

CASE STUDY: Corporate Equipment Allocation and Tracking System

1. Database Schema

Employee

```
CREATE TABLE employees (
    employee_id INT AUTO_INCREMENT PRIMARY KEY,
    name        VARCHAR(100) NOT NULL,
    department  VARCHAR(100) NOT NULL,
    email       VARCHAR(150) NOT NULL UNIQUE
);
```

Sample data

```
{
    "name": "Reception-Staff",
    "email": "recereceptionstaff@yopmail.com",
    "department": "Reception"
}
```

User

```
CREATE TABLE users (
    user_id    BIGINT AUTO_INCREMENT PRIMARY KEY,
    employee_id INT NOT NULL,
    passwordhash VARCHAR(150),
    role        ENUM('SYSTEM_ADMIN', 'RECEPTION_STAFF', 'INVENTORY_STAFF',
'MAINTENANCE_STAFF') NOT NULL,
    is_active   BOOLEAN DEFAULT TRUE,
    created_at  TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id)
);
```

Sample Data

```
{
    "userId": 1,
    "employeeId": 1,
```

```
"passwordHash": "$2a$10$VPwqtVUVKQpS168zD3RdSuqpXbD6hyUoRy/pzADacIdejLDmUG7Ru",
"role": "SYSTEM_ADMIN",
"isActive": true,
"createdAt": "2025-11-18T12:27:33Z"
}
```

Equipment Master

Equipment Types

```
CREATE TABLE equipment_types (
    type_id  INT AUTO_INCREMENT PRIMARY KEY,
    type_name VARCHAR(50) NOT NULL UNIQUE
);
```

Sample data

```
{
    "typeId": 1,
    "typeName": "Laptop"
}
```

Equipment Items

```
CREATE TABLE equipment_items (
    equipment_id  INT AUTO_INCREMENT PRIMARY KEY,
    serial_number VARCHAR(100) NOT NULL UNIQUE,
    type_id      INT NOT NULL,
    status        ENUM('AVAILABLE', 'ALLOCATED', 'UNDER_MAINTENANCE') NOT NULL
    DEFAULT 'AVAILABLE',
    created_at    DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (type_id) REFERENCES equipment_types(type_id)
);
```

Sample Data

```
{
    "equipmentId": 21,
    "serialNumber": "LAP-001",
    "typeId": 1,
    "status": "AVAILABLE",
    "createdAt": "2025-11-19T09:57:24Z"
}
```

Equipment Allocation

```
CREATE TABLE equipment_allocations (
    allocation_id    INT AUTO_INCREMENT PRIMARY KEY,
    equipment_id     INT NOT NULL,
    employee_id      INT NOT NULL,
    allocated_at     DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    expected_return_at DATETIME NOT NULL,
    returned_at      DATETIME DEFAULT NULL,
    condition_on_return ENUM('GOOD', 'DAMAGED') DEFAULT NULL,
    FOREIGN KEY (equipment_id) REFERENCES equipment_items(equipment_id),
    FOREIGN KEY (employee_id) REFERENCES employees(employee_id)
);
```

