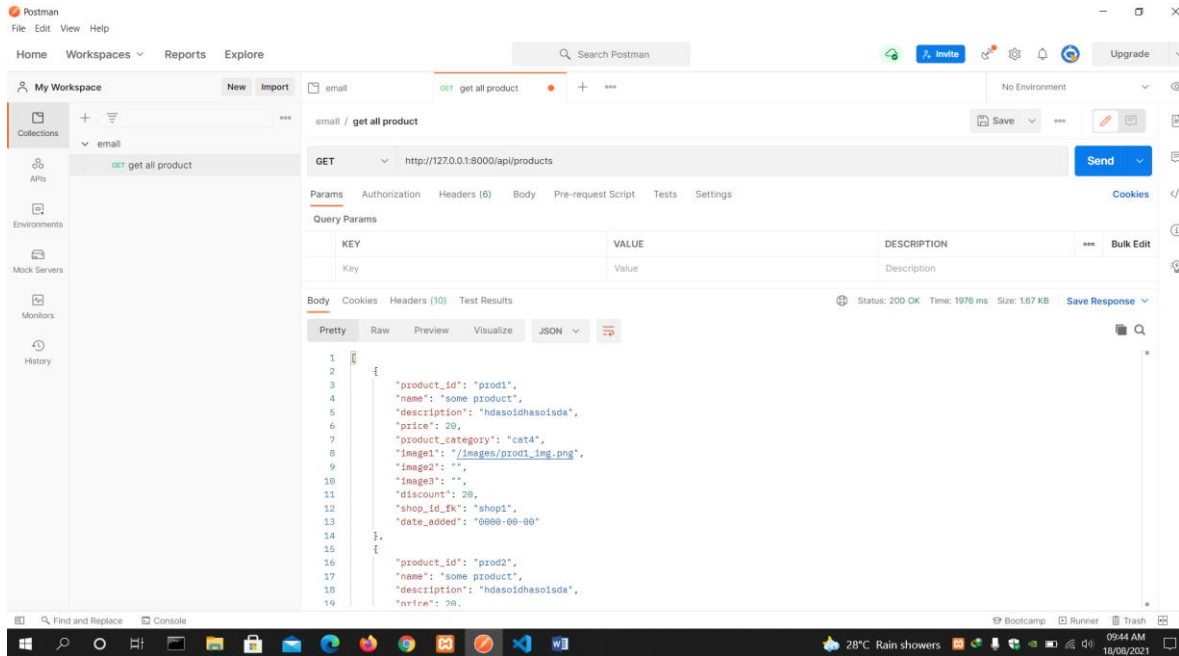


# Testing and request log of email backend system

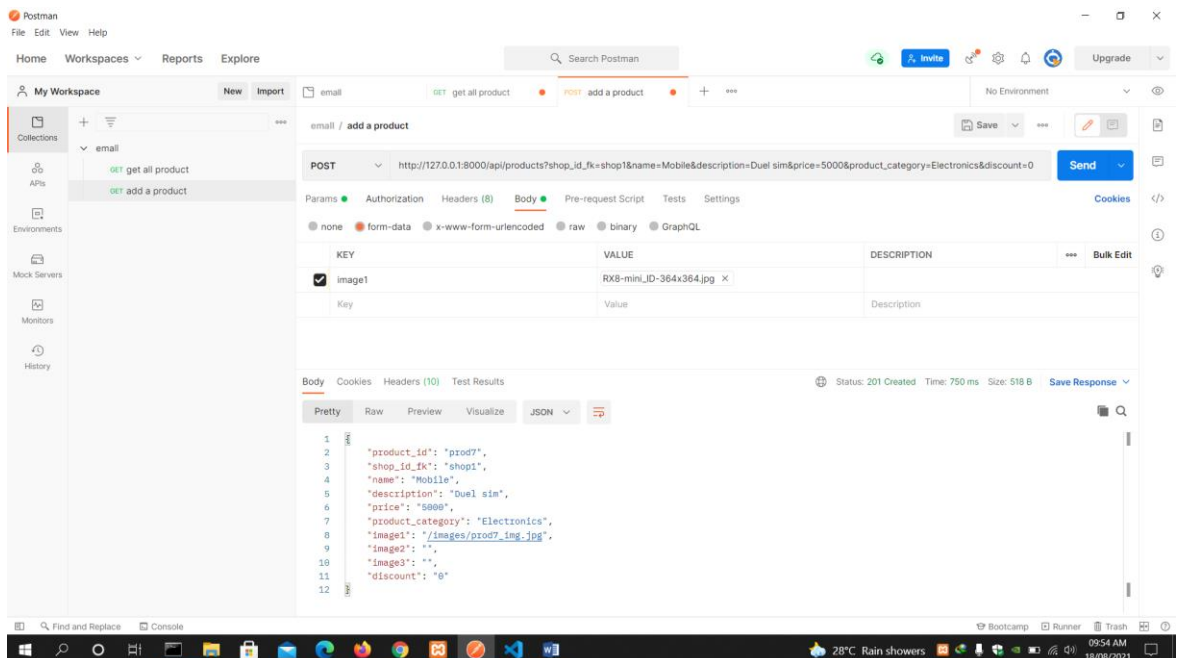
## Get all products



Postman interface showing a GET request to `http://127.0.0.1:8000/api/products`. The response is a JSON array of two product objects.

```
1 {
2   {
3     "product_id": "prod1",
4     "name": "some product",
5     "description": "hdsasoidhasoisda",
6     "price": 20,
7     "product_category": "cat4",
8     "image1": "/images/prod1_img.png",
9     "image2": "",
10    "image3": "",
11    "discount": 20,
12    "shop_id_fk": "shop1",
13    "date_added": "0000-00-00"
14  },
15  {
16    "product_id": "prod2",
17    "name": "some product",
18    "description": "hdsasoidhasoisda",
19    "price": 20,
```

## Add a product using post



Postman interface showing a POST request to `http://127.0.0.1:8000/api/products?shop_id_fk=shop1&name=Mobile&description=Duel sim&price=5000&product_category=Electronics&discount=0`. The response is a 201 Created status.

```
1 {
2   "product_id": "prod7",
3   "shop_id_fk": "shop1",
4   "name": "Mobile",
5   "description": "Duel sim",
6   "price": "5000",
7   "product_category": "Electronics",
8   "image1": "/images/prod7_img.jpg",
9   "image2": "",
10  "image3": "",
11  "discount": "0"
12 }
```

## Get all categories

The screenshot shows the Postman application interface. On the left sidebar, under 'My Workspace', there is a collection named 'email' containing three requests: 'get all products', 'add a product', and 'get all categories'. The 'get