API DOCUMENTATION

FOR PERFORMING REST API REQUESTS ON LAPTOP STORE DATABASE V1.2

Prepared by

Tadeo Bennett

University of Belize

Prepared for

Mr. Lester Pech

University of Belize

CMPS4191: Advanced Web-Technologies

October 20, 2023

Table of Contents

Introduction 1
Getting Started 1
Endpoints 2
- GET Get all users 2
- GET Get all laptops 3
- GET Get users by ID 4
- GET Get laptops by ID 5
- GET Get all employees 6
- GET Get all customers 7
- POST Add new user 8
- PUT Update user by ID 9
- PUT Update laptop by ID 10
- DELETE Delete user by ID 11
- DELETE Delete laptop by ID 12
- Sorting and Pagination 13

Response Format 15
- API Structure 15
- HTTP Response Codes Usage 15
- Response Codes and Their Meanings 16
  - Common Errors 16
  - GET Requests Errors 16
  - POST Request Errors 16
  - PUT Request Errors 16
  - DELETE Request Errors 17
  - Success Response Codes 17

Support 17
Changelog 18
Appendix 18

Introduction

This API enables developers to perform operations such as GET, POST, PUT, and DELETE on the users of the laptop store database. Its primary purpose is to enable developers to understand the requests' makeup and perform operations that yield expected results from the database. In addition, any error during the process is communicated to the developer to allow easy troubleshooting. This API is designed for any developer with knowledge of REST APIs, PHP, and MySQL.

These request examples can also be found in the online documentation.

Getting Started

If permitted, you can clone the repo from GitHub. In the extras directory, the file laptopstore_sql_dump must be used to create the laptopstore database. First, you'll need to create the "laptopstore" database and then run the file's contents in that database. Include the necessary login credentials for your mysql database in the index.php file. If Apache is used, I recommend having the root directory set up as the parent directory of the repo. It should look like the following:

root/CMPS4191_adv_web_RESTAPI/…

If required, ensure that your port is provided in the URI. It should look like the following:

127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/…

Endpoints

1. GET Get all users

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users

Example Request:

curl --location 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json''

Example Response:


```
{
  "rc": 50,
  "message": "Success",
  "data": [
    {
      "user_id": 1,
      "role_id": 1,
      "firstname": "Tadeo",
      "lastname": "Bennett",
      "username": "TBennett",
      "email": "tadeo@gmail.com",
      "address": 4,
      "phone": "",
      "age": 21,
      "password": "Tadeo2002",
      "member": 0,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    },
    {
      "user_id": 2,
      "role_id": 2,
      "firstname": "William",
      "lastname": "Locario",
      "username": "WLocario",
      "email": "william@gmail.com",
      "address": 6,
      "phone": "",
      "age": 22,
      "password": "William2002",
      "member": 1,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    },
    {
      "user_id": 2,
      "role_id": 2,
      "firstname": "William",
      "lastname": "Locario",
      "username": "WLocario",
      "email": "william@gmail.com",
      "address": 6,
      "phone": "",
      "age": 22,
      "password": "William2002",
      "member": 1,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    },
    {
      "user_id": 3,
      "role_id": 2,
      "firstname": "Victor",
      "lastname": "Castillo",
      "username": "VCastillo",
      "email": "victor@gmail.com",
      "address": 2,
      "phone": "",
      "age": 19,
      "password": "Victor2002",
      "member": 0,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    }
  ]
}
```