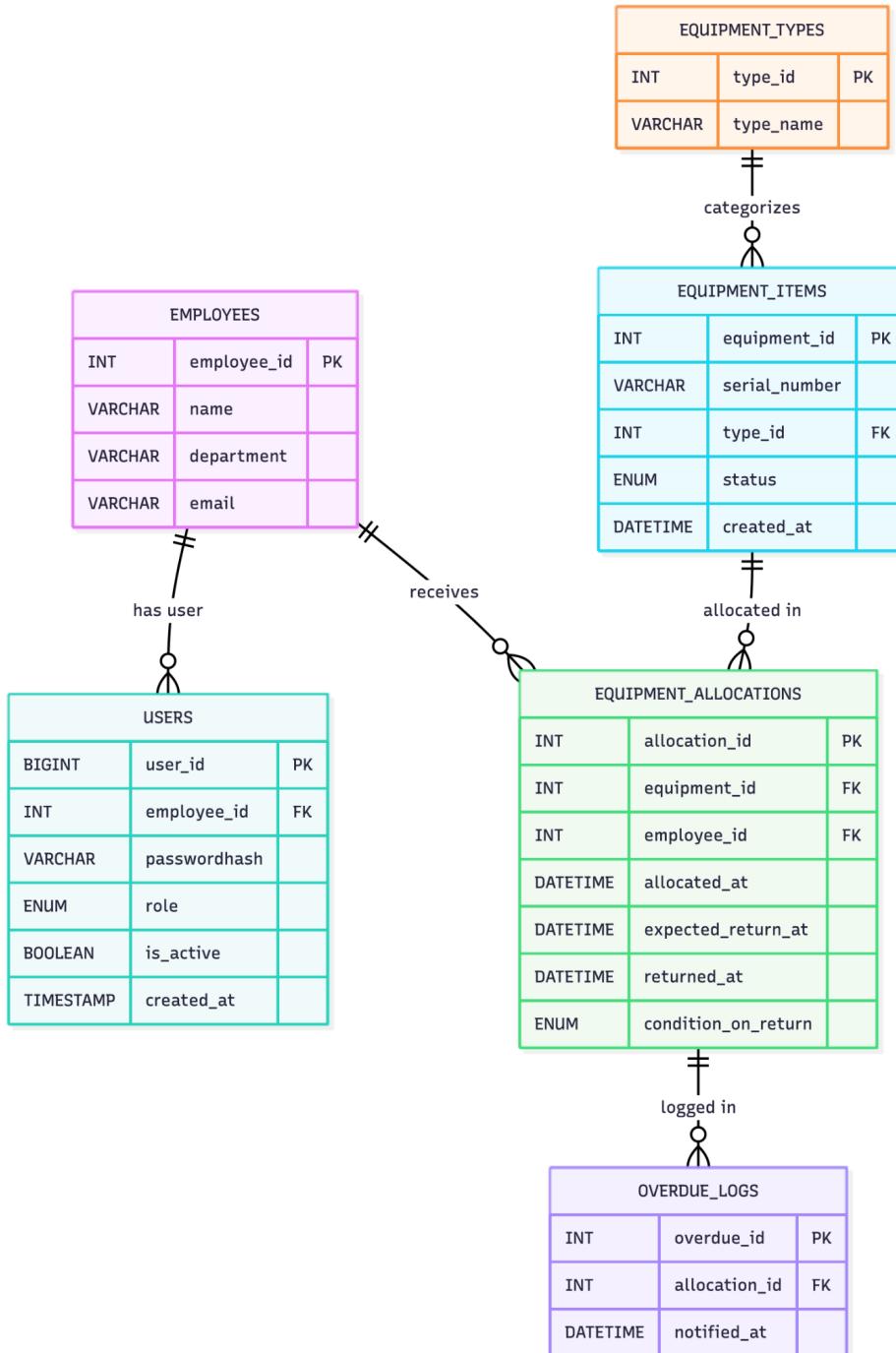
Sample data

```
{
    "allocationId": 1,
    "equipmentId": 21,
    "employeeId": 3,
    "allocatedAt": "2025-11-22T09:43:51Z",
    "expectedReturnAt": "2025-11-23T10:30:00Z",
    "returnedAt": "2025-11-22T10:01:24Z",
    "conditionOnReturn": "GOOD"
}
```

Logs

```
CREATE TABLE overdue_logs (
    overdue_id    INT AUTO_INCREMENT PRIMARY KEY,
    allocation_id INT NOT NULL,
    notified_at   DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    lastRemainderAt DATETIME DEFAULT NULL,
    FOREIGN KEY (allocation_id) REFERENCES equipment_allocations(allocation_id)
);
```

