# National University of Technology



Computer Science Department Semester Spring 2024

**Project: Device Management System**

# Course: DBMS

**Submitted By:**

**Ali javed Afridi**

**Muhammad Souban Farooq**

**Zaid Nabi**

**Shazaib**

**Muhammad Rafay**

**Submitted To:**

**Mam Amna Ikram**

**CONTENT**

1. **Introduction**
2. **Table in SQL**
3. **Output**
4. **Used Html for frontend**
5. **Used flask to integrate the backend and frontend.**

# **Overview**

The Device Management System is designed to manage a variety of electronic devices and their associated information. It includes tables for device types, individual devices, their specifications, status updates, and usage history. This documentation covers the creation of the database schema, sample data insertion, and queries to retrieve relevant information from the database.

**Introduction:**

The Device Management System is a comprehensive solution for managing electronic devices within an organization. It aims to streamline the process of tracking and maintaining various types of devices, ranging from laptops and smartphones to printers and networking equipment. This system enables efficient management of device specifications, status updates, and usage history, ensuring that devices are utilized optimally and maintained properly.

The project involves setting up a relational database to store detailed information about each device, its specifications, current status, and usage history.

**Creating Database:**

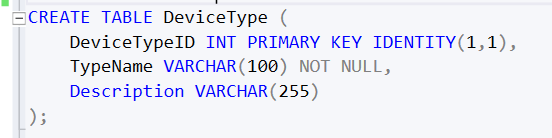
****

**Database Schema:**

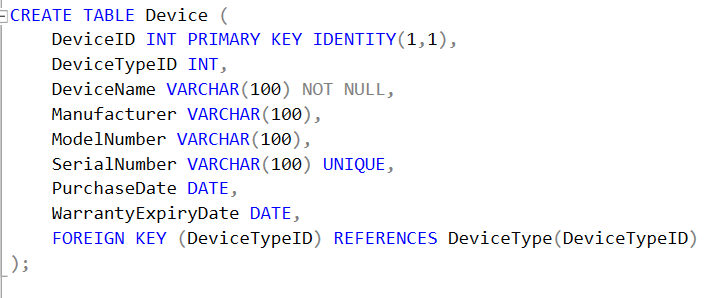
**Tables**:

1. **DeviceType**
2. **Device**
3. **DeviceSpecification**
4. **DeviceStatus**
5. **DeviceUsageHistory**

