

# Flipr Placement – Backend Development Task

E-commerce Backend Development

By Goyam Nahar Jain

**Email:** jaingoyamnahar@gmail.com

**Phone:** 9407529992

**Linkedin-** <https://www.linkedin.com/in/goyam-nahar-jain-019608275/>

- Create a database in PostgreSQL. ( Ecommerce )
- Restore the SQL file into the database.
- Open the command prompt in the project folder directory.
- Create Virtual Enviroment - `python -m venv venv`
- Activate the virtual environment with - `venv\Scripts\activate`.
- Install the required dependencies using - `pip install -r requirements.txt`.
- Run the Python application with - `python app.py`.
- And refer to the Screenshots folder

now on postman

## 1. Sign Up User - http://localhost:5000/signup (POST)

Body, raw, JSON:

```
{  
  "name": "Goyam Nahar Jain",  
  "email": "jaingoyamnahar@gmail.com",  
  "password": "Goyam@12345",  
  "address": "438, Clerk Colony"  
}
```

The screenshot displays a REST client interface with a single tab for the endpoint `http://localhost:5000/signup`. The request is a `POST` method. The `Body` tab is selected, showing the raw JSON payload: `{"name": "Goyam Nahar Jain", "email": "jaingoyamnahar@gmail.com", "password": "Goyam@12345", "address": "438, Clerk Colony"}`. The `Headers` tab shows 9 headers. The `Test Results` tab shows a status of `201 CREATED` with a time of `422 ms` and a size of `224 B`. The `Body` tab is also selected for the response, showing the raw JSON payload: `{"message": "Signup successful.", "user_id": 1}`.

POST http://localhost:5000/signup

http://localhost:5000/signup

POST http://localhost:5000/signup

Params Authorization Headers (9) Body Pre-request Script Tests Settings

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

```
1  
2 "name": "Goyam Nahar Jain",  
3 "email": "jaingoyamnahar@gmail.com",  
4 "password": "Goyam@12345",  
5 "address": "438, Clerk Colony"  
6
```

Body Cookies Headers (5) Test Results

Status: 201 CREATED Time: 422 ms Size: 224 B Save Response

Pretty Raw Preview Visualize JSON

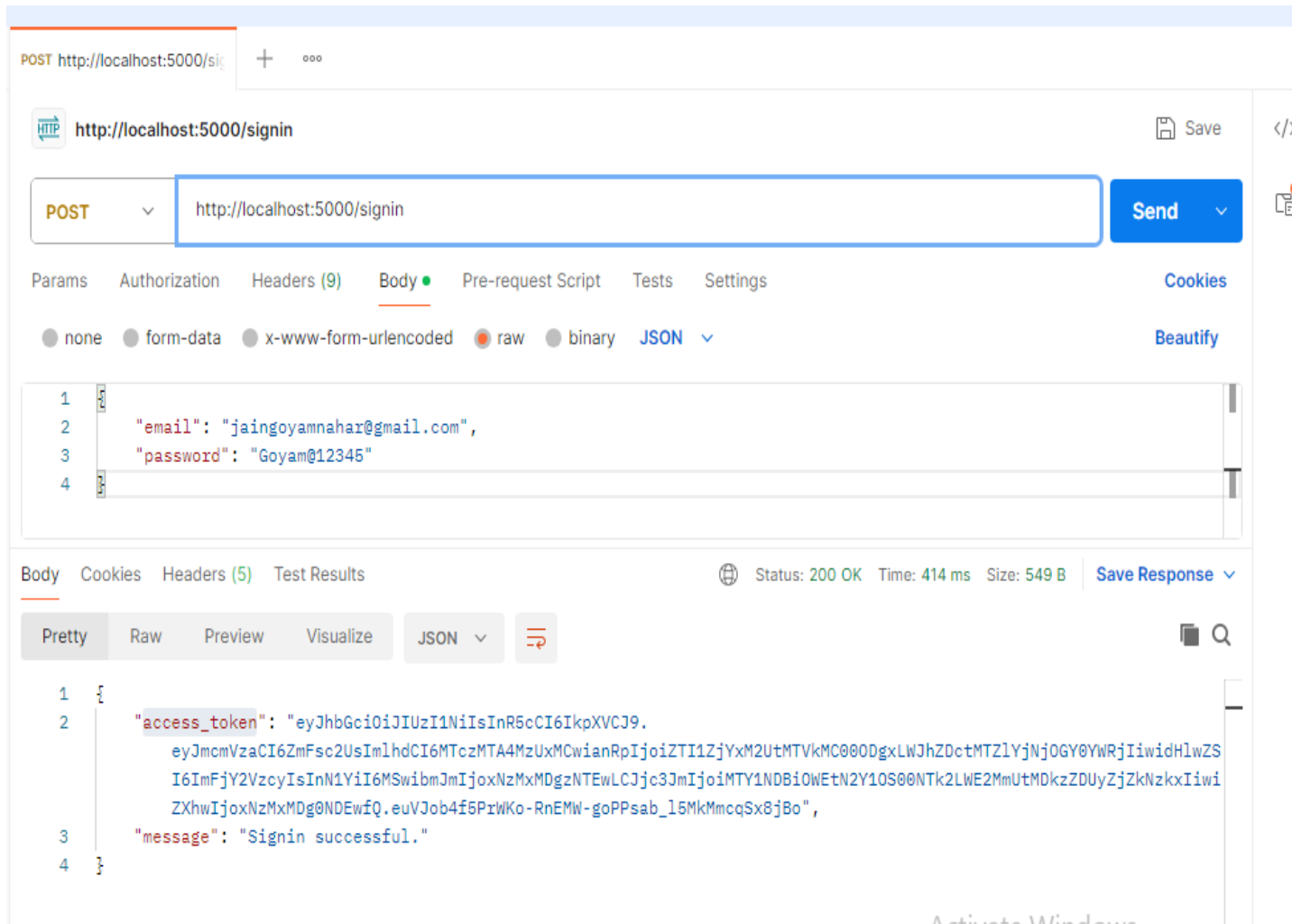
```
1  
2 "message": "Signup successful.",  
3 "user_id": 1  
4
```

