

# Grocery App -

## Admin Functionality -

### 1) Register Grocery Item -

The screenshot shows a REST client interface with a POST request to `http://localhost:14101/GroceryDirectory/V1/Admin/Grocery/Register`. The request body is a JSON object: `{ "ItemName": "Sugar", "Price": 45, "Quantity": 200 }`. The response status is 201 Created, with a time of 977 ms and a size of 96 B. The response body is `{ "ItemName": "Sugar" }`.

POST Register... GET Get Groc... DEL Delete Gr... PATCH Update... POST Register... + ... No Environment

Grocery / Register Grocery Item Save ...

POST http://localhost:14101/GroceryDirectory/V1/Admin/Grocery/Register Send

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

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

```
1 {
2   "ItemName": "Sugar",
3   "Price": 45,
4   "Quantity": 200
5 }
```

Body Cookies Headers (2) Test Results Status: 201 Created Time: 977 ms Size: 96 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "ItemName": "Sugar"
3 }
```

Runner 20°C 00:28 05-12-2023

### DB -

The screenshot shows a database query tool with a SQL query: `SELECT * FROM public.items ORDER BY item_name ASC`. The results are displayed in a table with 5 rows and 7 columns: `item_name`, `item_price`, `quantity`, `no_of_available_items`, `no_of_purchased_items`, and `status`.

Query Query History Scratch Pad

```
1 SELECT * FROM public.items
2 ORDER BY item_name ASC
```

Data Output Messages Notifications

	item_name [PK] character varying (50)	item_price integer	quantity integer	no_of_available_items integer	no_of_purchased_items integer	status character varying
1	Apple	20	1000	1000	0	IN_STOCK
2	Mango	50	10	10	0	IN_STOCK
3	Rice	200	190	190	0	IN_STOCK
4	Sugar	45	200	200	0	IN_STOCK
5	Wheat	2000	1990	1990	0	IN_STOCK

Total rows: 5 of 5 Query complete 00:00:00.083 Ln 1, Col 0031 05-12-2023

## 2) Retrieve Grocery Item -

The screenshot shows a REST client interface with a tab titled "GET Get Groc...". The request is a GET method to the URL `http://localhost:14101/GroceryDirectory/V1/Admin/Grocery/Retrieve`. The response status is 200 OK, with a time of 897 ms and a size of 667 B. The response body is displayed in JSON format, showing two items: Mango and Rice.

```
1 {
2   {
3     "ItemName": "Mango",
4     "Price": 50,
5     "Quantity": 10,
6     "NumberOfAvailableItems": 10,
7     "NumberOfSoldItems": 0,
8     "Status": "IN_STOCK"
9   },
10  {
11    "ItemName": "Rice",
12    "Price": 200,
13    "Quantity": 190,
14    "NumberOfAvailableItems": 190,
15    "NumberOfSoldItems": 0,
16    "Status": "IN_STOCK"
17  }
18 }
```

The screenshot shows a REST client interface with a tab titled "GET Get Groc...". The request is a GET method to the URL `http://localhost:14101/GroceryDirectory/V1/Admin/Grocery/Retrieve`. The response status is 200 OK, with a time of 897 ms and a size of 667 B. The response body is displayed in JSON format, showing two items: Sugar and Apple.

```
18 {
19   {
20     "ItemName": "Sugar",
21     "Price": 45,
22     "Quantity": 200,
23     "NumberOfAvailableItems": 200,
24     "NumberOfSoldItems": 0,
25     "Status": "IN_STOCK"
26   },
27   {
28     "ItemName": "Apple",
29     "Price": 20,
30     "Quantity": 1000,
31     "NumberOfAvailableItems": 1000,
32     "NumberOfSoldItems": 0,
33     "Status": "IN_STOCK"
34   }
35 }
```

POST Register...GET Get Groc...XDEL Delete Gr...PATCH Update...POST Register...+...No Environment

Grocery / Get Grocery ItemSaveSend

GEThttp://localhost:14101/GroceryDirectory/V1/Admin/Grocery/Retrieve

ParamsAuthorizationHeaders (8)Body●Pre-request ScriptTestsSettingsCookies</>

BodyCookiesHeaders (2)Test ResultsStatus: 200 OKTime: 897 msSize: 667 BSave Response<?>

PrettyRawPreviewVisualizeJSON≡

