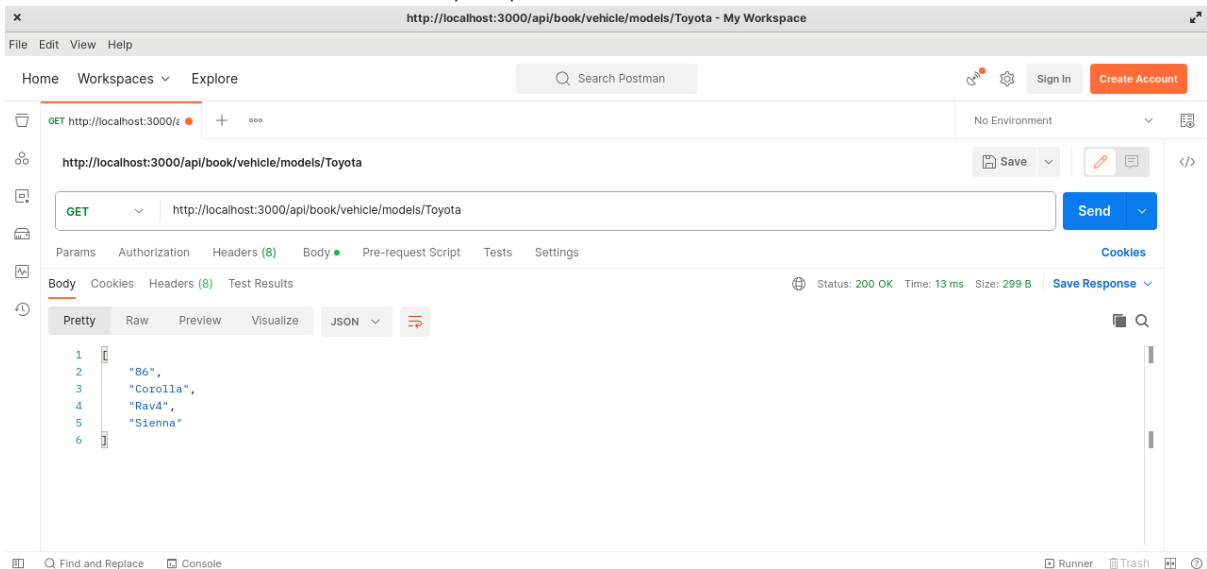
a. Get All Vehicle Makes



Postman interface showing a GET request to `http://localhost:3000/api/book/vehicle/makes`. The response is a JSON array of vehicle makes.

```
1 ["Chrysler",
2  "Ford",
3  "Isuzu",
4  "Jeep",
5  "Mazda",
6  "Nissan",
7  "Porsche",
8  "Toyota"]
```

b. Get All Distinct Vehicle Models by a Specific Make



Postman interface showing a GET request to `http://localhost:3000/api/book/vehicle/models/Toyota`. The response is a JSON array of distinct vehicle models for Toyota.

```
1 ["B6",
2  "Corolla",
3  "Rav4",
4  "Sienna"]
```

c. Get All Colours for a Specific Vehicle

The screenshot shows the Postman application interface. At the top, the browser tab is titled "http://localhost:3000/api/book/vehicle/colours/Toyota/Corolla - My Workspace". The main workspace displays a GET request to the URL "http://localhost:3000/api/book/vehicle/colours/Toyota/Corolla". The request has been sent, and the response is visible in the "Body" tab, which is set to "Pretty" view. The response status is "200 OK" with a time of "11 ms" and a size of "377 B". The response body is a JSON object containing an array of colors and an ID.

```
1 {
2   "colours": [
3     "Sparkling Brown",
4     "Sunfire Red",
5     "Seaweed Green",
6     "Pearl Yellow"
7   ],
8   "_id": "648da7eebe79f20f3cada90b"
9 }
```

The bottom of the interface shows a "Find and Replace" bar and a "Console" tab.