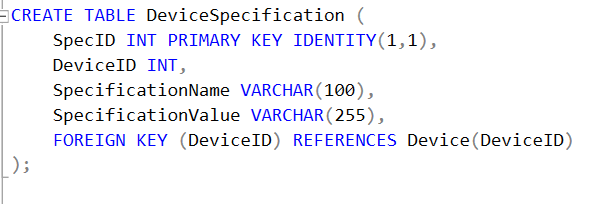
**DeviceType:**

****

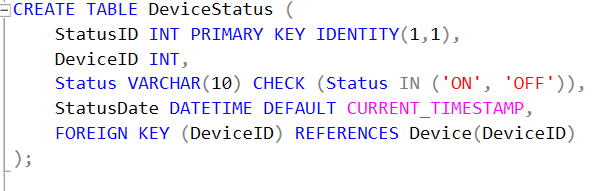
**Device:**

****

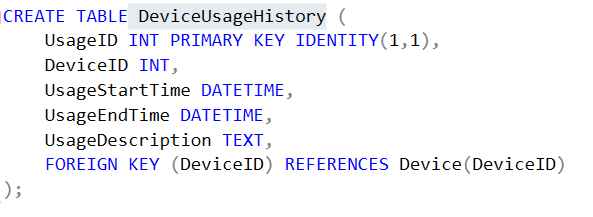
**DeviceSpecification:**

****

**DeviceStatus:**

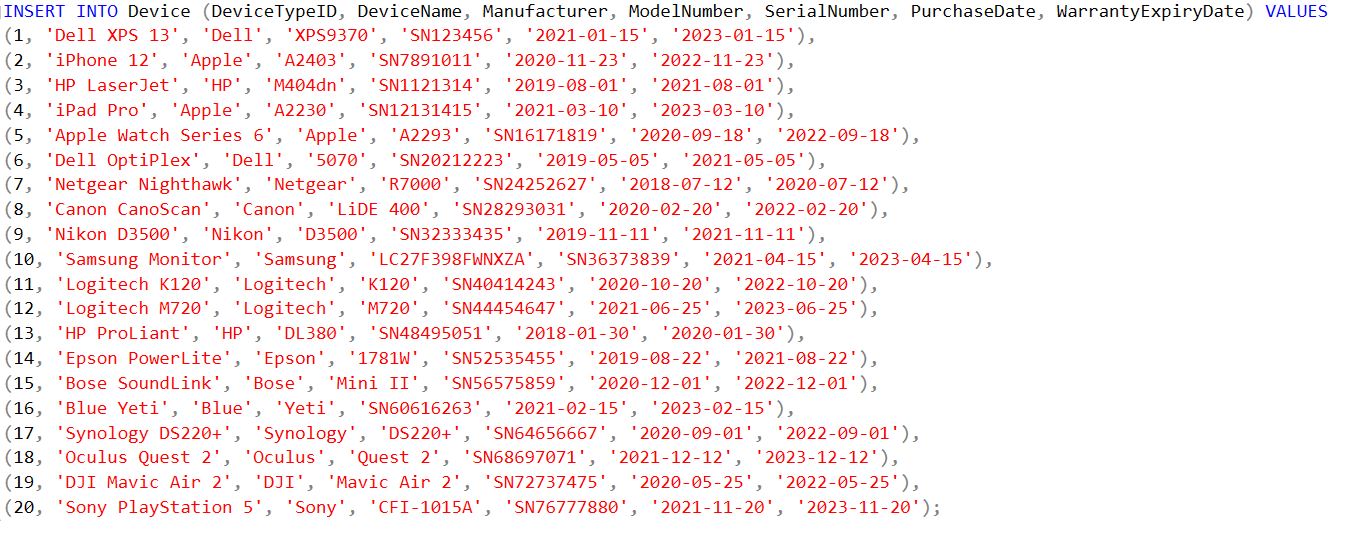
****

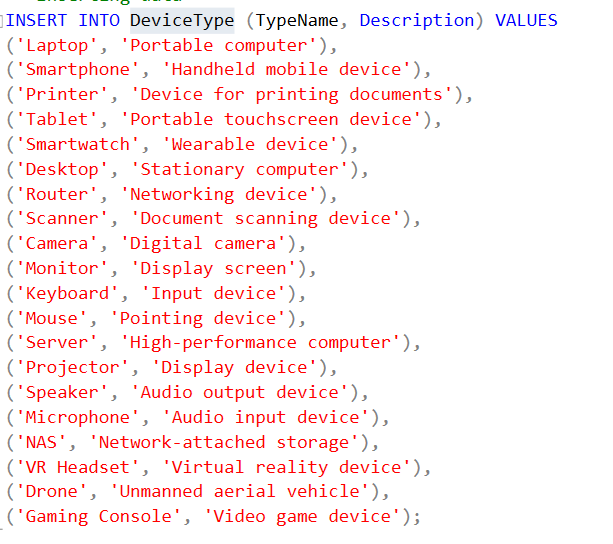
**DeviceUsageHistory:**

****

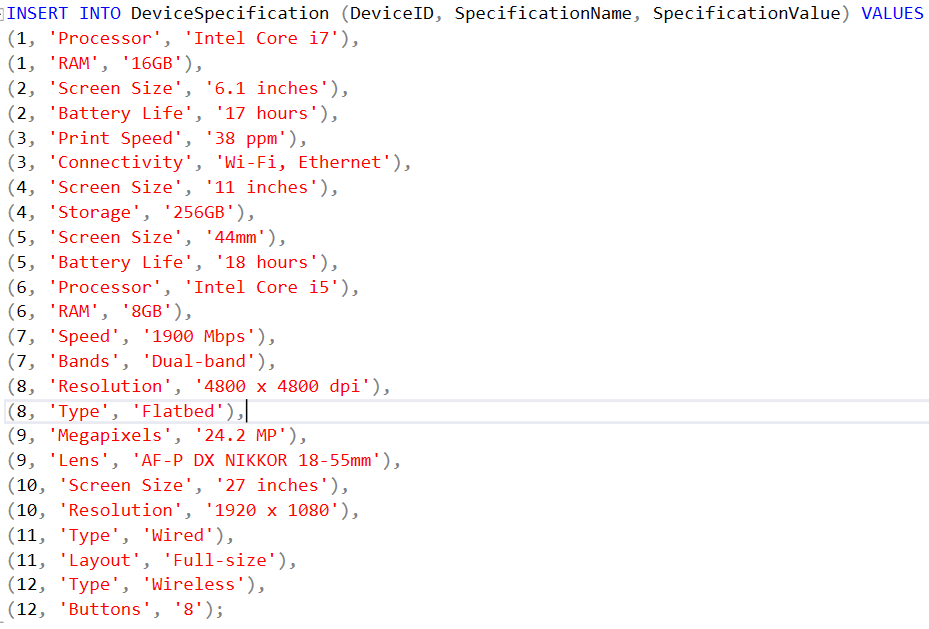
**Inserting Data in Tables:**

**Device (This Table is in Normalized Form )**

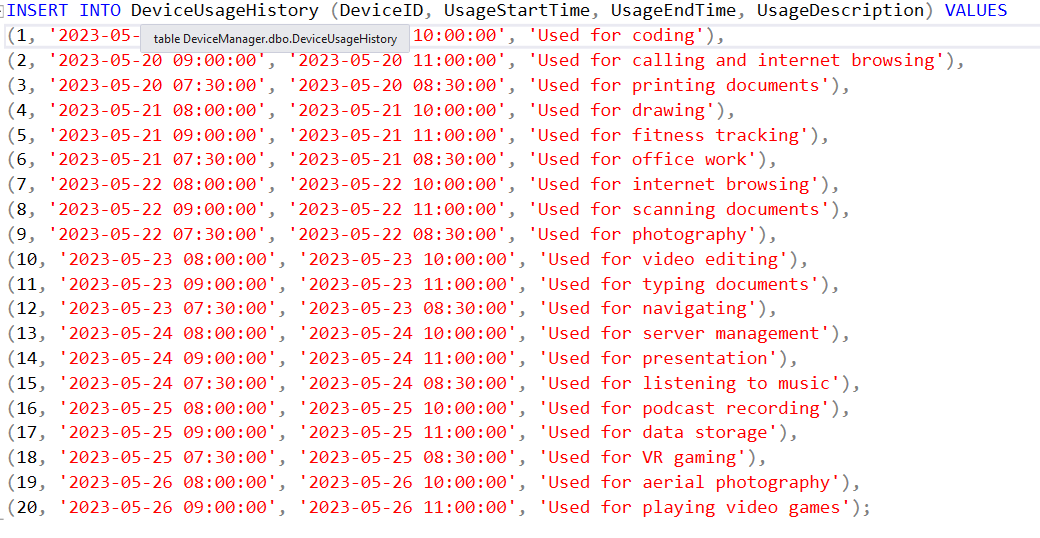
**DeviceType:**

****

**DeviceSpecification**

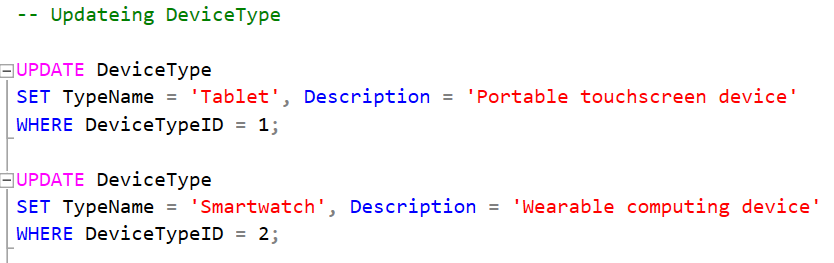
****

**DeviceUsageHistory:**

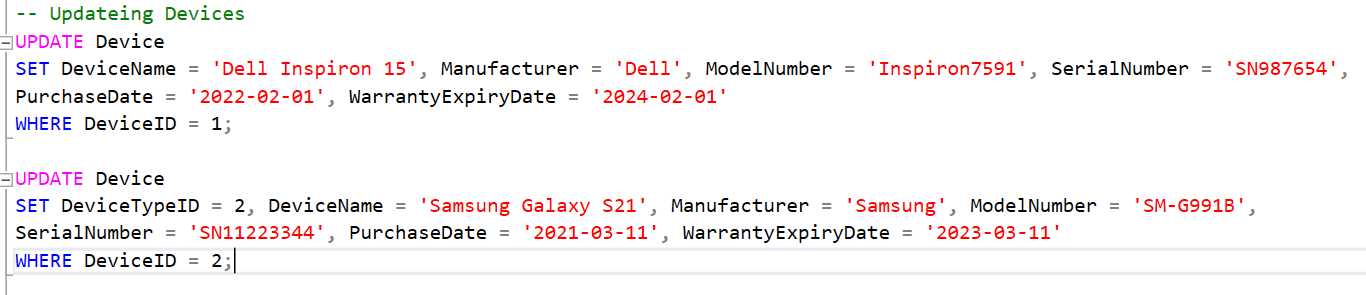
****

**Update Queries:**

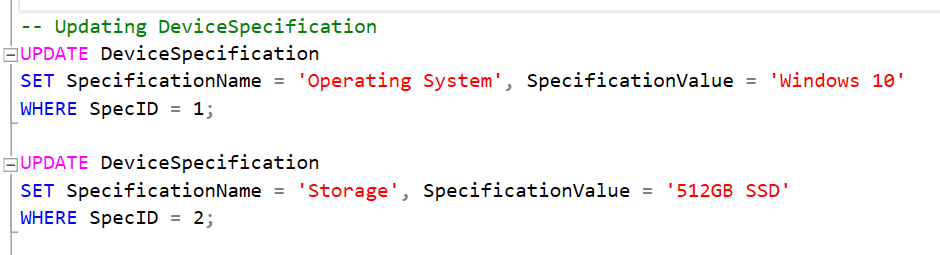
