

POSTMAN TESTING

1.Mohammad

Admin login: Insert into the table

The screenshot shows the Postman interface for a POST request to `localhost:8081/api/v1/admins`. The Body tab is selected, containing the following JSON payload:

```
1 {
2   "admin": "admin",
3   "password": "Frozen123@23"
4 }
```

The response shows a Result Grid with one row of data:

	admin_id	admin	password
▶	1	admin	Frozen123@23
*	NULL	NULL	NULL

Admin login authentication

The screenshot shows the Postman interface for a POST request to `localhost:8081/api/v1/admin`. The Body tab is selected, containing the same JSON payload as the previous screenshot:

```
1 {
2   "admin": "admin",
3   "password": "Frozen123@23"
4 }
```

The response status is 202 Accepted, with a response time of 379 ms and a size of 328 B. The response body is:

```
1 {
2   "admin": "admin",
3   "password": "Frozen123@23",
4   "adminAuthenticated": true
5 }
```

2.Sravan

User Registration

The screenshot shows the Postman interface for a POST request to `localhost:8081/api/users`. The request body is a JSON object:

```
1 {
2   "first_name": "jay",
3   "last_name": "jones",
4   "mobile_no": "9988336678",
5   "age": "86",
6   "username": "jay@example.com",
7   "password": "Frozen123@28",
8   "gender": "male"
9 }
```

The response is a database result grid showing two rows of data:

	user_id	age	first_name	gender	last_name	mobile_no	password	username
▶	1	22	Lee	Male	Cooper	9988776655	Lee123@12	lee@email.com
*	2	86	jay	male	jones	9988336678	Frozen123@28	jay@example.com
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

User login

The screenshot shows the Postman interface for a POST request to `localhost:8081/api/user`. The request body is a JSON object:

```
1 {
2   "username": "lee@email.com",
3   "password": "Lee123@12"
4 }
```

The response status is 202 Accepted with a response body:

```
1 {
2   "username": "lee@email.com",
3   "password": "Lee123@12",
4   "authenticated": true
5 }
```

3.Navya

Get by username

The screenshot shows the Postman application interface. At the top, there's a header bar with the URL `localhost:8081/api/users/lee@example.com`, a 'Save' button, and a 'Send' button. Below the header is a search bar containing the same URL. The main workspace has a 'GET' method selected, with the URL `localhost:8081/api/users/lee@email.com`. Underneath the method, tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' are visible, with 'Headers (6)' currently active. To the right of these tabs is a 'Cookies' tab. A 'Query Params' section contains two rows: one with 'Key' and 'Value' columns, and another with 'Key' and 'Value' columns. Below this is a row of buttons for 'Body', 'Cookies', 'Headers (9)', and 'Test Results', with 'Body' being the active tab. To the right of these buttons are status indicators: a globe icon, '200 OK', '184 ms', '449 B', and a 'Save Response' button. Below the status indicators is a toolbar with 'Pretty', 'Raw', 'Preview', 'Visualize', and 'JSON' dropdown buttons, along with a search icon. The main content area displays a JSON response with 10 numbered lines. The response body is as follows:

```
1  "userId": 1,
2  "first_name": "Lee",
3  "last_name": "Cooper",
4  "mobile_no": "9988776655",
5  "age": 22,
6  "username": "lee@email.com",
7  "password": "Lee123@12",
8  "gender": "Male"
```

Update by username

4. Manasa

Insert into the contact us table

The screenshot shows a POST request to `localhost:8081/api/v3/contactUs`. The Body tab contains the following JSON:

```
1 {
2   "name": "Jayanth",
3   "emailId": "jayanth@gmail.com",
4   "contactNo": 1234567891,
5   "description": "regarding policies"
6 }
```

The Result Grid displays the following data:

c_id	contact_no	description	email_id	name
1	9988776655	Good Service	lee@emailcom	lee
2	1234567891	regarding policies	jayanth@gmail.com	Jayanth
*	NULL	NULL	NULL	NULL

