

1-Register End-Point:

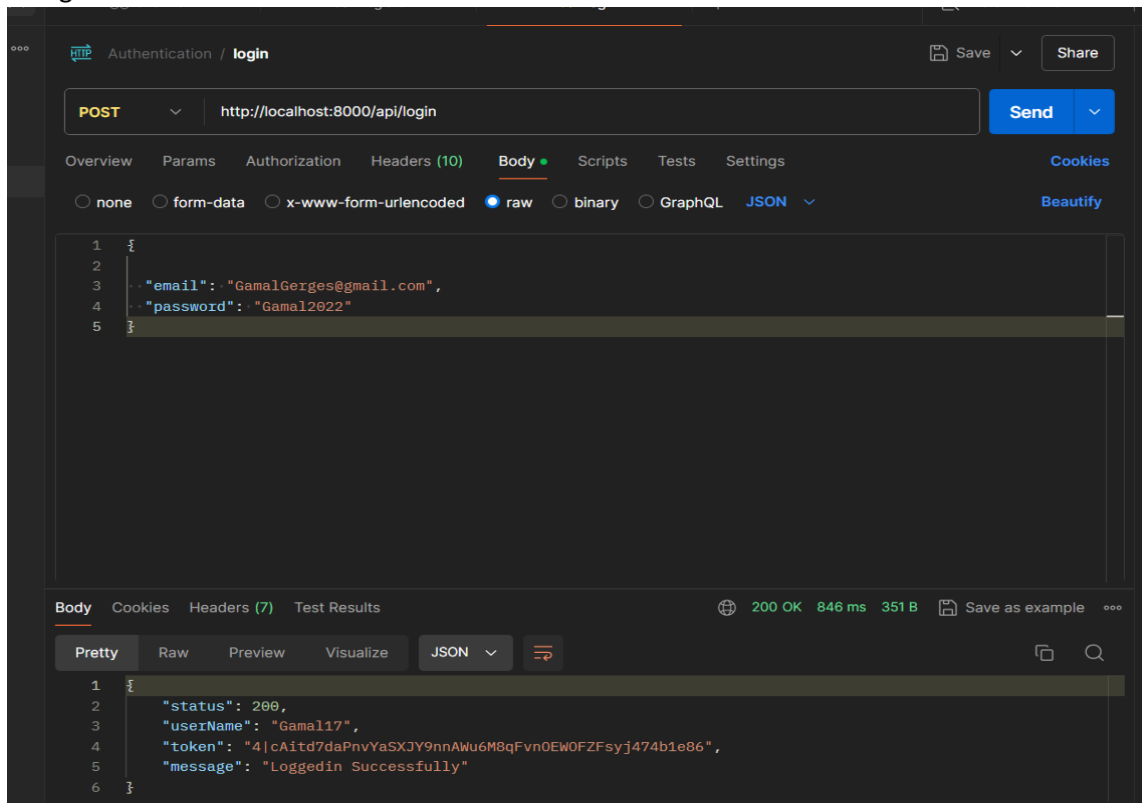
The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:8000/api/register
- Body (raw):**