**DeviceTypes**

****

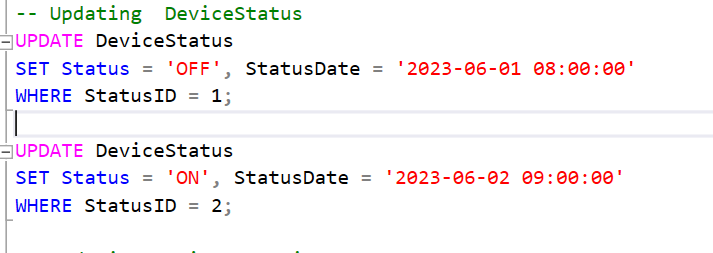
**Devices**

****

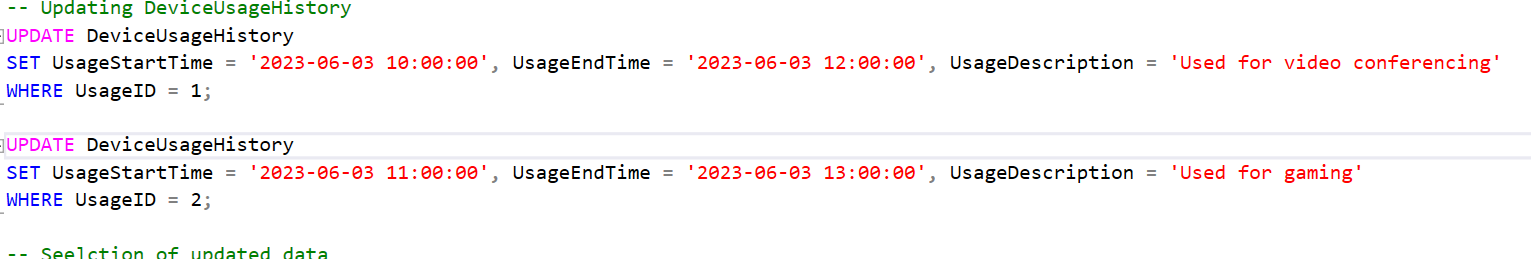
**DeviceSpecification**

****

**DeviceStatus**

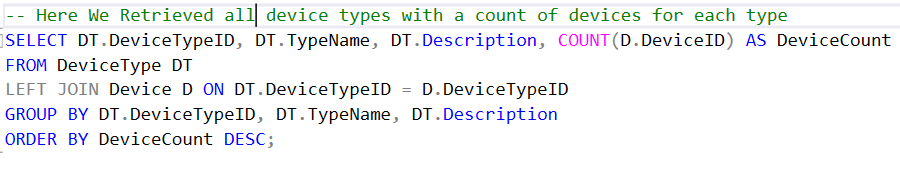
****

**DeviceUsageHistory**

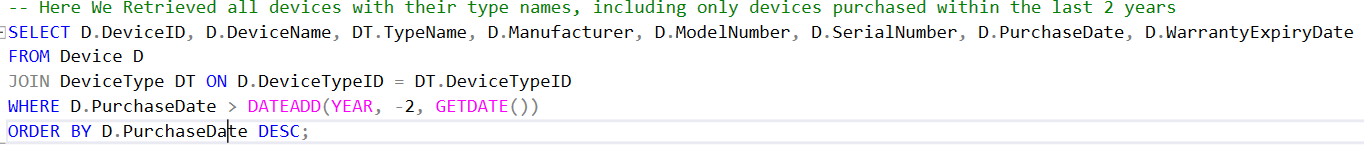
****

**Retrieving Data From Tables Using all the Queries:**

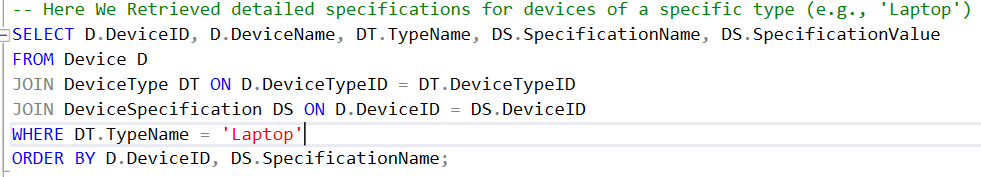
**🡪Here we retrieved all device types with a count of devices for each type**

****

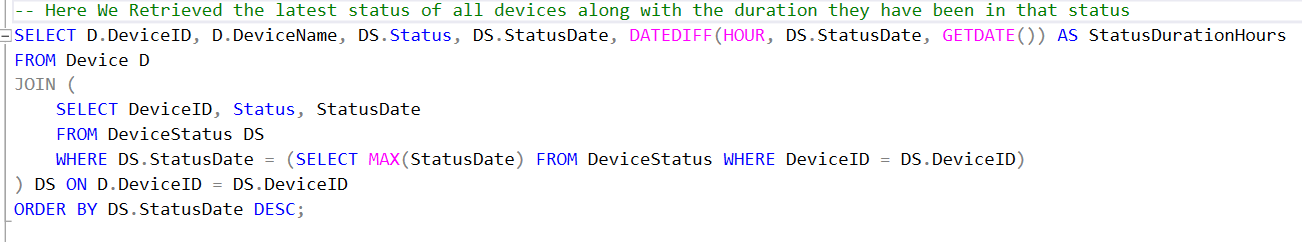
**🡪Here We Retrieved all devices with their type names, including only devices purchased within the last 2 years**

****

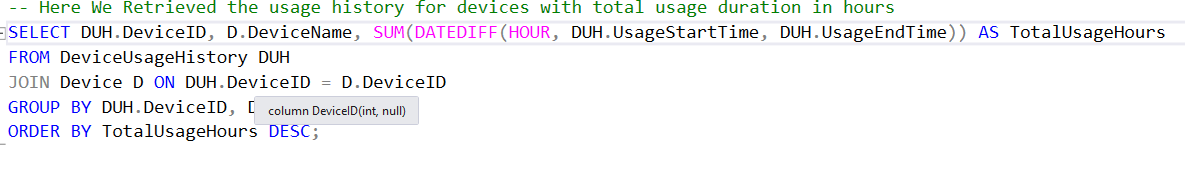
**🡪Here We Retrieved detailed specifications for devices of a specific type (e.g., 'Laptop')**