2. GET Get all laptops

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/laptops

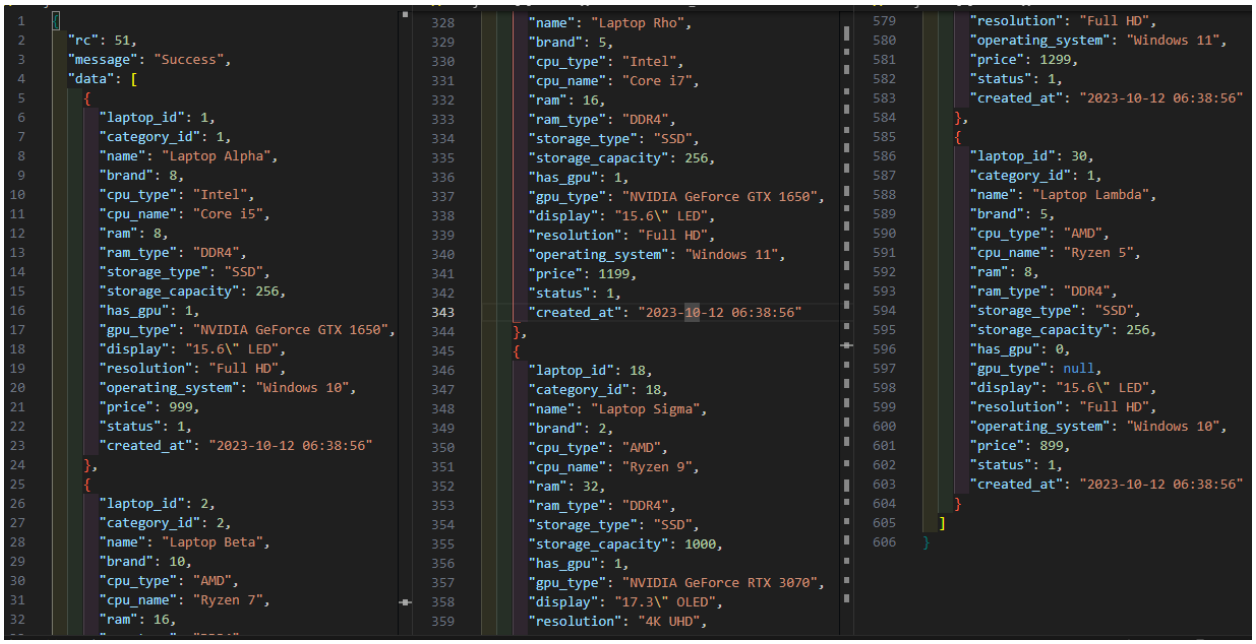
Example Request:

curl --location 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/laptops' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json''

Example Response:



```
{
  "rc": 51,
  "message": "Success",
  "data": [
    {
      "laptop_id": 1,
      "category_id": 1,
      "name": "Laptop Alpha",
      "brand": 8,
      "cpu_type": "Intel",
      "cpu_name": "Core i5",
      "ram": 8,
      "ram_type": "DDR4",
      "storage_type": "SSD",
      "storage_capacity": 256,
      "has_gpu": 1,
      "gpu_type": "NVIDIA GeForce GTX 1650",
      "display": "15.6\" LED",
      "resolution": "Full HD",
      "operating_system": "Windows 10",
      "price": 999,
      "status": 1,
      "created_at": "2023-10-12 06:38:56"
    },
    {
      "laptop_id": 2,
      "category_id": 2,
      "name": "Laptop Beta",
      "brand": 10,
      "cpu_type": "AMD",
      "cpu_name": "Ryzen 7",
      "ram": 16,
      "name": "Laptop Rho",
      "brand": 5,
      "cpu_type": "Intel",
      "cpu_name": "Core i7",
      "ram": 16,
      "ram_type": "DDR4",
      "storage_type": "SSD",
      "storage_capacity": 256,
      "has_gpu": 1,
      "gpu_type": "NVIDIA GeForce GTX 1650",
      "display": "15.6\" LED",
      "resolution": "Full HD",
      "operating_system": "Windows 11",
      "price": 1199,
      "status": 1,
      "created_at": "2023-10-12 06:38:56"
    },
    {
      "laptop_id": 18,
      "category_id": 18,
      "name": "Laptop Sigma",
      "brand": 2,
      "cpu_type": "AMD",
      "cpu_name": "Ryzen 9",
      "ram": 32,
      "ram_type": "DDR4",
      "storage_type": "SSD",
      "storage_capacity": 1000,
      "has_gpu": 1,
      "gpu_type": "NVIDIA GeForce RTX 3070",
      "display": "17.3\" OLED",
      "resolution": "4K UHD",
      "resolution": "Full HD",
      "operating_system": "Windows 11",
      "price": 1299,
      "status": 1,
      "created_at": "2023-10-12 06:38:56"
    },
    {
      "laptop_id": 30,
      "category_id": 1,
      "name": "Laptop Lambda",
      "brand": 5,
      "cpu_type": "AMD",
      "cpu_name": "Ryzen 5",
      "ram": 8,
      "ram_type": "DDR4",
      "storage_type": "SSD",
      "storage_capacity": 256,
      "has_gpu": 0,
      "gpu_type": null,
      "display": "15.6\" LED",
      "resolution": "Full HD",
      "operating_system": "Windows 10",
      "price": 899,
      "status": 1,
      "created_at": "2023-10-12 06:38:56"
    }
  ]
}
```