```
26  {
27    "ItemName": "Apple",
28    "Price": 20,
29    "Quantity": 1000,
30    "NumberOfAvailableItems": 1000,
31    "NumberOfSoldItems": 0,
32    "Status": "IN_STOCK"
33  },
34  {
35    "ItemName": "Wheat",
36    "Price": 2000,
37    "Quantity": 1990,
38    "NumberOfAvailableItems": 1990,
39    "NumberOfSoldItems": 0,
40    "Status": "IN_STOCK"
41  }
42 }
```

Runner19°C00:4405-12-2023ENG

### 3) Remove grocery items from the system

The screenshot shows a REST client interface with a tab titled "Delete Grocery Item". The request method is "DELETE" and the URL is "http://localhost:14101/GroceryDirectory/V1/Admin/Grocery/Apple/Delete". The status bar indicates "Status: 204 No Content", "Time: 42 ms", and "Size: 27 B". The response body is empty, showing only a "1" in the Pretty view.

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body

1

### DB - (Deleted Apple)

The screenshot shows a database query interface with a query editor containing the following SQL:

```
1 SELECT * FROM public.items
2 ORDER BY item_name ASC
```

The query has been executed, and the results are displayed in the Data Output tab. The results show 4 rows of data:

	item_name [PK] character varying (50)	item_price integer	quantity integer	no_of_available_items integer	no_of_purchased_items integer	status character varying
1	Mango	50	10	10	0	IN_STOCK
2	Rice	200	190	190	0	IN_STOCK
3	Sugar	45	200	200	0	IN_STOCK
4	Wheat	2000	1990	1990	0	IN_STOCK

Successfully run. Total query runtime: 92 msec. 4 rows affected.

Total rows: 4 of 4 Query complete 00:00:00.092 Ln 1, Col 1

#### 4) Update details (e.g., name, price, manage inventory) of existing grocery items

The screenshot shows a REST client interface with a PATCH request to `http://localhost:14101/GroceryDirectory/V1/Admin/Grocery/Mango/Update`. The request body is a JSON object: `{ "itemName": "Mango", "Price": 130, "Quantity": 250 }`. The response status is 200 OK, and the response body is a JSON object: `{ "itemName": "Mango", "Price": 130, "Quantity": 250, "NumberOfAvailableItems": 250, "NumberOfSoldItems": 0, "Status": "IN_STOCK" }`.

```
1 {
2   "itemName": "Mango",
3   "Price": 130,
4   "Quantity": 250
5 }
```

```
1 {
2   "itemName": "Mango",
3   "Price": 130,
4   "Quantity": 250,
5   "NumberOfAvailableItems": 250,
6   "NumberOfSoldItems": 0,
7   "Status": "IN_STOCK"
8 }
```

DB -

The screenshot shows a database query result for the query `SELECT * FROM public.items ORDER BY item_name ASC`. The result is a table with 4 rows and 6 columns: `item_name`, `item_price`, `quantity`, `no_of_available_items`, `no_of_purchased_items`, and `status`.

	item_name [PK] character varying (50)	item_price integer	quantity integer	no_of_available_items integer	no_of_purchased_items integer	status character varying
1	Mango	130	250	250	0	IN_STOCK
2	Rice	200	190	190	0	IN_STOCK
3	Sugar	45	200	200	0	IN_STOCK
4	Wheat	2000	1990	1990	0	IN_STOCK

Successfully run. Total query runtime: 79 msec. 4 rows affected.

Total rows: 4 of 4 Query complete 00:00:00.079 Ln 1, Col 1

# User Responsibilities:

## 1) View the list of available grocery items

### DB - (Consumed Sugar)

Query Query History