3. Database Testing

A. Headings

a. Find One By Page

```
+ x webuser webuser
phoenix> db.headings.findOne({page: "home"})
{
  _id: ObjectId("64859e06b521bf085a554dde"),
  page: 'home',
  heading: 'The Future Looks Bright',
  subheading: 'Our Vision of Tomorrow is Safe',
  isunderline: false,
  islight: true
}
phoenix>
```

B. Locations

a. Find All

```
phoenix> db.locations.find()
[
  {
    _id: ObjectId("6485a758b521bf085a554de5"),
    name: 'Auckland',
    phone: '3075089',
    address: {
      street: '289 Greenland Ave',
      city: 'Hamilton',
      country: 'New Zealand',
      postcode: 3582
    },
    hours: {
      sunday: [ '11am', '10pm' ],
      monday: [ '9am', '10am' ],
      tuesday: [ '9am', '10pm' ],
      wednesday: [ '9am', '10pm' ],
      thursday: [ '9am', '10pm' ],
      friday: [ '9am', '10pm' ],
      saturday: [ '9am', '10pm' ]
    },
    link: 'https://www.google.com/maps/place/Auckland/@-36.8588062,174.3764653,10z/data=!3m1!4b1!4m6!3m5!1s0x6d6d47fb5a9ce6fb:0x500ef6143a29917!8m2!3d-36.8508827!4d174.7644881!16zL20vMD0EydHM?entry=ttu'
  },
  {
    _id: ObjectId("6485a60eb521bf485a554de4"),
    name: 'Head Office',
    phone: '3035026',
    isheadoffice: true,
    address: {
      street: '15 King Street',
      city: 'Hamilton',
      country: 'New Zealand',
      postcode: 3159
    },
    hours: {
      sunday: [ '9am', '10pm' ],
      monday: [ '9am', '10am' ],
      tuesday: [ '9am', '10pm' ],
      wednesday: [ '9am', '10pm' ],
      thursday: [ '9am', '10pm' ],
      friday: [ '9am', '10pm' ],
      saturday: [ '9am', '10pm' ]
    },
    email: 'headoffice@phoenix.co.nz',
    link: 'https://www.google.com/maps/place/Hamilton/@-37.7776566,175.1659885,12z/data=!3m1!4b1!4m6!3m5!1s0x6d6d227487a63621:0x500ef6143a29903!8m2!3d-37.7825893!4d175.2527624!16zL20vMD0x3X3M?entry=ttu'
  }
]
```

b. Find All Store Names

```
phoenix> db.locations.find({}, {"name": 1}).pretty()
[
  { _id: ObjectId("6485a758b521bf085a554de5"), name: 'Auckland' },
  { _id: ObjectId("6485a60eb521bf485a554de4"), name: 'Head Office' },
  { _id: ObjectId("648dbcaf01b4bd8da1b10a5f"), name: 'Tauranga' },
  { _id: ObjectId("648dbcf901b4bd8da1b10a61"), name: 'Wellington' },
  { _id: ObjectId("648dbd8a01b4bd8da1b10a62"), name: 'New Plymouth' }
]
phoenix>
```

Find By Store Name

```

phoenix> db.locations.find({"name": "Auckland"}).pretty()
[
  {
    _id: ObjectId("6485a758b521bf085a554de5"),
    name: 'Auckland',
    phone: '3075089',
    address: {
      street: '289 Greenland Ave',
      city: 'Hamilton',
      country: 'New Zealand',
      postcode: 3582
    },
    hours: {
      sunday: [ '11am', '10pm' ],
      monday: [ '9am', '10am' ],
      tuesday: [ '9am', '10pm' ],
      wednesday: [ '9am', '10pm' ],
      thursday: [ '9am', '10pm' ],
      friday: [ '9am', '10pm' ],
      saturday: [ '9am', '10pm' ]
    },
    link: 'https://www.google.com/maps/place/Auckland/@-36.8588062,174.3764653,10z/data=!3m1!4b1!4m6!3m5!1s0x6d0d47fb5a9ce6fb:0x500ef6143a29917!8m2!3d-36.8508827!4d174.7644881!16zL20vMDEydHM?entry=ttu'
  }
]
phoenix>

