University of Windsor

MASTER OF APPLIED COMPUTING-ADVANCE COMPUTING CONCEPT

Smartphone Recommendation System (Group 9)

INSTRUCTOR: DR. OLENA SYROTKINA

PREPARED BY:

Krutarth Patel

Shiv Anand Patel

Mitansh Sharma

Kaushal Mahendrakumar Patel

Smit Patel

Student	Tasks
Krutarth	• **Search** • **Filter** • Filter.java • DataLoader.java • displaySuggestions,loadSmartphoneData, searchSmartphones methods in SmartphoneRecommendationGUI.java
Shiv	• **Data validation** • **Spell checking** • CuckooHashTable.java • EditDistance.java • SpellCheckerGUI.java
Mitansh	• **Word completion** • **Statistics** • TrieWordCompletion.java • displayTrendingSearches method in SmartphoneRecommendationGUI.java
Kaushal	• **Frequency count** • **Search frequency** • Smartphone.java • searchFrequency method in SmartphoneRecommendationGUI.java
Smit	• **Page ranking** • Vocabularyloader.java • sortedEntries method in SmartphoneRecommendationGUI.java

1. Web crawler and Html parser

The task of the web crawler is to crawl for different data of mobile phones like name , price, key features, and many more data related to smartphones and this is developed by all group members from assignment 1.

Html parser is used to parse the html data into text so which is helpful to crawl all the pages and get the content.

In our GUI as user click the Open web crawler button the crawler opens in new tab in which user can input the website name to crawl also if user give wrong input then it also shows the url is wrong which is related to error handling.