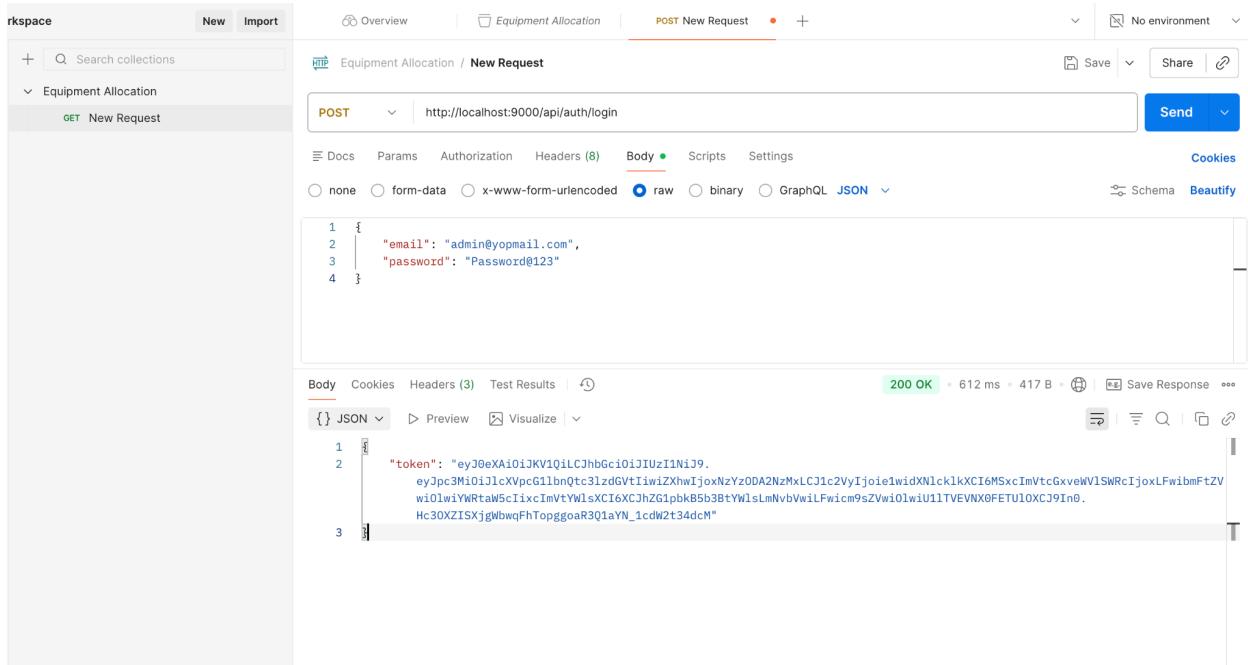
2. ER Diagram



3. Output screenshots

3.1 Login and staff registration

3.1.1 Login as System Admin



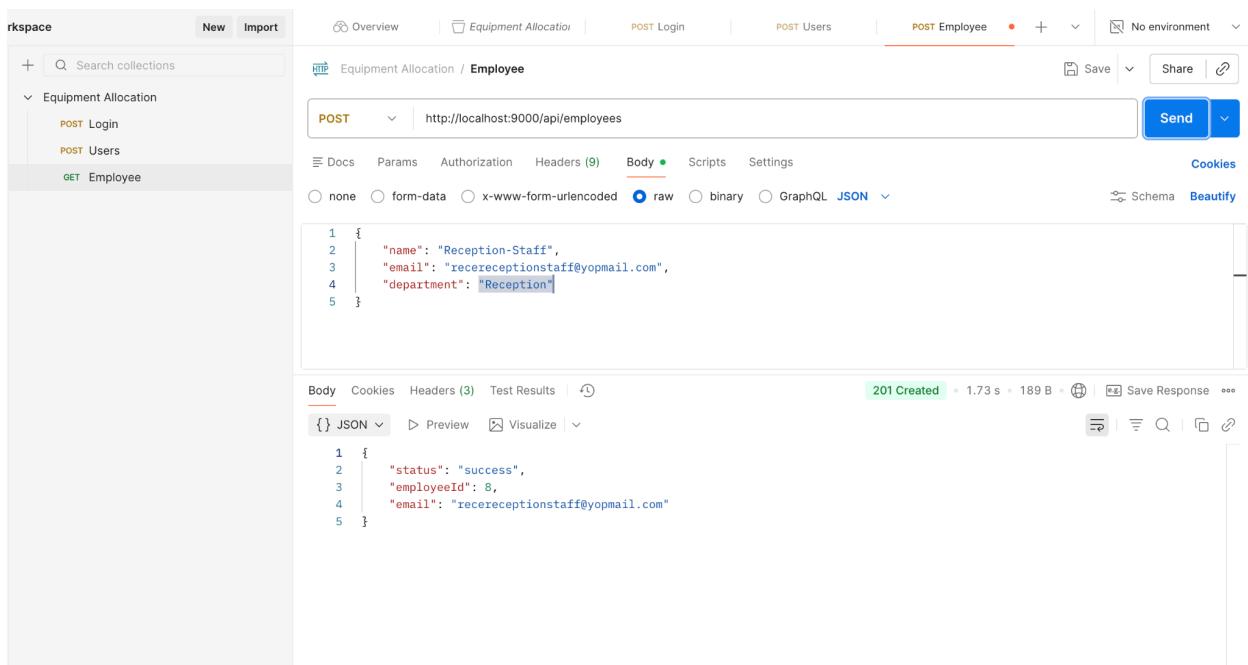
The screenshot shows the Rackspace API interface. A POST request is being made to `http://localhost:9000/api/auth/login`. The request body is a JSON object:

```
1 {
2   "email": "admin@yopmail.com",
3   "password": "Password@123"
4 }
```