****

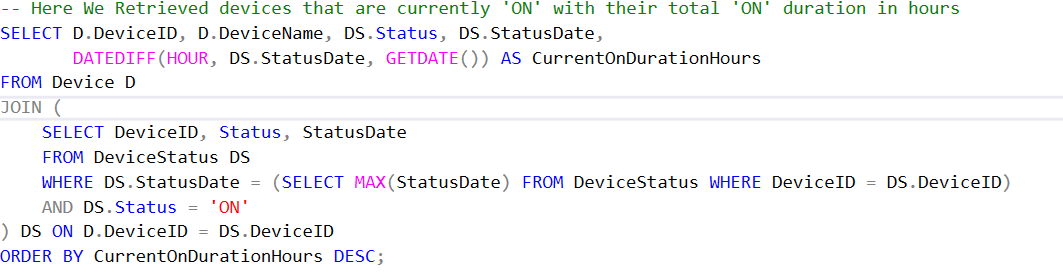
**🡪Here we retrieved the latest status of all devices along with the duration they have been in that status**

****

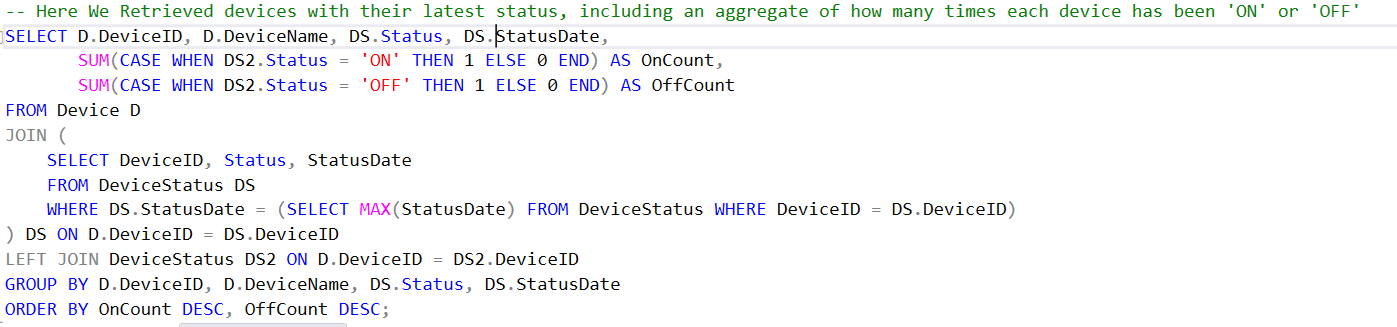
**🡪Here we retrieved the usage history for devices with total usage duration in hours**

****

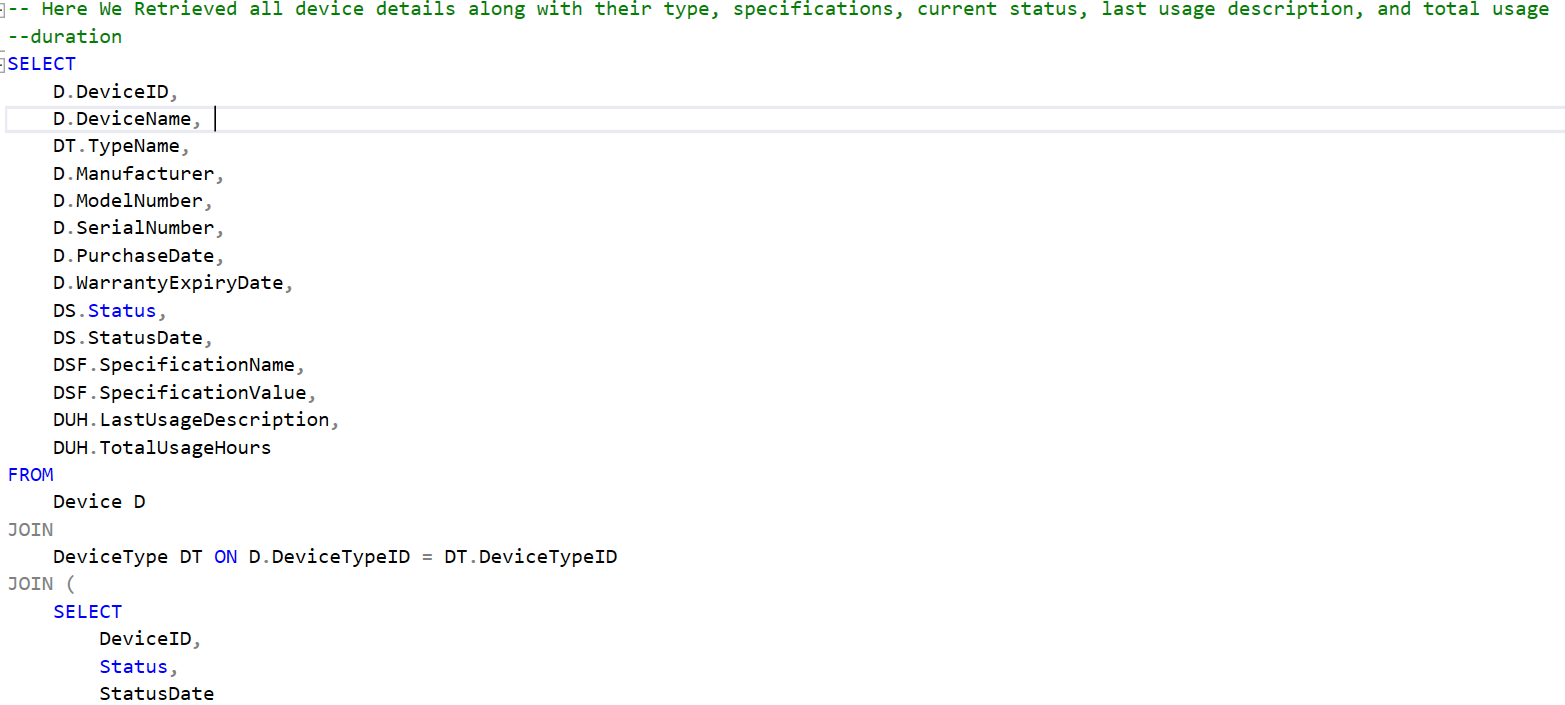
**🡪Here We Retrieved devices that are currently 'ON' with their total 'ON' duration in hours**