## 2. Sign in user - <http://localhost:5000/signin> (POST)

Body, raw, JSON:

```
{
  "email": "jaingoyamnahar@gmail.com",
  "password": "Goyam@12345"
}
```

Copy access\_token from the output



### 3. Add Product - http://localhost:5000/addproduct (POST)

Header:

Key: Authorization

Value: Bearer (access-token)

Body, raw, JSON:

```
{  
  "name": "Mobile",  
  "description": "Best Quality",  
  "price": 15000,  
  "category": "Electronics"  
}
```

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

- URL:** http://localhost:5000/addproduct
- Method:** POST
- Body:** A JSON object: 

```
{  
  "name": "Mobile",  
  "description": "Best Quality",  
  "price": 15000,  
  "category": "Electronics"  
}
```
- Status:** 201 CREATED
- Time:** 20 ms
- Size:** 236 B
- Response Body:** A JSON object: 

```
{  
  "message": "Product added successfully.",  
  "product_id": 1  
}
```

#### 4. Update Product - http://localhost:5000/updateproduct/1 ( PUT )

Headers: Add the Authorization header with the JWT token.

```
{  
  "name": "Updated Mobile",  
  "price": 12000  
}
```

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

- Request Method:** PUT
- Request URL:** http://127.0.0.1:5000/updateproduct/1
- Request Body (JSON):**

```
1 {  
2   "name": "Updated Mobile",  
3   "price": 12000  
4 }  
5
```
- Response Status:** 200 OK
- Response Time:** 48 ms
- Response Size:** 214 B
- Response Body (JSON):**

```
1 {  
2   "message": "Product updated successfully."  
3 }
```

An "Activate Windows" watermark is visible in the bottom right corner of the image.

## 5. Delete Product - http://localhost:5000/deleteproduct/1 ( DELETE )

Headers: Add the Authorization header with the JWT token.

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

- Request Bar:** Shows the HTTP method **DELETE** and the URL **http://localhost:5000/deleteproduct/1**. A **Send** button is on the right.
- Headers Tab:** Labeled **Headers (10)**, it contains a table with headers. The **Authorization** header is selected with a checkbox and contains the value **Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6Zm...**.
- Response Section:** Labeled **Body**, it shows the response status **Status: 200 OK**, **Time: 30 ms**, and **Size: 214 B**. A **Save Response** button is present.
- Response Body:** The **Pretty** tab is active, displaying the JSON response: 

```
1 {
2   "message": "Product deleted successfully."
3 }
```

An "Activate Windows" watermark is visible in the bottom right corner of the image.

## 6. Get All Products - http://localhost:5000/products (GET)

Headers: Add the Authorization header with the JWT token.

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

- Request Bar:** Method **GET** and URL **http://localhost:5000/products**. A **Send** button is on the right.
- Tabs:** Params, Authorization, **Headers (10)**, Body, Pre-request Script, Tests, Settings, and Cookies.
- Headers Section:** Labeled "Headers" with a sub-label "8 hidden". It contains a table with 3 visible headers:

	Key	Value	Bulk Edit
<input type="checkbox"/>	image		
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmF...	
	Key	Value	

- Response Section:** Tabs include Body, Cookies, **Headers (5)**, and Test Results. Status: **200 OK**, Time: **9 ms**, Size: **337 B**. A **Save Response** button is present.
- Body Tab:** Shows the response in **JSON** format, rendered as **Pretty** text:

```
1 {
2   "products": [
3     {
4       "category": "Electronics",
5       "description": "Best Quality",
6       "id": 1,
7       "name": "Mobile",
8       "price": "15000.00"
9     }
10  ]
11 }
```

An "Activate Windows" watermark is visible in the bottom right corner of the response area.