The response status is `200 OK`, with a response time of 612 ms and a response size of 417 B. The response body contains a token:

```
1 {
2   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiIcXVpcG1bnNtc3LzdGVtIiwidXNzIjoxNzYzODA2NzMxLCJ1c2VtIjoie1widXNlcklkXCi6MsxcImVtcGxvW1SWRcIjoxLFWibmFtZVwiOlwiYRtaW5ciixcImVtVlsxCi6xCJzhZG1pbk85b3ByWlsLmNvbVwiflwicm9sZVwiOlwiU11tVEVNX0FETU10XC39In0.Hc30XZISXjgWbwqFhTopggoaR3Q1aYN_1cdW2t34dcM"
```

3.1.2 Create and Register Reception Staff



The screenshot shows the Rackspace API interface. A POST request is being made to `http://localhost:9000/api/employees`. The request body is a JSON object:

```
1 {
2   "name": "Reception-Staff",
3   "email": "recreceptionstaff@yopmail.com",
4   "department": "Reception"
5 }
```

The response status is `201 Created`, with a response time of 1.73 s and a response size of 189 B. The response body indicates success:

```
1 {
2   "status": "success",
3   "employeeId": 8,
4   "email": "recreceptionstaff@yopmail.com"
5 }
```

The screenshot shows the Rackspace API Explorer interface. On the left, the sidebar lists collections: Equipment Allocation (with POST Login, POST Users, and GET Employee), Overview, Equipment Allocation Allocation, POST Login, POST Users, POST Employee, and No environment. The main area shows a POST request to `http://localhost:9000/api/users`. The request body is:

```

1 {
2     "email": "receptionstaff@yopmail.com",
3     "password": "Password0123",
4     "role": "RECEPTION_STAFF"
5 }

```

The response status is 201 Created, with a response time of 506 ms and 173 B. The response body is:

```

1 {
2     "status": "success",
3     "employeeId": 8,
4     "role": "RECEPTION_STAFF"
5 }

```

3.2 Equipment Types

The screenshot shows the Rackspace API Explorer interface. On the left, the sidebar lists collections: Equipment Allocation (with POST Login, POST Users, POST Employee, and GET Equipment-types), Overview, Equipment Allocation Allocation, POST Login, POST Users, POST Employee, and No environment. The main area shows a GET request to `http://localhost:9000/api/equipment-types`. The request body is empty, indicated by the message "This request does not have a body".

The response status is 200 OK, with a response time of 254 ms and 276 B. The response body is:

Root / 4	
typeid	3
typeName	Tablet

3.3 Equipments

The screenshot shows the Postman interface with the following details:

- Collection:** rkspace
- Request Type:** GET
- URL:** http://localhost:9000/api/equipment
- Headers:** User-Agent (PostmanRuntime/7.49.1), Accept (*/*), Accept-Encoding (gzip, deflate, br), Connection (keep-alive), Authorization (Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9....)
- Body:** (Empty)
- Test Results:** 200 OK, 236 ms, 1.19 KB
- Response Data:** A table showing 10 equipment entries:

equipmentId	serialNumber	typId	status	createdAt
0 21	LAP-001	1	AVAILABLE	2025-11-19T09:57:24Z
1 22	LAP-002	1	AVAILABLE	2025-11-19T10:50:19Z
2 23	LAP-003	1	AVAILABLE	2025-11-19T10:50:19Z
3 24	PRO-101	2	AVAILABLE	2025-11-19T10:50:19Z
4 25	PRO-102	2	AVAILABLE	2025-11-19T10:50:19Z
5 26	TAB-501	3	AVAILABLE	2025-11-19T10:50:19Z
6 27	TAB-502	3	AVAILABLE	2025-11-19T10:50:19Z
7 28	IPD-901	4	AVAILABLE	2025-11-19T10:50:19Z
8 29	CAM-777	5	AVAILABLE	2025-11-19T10:50:19Z
9 30	CAM-778	5	AVAILABLE	2025-11-19T10:50:19Z