all categories' request is selected. The main panel displays the details of this GET request to the endpoint `http://127.0.0.1:8000/api/categories`. The 'Query Params' section is empty. The 'Body' tab is active, showing a JSON array of five category objects in 'Pretty' format. The status bar at the bottom indicates a successful response with status 200 OK, time 304 ms, and size 644 B.

```
1 {
2   {
3     "category_id": "cat1",
4     "name": "Bags"
5   },
6   {
7     "category_id": "cat2",
8     "name": "Beauty Shop"
9   },
10  {
11    "category_id": "cat3",
12    "name": "Clothing"
13  },
14  {
15    "category_id": "cat4",
16    "name": "Electronics"
17  },
18  {
19    "category_id": "cat5",
```

## Add a category using post

The screenshot shows the Postman application interface with a new POST request created. The request is named 'add a category' and is targeted to the endpoint `http://127.0.0.1:8000/api/categories?name=Medicine`. The 'Query Params' section shows a single parameter 'name' with the value 'Medicine'. The 'Body' tab is active, displaying a JSON object in 'Pretty' format: `{ "category_id": "cat9", "name": "Medicine" }`. The status bar at the bottom shows the response status as 201 Created, with a time of 335 ms and a size of 352 B.

```
1 {
2   "category_id": "cat9",
3   "name": "Medicine"
4 }
```

## Get all subcategories

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/api/subcategories`. The response is a JSON array of three subcategory objects. The status is 200 OK, and the response size is 2.46 KB.

