Laurentian University
Université Laurentienne

BHARTI SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

Project Report

Course Code & Title: CPSC 5207 Introduction to Cloud Technologies

Faculty Advisor:

Jaspreet Bhatia
Fall, 2024

PriceTrack: Amazon Product Price Tracker

Group no: 3

Group Members and Details

Name	Student ID
Shazzad Sakim Sourav	0446233
Sayed Bin Taher	0456304
Ashif Ahmed	0438524
Faizan Raza	0446743
Amber Mallick	0441063

Table of Contents

Summary

The Amazon Product Price Tracking System is a robust platform designed to monitor price changes of Amazon products. It tracks historical price trends over 120 days and notifies users of price drops, allowing informed purchase decisions. The system leverages Google Cloud Platform (GCP) for hosting, scalable data storage, and periodic updates. Key features include a user-friendly interface for product tracking, automated data extraction using APIs, and advanced notification services.

1. Introduction

The rise of e-commerce platforms like Amazon has made it challenging for users to track price changes across multiple products. This project addresses this problem by offering a centralized system to monitor Amazon product prices, analyze historical trends, and provide price drop notifications, all through an intuitive web interface.

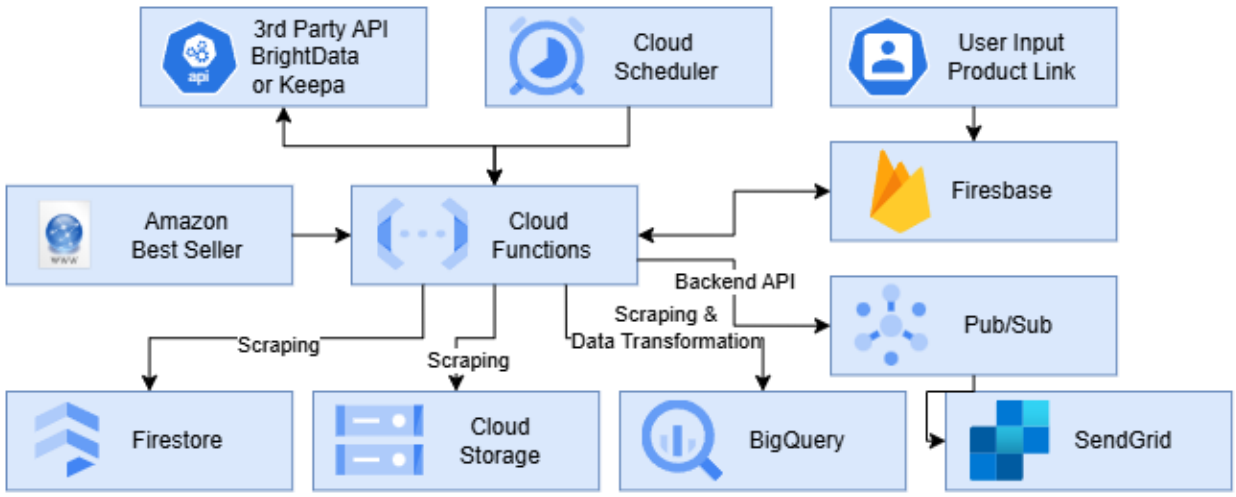
2. Objectives

- **Track Prices**: Monitor popular and user-specified Amazon products.
- **Analyze Trends**: Maintain and visualize a 120-day historical price record.
- **Notify Users**: Alert users when prices fall below predefined thresholds.
- **Scalability**: Design a cloud-based solution capable of handling large datasets and user interactions.

3. System Architecture

The architecture ensures scalability, reliability, and real-time performance through a modular design.

Architecture Pipeline



3.1 Data Extraction & Collection

- **Web Scraper Service (Cloud Functions)**:
    - Leverages **Bright Data** or **Keepa API** for scraping product prices and metadata.
    - Uses provided URLs of popular product sections for automated scraping.
    - Handles on-demand scraping for user-submitted product URLs.
- **Product Metadata Storage (Firestore)**:
    - Stores product details like names, categories, and image URLs.

3.2 Data Ingestion and Storage

- **BigQuery (Price Data Storage)**:
    - Stores price data for efficient retrieval and analysis.
    - Enables querying of price trends over a 120-day window.
- **Cloud Storage (Product Images)**:
    - Stores product images and large assets for fast and scalable access.
- **Firestore**:
    - Stores product metadata and user preferences for price alerts.