3.4 Create allocation

3.4.1 Login as Reception Staff

The screenshot shows the Postman interface with the following details:

- Collection:** rkspace
- Request Type:** POST
- URL:** http://localhost:9000/api/auth/login
- Headers:** (None selected)
- Body:** (Raw JSON)

```
1 {
2   "email": "receptionstaff@yopmail.com",
3   "password": "Password@123"
4 }
```
- Test Results:** 200 OK, 675 ms, 446 B
- Response Data:** A JSON object containing a token:

```
1 {
2   "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.
eyJpc3Mi0iJlcXVpcG1ibnQtc3Izd0VtIwiZXhwIjoxNzY0DA4MDQzLCJ1c2VyIjoie1widXNlcklkXCI6NSxcImVtcGxveWlSWRcIjo4LFwibmFtZV
wi01wiUmvjZXBoaw9uLVN0YWZmXClxJC1bwFpbFw10lwicVjZXB0aW9uc3RhZmZAeW9wbFpbC5jb21cIixcInJvbGvciJpcI1JF00VQE1PTI9TVEFG
RlwifsfJ9.uHJbX22FjqzbwZ1xx0uuOPTQdIkQ4a82jHBHdoimg"
```

3.4.2 Create equipment allocation requests, recording employee id along with the expected return date.

The screenshot shows the Rackspace API interface. On the left, there's a sidebar with a tree view of collections: Equipment Allocation (with sub-options like POST Login, POST Users, POST Employee, GET Equipment-types, GET Equipments, and GET Allocation), and other collections like Overview, Equipment, POST Login, POST Users, POST Employee, GET Equipment, and POST Allocation. The main area shows a POST request to 'Equipment Allocation / Allocation' with the URL 'http://localhost:9000/api/allocations'. The 'Body' tab is selected, displaying the following JSON payload:

```
1 {
2   "equipmentId": 21,
3   "employeeId": 3,
4   "expectedReturnAt": "2025-11-23T10:30:00Z"
5 }
6
```

Below the body, the response is shown as a 201 Created status with a timestamp of 2.04 s and 131 B. The response body is also displayed as JSON:

```
1 {
2   "allocationId": 1
3 }
```

3.4.3 Mail notification to employee and inventory about allocation

The screenshot shows an email inbox. A new email from 'sownthariponnusamy@gmail.com' to 'jerry' is highlighted. The subject is 'Allocation Created - Laptop'. The email content is as follows:

Allocation Created - Laptop

S [sownthariponnusamy@gmail.com](#)
to jerry ▾

2 of 897 2:13 PM (3 minutes ago) [star](#) [smile](#) [forward](#) [more](#)

Equipment Allocation Created

Hello Jerry,

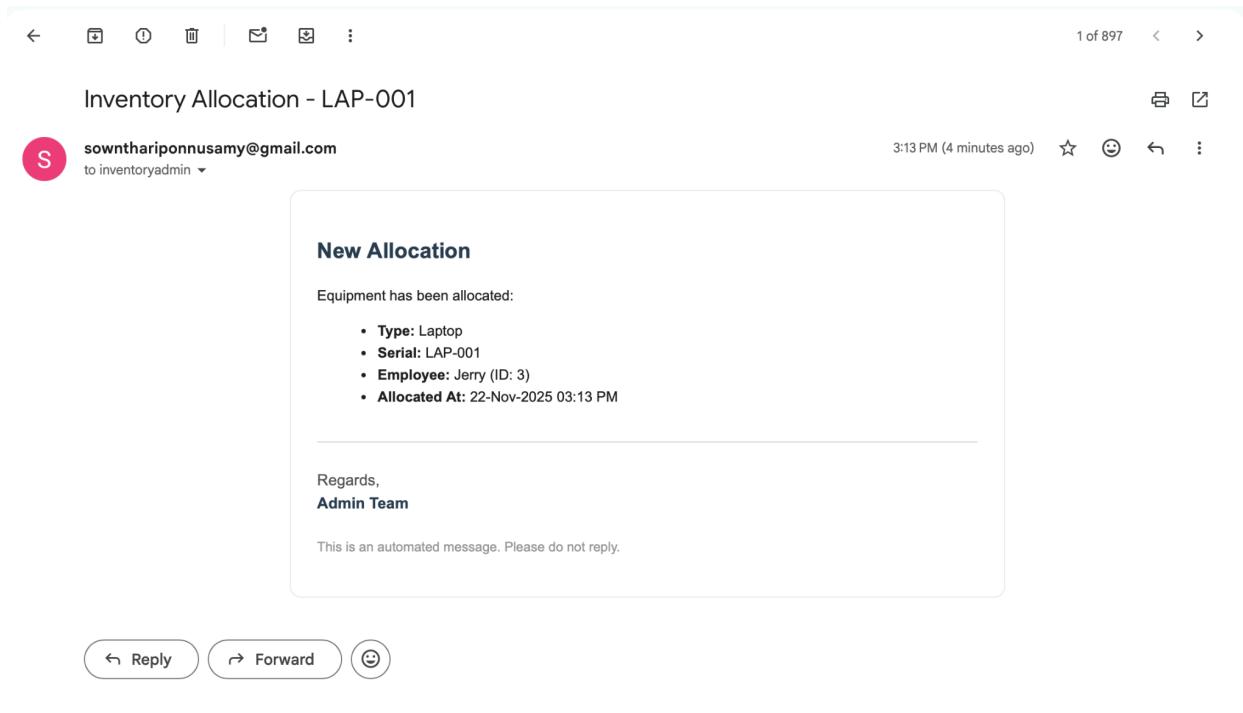
Your equipment has been allocated with the following details:

