

API DOCUMENTATION

FOR PERFORMING REST API REQUESTS ON LAPTOP

STORE DATABASE V1.2

University of Belize



Prepared by

Tadeo Bennett

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:

```
1  {
2    "rc": 50,
3    "message": "Success",
4    "data": [
5      {
6        "user_id": 1,
7        "role_id": 1,
8        "firstname": "Tadeo",
9        "lastname": "Bennett",
10       "username": "TBennett",
11       "email": "tadeo@gmail.com",
12       "address": 4,
13       "phone": "",
14       "age": 21,
15       "password": "Tadeo2002",
16       "member": 0,
17       "status": 1,
18       "created_at": "2023-10-12 06:38:55"
19     },
20     {
21       "user_id": 2,
22       "role_id": 2,
23       "firstname": "William",
24       "lastname": "Locario",
25       "username": "WLocario",
26       "email": "william@gmail.com",
27       "address": 6,
28       "phone": "",
29       "age": 22,
30       "password": "William2002",
31       "member": 1,
32       "status": 1,
33       "created_at": "2023-10-12 06:38:55"
34     },
35     {
36       "user_id": 3,
37       "role_id": 2,
38       "firstname": "Victor",
39       "lastname": "Castillo",
40       "username": "VCastillo",
41       "email": "victor@gmail.com",
42       "address": 2,
43       "phone": "",
44       "age": 19,
45       "password": "Victor2002",
46       "member": 0,
47       "status": 1,
48       "created_at": "2023-10-12 06:38:55"
49     }
50   ]
51 }
```

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:

```
1  {
2    "rc": 51,
3    "message": "Success",
4    "data": [
5      {
6        "laptop_id": 1,
7        "category_id": 1,
8        "name": "Laptop Alpha",
9        "brand": 8,
10       "cpu_type": "Intel",
11       "cpu_name": "Core i5",
12       "ram": 8,
13       "ram_type": "DDR4",
14       "storage_type": "SSD",
15       "storage_capacity": 256,
16       "has_gpu": 1,
17       "gpu_type": "NVIDIA GeForce GTX 1650",
18       "display": "15.6\" LED",
19       "resolution": "Full HD",
20       "operating_system": "Windows 10",
21       "price": 999,
22       "status": 1,
23       "created_at": "2023-10-12 06:38:56"
24     },
25     {
26       "laptop_id": 2,
27       "category_id": 2,
28       "name": "Laptop Beta",
29       "brand": 10,
30       "cpu_type": "AMD",
31       "cpu_name": "Ryzen 7",
32       "ram": 16,
33       "ram_type": "DDR4",
34       "storage_type": "SSD",
35       "storage_capacity": 1000,
36       "has_gpu": 1,
37       "gpu_type": "NVIDIA GeForce RTX 3070",
38       "display": "17.3\" OLED",
39       "resolution": "4K UHD",
40       "operating_system": "Windows 11",
41       "price": 1299,
42       "status": 1,
43       "created_at": "2023-10-12 06:38:56"
44     },
45     {
46       "laptop_id": 18,
47       "category_id": 18,
48       "name": "Laptop Sigma",
49       "brand": 2,
50       "cpu_type": "AMD",
51       "cpu_name": "Ryzen 9",
52       "ram": 32,
53       "ram_type": "DDR4",
54       "storage_type": "SSD",
55       "storage_capacity": 1000,
56       "has_gpu": 1,
57       "gpu_type": "NVIDIA GeForce RTX 3070",
58       "display": "17.3\" OLED",
59       "resolution": "4K UHD",
60       "operating_system": "Windows 10",
61       "price": 899,
62       "status": 1,
63       "created_at": "2023-10-12 06:38:56"
64     },
65     {
66       "laptop_id": 30,
67       "category_id": 1,
68       "name": "Laptop Lambda",
69       "brand": 5,
70       "cpu_type": "AMD",
71       "cpu_name": "Ryzen 5",
72       "ram": 8,
73       "ram_type": "DDR4",
74       "storage_type": "SSD",
75       "storage_capacity": 256,
76       "has_gpu": 0,
77       "gpu_type": null,
78       "display": "15.6\" LED",
79       "resolution": "Full HD",
80       "operating_system": "Windows 10",
81       "price": 899,
82       "status": 1,
83       "created_at": "2023-10-12 06:38:56"
84     },
85     {
86       "laptop_id": 30,
87       "category_id": 1,
88       "name": "Laptop Lambda",
89       "brand": 5,
90       "cpu_type": "AMD",
91       "cpu_name": "Ryzen 5",
92       "ram": 8,
93       "ram_type": "DDR4",
94       "storage_type": "SSD",
95       "storage_capacity": 256,
96       "has_gpu": 0,
97       "gpu_type": null,
98       "display": "15.6\" LED",
99       "resolution": "Full HD",
100      "operating_system": "Windows 10",
101      "price": 899,
102      "status": 1,
103      "created_at": "2023-10-12 06:38:56"
104    },
105    ]
106  }
```