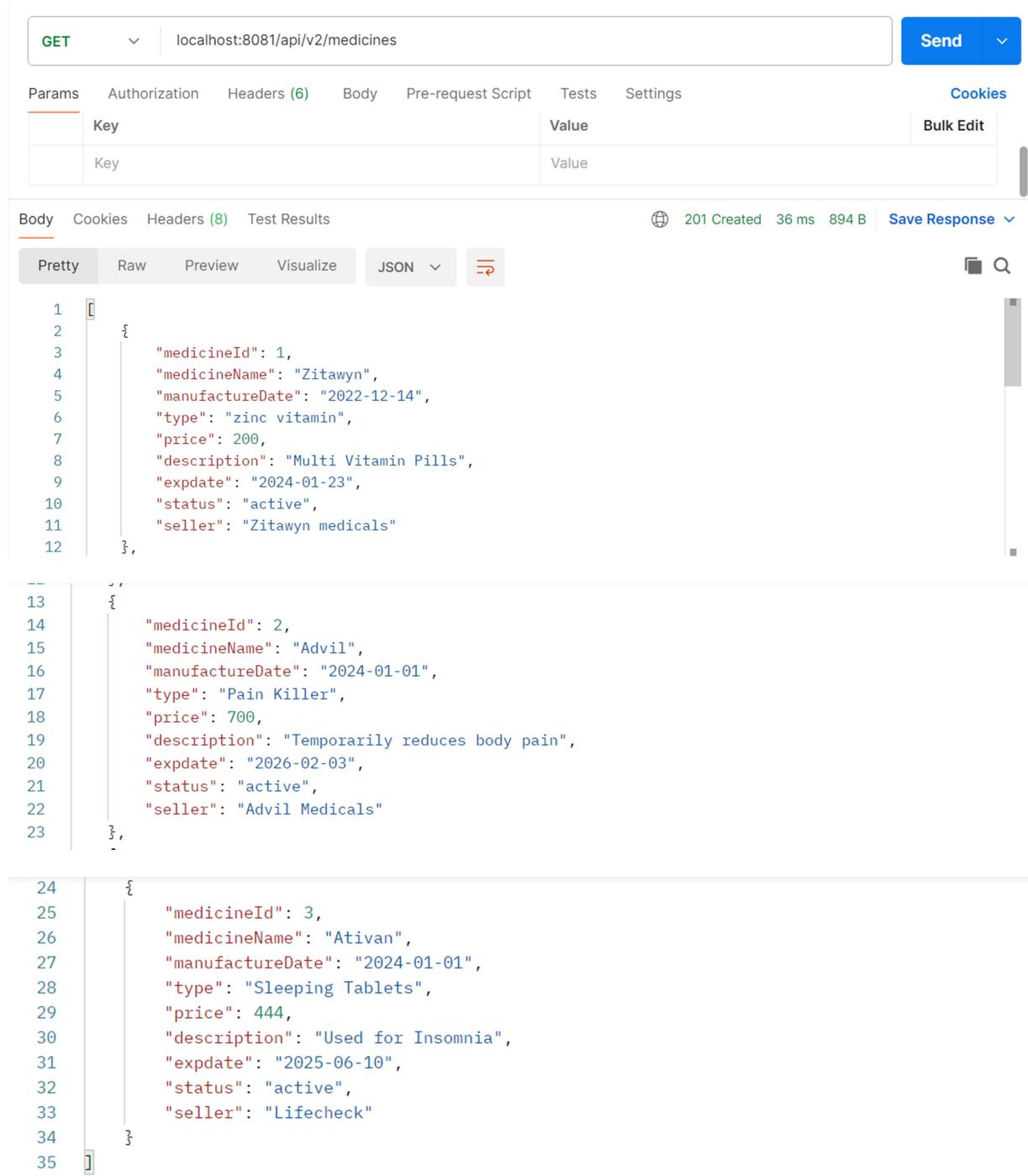
Get all

The screenshot shows a GET request to `localhost:8081/api/v3/contactUs`. The Body tab indicates "This request does not have a body". The Response tab shows the following JSON:

```
2 {
3   "cId": 1,
4   "name": "lee",
5   "emailId": "lee@emailcom",
6   "contactNo": 9988776655,
7   "description": "Good Service"
8 },
9 {
10  "cId": 2,
11  "name": "Jayanth",
12  "emailId": "jayanth@gmail.com",
13  "contactNo": 1234567891,
```

