Sean Umali 24051763

Joshua Rosas 23739594

Ayan Das 24073854

Project Overview/Proposal: To develop a platform that aggregates, cleans and analyzes and visually presents product pricing data from various e-commerce websites, enabling users to make informed purchasing decisions through interactive data visualizations.

Detailed Timeline and Milestone:

Week 1: Setup and Data collection

- **Days 1-2:** Project setup, including environment setup (Python, Jupyter Notebook, and necessary libraries such as Beautiful Soup for webscraping)
- **Days 3-4:** Develop web scraping scripts for the first e-commerce sites (e.g. Target, Walmart, etc.), focus on key categories to ensure depth before breadth in data collection.
- **Day 5-7:** Start scraping the second e-commerce site (e.g. shopify, etsy, etc.), aiming to gather a diverse dataset from multiple retailers.

Week 2: Data cleaning and initial analysis

- **Days 8-10:** Begin data cleaning process focusing on handling missing values, removing duplicates, and standardizing data formats using python scripts.
- **Days 11-14:** Perform the preliminary data analysis to identify basic pricing trends and outliers. Start conceptualizing the visualization designs.

Week 3: Advanced analysis and visualization development

- **Days 15-17:** Deepen Data analysis with more complex statistical methods or introductory machine learning models to detect pricing patterns or predict trends.
- **Days 18-21:** Select and begin using a visualization library (matplotlib for simplicity, plotly for interactivity, seaborn for creativity) → D3.js will be utilized in the later phase of the development to develop the visualizations (bar charts (horizontal and vertical), scatter plots, line charts, heatmaps, histograms etc.)

Week 4: Development sprint for backend and frontend

- **Days 22-24:** Start development of the frontend with Next.js for interactivity and user engagement. Design the basic UI/UX of the website where the data will be rendered.
- **Days 25-27:** Develop a backend API using Express (or flask/FastApi depending on what may be needed) to serve the preprocessed data to the frontend, additionally need to take into consideration performance (request and response timing should not take too long and if in the case that it does, pre-rendering/loading animation should be implemented)

- **Days 28-30:** Integrate the frontend with the backend, ensuring data is correctly fetched and displayed through the visualization. All the functionalities needs to be completed in regards to implementation.

Week 5: Testing, Refinement, Deployment

- **Day 31-33:** Perform comprehensive testing (unit testing, integration testing and ensuring all the necessary features of the website work as intended), ensuring reliability, usability and scalability (leaving room for stretch features to be implemented in the future)
- **Day 34-35:** Deploy the frontend on Vercel and backend on AWS.

Technical and Data Pipeline Details

Data collection and processing:

- **Collection:** Use beautiful soup for webscraping. Aim to automate and schedule scraping to refresh the dataset periodically
- **Cleaning and processing:** python (pandas) for data manipulation. Incorporate basic cleaning techniques initially, with potential to use more sophisticated methods as needed.

Data analysis:

- **Tools:** Python (scipy, numpy) for statistical analysis and basic machine learning models if within time remains.
- **Integration:** Explore using technologies like LastmileAi (contains collection of LLMS models) for advanced analytics and predictive modeling

Visualization and user interface:

- **Visualization:** Matplotlib/plotly/seaborn for generating interactive charts. Consider plotly dash for web-based interactive dashboards
- **Frontend:** Next.js for building the user interface, with node fetch API to fetch data from the backend

Backend:

- **Framework:** Express or FastAPI or Flask
- **Data management:** Either postgresql or MongoDB

Potential References/Technologies that is being considered for LLMS model:

- https://www.llamaindex.ai/open-source
- https://lastmileai.dev/
- https://modal.com/
- https://python.langchain.com/docs/integrations/llms/modal