## 7. Add to Cart - http://localhost:5000/cart/add (POST)

Headers: Add the Authorization header with the JWT token.

Body, raw, JSON:

```
{  
  "product_id": 1,  
  "quantity": 2  
}
```

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

- Request:**
  - Method: GET (dropdown)
  - URL: http://localhost:5000/cart
  - Buttons: Send, dropdown
- Request Headers:**
  - Params, Authorization, Headers (10), Body (selected), Pre-request Script, Tests, Settings, Cookies
  - Headers tab: 8 hidden
  - Table:
- Response:**
  - Body, Cookies, Headers (5), Test Results
  - Status: 200 OK, Time: 10 ms, Size: 371 B, Save Response
  - Body tab: Pretty, Raw, Preview, Visualize, JSON (selected)
  - JSON content:

	Key	Value	Bulk Edit
<input type="checkbox"/>	image		
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmF...	
	Key	Value	

```
1  
2  "cart": [  
3    {  
4      "description": "Best Quality",  
5      "price": "15000.00",  
6      "product_name": "Mobile",  
7      "quantity": 4,  
8      "total": "60000.00"  
9    }  
10 ]
```



## 8. Update Cart - http://localhost:5000/cart/update ( PUT )

Headers: Add the Authorization header with the JWT token.

Body, raw, JSON:

```
{  
  "product_id": 1,  
  "quantity": 3  
}
```

The screenshot shows a REST client interface with a PUT request to `http://localhost:5000/cart/update`. The request body is a JSON object: `{ "product_id": 2, "quantity": 3 }`. The response status is 200 OK, and the response body is a JSON object: `{ "message": "Cart updated successfully." }`.

PUT `http://localhost:5000/cart/update` Send

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies

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

```
1 {  
2   "product_id": 2,  
3   "quantity": 3  
4 }  
5
```

Body Cookies Headers (5) Test Results Status: 200 OK Time: 11 ms Size: 211 B Save Response

Pretty Raw Preview Visualize **JSON**

```
1 {  
2   "message": "Cart updated successfully."  
3 }
```

Activate Windows  
Go to Settings to activate Windows.

## 9. Delete Product from Cart - `http://localhost:5000/cart/delete` ( DELETE )

Headers: Add the Authorization header with the JWT token.

Body, raw, JSON:

```
{  
  "product_id": 2  
}
```

The screenshot shows a REST client interface with a DELETE request to `http://localhost:5000/cart/delete`. The request body is a JSON object: `{ "product_id": 2 }`. The response status is 200 OK, and the response body is a JSON object: `{ "message": "Product removed from cart." }`.

**Request:**

- Method: DELETE
- URL: `http://localhost:5000/cart/delete`
- Body: `{ "product_id": 2 }`

**Response:**

- Status: 200 OK
- Time: 13 ms
- Size: 211 B
- Body: `{ "message": "Product removed from cart." }`

Activate Windows  
Go to Settings to activate Windows.

## 10. Get Cart - http://localhost:5000/cart (GET)

Headers: Add the Authorization header with the JWT token.

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

- Request Bar:** Method `GET` and URL `http://localhost:5000/cart`. A `Send` button is on the right.
- Headers Tab:** Labeled "Headers (10)". It contains a table with one header entry:

Key	Value
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6Zm...
- Response Bar:** Status `200 OK`, Time `43 ms`, Size `381 B`. A `Save Response` button is on the right.
- Body Tab:** Labeled "Body". It shows the response in JSON format:

```
1  {
2    "cart": [
3      {
4        "description": "Best Performance",
5        "price": "15000000.00",
6        "product_name": "Audi",
7        "quantity": 2,
8        "total": "30000000.00"
9      }
10   ],
11   "total_amount": "30000000.00"
12 }
```

## 11. Place Order - http://localhost:5000/placeorder (POST)

Headers: Add the Authorization header with the JWT token.