3. GET Get users by ID

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/3

Example Request:

curl --location 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/3' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json''

Example Response:


```
{
  "rc": 52,
  "message": "Success",
  "data": [
    {
      "user_id": 3,
      "role_id": 2,
      "firstname": "Victor",
      "lastname": "Castillo",
      "username": "VCastillo",
      "email": "victor@gmail.com",
      "address": 2,
      "phone": "",
      "age": 19,
      "password": "Victor2002",
      "member": 0,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    }
  ]
}
```

4. GET Get laptops by ID

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/laptops/3

Example Request:

curl --location 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/laptops/3' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json''

Example Response:


```
{
"rc": 53,
"message": "Success",
"data": [
{
"laptop_id": 3,
"category_id": 3,
"name": "Laptop Gamma",
"brand": 3,
"cpu_type": "Intel",
"cpu_name": "Core i7",
"ram": 16,
"ram_type": "DDR4",
"storage_type": "SSD",
"storage_capacity": 512,
"has_gpu": 1,
"gpu_type": "NVIDIA GeForce RTX 3060",
"display": "15.6\" IPS",
"resolution": "Full HD",
"operating_system": "Windows 10",
"price": 1299,
"status": 1,
"created_at": "2023-10-12 06:38:56"
}
]
}
```

5. GET Get all employees

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/employees

Example Request:

curl --location 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/employees' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json''

Example Response:


```
{
"rc": 54,
"message": "Success",
"data": [
{
"user_id": 1,
"role_id": 1,
"firstname": "Tadeo",
"lastname": "Bennett",
"username": "TBennett",
"email": "tadeo@gmail.com",
"address": 4,
"phone": "",
"age": 21,
"password": "Tadeo2002",
"member": 0,
"status": 1,
"created_at": "2023-10-12 06:38:55"
}
]
}
```

6. GET Get all customers

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/customers

Example Request:

curl --location 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/customers' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json''

Example Response:


```
{
  "rc": 55,
  "message": "Success",
  "data": [
    {
      "user_id": 2,
      "role_id": 2,
      "firstname": "William",
      "lastname": "Locario",
      "username": "WLocario",
      "email": "william@gmail.com",
      "address": 6,
      "phone": "",
      "age": 22,
      "password": "William2002",
      "member": 1,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    },
    {
      "user_id": 3,
      "role_id": 2,
      "firstname": "Victor",
      "lastname": "Castillo",
      "username": "VCastillo",
      "email": "victor@gmail.com",
      "address": 2,
      "phone": "",
      "age": 19,
      "password": "Victor2002",
      "member": 0,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    }
  ]
}
```

7. POST Add new user

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/

Example Request:

curl --location 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json'' \

--form 'firstname="John"'' \

--form 'lastname="Doe"'' \

--form 'email="john@gmail.com"'' \

--form 'address="3"'' \

--form 'username="JDoe"'' \

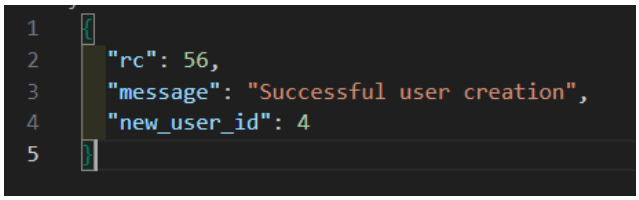
--form 'status="1"'' \

--form 'role_id="1"''

Other optional fields(data type specified):

password(string), member(int)

Example Response:



8. PUT Update user by ID

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/2

Example Request:

curl --location --request PUT 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/2' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json’'\

--form 'firstname="Luke"' \

--form 'lastname="SKyWalker"' \

--form 'username="LSkyWalker"' \

--form 'email="luke@gmail.com"'

Other optional fields(data type specified):

role_id(int), address(int), password(string), member(int), status(int)