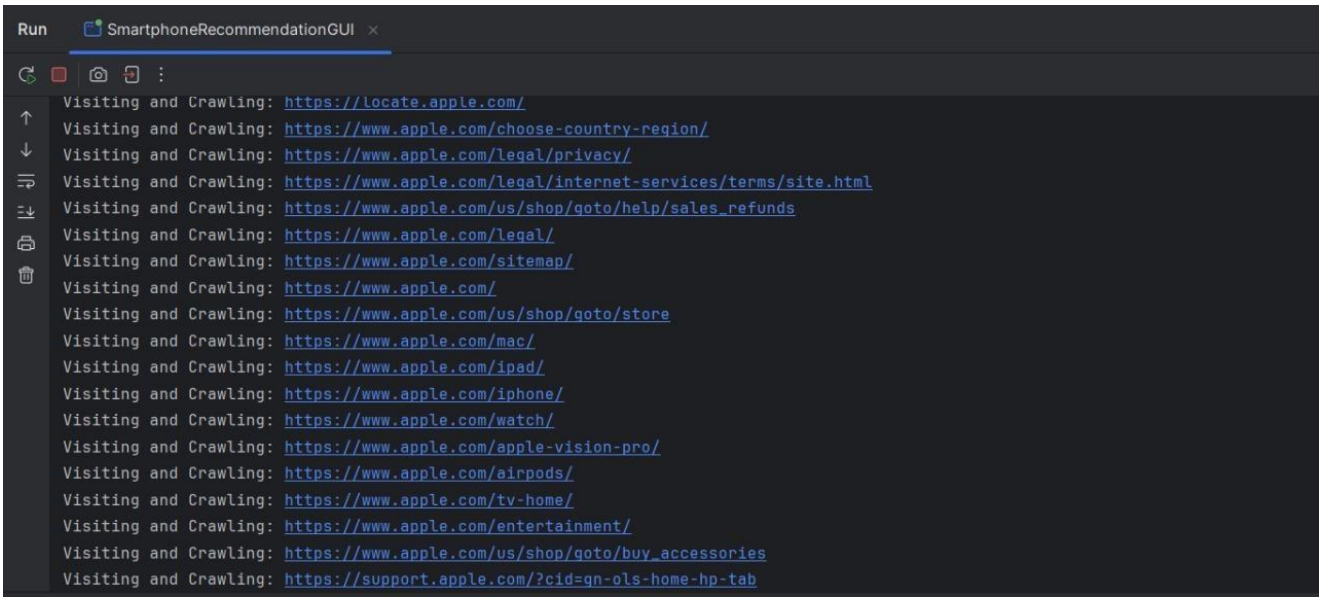
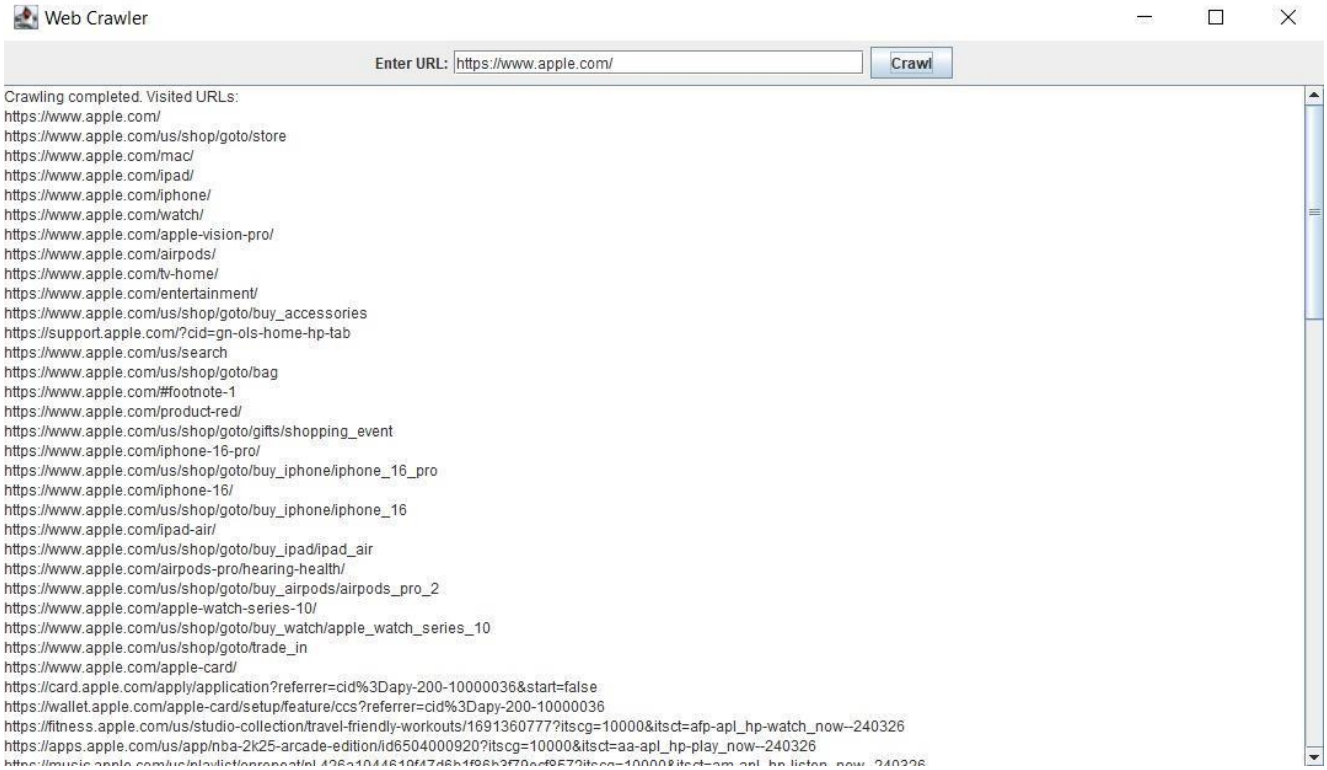


Figure 1.1 Web crawler with error handling

When the user enters the proper link for the smartphone website then the webcrawler starts crawling the website and displays the result along with parsing the data to data.csv file.