5.Priyanka M V

Get all from medicine table

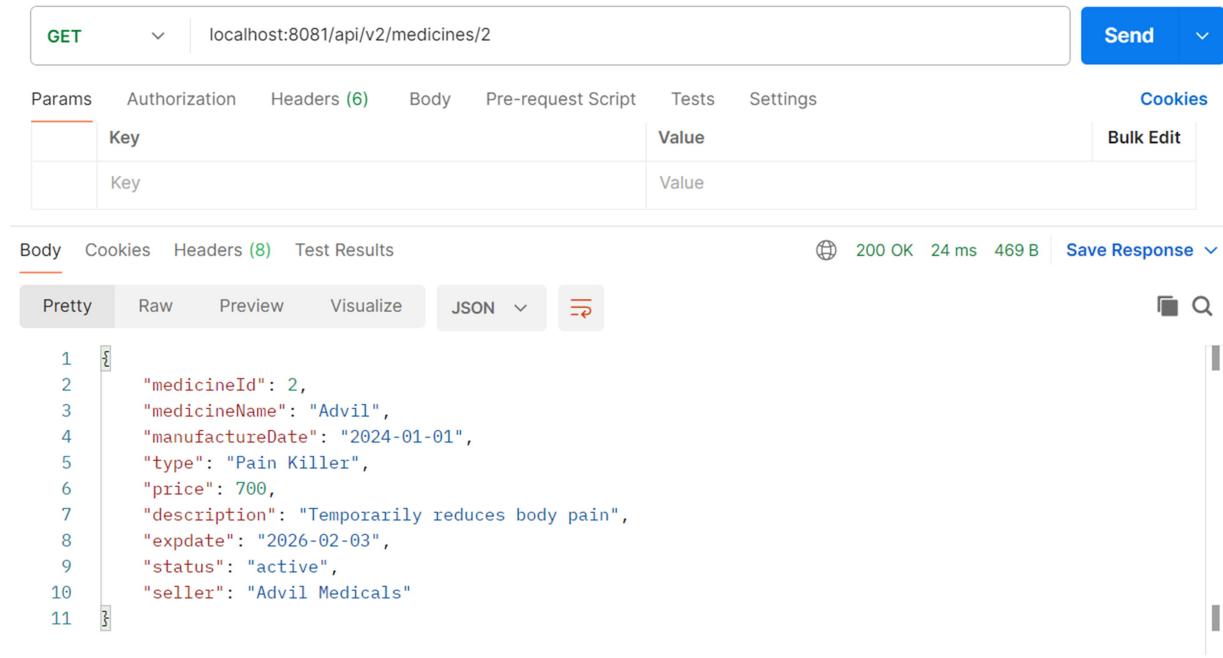


The screenshot shows the Postman application interface. At the top, there is a header bar with 'GET' selected, the URL 'localhost:8081/api/v2/medicines', and a 'Send' button. Below the header are tabs for 'Params', 'Authorization', 'Headers (6)', 'Body', 'Pre-request Script', 'Tests', and 'Settings'. A 'Cookies' section is also present. Under the 'Params' tab, there are two rows for 'Key' and 'Value'. The 'Body' tab is selected, showing a JSON response with three objects. The 'Pretty' tab is active, displaying the JSON with line numbers and indentation. The response body is as follows:

```
1 [ { 2   "medicineId": 1, 3   "medicineName": "Zitawyn", 4   "manufactureDate": "2022-12-14", 5   "type": "zinc vitamin", 6   "price": 200, 7   "description": "Multi Vitamin Pills", 8   "expdate": "2024-01-23", 9   "status": "active", 10  "seller": "Zitawyn medicals" 11 }, 12 13  { 14    "medicineId": 2, 15    "medicineName": "Advil", 16    "manufactureDate": "2024-01-01", 17    "type": "Pain Killer", 18    "price": 700, 19    "description": "Temporarily reduces body pain", 20    "expdate": "2026-02-03", 21    "status": "active", 22    "seller": "Advil Medicals" 23 }, 24  { 25    "medicineId": 3, 26    "medicineName": "Ativan", 27    "manufactureDate": "2024-01-01", 28    "type": "Sleeping Tablets", 29    "price": 444, 30    "description": "Used for Insomnia", 31    "expdate": "2025-06-10", 32    "status": "active", 33    "seller": "Lifecheck" 34 } ]
```

Below the JSON response, there is a status bar showing '201 Created' with a timestamp of '36 ms' and a size of '894 B', along with a 'Save Response' dropdown.

Get by id the medicine



GET <localhost:8081/api/v2/medicines/2> Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Key	Value
Key	Value

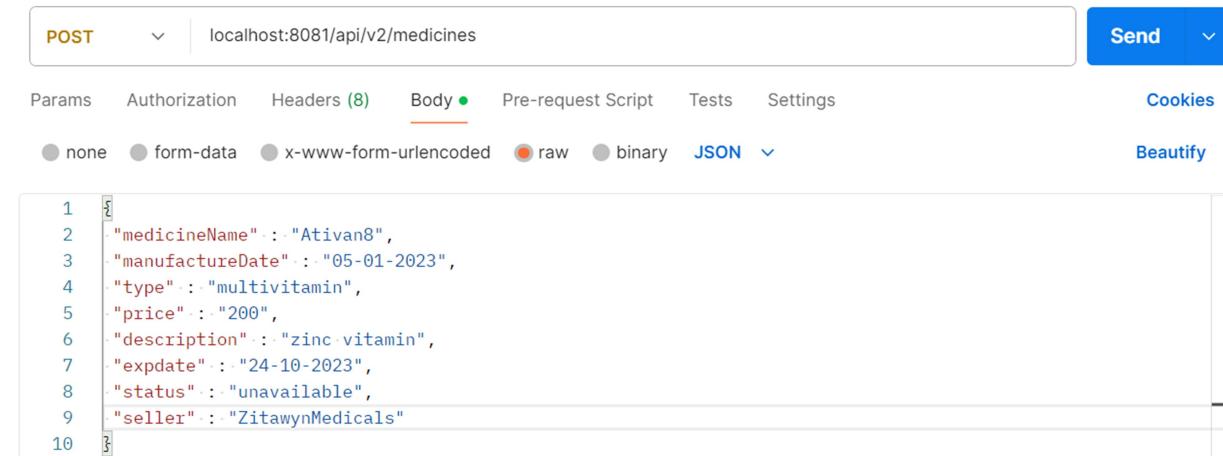
Body Cookies Headers (8) Test Results 200 OK 24 ms 469 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {"medicineId": 2,
2 "medicineName": "Advil",
3 "manufactureDate": "2024-01-01",
4 "type": "Pain Killer",
5 "price": 700,
6 "description": "Temporarily reduces body pain",
7 "expdate": "2026-02-03",
8 "status": "active",
9 "seller": "Advil Medicals"
10
11 }
```

6.Lokesh

Add medicine



POST <localhost:8081/api/v2/medicines> Send

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

none form-data x-www-form-urlencoded raw binary JSON Beautify