KEY	VALUE	DESCRIPTION
Key	Value	Description

```
1 {
2   "sub_cat_id": "sub1",
3   "category_id_fk": "cat3",
4   "name": "Man's Clothing"
5 },
6 {
7   "sub_cat_id": "sub18",
8   "category_id_fk": "cat4",
9   "name": "TV"
10 },
11 {
12   "sub_cat_id": "sub11",
13   "category_id_fk": "cat8",
14   "name": "Man's Watch"
15 },
16 {
17   "sub_cat_id": "sub12",
18   "category_id_fk": "cat8",
19 }
```

## Add a subcategory

The screenshot shows the Postman interface with a POST request to `http://127.0.0.1:8000/api/subcategories?category_id_fk=cat9&name=First Aid`. The response is a JSON object representing the newly created subcategory. The status is 201 Created, and the response size is 377 B.

KEY	VALUE	DESCRIPTION
category_id_fk	cat9	
name	First Aid	
Key	Value	Description

```
1 {
2   "category_id_fk": "cat9",
3   "name": "First Aid",
4   "sub_cat_id": "sub34"
5 }
```

## Get all childcategories

The screenshot shows the Postman interface with a GET request configured to `http://127.0.0.1:8000/api/childcategories`. The request is sent, and the response is displayed in the 'Body' tab, showing a JSON array of three child categories.

**Request:**

- Method: GET
- URL: `http://127.0.0.1:8000/api/childcategories`

**Response (JSON):**

```
[{"child_cat_id": "child1", "name": "T-shirt", "sub_cat_id_fk": "sub1"}, {"child_cat_id": "child10", "name": "Sandals", "sub_cat_id_fk": "sub3"}, {"child_cat_id": "child11", "name": "Sandals", "sub_cat_id_fk": "sub4"}]
```

## Add a childcategory

The screenshot shows the Postman interface with a POST request configured to `http://127.0.0.1:8000/api/childcategories?sub_cat_id_fk=sub34&name=Antiseptic`. The request is sent, and the response is displayed in the 'Body' tab, showing a JSON object representing the newly added child category.

**Request:**

- Method: POST
- URL: `http://127.0.0.1:8000/api/childcategories?sub_cat_id_fk=sub34&name=Antiseptic`

**Response (JSON):**

```
{"sub_cat_id_fk": "sub34", "name": "Antiseptic", "child_cat_id": "child20"}
```

## Get all shops

Postman interface showing a GET request to `http://127.0.0.1:8000/api/shops`. The response is a JSON array of three shop objects.

**Query Params:**

KEY	VALUE	DESCRIPTION
Key	Value	Description

**Body:**

```
1 {
2   {
3     "shop_id": "shop1",
4     "shop_name": "Walton",
5     "category_id_fk": "cat4",
6     "logo": "/logos/shop1_img.jpg"
7   },
8   {
9     "shop_id": "shop2",
10    "shop_name": "Apple",
11    "category_id_fk": "cat4",
12    "logo": "/logos/shop2_img.jpg"
13  },
14  {
15    "shop_id": "shop3",
16    "shop_name": "Pharmacy.com.bd",
17    "category_id_fk": "cat9",
18    "logo": "/logos/shop3_img.jpg"
19  }
20 }
```

## Add a shop

Postman interface showing a POST request to `http://127.0.0.1:8000/api/shops?category_id_fk=cat9&shop_name=Pharmacy.com.bd`. The response is a 201 Created status.

**Body:**

```
1 {
2   "category_id_fk": "cat9",
3   "shop_name": "Pharmacy.com.bd",
4   "shop_id": "shop3",
5   "logo": "/logos/shop3_img.jpg"
6 }
```

## Get all shopparents

Postman interface showing a GET request to `http://127.0.0.1:8000/api/shopparents`. The response is a JSON array of two shop parent objects.

```
GET http://127.0.0.1:8000/api/shopparents
```