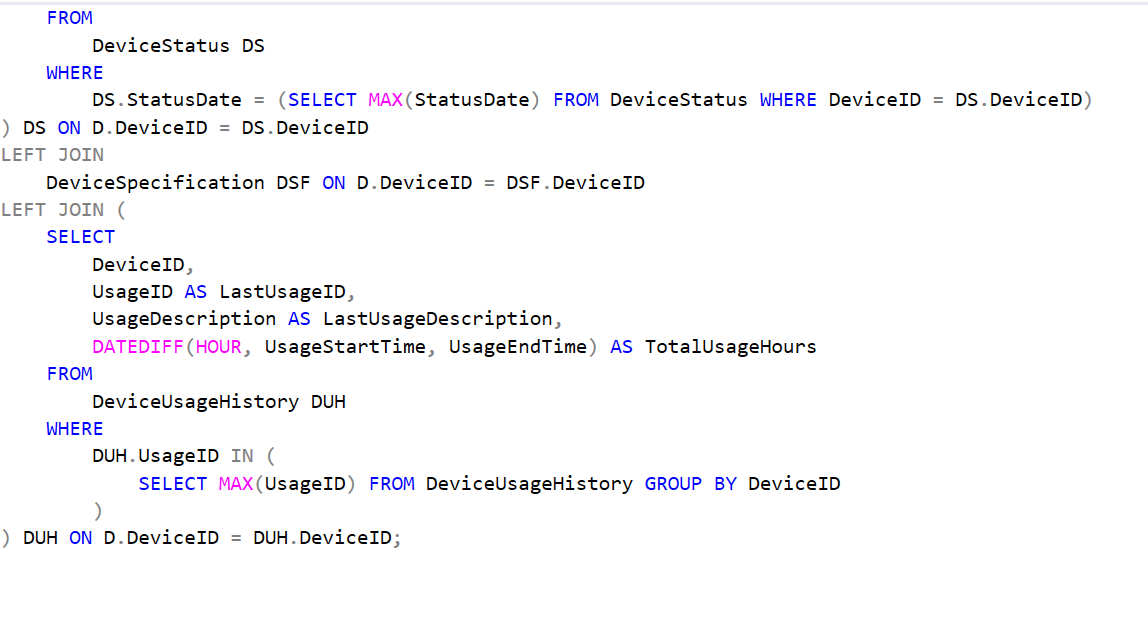
****

**🡪Here We Retrieved devices with their latest status, including an aggregate of how many times each device has been 'ON' or 'OFF'**

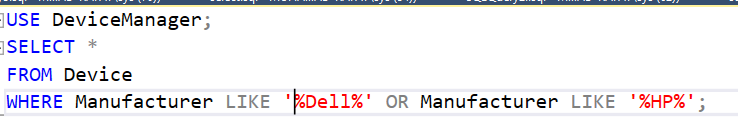
****

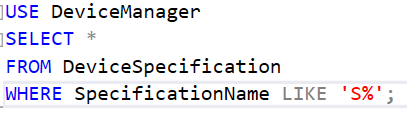
**🡪Here we retrieved all device details along with their type, specifications, current status, last usage description, and total usage duration**