Body, raw, JSON:

```
{  
  "shipping_address": "438, Clerk, Colony, Indore ( 9407529992 )"  
}
```

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

- Request Bar:** Method **POST**, URL `http://localhost:5000/placeorder`, and a **Send** button.
- Request Body:** The **Body** tab is selected. The format is set to **JSON**. The body content is:

```
{  
  "shipping_address": "438, Clerk, Colony, Indore ( 9407529992 )"  
}
```
- Response Section:** The **Body** tab is selected. The status is **200 OK**, time is **39 ms**, and size is **228 B**. The response body is:

```
{  
  "message": "Order placed successfully.",  
  "order_id": 1  
}
```

An "Activate Windows" watermark is visible in the bottom right corner of the image.

## 12. Get All Orders - http://localhost:5000/getallorders ( GET )

Headers: Add the Authorization header with the JWT token.

The screenshot shows a REST client interface with the following components:

- Request Bar:** Method: GET, URL: http://localhost:5000/getallorders, Send button.
- Headers Tab:** Headers (10), 8 hidden. Table with 4 columns: Key, Value, Bulk Edit.
- Headers Table:**

	Key	Value	Bulk Edit
<input type="checkbox"/>	image		
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6Zm...	
- Response Bar:** Body, Cookies, Headers (5), Test Results. Status: 200 OK, Time: 14 ms, Size: 447 B, Save Response button.
- Response Body:** Pretty, Raw, Preview, Visualize, JSON dropdown, Copy icon. JSON response:

```
1 {
2   "orders": [
3     {
4       "order_id": 1,
5       "products": [
6         {
7           "product_name": "Mobile",
8           "quantity": 4
9         }
10      ],
11      "shipping_address": "438, Clerk, Colony, Indore ( 9407529992 )",
12    }
13  ]
14 }
```

### 13. Get Orders by Customer ID - `http://localhost:5000/orders/customer/{customerId}` ( GET )

Headers: Add the Authorization header with the JWT token.

The screenshot shows a REST client interface with the following components:

- Request Bar:** Method: GET, URL: `http://localhost:5000/orders/customer/1`, Send button.
- Headers Tab:** Contains a table with headers.
- Response Bar:** Status: 200 OK, Time: 19 ms, Size: 425 B, Save Response button.
- Body Tab:** Shows the response body in JSON format.

Key	Value	Bulk Edit
<input type="checkbox"/> image		
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6Zm...	

```
1  {
2    "orders": [
3      {
4        "order_id": 1,
5        "products": [
6          {
7            "product_name": "Audi",
8            "quantity": 4
9          }
10       ],
11       "shipping_address": "438, Clerk, Colony, Indore ( 9407529992 )",
12       "status": "Pending"
13     }
14   ]
15 }
```

Activate Windows  
Go to Settings to activate Windows

## DataBase Screenshot

The screenshot displays the pgAdmin 4 interface. The left sidebar shows the 'Object Explorer' with a tree view of the database structure. The 'public' schema is selected, and the 'users' table is highlighted. The main pane shows a SQL query: `SELECT * FROM public.users ORDER BY id ASC`. Below the query, the 'Data Output' tab displays the results of the query in a table format.

id	name	email	password	address
1	Goyam Nahar Jain	jaingoyamnahar@gmail.com	\$2b\$12\$5trFb9wF3T2/YJWfTpk/MOnI3A6LS0sB4Ppr0xgbw0RU0diuwm...	438, Clerk Colony
2	Goyam Nahar Jain 2	goyamnaharjain@gmail.com	\$2b\$12\$.LbNnH.8GoXhCKdgOZ../fyez1PPYXY9Cr7W19HuAfnHef839VC	277, Clerk Colony
3	Goyam Nahar Jain 3	goyamnaharjain123@gmail.com	\$2b\$12\$DuVU6g3fhDp.35ph2JQ1Cea/.mSqHHE0q5HYORirW2DaD7eyWN...	278, Clerk Colony
4	Goyam Nahar Jain 4	goyamnahar123@gmail.com	\$2b\$12\$unihQNWCSMxjPefYiqH5d.3gtknson6rOeu.Hwu3J0zok05ZRtDAa	279, Clerk Colony
5	Goyam Nahar Jain 5	goyam.work1@gmail.com	\$2b\$12\$Q/zaekrWbtfaoyGNwk3OI0N2svf3PH.TE/UYIGw651RNFUfWAJ...	280, Clerk Colony
6	Goyam Nahar Jain 6	goyam.work@gmail.com	\$2b\$12\$T1BjwztaWXXI4N9bYHwiy.oG6AZPSYivB6NGxsKacRMlb9yettB...	281, Clerk Colony

And if you face any error or if any API is not working, just contact me. I can solve the error

Goyam Nahar Jain

Email: [jaingoyamnahar@gmail.com](mailto:jaingoyamnahar@gmail.com)

Phone: 9407529992

LinkedIn: <https://www.linkedin.com/in/goyam-nahar-jain-019608275/>

# THANKS