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

Query Params

KEY	VALUE	DESCRIPTION
-----	-------	-------------

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 185 ms Size: 853 B Save Response

```
2 {
3   "email": "iftekhar2@gmail.com",
4   "dob": "1997-03-02T00:00:00.000000Z",
5   "address": "dhaka",
6   "NID": 2131233,
7   "first_name": "Iftekhar",
8   "last_name": "Hyder",
9   "number": 1213123,
10  "password": "$2y$18$Bua12Gy3iU9Ev.FqnG7NuTVMb37j6uBpg8ko54m7Afy_4EvKC2",
11  "type": "staff",
12  "shop_id_fk": "shop1"
13 }
14 [
15   {
16     "email": "mazbaul7@gmail.com",
17     "dob": "1970-01-01T00:00:00.000000Z",
18     "address": "dinajpur",
19     "NID": 123456,
20     "first_name": "Mazbaul",
21     "last_name": "Hoque",
22     "number": 123456789,
23     "password": "$2y$18$mGb8YAQxepZ3cW4dAPQh.Mq.K16uNZi54nCBEdyI41ft.T8fRwz2",
24     "type": "owner",
25     "shop_id_fk": "shop3"
26   }
27 ]
```

## Add a shopparent

Postman interface showing a POST request to `http://127.0.0.1:8000/api/shopparents?shop_id_fk=shop3&email=mazbaul7@gmail.com&NID=123456&dob=23/9/1998&address=dinajpur&first_name=`. The response is a JSON object representing the created shop parent.

```
POST http://127.0.0.1:8000/api/shopparents?shop_id_fk=shop3&email=mazbaul7@gmail.com&NID=123456&dob=23/9/1998&address=dinajpur&first_name=
```

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

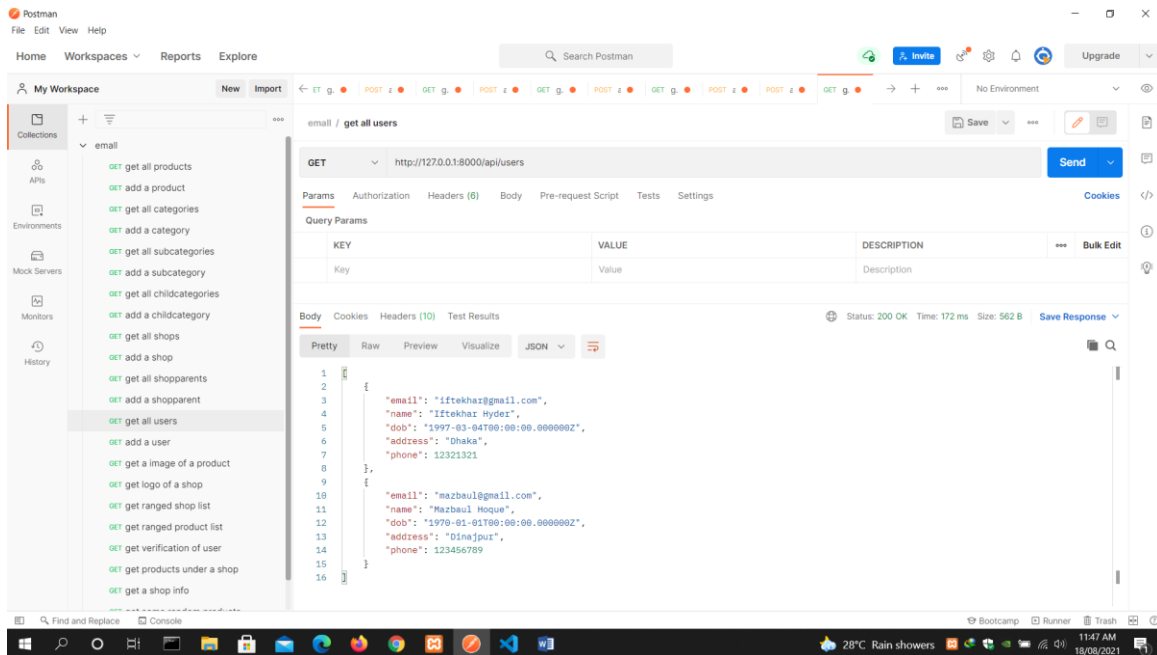
shop_id_fk	shop3
email	mazbaul7@gmail.com
NID	123456
dob	23/9/1998
address	dinajpur
first_name	Mazbaul
last_name	Hoque
number	0123456789
password	owner7
type	owner