```

c. Find Head Office

```

phoenix> db.locations.find({"isheadoffice": true}).pretty()
[
  {
    _id: ObjectId("6485a60eb521bf485a554de4"),
    name: 'Head Office',
    phone: '3035026',
    isheadoffice: true,
    address: {
      street: '15 King Street',
      city: 'Hamilton',
      country: 'New Zealand',
      postcode: 3159
    },
    hours: {
      sunday: [ '9am', '10pm' ],
      monday: [ '9am', '10am' ],
      tuesday: [ '9am', '10pm' ],
      wednesday: [ '9am', '10pm' ],
      thursday: [ '9am', '10pm' ],
      friday: [ '9am', '10pm' ],
      saturday: [ '9am', '10pm' ]
    },
    email: 'headoffice@phoenix.co.nz',
    link: 'https://www.google.com/maps/place/Hamilton/@-37.7776566,175.1659885,12z/data=!3m1!4b1!4m6!3m5!1s0x6d6d227487a63621:0x500ef6143a29903!8m2!3d-37.7825893!4d175.2527624!16zL20vMGx3X3M?entry=ttu'
  }
]

```

C. Vehicles

a. Find By ID

```
phoenix> db.vehicles.find({"_id": ObjectId("648d99cdbe79f20f3cada904")}).pretty()
[
  {
    _id: ObjectId("648d99cdbe79f20f3cada904"),
    icon: {
      category: 'suvs',
      type: '4 x 4',
      shape: 'Double Cab Pickup',
      iconimg: 'icon.iconimg_1687001549818_2021 Ford Explorer Icon.png'
    },
    preview: {
      features: [
        'Push Start',
        '3.5 Ton Towing Capacity',
        'Power Roller Shutter'
      ],
      description: [
        'The Ford Ranger is the pick-up truck always ready to work or play, with the power and capability for the toughest jobs, it also has the comfort and smart safety features you want for your family.'
      ],
      price: 52500,
      edition: 'Standard',
      previewimg: 'preview.previewimg_1687001549820_2021 Ford Explorer Preview.png'
    },
    colours: [
      'Cactus Gray',
      'Carbonized Gray',
      'Cyber Orange',
      'Velocity Blue'
    ],
    make: 'Ford',
    model: 'Explorer',
    year: 2021,
    __v: 0
  }
]
```

b. Find By Category

```
phoenix> db.vehicles.find({"icon.category": "cars"}).pretty()
[
  {
    _id: ObjectId("648da7eebe79f20f3cada90b"),
    icon: {
      category: 'cars',
      type: '2 Wheel Drive',
      shape: 'Hatchback',
      iconimg: 'icon.iconimg_1687005166699_2013 Toyota Corolla GX Icon.png'
    },
    preview: {
      features: [
        'Hand-Treated Materials',
        'Electric Mirrors',
        'Emersive Sound System'
      ],
      description: [
        'Just about the only thing we didn't update in the latest Toyota Corolla is its reputation for exceptional driving. The European-inspired design, clever technology and thoughtful features make every model in the Toyota Corolla range a joy to drive.'
      ],
      price: 88769,
      edition: 'GX',
      previewimg: 'preview.previewimg_1687005166699_2013 Toyota Corolla GX Preview.png'
    },
    colours: [
      'Sparkling Brown',
      'Sunfire Red',
      'Seaweed Green',
      'Pearl Yellow'
    ],
    make: 'Toyota',
    model: 'Corolla',
    year: 2013,
    __v: 0
  },
  {
    _id: ObjectId("648dab75be79f20f3cada90c"),
    icon: {
      category: 'cars',
      type: '2 Wheel Drive',
      shape: 'Coupe',
      iconimg: 'icon.iconimg_1687006069983_2015 Toyota 86 Icon.png'
    },
    preview: {
      features: [ 'Hand-Treated Materials', 'Luxurious Design', 'Racing Tires' ]
    }
  }
]
```


