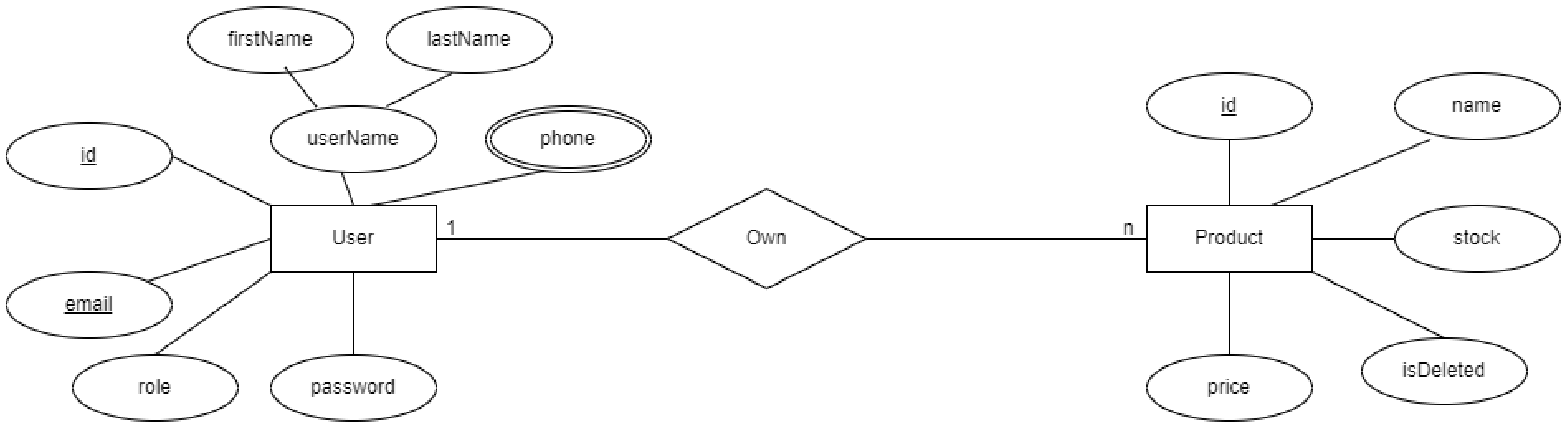
Assignment 7

Part1: Design a schema for the following ERD. (Use any design tool you want) (1 Grade)



Part2: Create APIs using Express.js with MySQL (Ensure you apply the folder structure we discussed)

A- Generate Tables API

1. **Using the schema you designed in the previous question, create the necessary tables for users and products, defining the appropriate columns, data types, and relationships. (1 Grade)**
   - **URL**: POST /DB/create-tables

```
{
  "message": "Tables created successfully."
}
```

B- User APIs

1. **Insert a new user and make sure that the email does not exist before (Get the user data from the body). (1 Grade)**
   - **URL:** POST /user/signup

```
{
  "message": "Email already exists."
}
```

```
{
  "message": "User added successfully."
}
```

2. **Login user using email and password that will be provided in the request body. (0.5 Grade)**
   - **URL:** POST /user/login

```
{
  "message": "done."
}
```

```
{
  "message": "Invalid credentials."
}
```

3. **Alter the user table to add a createdAt column (timestamp). The user ID will be provided in the request body. (Make sure the user has the 'admin' role before allowing this change). (1 Grade)**
   - **URL:** POST /user/alter-table

```
{
  "message": "done."
}
```

```
{
  "message": "You don't have access."
}
```

4. **Truncate the products table to remove all records. The user ID will be provided in the request body. (Make sure the user has the 'admin' role before performing this operation). (1 Grade)**
   - **URL:** POST /user/truncate-table

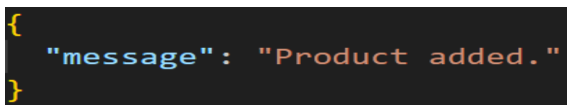
```
{
  "message": "done."
}
```

```
{
  "message": "You don't have access."
}
```