```
1 SELECT * FROM public.items
2 ORDER BY item_name ASC
```

Scratch Pad x

Data Output Messages Notifications

	Item_name [PK] character varying (50)	Item_price integer	quantity integer	no_of_available_items integer	no_of_purchased_items integer	status character varying
1	Mango	130	250	250	0	IN_STOCK
2	Rice	200	190	190	0	IN_STOCK
3	Sugar	45	200	0	200	OUT_OF_STOCK
4	Wheat	2000	1990	1990	0	IN_STOCK

Total rows: 4 of 4 Query complete 00:00:00.079 Ln 1, Col 1

POST Regi... GET Get G... DEL Delet... PATCH Up... POST Regi... GET Get G... No Environment

Grocery / Get Grocery Item Copy Save

GET http://localhost:14101/GroceryDirectory/V1/User/Grocery/Retrieve Send

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

Body Cookies Headers (2) Test Results Status: 200 OK Time: 26 ms Size: 432 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "ItemName": "Rice",
4     "Price": 200,
5     "Quantity": 190,
6     "NumberOfAvailableItems": 190,
7     "NumberOfSoldItems": 0,
8     "Status": "IN_STOCK"
9   },
10  {
11    "ItemName": "Wheat",
12    "Price": 2000,
13    "Quantity": 1990,
14    "NumberOfAvailableItems": 1990,
15    "NumberOfSoldItems": 0,
16    "Status": "IN_STOCK"
17  }
18 }
```

Runner 01:03 05-12-2023

POST Regl...GET Get G...DEL Delet...PATCH Up...POST Regl...GET Get G...+...No Environment

Grocery / Get Grocery Item CopySave...

GEThttp://localhost:14101/GroceryDirectory/V1/User/Grocery/RetrieveSend

ParamsAuthorizationHeaders (8)Body●Pre-request ScriptTestsSettingsCookies</>

BodyCookiesHeaders (2)Test ResultsStatus: 200 OKTime: 26 msSize: 432 BSave Response

PrettyRawPreviewVisualizeJSON...

```
10  {
11    "ItemName": "Wheat",
12    "Price": 2000,
13    "Quantity": 1990,
14    "NumberOfAvailableItems": 1990,
15    "NumberOfSoldItems": 0,
16    "Status": "IN_STOCK"
17  },
18  {
19    "ItemName": "Mango",
20    "Price": 130,
21    "Quantity": 250,
22    "NumberOfAvailableItems": 250,
23    "NumberOfSoldItems": 0,
24    "Status": "IN_STOCK"
25  }
26 }
```

Runner19°C01:0305-12-2023ENG

## 2) Register User -

The screenshot shows a REST client interface with a POST request to `http://localhost:14101/GroceryDirectory/V1/User/Register`. The request body is a JSON object: `{ "FirstName": "Rohan", "LastName": "Badade", "Email": "badaderohan@gmail.com" }`. The response status is 201 Created, with a time of 1700 ms and a size of 116 B. The response body is a JSON object: `{ "UserId": "RohBad2", "FirstName": "Rohan" }`.

POST Regl... GET Get G... DEL Delet... PATCH Up... POST Regl... GET Get G... + ... No Environment

Grocery / Register User Item Save ...

POST http://localhost:14101/GroceryDirectory/V1/User/Register Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies </> Beautify

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

```
1 {
2   "FirstName": "Rohan",
3   "LastName": "Badade",
4   "Email": "badaderohan@gmail.com"
5 }
```

Body Cookies Headers (2) Test Results Status: 201 Created Time: 1700 ms Size: 116 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "UserId": "RohBad2",
3   "FirstName": "Rohan"
4 }
```

Runner 01:11 05-12-2023

## DB -

The screenshot shows a database query runner interface. The query is `SELECT * FROM public.user_details ORDER BY user_id ASC`. The query has been successfully executed, resulting in 1 row affected. The data output table shows the following information:

	user_id [PK] character varying	first_name character varying	last_name character varying	email character varying	is_email_validated boolean
1	RohBad2	Rohan	Badade	badaderohan@gmail.com	false

Query Query History Scratch Pad