c. Update By ID

```
phoenix> db.vehicles.updateOne({"make": "Jeep", "model": "Wagoneer"}, {$set: {"make": "Jeep"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
phoenix> db.vehicles.findOne({"make": "Jeep", "model": "Wagoneer"})
{
  _id: ObjectId("648d9fdfbe79f20f3cada909"),
  icon: {
    category: 'suvs',
    type: '4 x 4',
    shape: 'SUV',
    iconimg: 'icon.iconimg_1687003103029_2022 Jeep Grand Wagoneer Icon.png'
  },
  preview: {
    features: [
      'Hand-Treated Materials',
      'Cinematic Sound',
      'Full Colour, Complete Clarity'
    ],
    description: [
      'Built for those who want to share a lifetime of success with the people they hold dear. The All-New G
      rand Wagoneer and Wagoneer reinterpret their own legacies and take the premium SUV into uncharted territorie
      s of style, comfort and adventure. These are more than sport utility vehicles. They're the means to make exc
      eptional memories and create your own legend.'
    ],
    price: 88769,
    edition: 'Standard',
    previewimg: 'preview.previewimg_1687003103030_2022 Jeep Grand Wagoneer Preview.png'
  },
  colours: [ 'Silver Zynith', 'Baltic Gray', 'Valvet Red', 'Bright White' ],
  make: 'Jeep',
  model: 'Wagoneer',
  year: 2022,
  __v: 0
}
phoenix>
```

d. Delete By ID

```
phoenix> db.vehicles.remove({"_id": ObjectId("648d9d19be79f20f3cada908")})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkW
rite.
{ acknowledged: true, deletedCount: 1 }
phoenix> █
```

D. Book

a. Get All Vehicle Makes

```
phoenix> db.vehicles.find({}, {"make": 1}).pretty()
[
  { _id: ObjectId("648d9fdfbe79f20f3cada909"), make: 'Jeep' },
  { _id: ObjectId("648da20bbe79f20f3cada90a"), make: 'Porsche' },
  { _id: ObjectId("648da7eebe79f20f3cada90b"), make: 'Toyota' },
  { _id: ObjectId("648dab75be79f20f3cada90c"), make: 'Toyota' },
  { _id: ObjectId("648dacc1be79f20f3cada90d"), make: 'Nissan' },
  { _id: ObjectId("648dadabbe79f20f3cada90e"), make: 'Ford' },
  { _id: ObjectId("648daf09be79f20f3cada90f"), make: 'Chrysler' },
  { _id: ObjectId("648db0e5be79f20f3cada910"), make: 'Chrysler' },
  { _id: ObjectId("648db1e9be79f20f3cada911"), make: 'Ford' },
  { _id: ObjectId("648db302be79f20f3cada913"), make: 'Toyota' },
  { _id: ObjectId("648db461be79f20f3cada914"), make: 'Ford' },
  { _id: ObjectId("648db61bbe79f20f3cada916"), make: 'Isuzu' },
  { _id: ObjectId("648db871be79f20f3cada917"), make: 'Mazda' },
  { _id: ObjectId("648dba2ebe79f20f3cada918"), make: 'Nissan' }
]
phoenix>
```

b. Get All Distinct Vehicle Models by a Specific Make

```
phoenix> db.vehicles.distinct("model", {"make": "Toyota"})
[ '86', 'Corolla', 'Sienna' ]
phoenix> █
```

c. Get All Colours for a Specific Vehicle

```
phoenix> db.vehicles.distinct("colours", {"make": "Toyota", "model": "86"})
[ 'Beach Brown', 'Diamond White', 'Graphite Gray', 'Sparkling Red' ]
phoenix>
```

References

DCodeMania. (2021). *#01 MEVN Full Stack Web Application Tutorial | Working on Back-End* [Video]. YouTube.

https://www.youtube.com/watch?v=-gdG_P85fFQ&ab_channel=DCodeMania