Assignment 7

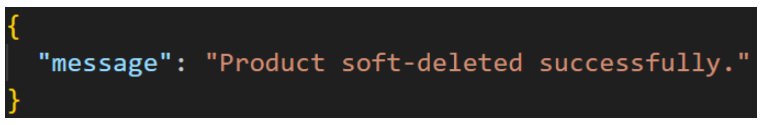
C- Product APIs

1. **Insert a new product (Get the product data from the body). (0.5 Grade)**
   - **URL:** POST /products

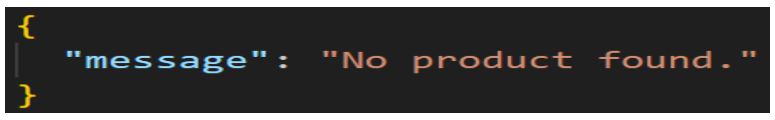


```
{
  "message": "Product added."
}
```

2. **Mark a product as deleted (soft delete) without actually removing it from the database. (0.5 Grade)**
   - **URL:** PATCH /products/soft-delete/:id

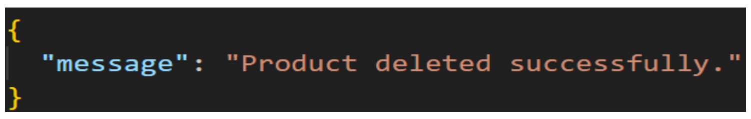


```
{
  "message": "Product soft-deleted successfully."
}
```

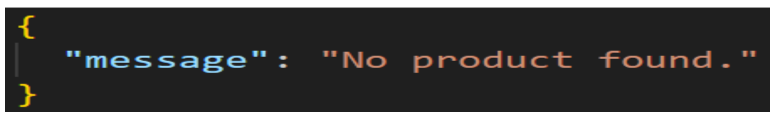


```
{
  "message": "No product found."
}
```

3. **Delete a product by its id (0.5 Grade)**
   - **URL:** DELETE /products/:id

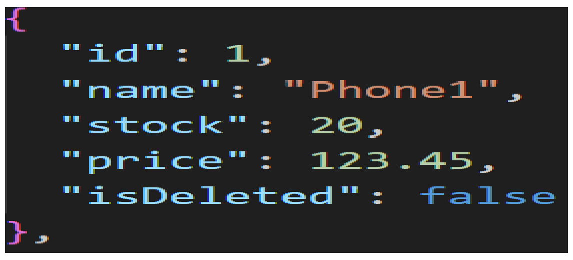


```
{
  "message": "Product deleted successfully."
}
```

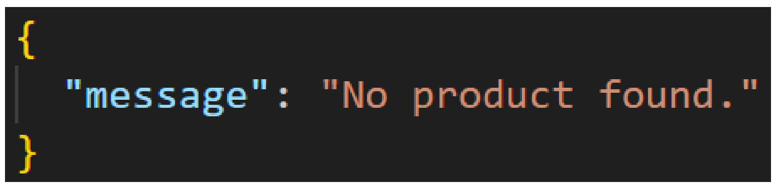


```
{
  "message": "No product found."
}
```

4. **Search for products by name (use like) (0.5 Grade)**
   - **URL:** GET /products/search (for example => /products/search?name=phone)

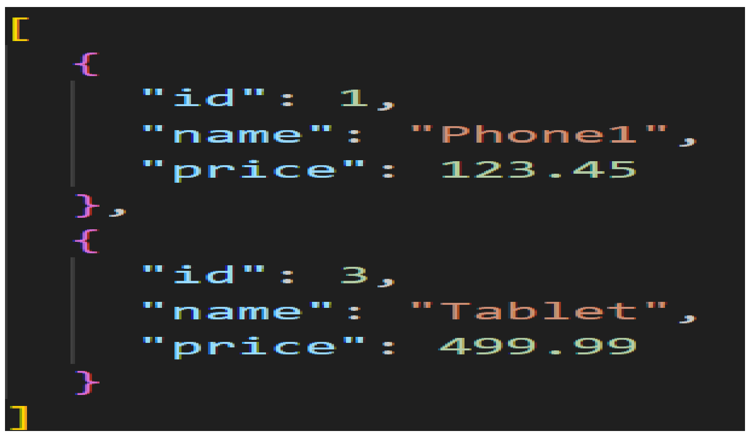


```
{
    "id": 1,
    "name": "Phone1",
    "stock": 20,
    "price": 123.45,
    "isDeleted": false
},
```