Example Response:


```
{
  "rc": 57,
  "message": "Success. User Updated",
  "data": [
    {
      "user_id": 2,
      "role_id": 2,
      "firstname": "William",
      "lastname": "Locario",
      "username": "WLocario",
      "email": "william@gmail.com",
      "address": 6,
      "phone": "",
      "age": 22,
      "password": "William2002",
      "member": 1,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    }
  ],
  "newdata": [
    {
      "user_id": 2,
      "role_id": 2,
      "firstname": "Luke",
      "lastname": "SKyWalker",
      "username": "LSkyWalker",
      "email": "luke@gmail.com",
      "address": 6,
      "phone": "",
      "age": 22,
      "password": "William2002",
      "member": 1,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    }
  ]
}
```

9. PUT Update laptop by ID

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/laptops/4

Example Request:

curl --location --request PUT 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/laptops/4' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json’'\

--form 'name="MyNewLaptop 23"' \

--form 'ram="64"'

Other optional fields(data type specified):

laptop_id (int), category_id (int), name (string), brand (int), cpu_type (string), cpu_name (string), ram (int), ram_type (string), storage_type (string), storage_capacity (int), has_gpu (bool), gpu_type (string), display (string), resolution (string), operating_system (string), price (int), status (bool), created_at (timestamp)

Example Response:

10. DELETE Delete user by ID

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/3

Example Request:

curl --location --request DELETE 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/users/3' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json''

Example Response:


```
{
  "rc": 59,
  "message": "Success. User deleted with id 3",
  "data": [
    {
      "user_id": 3,
      "role_id": 2,
      "firstname": "Victor",
      "lastname": "Castillo",
      "username": "VCastillo",
      "email": "victor@gmail.com",
      "address": 2,
      "phone": "",
      "age": 19,
      "password": "Victor2002",
      "member": 0,
      "status": 1,
      "created_at": "2023-10-12 06:38:55"
    }
  ]
}
```

11. DELETE Delete laptop by ID

http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/laptops/25

Example Request:
curl --location --request DELETE 'http://127.0.0.1:8036/CMPS4191_adv_web_RESTAPI/laptops/25' \

--header 'API-KEY: awt_Klw8!LhJ!2e,+?R%;#NZ_2967926746'

--header 'Content-Type: application/json''

Example Response:


```
{
  "rc": 60,
  "message": "Success. Laptop deleted with id 25",
  "data": [
    {
      "laptop_id": 25,
      "category_id": 5,
      "name": "Laptop Zeta",
      "brand": 10,
      "cpu_type": "Intel",
      "cpu_name": "Core i5",
      "ram": 8,
      "ram_type": "DDR4",
      "storage_type": "SSD",
      "storage_capacity": 256,
      "has_gpu": 1,
      "gpu_type": "NVIDIA GeForce GTX 1650",
      "display": "15.6\" LED",
      "resolution": "Full HD",
      "operating_system": "Windows 11",
      "price": 1199,
      "status": 1,
      "created_at": "2023-10-12 06:38:56"
    }
  ]
}
```

Sorting and Pagination

http://127.0.0.1:8036/demo3/laptops

Example Request:

curl --location --request GET 'http://127.0.0.1:8036/demo3/laptops' \

--header 'API-KEY:
awt_[9H<TzE5pIhW08tS(yF=Qo?{_0227029b5e7013d468d8155a47f1ec2b38f9f129aaadb9a668dd956dae443540' \