- Type: Laptop
- Serial: LAP-001
- Allocated At: 22-Nov-2025 03:13 PM

Regards,
Admin Team

This is an automated message. Please do not reply.

Reply Forward



3.5 Equipment Return

3.5.1 Equipment return with condition as “GOOD”, automated notification to inventory team

The screenshot shows a REST API tool interface. The URL is `http://localhost:9000/api/allocations/1/return`. The method selected is `PATCH`. The request body is:

```
1 {
2   "conditionOnReturn": "GOOD"
3 }
```

The response status is `200 OK` with a response time of `1.41 s`, a size of `129 B`, and a `Save Response` button. The response body is:

```
1 {
2   "status": "returned"
3 }
```

Returned GOOD - LAP-001

sownthariponnusamy@gmail.com
to inventoryadmin ▾

3:31 PM (0 minutes ago) ☆ 😊 ↳ ⋮

Returned - GOOD Condition

The equipment **LAP-001** was returned in good condition.

Returned At: 22-Nov-2025 03:31 PM

Regards,
Admin Team

This is an automated message. Please do not reply.

Reply Forward ...

3.5.2 Equipment return as “DAMAGED”, automated notification to Maintenance team and Equipment status as “UNDER MAINTENANCE” and inventory also gets notified

HTTP Equipment Allocation / Allocation

PATCH http://localhost:9000/api/allocations/2/return

Body (raw)

```
{
  "conditionOnReturn": "DAMAGED",
  "notes": "HDMI and Speaker is not working properly"
}
```

200 OK 1.13 s 129 B

```
{
  "status": "returned"
}
```

1 of 902 < >

Returned DAMAGED - LAP-002

 sownthariponnuusamy@gmail.com
to inventoryadmin ▾

3:37 PM (0 minutes ago)    

Returned - DAMAGED

The equipment **LAP-002** was returned damaged.