Body Cookies Headers (10) Test Results

Status: 201 Created Time: 393 ms Size: 584 B Save Response

```
2 {
3   "shop_id_fk": "shop3",
4   "email": "mazbaul7@gmail.com",
5   "dob": "1970-01-01T00:00:00.000000Z",
6   "NID": 123456,
7   "first_name": "Mazbaul",
8   "last_name": "Hoque",
9   "number": 123456789,
10  "password": "$2y$18$mGb8YAQxepZ3cW4dAPQh.Mq.K16uNZi54nCBEdyI41ft.T8fRwz2",
11  "type": "owner",
12  "shop_id_fk": "shop3"
13 }
```

## Get all users



Postman interface showing a GET request to `http://127.0.0.1:8000/api/users`. The response is a JSON array of two user objects.

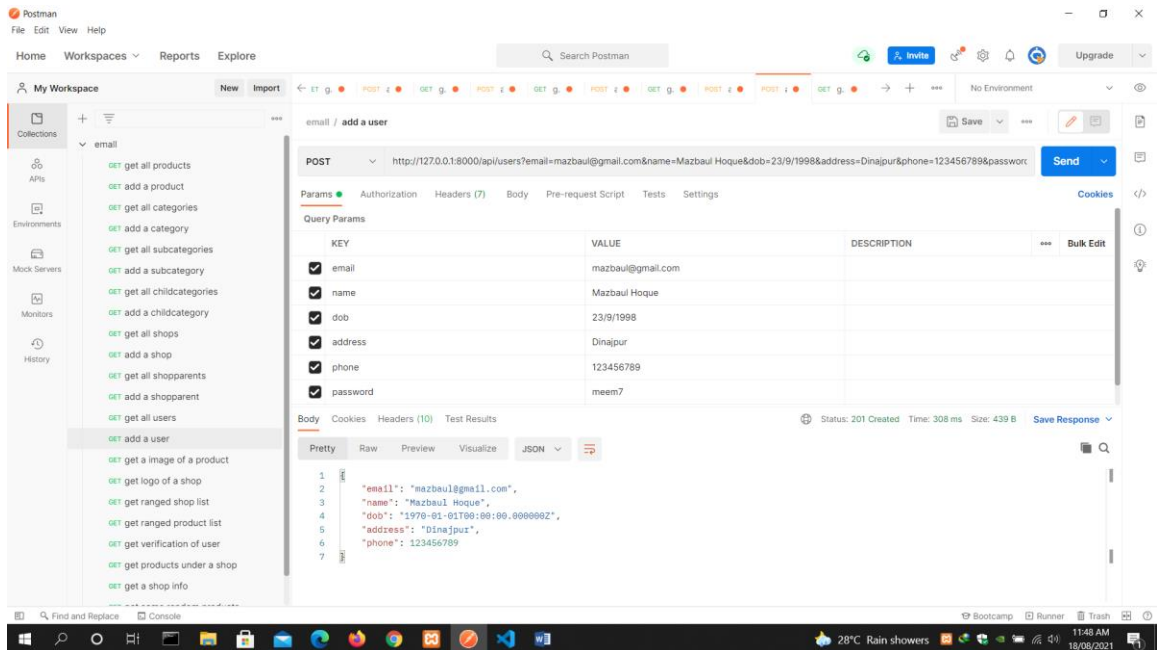
**Query Params**

KEY	VALUE	DESCRIPTION
Key	Value	Description

**Body**

```
1 {
2   {
3     "email": "iftexhar@gmail.com",
4     "name": "Iftexhar Hyder",
5     "dob": "1997-03-04T09:00:00.000000Z",
6     "address": "Dhaka",
7     "phone": 12321321
8   },
9   {
10    "email": "mazbaul@gmail.com",
11    "name": "Mazbaul Hoque",
12    "dob": "1979-01-01T00:00:00.000000Z",
13    "address": "Dinajpur",
14    "phone": 123456789
15  }
16 }
```

## Add a user



Postman interface showing a POST request to `http://127.0.0.1:8000/api/users?email=mazbaul@gmail.com&name=Mazbaul Hoque&dob=23/9/1998&address=Dinajpur&phone=123456789&password=meem7`. The response is a 201 Created status.