Web Crawler
Enter URL: https://www.apple.com/
Crawl
Crawling completed. Visited URLs:
https://www.apple.com/
https://www.apple.com/us/shop/goto/store
https://www.apple.com/mac/
https://www.apple.com/ipad/
https://www.apple.com/iphone/
https://www.apple.com/watch/
https://www.apple.com/apple-vision-pro/
https://www.apple.com/airpods/
https://www.apple.com/tv-home/
https://www.apple.com/entertainment/
https://www.apple.com/us/shop/goto/buy_accessories
https://support.apple.com/?cid=gn-ols-home-hp-tab
https://www.apple.com/us/search
https://www.apple.com/us/shop/goto/bag
https://www.apple.com/#footnote-1
https://www.apple.com/product-red/
https://www.apple.com/us/shop/goto/gifts/shopping_event
https://www.apple.com/iphone-16-pro/
https://www.apple.com/us/shop/goto/buy_iphone/iphone_16_pro
https://www.apple.com/iphone-16/
https://www.apple.com/us/shop/goto/buy_iphone/iphone_16
https://www.apple.com/ipad-air/
https://www.apple.com/us/shop/goto/buy_ipad/ipad_air
https://www.apple.com/airpods-pro/hearing-health/
https://www.apple.com/us/shop/goto/buy_airpods/airpods_pro_2
https://www.apple.com/apple-watch-series-10/
https://www.apple.com/us/shop/goto/buy_watch/apple_watch_series_10
https://www.apple.com/us/shop/goto/trade_in
https://www.apple.com/apple-card/
https://card.apple.com/apply/application?referrer=cid%3Dapy-200-10000036&start=false
https://wallet.apple.com/apple-card/setup/feature/ccs?referrer=cid%3Dapy-200-10000036
https://fitness.apple.com/us/studio-collection/travel-friendly-workouts/1691360777?itscg=10000&itsct=afp-apl_hp-watch_now--240326
https://apps.apple.com/us/app/nba-2k25-arcade-edition/id6504000920?itscg=10000&itsct=aa-apl_hp-play_now--240326

2. Search & Filter Functionality

The filter functionality processes a list of smartphones and narrows it down based on user-defined criteria such as brand, price range, features, and storage options.

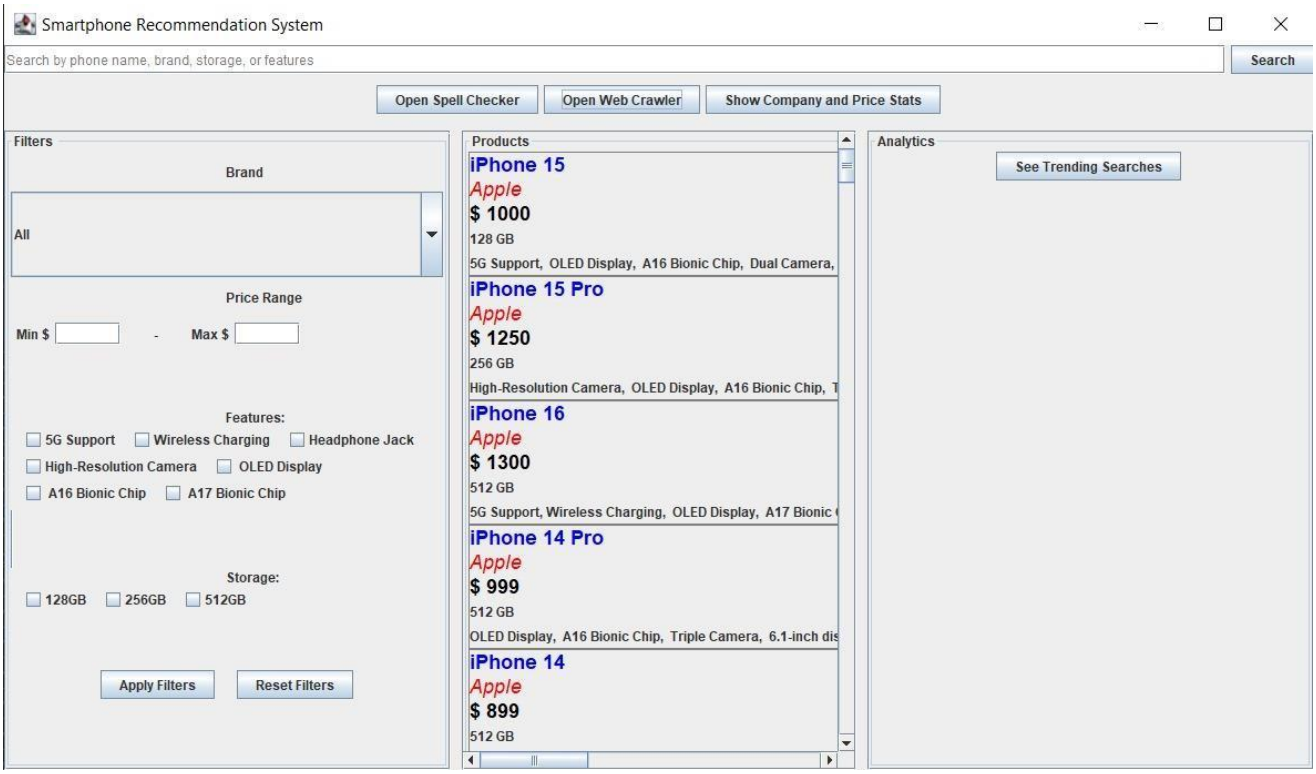
Brand Filtering: Compares the smartphone's brand to the selected brand. If "All" is selected, skips the brand filter to include all brands.