```
1 {
2   "medicineName": "Ativan8",
3   "manufactureDate": "05-01-2023",
4   "type": "multivitamin",
5   "price": "200",
6   "description": "zinc vitamin",
7   "expdate": "24-10-2023",
8   "status": "unavailable",
9   "seller": "ZitawynMedicals"
10 }
```

Body Cookies Headers (8) Test Results 201 Created 49 ms 466 B Save Response

Pretty Raw Preview Visualize JSON 🔗 🖨️ 🔎

```

1  {
2    "medicineId": 5,
3    "medicineName": "Ativan8",
4    "manufactureDate": "05-01-2023",
5    "type": "multivitamin",
6    "price": 200,
7    "description": "zinc vitamin",
8    "expdate": "24-10-2023",
9    "status": "unavailable",
10   "seller": "ZitawynMedicals"
11 }

```

Delete medicine

DELETE localhost:8081/api/v2/medicines/5 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

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

This request does not have a body

Body Cookies Headers (8) Test Results 202 Accepted 85 ms 280 B Save Response

Pretty Raw Preview Visualize Text JSON 🔗 🖨️ 🔎

```

1 deleted successfully

```

7.Pravin

Update medicine

PUT localhost:8081/api/v2/medicines Send

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

none form-data x-www-form-urlencoded raw binary JSON Beautify

```

1  {
2    "medicineId": "6",
3    "medicineName": "Zinc Tablet",
4    "manufactureDate": "05-01-2023",
5    "type": "multivitamin",
6    "price": "200",
7    "description": "zinc vitamin",
8    "expdate": "24-10-2023",
9    "status": "unavailable",
10   "seller": "ZitawynMedicals"
11 }

```

Body Cookies Headers (8) Test Results

200 OK 63 ms 465 B Save Response

Pretty

Raw

Preview

Visualize

JSON ▾



```
1 {
2   "medicineId": 6,
3   "medicineName": "Zinc Tablet",
4   "manufactureDate": "05-01-2023",
5   "type": "multivitamin",
6   "price": 200,
7   "description": "zinc vitamin",
8   "expdate": "24-10-2023",
9   "status": "unavailable",
10  "seller": "ZitawynMedicals"
11 }
```

8.Rakesh

Get all from cart

GET



localhost:8081/api/v5/cart

Send ▾

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

Key

Value

Bulk Edit

Key

Value

Body

Cookies

Headers (8)

Test Results

200 OK

23 ms

344 B

Save Response ▾

Pretty

Raw

Preview

Visualize

JSON ▾



```
1 [
2   {
3     "medicineId": 1,
4     "medicineName": "Zitawyn",
5     "price": 200,
6     "description": "Multi Vitamin Pills"
7   }
8 ]
```

Add to cart

The screenshot shows a POST request to `localhost:8081/api/v5/cart`. The Body tab is selected, containing the following JSON payload:

```
1
2   {
3     "medicineId": 2,
4     "medicineName": "Advil",
5     "price": 700,
6     "description": "Temporarily reduces body pain"
7   }
8
```

The Result Grid shows the following data:

	medicine_id	description	medicine_name	price
▶	1	Multi Vitamin Pills	Zitawyn	200
▶	2	Temporarily reduces body pain	Advil	700
*	NULL	NULL	NULL	NULL

Get by id

The screenshot shows a GET request to `localhost:8081/api/v5/cart/1`. The Response tab displays the following JSON data:

```
1
2   {
3     "medicineId": 1,
4     "medicineName": "Zitawyn",
5     "price": 200,
6     "description": "Multi Vitamin Pills"
7 }
```

Delete from cart

The screenshot shows the Postman interface with the following details:

- Method:** DELETE
- URL:** localhost:8081/api/v5/cart/2
- Status:** 200 OK (70 ms, 274 B)
- Body:** 1 deleted successfully
- Result Grid:** A table showing the deleted item:

	medicine_id	description	medicine_name	price
▶	1	Multi Vitamin Pills	Zitawyn	200
*	NULL	NULL	NULL	NULL