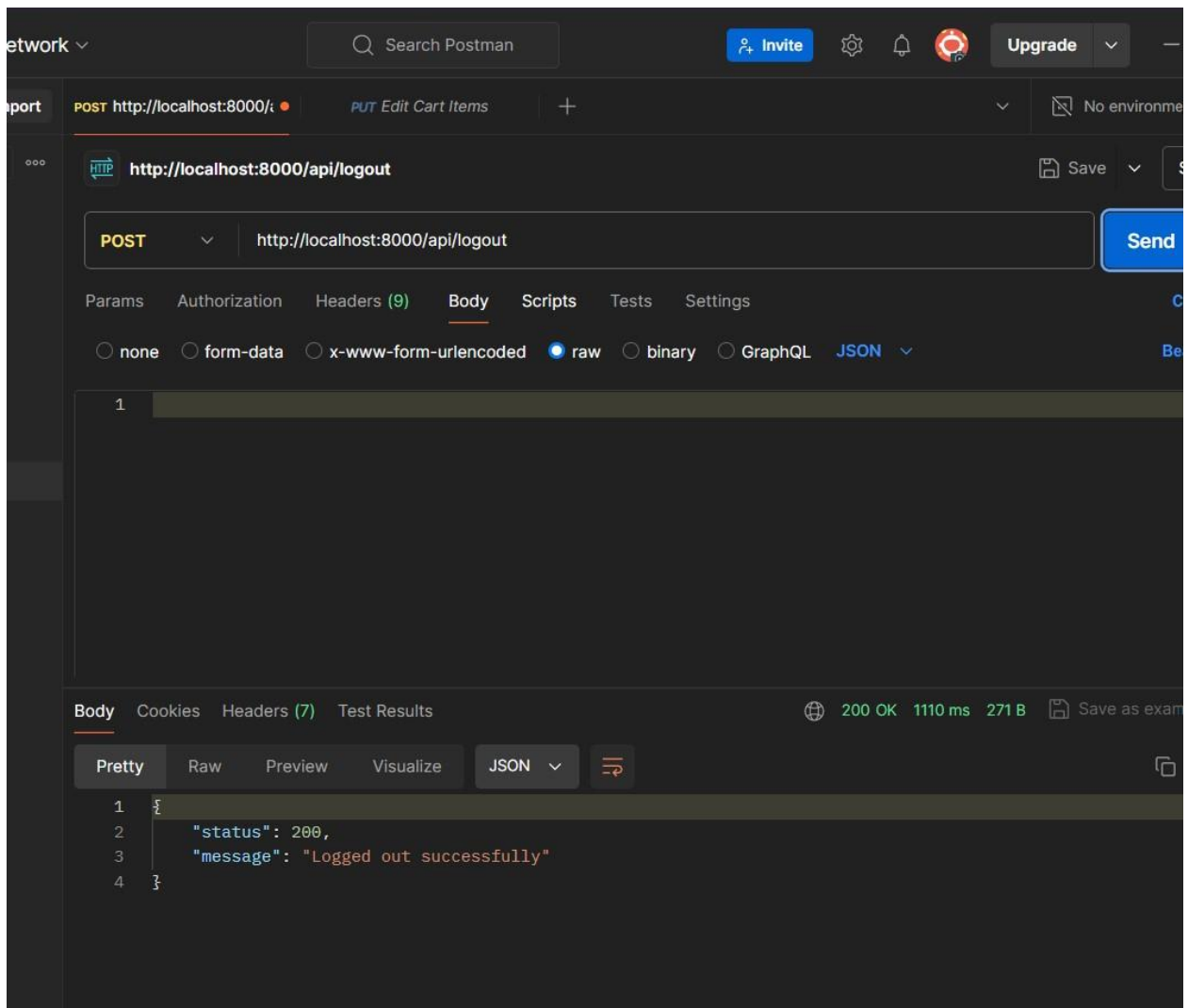
```
1 {  
2   "name": "Gamal17",  
3   "email": "GamalGerges@gmail.com",  
4   "password": "Gamal2022",  
5   "password_confirmation": "Gamal2022"  
6 }  
7
```
- Response (JSON):**

```
1 {  
2   "status": 200,  
3   "userName": "Gamal17",  
4   "token": "3|oxqrVog00V8lyxm4XtrVJx43mQJ1j3KR80DnuK6A6fca5f6d",  
5   "message": "Registered Successfully"  
6 }
```
- Status:** 200 OK, 855 ms, 353 B
- Buttons:** Save, Share, Beautify, Pretty, Raw, Preview, Visualize, JSON, Copy, Search.

2-Login End-Point:



3-Logout End Point



4-Get Products End-Point

The screenshot displays a REST client interface with a GET request to `http://localhost:8000/api/products/`. The response is a 200 OK status with a response time of 114 ms and a body size of 1.82 KB. The response body is shown in JSON format, containing an array of product data.

Request:

```
GET http://localhost:8000/api/products/
```

Response:

```
{
  "data": [
    {
      "id": 5,
      "name": "Dell Vostro 3520",
      "image": "https://m.media-amazon.com/images/I/71JPY7gSTyL._AC_SX679_.jpg",
      "shortDescription": "Dell Vostro 3520 Laptop - 12th Intel Core i5-1235U 10-Cores, 8GB RAM, 512GB SSD",
      "price": "20499.00"
    },
    {
      "id": 4,

```

5- Get Product Detail :

The screenshot displays a REST client interface with a dark theme. At the top, there are tabs for different requests: "POST register", "POST login", and "GET products". The "GET products" tab is active. Below the tabs, the URL bar shows "http://localhost:8000/api/products/1" with a "Send" button to its right. The interface has several tabs: "Overview", "Params", "Authorization", "Headers (7)", "Body", "Scripts", "Tests", "Settings", and "Cookies". The "Overview" tab is selected, showing the title "products" and a link to "View complete documentation". Below this is a text input field containing the URL "http://localhost:8000/api/products/". The "Code Snippet" section is also visible. The bottom section shows the response body, with tabs for "Body", "Cookies", "Headers (7)", and "Test Results". The "Body" tab is active, displaying a JSON response in "Pretty" format. The response status is "200 OK" with a response time of "137 ms" and a size of "699 B".

```
{
  "data": {
    "id": 1,
    "name": "2022 Apple MacBook ",
    "image": "https://m.media-amazon.com/images/I/71uJwr1IrZL._AC_SX679_.jpg",
    "shortDescription": "Air laptop with M2 chip: 13.6-inch Liquid Retina display, 8GB RAM, 256GB SSD storage",
    "detailedDescription": "2022 Apple MacBook Air laptop with M2 chip: 13.6-inch Liquid Retina display, 8GB RAM, 256GB SSD storage, 1080p FaceTime HD camera. Works with iPhone and iPad; Midnight; English",
    "price": "57950.00",
    "quantity": 50
  }
}
```