Price Range Filtering: Ensures the smartphone's price is between the specified minimum and maximum price values.

Features Filtering: Checks if the smartphone has all the features specified in required Features. Uses the contains All method for matching.

Storage Filtering: Matches the smartphone's storage capacity to the selected options. Uses the contains method to check if the smartphone's storage is in the list of selected storage options.

The implementation uses the List data structure to hold smartphones and their attributes. The Filter logic applies multiple filtering conditions efficiently, using containment checking (contains All and contains) for features and storage validation, and comparison for price and brand.



3. Spell Checker and Data Validation

We have used Cuckoo hashing for Spell Checker

Storing Valid Words: Created a cuckoo hash table to store valid words such as smartphone brands, features, and storage options. This table helps quickly check if the user input matches any of the valid options.

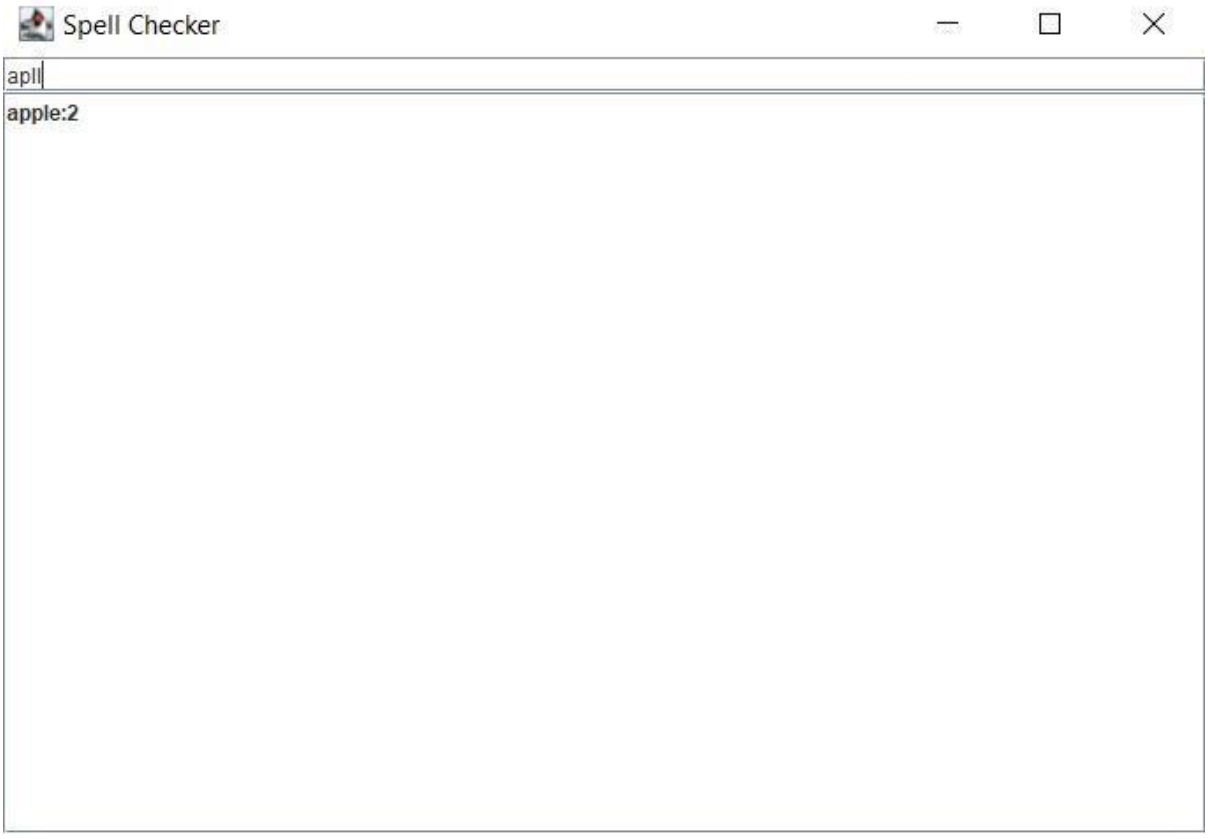
Two Hash Functions: Implemented two hash functions (hash1 and hash2). Each word is hashed into one of two possible positions in the table. If the first position is already taken, the word that is there gets moved to the second position and so on.