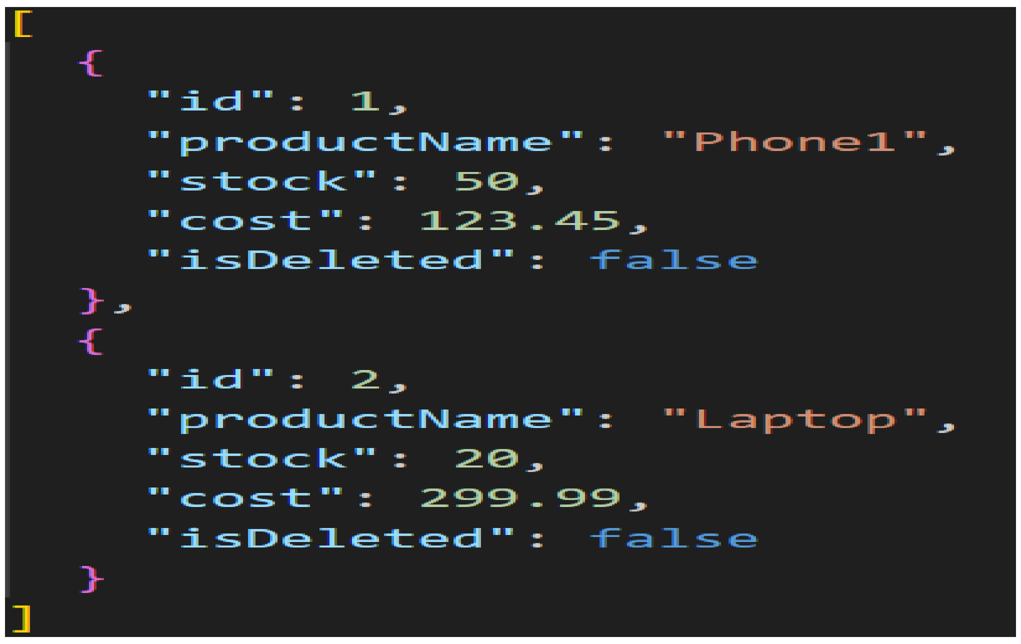
```
{
  "message": "No product found."
}
```

5. **Select products with id values that match a set of IDs and return only id, name and price. (0.5 Grade)**
   - **URL:** GET /products/in (for example => /products/in?ids=1,3)



```
[
  {
    "id": 1,
    "name": "Phone1",
    "price": 123.45
  },
  {
    "id": 3,
    "name": "Tablet",
    "price": 499.99
  }
]
```

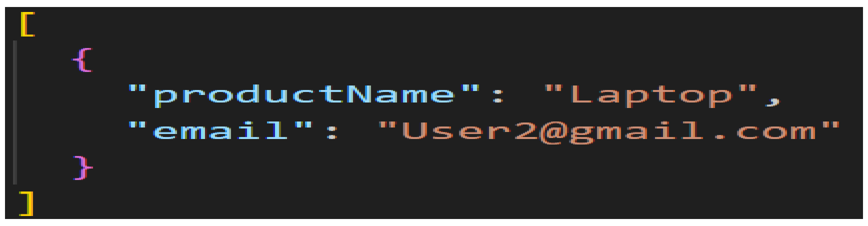
6. **Get all products that are not soft-deleted but alias the “name” as “productName” and the “price” as “cost” (0.5 Grade)**
   - **URL:** GET /products



```
[
  {
    "id": 1,
    "productName": "Phone1",
    "stock": 50,
    "cost": 123.45,
    "isDeleted": false
  },
  {
    "id": 2,
    "productName": "Laptop",
    "stock": 20,
    "cost": 299.99,
    "isDeleted": false
  }
]
```

