



ಭಾರತೀಯ ಮಾಹಿತಿ ತಂತ್ರಜ್ಞಾನ ಸಂಸ್ಥೆ ರಾಯಚೂರು
भारतीय सूचना प्रौद्योगिकी संस्थान रायचूर
Indian Institute of Information Technology Raichur

COMPUTER SCIENCE AND ENGINEERING

Deal Finder

BY

Deva Anand M (CS23B1020)

A Smaran Reddy(CS23B1011)

(Deal Finder)

OCT/2024

CONTENTS

ABSTRACT

1. INTRODUCTION

2. AIM, MOTIVATION and OBJECTIVES

3. PROPOSED METHODOLOGY

4. RESULTS

5. CONCLUSION

CONTRIBUTIONS by TEAM MEMBERS

REFERENCES

ABSTRACT

This Python project automates price comparison between two leading e-commerce platforms, Amazon and Flipkart. By allowing users to input a product name, the program scrapes the product prices from both websites and identifies the lowest price. It then displays the comparison results, highlighting the best deal, and provides a direct link to the product for easy access. This tool simplifies online shopping, saving time and ensuring cost-effective purchases.

1.INTRODUCTION

This Python project focuses on creating a user-friendly tool that simplifies price comparison between two major e-commerce platforms, Amazon and Flipkart. Users can input the name of a product, and the tool will automatically scrape the latest prices from both websites. Once the prices are retrieved, the program will compare them and display the lowest available price, accompanied by a direct link to the product. This streamlined process empowers users to make informed purchasing decisions quickly, ensuring they get the best value without the need to manually browse through multiple websites.

2.AIM, MOTIVATION, and OBJECTIVES

- Aim :- This project aims to create a Python application that efficiently scrapes product prices from Amazon and Flipkart, compares them, and presents the user with the lowest price and direct product link.

-Objective :- Online shopping has grown rapidly, and consumers often search multiple platforms for the best prices. Manually comparing prices can be time-consuming. This project seeks to ease that task by offering an automated solution, empowering users to make smarter, cost-effective purchases.

-Motivation :- To develop a tool that allows users to input a product name, retrieve real-time prices from Amazon and Flipkart, and provide the most affordable option along with a convenient purchasing link.

3.PROPOSED METHODOLOGY

3.1 Modules Used

i) User Input and Query Preparation:

The user provides a product name, which is formatted by replacing spaces with +. This prepares the query for compatibility with Amazon and Flipkart URLs.

ii) Dynamic URL Construction:

The formatted product name is inserted into dynamic URLs for Amazon and Flipkart. These URLs will fetch the search results for the desired product.

iii) Web Scraping Setup:

HTTP requests are sent using the `requests` library, with randomized user-agent headers to simulate a real browser. The `get_page(url)` function retrieves the HTML content from each platform.

iv) Extracting Price Data:

Using `BeautifulSoup`, the HTML is parsed to locate the product price. The scraper identifies platform-specific elements for price extraction from both Amazon and Flipkart.

v) Price Normalization:

Extracted prices are cleaned by removing currency symbols and commas. The `clean_price(price_str)` function converts the price string into a float for comparison.

vi) Comparing Prices:

The `track_prices(product_name)` function compares the prices from both platforms. It records the lowest price, platform name, and product URL.

vii) Displaying Results:

The lowest price, platform, and corresponding URL are displayed to the user. This ensures users can easily access the best deal.

viii) Error Handling and Logging:

Errors such as failed requests or invalid prices are logged using the `logging` module. The log helps track successes, errors, and price comparisons for debugging.

3.2 GUI:

i) Designing the User Interface (UI):

- Use `st.title()`, `st.header()`, `st.text()`, `st.text_input()` for user input fields.
- Utilize Streamlit's widget functions like `st.button()` to trigger actions (e.g., search product).
- Display images and formatted text using `st.image()`, `st.write()`, etc.

ii) Backend Logic Integration:

- Define functions to perform web scraping using libraries like `requests` and `BeautifulSoup`.
- Randomize user-agent headers to avoid getting blocked.

- Parse and extract required data (e.g., product prices) from HTML responses.

iii) Error Handling & Logging:

- Implement error handling (HTTP errors, timeouts) and use logging to track issues during requests.
- Ensure that logs are written to a file using `logging` for debugging.

iv) Display Results:

- Display price comparisons or other data in a readable format using Streamlit's `st.write()`.
- Show images and structured price details for each platform (Amazon, Flipkart).
- Highlight the lowest price and provide a link to the best deal.

v) Interactive User Input:

- Collect product names or search queries through `st.text_input()`.
- Trigger the scraping and comparison process when the user clicks a button (`st.button()`).

vi) Deployment:

- Test the app locally by running `streamlit run <app_name>.py`.
- Deploy the app using platforms like Streamlit Cloud, AWS, or Heroku for easy public access.

4.RESULTS

Deal Finder

Enter the product name

Find Deal

Deal Finder

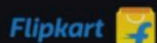
Enter the product name

iPhone 15

Find Deal



Amazon - Current price: ₹69900.0



Flipkart - Current price: ₹58999.0

Best Deal : ₹58999.0 at Flipkart

[Link to product](#)

5.CONCLUSION

This Python project effectively streamlines the price comparison process for online shopping by scraping data from Amazon and Flipkart. It empowers

users to effortlessly find the best deals by inputting a product name and receiving real-time price comparisons. Through efficient error handling and logging, the application ensures reliability, ultimately helping users make informed purchasing decisions with ease.

CONTRIBUTIONS by TEAM MEMBERS

1. Deva anand M & CS23B1020 -Report writing, code(50%),front end and back end(50%)
2. Smaran reddy & CS23B1011 - Presentation slides,code(50%) ,Front end and Back end(50%)

REFERENCES

<https://docs.streamlit.io/develop/tutorials>

<https://docs.streamlit.io/get-started>

<https://www.simplilearn.com/tutorials/python-tutorial/web-scraping-with-python>

<https://realpython.com/learning-paths/python-web-scraping/>

<https://www.scrapingbee.com/blog/web-scraping-101-with-python/>