--data '{

"order": {"brand":"ASC", "ram": "DESC"},

"paging": {"start": 0, "end": 3}

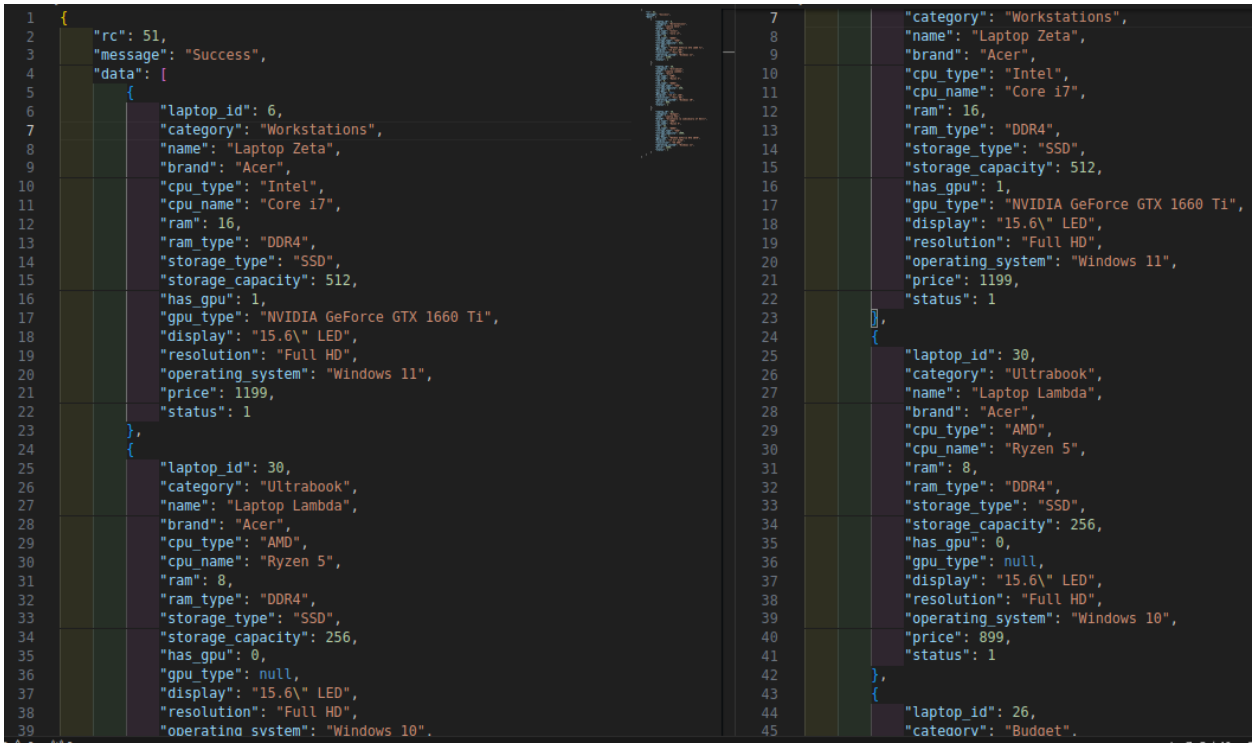
}'

Other Optional keys for “order”: "laptop_id", "category", "name", "brand", "cpu_type", "cpu_name", "ram", "ram_type", "storage_type", "storage_capacity", "has_gpu", "gpu_type", "display", "resolution", "operating_system", "price"

Paging: For the value of the “start” key, specify which record in the list you would like to start showing results from. For the “end” key, specify the record in the list to stop showing results at. To avoid errors, the end key value must be greater than the start key value.

Ordering: Only values of “ASC” or “DESC” including their lowercase equivalents, can have results.

