

Project Idea: Price Tracker

Project Name: The Price is Right... Enterprice... Check it Out... The Bazaar...

Team Members: Nevan Naug, Anna Maximova, Ashdika Siddiquee, Dilpreet Bansi, Samiul Hossain
Team 2

Vision Statement

Description and Scope of the Project

This project is a website that gathers product pricing data from various sources, tracking price changes over time. The goal is to keep consumers informed and hold companies accountable for monopolies or deceptive pricing. If APIs are unavailable, AI-driven web scraping will collect the data and store it in a database.

Users

- **Consumers:** Shoppers comparing prices to make informed purchases.
- **Retailers:** Businesses tracking competition and pricing trends.
- **Distributors:** Bulk buyers optimizing their pricing strategies.
- **Manufacturers:** Producers evaluating market conditions for better pricing.

Purpose of the Project

- Promote fair competition by exposing price manipulation.
- Help consumers compare prices and avoid overpriced products.
- Implement a product-quality rating system for informed choices.
- Detect and highlight fake deals where prices are artificially inflated before sales.
- Give new businesses insights into market trends and price movements.

What Differentiates Our Product

Most price trackers focus on electronics or one-time purchases. Our platform prioritizes daily essentials like groceries, where companies often increase prices due to consumer dependency. Additionally, we estimate production costs vs. selling prices to reveal corporate profit margins, showing which companies are overcharging consumers.

Why Our Product is Valuable

Large corporations exploit pricing strategies to maximize profits, leaving consumers and small businesses at a disadvantage. By making pricing data public, we level the playing field and ensure transparency, competition, and accountability in the market.

Success Criteria

- Gather and aggregate large amounts of pricing data efficiently.
- Display data in a clear, user-friendly format with filtering and sorting.
- Provide tools for searching, comparing, and analyzing price trends.

Big user stories

- **View Each Store and its Inventory/Prices**

(In these user stories, users can see available products and their prices at different stores)

- Use APIs and scrub websites to gather data on product prices
- Allow users to submit data on products and prices.
- Verify user inputs or require them to add a link/proof.
- Track prices of Consumer/Customer Goods.
- Track prices of Manufactured Goods/Products being sold.

- **Compare prices/availability from different stores on the same Items**

(In these user stories, users can compare prices across stores to find the best deals)

- Compare product prices from different stores.
- Compare profit margins on each item.
- Compare prices of the same product in different stores.
- Compare profits gained from products Year over Year.
- Sort by lowest to highest, percentage of change, date, name, etc.

- **Create a grocery list and select a store with the cheapest overall total**

(In these user stories, users can track and analyze price trends to make cost-effective shopping decisions)

- Visualize and display data in easy-to-digest forms such as graphs.
- Different types of graphs
- Graph prices and changes over a Monthly, Yearly, 5-Year, etc., period.
- Add products to a watchlist to track
- Highlight the highest gains or price drops
- Spotlight on the front page

Iteration 1 detailed user stories

1. Compare product prices from different stores.

As a shopper, they would want to compare the prices of the same product across multiple stores so that they can find the best deal.

- i. Users can search for a product by name or category
- ii. Display a list of stores selling the product with their prices.
- iii. Users can see price differences between stores in a table format.
- iv. Allow filtering by store name, location, and stock availability.
- v. Data is updated periodically to ensure accuracy.

Priority: High

Estimated Time to complete task: 3 days

2. Compare profit margins on each item.

As a consumer, they want to see how much a product costs to produce vs. how much it is sold for so that they can identify overpriced items.

- i. The system calculates estimated production costs for products.
- ii. Users can see profit margin percentages for each item.
- iii. A visual indicator (e.g., "High Markup" label) highlights products with high profit margins.
- iv. Users can filter/search products by profit margin range.

Priority: Medium

Estimated Time to complete task: 4 days

3. Compare prices of the same product in different stores.

As a shopper, they would want to compare a product's price trend with different stores over time so that they can decide where and when to buy.

- i. Users can select a product and see its price history per store
- ii. Display a graph of price changes over time.
- iii. Allow filtering by store, location, and time range (e.g., last month, last year).
- iv. Notify users when a product drops in price at a specific store.

Priority: High

Estimated Time to complete task: 5 days

4. Compare profits gained from products Year over Year.

As a business owner, they would want to track profit changes on a product over different years so that they can understand market trends.

- i. Users can select a product and view its yearly profit changes.
- ii. The system calculates % increase or decrease in profit margins over time.
- iii. Graph visualization shows profit trends per year.
- iv. Users can compare multiple years side by side.

Priority: Medium

Estimated Time to complete task: 3 days

5. Sort Products by lowest to highest price, percentage of change, date, name, etc.

As a shopper, they would want to sort products by different criteria so that they can quickly find what they need.

- i. Users can sort by:
 - Price (Low → High, High → Low)
 - Percentage change in price
 - Date of last price update
 - Product name (A-Z, Z-A)
- ii. Sorting applies to search results and comparison lists.
- iii. The sorting should be fast and efficient even with large datasets.

Priority: High

Estimated Time to complete task: 2 days

Iteration 2

Revised Big User Stories:

1. **See the availability of items at different stores and their prices**
 - The user should be able to look at an item and see where it's available.
 - The user should be able to look at the price of the item at said store.
2. **Compare the prices of items at different stores**
 - The program should compare the prices of the items at different stores.
 - The compared price should be presented to the user.
3. **Create a grocery list and get the store with the cheapest total**
 - The user should be able to make a list with various items.
 - The user should be told what store would be the cheapest to go to.

Revised Small User Stories:

1. **The creation of a database system with stub information.**
 - The database should use postgresql.
 - The database should have stub data covering the majority of tests.
 - Priority: High
 - Estimated Time to complete the task: 5 days
 - Actual time spent to complete this task: 3 days
2. **Creation of a simple authentication system.**
 - The user should be able to create an account with a password.
 - The user should be able to sign in.
 - Priority: High
 - Estimated Time to complete this task: 6 days
 - Actual time spent to complete this task: 4 days
3. **Creation of list function and the ability to add products to a list**
 - The user should be able to create a list and add products to it.
 - The user should be able to access the products information from the list.
 - Priority: High
 - Estimated Time to complete this task: 1 day
 - Actual time spent to complete this task: 1 day
4. **Creation of a simple GUI for overall functions.**
 - There should be GUI for the components of the program.
 - Priority: Medium
 - Estimated Time to complete the task: 4 days
 - Actual time spent to complete this task: 5 days
5. **Creation of comparison functions.**
 - The program should be able to compare the various prices of the products.
 - Priority: High
 - Estimated Time to complete the task: 1 day
 - Actual time spent to complete this task: 1 day