User Input Checking: When a user types a search query, the input is checked against the hash table. I hash each word using both hash functions to check if the word exists in either of the two positions in the table.

If a word is not found in the hash table, it's flagged as a potential misspelling. I then prompt the user to provide suggestions based on Edit Distance.

Used the Edit distance algorithm to calculate and give the suggestions. This helps identify words that are close to valid entries.

Suggestions for Misspellings: When a user types a misspelled word, the algorithm suggests corrections based on words in the hash table with a small edit distance.



Data Validation

Validating User Input: Used the cuckoo hash table to validate user inputs, like checking if a selected brand or feature exists in the predefined list. If it doesn't match any entry, it's flagged as invalid.

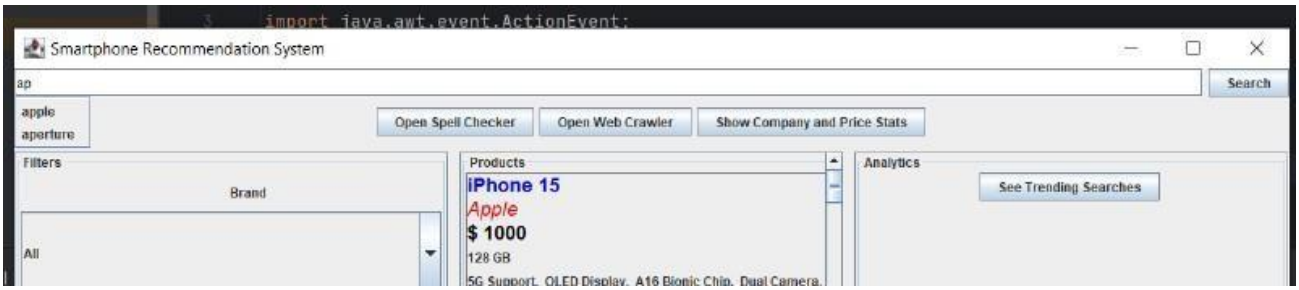
4. Word Completion

We have used **Trie-based** Word Completion.

Storing Words in a Trie: A Trie is used to store valid words such as smartphone brands, features, and storage options. Each character in a word is stored as a node, forming a tree structure.

Finding Words by Prefix: When a user enters a prefix, the Trie is searched to locate the node corresponding to the last character of the prefix. All valid words starting from that node are collected.

Recursive Search: From the prefix node, the algorithm explores all possible paths to gather complete words. This ensures all words starting with the prefix are found.