3.3 Data Processing & Scheduling

- **Cloud Scheduler (Price Update Trigger)**:
    - Triggers automated API calls for periodic updates to predefined popular product URLs and tracked items.
- **Cloud Functions (Data Transformation)**:
    - Processes raw data into trends, including average prices, min/max values, and changes over time.

3.4 User Interaction Platform

- **Frontend User Interface (Firebase Hosting)**:
    - The web interface was built using React and hosted via Firebase Hosting.
    - Allows users to search products, view trends, and subscribe to notifications.
- **Backend API (Cloud Functions)**:
    - Serves as the bridge between frontend requests and backend operations, like fetching trends or adding new products.

3.5 Price Drop Alerts & Notifications

- **Pub/Sub (Notification Trigger)**:
    - Monitors pricing conditions and triggers notifications for users.
- **Notification Services**:

    - Delivers alerts using **Firebase Cloud Messaging** (push notifications) or **SendGrid** (email).

4. Detailed Workflow

Product Identification:

- Scrape provided URLs of popular product sections using Bright Data/Keepa.
- Users can submit individual product URLs to add to the tracking queue.
- New products are added to Firestore for tracking.

Price Extraction & Storage:

- Cloud Scheduler triggers periodic API calls to update prices for tracked products and those from predefined URLs.
- Data is ingested into BigQuery for storage and analysis.

Frontend Interaction:

- Users access a web interface to:
    - Search for products and view trends.
    - Add new URLs for tracking.
    - Manage price drop alerts.

Price Drop Notification:

- Pub/Sub monitors price thresholds and triggers notifications via Firebase Cloud Messaging or SendGrid.

5. Scalability & Monitoring

- **Cloud Monitoring** tracks API performance, scraping operations, and notification delivery.
- Auto-scaling **Cloud Functions** handles dynamic workloads from user actions and periodic updates.

6. Data Storage and Management Architecture

6.1 Firestore Schema

Firestore will be used to store metadata and user-related data. Below is the schema:

Collections and Documents:

1. **Products Collection**

- Stores metadata about the products being tracked.

Document ID: Unique Product ID (e.g., ASIN or generated UUID)

Field	Type	Description
productName	String	Name of the product
category	String	Category of the product
imageUrl	String	URL of the product image
url	String	Product URL
lastUpdated	Timestamp	Last update timestamp
status	String	Status (active, inactive, etc.)

2. **Users Collection**
   - Stores user information and their subscriptions to price alerts.

Document ID: User ID (UUID or email)

Field	Type	Description
email	String	User's email
preferences	Map	User-specific settings (e.g., notifications)
subscriptions	Array<String>	List of product IDs the user is subscribed to

3. **Subscriptions Collection**
   - Links users to product price alerts.

Document ID: Auto-generated

Field	Type	Description
userId	String	ID of the subscribed user
productId	String	ID of the subscribed product
priceThreshold	Float	User-defined price threshold

6.1.1. Products Collection (Firestore)

Document ID	Product Name	Category	Image URL	URL	Last Updated	Status
B08N5WRWNW	Sony WH-1000XM4 Wireless Headphones	Electronics	https://example.com/images/sony.jpg	https://www.amazon.com/dp/B08N5WRWNW	2024-11-25T12:00:00Z	active
B09G3HRPZG	Apple AirPods Pro (2nd Gen)	Electronics	https://example.com/images/airpods.jpg	https://www.amazon.com/dp/B09G3HRPZG	2024-11-20T10:00:00Z	active

B07FZ8S74R	Instant Pot Duo 7-in-1	Home & Kitchen	https://example.com/images/instantpot.jpg	https://www.amazon.com/dp/B07FZ8S74R	2024-11-21T11:00:00Z	active
B08164VTWH	Samsung Galaxy Watch 4	Electronics	https://example.com/images/samsung.jpg	https://www.amazon.com/dp/B08164VTWH	2024-11-22T08:30:00Z	active
B08P2DKZ1D	Lenovo IdeaPad 3	Computers	https://example.com/images/lenovo.jpg	https://www.amazon.com/dp/B08P2DKZ1D	2024-11-23T09:00:00Z	active
B07WC5VNGL	Logitech MX Master 3	Accessories	https://example.com/images/logitech.jpg	https://www.amazon.com/dp/B07WC5VNGL	2024-11-24T14:00:00Z	active
B082LZ1NXJ	Ninja Foodi 9-in-1	Home & Kitchen	https://example.com/images/ninja.jpg	https://www.amazon.com/dp/B082LZ1NXJ	2024-11-21T13:00:00Z	active
B07C1PWSTY	Amazon Echo Dot (4th Gen)	Electronics	https://example.com/images/echo.jpg	https://www.amazon.com/dp/B07C1PWSTY	2024-11-20T15:00:00Z	active
B097QKT4F2	Fitbit Versa 3	Electronics	https://example.com/images/fitbit.jpg	https://www.amazon.com/dp/B097QKT4F2	2024-11-18T17:00:00Z	active
B08FC5L3RG	ASUS ROG Zephyrus Gaming Laptop	Computers	https://example.com/images/asus.jpg	https://www.amazon.com/dp/B08FC5L3RG	2024-11-19T16:00:00Z	active