Notes: HDMI and Speaker is not working properly

Returned At: 22-Nov-2025 03:37 PM

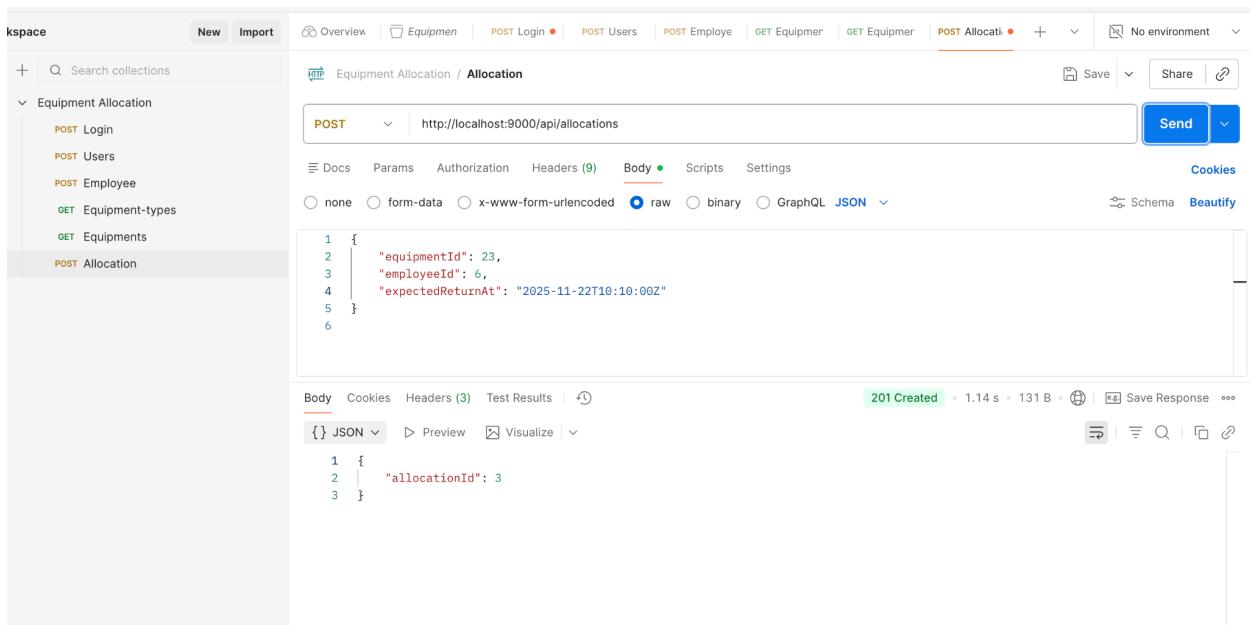
Regards,
Admin Team

This is an automated message. Please do not reply.

3.6 Overdue Notification

3.6.1 Equipment return on 22.11.2025 3.40PM



The screenshot shows the Postman application interface. The left sidebar has a tree view with 'Equipment Allocation' expanded, showing 'POST Login', 'POST Users', 'POST Employee', 'GET Equipment-types', 'GET Equipments', and 'POST Allocation'. The 'POST Allocation' item is selected and highlighted in grey. The main workspace shows a POST request to 'http://localhost:9000/api/allocations'. The 'Body' tab is selected, showing the following JSON payload:

```

1 {
2   "equipmentId": 23,
3   "employeeId": 6,
4   "expectedReturnAt": "2025-11-22T10:10:00Z"
5 }

```

Below the request, the response status is shown as '201 Created' with a timestamp of '1.14 s' and a size of '131 B'. There are also buttons for 'Save Response' and 'Copy'.

Equipment Overdue

sownthariponnuusamy@gmail.com
to tom ▾

4:11 PM (0 minutes ago)

Equipment Overdue

Hello Tom,

Your equipment LAP-003 is overdue by **0 days**.

Due Date: 22-Nov-2025 03:40 PM

Please return it as soon as possible.

Regards,
Admin Team

This is an automated message. Please do not reply.