**Query Params**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> email	mazbaul@gmail.com	
<input checked="" type="checkbox"/> name	Mazbaul Hoque	
<input checked="" type="checkbox"/> dob	23/9/1998	
<input checked="" type="checkbox"/> address	Dinajpur	
<input checked="" type="checkbox"/> phone	123456789	
<input checked="" type="checkbox"/> password	meem7	

**Body**

```
1 {
2   "email": "mazbaul@gmail.com",
3   "name": "Mazbaul Hoque",
4   "dob": "1979-01-01T00:00:00.000000Z",
5   "address": "Dinajpur",
6   "phone": 123456789
7 }
```

## get an image of a product

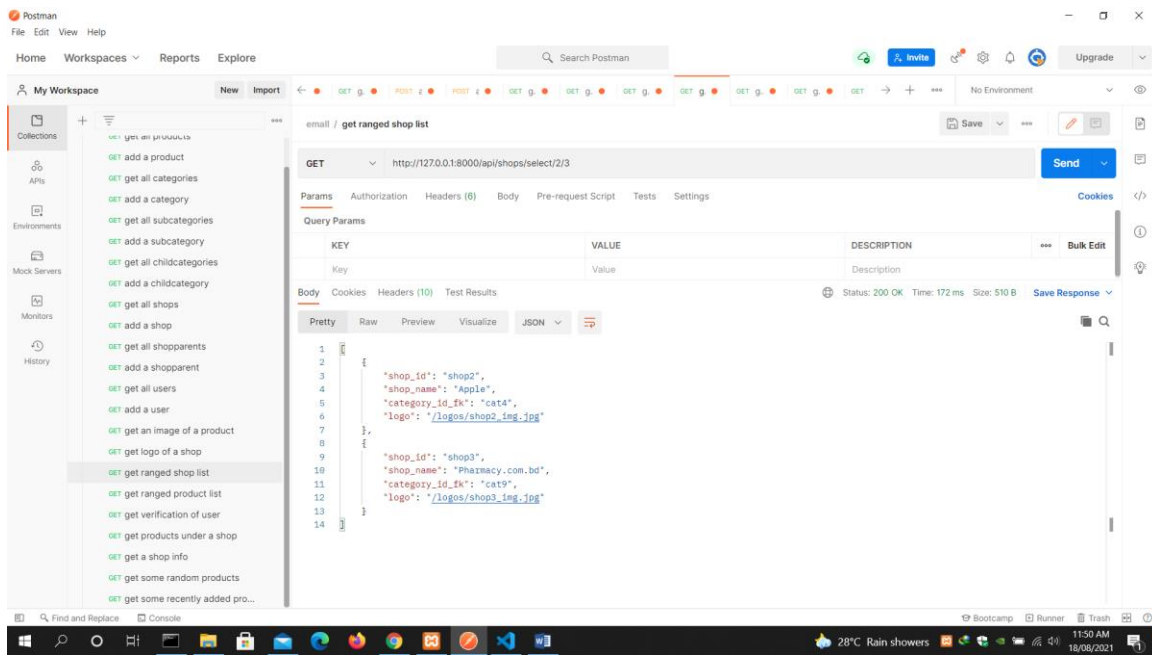
The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API endpoints under the 'email' collection. The main panel displays a GET request to the URL 'http://127.0.0.1:8000/api/images/prod1/1'. The request is saved, and the response is shown in the 'Body' tab, which is a large image. The status bar at the bottom indicates a 200 OK status, 196 ms response time, and a 66.32 KB response size.

## get logo of a shop

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API endpoints under the 'email' collection. The main panel displays a GET request to the URL 'http://127.0.0.1:8000/api/logos/shop3'. The request is saved, and the response is shown in the 'Body' tab, which is a large image. The status bar at the bottom indicates a 200 OK status, 139 ms response time, and a 6.4 KB response size.