6.1.2. Users Collection (Firestore)

Document ID	Email	Preferences	Subscriptions
user1	johndoe@example.com	{ "notifications": true }	["B08N5WRWNW", "B09G3HRPZG", "B08164VTWH"]
user2	janedoe@example.com	{ "notifications": false }	["B07WC5VNGL", "B082LZ1NXJ", "B08FC5L3RG"]
user3	sam@example.com	{ "notifications": true }	["B097QKT4F2", "B07C1PWSTY", "B08P2DKZ1D"]
user4	alice@example.com	{ "notifications": false }	["B08N5WRWNW", "B08164VTWH", "B07FZ8S74R"]
user5	bob@example.com	{ "notifications": true }	["B07C1PWSTY", "B08P2DKZ1D", "B08N5WRWNW"]
user6	tom@example.com	{ "notifications": true }	["B09G3HRPZG", "B082LZ1NXJ", "B08FC5L3RG"]
user7	mary@example.com	{ "notifications": false }	["B097QKT4F2", "B08P2DKZ1D", "B08FC5L3RG"]
user8	anna@example.com	{ "notifications": true }	["B07WC5VNGL", "B08164VTWH", "B09G3HRPZG"]

user9	mike@example.com	{ "notifications": true }	["B07FZ8S74R", "B07WC5VNGL", "B08N5WRWNW"]
user10	emily@example.com	{ "notifications": false }	["B082LZ1NXJ", "B07C1PWSTY", "B097QKT4F2"]

6.1.3. Subscriptions Collection (Firestore)

Document ID	User ID	Product ID	Price Threshold
sub1	user1	B08N5WRWNW	300.00
sub2	user1	B09G3HRPZG	250.00
sub3	user2	B07WC5VNGL	80.00
sub4	user3	B097QKT4F2	200.00
sub5	user3	B08P2DKZ1D	500.00
sub6	user4	B07FZ8S74R	100.00
sub7	user5	B08P2DKZ1D	450.00
sub8	user6	B082LZ1NXJ	150.00
sub9	user7	B08FC5L3RG	1200.00
sub10	user8	B08164VTWH	180.00

Products Collection (products.json)


```
[
    {
      "productId": "B08N5WRWNW",
      "productName": "Sony WH-1000XM4 Wireless Headphones",
      "category": "Electronics",
      "imageUrl": "https://example.com/images/sony.jpg",
      "url": "https://www.amazon.com/dp/B08N5WRWNW",
      "lastUpdated": "2024-11-25T12:00:00Z",
      "status": "active"
    },
    {
      "productId": "B09G3HRPZG",
      "productName": "Apple AirPods Pro (2nd Gen)",
      "category": "Electronics",
      "imageUrl": "https://example.com/images/airpods.jpg",
      "url": "https://www.amazon.com/dp/B09G3HRPZG",
      "lastUpdated": "2024-11-20T10:00:00Z",
      "status": "active"
    },
    ...
  ]
```

Users Collection (users.json)

```
[
    {
      "userId": "user1",
      "email": "johndoe@example.com",
      "preferences": { "notifications": true },
      "subscriptions": ["B08N5WRWNW", "B09G3HRPZG", "B08164VTWH"]
    },
    {
      "userId": "user2",
      "email": "janedoe@example.com",
      "preferences": { "notifications": false },
      "subscriptions": ["B07WC5VNGL", "B082LZ1NXJ", "B08FC5L3RG"]
    },
    ...
  ]
```

Subscriptions Collection (subscriptions.json)

```
[
    {
      "subscriptionId": "sub1",
      "userId": "user1",
      "productId": "B08N5WRWNW",
      "priceThreshold": 300.00
    },
    {
      "subscriptionId": "sub2",
      "userId": "user1",
      "productId": "B09G3HRPZG",
      "priceThreshold": 250.00
    },
    ...
  ]
```