3.7 Kafka Consumer logs

```
14:46:24.866 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-21] INFO o.a.k.c.c.i.ConsumerCoordinator - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] became follower at term 0
14:46:24.933 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-22] INFO o.a.k.c.c.i.ConsumerCoordinator - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] became follower at term 0
14:46:24.934 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-22] INFO o.a.k.c.c.i.ConsumerCoordinator - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] became follower at term 0
14:46:24.937 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-22] INFO o.a.k.c.c.i.ConsumerRebalanceListenerInvoker - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] rebalance([{"topic": "notification-topic", "partition": 0, "leader": null, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 1, "leader": null, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 2, "leader": null, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 3, "leader": null, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 4, "leader": null, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 5, "leader": null, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 6, "leader": null, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 7, "leader": null, "newLeader": null, "oldLeader": null}], [{"topic": "notification-topic", "partition": 0, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 1, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 2, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 3, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 4, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 5, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 6, "newLeader": null, "oldLeader": null}, {"topic": "notification-topic", "partition": 7, "newLeader": null, "oldLeader": null}], [{}]
14:46:24.953 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-22] INFO o.a.k.c.c.i.ConsumerCoordinator - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] became follower at term 0
14:46:24.969 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-22] INFO o.a.k.c.c.i.SubscriptionState - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] became follower at term 0
14:46:24.974 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-22] INFO o.a.k.c.c.i.SubscriptionState - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] became follower at term 0
14:46:24.979 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-22] INFO o.a.k.c.c.i.SubscriptionState - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] became follower at term 0
14:55:26.609 [KafkaConsumerStreamSystem-akka.kafka.default-dispatcher-8] INFO o.apache.kafka.clients.NetworkClient - [Consumer clientId=consumer-notification-consumer-group-0, groupId=notification-consumer-group] Event received: AllocationCreatedEvent(Laptop,LAP-001,3,Jerry,jerry@yopmail.com,2025-11-22T09:43:52.456110Z,2025-11-23T10:30:00Z,AllocationCreatedEvent,Some(Additiona
[Kafka] Event received: AllocationCreatedEvent(Laptop,LAP-001,3,Jerry,jerry@yopmail.com,2025-11-22T09:43:52.456110Z,2025-11-23T10:30:00Z,AllocationCreatedEvent)
[Kafka] Event received: AllocationCreatedEvent(Laptop,LAP-002,7,Pikko,pikko@yopmail.com,2025-11-22T09:54:36.673772Z,2025-11-23T10:30:00Z,AllocationCreatedEvent,Some(Additiona
[Kafka] Event created: AllocationCreatedEvent(Laptop,LAP-002,7,Pikko,pikko@yopmail.com,2025-11-22T09:54:36.673772Z,2025-11-23T10:30:00Z,AllocationCreatedEvent)
[Kafka] Event received: AllocationCreatedEvent(Laptop,LAP-003,6,Tom,tom@yopmail.com,2025-11-22T09:57:19.513777Z,2025-11-22T10:10:00Z,AllocationCreatedEvent,Some(Additiona
[Kafka] Event created: AllocationCreatedEvent(Laptop,LAP-003,6,Tom,tom@yopmail.com,2025-11-22T09:57:19.513777Z,2025-11-22T10:10:00Z,AllocationCreatedEvent)
[Kafka] Event received: AllocationReturnedEvent(Laptop,LAP-001,3,Jerry,jerry@yopmail.com,2025-11-22T10:01:24.412553Z,6000,None,AllocationReturnedEvent,Some(AdditionalInfo)
[Kafka] Returned 6000: AllocationReturnedEvent(Laptop,LAP-001,3,Jerry,jerry@yopmail.com,2025-11-22T10:01:24.412553Z,6000,None,AllocationReturnedEvent,Some(Additiona
[Kafka] Event received: AllocationReturnedEvent(Laptop,LAP-002,7,Pikko,pikko@yopmail.com,2025-11-22T10:07:24.669541Z,DAMAGED,DAMAGED,Some(HMDI) and Speaker is not working proper
[Kafka] Event returned: AllocationReturnedEvent(Laptop,LAP-002,7,Pikko,pikko@yopmail.com,2025-11-22T10:07:24.669541Z,DAMAGED,DAMAGED,Some(HMDI) and Speaker is not wor
[INFO] [akkaDeadLetter][11/22/2025 15:37:25.829] [KafkaConsumerStreamSystem-akka.actor.default-dispatcher-38] [akka://KafkaConsumerStreamSystem/user/maintenanceActor] M
[Kafka] Event received: AllocationCreatedEvent(Projector,PRO-101,6,Tom,tom@yopmail.com,2025-11-22T10:22:08.021305Z,2025-11-22T10:10:00Z,AllocationCreatedEvent,Some(Additiona
[Kafka] Event created: AllocationCreatedEvent(Projector,PRO-101,6,Tom,tom@yopmail.com,2025-11-22T10:22:08.021305Z,2025-11-22T10:10:00Z,AllocationCreatedEvent)
[Kafka] Event received: AllocationOverdueEvent(Projector,PRO-101,6,Tom,tom@yopmail.com,0,2025-11-22T10:10:00Z,AllocationOverdueEvent,Some(AdditionalInfo(Some(inventory
[Kafka] Overdue: AllocationOverdueEvent(Projector,PRO-101,6,Tom,tom@yopmail.com,0,2025-11-22T10:10:00Z,AllocationOverdueEvent,Some(AdditionalInfo(Some(inventor
[Kafka] Event received: AllocationOverdueEvent(Laptop,LAP-003,6,Tom,tom@yopmail.com,0,2025-11-22T10:10:00Z,AllocationOverdueEvent,Some(AdditionalInfo(Some(inventoryadm
[Kafka] Overdue: AllocationOverdueEvent(Laptop,LAP-003,6,Tom,tom@yopmail.com,0,2025-11-22T10:10:00Z,AllocationOverdueEvent,Some(AdditionalInfo(Some(inventoryadm
```