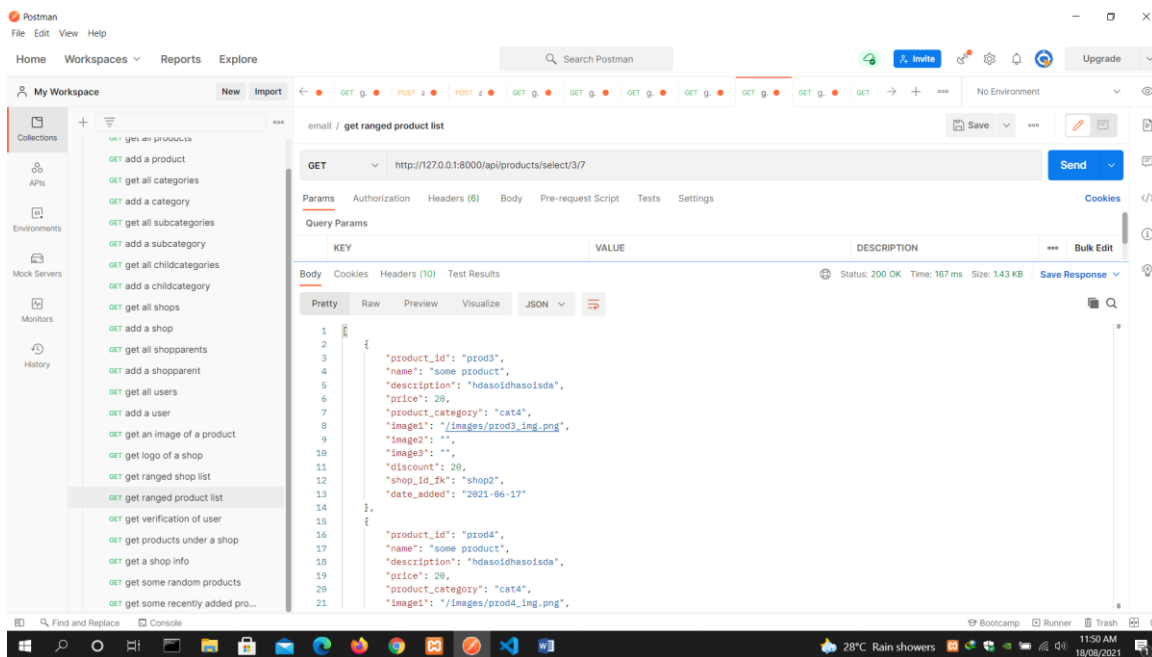
## get ranged shop list



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/api/shops/select/2/3`. The response is a JSON array of two shop objects. The status is 200 OK, and the response size is 510 B.

```
1 {
2   "shop_id": "shop2",
3   "shop_name": "Apple",
4   "category_id_fk": "cat4",
5   "logo": "/logos/shop2_img.jpg"
6 },
7 {
8   "shop_id": "shop3",
9   "shop_name": "Pharmacy.com.bd",
10  "category_id_fk": "cat9",
11  "logo": "/logos/shop3_img.jpg"
12 }
13 ]
14 }
```

## get ranged product list



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/api/products/select/3/7`. The response is a JSON array of two product objects. The status is 200 OK, and the response size is 1.43 KB.

```
1 {
2   "product_id": "prod3",
3   "name": "some product",
4   "description": "hdsasoidhasoisda",
5   "price": 20,
6   "product_category": "cat4",
7   "image1": "/images/prod3_img.png",
8   "image2": "",
9   "image3": "",
10  "discount": 20,
11  "shop_id_fk": "shop2",
12  "date_added": "2021-06-17"
13 },
14 {
15   "product_id": "prod4",
16   "name": "some product",
17   "description": "hdsasoidhasoisda",
18   "price": 20,
19   "product_category": "cat4",
20   "image1": "/images/prod4_img.png",
21 }
```

## get verification of user

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API endpoints. The main panel displays a GET request named 'get verification of user' with the URL 'http://127.0.0.1:8000/api/users/mazbaul@gmail.com/meem?'. The 'Query Params' section is empty. The 'Body' tab is selected, showing a '1 Match' result. The status bar at the bottom indicates a successful response with status 200 OK, time 211 ms, and size 320 B.