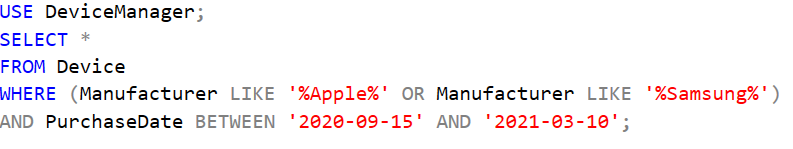
****

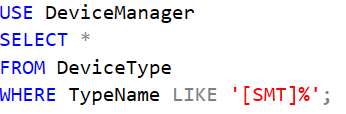
****

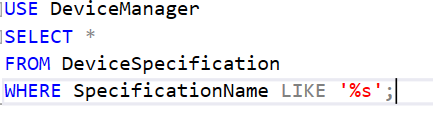
**Specific queries**

****

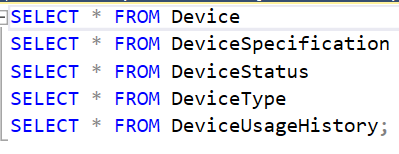
**+**

****

****

****

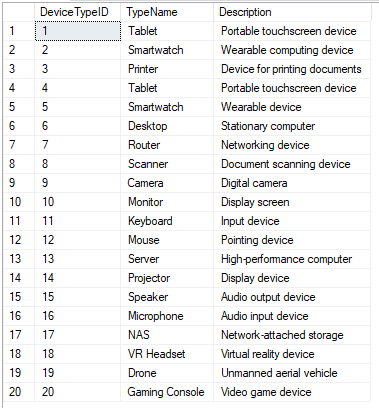
**Output:**

****

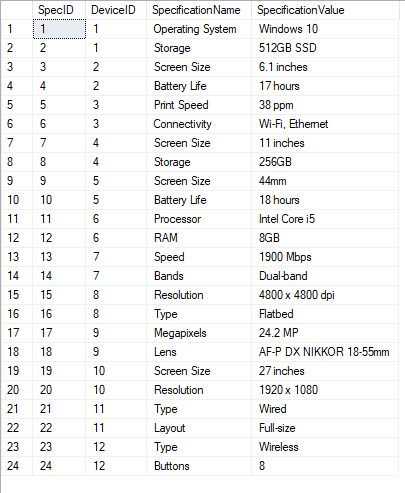
**Device**



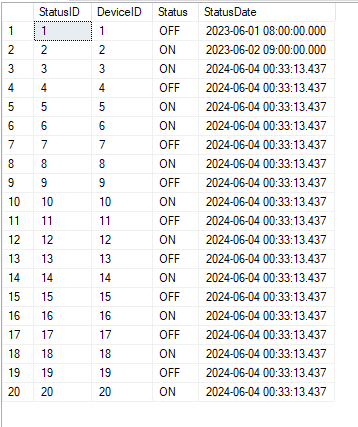
**DeviceType**

****

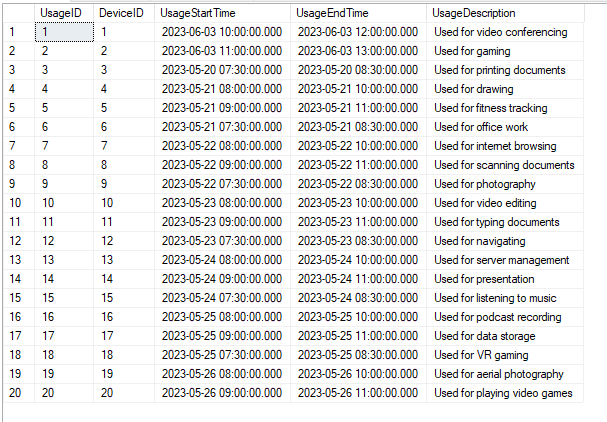
**Device Specification:**

****

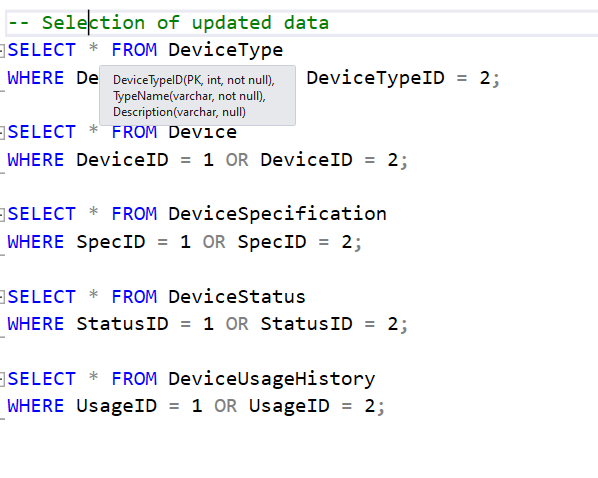
**DeviceStatus**

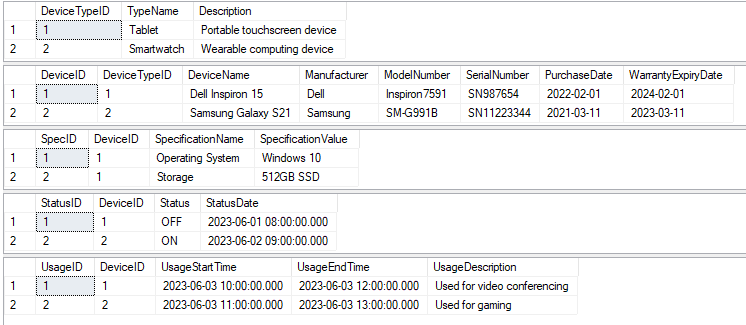
****

**DeviceUsageHistory**

****

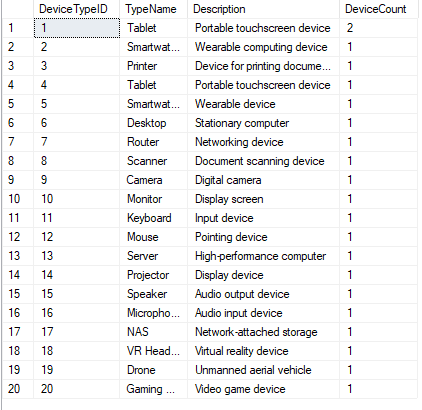
**Selecting the updated queries**

****

****

**Output of data retrieved from database**

**🡪Here we retrieved all device types with a count of devices for each type**

****

**🡪Here We Retrieved all devices with their type names, including only devices purchased within the last 2 years**

****

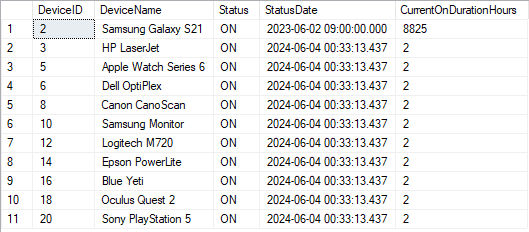
**🡪Here we retrieved the latest status of all devices along with the duration they have been in that status**

****

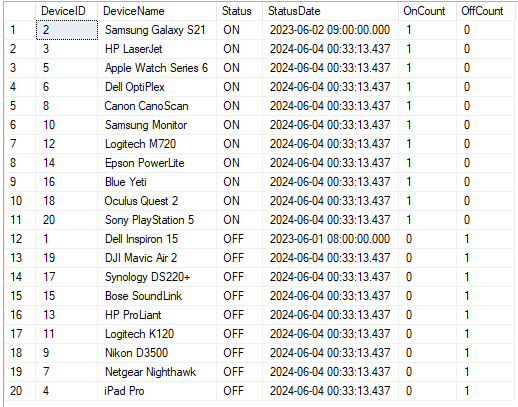
**🡪Here we retrieved the usage history for devices with total usage duration in hours**

****

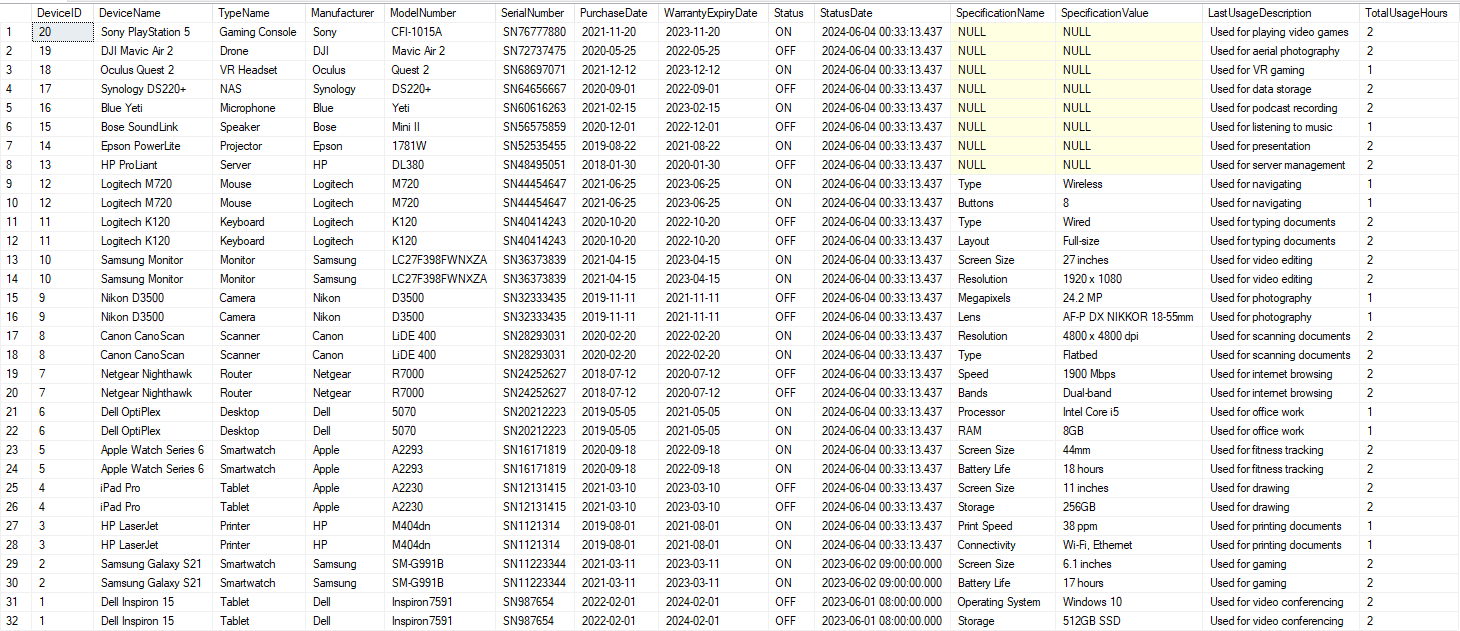
**🡪Here We Retrieved devices that are currently 'ON' with their total 'ON' duration in hours**