```
1 SELECT * FROM public.user_details
2 ORDER BY user_id ASC
```

Data Output Messages Notifications

user\_id [PK] character varying first\_name character varying last\_name character varying email character varying is\_email\_validated boolean

1 RohBad2 Rohan Badade badaderohan@gmail.com false

Successfully run. Total query runtime: 113 msec. 1 rows affected.

Total rows: 1 of 1 Query complete 00:00:00.113 Ln 2, Col 22

01:12 05-12-2023

## 3) Ability to book multiple grocery items in a single order



POST R... GET Ge... DEL De... PATCH U... POST R... GET Ge... POST R... + ... No Environment

Grocery / Register Booking Details

POST http://localhost:14101/GroceryDirectory/V1/User/Book Send

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

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

```
1 {
2   "UserId": "RohBad2",
3   "SendOrderDetailsOnEmail": "NO",
4   "GroceryItems": [
5     {
6       "ItemName": "Mango",
7       "Quantity": 100
8     },
9     {
10      "ItemName": "Rice",
11      "Quantity": 190
12    }
13  ]
14 }
```

Body Cookies Headers (2) Test Results Status: 200 OK Time: 69 ms Size: 315 B Save Response

Runner

19°C 01:14 05-12-2023

POST R... GET Ge... DEL De... PATCH U... POST R... GET Ge... POST R... + ... No Environment

Grocery / Register Booking Details

POST http://localhost:14101/GroceryDirectory/V1/User/Book Send

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

Body Cookies Headers (2) Test Results Status: 200 OK Time: 69 ms Size: 315 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "purchasedGroceryItemsList": [
3     {
4       "ItemName": "Mango",
5       "NumberOfUnitsPurchased": 100,
6       "TotalPrice": 13000
7     },
8     {
9       "ItemName": "Rice",
10      "NumberOfUnitsPurchased": 190,
11      "TotalPrice": 30000
12    }
13  ],
14   "ReceiptId": "ed54ecfd-4f15-4be8-af05-27780ca04d04",
15   "TotalBillAmount": 61000
16 }
```

Runner

19°C 01:14 05-12-2023

DB -

## Items -

Query

Query History

Scratch Pad

1

2

SELECT \* FROM public.items

ORDER BY item\_name ASC

Data Output

Messages

Notifications

item\_name

item\_price

quantity

no\_of\_available\_items

no\_of\_purchased\_items

status

[PK] character varying (50)

integer

integer

integer

integer

character varying

1

2

3

4

Mango

130

250

150

100

IN\_STOCK

Rice

200

190

0

190

OUT\_OF\_STOCK

Sugar

45

200

0

200

OUT\_OF\_STOCK

Wheat

2000

1990

1990

0

IN\_STOCK

Total rows: 4 of 4

Query complete 00:00:00.077

Ln 1, Col 1

✓ Successfully run. Total query runtime: 77 msec. 4 rows affected.

## Booking Details -

Query

Query History

Scratch Pad

1

2

SELECT \* FROM public.booking\_details

ORDER BY receipt\_id ASC

Data Output

Messages

Notifications

receipt\_id

user\_id

bill\_amount

[PK] uuid

character varying

integer

1

ed54ecfd-4f15-4be8-af05-27780ca04d04

RohBad2

51000

Total rows: 1 of 1

Query complete 00:00:00.179

Ln 1, Col 1

## Purchased Goods -

Query Query History 

Scratch Pad x 

```
1 SELECT * FROM public.purchased_goods
2 ORDER BY id ASC
```

Data Output Messages Notifications 

	receipt_id uuid	item_name character varying	no_of_purchased_units integer	total_price integer	id [PK] bigint
1	ed54ecd4-f1f5-4be8-af05-27780ca04d04	Mango	100	13000	9
2	ed54ecd4-f1f5-4be8-af05-27780ca04d04	Rice	190	38000	10

✓ Successfully run. Total query runtime: 169 msec. 2 rows affected. ✕

Total rows: 2 of 2    Query complete 00:00:00.169

Ln 1, Col 1