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body: 1 Match

## get products under a shop

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API endpoints. The main panel displays a GET request named 'get products under a shop' with the URL 'http://127.0.0.1:8000/api/shops/products/Walton'. The 'Query Params' section is empty. The 'Body' tab is selected, showing a JSON response with two product objects. The status bar at the bottom indicates a successful response with status 200 OK, time 164 ms, and size 997 B.

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body: Pretty Raw Preview Visualize JSON

```
1 {
2   {
3     "product_id": "prod1",
4     "name": "some product",
5     "description": "hdsasoidhasoisa",
6     "price": 20,
7     "product_category": "cat4",
8     "image1": "/images/prod1_img.png",
9     "image2": "",
10    "image3": "",
11    "discount": 20,
12    "shop_id_fk": "shop1",
13    "date_added": "0000-00-00"
14  },
15  {
16    "product_id": "prod2",
17    "name": "some product",
18    "description": "hdsasoidhasoisa",
19    "price": 20,
20    "product_category": "cat4",
21    "image1": "/images/prod2_img.png",
```

## get a shop info

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API endpoints. The main panel displays a GET request to the URL `http://127.0.0.1:8000/api/shops/getPharmacy.com.bd`. The request is configured with a 'Key' and 'Value' in the 'Query Params' section. The response is shown in the 'Body' tab, displaying a JSON object with the following data:

```
{  "shop_id": "shop3",  "shop_name": "Pharmacy.com.bd",  "category_id_fk": "cat9",  "logo": "/logos/shop3_img.jpg"}
```

The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 175 ms, and a size of 412 B.

## get some random products

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various API endpoints. The main panel displays a GET request to the URL `http://127.0.0.1:8000/api/products/random/3`. The request is configured with a 'Key' and 'Value' in the 'Query Params' section. The response is shown in the 'Body' tab, displaying a JSON array of three product objects:

```
[  {    "product_id": "prod1",    "name": "some product",    "description": "hdsasidhasoisda",    "price": 20,    "product_category": "cat4",    "image1": "/images/prod1_img.png",    "image2": "",    "image3": "",    "discount": 20,    "shop_id_fk": "shop1",    "date_added": "0000-00-00"  },  {    "product_id": "prod5",    "name": "some product",    "description": "hdsasidhasoisda",    "price": 20,    "product_category": "cat4",    "image1": "/images/prod5_img.png",    "image2": ""  }]
```

The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 182 ms, and a size of 997 B.

## get some recently added products

The image shows the Postman REST client interface. The workspace is named 'My Workspace'. A collection of API requests is visible on the left, including 'get some recently added products', 'add a product', 'get all categories', 'add a category', 'get all subcategories', 'add a subcategory', 'get all childcategories', 'add a childcategory', 'get all shops', 'add a shop', 'get all shopparents', 'add a shopparent', 'get all users', 'add a user', 'get an image of a product', 'get logo of a shop', 'get ranged shop list', 'get ranged product list', 'get verification of user', 'get products under a shop', 'get a shop info', 'get some random products', and 'get some recently added pro...'. The selected request is 'get some recently added products', which is a GET request to the endpoint 'http://127.0.0.1:8000/api/products/recent/3'. The response is a JSON array of two product objects, with a status of 200 OK, time of 198 ms, and size of 1005 B. The response body is displayed in the 'Pretty' format.

```
1 {
2   {
3     "product_id": "prod5",
4     "name": "some product",
5     "description": "hdasoldhasoisda",
6     "price": 20,
7     "product_category": "cat4",
8     "image1": "/images/prod5_img.png",
9     "image2": "",
10    "image3": "",
11    "discount": 20,
12    "shop_id_fk": "shop2",
13    "date_added": "2021-08-31"
14  },
15  {
16    "product_id": "prod4",
17    "name": "some product",
18    "description": "hdasoldhasoisda",
19    "price": 20,
20    "product_category": "cat4",
21    "image1": "/images/prod4_img.png",
```