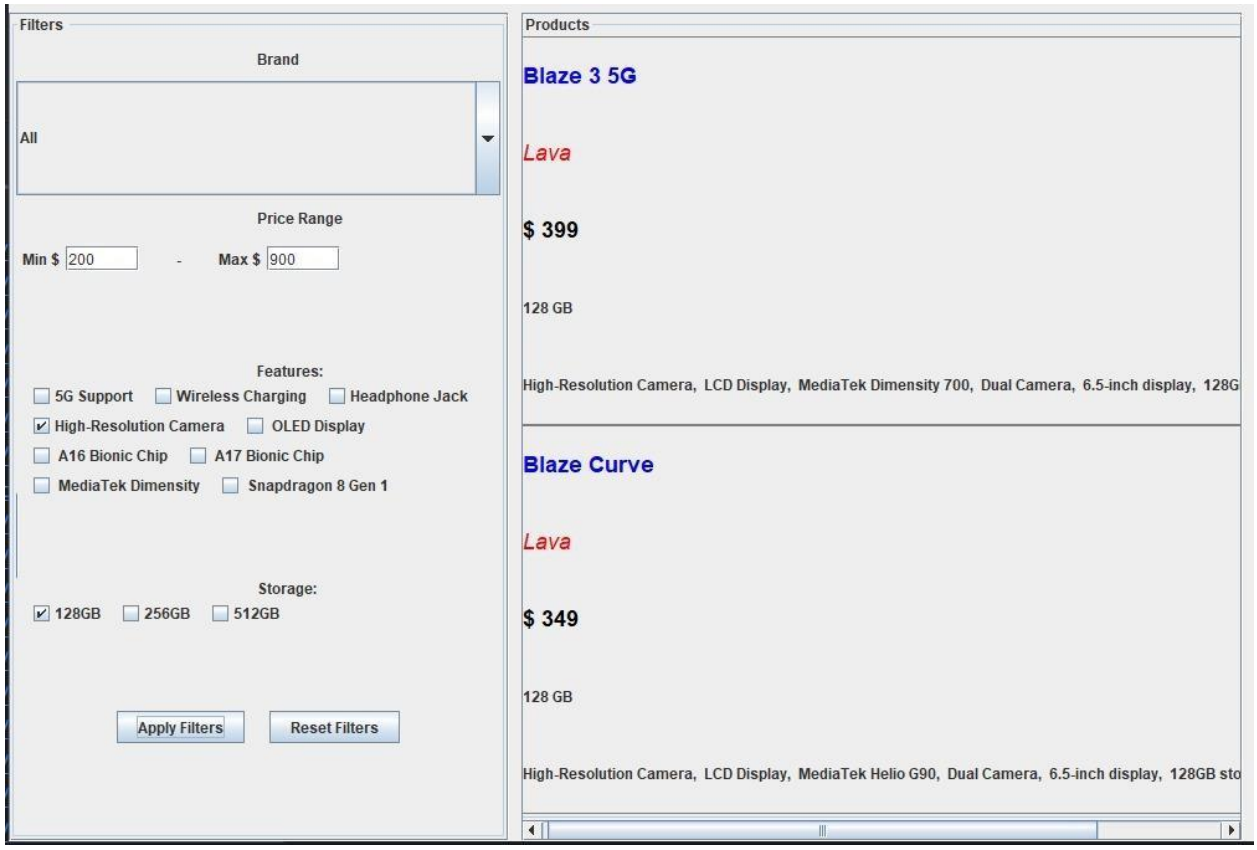


User Input Handling: Users type a prefix, and the program searches the Trie to find matching words. If no words are found, it informs the user.

Efficient Searches: The Trie structure allows fast lookups, even for large word sets, making it ideal for word completion.

Real-Time Suggestions: The program displays all words starting with the given prefix along with their count, helping users find what they're looking for quickly.

Smartphone Recommendation System



5. Statistics

When the statistics button is clicked, a new window titled "Company and Price Statistics" is displayed.