****

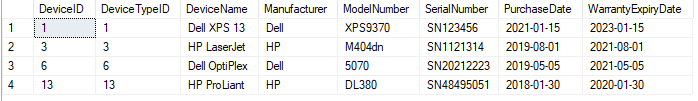
**🡪Here We Retrieved devices with their latest status, including an aggregate of how many times each device has been 'ON' or 'OFF'**

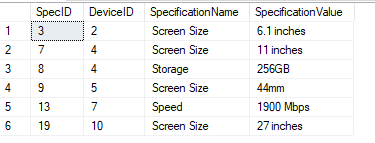
****

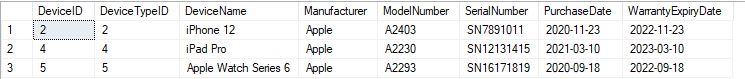
**🡪Here we retrieved all device details along with their type, specifications, current status, last usage description, and total usage duration**

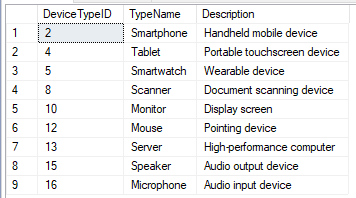
****

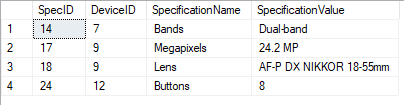
**Output of specific queries used:**

****

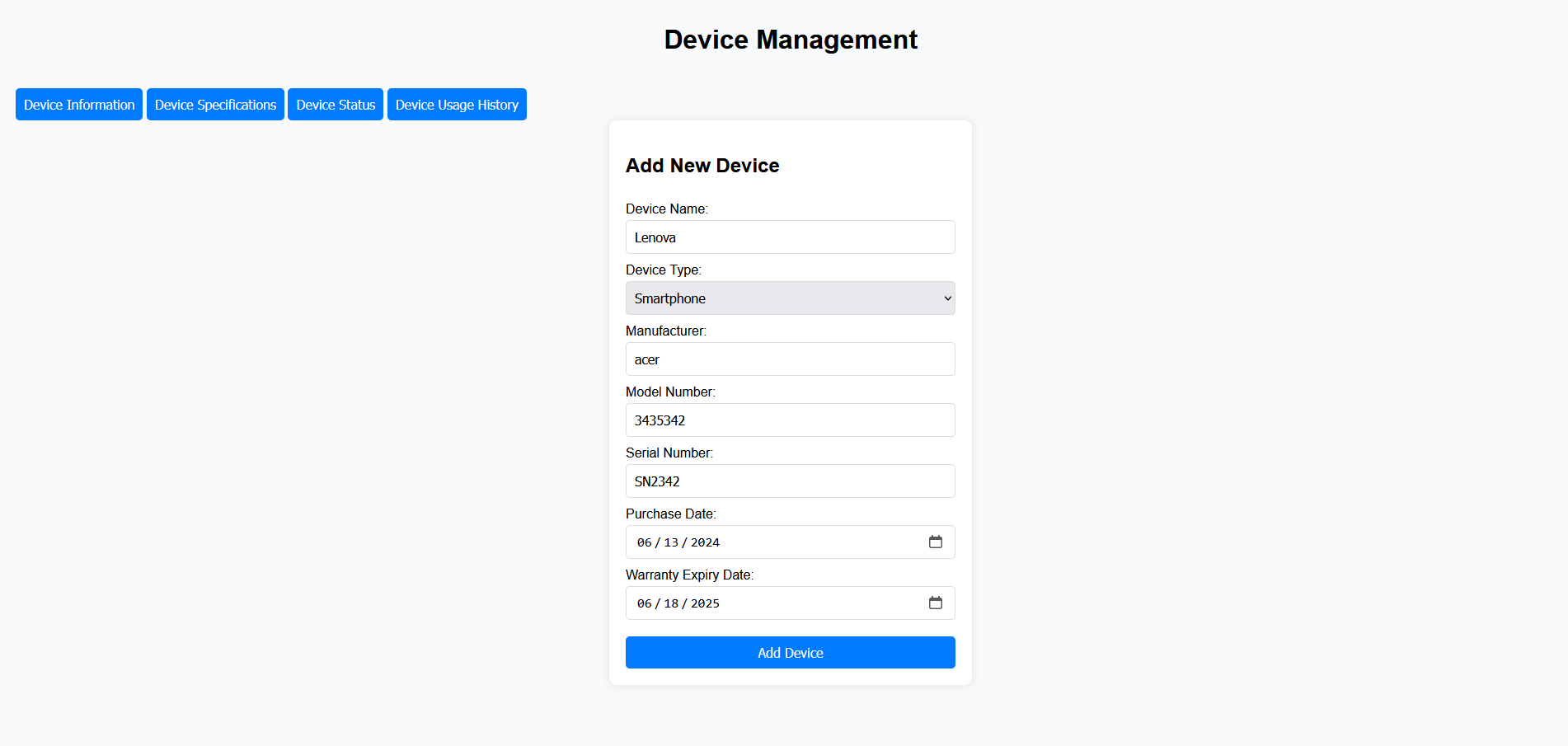
****

****

****

****

**Used Html for fronted**

****

In this page if we click the buttons the tables will be shown that we have created in database.

**Used flask to integrate the backend and frontend**

### **Flask Integration**

Flask is a lightweight web framework for Python that helps develop web applications. In the Device Management System, Flask connects the backend database with the frontend user interface, making it easy for users to interact with the system.

**Backend with Flask:**

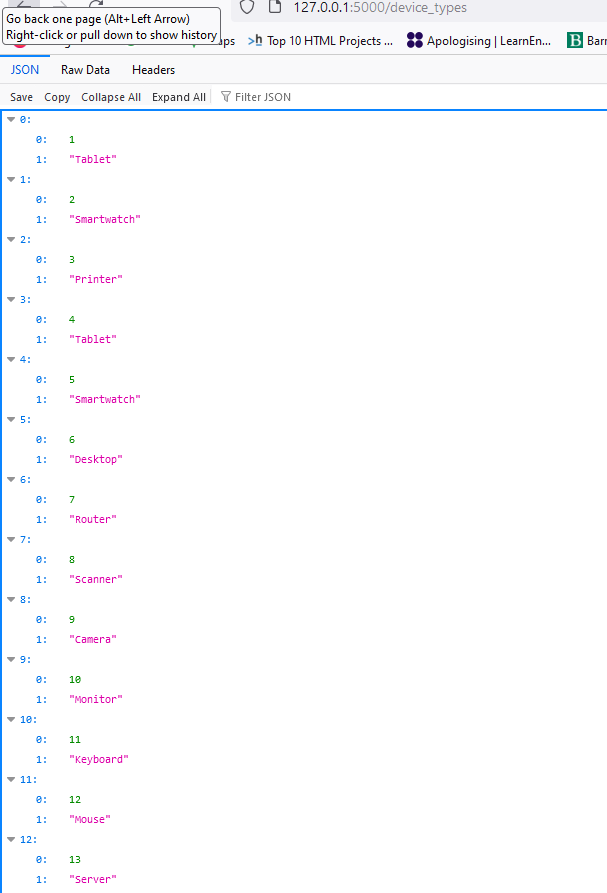
* **API Endpoints:** Flask creates routes (API endpoints) that the frontend can call to get or send data. This allows the frontend to add, view, update, or delete device information.
* **Database Interaction:** Flask works with SQLAlchemy (open-source SQL toolkit and object-relational mapper for the Python programming language ), a tool that helps interact with the database. It retrieves, inserts, updates, and deletes data efficiently.
* **Data Processing:** Flask handles the calculations and logic, like figuring out how long a device has been used or its current status.