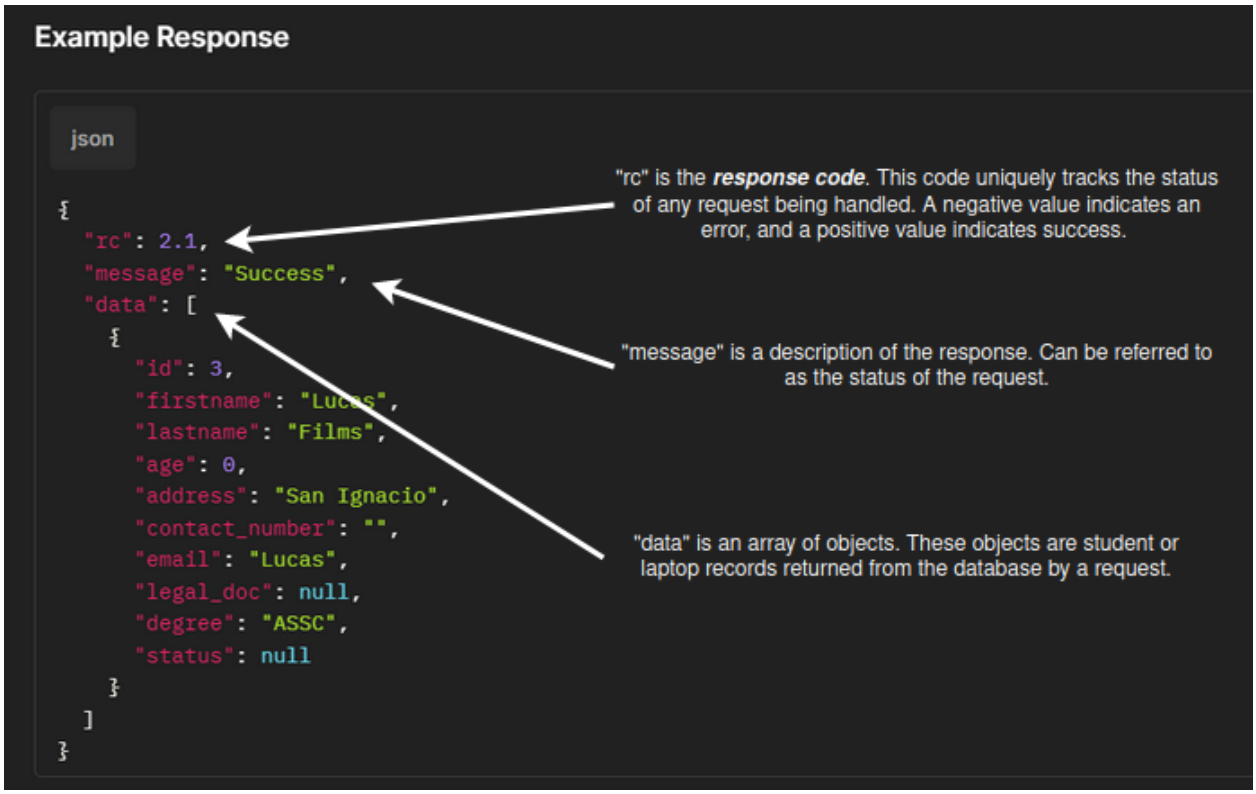
Example Response:




```
{
    "rc": 51,
    "message": "Success",
    "data": [
        {
            "laptop_id": 6,
            "category": "Workstations",
            "name": "Laptop Zeta",
            "brand": "Acer",
            "cpu_type": "Intel",
            "cpu_name": "Core i7",
            "ram": 16,
            "ram_type": "DDR4",
            "storage_type": "SSD",
            "storage_capacity": 512,
            "has_gpu": 1,
            "gpu_type": "NVIDIA GeForce GTX 1660 Ti",
            "display": "15.6\" LED",
            "resolution": "Full HD",
            "operating_system": "Windows 11",
            "price": 1199,
            "status": 1
        },
        {
            "laptop_id": 30,
            "category": "Ultrabook",
            "name": "Laptop Lambda",
            "brand": "Acer",
            "cpu_type": "AMD",
            "cpu_name": "Ryzen 5",
            "ram": 8,
            "ram_type": "DDR4",
            "storage_type": "SSD",
            "storage_capacity": 256,
            "has_gpu": 0,
            "gpu_type": null,
            "display": "15.6\" LED",
            "resolution": "Full HD",
            "operating_system": "Windows 10",
            "price": 899,
            "status": 1
        },
        {
            "laptop_id": 26,
            "category": "Budget",
```

Response Format

API Structure



HTTP Response Codes Usage

Two response codes are used in the current version of the API:

HTTP Response Code 200

- 200 OK: The request was successful.

HTTP Response Code 400

- The client's request is incomplete.

HTTP Response Code 401

- Access Denied to client requesting resource.

HTTP Response Code 429

- The client's request is malformed.

HTTP Response Code 500

- Internal Server Error

Response Codes and Their Meanings

Common Errors

-1: Invalid Request. Unknown resource Requested;

-2: No access to parent resource.

-3: Unsupported Request Method. Could not find the request method GET, POST, PUT, or DELETE

-4: No Database connection; Database connection was lost

-5: Query Execution Errors

-6: Query successful but no records retrieved

GET Requests Errors

-11: Invalid GET request for resource “users”

-12: No access to sub resource “employees”

-13: No access to sub resource “customers”

-14: No access to “/” sub resource using method GET.