6.2 Cloud Storage Setup

Bucket Name: product-images

Folder Structure

Each product gets its own folder, and all images associated with that product are stored in that folder.

```
gs://product-images/
  ├── products/
      ├── B08N5WRWNW/
          ├── image1.jpg
          ├── image2.jpg
          ├── image3.jpg
          ├── metadata.json
      ├── B09G3HRPZG/
          ├── image1.jpg
          ├── image2.jpg
          ├── image3.jpg
          ├── metadata.json
      ├── B07FZ8S74R/
          ├── image1.jpg
          ├── image2.jpg
          ├── image3.jpg
          ├── metadata.json
      ...
```

Content of Each Folder

Images

- Each folder will contain the product's images (image1.jpg, image2.jpg, image3.jpg, etc.).
- Images are stored in the same folder as their corresponding metadata.

Metadata File

- **Filename**: metadata.json
- **Purpose**: Stores metadata for all images in the product's folder.

Metadata File (metadata.json)


```
{
    "productId": "B08N5WRWNW",
    "images": [
      {
        "fileName": "image1.jpg",
        "path": "gs://product-images/products/B08N5WRWNW/image1.jpg",
        "uploadTime": "2024-11-25T12:00:00Z",
        "contentType": "image/jpeg",
        "size": "124 KB"
      },
      {
        "fileName": "image2.jpg",
        "path": "gs://product-images/products/B08N5WRWNW/image2.jpg",
        "uploadTime": "2024-11-25T12:05:00Z",
        "contentType": "image/jpeg",
        "size": "150 KB"
      },
      {
        "fileName": "image3.jpg",
        "path": "gs://product-images/products/B08N5WRWNW/image3.jpg",
        "uploadTime": "2024-11-25T12:10:00Z",
        "contentType": "image/jpeg",
        "size": "130 KB"
      }
    ]
  }
```


Advantages of This Structure

1. **Organized by Product**:
    - All images and metadata for a specific product are grouped in a single folder.
2. **Metadata Accessibility**:
    - Metadata stored as a JSON file allows easy programmatic access and updates.
3. **Scalable**:
    - Supports thousands of products, with each having its own subfolder.
4. **Decoupled Storage**:
    - Cloud Storage serves as the single source for both images and metadata, reducing dependency on external systems like Firestore.

6.3 BigQuery Schema for Price Analytics

Dataset Name: price_tracking

Table Name: product_prices

Column Name	Type	Description
productId	STRING	Unique ID of the product (matches Firestore).
date	DATE	Date of the price record.
price	FLOAT	Price of the product.
currency	STRING	Currency of the price (e.g., USD).
priceType	STRING	Type of price (regular, sale, etc.).
timestamp	TIMESTAMP	Time when the price was recorded.

Sample Data for product_prices

Below is the data for 10 products, each with 2-3 price records to reflect changes over time.

productId	date	price	currency	priceType	timestamp
B08N5WRWNW	2024-11-01	299.99	USD	regular	2024-11-01T10:00:00Z
B08N5WRWNW	2024-11-10	279.99	USD	sale	2024-11-10T11:00:00Z
B09G3HRPZG	2024-11-05	249.99	USD	regular	2024-11-05T09:00:00Z
B09G3HRPZG	2024-11-12	229.99	USD	sale	2024-11-12T08:00:00Z
B07FZ8S74R	2024-11-02	99.99	USD	regular	2024-11-02T12:00:00Z
B07FZ8S74R	2024-11-08	89.99	USD	sale	2024-11-08T08:00:00Z
B08164VTWH	2024-11-03	199.99	USD	regular	2024-11-03T07:00:00Z
B08164VTWH	2024-11-15	179.99	USD	sale	2024-11-15T10:30:00Z
B08P2DKZ1D	2024-11-04	499.99	USD	regular	2024-11-04T08:30:00Z
B08P2DKZ1D	2024-11-18	449.99	USD	sale	2024-11-18T09:00:00Z
B07WC5VNGL	2024-11-06	99.99	USD	regular	2024-11-06T14:00:00Z
B07WC5VNGL	2024-11-14	89.99	USD	sale	2024-11-14T11:30:00Z
B082LZ1NXJ	2024-11-07	229.99	USD	regular	2024-11-07T13:00:00Z
B082LZ1NXJ	2024-11-16	209.99	USD	sale	2024-11-16T15:00:00Z
B07C1PWSTY	2024-11-09	49.99	USD	regular	2024-11-09T12:00:00Z

