**UNIVERSITY OF ENGINEERING & TECHNOLOGY MARDAN**

**(UETM)**

Name: [Mumtaz Ali](https://www.linkedin.com/in/mumtazali12/)

Subject: [Python Developer Skill Assignment](https://docs.google.com/document/d/1gtGCS2FzLkvPbO0h_KfH_7V5zt7BV2n7uyCVbLfCKO4/edit?tab=t.0#heading=h.bh02es41r6ml)

**Python Script for Amazon Scraping**

**Objective 🎯**

The goal of this project is to develop a Python script that scrapes product details from Amazon based on a set of search queries. The script should emphasize modularity, efficiency, and readability.

**Requirements 📄**

**1. Read Search Queries**

Input File: user\_queries.json

Structure: An array of search queries (e.g., ["headphones", "smartphones", ...]).



Total Queries: 10 queries.

**2. Scraping**

Pages to Scrape: The first 20 pages of Amazon search results for each query.

URL Format: For example, <https://www.amazon.com/s?k=headphones>.



**Extracted Details:**

**Title**

**Total reviews**

**Price**

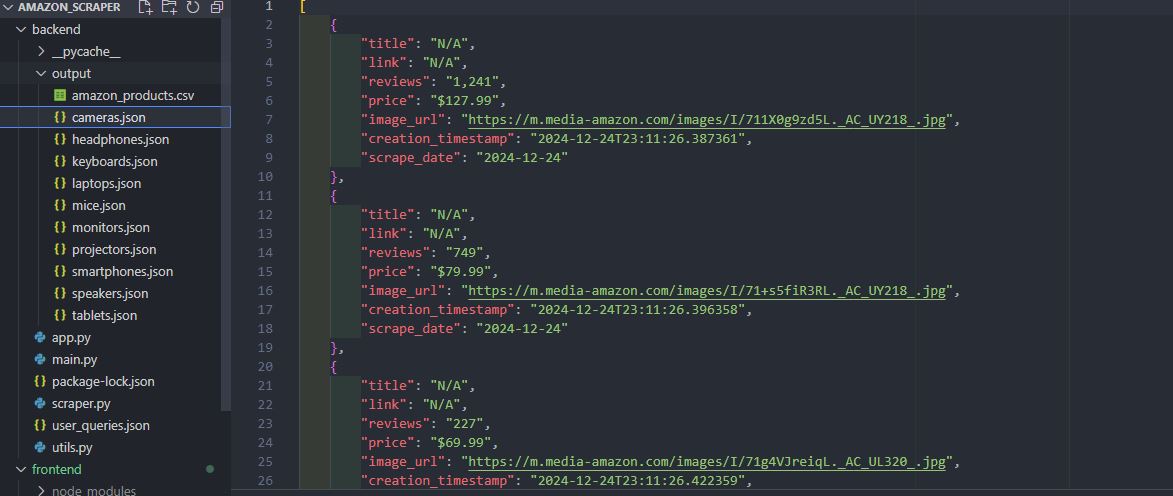
**Image URL**

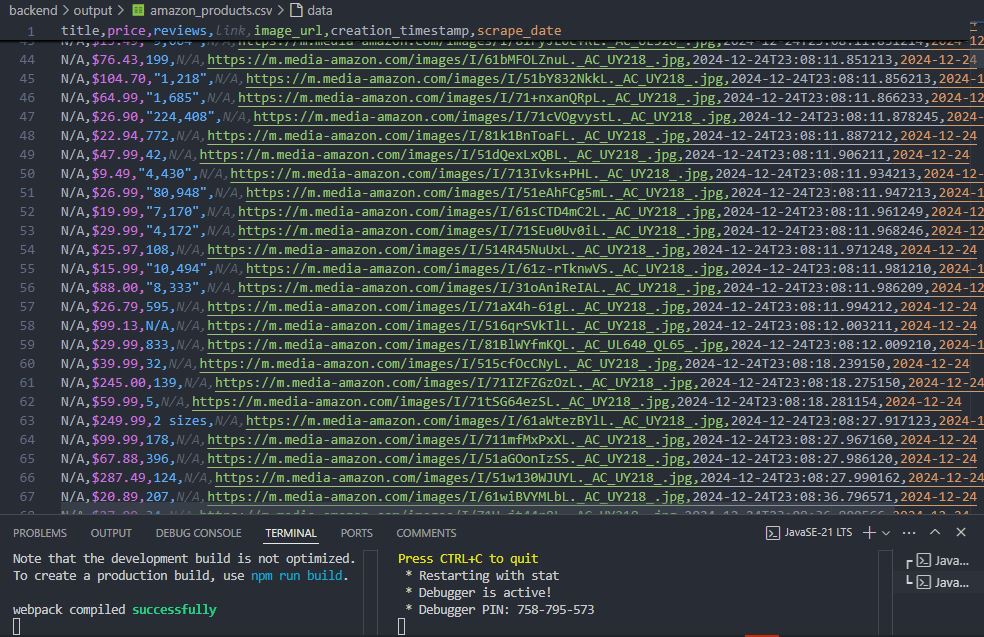
Additional fields (optional): Creation and update timestamps, scrape date.