**Frontend with Flask:**

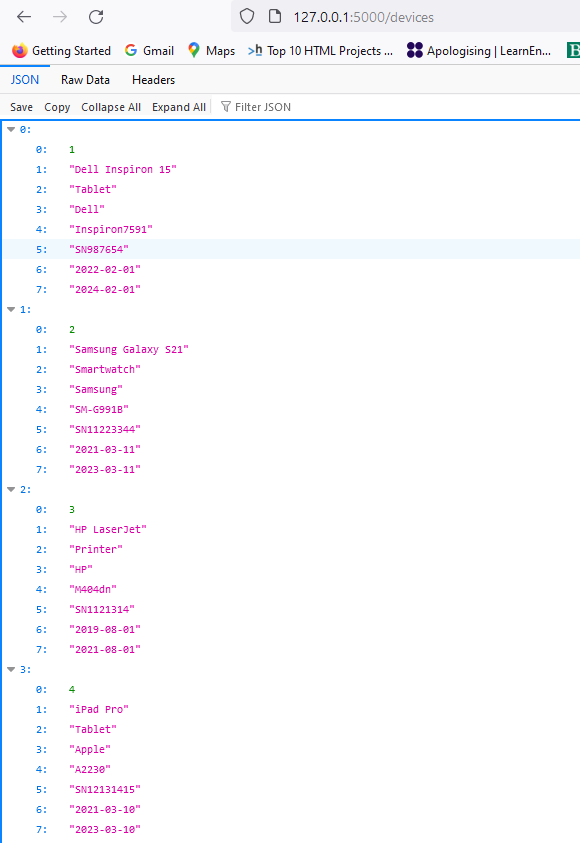
* **Templates:** Flask uses Jinja2 to create HTML pages. Data from the backend is inserted into these pages, so the information is always up-to-date.
* **Routing:** Flask defines URLs that users can visit. Each URL corresponds to a specific page in the application.
* **AJAX Calls:** AJAX is used for more interactive features. It allows the frontend to request data from the backend without reloading the entire page, making the app faster and smoother.

**In the Last we used to call the table data using API:**

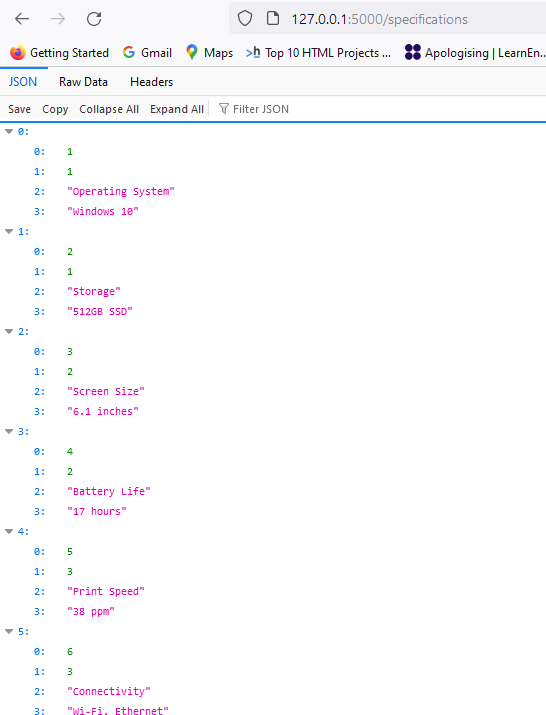
**/DeviceTypes**

****

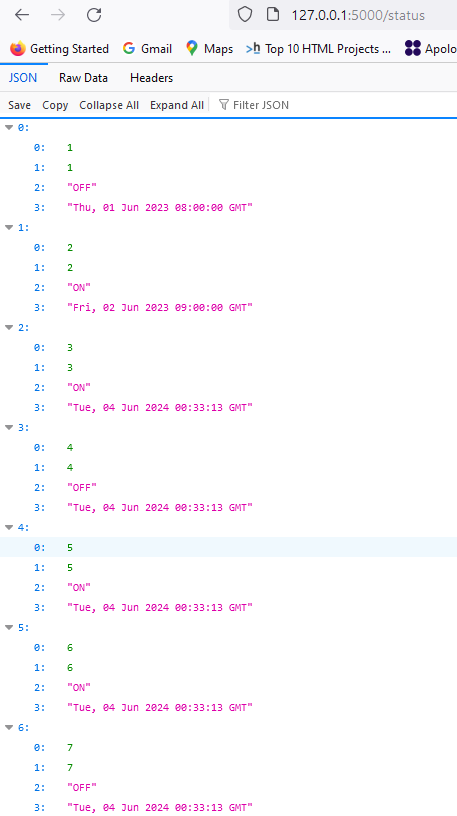
**/Devices**

****

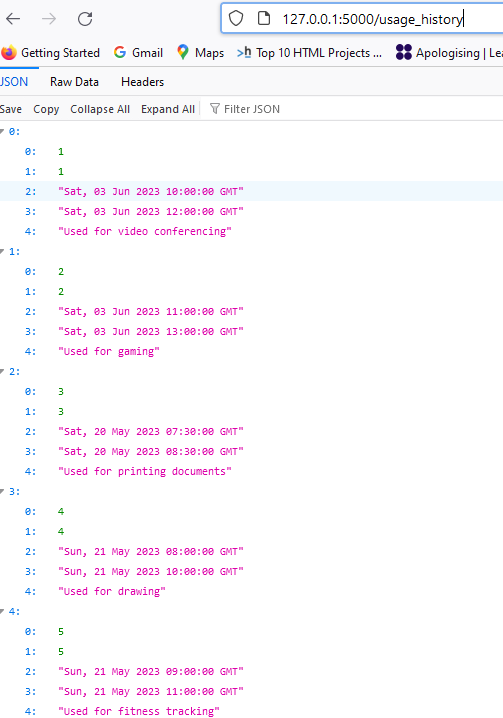
**/specifications:**

****

**/status**

****

**/usage\_history**

****