-15: All users’ details was not found

-16: User details not found for provided ID

-17: Employee details not found

-18: Customer details not found

-19:- Invalid GET request for resource “laptops”

-20: No access to sub resource “/” using method GET for laptops

-21: Laptops’ details not found

-22: Laptop details not found for provided ID

POST Request Errors

-23: Invalid POST request for resource “users”

-24: No access to resource “users” using method POST

-25:

-26:

PUT Request Errors

-27: Invalid PUT request for resource “users”

-28: No access to resource “users” using method PUT

-29: Invalid PUT request for resource “laptops”

-30: No access to resource “laptops” using method PUT

DELETE Request Errors

-31: Invalid DELETE request for resource “users”

-32: No access to resource “users” using method DELETE

-33: Invalid DELETE request for resource “laptops”

-34: No access to resource “laptops” using method DELETE

Success Response Codes

50: Successful GET request for all users’ details

51: Successful GET request for all laptops’ details

52: Successful GET request for user with the provided ID

53: Successful GET request for laptop with the provided ID

54: Successful GET request for all employees’ details

55: Successful GET request for all customers’ details

56: Successful POST request for creating a new user

57: Successful PUT request for updating a user with provided ID

58: Successful PUT request for updating a laptop with provided ID

59: Successful DELETE request for deleting a user with provided ID

60: Successful DELETE request for deleting a laptop with provided ID

100: Test function is working.

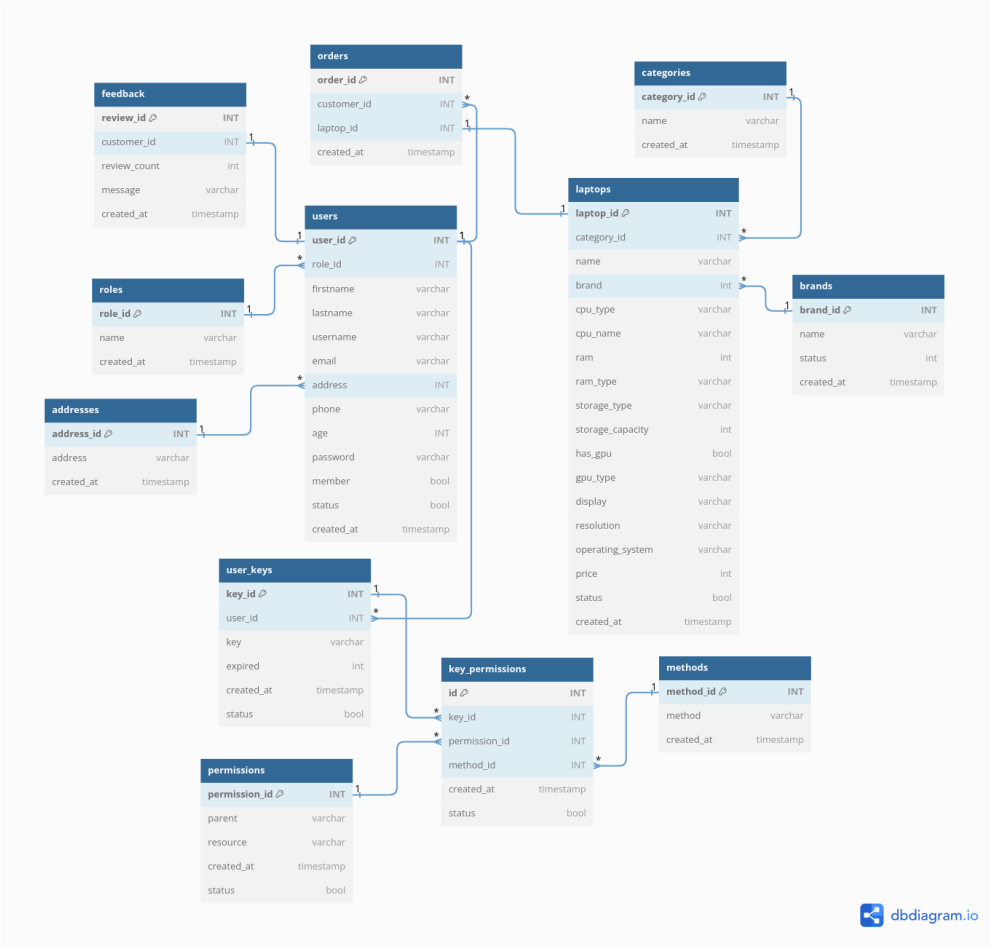
Support

If there are unhandled errors, kindly email me at 2021154344@ub.edu.bz. Ensure you have proper descriptions of the encountered problem and the events leading up to it. If you can, include pictures to demonstrate further.

Changelog

API updates, new features, and bug fixes are communicated in the commits of the git repository.

Appendix



Database Entity Relationship Diagram (ERD)