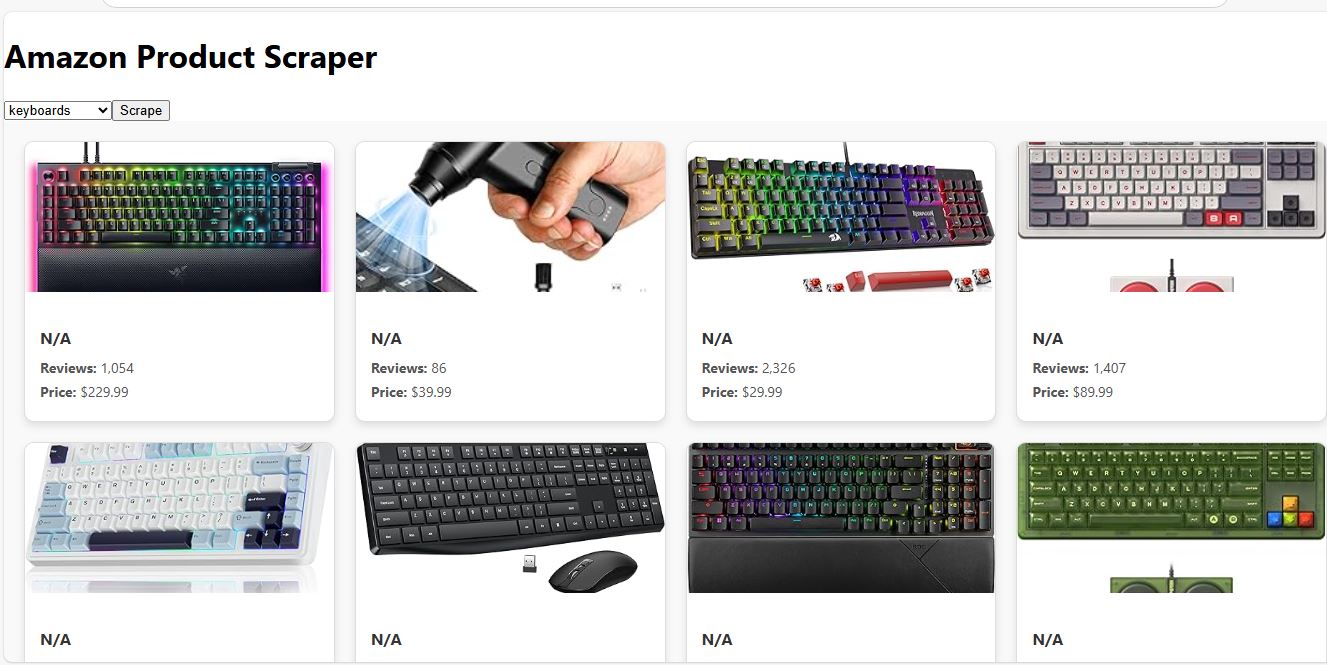
**3. Output**

File Naming: JSON files named after the search query (e.g., headphones.json).

Content Structure: Each output file should contain an array of product details.







**Suggested Structure**

Input Module: For reading input queries from user\_queries.json.

Scraping Module: A class or function dedicated to fetching and parsing web pages.

Data Model: A representation of the product, including all relevant fields.

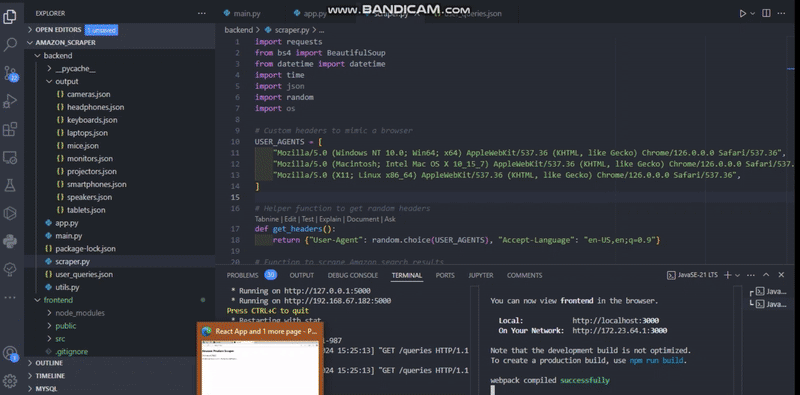
Output Module: For saving the scraped data to JSON files.

**User Interface**

**Frontend Design**

* **Technology**: React App
* **Functionality**: Display the scraped data in a tabular format.