B07C1PWSTY	2024-11-20	44.99	USD	sale	2024-11-20T16:00:00Z
B097QKT4F2	2024-11-11	199.99	USD	regular	2024-11-11T17:00:00Z
B097QKT4F2	2024-11-19	179.99	USD	sale	2024-11-19T18:00:00Z
B08FC5L3RG	2024-11-13	1499.99	USD	regular	2024-11-13T16:00:00Z
B08FC5L3RG	2024-11-21	1399.99	USD	sale	2024-11-21T18:00:00Z

Explanation

- Each productId links to a product in Firestore.
- The date field helps with time-based trend analysis.
- The priceType differentiates between regular prices and discounts.
- The timestamp adds granularity, showing exactly when the price was recorded.

Why This Schema

1. **Minimal Redundancy**:
    - Metadata like productName or category is fetched from Firestore when needed, avoiding duplication.
2. **Focused Analytics**:
    - BigQuery stores only price-related data, which is the primary focus of this project.
3. **Scalability**:
    - BigQuery's ability to handle large datasets ensures smooth scaling for hundreds or thousands of products.

7. IAM Policy for Collaborative Team Management

Roles and Responsibilities:

Role	**Responsibilities**	**Suggested GCP Role**
Project Admin	Oversees the project, resources, billing, IAM.	roles/owner or roles/editor
Developer	Develops/deploys Cloud Functions and APIs.	roles/cloudfunctions.developer
Data Engineer	Manages BigQuery, Firestore, and Cloud Storage.	roles/bigquery.dataEditor, etc.

DevOps Specialist	Manages pipelines, logging, monitoring.	roles/logging.admin, roles/monitoring.editor
Security Specialist	Manages API keys and secrets, ensures resource security.	roles/iam.securityReviewer

9. Challenges and Mitigations

Popular Product Scraping:

- Use the provided URLs of popular product sections to define scraping targets.
- Paid APIs (Bright Data or Keepa) bypass Amazon's rate limits and CAPTCHAs.

Anti-Scraping Measures:

- Bright Data manages proxy rotation to avoid detection.
- Keepa API provides compliant, structured data for Amazon products.

Data Privacy:

- Ensure GDPR and other data privacy compliance standards.

8. Cost Estimation and Optimization

Assuming 10,000 users will use and 100,000 products will be stored:

GCP Costs:

Service	**Cost Driver**	**Estimated Cost**
Cloud Functions	Triggers for scrapers, APIs	$150–300/month
Firestore	Metadata for 100K products, frequent queries	$100–200/month
BigQuery	Storage and queries for 100K products	$100–250/month
Cloud Storage	10GB images, 150K reads	$30–50/month
Pub/Sub	Notifications for 10K users	$5–10/month
Firebase Hosting	Web UI hosting	$10–20/month

SendGrid	Email notifications	$10–50/month

Paid API Costs:

Service	**Pricing Model**	**Estimated Monthly Cost**
Bright Data	$25/GB of traffic (10GB usage estimated)	$250–500/month
Keepa API	€0.01/request (100K requests estimated)	€1,000 (~$1,080/month)

Total Estimated Costs:

Phase	**Estimated Monthly Cost**
Development	~$500–1,000
Production	~$1,400–2,300

9. Why we chose GCP over other cloud providers and on-premise solution

Criteria	**Google Cloud Platform (GCP)**	**AWS**	**Azure**	**On-Premise**
Scalability	Auto-scaling services like **Cloud Functions** and **BigQuery** seamlessly handle fluctuating workloads.	Auto-scaling available but may require more manual configuration, e.g., Lambda scaling policies.	Good scaling, but some services (e.g., Synapse Analytics) are less intuitive.	Limited scalability; requires upfront hardware investment for peak load capacity.
Cost	**Pay-as-you-go**, sustained-use discounts, competitive pricing, and efficient cost monitoring tools.	Often more expensive for managed services like DynamoDB and Lambda.	Cost-effecti ve for hybrid cloud but can become costly for	High upfront costs for hardware, maintenance, power,