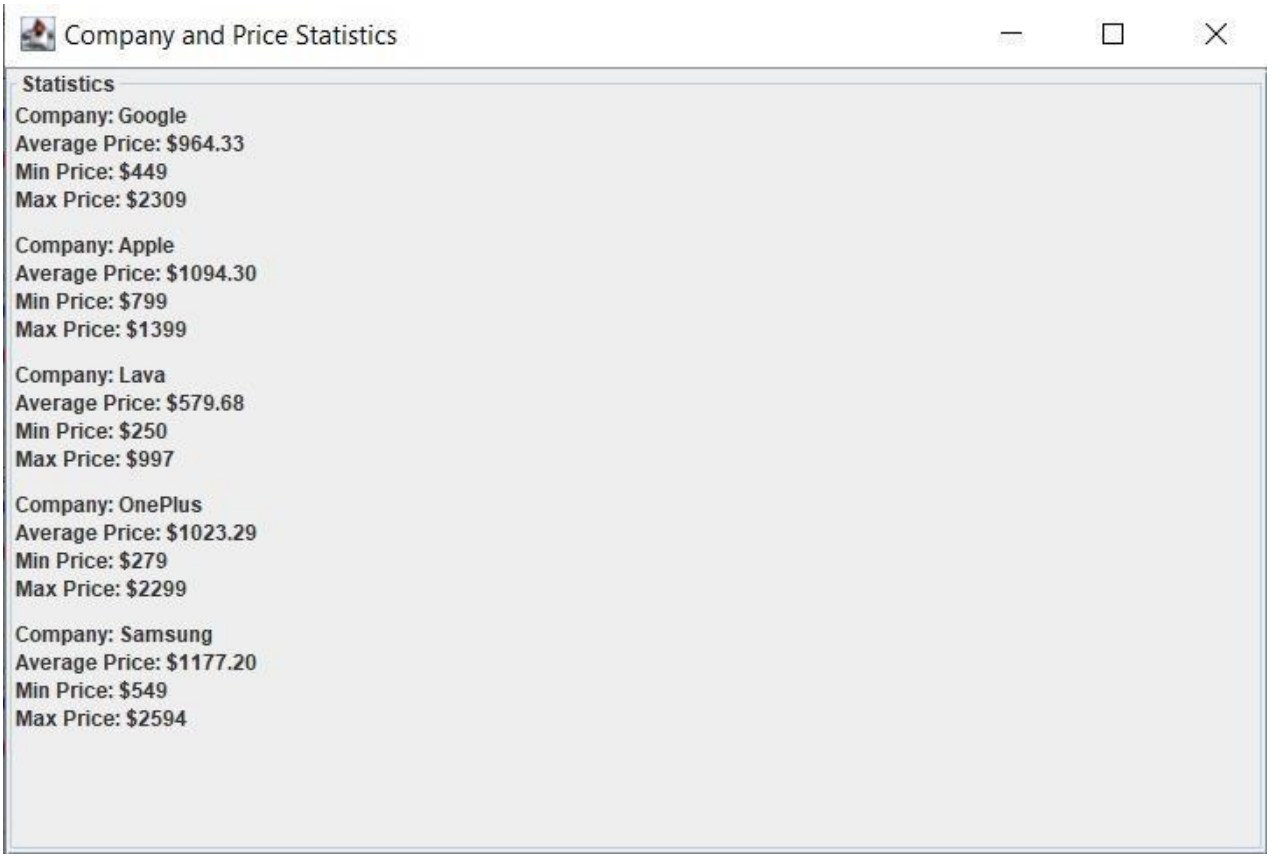
A HashMap groups smartphone prices by brand. Each brand is a key, and its value is a list of prices for that brand.

Average Price: The mean price of all smartphones for the brand.

Minimum Price: The lowest price among the brand's smartphones.

Maximum Price: The highest price for the brand.

A vertical layout displays the statistics. Each brand's name, average price, minimum price, and maximum price.



6. Frequency Count and Search Frequency

Frequency Count: Tracks how many times each word or phrase appears in user searches. This helps identify popular keywords or frequently searched terms.

Search Frequency Storage: A data structure (e.g., HashMap) stores words as keys and their search frequencies as values. Each time a user searches for a word, its count is increased.

Real-Time Updates: The frequency count is updated dynamically as users enter new search queries. This ensures accurate tracking of search trends over time.

Search Suggestion Integration: Frequently searched terms can be highlighted in autocomplete suggestions, prioritizing the most popular searches.

Benefits: Helps refine search optimization by focusing on popular terms. Provides insights into user behavior, enabling better personalization and decision-makin

7. Ranking

Purpose of Ranking: Ranking determines the order in which results are displayed based on their relevance or importance to the user's query.

Ranking Factors: Results can be ranked based on various criteria, such as:

- **Search Frequency**: Popular words or items (those searched more often) appear higher.
- **Relevance**: Matches closer to the user's query or preferences are prioritized.

Dynamic Ranking: Ranking is updated dynamically as new data (e.g., search frequency or user interactions) is collected.

Integration with Suggestions: Frequently searched and highly ranked items can appear at the top of autocomplete or suggestion lists.

Benefits of Ranking: Ensures the most relevant and valuable results are presented. Increases the efficiency and effectiveness of the search process for users.