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:

```
1  {  
2    "rc": 52,  
3    "message": "Success",  
4    "data": [  
5      {  
6        "user_id": 3,  
7        "role_id": 2,  
8        "firstname": "Victor",  
9        "lastname": "Castillo",  
10       "username": "VCastillo",  
11       "email": "victor@gmail.com",  
12       "address": 2,  
13       "phone": "",  
14       "age": 19,  
15       "password": "Victor2002",  
16       "member": 0,  
17       "status": 1,  
18       "created_at": "2023-10-12 06:38:55"  
19     }  
20   ]  
21 }
```

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:

```
1  {  
2    "rc": 53,  
3    "message": "Success",  
4    "data": [  
5      {  
6        "laptop_id": 3,  
7        "category_id": 3,  
8        "name": "Laptop Gamma",  
9        "brand": 3,  
10       "cpu_type": "Intel",  
11       "cpu_name": "Core i7",  
12       "ram": 16,  
13       "ram_type": "DDR4",  
14       "storage_type": "SSD",  
15       "storage_capacity": 512,  
16       "has_gpu": 1,  
17       "gpu_type": "NVIDIA GeForce RTX 3060",  
18       "display": "15.6\" IPS",  
19       "resolution": "Full HD",  
20       "operating_system": "Windows 10",  
21       "price": 1299,  
22       "status": 1,  
23       "created_at": "2023-10-12 06:38:56"  
24     }  
25   ]  
26 }
```

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:

```
1 {  
2   "rc": 54,  
3   "message": "Success",  
4   "data": [  
5     {  
6       "user_id": 1,  
7       "role_id": 1,  
8       "firstname": "Tadeo",  
9       "lastname": "Bennett",  
10      "username": "TBennett",  
11      "email": "tadeo@gmail.com",  
12      "address": 4,  
13      "phone": "",  
14      "age": 21,  
15      "password": "Tadeo2002",  
16      "member": 0,  
17      "status": 1,  
18      "created_at": "2023-10-12 06:38:55"  
19    }  
20  ]  
21 }
```


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:

```
1 {  
2   "rc": 55,  
3   "message": "Success",  
4   "data": [  
5     {  
6       "user_id": 2,  
7       "role_id": 2,  
8       "firstname": "William",  
9       "lastname": "Locario",  
10      "username": "WLocario",  
11      "email": "william@gmail.com",  
12      "address": 6,  
13      "phone": "",  
14      "age": 22,  
15      "password": "William2002",  
16      "member": 1,  
17      "status": 1,  
18      "created_at": "2023-10-12 06:38:55"  
19    },  
20    {  
21      "user_id": 3,  
22      "role_id": 2,  
23      "firstname": "Victor",  
24      "lastname": "Castillo",  
25      "username": "VCastillo",  
26      "email": "victor@gmail.com",  
27      "address": 2,  
28      "phone": "",  
29      "age": 19,  
30      "password": "Victor2002",  
31      "member": 0,  
32      "status": 1,  
33      "created_at": "2023-10-12 06:38:55"  
34    }  
35  ]  
36 }
```

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:

```
1  {
2    "rc": 56,
3    "message": "Successful user creation",
4    "new_user_id": 4
5  }
```

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:

```
1  {
2    "rc": 57,
3    "message": "Success. User Updated",
4    "data": [
5      {
6        "user_id": 2,
7        "role_id": 2,
8        "firstname": "William",
9        "lastname": "Locario",
10       "username": "WLocario",
11       "email": "william@gmail.com",
12       "address": 6,
13       "phone": "",
14       "age": 22,
15       "password": "William2002",
16       "member": 1,
17       "status": 1,
18       "created_at": "2023-10-12 06:38:55"
19     }
20   ],
21   "newdata": [
22     {
23       "user_id": 2,
24       "role_id": 2,
25       "firstname": "Luke",
26       "lastname": "SKyWalker",
27       "username": "LSkyWalker",
28       "email": "luke@gmail.com",
29       "address": 6,
30       "phone": "",
31       "age": 22,
32       "password": "William2002",
33       "member": 1,
34       "status": 1,
35       "created_at": "2023-10-12 06:38:55"
36     }
37   ]
38 }
```