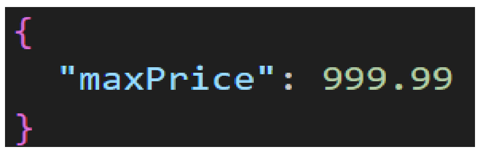
Assignment 7

7. **Get all products with their corresponding user (owner). (Just return the name of product as "productName" and the user email for each product) (0.5 Grade)**
    - **URL:** GET /products/products-with-users



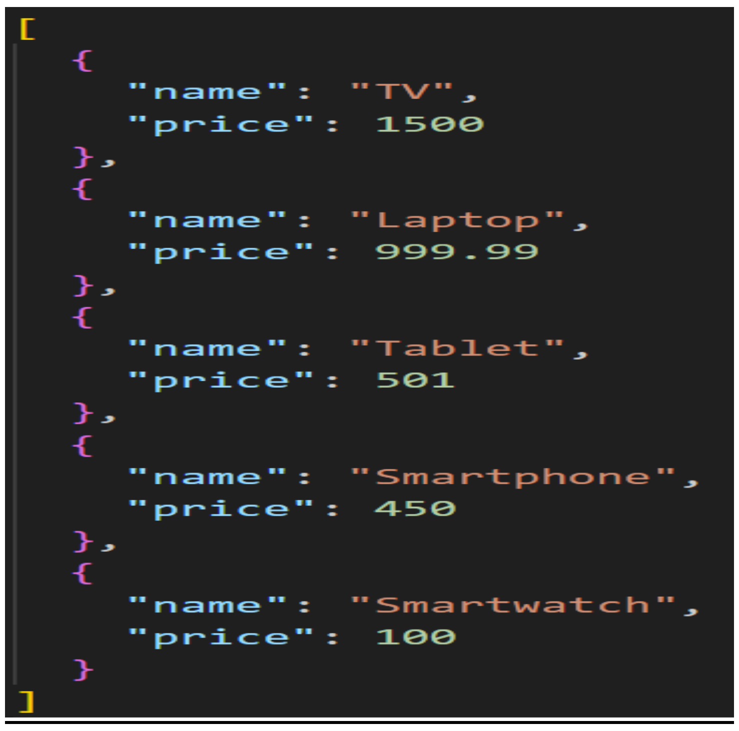
```
[
  {
    "productName": "Laptop",
    "email": "User2@gmail.com"
  }
]
```

8. **Get the maximum price of all products. (0.5 Grade)**
    - **URL:** GET /products/max-price



```
{
  "maxPrice": 999.99
}
```

9. **Get Top 5 Most Expensive Products (return only the name and price) (0.5 Grade)**
    - **URL:** GET /products/top-expensive



```
[
  {
    "name": "TV",
    "price": 1500
  },
  {
    "name": "Laptop",
    "price": 999.99
  },
  {
    "name": "Tablet",
    "price": 501
  },
  {
    "name": "Smartphone",
    "price": 450
  },
  {
    "name": "Smartwatch",
    "price": 100
  }
]
```

⚠️ Important Notes about postman

1. **Name the endpoint with a meaningful name like 'Add User', not dummy names.**
2. **Save your changes on each request(ctrl+s).**
3. **Include the Postman collection link (export your Postman collection) in the email with your assignment link**

Bonus (2 Grades)

How to deliver the bonus?

1- Solve the problem Average Selling Price on **LeetCode**

2- Inside your assignment folder, create a **SEPARATE FILE** and name it "bonus.txt"

3- Copy the code that you have submitted on the website inside "bonus.txt" file