**Storefront Backend**

**Requirements:**create APIs and database for small storefront.

**Database:**

Create 4 tables:

1. **users**

* id => serial primary key
* name => varchar(100)
* email => varchar
* hash\_password => varcher

1. **products**

* id => serial primary key
* name => varchar(100)
* description => varchar
* price => float
* category => varchar(50)

1. **orders**

* id => serial primary key
* status => varchar(50) (active or complete)
* user\_id => bigint foreign key

1. **order\_products**

* id => serial primary key
* order\_id => bigint foreign key
* product\_id => bigint foreign key
* quantity => integer
* price => float (total price of product (including quantity))

**Database diagram:**

Graphical user interface, application, table, Excel

Description automatically generated

**API Endpoints:**

* **users**:

<http://localhost:3000/users> => post method to create user (email should be unique)

Graphical user interface

Description automatically generated

<http://localhost:3000/users/login> => post method to login with email and password then return user token

A screenshot of a computer

Description automatically generated

<http://localhost:3000/users> => get method that returned all user and request need bearer token (send the token that returned from login method)

Graphical user interface

Description automatically generated

<http://localhost:3000/users/:id> => get method to get user information by id (the request needs to pass the bearer token for authentication).

Graphical user interface

Description automatically generated

<http://localhost:3000/users> => put method to update user information (you should send bearer token and it must be the token for the user that need to update his information)

Graphical user interface

Description automatically generated

<http://localhost:3000/users> => delete method that delete user by id (you should send bearer token and it must be the token for the user that need to delete him self)

**Graphical user interface

Description automatically generated**

* **products:**

<http://localhost:3000/products> => post method to create product (the request needs to pass the bearer token for authentication).

Graphical user interface

Description automatically generated

<http://localhost:3000/products> => put method to update product information

Graphical user interface

Description automatically generated

<http://loacalhost:3000/products> => get method to get all products

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

<http://localhost:3000/products/:id> => get method to get product by id

Graphical user interface

Description automatically generated

<http://localhost:3000/products/:id> => delete method to delete product by id

Graphical user interface

Description automatically generated

* **orders**:

<http://localhost:3000/orders> => post method to create order (the request needs to pass the bearer token for authentication)

A screenshot of a computer

Description automatically generated

<http://localhost:3000/orders> => put method to update order (the request needs to pass the bearer token for authentication)

A screenshot of a computer

Description automatically generated with medium confidence

<http://localhost:3000/orders> => get method to get all orders (the request needs to pass the bearer token for authentication)

A screenshot of a computer

Description automatically generated with medium confidence

<http://localhost:3000/orders/:id> => get method to get order by id (the request needs to pass the bearer token for authentication)

Graphical user interface

Description automatically generated

<http://localhost:3000/ordersUser/:user_id> => get method to get orders by user id (the request needs to pass the bearer token for authentication).

Graphical user interface

Description automatically generated

<http://localhost:3000/orders/:id> => delete method to delete order by id (the request needs to pass the bearer token for authentication)

Graphical user interface

Description automatically generated

* **order\_products:**

<http://loacalhost:3000/orderProducts> => post method to create order\_product

A screenshot of a computer

Description automatically generated with medium confidence

<http://localhost:3000/orderProducts> => put method to update order\_product

A screenshot of a computer

Description automatically generated with medium confidence

<http://localhost:3000/orderProducts> => get method to get all order\_products

A screenshot of a computer

Description automatically generated

Graphical user interface

Description automatically generated

<http://localhost:3000/orderProducts/:id> => get method to get order\_product by id

Graphical user interface

Description automatically generated

<http://localhost:3000/orderProducts/:id> => delete method to delete order\_product by id

Graphical user interface

Description automatically generated

* **Dashboard:**

<http://localhost:3000/usersOrders> => get method to get orders with user id and user name (the request needs to pass the bearer token for authentication)

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

<http://localhost:3000/topOrderedProducts> => get method to get list of products in descending order according to the number of times they are found in orders (the request needs to pass the bearer token for authentication)

A screenshot of a computer

Description automatically generated with medium confidence