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:

```
1  {
2    "rc": 58,
3    "message": "Success. Laptop Updated",
4    "data": [
5      {
6        "laptop_id": 4,
7        "category_id": 4,
8        "name": "Laptop Delta",
9        "brand": 15,
10       "cpu_type": "AMD",
11       "cpu_name": "Ryzen 5",
12       "ram": 12,
13       "ram_type": "DDR4",
14       "storage_type": "HDD",
15       "storage_capacity": 128,
16       "has_gpu": 0,
17       "gpu_type": null,
18       "display": "14\" LCD",
19       "resolution": "HD",
20       "operating_system": "Linux",
21       "price": 699,
22       "status": 1,
23       "created_at": "2023-10-12 06:38:56"
24     }
25   ],
26   "newdata": [
27     {
28       "laptop_id": 4,
29       "category_id": 4,
30       "name": "MyNewLaptop 23",
31       "brand": 15,
32       "cpu_type": "AMD",
33       "cpu_name": "Ryzen 5",
34       "ram": 64,
35       "ram_type": "DDR4",
36       "storage_type": "HDD",
37       "storage_capacity": 128,
38       "has_gpu": 0,
39       "gpu_type": null,
40       "display": "14\" LCD",
41       "resolution": "HD",
42       "operating_system": "Linux",
43       "price": 699,
44       "status": 1,
45       "created_at": "2023-10-12 06:38:56"
46     }
47   ]
48 }
```

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:

```
1 {  
2   "rc": 59,  
3   "message": "Success. User deleted with id 3",  
4   "data": [  
5     {  
6       "user_id": 3,  
7       "role_id": 2,  
8       "firstname": "Victor",  
9       "lastname": "Castillo",  
10      "username": "VCastillo",  
11      "email": "victor@gmail.com",  
12      "address": 2,  
13      "phone": "",  
14      "age": 19,  
15      "password": "Victor2002",  
16      "member": 0,  
17      "status": 1,  
18      "created_at": "2023-10-12 06:38:55"  
19    }  
20  ]  
21 }
```

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:

```
1 {  
2   "rc": 60,  
3   "message": "Success. Laptop deleted with id 25",  
4   "data": [  
5     {  
6       "laptop_id": 25,  
7       "category_id": 5,  
8       "name": "Laptop Zeta",  
9       "brand": 10,  
10      "cpu_type": "Intel",  
11      "cpu_name": "Core i5",  
12      "ram": 8,  
13      "ram_type": "DDR4",  
14      "storage_type": "SSD",  
15      "storage_capacity": 256,  
16      "has_gpu": 1,  
17      "gpu_type": "NVIDIA GeForce GTX 1650",  
18      "display": "15.6\" LED",  
19      "resolution": "Full HD",  
20      "operating_system": "Windows 11",  
21      "price": 1199,  
22      "status": 1,  
23      "created_at": "2023-10-12 06:38:56"  
24    }  
25  ]  
26 }
```

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?{_0227029b5e7013d468d8155a47f1ec2b38f9f129aaadb9a668dd956da
e443540' \
--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:

```

1  {
2    "rc": 51,
3    "message": "Success",
4    "data": [
5      {
6        "laptop_id": 6,
7        "category": "Workstations",
8        "name": "Laptop Zeta",
9        "brand": "Acer",
10       "cpu_type": "Intel",
11       "cpu_name": "Core i7",
12       "ram": 16,
13       "ram_type": "DDR4",
14       "storage_type": "SSD",
15       "storage_capacity": 512,
16       "has_gpu": 1,
17       "gpu_type": "NVIDIA GeForce GTX 1660 Ti",
18       "display": "15.6\" LED",
19       "resolution": "Full HD",
20       "operating_system": "Windows 11",
21       "price": 1199,
22       "status": 1
23     },
24     {
25       "laptop_id": 30,
26       "category": "Ultrabook",
27       "name": "Laptop Lambda",
28       "brand": "Acer",
29       "cpu_type": "AMD",
30       "cpu_name": "Ryzen 5",
31       "ram": 8,
32       "ram_type": "DDR4",
33       "storage_type": "SSD",
34       "storage_capacity": 256,
35       "has_gpu": 0,
36       "gpu_type": null,
37       "display": "15.6\" LED",
38       "resolution": "Full HD",
39       "operating system": "Windows 10".
40     },
41     {
42       "category": "Workstations",
43       "name": "Laptop Zeta",
44       "brand": "Acer",
45       "cpu_type": "Intel",
46       "cpu_name": "Core i7",
47       "ram": 16,
48       "ram_type": "DDR4",
49       "storage_type": "SSD",
50       "storage_capacity": 512,
51       "has_gpu": 1,
52       "gpu_type": "NVIDIA GeForce GTX 1660 Ti",
53       "display": "15.6\" LED",
54       "resolution": "Full HD",
55       "operating_system": "Windows 11",
56       "price": 1199,
57       "status": 1
58     },
59     {
60       "laptop_id": 30,
61       "category": "Ultrabook",
62       "name": "Laptop Lambda",
63       "brand": "Acer",
64       "cpu_type": "AMD",
65       "cpu_name": "Ryzen 5",
66       "ram": 8,
67       "ram_type": "DDR4",
68       "storage_type": "SSD",
69       "storage_capacity": 256,
70       "has_gpu": 0,
71       "gpu_type": null,
72       "display": "15.6\" LED",
73       "resolution": "Full HD",
74       "operating_system": "Windows 10",
75       "price": 899,
76       "status": 1
77     },
78     {
79       "laptop_id": 26,
80       "category": "Budget",
81       "name": "Laptop Alpha",
82       "brand": "Acer",
83       "cpu_type": "Intel",
84       "cpu_name": "Core i3",
85       "ram": 8,
86       "ram_type": "DDR4",
87       "storage_type": "HDD",
88       "storage_capacity": 1000,
89       "has_gpu": 0,
90       "gpu_type": null,
91       "display": "15.6\" LED",
92       "resolution": "Full HD",
93       "operating_system": "Windows 10",
94       "price": 499,
95       "status": 1
96     }
97   ]
98 }