**Deploy :**

****

<https://amazon-scraper-m2h2defbh-engrmumtazali0112s-projects.vercel.app>

**Other Questions**

**1. What do you know about Microsoft Azure and AWS? Have you ever worked with cloud technologies? If so, please describe.**

**Microsoft Azure**:

* A cloud computing platform by Microsoft offering services like computing, analytics, storage, and networking.
* Supports various programming languages and frameworks.
* Key services include Azure Virtual Machines, Azure App Services, Azure SQL Database, and Azure Functions.

**Amazon Web Services (AWS)**:

* A comprehensive cloud platform by Amazon with over 200 services.
* Offers computing (EC2), storage (S3), databases (RDS), machine learning, and more.
* Known for scalability, flexibility, and a pay-as-you-go pricing model.

**Experience with Cloud Technologies**:

* I have worked with both Azure and AWS, deploying web applications and managing databases.
* Used AWS Lambda for serverless applications and S3 for storage. In Azure, I utilized Azure Functions and Blob Storage.

**2. What is FIFO in AWS queue? What is the purpose of Queue?**

**FIFO in AWS Queue**:

* FIFO stands for "First-In-First-Out," ensuring messages are processed in the order they are sent.
* AWS offers FIFO queues through Amazon SQS, guaranteeing exactly-once delivery.

**Purpose of Queue**:

* Queues decouple components of a distributed system, allowing asynchronous communication.
* They manage workloads by storing messages until they can be processed, improving scalability and reliability.

**3. Have you ever set up a CI/CD pipeline? Do you know how to deploy Lambda functions/Create them?**

**Setting Up CI/CD Pipeline**:

* Yes, I have set up CI/CD pipelines using AWS CodePipeline, Jenkins, and GitHub Actions.
* A typical pipeline includes stages like code commit, build, test, and deployment.

**Deploying Lambda Functions**:

* Lambda functions can be deployed via:
  + **AWS Management Console**: Manually create functions.
  + **AWS CLI**: Deploy functions using command-line commands.
  + **Infrastructure as Code**: Use CloudFormation or Terraform for deployment.
  + **Serverless Framework**: Simplifies deployment management.

**Creating Lambda Functions**:

* Define the function code and runtime (e.g., Python, Node.js).
* Configure triggers (like API Gateway) and permissions (IAM roles) for execution.

🚀 Excited to Showcase My Latest Project: Amazon Product Scraper 🛍️

**#Task** :https://docs.google.com/document/d/1gtGCS2FzLkvPbO0h\_KfH\_7V5zt7BV2n7uyCVbLfCKO4/edit?usp=sharing

**#Ducoments**

I am thrilled to share the culmination of my work as a Python Developer, where I developed a full-stack application for scraping and displaying product details from Amazon in real-time. This project demonstrates my expertise in Python, web scraping, and full-stack development, coupled with the power of cloud deployment. 💻

🌟 Project Highlights:

**#Backend**:

Built using Python and Flask, designed with modularity and efficiency.

Utilized BeautifulSoup for web scraping to fetch product details like:

📌 Product Name

⭐ Reviews

💰 Price

🖼️ Images

Data stored in JSON and CSV formats for versatility.

Error-handling to manage network and parsing issues seamlessly.

**#Frontend**:

Developed with React.js to display scraped data in an intuitive tabular format.

Fully responsive UI for a smooth user experience.

Cloud Deployment:

Backend deployed using Vercel, ensuring robust performance and scalability.

Frontend also hosted on Vercel for seamless integration and global accessibility.

💡 Key Technical Insights:

Scraped 20 pages per query to ensure detailed results.

Developed CI/CD pipelines for automatic deployment.

Cloud-ready with experience in AWS and Microsoft Azure for backend management.

🌐 Explore the Project

Demo: Amazon Scraper Live Demo

GitHub Repository: View Source Code

📚 Key Takeaways:

This project helped me solidify my skills in:

Full-stack development

Cloud deployment on Vercel

Real-time data scraping

Clean code practices following the DRY (Don’t Repeat Yourself) principle

💼 Skills & Tools:

Languages: Python, JavaScript (React.js)

Libraries/Frameworks: Flask, BeautifulSoup

Cloud Platforms: AWS, Vercel

Database: JSON, CSV

Version Control: Git/GitHub

✨ Why It Matters:

This project is not just about scraping data—it's about solving problems and providing actionable insights from e-commerce data. It highlights how technology can bridge gaps and empower decision-making.

**#Python** **#ReactJS** **#WebScraping** **#AWS** **#FullStackDevelopment** **#CloudComputing** **#DataDriven** **#Coding** **#PythonProjects** **#Flask** **#Vercel** **#React** **#Ecommerce** **#DataAnalysis** **#APIs** **#DataVisualization** **#OpenSource** **#Tech** **#Programming** **#Innovation** **#SoftwareDevelopment**