Project Report: Automating Graphics Card Sales Tracking

**Introduction**

This report outlines a project that aims to automate the process of tracking graphics card sales data using Bash scripts. The primary goal of this project is to facilitate the collection and recording of sales information for specific graphics card models, including RTX 3060, RTX 3070, RTX 3080, RTX 3090, and RX 6700, and store this data in an organized format. The project also involves setting up a local API server and using cURL for data retrieval.

**Project Objectives**

The key objectives of this project are as follows:

Automate the process of collecting sales data for specific graphics card models.

Store the sales data in a structured format within a text file called "sales.txt."

Implement a Bash script (exam.sh) to perform data retrieval and storage.

Create a Cron Job to execute the script periodically.

**Project Implementation**

Setting Up the API

The project begins by setting up an API server. The API is downloaded from a remote location using the wget command. The downloaded file is in .tar format, which is similar to a compressed ZIP file. To extract the content of the .tar file, the tar command is used with specific options: -x for extraction, -v for verbosity, and -f to specify the file name.

bash

Copy code

tar -xvf api.tar

Once extracted, the API script can be executed by providing execute permissions using chmod and launching it in the background using &.

bash

Copy code

chmod +x api

./api &

The API is now running locally at address 0.0.0.0 on port 5000. This allows for data retrieval and interaction with the API.

**Data Retrieval Using cURL**

To obtain sales data for specific graphics card models from the API, the cURL command is used. For example, to retrieve sales data for the RTX 3060 model:

**bash**

Copy code

curl "http://0.0.0.0:5000/rtx3060"

This command interacts with the API and fetches the sales information for the specified graphics card model.

**Bash Script and Data Storage**

A Bash script named exam.sh is created within a designated folder. The script is given execution permissions using the command chmod. This script is designed to retrieve sales data for multiple graphics card models and store this information in a structured format in the "sales.txt" file. The data format in "sales.txt" includes the date of scraping and sales figures for each graphics card model, separated by colons.

An example of the "sales.txt" format is as follows:

makefile

Copy code

Thu Apr 1 00:05:01 UTC 2021

rtx3060:18

rtx3070:20

rtx3080:20

rtx3090:2

rx6700:12

Thu Apr 1 00:06:01 UTC 2021

rtx3060:6

rtx3070:15

rtx3080:15

rtx3090:1

rx6700:15

To capture the current date and write it to the "sales.txt" file, the date command is used:

**bash**

Copy code

echo "$(date)" >> sales.txt

In compliance with project requirements, the script is expected to include a function and a loop (either for or while) for data retrieval and storage.

**Scheduling with Cron Job**

To automate the script execution, a Cron Job is created to run the Bash script every minute during the months of the training program. The Cron Job configuration is saved in a file called "cron.txt."

**Conclusion**

This project demonstrates the automation of graphics card sales data tracking using Bash scripts and APIs. It simplifies the process of data collection, storage, and retrieval, providing valuable sales information for specific graphics card models. The combination of Bash scripting, API interaction, and Cron Jobs offers an efficient solution for tracking and recording sales data.

Upon successful completion, users can rely on this project to maintain up-to-date records of graphics card sales trends and make informed decisions. Further improvements and contributions to the project can enhance its functionality and adaptability to evolving market demands.