```


Response Format

API Structure

Example Response

```

json
{
  "rc": 2.1,
  "message": "Success",
  "data": [
    {
      "id": 3,
      "firstname": "Lucas",
      "lastname": "Films",
      "age": 0,
      "address": "San Ignacio",
      "contact_number": "",
      "email": "Lucas",
      "legal_doc": null,
      "degree": "ASSC",
      "status": null
    }
  ]
}

```

"rc" is the **response code**. This code uniquely tracks the status of any request being handled. A negative value indicates an error, and a positive value indicates success.

"message" is a description of the response. Can be referred to as the status of the request.

"data" is an array of objects. These objects are student or laptop records returned from the database by a request.

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.

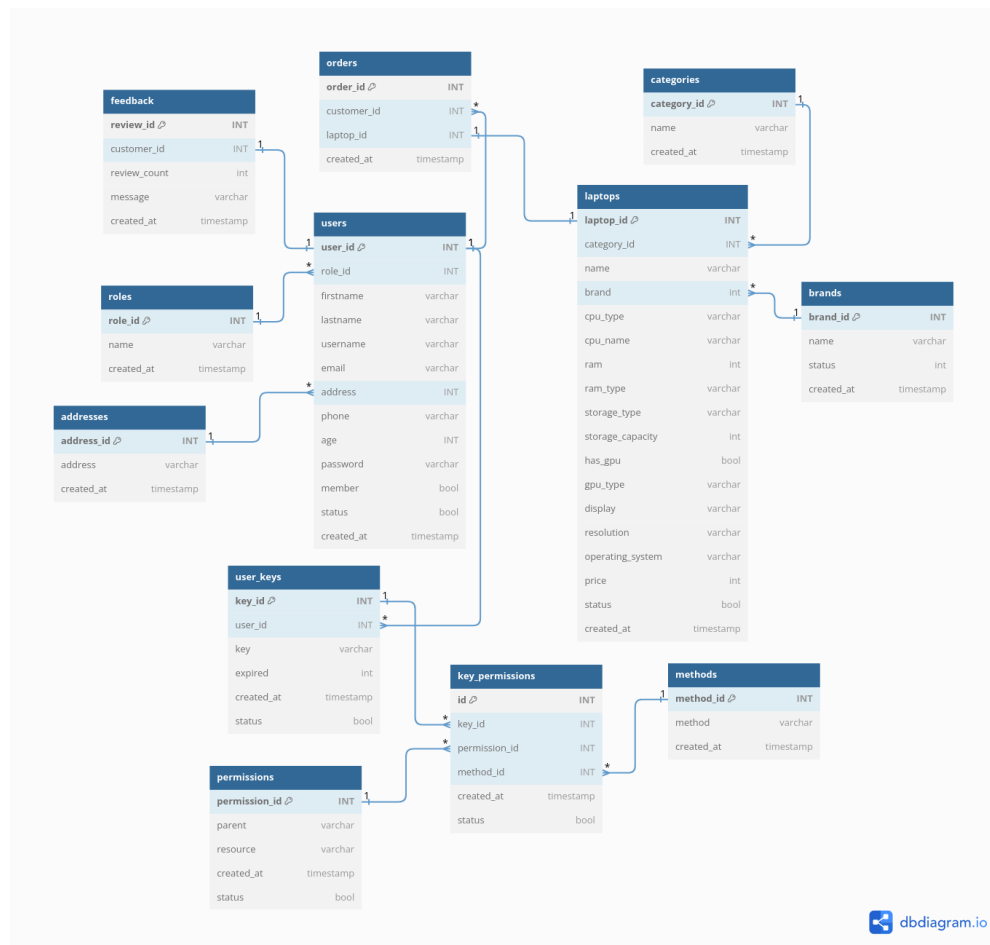
Support

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)