6- Add to Cart End Point

The screenshot displays a REST client interface for a POST request to `http://localhost:8000/api/cart/add`. The request body is a JSON object with `product_id` set to 1 and `quantity` set to 2. The response is a 201 status code, indicating successful creation, with a response time of 229 ms and a body size of 424 B. The response body is shown in a pretty-printed JSON format.

Request:

```
1 {
2   ... "product_id": 1,
3   ... "quantity": 2
4 }
```

Response:

```
1 {
2   "message": "Product added to cart successfully",
3   "cart": {
4     "user_id": 1,
5     "product_id": 1,
6     "product_quantity": 2,
7     "updated_at": "2024-07-06T19:59:58.000000Z",
8     "created_at": "2024-07-06T19:59:58.000000Z",
9     "id": 1
10  }
11 }
```

7-Edit Cart End-Point

The screenshot displays a REST client interface with the following details:

- Request:**
 - Method: `PUT`
 - URL: `http://localhost:8000/api/cart/edit/1`
 - Body (raw):

```
1 {
2   "quantity": 2,
3   "scope": "inc"
4 }
```
- Response:**
 - Status: `200 OK`
 - Time: `526 ms`
 - Size: `415 B`
 - Body (pretty):

```
1 {
2   "message": "Cart item updated successfully",
3   "cart": {
4     "id": 1,
5     "user_id": 1,
6     "product_id": 1,
7     "product_quantity": 3,
8     "created_at": "2024-07-06T19:59:58.000000Z",
9     "updated_at": "2024-07-06T20:25:09.000000Z"
10  }
11 }
```

8-Delete-Cart Item

The screenshot displays a REST client interface with the following details:

- Overview Tab:** Shows a list of requests including `DEL http://localhost:8000`, `PUT Edit Cart Items`, and `GET Get Cart Items`.
- Request Configuration:**
 - Method:** `DELETE`
 - URL:** `http://localhost:8000/api/cart/remove/1`
 - Body Type:** `raw` (selected)
- Response Tab:**
 - Status:** `200 OK`
 - Time:** `3.47 s`
 - Size:** `273 B`
 - Body:**

```
1 {  
2   "message": "Product removed from cart successfully"  
3 }
```


9-Place Order End Point

The image shows the Postman application interface. At the top, there's a tab bar with 'POST placeorder' selected. Below it, the URL bar shows 'http://localhost:8000/api/place-order'. The 'Headers' tab is active, displaying a list of headers with checkboxes. The 'Body' tab is also visible, showing a JSON response.

POST `http://localhost:8000/api/place-order`

Overview Params Authorization **Headers (11)** Body • Scripts Tests Settings

Header	Value
Cache-Control	no-cache
Postman-Token	<calculated when request is sent>
Content-Type	application/json
Content-Length	<calculated when request is sent>
Host	<calculated when request is sent>
User-Agent	PostmanRuntime/7.39.0
Accept	*/*
Accept-Encoding	gzip, deflate, br
Connection	keep-alive
Authorization	Bearer 2 KuYIkrRT11E76FVUcF8AummJIF...

Body Cookies (2) Headers (7) Test Results 200 OK 4.30 s 273 B Save as e

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": 200,
3   "message": "Order placed successfully"
4 }
```

10-View Cart Items Endpoint

The screenshot displays a REST client interface with the following details:

- URL:** `http://localhost:8000/api/cart/`
- Method:** `GET`
- Body:** `raw` (selected)
- Response:** `200 OK`, `2.97 s`, `1.56 KB`
- Body Content (JSON):**

```
{
  "message": "Cart retrieved successfully",
  "cart": [
    {
      "id": 1,
      "user_id": 1,
      "product_id": 1,
      "product_quantity": 2,
      "created_at": "2024-07-06T19:59:58.000000Z",
      "updated_at": "2024-07-06T19:59:58.000000Z",
      "product": {
        "id": 1,
```