			big data services.	cooling, and IT staff.
Big Data Tools	**BigQuery** excels in handling large-scale data analytics with built-in machine learning capabilities.	Athena and Redshift are robust but require more manual setup.	Synapse Analytics is less seamless for real-time queries compared to BigQuery.	Requires building a custom analytics solution, increasing complexity and costs.
Ease of Use	Intuitive services like **Firestore** and **Firebase Hosting** enable quick development and deployment.	Steeper learning curve for equivalent services like DynamoDB and S3.	Strong for Microsoft-based ecosystems but less user-friendly for non-Microsoft tools.	Requires a dedicated team to manage servers, databases, and updates.
Global Availability	Extensive global infrastructure ensures low latency and high availability for users worldwide.	Widely available but often higher latency for certain regions compared to GCP.	Strong global reach but slightly limited compared to AWS.	Localized servers mean latency and availability issues for a global user base.
Integration	Seamless integration with **Firebase**, **Firestore**, and APIs (e.g., Bright Data, Keepa) to build user-friendly systems.	Integration is possible but less straightforward without additional configuration.	Good integration with Microsoft services but limited third-party API support.	Limited integration capabilities; requires custom setups for APIs and third-party services.
Security	Built-in encryption at rest and in transit, **OAuth**, and compliance with GDPR and other standards.	Strong security, but AWS's complex service	Robust security with enterprise-gr	Security depends on internal IT expertise and

		structure makes configuration error-prone.	ade compliance options.	infrastructure investment.
Operational Overhead	Fully managed services reduce maintenance and operational costs significantly.	Managed services available but may require more custom configurations.	Similar to AWS but with more focus on enterprise setups.	High operational overhead for managing hardware, software, and IT staff.
Initial Investment	No upfront capital costs; **pay-as-you-go** model ensures cost efficiency during scaling.	Similar to GCP, but higher costs in the long run for equivalent services.	Similar to GCP but can be more expensive for hybrid or specific analytics workloads.	High upfront investment for hardware, setup, and IT staff; long lead times to scale or upgrade.
Suitability for the Project	Best fit due to managed services, cost-efficiency, and robust big data tools like BigQuery and Firestore.	Viable alternative but requires more manual configurations, increasing time and complexity.	Less suitable due to lack of seamless big data solutions and higher complexity.	Least suitable due to high costs, limited scalability, and operational challenges.

10. Potential Future Enhancements

- **Machine Learning Price Prediction**: Use Vertex AI to forecast trends and suggest the best purchase times.
- **Product Recommendations**: Suggest similar or alternative products based on price trends.

11. Conclusion

The updated system utilizes **provided URLs for popular product sections**, **paid APIs for reliable scraping**, and GCP's serverless architecture for scalability and cost-efficiency. This project is designed to provide users with robust, insightful, and user-friendly price tracking by addressing challenges like anti-scraping measures and integrating future enhancements.

12. References

1. https://firebase.google.com/docs/hosting
2. https://firebase.google.com/docs/hosting/custom-domain
3. https://firebase.google.com/docs/firestore
4. https://firebase.google.com/docs/firestore/security/get-started
5. https://firebase.google.com/docs/firestore/query-data/index-overview
6. https://cloud.google.com/functions/docs
7. https://firebase.google.com/docs/functions
8. https://cloud.google.com/functions/docs/concepts/events-triggers
9. https://cloud.google.com/storage/docs
10. https://firebase.google.com/docs/storage
11. https://cloud.google.com/storage/docs/access-control
12. https://firebase.google.com/docs/firestore/extend-with-bigquery
13. https://cloud.google.com/bigquery/docs
14. https://cloud.google.com/bigquery/docs/reference/standard-sql/query-syntax
15. https://firebase.google.com/docs/cli
16. https://cloud.google.com/products/calculator
17. https://firebase.google.com/pricing
18. https://cloud.google.com/bigquery/pricing
19. https://cloud.google.com/storage/docs/storage-classes
20. https://firebase.google.com/docs/admin/setup
21. https://cloud.google.com/storage/docs/reference/libraries
22. https://cloud.google.com/bigquery/docs/reference/libraries
23. https://cloud.google.com/free
24. https://firebase.google.com/docs
25. https://cloud.google.com/support
26. https://github.com/luminati-io
27. https://github.com/pkellz/bright-data-server
28. https://github.com/luminati-io/Awesome-Web-Scraping
29. https://github.com/yashpatel7025/PriceTracker
30. https://github.com/egemenberk/PriceGetter
31. https://github.com/Evasionn/price-tracker
32. https://github.com/akaszynski/keepa
33. https://github.com/kcoder666/keepaAPI

13. Repo

- https://github.com/shazzad-sourav/CloudTech.git